MARKOV DECISION PROCESSES

LODEWIJK KALLENBERG

UNIVERSITY OF LEIDEN

Preface

Branching out from operations research roots of the 1950's, Markov decision processes (MDPs) have gained recognition in such diverse fields as economics, telecommunication, engineering and ecology. These applications have been accompanied by many theoretical advances. Markov decision processes, also referred to as stochastic dynamic programming or stochastic control problems, are models for sequential decision making when outcomes are uncertain. The Markov decision process model consists of decision epochs, states, actions, transition probabilities and rewards. Choosing an action in a state generates a reward and determines the state at the next decision epoch through a transition probability function. Policies or strategies are prescriptions of which action to choose under any eventuality at every future decision epoch. Decision makers seek policies which are *optimal* in some sense.

These lecture notes aim to present a unified treatment of the theoretical and algorithmic aspects of Markov decision process models. It can serve as a text for an advanced undergraduate or graduate level course in operations research, econometrics or control engineering. As a prerequisite, the reader should have some background in linear algebra, real analysis, probability, and linear programming. Throughout the text there are a lot of examples. At the end of each chapter there is a section with bibliographic notes and a section with exercises. A solution manual is available on request (e-mail to kallenberg@math.leidenuniv.nl).

Chapter 1 introduces the *Markov decision process model* as a sequential decision model with actions, transitions, rewards and policies. We illustrate these concepts with nine different *applications*: red-black gambling, how-to-serve in tennis, optimal stopping, replacement problems, maintenance and repair, production control, optimal control of queues, stochastic scheduling, and the multi-armed bandit problem.

Chapter 2 deals with the *finite horizon model* with nonstationary transitions and rewards, and the *principle of dynamic programming*: *backward induction*. We present an equivalent stationary infinite horizon model. We also study under which conditions optimal policies are *monotone*, i.e. nondecreasing or nonincreasing in the ordering of the state space.

In chapter 3 the discounted rewards over an infinite horizon are studied. This results in the *optimality equation* and methods to solve this equation: *policy iteration, linear programming, value iteration* and *modified value iteration.* Furthermore, we study under which conditions monotone optimal policies exist.

Chapter 4 discusses the total rewards over an infinite horizon under the assumption that the transition matrices are *substochastic.* We first present some background material on square matrices, eigenvalues and the spectral radius. Then, we introduce the linear program and its correspondence to policies. We derive equivalent statements for the properties that the model is a so-called *contracting* or *normalized* dynamic programming model. Next, we present the *optimality equation* and results on the computations of *optimal transient policies.* For contracting dynamic programming results and algorithms can be formulated which are similar to the results and algorithms in the discounted reward model. Special sections are devoted to finite horizon and transient MDPs, to positive, negative and convergent MDPs, and to special models as red-black gambling and the optimal stopping problem.

Chapter 5 discusses the criterion of average rewards over an infinite horizon, in the most general case. Firstly, polynomial algorithms are developed to classify MDPs as irreducible or communicating. The distinction between unichain and multichain turns out to be $\mathcal{NP}$-complete, so there is no hope of a polynomial algorithm. Then, the stationary, the fundamental and the deviation matrices are introduced, and the internal relations and properties are derived. Next, an extension of a theorem by Blackwell and the Laurent series expansion are presented. These results are fundamental to analyze the relation between discounted, average and more sensitive optimality criteria. With these results, as in the discounted case but via a more complicated analysis, the optimality equation is derived and methods to solve this equation are presented (policy iteration, linear programming and value iteration).

In chapter 6 special cases of the average reward criterion (irreducible, unichain and communicating) are considered. In all these cases the optimality equation and the methods of policy iteration, linear programming and value iteration can be simplified. Furthermore, we present the method of modified value iteration for these special cases.

Chapter 7 introduces more sensitive optimality criteria: bias optimality, n-discount and n-average optimality, and Blackwell optimality. The criteria of n-discount and n-average optimality are equivalent. We present a unifying framework, based on the Laurent series expansion, to derive *sensitive discount optimality equations.* Using a lexicographic ordering of the Laurent series, we derive the policy iteration method for n-discount optimality. In the irreducible case, one can derive a sequence of nested linear programs to compute n-discount optimal policies for any n. Also for Blackwell optimality, even in the most general case, linear programming can be applied. However, then the elements are not real numbers, but lie in a much general ordered field, namely in an ordered field of rational functions. For bias optimality, an optimal policy can be found with a three-step linear programming approach. When in addition the model is a unichain MDP, the linear programs for bias optimality can be simplified. In this unichain case, we also derive a simple policy iteration method and turnpike results. The last sections of this chapter deal with some special optimality criteria. We consider overtaking, average overtaking and cumulative overtaking optimality. A next section deals with a weighted combination of the total discounted rewards and the long-run average rewards. For this criterion an optimal policy might not exist, even when we allow nonstationary randomized policies. We present an iterative algorithm for computing an ε-optimal nonstationary policy with a simple structure. Finally, we study an optimality criterion which is the sum of expected total discounted rewards with different one-step rewards and discount factors. It turns out that for this criterion an optimal deterministic policy exists with a first nonstationary part and then it becomes stationary. We present an algorithm to compute such policy.

In chapter 8, six of the applications introduced in chapter 1 (replacement problems, maintenance and repair, production and inventory control, optimal control of queues, stochastic scheduling and multi-armed bandit problems) are analyzed in much more detail. In most cases theoretical and computational (algorithmic) results are presented. It turns out that in many cases polynomial algorithms exist, e.g. of order $\mathcal{O}(N^3)$, where N is the number of states. Finally, we present separableMDP problems.

Chapter 9 deals with some other topics. We start with complexity results (e.g. MDPs are $\mathcal{P}$-complete, deterministic MDPs are in $\mathcal{NC}$), additional constraints (for discounted and average rewards, and for MDPs with sum of discounted rewards and different discount factors) and multiple objectives (both for discounted MDPs as well as for average MDPs). Then, the linear program approach for average rewards is revisited. Next, we consider mean-variance tradeoffs, followed by determinstic MDPs (models in which each action determines the next state with probability 1). In the last section of this chapter semi-Markov decision problems are analyzed.

The subject of the last chapter (chapter 10) is *stochastic games*, particularly the *two-person zero-sum* stochastic game. Then, both players may choose actions from their own action sets, resulting in transitions and rewards determined by both players. Zero-sum means that the reward for player 1 has to be payed by player 2. Hence, there is a conflicting situation: player 1 wants to maximize the rewards, while player 2 tries to minimize the rewards. We discuss the *value* of the game and the concept of optimal policies for discounted, total as well as for average rewards. We also derive mathematical programming formulations and iterative methods. In some special cases we can present finite solution methods to find the value and optimal policies. In the last section before the sections with the bibliographic notes and the exercises we discuss *two-person general-sum* stochastic games in which each player has his own reward function and tries to maximize his own payoff.

For these lecture notes a lot of material, collected over the years and from various sources is used. In the bibliographic notes is referred to many books, papers and reports. I close this preface by expressing my gratitude to Arie Hordijk, who introduced me to the topic of MDPs. Furthermore, he was my supervisor and after my PhD a colleague during many years.

Lodewijk Kallenberg

Leiden, October, 2016.

Contents

Chapter 1

Introduction

1.1 The MDP model

An MDP is a model for sequential decision making under uncertainty, taking into account both the short-term outcomes of current decisions and opportunities for making decisions in the future. While the notion of an MDP may appear quite simple, it encompasses a wide range of applications and has generated a rich mathematical theory. In an MDP model one can distinguish the following seven characteristics.

1. The state space

At any time point at which a decision has to be made, the state of the system is observed by the decision maker. The set of possible states is called the state space and will be denoted by S. The state space may be finite, denumerable, compact or even more general. In a finite state space, the number of states, i.e. $|S|$, will be denoted by N.

2. The action sets

When the decision maker observes that the system is in state i, he (we will refer to the decision maker as 'he') chooses an action from a certain action set that may depend on the observed state: the action set in state i is denoted by $A(i)$. Similarly to the state space the action sets may be finite, denumerable, compact or more general.

3. The decision time points

The time intervals between the decision points may be constant or random. In the first case the model is said to be a *Markov decision process*; when the times between consecutive decision points are random the model is called a *semi-Markov decision process*.

4. The immediate rewards (or costs)

Given the state of the system and the chosen action, an immediate reward (or cost) is earned (there is no essential difference between rewards and costs, namely: maximizing rewards is equivalent to minimizing costs). These rewards may in general depend on the decision time point, the observed state and the chosen action, but not on the history of the process. The immediate reward at decision time point t for an action a in state i will be denoted by $r_i^t(a)$; if the reward is independent of the time t, we will write $r_i(a)$ instead of $r_i^t(a)$.

5. The transition probabilities

Given the state of the system and the chosen action, the state at the next decision time point is determined by a transition law. These transitions only depend on the decision time point t, the observed state i and the chosen action a and not on the history of the process. This property is called the *Markov property*. If the transitions really depend on the decision time point, the problem is said to be *nonstationary*. If the state at time t is i and action a is chosen, we denote the probability that at the next time point the system is in state j by $p_{ij}^t(a)$. If the transitions are independent of the time points, the problem is called *stationary*, and the transition probabilities are denoted by $p_{ij}(a)$.

6. The planning horizon

The process has a planning horizon, which is the result of the time points at which the system has to be controlled. This horizon may be finite, infinite or of random length.

7. The optimality criterion

The objective of a Markov decision problem (or a semi-Markov decision problem) is to determine a policy, i.e. a decision rule for each decision time point and each history (including the present state) of the process, that optimizes the performance of the system. The performance is measured by a utility function. This utility function assigns to each policy a value, given the starting state of the process. In the next section we will explain the concept of a policy in more detail and we will present several optimality criteria.

Example 1.1 *Inventory model with backlog*

An inventory has to be managed over a planning horizon of T weeks. At the beginning of each week the manager observes the inventory on hand and has to decide how many units to order. We assume that orders can be delivered instantaneously and that there is a finite inventory capacity of B units. We also assume that the demands D_t in week t, $1 \leq t \leq T$, are independent random variables that have nonnegative integer values and that the numbers $p_j(t) := \mathbb{P}\{D_t = j\}$ are known for all $j \in \mathbb{N}_0$ and for $t = 1, 2, \dots, T$. If the demand during a period exceeds the inventory on hand, the shortage is backlogged in the next period. The optimization problem is: which inventory strategy minimizes the total expected costs?

If an order is made in week t, there is a fixed cost K_t and a cost k_t for each ordered unit. If at the end of week t there is a positive inventory, then there are inventory costs of h_t per unit; when there is a shortage, there are backlogging costs of q_t per unit. The data K_t, k_t, h_t, q_t and $p_j(t)$, $j \in \mathbb{N}$, are known for all $t \in \{1, 2, \dots, T\}$.

Let i, the state of the system, be the inventory at the start of week t (shortages are modeled as negative inventory), let the number of ordered units be a and let j be the inventory at the end of week t; so j is the state of the next decision time point.

Then, the following costs are involved, where we use the notation $\delta(x) = \begin{cases} 1 & \text{if } x \geq 1; \\ 0 & \text{if } x \leq 0. \end{cases}$

$$\begin{array}{ll} \text{ordering costs:} & K_t \cdot \delta(a) + k_t \cdot a; \\ \text{inventory costs:} & h_t \cdot \delta(j) \cdot j; \\ \text{backlogging costs:} & q_t \cdot \delta(-j) \cdot (-j). \end{array}$$

This inventory problem can be modeled as a nonstationary MDP over a finite planning horizon, with a denumerable state space and finite action sets:

$$S = \{\dots, -1, 0, 1, \dots, B\};\ A(i) = \{a \geq 0 \mid 0 \leq i + a \leq B\};$$

$$p^t_{ij}(a) = \begin{cases} p_{i+a-j}(t) & j \leq i + a; \\ 0 & B \geq j > i + a; \end{cases}$$

$$r^t_i(a) = -\{K_t \cdot \delta(a) + k_t \cdot a + \textstyle\sum_{j=0}^{i+a}\ p_j(t) \cdot h_t \cdot (i + a - j) + \sum_{j=i+a+1}^{\infty}\ p_j(t) \cdot q_t \cdot (j - i - a)\}.$$

1.2 Policies and optimality criteria

1.2.1 Policies

A *policy* R is a sequence of decision rules: $R = (\pi^1, \pi^2, \dots, \pi^t, \dots)$, where π^t is the decision rule at time point t, $t = 1, 2, \dots$. The *decision rule* π^t at time point t may depend on all available information on the system until time t, i.e. on the states at the time points $1, 2, \dots, t$ and the actions at the time points $1, 2, \dots, t - 1$.

The formal definition of a policy is as follows. Consider the Cartesian product

$$S \times A := \{(i, a) \mid i \in S,\ a \in A(i)\} \tag{1.1}$$

and let H_t denote the set of the possible *histories* of the system up to time point t, i.e.

$$H_t := \{h_t = (i_1, a_1, \dots, i_{t-1}, a_{t-1}, i_t) \mid (i_k, a_k) \in S \times A,\ 1 \le k \le t-1;\ i_t \in S\}. \tag{1.2}$$

A decision rule π^t at time point t is function on $H_t \times A := \{(h_t, a_t) \mid h_t \in H_t,\ a_t \in A(i_t)\}$, which gives the probability of the action to be taken at time t, given the history h_t, i.e.

$$\pi^t_{h_t a_t} \ge 0 \text{ for every } a_t \in A(i_t) \text{ and } \sum_{a_t} \pi^t_{h_t a_t} = 1 \text{ for every } h_t \in H_t. \tag{1.3}$$

Let C denote the set of all policies. A policy is said to be *memoryless* if the decision rule π^t is independent of $(i_1, a_1, \dots, i_{t-1}, a_{t-1})$ for every $t \in \mathbb{N}$. So, for a memoryless policy, the decision rule at time t depends - with regard to the history h_t - only on the state i_t; therefore the notation $\pi^t_{i_t a_t}$ is used instead of $\pi^t_{h_t a_t}$. We call $C(M)$ the set of the memoryless policies. Memoryless policies are also called *Markov policies.*

If a policy is memoryless and the decision rules are independent of the time point t, i.e. $\pi^1 = \pi^2 = \cdots$, then the policy is called *stationary.* Hence, a stationary policy is determined by a nonnegative function π on $S \times A$ such that $\sum_a \pi_{ia} = 1$ for every $i \in S$. The stationary policy $R = (\pi, \pi, \dots)$ is denoted by π^∞ (and sometimes by π). The set of stationary policies is notated by $C(S)$.

If the decision rule π of the stationary policy π^∞ is nonrandomized, i.e. for every $i \in S$, we have $\pi_{ia} = 1$ for exactly one action a_i and consequently $\pi_{ia} = 0$ for every $a \neq a_i$, then the policy is called *deterministic.* Hence, a deterministic policy can be described by a function f on S, where $f(i)$ is the chosen action a_i, $i \in S$. A deterministic policy is denoted by f^∞ (and sometimes by f). The set of deterministic policies is notated by by $C(D)$.

A matrix $P = (p_{ij})$ is a *transition matrix* if $p_{ij} \ge 0$ for all (i,j) and $\sum_j p_{ij} = 1$ for all i. For a Markov policy $R = (\pi^1, \pi^2, \dots)$ the transition matrix $P(\pi^t)$ and the reward vector $r(\pi^t)$ are defined by

$$\begin{aligned} \{P(\pi^t)\}_{ij} &:= \sum_a p^t_{ij}(a) \cdot \pi^t_{ia} \text{ for every } i \in S,\ j \in S \text{ and } t \in \mathbb{N}; && (1.4)\\ \{r(\pi^t)\}_i &:= \sum_a r^t_i(a) \cdot \pi^t_{ia} \text{ for every } i \in S \text{ and } t \in \mathbb{N}. && (1.5) \end{aligned}$$

Take any initial distribution β defined on the state space S, i.e. β_i is the probability that the system starts in state i, and take any policy R. Then, by the theorem of Ionescu Tulcea (see e.g. Bertsekas and Shreve [21], Proposition 7.28, p.140), there exists a unique probability measure $\mathbb{P}_{\beta,R}$ on H_∞, where

$$H_\infty := \{h_\infty = (i_1, a_1, i_2, a_2, \dots) \mid (i_k, a_k) \in S \times A,\ k = 1, 2, \dots\}. \tag{1.6}$$

If $\beta_i = 1$ for some $i \in S$, then we write $\mathbb{P}_{i,R}$ instead of $\mathbb{P}_{\beta,R}$.

Let the random variables X_t and Y_t denote the state and action at time t, $t = 1, 2, \dots$. Given an initial distribution β and a policy R, by the theorem of Ionescu Tulcea, for all $j \in S$, $a \in A(j)$

the notion $\mathbb{P}_{\beta,R}\{X_t = j, Y_t = a\}$ is well-defined as the probability that at time t the state is j and the action is a. Similarly, for all $j \in S$ the notion $\mathbb{P}_{\beta,R}\{X_t = j\}$ is well-defined as the probability that at time t the state is j. Furthermore, $\mathbb{P}_{\beta,R}\{X_t = j\} = \sum_a \mathbb{P}_{\beta,R}\{X_t = j, Y_t = a\}$.

Lemma 1.1

For any Markov policy $R = (\pi^1, \pi^2, \dots)$, any initial distribution β and any $t \in \mathbb{N}$, we have

(1) $\mathbb{P}_{\beta,R}\{X_t = j, Y_t = a\} = \sum_i \beta_i \cdot \{P(\pi^1)P(\pi^2)\cdots P(\pi^{t-1})\}_{ij} \cdot \pi^t_{ja}$ *for all $(j,a) \in S \times A$, where, if $t = 1$, $P(\pi^1)P(\pi^2)\cdots P(\pi^{t-1})$ is defined as the identity matrix I.*

(2) $\mathbb{E}_{\beta,R}\{r^t_{X_t}(Y_t)\} = \sum_i \beta_i \cdot \{P(\pi^1)P(\pi^2)\cdots P(\pi^{t-1}) \cdot r(\pi^t)\}_i$.

Proof

By induction on t. For $t = 1$,

$$\mathbb{P}_{\beta,R}\{X_t = j, Y_t = a\} = \beta_j \cdot \pi^1_{ja} = \sum_i \beta_i \cdot \{P(\pi^1)P(\pi^2)\cdots P(\pi^{t-1})\}_{ij} \cdot \pi^t_{ja}$$

and

$$\mathbb{E}_{\beta,R}\{r^t_{X_t}(Y_t)\} = \sum_{i,a} \beta_i \cdot \pi^1_{ia} \cdot r^1_i(a) = \sum_i \beta_i \cdot \{P(\pi^1)P(\pi^2)\cdots P(\pi^{t-1}) \cdot r(\pi^t)\}_i.$$

Assume that the results are true for t; we show that the results also hold for $t+1$:

$$\begin{aligned}
\mathbb{P}_{\beta,R}\{X_{t+1} = j, Y_{t+1} = a\} &= \textstyle\sum_{k,b} \mathbb{P}_{\beta,R}\{X_t = k, Y_t = b\} \cdot p^t_{kj}(b) \cdot \pi^{t+1}_{ja} \\
&= \textstyle\sum_{k,b,i} \beta_i \cdot \{P(\pi^1)P(\pi^2)\cdots P(\pi^{t-1})\}_{ik} \cdot \pi^t_{kb} \cdot p^t_{kj}(b) \cdot \pi^{t+1}_{ja} \\
&= \textstyle\sum_i \beta_i \cdot \sum_k \{P(\pi^1)P(\pi^2)\cdots P(\pi^{t-1})\}_{ik} \cdot \sum_b \pi^t_{kb} \cdot p^t_{kj}(b) \cdot \pi^{t+1}_{ja} \\
&= \textstyle\sum_i \beta_i \cdot \sum_k \{P(\pi^1)P(\pi^2)\cdots P(\pi^{t-1})\}_{ik} \cdot \{P(\pi^t)\}_{kj} \cdot \pi^{t+1}_{ja} \\
&= \textstyle\sum_i \beta_i \cdot \{P(\pi^1)P(\pi^2)\cdots P(\pi^t)\}_{ij} \cdot \pi^{t+1}_{ja}.
\end{aligned}$$

Furthermore, one has

$$\begin{aligned}
\mathbb{E}_{\beta,R}\{r^{t+1}_{X_{t+1}}(Y_{t+1})\} &= \textstyle\sum_{j,a} \mathbb{P}_{\beta,R}\{X_{t+1} = j, Y_{t+1} = a\} \cdot r^{t+1}_j(a) \\
&= \textstyle\sum_{j,a,i} \beta_i \cdot \{P(\pi^1)P(\pi^2)\cdots P(\pi^t)\}_{ij} \cdot \pi^{t+1}_{ja} \cdot r^{t+1}_j(a) \\
&= \textstyle\sum_i \beta_i \cdot \sum_j \{P(\pi^1)P(\pi^2)\cdots P(\pi^t)\}_{ij} \cdot \sum_a \pi^{t+1}_{ja} \cdot r^{t+1}_j(a) \\
&= \textstyle\sum_i \beta_i \cdot \sum_j \{P(\pi^1)P(\pi^2)\cdots P(\pi^t)\}_{ij} \cdot \{r(\pi^{t+1})\}_j \\
&= \textstyle\sum_i \beta_i \cdot \{P(\pi^1)P(\pi^2)\cdots P(\pi^t) r(\pi^{t+1})\}_i.
\end{aligned}$$

□

The next theorem shows that for any initial distribution β, any sequence of policies $R_1, R_2, \dots$ and any convex combination of the marginal distributions of $\mathbb{P}_{\beta,R_k}$, $k \in \mathbb{N}$, there exists a Markov policy R_* with the same marginal distribution.

Theorem 1.1

For any initial distribution β, any sequence of policies $R_1, R_2, \dots$ and any sequence of nonnegative real numbers $p_1, p_2, \dots$ satisfying $\sum_k p_k = 1$, there exists a Markov policy R_* such that

$$\mathbb{P}_{\beta,R_*}\{X_t = j, Y_t = a\} = \sum_k p_k \cdot \mathbb{P}_{\beta,R_k}\{X_t = j, Y_t = a\} \textit{ for all } (j,a) \in S \times A, \textit{ and all } t \in \mathbb{N}. \quad (1.7)$$

Proof

Define the Markov policy $R_* = (\pi^1, \pi^2, \dots)$ by

$$\pi^t_{ja} := \frac{\sum_k p_k \cdot \mathbb{P}_{\beta,R_k}\{X_t = j, Y_t = a\}}{\sum_k p_k \cdot \mathbb{P}_{\beta,R_k}\{X_t = j\}} \text{ for all } t \in \mathbb{N} \text{ and all } (j,a) \in S \times A. \tag{1.8}$$

In case the denominator is zero, take for $\pi^t_{ja},\ a \in A(j)$ arbitrary nonnegative numbers such that $\sum_a \ \pi^t_{ja} = 1,\ j \in S$. Take any $(j,a) \in S \times A$. We prove the theorem by induction on t. For $t = 1$, we obtain $\mathbb{P}_{\beta,R_*}\{X_1 = j\} = \beta_j$ and $\sum_k \ p_k \cdot \mathbb{P}_{\beta,R_k}\{X_1 = j\} = \sum_k \ p_k \cdot \beta_j = \beta_j$. If $\beta_j = 0$, then $\mathbb{P}_{\beta,R_*}\{X_1 = j, Y_1 = a\} = \sum_k \ p_k \cdot \mathbb{P}_{\beta,R_k}\{X_1 = j, Y_1 = a\} = 0$.

If $\beta_j \neq 0$, then from (1.8) it follows that

$$\begin{aligned}
\textstyle\sum_k p_k \cdot \mathbb{P}_{\beta,R_k}\{X_1 = j, Y_1 = a\} &= \textstyle\sum_k \ p_k \cdot \mathbb{P}_{\beta,R_k}\{X_1 = j\} \cdot \pi^1_{ja} = \beta_j \cdot \pi^1_{ja} \\
&= \mathbb{P}_{\beta,R_*}\{X_1 = j, Y_1 = a\}.
\end{aligned}$$

Assume that (1.7) is true for t. We shall prove that (1.7) is also true for $t+1$.

$$\begin{aligned}
\mathbb{P}_{\beta,R_*}\{X_{t+1} = j\} &= \textstyle\sum_{l,b} \mathbb{P}_{\beta,R_*}\{X_t = l, Y_b = b\} \cdot p^t_{lj}(b) \\
&= \textstyle\sum_{l,b,k} p_k \cdot \mathbb{P}_{\beta,R_k}\{X_t = l, Y_b = b\} \cdot p^t_{lj}(b) \\
&= \textstyle\sum_k p_k \cdot \sum_{l,b} \ \mathbb{P}_{\beta,R_k}\{X_t = l, Y_b = b\} \cdot p^t_{lj}(b) \\
&= \textstyle\sum_k p_k \cdot \mathbb{P}_{\beta,R_k}\{X_{t+1} = j\}.
\end{aligned}$$

If $\mathbb{P}_{\beta,R_*}\{X_{t+1} = j\} = 0$, then $\sum_k \ p_k \cdot \mathbb{P}_{\beta,R_k}\{X_{t+1} = j\} = 0$, and consequently,

$$\mathbb{P}_{\beta,R_*}\{X_{t+1} = j, Y_{t+1} = a\} = \textstyle\sum_k \ p_k \cdot \mathbb{P}_{\beta,R_k}\{X_{t+1} = j, Y_{t+1} = a\} = 0.$$

If $\mathbb{P}_{\beta,R_*}\{X_{t+1} = j\} \neq 0$, then

$$\begin{aligned}
\mathbb{P}_{\beta,R_*}\{X_{t+1} = j, Y_{t+1} = a\} &= \mathbb{P}_{\beta,R_*}\{X_{t+1} = j\} \cdot \pi^{t+1}_{ja} = \textstyle\sum_k \ p_k \cdot \mathbb{P}_{\beta,R_k}\{X_{t+1} = j\} \cdot \pi^{t+1}_{ja} \\
&= \textstyle\sum_k p_k \cdot \mathbb{P}_{\beta,R_k}\{X_{t+1} = j\} \cdot \frac{\sum_k \ p_k \cdot \mathbb{P}_{\beta,R_k}\{X_{t+1} = j, Y_{t+1} = a\}}{\sum_k \ p_k \cdot \mathbb{P}_{\beta,R_k}\{X_{t+1} = j\}} \\
&= \textstyle\sum_k p_k \cdot \mathbb{P}_{\beta,R_k}\{X_{t+1} = j, Y_{t+1} = a\}.
\end{aligned}$$

□

Corollary 1.1

For any starting state i and any policy R, there exists a Markov policy R_* such that

$$\mathbb{P}_{i,R_*}\{X_t = j, Y_t = a\} = \mathbb{P}_{i,R}\{X_t = j, Y_t = a\} \textit{ for all } t \in \mathbb{N} \textit{ and all } (j,a) \in S \times A,$$

and

$$\mathbb{E}_{i,R_*}\{r^t_{X_t}(Y_t)\} = \mathbb{E}_{i,R}\{r^t_{X_t}(Y_t)\} \textit{ for all } t \in \mathbb{N}.$$

1.2.2 Optimality criteria

We consider the following optimality criteria:

1. Total expected reward over a finite horizon.
2. Total expected discounted reward over an infinite horizon.
3. Total expected reward over an infinite horizon.
4. Average expected reward over an infinite horizon.
5. More sensitive optimality criteria over an infinite horizon.

Assumption 1.1
In infinite horizon models we assume that the immediate rewards and the transition probabilities are stationary, and we denote these quantities by $r_i(a)$ and $p_{ij}(a)$, respectively, for all i, j and a.

Total expected reward over a finite horizon

Consider an MDP with a finite planning horizon of T periods. For any policy R and any initial state $i \in S$, the *total expected reward* over the planning horizon is defined by:

$$v_i^T(R) := \sum_{t=1}^{T} \mathbb{E}_{i,R}\{r^t_{X_t}(Y_t)\} = \sum_{t=1}^{T} \sum_{j,a} \mathbb{P}_{i,R}\{X_t = j, Y_t = a\} \cdot r_j^t(a) \text{ for all } i \in S. \tag{1.9}$$

Interchanging the summation and the expectation in (1.9) is allowed, so $v_i^T(R)$ may also be defined as the expected total reward, i.e.

$$v_i^T(R) := \mathbb{E}_{i,R}\left\{\sum_{t=1}^{T} r^t_{X_t}(Y_t)\right\} \text{ for all } i \in S.$$

Let

$$v_i^T := sup_{R \in C}\ v_i^T(R) \text{ for all } i \in S, \tag{1.10}$$

or in vector notation, $v^T = sup_{R \in C}\ v^T(R)$. The vector v^T is called the *value vector*. From Corollary 1.1 and Lemma 1.1, it follows that

$$v^T = sup_{R \in C(M)}\ v^T(R) \tag{1.11}$$

and

$$v^T(R) = \sum_{t=1}^{T}\ P(\pi^1)P(\pi^2)\cdots P(\pi^{t-1}) \cdot r(\pi^t) \text{ for } R = (\pi^1, \pi^2, \cdots) \in C(M). \tag{1.12}$$

A policy R_* is called an *optimal policy* if

$$v^T(R_*) = v^T. \tag{1.13}$$

It is nontrivial that there exists an optimal policy: the supremum has to be attained and it has to be attained simultaneously for all starting states. It can be shown (see the next chapter) that an optimal Markov policy $R_* = (f_*^1, f_*^2, \cdots, f_*^T)$ exists, where f_*^t is a deterministic decision rule for $t = 1, 2, \ldots, T$.

Total expected discounted reward over an infinite horizon

Assume that an amount r earned at time point 1 is deposited in a bank with *interest rate* ρ. This amount becomes $(1+\rho)\cdot r$ at time point 2, $(1+\rho)^2\cdot r$ at time point 3, etc. Hence, for interest rate ρ, an amount r at time point 1 is comparable with $(1+\rho)^{t-1}\cdot r$ at time point $t,\ t=1,2,\dots$.

Define $\alpha := (1+\rho)^{-1}$ and call α the *discount factor*. Note that $\alpha \in (0,1)$. Then, conversely, an amount r received at time point t is considered as equivalent to the amount $\alpha^{t-1}\cdot r$ at time point 1, the so-called *discounted value.*

Hence, the reward $r_{X_t}(Y_t)$ at time point t has at time point 1 the discounted value $\alpha^{t-1}\cdot r_{X_t}(Y_t)$. The *total expected α-discounted reward*, given initial state i and policy R, is denoted by $v_i^\alpha(R)$ and defined by

$$v_i^\alpha(R) := \sum_{t=1}^{\infty} \mathbb{E}_{i,R}\{\alpha^{t-1}\cdot r_{X_t}(Y_t)\}. \tag{1.14}$$

Obviously, $v_i^\alpha(R) = \sum_{t=1}^{\infty} \alpha^{t-1} \sum_{j,a} \mathbb{P}_{i,R}\{X_t = j, Y_t = a\}\cdot r_j(a)$. Another way to consider the discounted reward is by the *expected total α-discounted reward*, i.e.

$$\mathbb{E}_{i,R}\left\{\sum_{t=1}^{\infty} \alpha^{t-1}\cdot r_{X_t}(Y_t)\right\}.$$

Since

$$\left|\sum_{t=1}^{\infty} \alpha^{t-1}\cdot r_{X_t}(Y_t)\right| \le \sum_{t=1}^{\infty} \alpha^{t-1}\cdot M = (1-\alpha)^{-1}\cdot M,$$

where $M = max_{i,a}|r_i(a)|$, the theorem of dominated convergence (e.g. Bauer [13] p. 71) implies

$$\mathbb{E}_{i,R}\left\{\sum_{t=1}^{\infty} \alpha^{t-1}\cdot r_{X_t}(Y_t)\right\} = \sum_{t=1}^{\infty} \mathbb{E}_{i,R}\{\alpha^{t-1}\cdot r_{X_t}(Y_t)\} = v_i^\alpha(R), \tag{1.15}$$

i.e. the expected total discounted reward and the total expected discounted reward criteria are equivalent.

Let $R = (\pi^1, \pi^2, \dots) \in C(M)$, then

$$v^\alpha(R) = \sum_{t=1}^{\infty} \alpha^{t-1}\cdot P(\pi^1)P(\pi^2)\cdots P(\pi^{t-1})\cdot r(\pi^t). \tag{1.16}$$

Hence, a stationary policy π^∞ satisfies

$$v^\alpha(\pi^\infty) = \sum_{t=1}^{\infty} \alpha^{t-1}P(\pi)^{t-1}r(\pi). \tag{1.17}$$

Like before, the *value vector* v^α is defined by

$$v^\alpha := sup_{R\in C}\ v^\alpha(R). \tag{1.18}$$

A policy R_* is an *optimal policy* if

$$v^\alpha(R_*) = v^\alpha. \tag{1.19}$$

In Chapter 3 we will show the existence of an optimal deterministic policy f_*^∞ for this criterion and we also will prove that the value vector v^α is the unique solution of the so-called *optimality equation*

$$x_i = max_{a \in A(i)} \Big\{ r_i(a) + \alpha \sum_j p_{ij}(a) x_j \Big\} \text{ for all } i \in S. \tag{1.20}$$

Furthermore, we will derive that f_*^∞ is an optimal policy if

$$r_i(f_*) + \alpha \sum_j p_{ij}(f_*) v_j^\alpha \geq r_i(a) + \alpha \sum_j p_{ij}(a) v_j^\alpha \text{ for all } a \in A(i) \text{ for all } i \in S. \tag{1.21}$$

Total expected reward over an infinite horizon

A logical definition of the *total expected reward* is the total expected discounted reward with discount factor $\alpha = 1$. So, given initial state i and policy R, we obtain $\sum_{t=1}^\infty \mathbb{E}_{i,R}\{r_{X_t}(Y_t)\}$. However, in general $\sum_{t=1}^\infty \mathbb{E}_{i,R}\{r_{X_t}(Y_t)\}$ may be not well-defined. Therefore, we consider this criterion under the following assumptions.

Assumption 1.2

(1) The model is *substochastic*, i.e. $\sum_j p_{ij}(a) \leq 1$ for all $(i,a) \in S \times A$.

(2) For any initial state i and any policy R, $\sum_{t=1}^\infty \mathbb{E}_{i,R}\{r_{X_t}(Y_t)\}$ is well-defined (possibly $\pm\infty$).

Under these assumptions the *total expected reward*, which we denote by $v_i(R)$ for initial state i and policy R, is well-defined by

$$v_i(R) := \sum_{t=1}^\infty \mathbb{E}_{i,R}\{r_{X_t}(Y_t)\}. \tag{1.22}$$

In this case, we also can write $v_i(R) = \sum_{t=1}^\infty \sum_{j,a} \mathbb{P}_{i,R}\{X_t = j, Y_t = a\} \cdot r_j(a)$. The *value vector*, denoted by v and the concept of an *optimal policy* are defined in the usual way:

$$v := sup_{R \in C}\ v(R). \tag{1.23}$$

A policy R_* is an *optimal policy* if

$$v(R_*) = v. \tag{1.24}$$

Under the additional assumption that every policy R is *transient*, i.e.

$$\sum_{t=1}^\infty \mathbb{P}_{i,R}\{X_t = j, Y_t = a\} < \infty \text{ for all } i, j \text{ and all } a,$$

it can be shown (cf. Kallenberg [148], chapter 3) that most properties of the discounted MDP model are valid for the total reward MDP model, taking discount factor $\alpha = 1$.

Average expected reward over an infinite horizon

In the criterion of average reward the limiting behavior of the average reward over the first T periods, i.e. $\frac{1}{T}\sum_{t=1}^T r_{X_t}(Y_t)$, is considered for $T \rightarrow \infty$. Since $\lim_{T\rightarrow\infty} \frac{1}{T}\sum_{t=1}^T r_{X_t}(Y_t)$ may not exist and interchanging limit and expectation is not allowed in general, there are four different evaluation measures which can be considered:

1. Lower limit of the average expected reward:

   $\phi_i(R) := \liminf_{T\rightarrow\infty} \frac{1}{T}\sum_{t=1}^T \mathbb{E}_{i,R}\{r_{X_t}(Y_t)\},\ i \in S$, with *value vector* $\phi := sup_{R\in C}\ \phi(R)$.

2. Upper limit of the average expected reward:

   $\overline{\phi}_i(R) := \limsup_{T\rightarrow\infty} \frac{1}{T}\sum_{t=1}^T \mathbb{E}_{i,R}\{r_{X_t}(Y_t)\},\ i \in S$, with *value vector* $\overline{\phi} := sup_{R\in C}\ \overline{\phi}(R)$.

3. Expectation of the lower limit of the average reward:

   $\psi_i(R) := \mathbb{E}_{i,R}\{\liminf_{T\rightarrow\infty} \frac{1}{T}\sum_{t=1}^T\ r_{X_t}(Y_t)\},\ i \in S$, with *value vector* $\psi := sup_{R\in C}\ \psi(R)$.

4. Expectation of the upper limit of the average reward:

   $\overline{\psi}_i(R) := \mathbb{E}_{i,R}\{\limsup_{T\rightarrow\infty} \frac{1}{T}\sum_{t=1}^T\ r_{X_t}(Y_t)\},\ i \in S$, with *value vector* $\overline{\psi} := sup_{R\in C}\ \overline{\psi}(R)$.

The next lemma shows the relation between these four criteria.

Lemma 1.2
$\psi_i(R) \leq \phi_i(R) \leq \overline{\phi}_i(R) \leq \overline{\psi}_i(R)$ for every state i and every policy R.

Proof
Take any state i and any policy R. The first inequality follow from Fatou's lemma (e.g. Bauer [13], p.126):

$$\psi_i(R) = \mathbb{E}_{i,R}\{\liminf_{T\rightarrow\infty} \frac{1}{T}\sum_{t=1}^T\ r_{X_t}(Y_t)\} \leq \liminf_{T\rightarrow\infty} \frac{1}{T}\sum_{t=1}^T \mathbb{E}_{i,R}\{r_{X_t}(Y_t)\} = \phi_i(R).$$

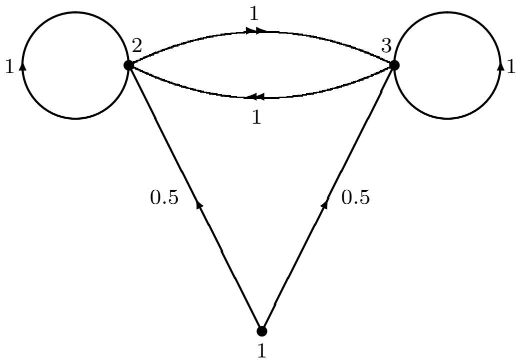
The second inequality ($\phi_i(R) \leq \overline{\phi}_i(R)$) is obvious. The third inequality is also a consequence of Fatou's lemma:

$$\overline{\phi}_i(R) = \limsup_{T\rightarrow\infty} \frac{1}{T}\sum_{t=1}^T \mathbb{E}_{i,R}\{r_{X_t}(Y_t)\} \leq \mathbb{E}_{i,R}\{\limsup_{T\rightarrow\infty} \frac{1}{T}\sum_{t=1}^T\ r_{X_t}(Y_t)\} = \overline{\psi}_i(R). \quad \square$$

We will present two examples to show that the quantities $\psi_i(R)$, $\phi_i(R)$, $\overline{\phi}_i(R)$ and $\overline{\psi}_i(R)$ may differ for some state i and some policy R. In the first example we show that $\psi_i(R) < \phi_i(R)$ and $\overline{\phi}_i(R) < \overline{\psi}_i(R)$ is possible; the second example shows that $\phi_i(R) < \overline{\phi}_i(R)$ is possible.

We use directed graphs to illustrate examples. The nodes of the graph represent the states. If the transition probability $p_{ij}(a)$ is positive there is an arc (i, j) from node i to node j; for $a = 1$ we use a simple arc, for $a = 2$ a double arc, etc.; next to the arc from node i to node j we note the transition probability $p_{ij}(a)$.

Example 1.2



Consider the following MDP:
$S = \{1, 2, 3\};\ A(1) = \{1\},\ A(2) = A(3) = \{1, 2\}.$
$p_{11}(1) = 0,\ p_{12}(1) = 0.5;\ p_{13}(1) = 0.5;\ r_1(1) = 0.$
$p_{21}(1) = 0,\ p_{22}(1) = 1;\ p_{23}(1) = 0;\ r_2(1) = 1.$
$p_{21}(2) = 0,\ p_{22}(2) = 0;\ p_{23}(2) = 1;\ r_2(2) = 1.$
$p_{31}(1) = 0,\ p_{32}(1) = 0;\ p_{33}(1) = 1;\ r_3(1) = 0.$
$p_{31}(2) = 0,\ p_{32}(2) = 1;\ p_{33}(2) = 0;\ r_3(2) = 0.$
If we start in state 1, we never return to state 1, but we will remain in the states 2 and state 3 for ever, independent the policy which is chosen.
Because of the reward structure, $\frac{1}{T}\sum_{t=1}^{T} r_{X_t}(Y_t)$ is the average number of visits to state 2 during the periods $1, 2, \dots, T$. Consider the policy $R = (\pi^1, \pi^2, \dots, \pi^t, \dots)$, where $\pi^1_{i1} := 1$ for $i = 1, 2, 3$, i.e. at time point $t = 1$ the first action is chosen in each state.
For $t \geq 2$ and history $h_t = (i_1, a_1, i_2, a_2, \dots, i_{t-1}, a_{t-1}, i_t)$, $\pi^t_{h_t a_t}$ is defined by:

$$\pi^t_{h_t a_t} := \begin{cases} \frac{k_t}{k_t+1} & \text{if } a_t = 1 \\ \frac{1}{k_t+1} & \text{if } a_t = 2 \end{cases} \quad \text{where } k_t := max\{k \geq 1 \mid i_{t-1} = i_{t-2} = \cdots = i_{t-k+1} = i_t,\ i_{t-k} \neq i_t\}.$$

So, k_t is the maximum number of periods we consecutively are in state i_t at the time points $t, t-1, \dots$. The definition of $\pi^t_{h_t a_t}$ implies that each time the system stays in the same state up to and including time point t, there is a higher probability to stay in this state for one more period. E.g. for $t = 5$ and $h_5 = (i_1, a_1, i_2, a_2, i_3, a_3, i_4, a_4, i_5) = (1, 1, 2, 2, 3, 2, 2, 2, 3)$, $k_5 = 1$, and for $t = 5$ and $h_5 = (i_1, a_1, i_2, a_2, i_3, a_3, i_4, a_4, i_5) = (1, 1, 2, 2, 3, 1, 3, 1, 3)$, $k_5 = 3$.
The probability to stay in state 2 for ever, given that we enter state 2 at time point t_0 is:

$$\mathbb{P}_R\{X_t = 2 \text{ for } t = t_0 + 1, t_0 + 2, \dots \mid X_{t_0} = 2 \text{ and } X_{t_0-1} \neq 2\} =$$
$$lim_{t\to\infty} \left\{\tfrac{1}{2} \cdot \tfrac{2}{3} \cdot \tfrac{3}{4} \cdots \tfrac{t-1}{t}\right\} = lim_{t\to\infty} \tfrac{1}{t} = 0.$$

Hence, with probability 1, a switch from state 2 to state 3 will occur at some time point; similarly, with probability 1, there is a switch from state 3 to state 2 at some time point. We even can compute the expected number of periods before such a switch occurs:

$$\mathbb{E}_R\{\text{number of consecutive stays in state } 2\} =$$
$$\sum_{k=1}^{\infty} k{\cdot}\mathbb{P}_R\{X_j = 2 \,\text{for}\; t = t_0 + 1, t_0 + 2, \dots, t_0 + k - 1;\ X_{t_0+k} \neq 2 \mid X_{t_0} = 2 \text{ and } X_{t_0-1} \neq 2\} =$$
$$\sum_{k=1}^{\infty} k \cdot \left\{\tfrac{1}{2} \cdot \tfrac{2}{3} \cdot \tfrac{3}{4} \cdots \tfrac{k-1}{k} \cdot \tfrac{1}{k+1}\right\} = \sum_{k=1}^{\infty} \tfrac{1}{k+1} = \infty.$$

So for this policy R, as long as we stay in state 2, we obtain a reward of 1 in each period. The expected number of stays in state 2 is infinite. Therefore, with probability 1, there is an infinite number of time points T at which the average reward $\frac{1}{T}\sum_{t=1}^{T} r_{X_t}(Y_t)$ is arbitrary close to 1. Similarly for state 3, with probability 1, there is an infinite number of time points T at which the average reward $\frac{1}{T}\sum_{t=1}^{T} r_{X_t}(Y_t)$ is arbitrary close to 0.

This implies for policy R that

$$\limsup_{T\to\infty} \frac{1}{T}\sum_{t=1}^{T} r_{X_t}(Y_t) = 1 \text{ with probability 1 and } \liminf{}_T \rightarrow \infty \frac{1}{T}\sum_{t=1}^{T} r_{X_t}(Y_t) = 0 \text{ with probability 1.}$$

From this we obtain

(1) $\psi_1(R) = \mathbb{E}_{1,R}\{\liminf_{T\to\infty} \frac{1}{T}\sum_{t=1}^{T} r_{X_t}(Y_t)\} = 0$;

(2) $\overline{\psi}_1(R) = \mathbb{E}_{1,R}\{\limsup_{T\to\infty} \frac{1}{T}\sum_{t=1}^{T} r_{X_t}(Y_t)\} = 1$.

If the process starts in state 1, then at any time point $t \geq 2$ - by symmetry - the probability to be in state 2 and earn 1 will be equal to the probability to be in state 3 and earn 0. So, $\mathbb{E}_{1,R}\left\{r_{X_t}(Y_t)\right\} = \frac{1}{2}$ for all $t \geq 2$. Hence,

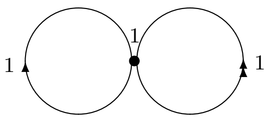
$\phi_1(R) = \liminf_{T\to\infty} \frac{1}{T}\sum_{t=1}^{T} \mathbb{E}_{i,R}\{r_{X_t}(Y_t)\} = \overline{\phi}_1(R) = \limsup_{T\to\infty} \frac{1}{T}\sum_{t=1}^{T} \mathbb{E}_{i,R}\{r_{X_t}(Y_t)\} = \frac{1}{2}$.

So, this example shows that

$$\psi_1(R) = 0 < \phi_1(R) = \frac{1}{2} = \overline{\phi}_1(R) < \overline{\psi}_1(R) = 1.$$

Example 1.3

Consider the following MDP:



$S = \{1\}$; $A(1) = \{1, 2\}$; $p_{11}(1) = p_{11}(2) = 1$; $r_1(1) = 1$, $r_1(2) = -1$.

Take the policy R that chooses action 1 at $t = 1$; action 2 at $t = 2, 3$; action 1 at $t = 4, 5, 6, 7$; action 2 at $t = 8, 9, \ldots, 15$.

In general: action 1 at $t = 2^{2k}, 2^{2k}+1, \ldots, 2^{2k}+2^{2k}-1$ for $k = 0, 1, \ldots$ and action 2 at $t = 2^{2k+1}, 2^{2k+1}+1, \ldots, 2^{2k+1}+2^{2k+1}-1$ for $k = 0, 1, \ldots$.

This gives a deterministic stream of rewards: $1; -1, -1; 1, 1, 1, 1; -1, -1, -1, -1, -1, -1, -1, -1; \ldots$ with total rewards $1;\ 0, -1;\ 0, 1, 2, 3;\ 2, 1,\ 0, -1, -2, -3, -4, -5; \ldots$.

To compute the limsup we take the time points $2^{2k-1}-1$ voor $k = 1, 2, \ldots$, i.e. the time points $1, 7, 31, 127, \ldots$; for the liminf we consider the time points $2^{2k}-1$ voor $k = 1, 2, \ldots$, i.e. the time points $3, 15, 63, 255, \ldots$.

Let T_k be the time points when we change the chosen action, i.e. $T_k = 2^k - 1$ for $k = 1, 2, \ldots$, and let A_k denote the total reward at the time points T_k, i.e. $A_1 = 1, A_2 = -1, A_3 = 3, A_4 = -5, \cdots$. It can be shown (this is left to the reader) that $|A_k| + |A_{k+1}| = 2^k$ and $|A_{k+1}| = 2 \cdot |A_k| + (-1)^k$. This implies that $|A_k| = \frac{1}{3}\{2^k - (-1)^k\}$. Since A_k is positive iff k is odd, we obtain

$$A_k = \frac{1}{3}\{(-1)^{k+1}2^k + 1\},\ k = 1, 2, \ldots.$$

Hence,

$$\phi_1(R) = liminf_{T\to\infty}\tfrac{1}{T}\textstyle\sum_{t=1}^{T} \mathbb{E}_{1,R}\{r_{X_t}(Y_t)\} = lim_{k\to\infty}\frac{A_{2k}}{2^{2k}-1} = lim_{k\to\infty}\frac{\frac{1}{3}\{-2^{2k}+1\}}{2^{2k}-1} = -\frac{1}{3}$$

and

$$\overline{\phi}_1(R) = limsup_{T\to\infty}\tfrac{1}{T}\textstyle\sum_{t=1}^{T} \mathbb{E}_{1,R}\{r_{X_t}(Y_t)\} = lim_{k\to\infty}\frac{A_{2k-1}}{2^{2k-1}-1} = lim_{k\to\infty}\frac{\frac{1}{3}\{2^{2k-1}+1\}}{2^{2k-1}-1} = +\frac{1}{3}.$$

Bierth [28] has shown that

$$\psi(\pi^\infty) = \phi(\pi^\infty) = \overline{\phi}(\pi^\infty) = \overline{\psi}(\pi^\infty) \text{ for every stationary policy } \pi^\infty \tag{1.25}$$

and that there exists a deterministic optimal policy which is optimal for all these four criteria. Hence, the four criteria are equivalent in the sense that an optimal deterministic policy for one criterion is also optimal for the other criteria.

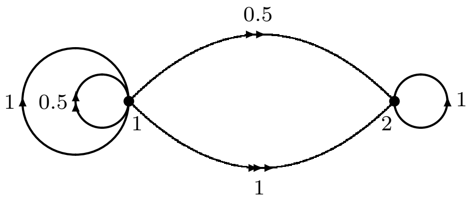
More sensitive optimality criteria over an infinite horizon

The average reward criterion has the disadvantage that it does not consider rewards earned in a finite number of periods. For example, the streams of rewards $0, 0, 0, 0, 0, \dots$ and $100, 100, 0, 0, 0, \dots$ have the same average value 0 although usually the second stream will be preferred. Hence, there is a need for criteria that select policies which are average optimal but also make the right 'early decisions' as well. There are several ways to create more sensitive criteria. One way is to consider discounting for discount factors that tend to 1. Another way is to use more subtle kinds of averaging. We present some of these criteria.

1. Bias optimality
A policy R_* is called bias optimal if $\lim_{\alpha\uparrow 1} \{v^\alpha(R_*) - v^\alpha\} = 0$.

2. Blackwell optimality
A policy R_* is Blackwell optimal if there exists an $\alpha_0 \in (0,1)$ such that $v^\alpha(R_*) = v^\alpha$ for all $\alpha \in \{\alpha_0, 1)$. From this definition it is clear that Blackwell optimality implies bias optimality. The next example shows deterministic policies f_1^∞, f_2^∞ and f_3^∞ such that f_1^∞ is average optimal but not bias-optimal, f_2^∞ is bias-optimal but not Blackwell optimal, and f_3^∞ is Blackwell optimal. Therefore, Blackwell optimality is more selective than bias-optimality which in his turn is more selective than average optimality.

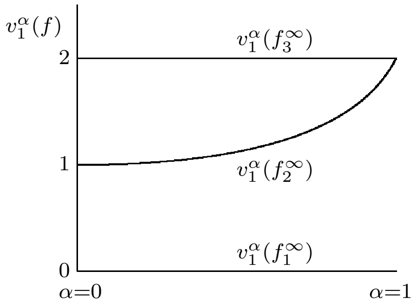
Example 1.4
Consider the following MDP:
$S = \{1,2\};\ A(1) = \{1,2,3\},\ A(2) = \{1\}$.
$p_{11}(1) = 1,\ p_{12}(1) = 0;\ p_{11}(2) =\ p_{12}(2) = 0.5;$
$p_{11}(3) = 0,\ p_{12}(3) = 1;\ p_{21}(1) = 0,\ p_{22}(1) = 1;$
$r_1(1) = 0,\ r_1(2) = 1,\ r_1(3) = 2,\ r_2(1) = 0.$



If the system is in state 2, the system stays in state 2 forever and no rewards are earned. In state 1 we have to choose between the actions 1, 2 and 3, which is denoted by the policies f_1^∞, f_2^∞ and f_3^∞, respectively. All policies have the same average reward (0 for both starting states) and the discounted reward for these policies only differ in state 1 (in state 2 the discounted reward are 0 for every discount factor α).
It is easy to see that for all α, we have $v_1^\alpha(f_1^\infty) = 0$ and $v_1^\alpha(f_3^\infty) = 2$. For the second policy, we there is an immediate reward 1 and the process stays in state 1 with probability 0.5 and moves

to state 2 with probability 0.5. Hence, we obtain $v_1^\alpha(f_2^\infty) = 1 + 0.5 \cdot \alpha \cdot v_1^\alpha(f_2^\infty) + 0.5 \cdot \alpha \cdot v_2^\alpha(f_2^\infty)$, so $v_1^\alpha(f_2^\infty) = \frac{2}{2-\alpha}$.

Alongside a picture shows these policies as function of the discount factor α. From this picture it is obvious that $v_1^\alpha = 2$ ($v^\alpha = sup_{f \in C(D)}\, v^\alpha(f)$, which is proved in the next chapter) and that the policy f_3^∞ is the only Blackwell optimal policy. Furthermore, we have $\lim_{\alpha\uparrow 1}\ \{v_1^\alpha(f_1^\infty) - v_1^\alpha\} = -2$, $\lim_{\alpha\uparrow 1}\ \{v_1^\alpha(f_2^\infty) - v_1^\alpha\} = \lim_{\alpha\uparrow 1}\ \{\frac{2}{2-\alpha} - 2\} = 0$, and $\lim_{\alpha\uparrow 1}\ \{v_1^\alpha(f_3^\infty) - v_1^\alpha\} = \lim_{\alpha\uparrow 1}\ \{2 - 2\} = 0$.
Hence, both the policies f_2^∞ and f_3^∞ are bias-optimal.



3. n-discount optimality

For $n = -1, 0, 1, \ldots$ the policy R_* is called n-discount optimal if

$$\lim_{\alpha\uparrow 1}\ (1-\alpha)^{-n}\{v^\alpha(R_*) - v^\alpha\} = 0.$$

Obviously, 0-discount optimality is the same as bias-optimality. It can be shown that that (-1)-discount optimality is equivalent to average optimality, and that Blackwell optimality is equivalent to n-discount optimality for all $n \geq |S| - 1 = N - 1$. In Chapter 7 we will shown that, for any $n = -1, 0, 1, \ldots$, an n-discount-optimal deterministic policy exists.

4. n-average optimality

Let R be any policy. For $t \in \mathbb{N}$ and $n = -1, 0, 1, \ldots$, we define the vector $v^{n,t}(R)$ inductively by

$$v^{n,t}(R) = \begin{cases} v^t(R) & \text{for } n = -1 \\ \sum_{s=1}^t v^{n-1,s}(R) & \text{for } n = 0, 1, \ldots \end{cases}$$

For $n = -1, 0, 1, \ldots$ a policy R_* is said to be n-average optimal if

$$\liminf_{T\to\infty} \frac{1}{T}\{v^{n,T}(R_*) - v^{n,T}(R)\} \geq 0 \text{ for all policies } R.$$

It can be shown that n-average optimality is equivalent to n-discount optimality. Hence, for any $n = -1, 0, 1, \ldots$, there exists an n-average optimal deterministic policy.

5. Overtaking optimality

A policy R_* is *overtaking optimal* if $\liminf_{T\to\infty}\{v^T(R_*) - v^T(R)\} \geq 0$ for all policies R. In contrast with the other criteria mentioned in this section, an overtaking optimal policy doesn't exist in general.

6. Average overtaking optimality

A policy R_* is *average overtaking optimal* if $\liminf_{T\to\infty} \frac{1}{T}\sum_{t=1}^T\{v^t(R_*) - v^t(R)\} \geq 0$ for all policies R. It is easy to verify that average overtaking optimality is equivalent to 0-average optimality and consequently to 0-discount optimality, which is bias optimality. So, also for the criterion of average overtaking optimality an optimal deterministic policy exists.

1.3 Examples

In Example 1.1 we have introduced an inventory model with backlogging. In this section we introduce other examples of MDPs: gambling, gaming, optimal stopping, replacement, maintenance and repair, production, optimal control of queues, stochastic scheduling and the so-called multi-armed bandit problem. For some of these models the optimal policy has a special structure, which we shall mention. In Chapter 8 most of these models are discussed in more detail and including the proofs of the structure of the optimal policies.

1.3.1 Red-black gambling

In the red-black gambling model a gambler with a fortune of i Euro may bet any amount $a \in \{1, 2, \dots, i\}$. He wins his amount a with probability p and he looses a Euro with probability $1 - p$. The gambler's goal is to reach a certain fortune N. The gambler continues until either he has reached his goal or he has lost all his money. The problem is to determine a policy that maximizes the probability to reach this goal.

This problem can be modeled as a substochastic MDP with the total reward criterion. At any time point t, the fortune of the gambler is considered as the state of the system. Since the gambling problem is over when the gambler has reached his goal or has lost all his money, there are no transitions when the game is in either state N or state 0. Maximizing the probability to reach the amount N is equivalent to assigning a reward 1 to state N and rewards 0 to the other states, and then maximizing the total expected reward. The MDP model for the gambling problem is formulated as follows.

$S = \{0, 1, \dots, N\};\ A(0) = A(N) = \{0\},\ A(i) = \{1, 2, \dots, min(i, N-i)\},\ 1 \leq i \leq N-1.$

For $1 \leq i \leq N-1, a \in A(i):\ p_{ij}(a) = \begin{cases} p & , j = i + a \\ 1 - p & , j = i - a \\ 0 & , j \neq i + a, i - a \end{cases}$ and $r_i(a) = 0.$

$p_{0j}(0) = p_{Nj}(0) = 0, j \in S;\ r_0(0) = 0, r_N(0) = 1.$

Since under any policy state N or state 0 is reached with probability 1, it is easy to verify that Assumption 1.2 of the total expected reward criterion over an infinite horizon is satisfied. Notice also that

$$v_i(R) = \sum_{t=1}^{\infty} \sum_{j,a} \mathbb{P}_{i,R}\{X_t = j, Y_t = a\} \cdot r_j(a) = \sum_{t=1}^{\infty} \mathbb{P}_{i,R}\{X_t = N\},$$

i.e. the total expected reward is equal to the probability to reach state N.
It can be shown that an optimal policy has the following intuitively obvious structure:
if $p > \frac{1}{2}$, then *timid play*, i.e. always bet the amount 1, is optimal;

if $p = \frac{1}{2}$, then any policy is optimal;

if $p < \frac{1}{2}$, then *bold play*, i.e. betting $min(i, N-i)$ in state i, is optimal.

1.3.2 Gaming: How to serve in tennis

The scoring in tennis is conventionally in steps from 0 to 15 to 30 to 40 to *game*. We simply use the numbers 0 through 4 for these scores. If the score reaches deuce, i.e. 40 - 40, the game is won by the player who has as first two points more than his opponent. Therefore, deuce is equivalent to 30 - 30, and similarly advantage server (receiver) is equivalent to 40 - 30 (30 - 40).

Hence, the scores can be represented by (i,j), $0 \le i,j \le 3$, excluding the pair (3,3) which is equivalent to (2,2), where i denotes the score of the server and j of the receiver. Furthermore, we use the states (4) and (5) for the case that the server or the receiver, respectively, wins the game. When the score is (i,j), the server may serve a first service ($s = 1$), or a second service ($s = 2$) in case the first serve is fault. This leads to the following 32 states:

$$\begin{cases} (i,j,s) \quad 0 \le i,j \le 3,\ (i,j) \ne (3,3),\ s = 1,2 & : \text{the states in which the game is going on} \\ (4) & : \text{the target state for the server} \\ (5) & : \text{the target state for the receiver} \end{cases}$$

For the sake of simplicity, suppose that the players can choose between two types of services: a *fast service* ($a = 1$) and a *slow service* ($a = 2$). The fast service is more likely to be fault, but also more difficult to return; the slow service is more accurate and easier to return correctly.

Let p_1 (p_2) be the probability that the fast (slow) service is good, i.e. lands in the given bounds of the court, and let q_1 (q_2) be the probability of winning the point by the server when the fast (slow) service is good. We make the following obvious assumptions: $p_1 \le p_2$ and $q_1 \ge q_2$.

Suppose the server chooses action a, where $a = 1$ or $a = 2$, for his first service. Then, the event that the server serves right and wins the point has probability p_aq_a; the event that the server serves right and loses the point has probability $p_a(1-q_a)$; the event that the server serves a fault and continues with his second service has probability $1-p_a$. For the second service, the server either wins or loses the point with probabilities p_aq_a and $1-p_aq_a$, respectively.

In the states which are no game point, i.e. $i \ne 3$ or $j \ne 3$, we have the following transition probabilities:

$$\begin{cases} p_{(i,j,1)(i+1,j,1)}(a) = p_aq_a; & p_{(i,j,2)(i+1,j,1)}(a) = p_aq_a; \\ p_{(i,j,1)(i,j+1,1)}(a) = p_a(1-q_a); & p_{(i,j,2)(i,j+1,1)}(a) = 1-p_aq_a; \\ p_{(i,j,1)(i,j,2)}(a) = 1-p_a. & \end{cases}$$

If $i = 3$, we obtain for $j = 0,1,2$:

$$\begin{cases} p_{(3,j,1)(4)}(a) = p_aq_a; & p_{(3,j,2)(4)}(a) = p_aq_a; \\ p_{(3,j,1)(3,j+1,1)}(a) = p_a(1-q_a); & p_{(3,j,2)(3,j+1,1)}(a) = 1-p_aq_a; \\ p_{(3,j,1)(3,j,2)}(a) = 1-p_a. & \end{cases}$$

Similarly, if $j = 3$, we obtain for $i = 0,1,2$:

$$\begin{cases} p_{(i,3,1)(i+1,3,1)}(a) = p_aq_a; & p_{(i,3,2)(i+1,3,1)}(a) = p_aq_a; \\ p_{(i,3,1)(5)}(a) = p_a(1-q_a); & p_{(i,3,2)(5)}(a) = 1-p_aq_a; \\ p_{(i,3,1)(i,3,2)}(a) = 1-p_a. & \end{cases}$$

Note that for $j = 2$ and $i = 2$ the states $(3, j+1, s)$ and $(i+1, 3, s)$ have to be considered as state $(2, 2, s)$ for $s = 1, 2$. When the game is over, i.e. in states (4) and (5), there are no transitions. So, this model is substochastic.

The optimization problem is: What kind of service should the server choose, given the score, in order to maximize the probability to win the game? For this aim the following reward structure is suitable. All rewards are equal to 0, except in the target state (4) of the server, in which state the reward is 1. As utility criterion the total expected reward will be used and it is easy to see that with this transition and reward structure the total expected reward equals the probability to win the game.

Let $x = \frac{p_1q_1}{p_2q_2}$. Then, x is the ratio of serving right and winning the point for the two possible actions $a = 1$ and $a = 2$. It can be shown (see exercise 4.12) that the optimal policy has the following structure in each state:

$$\begin{cases} \text{If } x \geq 1 & : \text{ always use the fast service} \\ \text{If } 1 - (p_2 - p_1) \leq x < 1 & : \text{ use the fast service as first service and the slow service as second} \\ \text{If } x < 1 - (p_2 - p_1) & : \text{ use always the slow service} \end{cases}$$

A similar problem is: Which service (the fast or the slow service) is the best to maximize the probability of winning the next point. It turns out that this problem has the same optimal policy. Hence, the optimal policy for winning the game is a *myopic policy*. Furthermore, this optimal policy is independent of the state and depends only on the data p_1, p_2, q_1 and q_2.

1.3.3 Optimal stopping

In an optimal stopping problem there are two actions for every state. The first action is stopping and the second corresponds to continue. If the stopping action 1 is chosen in state i, then a terminal reward r_i is earned and the process terminates. This termination is modeled by taking all transition probabilities equal to zero. If action 2 is chosen in state i, then a cost c_i is incurred and the probability of being in state j at the next time point is p_{ij}. Hence, the characteristics of the MDP model are:

$$\begin{aligned} &S = \{1, 2, \ldots, N\};\ A(i) = \{1, 2\} \text{ for all } i \in S; \\ &r_i(1) = r_i \text{ for all } i \in S;\ r_i(2) = -c_i \text{ for all } i \in S; \\ &p_{ij}(1) = 0 \text{ for all } i, j \in S;\ p_{ij}(2) = p_{ij} \text{ for all } i, j \in S. \end{aligned}$$

We are interested in finding an optimal *stopping policy*. A stopping policy R is a policy such that for any starting state i the process terminates in finite time with probability 1. Notice that for a stopping policy the total expected reward $v(R)$ is well-defined. As optimality criterion the total expected reward is considered.

Let v be the *value vector* of this model, i.e.

$$v_i = sup\{v_i(R) \mid R \text{ is a stopping policy}\},\ i \in S.$$

A stopping policy R_* is an *optimal policy* if $v(R_*) = v$.
Let

$$S_0 = \{i \in S \mid r_i \geq -c_i + \sum_j p_{ij} r_j\},$$

i.e. S_0 is the set of states in which immediate stopping is as least as good as continuing for one period and then choosing the stopping action. A *one-step look-head policy* is a policy which chooses the stopping action in state i if and only if $i \in S_0$. An optimal stopping problem is called *monotone* if $p_{ij} = 0$ for all $i \in S_0,\ j \notin S_0$, i.e. if S_0 is closed under P. It can be shown that in a monotone optimal stopping problem the one-step look-ahead policy is optimal.

Example 1.5 *Selling the house*
Someone wants to sell his house. He receives a price offer every week. Suppose successive offers are independent and have a value of j euros with probability p_j, for $j = 0, 1, \ldots, N$. We assume that an offer that is not immediately accepted can be accepted at any later time point. When the house remains unsold, then there are maintenance costs c during that week. What is an optimal policy for selling the house?

This problem can be modeled as an optimal stopping problem. Define the state space by $S := \{0, 1, \ldots, N\}$, where state i corresponds to the highest offer i so far. In state i there are two actions: accept the offer i (i.e. $r_i = i$) and stop, or continue with costs c and with transition probabilities

$$p_{ij} = \begin{cases} p_j & j > i; \\ 1 - \sum_{j>i} p_j & j = i; \\ 0 & j < i. \end{cases}$$

For this problem

$$S_0 = \left\{ i \in S \;\middle|\; i \geq -c + i \cdot [1 - \textstyle\sum_{j>i} p_j] + \sum_{j>i} j \cdot p_j \right\} = \left\{ i \in S \;\middle|\; c \geq \textstyle\sum_{j>i} (j-i) p_j \right\}.$$

Notice that $\sum_{j=i+1}^N (j-i)p_j = p_{i+1} + 2p_{i+2} + \cdots + (N-i)p_N$ is a monotone nonincreasing function of i. Let

$$i_* = min \left\{ i \;\middle|\; c \geq \textstyle\sum_{j>i} (j-i) p_j \right\}.$$

Then, $S_0 = \{i \in S \mid i \geq i_*\}$. Since $p_{ij} = 0,\ j < i$, the problem is monotone. An optimal policy accepts the first offer that is at least i_* (such a policy is called a *control-limit policy*). Since $\sum_{j>i}(j-i)p_j$ is the expected additional income above i in the next period, an offer is accepted if the the cost during the next week is at least the expected additional income of the offers next week. Hence, this policy has a obvious interpretation.

1.3.4 Replacement problems

Consider an item (e.g. a component of an electric system or a truck of a transportation company) that can be in one of a finite number of states, say the states $0, 1, \ldots, N$. Each state may be associated with some parameter, e.g. the age of the item. Suppose that at the beginning of each period the decision has to be made whether or not to replace the item. The motivation for replacing an item is to avoid 'bad' states with high costs.

Action 1 corresponds to replacing the item by a new one (the state of a new item is state 0 and the transition to the new item is instantaneous). For an item in state i a trade-in value s_i is received and a new item costs c.

Action 2 is to keep the item for (at least) one more period. Let p_{ij} be the probability that an item of state i is in state j at the beginning of the next period, and suppose that c_i is the maintenance cost for an item of state i during one period.
The characteristics of the MDP model are:

$$S = \{0, 1, \dots, N\};\ A(0) = \{2\},\ A(i) = \{1, 2\},\ 1 \le i \le N;$$
$$p_{ij}(1) = p_{0j},\ 1 \le i \le N,\ j \in S;\ p_{ij}(2) = p_{ij},\ 0 \le i \le N,\ j \in S;$$
$$r_i(1) = s_i - c - c_0,\ 1 \le i \le N;\ r_i(2) = -c_i,\ 0 \le i \le N.$$

Many replacement problems have an optimal *control-limit policy*, i.e. the item is replaced by a new one when the state (age) is at least a given number i_*.

1.3.5 Maintenance and repair

Consider a series system of n unreliable components, maintained by a single repairman. Each of the components may be either working or failed. The state space can be represented by a vector $x = (x_1, x_2, \dots, x_n)$, where $x_i = 1$ (working) or 0 (failed) for $i = 1, 2, \dots, n$. The system is functioning if and only if the state is $(1, 1, \dots, 1)$.

The failure time and repair time of component $i,\ 1 \le i \le n$, are exponentially distributed with rates λ_i and μ_i, respectively, and independently of the state of the other components. Notice that, by the memoryless property of the exponential distribution, the elapsed time that a working component operates or a failed component is under repair is not relevant for the description of a state.

It is assumed that the repairman may change instantaneously among failed components. That is, for example, if component i fails while component j is being repaired, the repairman may switch instantaneously from j to i, or to any other failed component.

The objective is to find a policy which assigns the repairman to a failed component in such a way that the average expected time that the system is functioning is maximized.

This problem is a finite state *continuous-time Markov decision problem.* In a continuous-time Markov decision problem any deterministic policy f^∞ generates a continuous-time Markov chain, which is a stochastic process that stays in state i for an exponential time $T_i(f)$ after which it moves to some other state j with transition probability $p_{ij}(f)$.

Let the deterministic policy f^∞ assign the repairman to component i in state x. Then, we denote this assignment by $f(x) = i$. We also use the following notation:

$$(1_k, x) := (x_1, x_2, \dots, x_{k-1}, 1, x_{k+1}, \dots, x_n);\ C_1(x) := \{i \mid x_i = 1\};\ \lambda_1(x) := \textstyle\sum_{i \in C_1(x)}\ \lambda_i;$$
$$(0_k, x) := (x_1, x_2, \dots, x_{k-1}, 0, x_{k+1}, \dots, x_n);\ C_0(x) := \{i \mid x_i = 0\}.$$

Given policy f^∞, the Markov chain remains in state x during an exponentially distributed time with rate $\lambda_1(x) + \mu_{f(x)}$. The transition probabilities of the Markov chain satisfy

$$p_{x,(1_{f(x)},x)}(f(x)) = \tfrac{\mu_{f(x)}}{\lambda_1(x)+\mu_{f(x)}};\ p_{x,(0_k,x)}(f(x)) = \tfrac{\lambda_k}{\lambda_1(x)+\mu_{f(x)}},\ k \in C_1(x);$$

$$p_{x,y)}(f(x)) = 0 \text{ for } y \neq (1_{f(x)}, x) \text{ and } y \neq (0_k, x) \text{ for some } k \in C_1(x).$$

The following results can be shown:

(1) An optimal policy can be found in the class of deterministic policies that never leave the repairman idle when there is a failed component.

(2) Maximizing the average expected time that the system is functioning, is equivalent to minimizing the time until the functioning state $(1, 1, \dots, 1)$ is reached.

(3) The optimal policy is irrespective of the repair rates $\mu_i,\ 1 \le i \le n$, and is the policy that assigns the repairman to the failed component with the smallest failure rate λ_i (*SFR policy*), i.e. the failed component with the longest expected lifetime, which is $\frac{1}{\lambda_i}$.

The results (1) and (2) are intuitively clear; however, result (3) is rather counterintuitive.

1.3.6 Production control

Consider a production process of a certain item over a planning horizon of T periods. Let the demand in period t be known and deterministic, say $D_t,\ 1 \le t \le T$. The production in period t has a capacity b_t, and let $c_t(a)$ denote the production cost for the production of a units in period $t,\ 1 \le t \le T$. In each period the demand has to be fulfilled, so shortages are not allowed and there is no backlogging. There are inventory costs $h_t(i)$ in period t, when the inventory at the end of period t is equal to $i,\ 1 \le t \le T$.

The aim is to determine the production in the various periods so as to satisfy the demands at minimum total costs.

The definition of the states is a little tricky for this model. We use a two-dimensional description, namely (i, t) to denote the situation of having i units inventory at the beginning of period t. Actions correspond to production. When in state (i, t) action a is selected, then this action has to satisfy the following three conditions:

(1) $0 \le a \le b_t$: the capacity constraint.

(2) $D_t \le i + a$: the demand requirement.

(3) $i + a \le \sum_{s=t}^T D_s$: the total production may not exceed the total demand for the remaining periods.

Hence, the MDP model for this production problem is:

$$S = \{(i, t) \mid 0 \le i \le \textstyle\sum_{s=1}^T D_s;\ 1 \le t \le T\};$$

$$A[(i, t)] = \{a \mid 0 \le a \le b_t;\ D_t \le i + a \le \textstyle\sum_{s=t}^T D_s\},\ (i, t) \in S;$$

$$p_{(i,t)(j,s}(a) = \begin{cases} 1 & \text{if } j = i + a - D_t \text{ and } s = t + 1 \\ 0 & \text{otherwise} \end{cases} \quad (i, t), (j, s) \in S,\ a \in A[(i, t)];$$

$$r_{(i,t)}(a) = -\{c_t(a) + h_t(i + a - D_t)\},\ (i, t) \in S,\ a \in A[(i, t)].$$

1.3.7 Optimal control of queues

Consider a single server queueing system where customers arrive according to a Poisson process and where the service time of a customer is exponentially distributed: the so-called $M/M/1$ queue. Suppose that the arrival and service rates can be controlled by a finite number of actions.

We say that the system is in state i when there are i customers in the system. Action a in state i means that the arrival and the service rates are $\lambda_i(a)$ and $\mu_i(a)$, respectively. Any customer has waiting cost c per time unit and when a customer enters the system a reward r is incurred.

For this model several variations can be considered, changing the assumptions about the decision time points, for example. We discuss two models, *continuous control* and *semi-Markov control*

Continuous control
In continuous control the parameters can be controlled at any time. If the system is in state i and action a is chosen, then the expectation of the interarrival time between new customers is $\frac{1}{\lambda_i(a)}$ and the expectation of the service time equals $\frac{1}{\mu_i(a)}$. It follows from the lack-of-memory property of the exponential distribution that this yields a valid stochastic decision model.

One can approach this model by *time discretization.* Then, a discrete MDP approximation scheme can be obtained by using time points $t \cdot h,\ t \in \mathbb{N}_0$, where h is a sufficiently small positive number, called the *step-size.* Sufficiently small means

$$0 < h < min_{(i,a)} \Big\{ min\Big\{ \frac{1}{\lambda_i(a)}, \frac{1}{\mu_i(a)} \Big\} \Big\},$$

in which case the so-called first order approximation of the transition probabilities is allowed. In doing so, we obtain the following MDP model:

$S = \mathbb{N}_0;\ A(i) = \{1, 2, \dots, m\};\ r_i(a) = \{r \cdot \lambda_i(a) - c \cdot i\} \cdot h,\ i \in S,\ a \in A(i);$

$$p_{ij}(a) = \begin{cases} \lambda_i(a) \cdot h & j = i+1 \\ \delta(i) \cdot \mu_i(a) \cdot h & j = i-1 \\ 1 - \lambda_i(a) \cdot h - \delta(i) \cdot \mu_i(a) \cdot h & j = i \\ 0 & \text{otherwise} \end{cases} \quad i \in S,\ a \in A(i), \text{where } \delta(i) = \begin{cases} 1 & \text{if } i \geq 1 \\ 0 & \text{if } i = 0 \end{cases}$$

Semi-Markov control
Another natural model can be obtained by using the arrival and departure times as the decision time points. Then, we have a semi-Markov decision problem in which the time until the next decision is a random variable which depends only on the current state and the chosen action. In our model the time until the next decision time point is the minimum of two negative exponential distributions, which time has also a negative exponential distribution with as rate the sum of the two negative exponential distributions. Hence, if i is the current state and action a is chosen, then this exponential distribution has parameter

$$\nu_i(a) = \lambda_i(a) + \delta(i) \cdot \mu_i(a).$$

The transition probabilities satisfy

$$p_{ij}(a) = \begin{cases} \frac{\delta(i)\mu_i(a)}{\nu_i(a)} & j = i-1 \\ \frac{\lambda_i(a)}{\nu_i(a)} & j = i+1 \\ 0 & j \neq i-1, j-1 \end{cases} \qquad i \in S,\ a \in A(i)$$

Let $r_i(a)$ be the expected reward until the next decision time point. Then,

$$r_i(a) = \frac{r \cdot \lambda_i(a) - c \cdot i}{\nu_i(a)},\ i \in S,\ a \in A(i).$$

By the technique of *uniformization*, a semi-Markov model can be transformed into an equivalent MDP with equidistant decision epochs. Define

$$\nu = max_{i,a}\ \nu_i(a)$$

and define the transition probabilities p' and the one-step reward r' for the MDP by

$$p'_{ij}(a) = \begin{cases} \frac{\nu_i(a)p_{ij}(a)}{\nu} & j \neq i \\ \frac{\nu - \nu_i(a)}{\nu} & j = i \end{cases} \quad i \in S,\ a \in A(i);\ r'_i(a) = \frac{r_i(a)\nu_i(a)}{\nu},\ i \in S,\ a \in A(i).$$

It can be shown that these models are equivalent.

1.3.8 Stochastic scheduling

In a scheduling problem, jobs have to be processed on a number of machines. Each machine can only process one job at a time. Each job i has a given processing time T_{ij} on machine j. In stochastic scheduling, these processing times are random variables. At certain time points decisions have to be made, e.g. which job is assigned to which machine. There is a utility function by which different policies can be measured, and we want to find a policy that optimizes the utility function.

There are two types of models: *customer assignment models*, in which each arriving customer has to be assigned to one of the queues and *server assignment models*, where servers have to be assigned to one of the queues of customers.

We do not explicitly present the MDP model for these general stochastic scheduling models. We confine ourselves to the formulation of some variants for which the optimal policy has a nice structure.

One server allocation to parallel queues with preemption: μc-rule

Customers arrive at a system of m parallel queues and one server. The system operates at discrete time points, i.e. arrival times and service times take values in the set $\{1, 2, \dots\}$. Furthermore, the arrival times are arbitrary and the service time T_i, for a customer in queue i, is geometrically distributed with rate μ_i, i.e.

$$\mathbb{P}\{T_i = n\} = (1-\mu_i)^{n-1} \cdot \mu_i,\ n \in \mathbb{N},\ \text{with } \mu_i \in (0,1),\ 1 \leq i \leq m.$$

Then,

$$\mathbb{E}\{T_i\} = \sum_{n=1}^{\infty} \mathbb{P}\{T_i = n\} \cdot n = \mu_i \cdot \sum_{n=1}^{\infty} (1-\mu_i)^{n-1} \cdot n = \mu_i^{-1}.$$

At any time point $t = 1, 2, \ldots$ the server chooses a customer from one of the queues: this is an example of a server assignment model. Services may be interrupted and resumed later on (*preemption*). For each customer in queue i, a cost c_i is charged per unit of time that this customer is in the system. A policy is a rule to assign each server to one of the queues. As optimization problem we consider: which policy minimizes the total cost in T periods?

Let $N_i^t(R)$ be the number of customers in period t in queue i, if policy R is used. Then, the performance measure is

$$min_R \ \mathbb{E}\left\{ \sum_{t=1}^{T} \sum_{i=1}^{m} c_i \cdot N_i^t(R) \right\}.$$

It can be shown that the so-called μc-rule is an optimal policy. This rule assigns the server to queue k, where k is a nonempty queue satisfying

$$\mu_k c_k = max_i\{\mu_i c_i \mid \text{queue } i \text{ is nonempty}\}.$$

Note that $\mu_i c_i$ is the expected cost per unit of service time for a customer in queue i, and by using the μc-rule, the largest reduction of the expected cost in the next period is obtained.

Poisson arrivals and two servers: threshold policy

Consider a system with two servers where the customers arrive according to a Poisson process with rate λ, and where there is only one queue. The service times are assumed to be exponentially distributed with the respective rates μ_1 (for server 1) and μ_2 (for server 2), where $\mu_1 \geq \mu_2$. When one of the servers becomes available, the decision has to be taken whether or not to send the customer to this server.

This is a customer assignment model. The model is not discrete, but continuous in time. For policy R, let $N^t(R)$ be the number of customers in the system at time t. As performance measure the total discounted costs are used, i.e.

$$min_R \ \mathbb{E}\left\{ \int_0^{\infty} e^{-\alpha t} N^t(R) dt \right\},$$

where $\alpha > 0$, which is the continuous analogon of the total discounted costs in the discrete case.

For this model an optimal *threshold policy* exists, namely server 1 will always be used when it becomes available, and the slower server, server 2, is only used when the total number of customers in the queue exceeds some threshold number n.

1.3.9 Multi-armed bandit problem

The multi-armed bandit problem is a model for dynamic allocation of a resource to one of n independent alternative projects. The terminology 'multi-armed bandit' comes from the interpretation of the projects as arms of a gambling machine.

Any project may be in one of a finite number of states, say project j in the set $S_j,\ j = 1, 2, \dots, n$. Hence, the state space S is the Cartesian product

$$S = S_1 \times S_2 \times \cdots \times S_n.$$

Each state $i = (i_1, i_2, \dots, i_n)$ has the same action set $A = \{1, 2, \dots, n\}$, where action a means that project a is chosen, $a = 1, 2, \dots, n$. So, at each stage one can be working on exactly one of the projects.

When project a is chosen in state i - the chosen project is called the *active project* - the immediate reward and the transition probabilities only depend on the active project, whereas the states of the remaining projects are frozen. As utility function the total discounted reward is chosen.

There are many applications of this model, e.g. in machine scheduling, in the control of queueing systems and in medicine, when dealing with the selection of decision trials.

It can be shown that an optimal policy is the policy that selects project a in state $i = (i_1, i_2, \dots, i_n)$, where a satisfies

$$G_a(i_a) = max_{1 \le k \le n}\ G_k(i_k)$$

for certain numbers $G_k(i_k),\ i_k \in S_k,\ 1 \le k \le n$. Such a policy is called an *index policy.* Surprisingly, these numbers $G_k(i_k)$ only depend on project k and not on the other projects. This result is a fundamental contribution made by Gittins and therefore these indices are called the *Gittins indices.*

As a consequence, the multi-armed bandit problem can be solved by a sequence of n one-armed bandit problems. This is a *decomposition* result by which the dimensionality of the problem is reduced considerably. Algorithms with complexity $\mathcal{O}\big(\sum_{k=1}^{n} n_k^3\big)$, where $n_k = |S_k|,\ 1 \le k \le n$, do exist for the computation of all indices.

1.4 Bibliographic notes

Bellman's book [17] can be considered as the starting point for the study of Markov decision processes. However, as early as 1953, Shapley's paper [267] on stochastic games includes as a special case the discounted Markov decision process. Around 1960 the basics for solution methods for MDPs were developed in publications as Howard [134], De Ghellinck [51], d'Epenoux [67], Manne [193] and Blackwell [29]. Since the early sixties, many results on MDPs have been published in numerous journals, monographs, books and proceedings. Around 1970 a first series of books was published, e.g. Derman [69], Mine and Osaki [200] and Ross [236]. In 1994, the rather comprehensive book by Puterman was published ([227]).

The result mentioned in Corollary 1.1 on the sufficiency of Markov policies for performance measures that only depend on the marginal distributions is due to Derman and Strauch ([71]). The extension to Theorem 1.1 was given by Strauch and Veinott ([286]).

The relation between the four criteria for average rewards and the result that these four criteria are equivalent for stationary policies is due to Bierth ([28]).

In a fundamental paper Blackwell ([29]) introduced the concepts of bias optimality (Blackwell called it *nearly optimal*) and Blackwell optimality. An algorithm for finding a Blackwell optimal policy was constructed by Miller and Veinott ([199]). The n-discount optimality criterion was proposed in Veinott [311]. He also showed that Blackwell optimality is equivalent to n-discount optimality for all $n \geq |S| - 1$.

The concept of n-average optimality was announced in Veinott [310], which is an abstract of a preliminary report. This report was never published. In Sladky [273] a proof is given of the equivalence between n-average optimality and n-discount optimality.

The criterion of overtaking optimality was proposed by Denardo and Rothblum ([66]). For this criterion no optimal policy may exist. Denardo and Rothblum also provided conditions under which an optimal policy exists. The concept of average overtaking optimality was proposed by Veinott ([308]), where he used the terminology *optimal*. He presented an algorithm for finding a bias-optimal policy, showed that an average overtaking policy is bias-optimal and conjectured that the converse was also true. This conjecture has been proven by Denardo and Miller ([65]).

There is an extensive literature on examples of MDP models. Seminal papers on inventory models are written by Scarf ([251],[252]), Iglehart ([138],[139]) and Veinott ([309]).

A standard reference on gambling is Dubins and Savage [75], who have shown for example that the bold policy is optimal if $p \leq \frac{1}{2}$. The optimality of the timid policy for $p \geq \frac{1}{2}$ is due to Ross ([238] and [239]). The example of the tennis game is due to Norman ([206]) and Prussing ([226]).

A dynamic programming approach for optimal stopping problems can be found in Breiman [32], who showed the optimality of control-limit policies. The house selling example (Example 1.5) comes from Ross ([236]).

There are a lot of references on replacement and repair models. The survey of Sherif and Smithn ([269]) contains over 500 references. Results on the optimality of control-limit policies can be found in Derman [68], Kolesar [170], Derman [69], Ross [236] and Kao [154]. Our presentation of the n-component series system with exponential distributions is based on Katehakis and Derman [159]. They showed the optimality of the *SFR-policy*. This result was first conjectured by Smith ([275]).

There is a close relation between production control problems and flows in networks. For more detailed information about this subject we refer the reader to Chapter 5 in Denardo [63].

The literature on optimal control of queues is also quite extensive. Markov decision processes with continuous time parameter were introduced by Bellmann ([17], Chapter 11). For time discretization we refer to Hordijk and Van Dijk [133]. The technique of uniformization was already suggested by Howard ([134], page 113). Schweitzer [256] has generalized this idea for general non-exponential mean holding times and has explicitly given the data transformations mentioned at the end of Section 1.3.7.

For reviews on stochastic scheduling we refer to Weiss [321], Walrand [318] (Chapter 8), and

Righter [235]. The optimality of the μc-rule is due to Baras, Ma and Makowsky ([9]), see also Buyukkoc, Varaiya and Walrand [36]. The structural result of an optimal threshold policy in the two server model with Poisson arrivals is from Lin and Kumar ([180]).

The most fundamental contribution on multi-armed bandit problems has been made by Gittins ([105], [104]). The importance of Gittins' work had not been recognized in the seventies. The re-discovery is due to Whittle ([332]) who has given an easier and more natural proof. Other proofs are given by Ross ([239]), Varaiya, Walrand and Buyukkoc ([306]), Tsitsiklis ([291]) and Weber ([320]). Several methods are developed for the computation of the Gittins indices: Varaiya, Walrand and Buyukkoc [306], Chen and Katehakis [38], Kallenberg [149], Katehakis and Veinott [162], Ben-Israel and Flåm [20], and Liu and Liu [184].

1.5 Exercises

Exercise 1.1 *Inventory model without backlogging*
Consider a finite horizon nonstationary inventory model. If demands exceed the supply, then there is no backlogging. For this model we have the following notations:

T = the number of periods in the planning horizon;
$p_j(t)$ = the probability of demand j in period t, $j = 0, 1, \ldots;$
c = the cost price of an item;
h = the holding cost of an item that is unsold at the end of a period;
p = the penalty cost of an item that cannot be delivered during a period;
B = the finite inventory capacity.

The optimization problem is: which inventory strategy minimizes the total expected costs?
Formulate this model as a Markov decision model.

Exercise 1.2 *Number of Markov policies*
Let $N = |S|$ and $m_i = |A(i)|,\ i \in S$.
What is the number of nonrandomized Markov policies in this finite horizon MDP with T periods?

Exercise 1.3 *n-discount optimality*
Show that n-discount optimality implies $(n-1)$-discount optimality for $n = 0, 1, \ldots$.

Exercise 1.4 *Red-black gambling with $p = \frac{1}{2}$*
Consider the red-black gambling model with $p = \frac{1}{2}$. Let f_1^∞ be the deterministic policy betting 1 euro in every round of the game. Show that policy f_1^∞ satisfies $v_i(f_1^\infty) = \frac{i}{N},\ 0 \le i \le N$.

<u>Hint:</u>

Derive a recurrence relation for $v_i(f_1^\infty)$ and show that $v_i(f_1^\infty) = \frac{i}{N},\ 0 \le i \le N.$ is the unique solution of this recurrence relation.

Exercise 1.5 *Automobile replacement problem*

Suppose that we review a car every month and that the decision is made either to keep the present car or to trade in the car for another car of a certain age. The age of a car is measured in months. In order to keep the state space finite, we assume that there is a largest age N, i.e. a car of age N will always be reset by another car. Furthermore, we assume that a car of age i has a probability p_i of a breakdown in which case it ends up in state N.

Suppose that we have the following costs and rewards:

b_i = cost of buying a car of age i;

t_i = trade-in value of a car of age i;

c_i = expected maintenance cost in the next month for a car of age i.

Formulate this automobile replacement problem as an MDP.

Exercise 1.6 *Production problem*

Consider the following variant of the production problem. Let the demand D_t in period t be stochastic with $p_j(t) = \mathbb{P}\{D_t = j\},\ j = 0, 1, \dots, N_t$ and $1 \leq t \leq T$. Because of the uncertainties it is no longer possible to satisfy the demands with probability 1. Therefore, we require that the demand in period t has to be satisfied with probability at least $\alpha_t,\ 1 \leq t \leq T$. The production in period t has a capacity of $b_t,\ 1 \leq t \leq T$. Let $c_t(a)$ denote the production cost for the production of a units in period $t,\ 1 \leq t \leq T$. There are also inventory costs $h_t(i)$ in period t, when the inventory at the end of period t is equal to $i,\ 1 \leq t \leq T$.

Formulate the MDP model for this variant of the production problem.

Exercise 1.7 *Queueing problem*

Consider a single server queueing system with a finite capacity N. The service time is a negative exponential distribution with parameter μ. The system manager can control the system by increasing or decreasing the price he charges for the service facility in order to encourage or discourage the arrival of customers.

Assume that the manager must choose one of a finite number of prices, say $p_1, p_2, \cdots, p_m$, where $0 < p_1 < p_2 < \cdots < p_m$. If there are i customers in the system and he chooses p_a, then the arriving process is a Poisson process with parameter λ_a, an arriving customer has to pay p_a and the system manager has c_i as waiting cost per time unit.

It is quite natural to assume that:

(1) $\lambda_1 > \lambda_2 > \cdots > \lambda_m$ (lower prices give more arrivals).

(2) $0 \leq c_0 \leq c_1 \leq \cdots \leq c_N$ (more costs for more customers).

(3) $p_m > c_N$ (positive net reward for the manager for each arriving customer).

a. Give the specifications for the time discretization approach.

b. Give the specifications for the semi-Markov approach. Apply uniformization to obtain an equivalent MDP model.

Exercise 1.8 *Stochastic scheduling: μc-rule*

Assume that m customers are present at the service station and have to be processed nonpreemptively by one server. Let μ_i^{-1}be the expected service time and c_i the cost per unit time for customer i. Show, by an interchanging argument, that the μc-rule is optimal for scheduling the jobs in order to minimize the total expected costs.

Chapter 2

Finite Horizon

2.1 Introduction

A system with rewards $r_i^t(a)$ and transition probabilities $p_{ij}^t(a)$ has to be controlled over a planning horizon of T periods. As you see in the notation, these rewards and transition probabilities may be nonstationary, i.e. may depend on the period t. As utility function the total expected reward is considered as defined in (1.9).

In section 2.2 we shown that an optimal Markov policy with deterministic but in general nonstationary decision rules exists. Furthermore, we show that such an optimal policy can be computed by *backward induction.*, based on the *principle of optimality.*

In section 2.3 an alternative stationary substochastic model over an infinite horizon is described. The utility function of this model is the total expected reward. This infinite horizon stationary model is equivalent to the finite horizon nonstationary model in the sense that there is equivalence between the policies in both models such that equivalent policies have the same value of their utility functions. Hence, results of the infinite horizon model, which is discussed in Chapter 4, can be applied to the finite horizon model.

In section 2.4 we study under which conditions optimal policies are *monotone*, i.e. nondecreasing or nonincreasing. Such a concept is worthwhile if there is a natural *ordering* in the state space. Knowledge about the monotone structure of optimal policies enables us to find such policy with less computational effort than without the monotone structure.

We close this chapter with bibliographic notes (section 2.5) and exercises (section 2.6).

2.2 Backward induction

In this section we show how to compute an optimal policy by backward induction. Backward induction is an iterative procedure. Starting at the end of the planning horizon one computes iteratively the values for the previous periods. Then, after T iterations, where T is the number of periods in the planning horizon, an optimal policy is found.
The notation $r(f^t)$ and $P(f^t)$, as defined in (1.5) and (1.4) respectively, is used for the reward vector and transition matrix of a deterministic decision rule f^t at decision time point t. In a finite planning horizon with T periods only the decision rules for the first T decision time points are relevant. Hence, we write $R = (\pi^1, \pi^2, \dots, \pi^T)$.

Theorem 2.1
Let $x_i^{T+1} = 0$ for all $i \in S$. Let for $t = T, T-1, \dots, 1$ consecutively, respectivelely a deterministic decision rule f^t and a vector x^t be defined as

$$\{r(f^t)\}_i + \{P(f^t)x^{t+1}\}_i = max_{a\in A(i)}\ \{r_i^t(a) + \sum_j\ p_{ij}^t(a)x_j^{t+1}\},\ \textit{for all } i \in S \tag{2.1}$$

and

$$x^t = r(f^t) + P(f^t)x^{t+1}.$$

Then, $R_* = (f^1, f^2, \dots, f^T)$ is an optimal policy and x^1 is the value vector v^T.

Proof
We use induction on T. Let $R = (\pi^1, \pi^2, \dots, \pi^T)$ be an arbitrary policy.
For $T = 1$, we obtain

$$\begin{aligned} v_i^T(R) &= \textstyle\sum_{j,a}\ \mathbb{P}\{X_1 = j, Y_1 = a\} \cdot r_j^1(a) = \sum_a\ r_i^1(a)\pi_{ia}^1 \\ &\leq\ max_{a\in A(i)}\ r_i^1(a) = x_i^1 = v_i^1(R_*),\ i \in S. \end{aligned}$$

Assume that the result has been shown for $T = 1, 2, \dots, t$. Take an arbitrary state i.
From Corollary 1.1 it follows that there exists a Markov policy $\overline{R}$ such that $v_i^{t+1}(\overline{R}) = v_i^{t+1}(R)$.
Let $\overline{R} = (\sigma^1, \sigma^2, \dots, \sigma^{t+1})$. Define the Markov policy $R' = (\rho^1, \rho^2, \dots, \rho^t)$ by $\rho_{ja}^k = \sigma_{ja}^{k+1}$ for all $(j, a) \in S \times A$ and for $k = 1, 2, \dots, t$. From the induction assumption it follows that $v_j^t(R') \leq x_j^2$ for all $j \in S$, because for a planning horizon of $t+1$ periods x^2 is the same as x^1 for a planning horizon of t periods. Hence,

$$\begin{aligned} v_i^{t+1}(R) &= v_i^{t+1}(\overline{R}) = \textstyle\sum_a\ \sigma_{ia}^1\{r_i^1(a) + \sum_j\ p_{ij}^1(a)v_j^t(R')\} \\ &\leq \textstyle\sum_a\ \sigma_{ia}^1\{r_i^1(a) + \sum_j\ p_{ij}^1(a)x_j^2\} \leq max_a\ \{r_i^1(a) + \sum_j\ p_{ij}^1(a)x_j^2\} = x_i^1. \end{aligned}$$

On the other hand,

$$\begin{aligned} x^1 &= r(f^1) + P(f^1)x^2 = r(f^1) + P(f^1)\{r(f^2) + P(f^2)x^3\} \\ &= \cdots = \textstyle\sum_{s=1}^{t+1}\ \{P(f^1)P(f^2)\cdots P(f^{s-1})r(f^s)\} = v^{t+1}(R_*), \end{aligned}$$

i.e. $v^{t+1}(R_*) = x^1 \geq v^{t+1}(R)$, i.e. R_* is an optimal policy and x^1 is the value vector. □

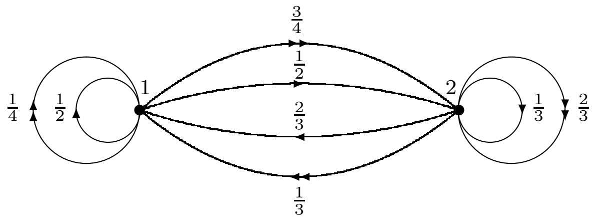
Algorithm 2.1 *Determination of an optimal policy for a nonstationary MDP over T periods*
Input: Instance of a finite nonstationary MDP and the time horizon T.
Output: Optimal Markov policy $R_* = (f^1, f^2, \dots, f^T)$ and the value vector v^T.

```
1. $x := 0$

2. for $t = T, T-1, \dots, 1$ do
   begin
   (1) take $f^t$ such that $\{r(f^t) + P(f^t)x\}_i = max_{a \in A(i)} \; \{r_i^t(a) + \sum_j \; p_{ij}^t(a)x_j\}$ for all $i \in S$
   (2) $x := r(f^t) + P(f^t)x$
   end

3. $R_* := (f^1, f^2, \dots, f^T)$ is an optimal policy and $x$ is the value vector.
```

Example 2.1
Consider an MDP with the following data:
$S = \{1, 2\};\ A(1) = A(2) = \{1, 2\};\ T = 3.$
$p_{11}(1) = \frac{1}{2}; p_{12}(1) = \frac{1}{2}; r_1(1) = 1;$
$p_{11}(2) = \frac{1}{4}; p_{12}(2) = \frac{3}{4}; r_1(2) = 0;$
$p_{21}(1) = \frac{2}{3}; p_{22}(1) = \frac{1}{3}; r_2(1) = 2;$
$p_{21}(2) = \frac{1}{3}; p_{22}(2) = \frac{2}{3}; r_2(2) = 5.$
Start with $x_1 = x_2 = 0$.



$t = 3 : i = 1 : max\{1, 0\} = 1;\ f^3(1) = 1;\ x_1 = 1.$
$\quad i = 2 : max\{2, 5\} = 5;\ f^3(2) = 2;\ x_2 = 5.$
$t = 2 : i = 1 : max\{1 + \frac{1}{2} \cdot 1 + \frac{1}{2} \cdot 5, 0 + \frac{1}{4} \cdot 1 + \frac{3}{4} \cdot 5\} = 4;\ f^2(1) = 1 \text{ (or 2)};\ x_1 = 4.$
$\quad i = 2 : max\{2 + \frac{2}{3} \cdot 1 + \frac{1}{3} \cdot 5, 5 + \frac{1}{3} \cdot 1 + \frac{2}{3} \cdot 5\} = \frac{26}{3};\ f^2(2) = 2;\ x_2 = \frac{26}{3}.$
$t = 1 : i = 1 : max\{1 + \frac{1}{2} \cdot 4 + \frac{1}{2} \cdot \frac{26}{3}, 0 + \frac{1}{4} \cdot 4 + \frac{3}{4} \cdot \frac{26}{3}\} = \frac{15}{2};\ f^1(1) = 2;\ x_1 = \frac{15}{2}.$
$\quad i = 2 : max\{2 + \frac{2}{3} \cdot 4 + \frac{1}{3} \cdot \frac{26}{3}, 5 + \frac{1}{3} \cdot 4 + \frac{2}{3} \cdot \frac{26}{3}\} = \frac{109}{9};\ f^1(2) = 2;\ x_2 = \frac{109}{9}.$
$R_* = (f^1, f^2, f^3)$ is an optimal policy and $x = (\frac{15}{2}, \frac{109}{9})$ is the value vector.

Application 2.1 *Scheduling*
Suppose that N jobs have to be processed on one machine. Assume that the machine can process at most one job at a time, that job j has processing time p_j and that $c_j(t)$ is the cost if job j is completed at time t.
A strategy R corresponds to a permutation of the N jobs, say $R = \{i_1, i_2, \dots, i_N\}$. Given strategy $R = \{i_1, i_2, \dots, i_N\}$, job i_k has completion time $\sum_{j=1}^k p_{i_j}$. Hence, the corresponding cost is $c_{i_k}(\sum_{j=1}^k p_{i_j})$ and the total costs of this policy are $\sum_{k=1}^N c_{i_k}(\sum_{j=1}^k \; p_{i_j})$. Which order of the jobs minimizes the total costs?

This problem can be modeled as a finite horizon MDP with a layered state space. The states are the 2^N subsets of $\{1, 2, \dots, N\}$. Layer 1 consists of the single state $\{1, 2, \dots, N\}$, layer 2 has the N states $\{1, 2, \dots, N\} \backslash \{j\},\ 1 \le j \le N$, and so on until layer $N+1$, which consists of the empty state $\emptyset$. Any path from $\{1, 2, \dots, N\}$ to $\emptyset$ corresponds to a permutation: at each stage, the job which is deleted from the state is chosen as scheduled at this stage on the machine.
When job j is chosen, i.e. deleted from a subset $J \subseteq \{1, 2, \dots, N\}$, the jobs from $\{1, 2, \dots, N\} \backslash J$ are already scheduled on the machine. Hence, the completion time of job j is $\sum_{i \notin J} p_i \ + \ p_j$ with costs $c_j(\sum_{i \notin J} p_i + p_j)$. Therefore, this scheduling problem is equivalent to a *layered shortest path problem*, which can be solved as an MDP with finite horizon (see Exercise 2.1).
Formally, in state $J \subseteq \{1, 2, \dots, N\}$ the action set $A(J)$ satisfies $A(J) = \{j \mid j \in J\}$ and, if action j is chosen, the immediate costs are $c_j(\sum_{i \notin J} p_i + p_j)$ and there is a deterministic transition to state $J \backslash \{j\}$ in the next layer.

2.3 An equivalent stationary infinite horizon model

In this section we present a stationary infinite horizon model which is equivalent to the standard nonstationary finite horizon model. The reason behind the infinite model is to copy the state space for each period, to make transitions from a period to the next period and to take in the last period only absorbing states without rewards. Therefore, consider the following stationary MDP with infinite horizon for which the state space, action sets, immediate rewards and transition probabilities, denoted by $S^*,\ A^*,\ r^*$ and p^* respectively, are given by:

$$\begin{aligned}
S^* &= \{(i,t) \mid i \in S,\ t = 1, 2, \dots, T+1\} \\
A^*\{(i,t)\} &= \begin{cases} A(i) & i \in S,\ t = 1, 2, \dots, T \\ \{1\} & i \in S,\ t = T+1 \end{cases} \\
r^*_{(i,t)}(a) &= \begin{cases} r^t_i(a) & i \in S,\ t = 1, 2, \dots, T,\ a \in A(i) \\ 0 & i \in S,\ t = T+1,\ a = 1 \end{cases} \\
p^*_{(i,t)(j,s)}(a) &= \begin{cases} p^t_{ij}(a) & i \in S,\ t = 1, 2, \dots, T,\ a \in A(i),\ j \in S,\ s = t+1 \\ 0 & \text{elsewhere} \end{cases} \\
p^*_{(i,T+1)(j,s)}(1) &= \begin{cases} 1 & i \in S,\ j = i,\ s = T+1 \\ 0 & \text{elsewhere} \end{cases}
\end{aligned}$$

Hence, this new infinite horizon model has a layered state space - as in Application 2.1 with transitions from (i,t) to $(j, t+1)$ until we reach a state in layer $(\cdot, T+1)$. All states of this last layer are absorbing. This infinite horizon model is a so-called *transient MDP* (see Chapter 4). For initial state (i,t) and policy R the total expected reward over the infinite horizon is denoted by $v_{(i.t)}(R)$. Any Markov policy $R = (\pi^1, \pi^2, \dots, \pi^T)$ of the finite horizon model corresponds to a stationary policy π^∞ of the infinite horizon model by

$$\pi_{(i.t)}(a) = \begin{cases} \pi^t_i(a) & i \in S,\ t = 1, 2, \dots, T,\ a \in A(i) \\ 1 & i \in S,\ t = T+1,\ a = 1 \end{cases}$$

The next Lemma shows that for these corresponding policies the respective utility functions have the same value.

Lemma 2.1
Let $R = (\pi^1, \pi^2, \dots, \pi^T)$ be a Markov policy of the finite horizon model with corresponding stationary policy π^∞ of the infinite horizon model. Then, $v_i^T(R) = v_{(i,1)}(\pi^\infty)$ for all $i \in S$.

Proof
By induction on t it is easy to show that for all i, j, t

$$\big\{[P^*(\pi)]^{t-1}\big\}_{(i,1)(j,t)} = \{P(\pi^1)P(\pi^2)\cdots P(\pi^{t-1})\}_{ij} \text{ and } r^*_{(j,t)}(\pi) = r_j(\pi^t),\ t \le T.$$

$$\textstyle\sum_j \big\{[P^*(\pi)]^T\big\}_{(i,1)(j,T+1)} = 1 \text{ and } r^*_{(j,T+1)}(\pi) = 0.$$

Hence,

$$\begin{aligned} v_{(i,1)}(\pi^\infty) = &\ \textstyle\sum_{t=1}^\infty \big\{[P^*(\pi)]^{t-1}r^*\pi)\big\}_{(i,1)} = \sum_{t=1}^T \big\{[P^*(\pi)]^{t-1}r^*(\pi)\big\}_{(i,1)} \\ = &\ \textstyle\sum_{t=1}^T \sum_{j\in S} \{[P^*(\pi)]^{t-1}\}_{(i,1)(j,t)} r^*_{(j,t)}(\pi) \\ = &\ \textstyle\sum_{t=1}^T \sum_{j\in S} [P(\pi^1)P(\pi^2)\cdots P(\pi^{t-1})]_{ij} r_j(\pi^t) \\ = &\ \textstyle\sum_{t=1}^T \big\{P(\pi^1)P(\pi^2)\cdots P(\pi^{t-1})r(\pi^t)\big\}_i = v_i^T(R),\ i \in S. \end{aligned}$$ □

Since the finite horizon model has an optimal policy in the class of Markov policies with deterministic decision rules, the corresponding infinite horizon transient MDP model has an optimal policy in the class of deterministic policies. By the method of linear programming for MDPs (see the next chapters), MDPs with additional constraints on the state-action frequencies can also be handled.

2.4 Monotone optimal policies

In this section we study under which conditions optimal policies are *monotone*, i.e. nondecreasing or nonincreasing, in the ordering of state space. Such concept is worthwhile if there is a natural *ordering* in the state space. Knowlegde about the structure of optimal policies enables us to find such policies with less computational effort. In section 1.3 we have encountered several examples of special models with structured optimal policies, e.g. control-limit policies.

For the proof of the optimality of monotone policies, the following lemma is important.

Lemma 2.2
Let $y, z : S \to \mathbb{R}_+$ satisfy $\sum_{j=k}^N y_j \ge \sum_{j=k}^N z_j,\ 2 \le k \le N$ and $\sum_{j=1}^N y_j = \sum_{j=1}^N z_j$, and let $v : S \to \mathbb{R}$ satisfy $v_{j+1} \ge v_j,\ j = 1, 2, \dots, N-1$. Then, $\sum_{j=1}^N v_j y_j \ge \sum_{j=1}^N v_j z_j$.

Proof
Define $v_0 := 0$. Then,

$$\begin{aligned}\textstyle\sum_{j=1}^N v_j y_j = &\textstyle\sum_{j=1}^N y_j \big\{ \sum_{k=1}^j (v_k - v_{k-1}) \big\} = \sum_{k=1}^N (v_k - v_{k-1}) \big\{ \sum_{j=k}^N y_j \big\} \\ = \;&\textstyle v_1 \sum_{j=1}^N y_j + \sum_{k=2}^N (v_k - v_{k-1}) \big\{ \sum_{j=k}^N y_j \big\} \\ \geq \;&\textstyle v_1 \sum_{j=1}^N z_j + \sum_{k=2}^N (v_k - v_{k-1}) \big\{ \sum_{j=k}^N z_j \big\} = \sum_{j=1}^N v_j z_j. \end{aligned}$$ □

Let X and Y be ordered sets and let $f(x,y)$ a real-valued function on $X \times Y$. The function f is said to be *supermodular* (also called *superadditive*) if for any $x_1, x_2 \in X$ and $y_1, y_2 \in Y$ with $x_1 \geq x_2$ and $y_1 \geq y_2$

$$f(x_1,y_1) + f(x_2,y_2) \geq f(x_1,y_2) + f(x_2,y_1).$$

If the reverse inequality holds, i.e. if for any $x_1, x_2 \in X$ and $y_1, y_2 \in Y$ with $x_1 \geq x_2$ and $y_1 \geq y_2$

$$f(x_1,y_1) + f(x_2,y_2) \leq f(x_1,y_2) + f(x_2,y_1),$$

the function f is called *submodular* or *subadditive.*

If $X = Y = \mathbb{R}$ and $f(x,y)$ is twice differentiable and supermodular, then $\frac{\partial^2 f(x,y)}{\partial x \partial y} \geq 0$ for all x and y (see Exercise 2.8).

Examples of supermodular functions on $\mathbb{R} \times \mathbb{R}$ are (see Exercise 2.6):
(1) $f(x,y) = (x+y)^2$.
(2) $f(x,y) = xy$.
(3) $f(x,y) = g(x+y)$ for any convex function g.

Argmax stands for the argument of the maximum, i.e. the set of points of the given argument for which the given function attains its maximum value. For example, $argmax_{x \in \mathbb{R}}\ (1 - |x|) = \{0\}$. For a fixed $x \in X$, we say that $Y(x) = \{y \in Y \mid y \in argmax\, f(x,y)\}$ and $y(x) = max\{y \in Y(x)\}$.

Lemma 2.3

Suppose f is supermodular on $X \times Y$ and for each $x \in X$ $max_{y \in Y}\ f(x,y)$ exists. Then, $y(x)$ is nondecreasing in x.

Proof

Let $x_1 \geq x_2$ and choose $y \leq y(x_2)$. Then,

$$f\big(x_1, y(x_2)\big) + f(x_2, y) \geq f(x_1, y) + f\big(x_2, y(x_2)\big),$$

i.e.

$$f\big(x_1, y(x_2)\big) \geq f(x_1, y) + \{f\big(x_2, y(x_2)\big) - f(x_2, y)\} \geq f(x_1, y),$$

the last inequality by the definition of $y(x_2)$. Hence, $f(x_1, y(x_2)) \geq f(x_1, y)$ for all $y \leq y(x_2)$. By the definition of $y(x_1)$, we have $y(x_1) \geq y(x_2)$, implying that $y(x)$ is nondecreasing in x. □

Lemma 2.4

Suppose f is submodular on $X \times Y$ and for each $x \in X$ $max_{y \in Y}\ f(x,y)$ exists. Then, $y(x)$ is nonincreasing in x.

Proof

Let $x_1 \geq x_2$ and choose $y \leq y(x_1)$. Then,

$$f\big(x_2, y(x_1)\big) + f(x_1, y) \geq f(x_2, y) + f\big(x_1, y(x_1)\big),$$

i.e.

$$f\big(x_2, y(x_1)\big) \geq f(x_2, y) + \{f\big(x_1, y(x_1)\big) - f(x_1, y)\} \geq f(x_2, y),$$

the last inequality by the definition of $y(x_1)$. Hence, $f(x_2, y(x_1)) \geq f(x_2, y)$ for all $y \leq y(x_1)$. By the definition of $y(x_2)$, we have $y(x_2) \geq y(x_1)$, implying that $y(x)$ is nonincreasing in x. □

We will show the existence of optimal monotone policies under certain assumption. Firstly, we consider the nondecreasing case.

Assumption 2.1

(A1) $S = \{1, 2, \dots, N\}$, ordered in the natural way;

(A2) $r_i^t(a)$ is nondecreasing in i for all a and t;

(A3) $\sum_{j=k}^N p_{ij}^t(a)$ is nondecreasing in i for all k, a and t.

Theorem 2.2

Under Assumption 2.1, the function x_i^t, defined in Theorem 2.1, is nondecreasing in i for all t.

Proof

Apply backward induction on t. For $t = T + 1:\ x_i^{T+1} = 0$ for all i, so the result is true.

Assume that the result holds for $t + 1$ and consider $x^t = r(f^t) + P(f^t)x^{t+1}$.

Let $i_1 \geq i_2$, and let $y_j = p_{i_1 j}^t(f^t(i_2))$ and $z_j = p_{i_2 j}^t(f^t(i_2))$.

From Assumption 2.1 (A3), we obtain for all k

$$\sum_{j=k}^N y_j = \sum_{j=k}^N p_{i_1 j}^t\big(f^t(i_2)\big) \geq \sum_{j=k}^N p_{i_2 j}^t\big(f^t(i_2)\big) = \sum_{j=k}^N z_j.$$

Notice that $\sum_{j=1}^N y_j = \sum_{j=1}^N z_j = 1$ and that, by induction hypothesis, $x_{j+1}^{t+1} \geq x_j^{t+1}$ for $j = 1, 2, \dots, N-1$. Applying Lemma 2.2 yields

$$\sum_{j=1}^N p_{i_1 j}^t(f^t(i_2)) x_j^{t+1} \geq \sum_{j=1}^N p_{i_2 j}^t(f^t(i_2)) x_j^{t+1}.$$

Hence, using Assumption 2.1 (A2),

$$\begin{aligned} x_{i_1}^t &= max_{a \in A}\{r_{i_1}^t(a) + \textstyle\sum_{j=1}^N\ p_{i_1 j}^t(a) x_j^{t+1}\} \geq r_{i_1}^t(f^t(i_2)) + \textstyle\sum_{j=1}^N\ p_{i_1 j}^t(f^t(i_2)) x_j^{t+1} \\ &\geq r_{i_2}^t(f^t(i_2)) + \textstyle\sum_{j=1}^N\ p_{i_2 j}^t(f^t(i_2)) x_j^{t+1} = x_{i_2}^t. \end{aligned}$$

□

Assumption 2.2

(A4) The action set $A(i) = A = \{1, 2, \dots, M\},\ i \in S$, where A is ordered in the natural way;

(A5) $r_i^t(a)$ is supermodular on $S \times A$ for $t = 1, 2, \dots, T$;

(A6) $\sum_{j=k}^N p_{ij}^t(a)$ is supermodular on $S \times A$ for $t = 1, 2, \dots, T$ and for all $k \in S$.

Theorem 2.3

Let Assumption 2.1 and 2.2 hold. Then, there exists an optimal policy $R_* = (f^1, f^2, \dots, f^T)$, where $f^t(i)$ is nondecreasing in i for $t = 1, 2, \dots, T$.

Proof

Take any $1 \le t \le T$. We first prove that $s_i^t(a) := r_i^t(a) + \sum_{j=1}^N p_{ij}^t(a)x_j^{t+1}$ is supermodular on $S \times A$. Let $i_1 \ge i_2,\ a_1 \ge a_2$, and let $y_j = p_{i_1j}^t(a_1) + p_{i_2j}^t(a_2),\ z_j = p_{i_1j}^t(a_2) + p_{i_2j}^t(a_1),\ j \in S$. By Assumption 2.2 (A6), for all $k \in S$, we have $\sum_{j=k}^N y_j \ge \sum_{j=k}^N z_j$. Since $\sum_{j=1}^N y_j = \sum_{j=1}^N z_j = 2$, and because x_i^{t+1} is nondecreasing in i (see Theorem 2.2), applying Lemma 2.2 yields

$$\sum_{j=1}^N \{p_{i_1j}^t(a_1) + p_{i_2j}^t(a_2)\}x_j^{t+1} \ge \sum_{j=1}^N \{p_{i_1j}^t(a_2) + p_{i_2j}^t(a_1)\}x_j^{t+1},$$

implying the supermodularity of $\sum_{j=1}^N p_{ij}^t(a)x_j^{t+1}$. Because $r_i^t(a)$ is supermodular by Assumption 2.2 (A5) and because the sum of supermodular functions is also supermodular (see Exercise 2.5), $s_i^t(a)$ is a supermodular function on $S \times A$. If the action $f^t(i)$ in formula (2.1) is not unique, take the largest optimal action. Then, applying Lemma 2.3 yields the result that $f^t(i)$ is nondecreasing in i. □

Algorithm 2.2 *Determination of an optimal policy with monotone decision rules for a nonstationary MDP over T periods under the assumptions 2.1 and 2.2.*

Input: Instance of a finite nonstationary MDP, which satisfies assumptions 2.1 and 2.2, and the time horizon T.

Output: Optimal Markov policy $R_* = (f^1, f^2, \dots, f^T)$ with nondecreasing decision rules $f^t(i)$, $i \in S$, $1 \le t \le T$, and the value vector v^T.

```
1. $x := 0$.

2. for $t = T, T-1, \dots, 1$ do
     begin $A := \{1, 2, \dots, M\}$
           for $i = 1, 2, \dots, N$ do
           begin (1) take $f^t(i)$ such that $\{r(f^t) + P(f^t)x\}_i = max_{a \in A}\ \{r_i^t(a) + \sum_j p_{ij}^t(a)x_j\}$
                     (if there is more than one optimizing action, take the largest)
                 (2) $y_i := \{r(f^t) + P(f^t)x\}_i$
                 (3) $A := \{a \mid f^t(i) \le a \le M\}$
           end
     $x := y$
     end
```

3. $R_* = (f^1, f^2, \dots, f^T)$ is an optimal policy with nondecreasing decision rules $f^t(i),\ i \in S$, $1 \le t \le T$, and the value vector $v^T := x$.

Remark

The advantage of this algorithm is that the maximization can be carried out over action sets which become smaller in the order of the states. If for some state i the action set consists of a singleton no optimization is needed in higher states.

Next, we consider the case in which the rewards are nonincreasing and submodular.

Assumption 2.3

(B1) $S = \{1, 2, \dots, N\}$, ordered in the natural way;

(B2) $r^t_i(a)$ is nonincreasing in i for all a and t;

(B3) $\sum_{j=k}^N p^t_{ij}(a)$ is nondecreasing in i for all k, a and t.

Theorem 2.4

Under Assumption 2.3, the function x^t_i, defined in Theorem 2.1, is nonincreasing in i for all t.

Proof

Apply backward induction on t. For $t = T+1:\ x^{T+1}_i = 0$ for all i, so the result is true.

Assume that the result holds for $t+1$ and consider $x^t = r(f^t) + P(f^t)x^{t+1}$.

Let $i_1 \ge i_2$, and let $y_j = p^t_{i_1j}(f^t(i_1))$ and $z_j = p^t_{i_2j}(f^t(i_1))$.

From Assumption 2.3 (B3), we obtain for all k

$$\sum_{j=k}^N y_j = \sum_{j=k}^N p^t_{i_1j}\big(f^t(i_1)\big) \ge \sum_{j=k}^N p^t_{i_2j}\big(f^t(i_1)\big) = \sum_{j=k}^N z_j.$$

Notice that $\sum_{j=1}^N y_j = \sum_{j=1}^N z_j = 1$ and that, by induction hypothesis, $-x^{t+1}_{j+1} \ge -x^{t+1}_j$ for $j = 1, 2, \dots, N-1$. Applying Lemma 2.2 yields

$$\sum_{j=1}^N p^t_{i_1j}(f^t(i_1))\{-x^{t+1}_j\} \ge \sum_{j=1}^N p^t_{i_1j}(f^t(i_2))\{-x^{t+1}_j\}$$

i.e.

$$\sum_{j=1}^N p^t_{i_2j}(f^t(i_1))x^{t+1} \ge \sum_{j=1}^N p^t_{i_1j}(f^t(i_1))x^{t+1}_j.$$

Hence, using Assumption 2.3 (B2),

$$\begin{aligned} x^t_{i_2} = \;& max_{a\in A}\{r^t_{i_2}(a) + \textstyle\sum_{j=1}^N p^t_{i_2j}(a)x^{t+1}_j\} \ge r^t_{i_2}(f^t(i_1)) + \textstyle\sum_{j=1}^N p^t_{i_2j}(f^t(i_1))x^{t+1}_j \\ \ge \;& r^t_{i_1}(f^t(i_1)) + \textstyle\sum_{j=1}^N p^t_{i_1j}(f^t(i_1))x^{t+1}_j = x^t_{i_1}. \end{aligned}$$

□

Assumption 2.4

(B4) The action set $A(i) = A = \{1, 2, \dots, M\},\ i \in S$, where A is ordered in the natural way;

(B5) $r^t_i(a)$ is submodular on $S \times A$ for $t = 1, 2, \dots, T$;

(B6) $\sum_{j=k}^N p^t_{ij}(a)$ is supermodular on $S \times A$ for $t = 1, 2, \dots, T$ and for all $k \in S$.

Theorem 2.5

Let Assumption 2.3 and 2.4 hold. Then, there exists an optimal policy $R_* = (f^1, f^2, \dots, f^T)$, where $f^t(i)$ is nonincreasing in i for $t = 1, 2, \dots, T$.

Proof

Take any $1 \leq t \leq T$. We first prove that $s^t_i(a) := r^t_i(a) + \sum_{j=1}^N p^t_{ij}(a)x^{t+1}_j$ is submodular on $S \times A$. Let $i_1 \geq i_2,\ a_1 \geq a_2$, and let $y_j = p^t_{i_1j}(a_1) + p^t_{i_2j}(a_2),\ z_j = p^t_{i_1j}(a_2) + p^t_{i_2j}(a_1),\ j \in S$. By Assumption 2.4 (B6), for all $k \in S$, we have $\sum_{j=k}^N y_j \geq \sum_{j=k}^N z_j$. Since $\sum_{j=1}^N y_j = \sum_{j=1}^N z_j = 2$, and because $-x^{t+1}_i$ is nondecreasing in i (see Theorem 2.4), applying Lemma 2.2 yields

$$\sum_{j=1}^N \{p^t_{i_1j}(a_1) + p^t_{i_2j}(a_2)\}\{-x^{t+1}_j\} \geq \sum_{j=1}^N \{p^t_{i_1j}(a_2) + p^t_{i_2j}(a_1)\}\{-x^{t+1}_j\},$$

i.e.

$$\sum_{j=1}^N \{p^t_{i_1j}(a_1) + p^t_{i_2j}(a_2)\}x^{t+1}_j \leq \sum_{j=1}^N \{p^t_{i_1j}(a_2) + p^t_{i_2j}(a_1)\}x^{t+1}_j.$$

This implies the submodularity of $\sum_{j=1}^N p^t_{ij}(a)x^{t+1}_j$. Because $r^t_i(a)$ is submodular by Assumption 2.4 (B5) and because the sum of submodular functions is also submodular, $s^t_i(a)$ is a submodular function on $S \times A$. If the action $f^t(i)$ in formula (2.1) is not unique, take the largest optimal action. Then, applying Lemma 2.4 yields the result that $f^t(i)$ is nonincreasing in i. □

Algorithm 2.3 *Determination an optimal policy with monotone decision rules for a nonstationary MDP over T periods under the assumptions 2.3 and 2.4.*

Input: Instance of a finite nonstationary MDP, which satisfies assumptions 2.3 and 2.4, and the time horizon T.

Output: Optimal Markov policy $R_* = (f^1, f^2, \dots, f^T)$ with nonincreasing decision rules $f^t(i)$, $i \in S$, $1 \leq t \leq T$, and the value vector v^T.

```
1. $x := 0$.

2. for $t = T, T-1, \dots, 1$ do
```

```
begin  $A := \{1, 2, \dots, M\}$
       for $i = 1, 2, \dots, N$ do
       begin (1) take $f^t(i)$ such that $\{r(f^t) + P(f^t)x\}_i = max_{a \in A} \{r_i^t(a) + \sum_j p_{ij}^t(a)x_j\}$
                 (if there is more than one optimizing action, take the largest)
             (2) $y_i := \{r(f^t) + P(f^t)x\}_i$
             (3) $A := \{a \mid 1 \leq a \leq f^t(i)\}$
       end
   $x := y$
   end
```

3. $R_* = (f^1, f^2, \dots, f^T)$ is an optimal policy with nonincreasing decision rules $f^t(i),\ i \in S$, $1 \leq t \leq T$, and the value vector $v^T := x$.

Finally, we provide alternative conditions which lead to a nondecreasing optimal policy and for which Algorithm 2.2 can be used.

Assumption 2.5
(C1) $S = \{1, 2, \dots, N\}$, ordered in the natural way;

(C2) $r_i^t(a)$ is nonincreasing in i for all a and t;

(C3) $\sum_{j=k}^N p_{ij}^t(a)$ is nondecreasing in i for all k, a and t.

(C4) The action set $A(i) = A = \{1, 2, \dots, M\},\ i \in S$, where A is ordered in the natural way;

(C5) $r_i^t(a)$ is supermodular on $S \times A$ for $t = 1, 2, \dots, T$;

(C6) $\sum_{j=1}^N p_{ij}^t(a)u_j$ is supermodular on $S \times A$ for $t = 1, 2, \dots, T$ and for every nonincreasing function u on S.

Theorem 2.6
Let Assumption 2.5 hold. Then, there exists an optimal policy $R_* = (f^1, f^2, \dots, f^T)$, where $f^t(i)$ is nondecreasing in i for $t = 1, 2, \dots, T$.

Proof
Take any $1 \leq t \leq T$. By Theorem 2.5, x^{t+1} is nonincreasing on S. Hence, by condition (C6), $\sum_{j=1}^N p_{ij}^t(a)x_j^{t+1}$ is supermodular on $S \times A$. Because $r_i^t(a)$ is also supermodular by Assumption 2.5 (C5), $s_i^t(a) := r_i^t(a) + \sum_{j=1}^N p_{ij}^t(a)x_j^{t+1}$ is a supermodular function on $S \times A$. If the action $f^t(i)$ in formula (2.1) is not unique, take the largest optimal action. Then, applying Lemma 2.3 yields the result that $f^t(i)$ is nondecreasing in i. □

2.5 Bibliographic notes

The principles of optimality and backward induction were presented in Bellman's book [17]. This book had an enormous impact in the field of dynamic programming. Hordijk ([124]) has shown that the principle of optimality together with the validity of backward induction may be viewed as a consequence of the duality theory of linear programming.

The equivalence between the standard nonstationary finite horizon and a stationary inifinite horizon model was presented in Kallenberg ([146], [147]). A related paper is due to Derman and Klein ([70]).

The development of monotone optimal policies is provided by the work of Serfozo ([263]) and Topkis ([289]). Our presentation follows Puterman ([227], section 4.7). Other contributions are given e.g. by Ross ([239]) and Heyman and Sobel ([117]).

2.6 Exercises

Exercise 2.1

Consider a layered network: i.e. the set of vertices $V = V_1 \cup V_2 \cup \cdots \cup V_p$, where $V_1 = \{1\}$, $V_p = \{N\}$, and all arcs (i, j) satisfy if $i \in V_k$, then $j \in V_{k+1}$ for some $k = 1, 2, \ldots, p-1$. Let arc (i, j) has length l_{ij}. Show that the problem of finding the shortest path (and its length) from vertex 1 to vertex N can be modeled as an MDP over a finite horizon.

Exercise 2.2

Consider a scheduling problem as in Application 2.1 with the data:
$N = 4;\ p_1 = 1,\ p_2 = 2,\ p_3 = 3,\ p_4 = 4;\ c_1(t) = max(0, t-2),\ c_2(t) = max(0, t-7)$,
$c_3(t) = max(0, t-5)$ and $c_4(t) = max(0, t-6)$.
a. Draw the layered network for this scheduling problem;
b. Compute an optimal ordering of the jobs by backward induction.

Exercise 2.3

Suppose you have an employee and at the beginning of each month you can decide on his salary for that month: either a low salary ($ 2300) or a high salary ($ 3000). Knowing his salary, the employee can decide to send in his resignation immediately.
The probability that he sends in his resignation depends on his salary: 40% for a low salary and 20% for a high salary. When the employee quits, a temporary employee has to be hired immediately for $ 4000 per month. When you have a temporary employee you will advertise each month for a new permanent employee.
The probability to find a new permanent employee (who can start at the beginning of the following month and will receive the same salary conditions as he original employee) depends on the advertising budget: 70% for advertising budget $ 300 and 90% for advertising budget $ 600. Each month you have to decide which salary is offered to an employee and if the employee resigns you have to choose the advertising budget. What is for you an optimal policy if only the next six months are considered?

Exercise 2.4

Construct the corresponding infinite horizon model for the finite horizon MDP of Exercise 2.3.

Exercise 2.5

Show that the sum of supermodular functions is supermodular.

Exercise 2.6

Show that the following functions on $\mathbb{R}^1 \times \mathbb{R}^1$ are supermodular:
a. $f(x,y) = (x+y)^2$.
b. $f(x,y) = xy$.
c. $f(x,y) = g(x+y)$ for any convex function g.

Exercise 2.7

Let $f(x,y)$ be a function on $X \times Y$, where $X = Y = \mathbb{Z}_+$, and suppose

$$f(i+1, a+1) + f(i,a) \geq f(i, a+1) + f(i+1, a) \text{ for all } i \in X \text{ and } a \in Y.$$

Show that $f(x,y)$ is superadditive.

Exercise 2.8

Let $f(x,y)$ be a twice differential function on $\mathbb{R}^1 \times \mathbb{R}^1$. Show that $f(x,y)$ is superadditive if and only if $\frac{\partial^2 f(x,y)}{\partial x \partial y}$ is a nonnegative function.

Hint: Consider $\int_{y_2}^{y_1} \left\{ \int_{x_2}^{x_1} \frac{\partial^2 f(x,y)}{\partial x \partial y} dx \right\} dy$.

Exercise 2.9

Let X and Y be ordered sets. Suppose $f(x,y)$ is a superadditive function on $X \times Y$ and for each $x \in X$, $min\, f(x,y)$ exists. For a fixed $x \in X$, define $\underline{Y}(x) = \{y \in Y \mid y \in argmin\, f(x,y)\}$ and $\underline{y}(x) = min\{y \in \underline{Y}(x)\}$. Show that $\underline{y}(x)$ is nonincreasing in x.

Chapter 3

Discounted rewards

3.1 Introduction

This chapter deals with the total expected discounted reward over an infinite planning horizon. We assume that the model is stationary. The criterion of the total expected discounted reward is quite natural when the planning horizon is rather large and returns at the present time are of more value than returns which are earned later in time. We recall that the total expected α-discounted reward, given initial state i, policy R and discount factor $\alpha \in (0,1)$, is denoted by $v_i^\alpha(R)$ and defined by

$$v_i^\alpha(R) := \sum_{t=1}^\infty \mathbb{E}_{i,R}\{\alpha^{t-1} \cdot r_{X_t}(Y_t)\} = \sum_{t=1}^\infty \alpha^{t-1} \sum_{j,a} \mathbb{P}_{i,R}\{X_t = j, Y_t = a\} \cdot r_j(a). \tag{3.1}$$

As already mentioned in section 1.2.2, by the theorem of dominated convergence, the expected total α-discounted reward, i.e.

$$\mathbb{E}_{i,R}\Big\{\sum_{t=1}^\infty \alpha^{t-1} \cdot r_{X_t}(Y_t)\Big\},$$

gives the same expression as (3.1). Hence, the expected total discounted reward criterion and the total expected discounted reward criterion are equivalent. We also recall that a stationary policy

π^∞ satisfies

$$v^\alpha(\pi^\infty) = \sum_{t=1}^{\infty} \alpha^{t-1} P(\pi)^{t-1} r(\pi). \tag{3.2}$$

Since $\Big\{I - \alpha P(\pi)\Big\} \cdot \Big\{I + \alpha P(\pi) + \cdots + \{\alpha P(\pi)\}^{t-1}\Big\} = I - \{\alpha P(\pi)\}^t$ and $\{\alpha P(\pi)\}^t \to 0$ for $t \to \infty$, we obtain

$$\sum_{t=1}^{\infty} \Big\{\alpha P(\pi)\Big\}^{t-1} = \Big\{I - \alpha P(\pi)\Big\}^{-1} \text{ and } v^\alpha(\pi^\infty) = \Big\{I - \alpha P(\pi)\Big\}^{-1} r(\pi).$$

The α-discounted value vector v^α is defined by

$$v^\alpha := sup_R \; v^\alpha(R). \tag{3.3}$$

A policy R_* is an optimal policy if $v^\alpha(R_*) = v^\alpha$.

From the mathematical point of view, the discounted reward criterion is good manageable: there is a very complete general theory. In this chapter we only discuss the case of finite state space and finite action sets, but the results can be extended to a much higher level of generality.

In this chapter, we first discuss the theory of monotone contraction mappings in the context of MDPs. Then, the optimality equation, bounds for the value vector and suboptimal actions are considered. Next, the classical methods (policy iteration, linear programming, value iteration) and the hybrid method of modified policy iteration are studied. Then, we discuss under which conditions monotone optimal policies exist (section 3.9). We close this chapter with bibliographic notes and exercises.

3.2 Monotone contraction mappings

To find an optimal policy and the α-discounted value vector v^α, the *optimality equation*

$$x_i = max_{a \in A(i)} \Big\{r_i(a) + \alpha \sum_j p_{ij}(a) x_j\Big\}, \; i \in S, \tag{3.4}$$

plays a central role. In the next section we will show that v^α is the unique solution of this equation. For the moment, we give the following intuitive argumentation. Suppose that at time point $t = 1$, given that the system is in state i, action $a \in A(i)$ is chosen; furthermore, suppose that from $t = 2$ on an optimal policy is followed. Then, the total expected α-discounted reward is equal to $r_i(a) + \alpha \sum_j p_{ij}(a) v_j^\alpha$. Since any optimal policy obtains at least this amount, we have

$$v_i^\alpha \geq max_{a \in A(i)} \Big\{r_i(a) + \alpha \sum_j p_{ij}(a) v_j^\alpha\Big\}, \; i \in S.$$

On the other hand, let a_i be the action chosen in state i by an optimal policy. Then,

$$v_i^\alpha = r_i(a_i) + \alpha \sum_j p_{ij}(a_i) v_j^\alpha \leq max_{a \in A(i)} \Big\{r_i(a) + \alpha \sum_j p_{ij}(a) v_j^\alpha\Big\}, \; i \in S.$$

Hence,

$$v_i^\alpha = max_{a\in A(i)} \Big\{ r_i(a) + \alpha \sum_j p_{ij}(a) v_j^\alpha \Big\},\ i \in S,$$

i.e. v^α is a solution of (3.4) and v^α is a fixed-point of the mapping $U : \mathbb{R}^N \to \mathbb{R}^N$, defined by

$$(Ux)_i := max_{a\in A(i)} \Big\{ r_i(a) + \alpha \sum_j p_{ij}(a) x_j \Big\},\ i \in S. \tag{3.5}$$

We will show that U is a contraction mapping. Hence, by the general theory of contracting mappings, v^α is the unique solution of (3.4) and can be computed by value iteration.

Contraction mappings

Let X be a Banach space[1] with norm $\| \cdot \|$, and let B be a mapping on X to itself. B is called a *contraction mapping* if for some $\beta \in (0, 1)$

$$\|Bx - By\| \leq \beta \cdot \|x - y\| \text{ for all } x, y \in X. \tag{3.6}$$

The number β is called the *contraction factor* of B. An element $x \in X$ is said to be a *fixed-point* of B if $Bx^* = x^*$. The next theorem[2] ensures the existence of a unique fixed-point for contraction mappings in a Banach space.

Theorem 3.1 *Banach Fixed-point Theorem*
Let X be a Banach space and suppose $B : X \to X$ is a contraction mapping. Then,
(1) $x^* = lim_{n\to\infty} B^n x$ exists for every $x \in X$ and x^* is a fixed-point of B.
(2) x^* is the unique fixed-point of B.

The next theorem provides bounds on the distance between the fixed-point x^* and the elements $B^n x$ for $n = 0, 1, 2, \dots$.

Theorem 3.2
Let X be a Banach space and suppose $B : X \to X$ is a contraction mapping with contraction factor β and fixed-point x^*. Then,
(1) $\|x^* - B^n x\| \leq \beta(1-\beta)^{-1} \cdot \|B^n x - B^{n-1} x\| \leq \beta^n (1-\beta)^{-1} \cdot \|Bx - x\|\ \forall x \in X,\ n \in \mathbb{N}$.
(2) $\|x^* - x\| \leq (1-\beta)^{-1} \cdot \|Bx - x\|\ \forall x \in X$.

Proof
(1) For $m > n \geq 1$, we have

$$\begin{aligned}
\|B^m x - B^n x\| &\leq \beta \cdot \|B^{m-1} x - B^{n-1} x\| \\
&\leq \beta \cdot \{\|B^{m-1} x - B^{m-2} x\| + \|B^{m-2} x - B^{m-3} x\| + \cdots + \|B^n x - B^{n-1} x\|\} \\
&\leq \beta \cdot \{\beta^{m-n-1} + \beta^{m-n-2} + \cdots + 1\} \cdot \|B^n x - B^{n-1} x\| \\
&\leq \beta(1-\beta)^{-1} \cdot \|B^n x - B^{n-1} x\|.
\end{aligned}$$

[1] For a definition of Banach space, see textbooks on Functional Analysis or Appendix C in Puterman [227].

[2] For a proof of the theorem, see textbooks on Functional Analysis or Puterman [227], p.150.

Hence, since $B^m x^* = x^*$, we obtain

$$\begin{aligned} \|x^* - B^n x\| &= \|B^m x^* - B^n x\| \leq \|B^m x^* - B^m x\| + \|B^m x - B^n x\| \\ &\leq \beta^m \cdot \|x^* - x\| + \beta(1-\beta)^{-1} \cdot \|B^n x - B^{n-1} x\| \text{ for } m > n \geq 1. \end{aligned}$$

Letting $m \to \infty$ yields $\|x^* - B^n x\| \leq \beta(1-\beta)^{-1} \cdot \|B^n x - B^{n-1} x\|$.

Because $\|B^n x - B^{n-1} x\| \leq \beta^{n-1} \cdot \|Bx - x\|$, we obtain the second inequality $\beta(1-\beta)^{-1} \cdot \|B^n x - B^{n-1} x\| \leq \beta^n (1-\beta)^{-1} \cdot \|Bx - x\|$.

(2) Apply the triangle inequality and part (1) for $n = 1$:

$$\begin{aligned} \|x^* - x\| &\leq \|x^* - Bx\| + \|Bx - x\| \leq \beta(1-\beta)^{-1} \cdot \|Bx - x\| + \|Bx - x\| \\ &= (1-\beta)^{-1} \cdot \|Bx - x\|. \end{aligned}$$

□

Remark:

The above theorem implies that the convergence rate of $B^n x$ to the fixed-point x^* is at least linear (cf. Stoer and Bulirsch ([283] p.251)). This kind of convergence is also called *geometric convergence.*

Monotonicity

Let X be a partially ordered set and $B : X \to X$. The mapping B is called *monotone* if $x \leq y$ implies $Bx \leq By$.

Theorem 3.3

Let X be a partially ordered Banach space. Suppose that $B :\ X \to X$ is a monotone contraction mapping with fixed-point x^*. Then,

(1) $Bx \leq x$ implies $x^* \leq Bx \leq x$.

(2) $Bx \geq x$ implies $x^* \geq Bx \geq x$.

Proof

(1) By the monotonicity of B, with induction on n, it can easily be verified that $x \geq Bx \geq \cdots \geq B^n x,\ n \in \mathbb{N}$. Therefore, we have $x^* = lim_{n\to\infty} B^n x \leq Bx \leq x$.

(2) The proof is similar to the proof of part (1).

□

It is easy to verify that the Euclidian N-space $\mathbb{R}^N$ with norm $\|x\|_\infty := max_{1\leq i \leq N} |x_i|$ (*supremum norm*) and with ordering $x \leq y$ if $x_i \leq y_i$ for all $1 \leq i \leq N$ is a partially ordered Banach space. Also, for $x \in \mathbb{R}^N$, we have $x \leq \|x\|_\infty \cdot e$, where e is the vector with all elements equal to 1.

Lemma 3.1

(1) Let $B : \mathbb{R}^N \to \mathbb{R}^N$ be a monotone contraction mapping with contraction factor β, and let d be a scalar. Then, $x \leq y + d \cdot e$ implies $Bx \leq By + \beta \cdot |d| \cdot e$.

(2) Let $B : \mathbb{R}^N \to \mathbb{R}^N$ be a mapping with the property that $x \leq y + d \cdot e$ implies $Bx \leq By + \beta \cdot |d| \cdot e$ for some $0 \leq \beta < 1$ and for all scalars d. Then, with respect to the supremum norm, B is a monotone contraction with contraction factor β.

Proof

(1) From the monotonicity of B it follows that

$$\begin{aligned} Bx &\leq B(y+d\cdot e) = B(y+d\cdot e) - By + By \leq \|B(y+d\cdot e) - By\|_\infty \cdot e + By \\ &\leq \beta \cdot \|(y+d\cdot e) - y\|_\infty \cdot e + By = \beta \cdot |d| \cdot e + By. \end{aligned}$$

(2) Taking $d = 0$ yields the monotonicity. Since $x - y \leq \|x-y\|_\infty \cdot e$ and $y - x \leq \|x-y\|_\infty \cdot e$, the property of B, mentioned in part (2) of the theorem, implies that $Bx - By \leq \beta \cdot \|x-y\|_\infty \cdot e$ and $By - Bx \leq \beta \cdot \|x-y\|_\infty \cdot e$, which yields $\|Bx - By\|_\infty \leq \beta \cdot \|x-y\|_\infty$. □

Lemma 3.2

Let $B: \mathbb{R}^N \to \mathbb{R}^N$ be a monotone contraction mapping with respect to the supremum norm and with contraction factor $+\beta$ and fixed-point x^*. Suppose that there exist scalars a and b such that $a \cdot e \leq Bx - x \leq b \cdot e$ for some $x \in \mathbb{R}^N$. Then,

$$x - (1-\beta)^{-1}|a| \cdot e \leq Bx - \beta(1-\beta)^{-1}|a| \cdot e \leq x^* \leq Bx + \beta(1-\beta)^{-1}|b| \cdot e \leq x + (1-\beta)^{-1}|b| \cdot e.$$

Proof

Since $Bx \leq x + b \cdot e \leq x + |b| \cdot e$, it follows from the monotonicity of B that

$$\begin{aligned} B^2x &\leq B(x+|b|\cdot e) = B(x+|b|\cdot e) - Bx + Bx \leq Bx + \|B(x+|b|\cdot e) - Bx\|_\infty \cdot e \\ &\leq Bx + \beta|b| \cdot e \leq x + (1+\beta)|b| \cdot e. \end{aligned}$$

Using the same arguments it can be shown (with induction on n) that

$$B^n x \leq Bx + (\beta + \cdots + \beta^{n-1})|b| \cdot e \leq x + (1 + \beta + \cdots + \beta^{n-1})|b| \cdot e, \ n \in \mathbb{N}.$$

By letting $n \to \infty$,

$$x^* \leq Bx + \beta(1-\beta)^{-1}|b| \cdot e \leq x + (1-\beta)^{-1}|b| \cdot e.$$

Because $Bx \geq x + a \cdot e \geq x - |a| \cdot e$, an analogous reasoning shows that

$$x^* \geq Bx - \beta(1-\beta)^{-1}|a| \cdot e \leq x - (1-\beta)^{-1}|a| \cdot e.$$

□

Corollary 3.1

Let B be a monotone contraction in $\mathbb{R}^N$ with respect to the supremum norm and with contraction factor β and fixed-point x^*. Then,

$$\begin{aligned} x - (1-\beta)^{-1}\|Bx - x\|_\infty \cdot e &\leq Bx - \beta(1-\beta)^{-1}\|Bx - x\|_\infty \cdot e \leq x^* \\ &\leq Bx + \beta(1-\beta)^{-1}\|Bx - x\|_\infty \cdot e \leq x + (1-\beta)^{-1}\|Bx - x\|_\infty \cdot e. \end{aligned}$$

Proof

Notice that $-\|Bx - x\|_\infty \cdot e \leq Bx - x \leq \|Bx - x\|_\infty \cdot e$ and apply Lemma 3.2. □

Lemma 3.3

Let $B: \mathbb{R}^N \to \mathbb{R}^N$ be a monotone contraction mapping with respect to the supremum norm and with contraction factor β, fixed-point x^* and with the property that $B(x + c \cdot e) = Bx + \beta c \cdot e$ for every $x \in \mathbb{R}^N$ and scalar c.

Suppose that there exist scalars a and b such that $a \cdot e \leq Bx - x \leq b \cdot e$ for some $x \in \mathbb{R}^N$. Then,

$$x + (1-\beta)^{-1}a \cdot e \leq Bx + \beta(1-\beta)^{-1}a \cdot e \leq x^* \leq Bx + \beta(1-\beta)^{-1}b \cdot e \leq x + (1-\beta)^{-1}b \cdot e.$$

Proof
By the monotonicity of B it follows from $Bx \leq x + b \cdot e$ that

$$B^2x \leq B(x + b \cdot e) = Bx + \beta \cdot e \leq x + (1+\beta)b \cdot e.$$

By induction on n, we obtain

$$B^nx \leq Bx + (\beta + \beta^2 + \cdots + \beta^{n-1})b \cdot e \leq x + (1 + \beta + \beta^2 + \cdots + \beta^{n-1})b \cdot e.$$

Taking the limit for $n \to \infty$ gives,

$$x^* \leq Bx + \beta(1-\beta)^{-1}b \cdot e \leq x + (1-\beta)^{-1}b \cdot e.$$

The proof of the lower bounds is similar. □

3.3 The optimality equation

In this section we discuss the optimality equation (3.4) for the α-discounted value vector v^α. We show that v^α is the unique solution of (3.4). Furthermore, we will derive bounds for the value vector. By these bounds suboptimality tests can be formulated to exclude nonoptimal actions. The results are obtained by applying the theory of monotone contraction mappings, as presented in Section 3.2. Besides the mapping U, defined in (3.5), we introduce a mapping $L_\pi : \mathbb{R}^N \to \mathbb{R}^N$ for any randomized decision rule π, defined by

$$L_\pi x := r(\pi) + \alpha P(\pi)x. \tag{3.7}$$

Let $f_x(i)$ be such that

$$r_i\big(f_x(i)\big) + \alpha \sum_j p_{ij}\big(f_x(i)\big)x_j = max_a\Big\{r_i(a) + \alpha \sum_j p_{ij}(a)x_j\Big\},\ i \in S.$$

Then,

$$L_{f_x}x = Ux = max_f\ L_fx,$$

where the maximization is taken over all deterministic decision rules f. Let $\|P(\pi)\|_\infty$ be the subordinate matrix norm[3], then $\|P(\pi)\|_\infty$ satisfies (see e.g. Stoer and Boelirsch [283], p. 178)

$$\|P(\pi)\|_\infty = max_i \sum_j p_{ij}(\pi) = 1.$$

Theorem 3.4
The mappings L_π and U are monotone contraction mappings with respect to the supremum norm and with contraction factor α.

Proof
Suppose that $x \geq y$. Let π be any stationary decision rule. Because $P(\pi) \geq 0$,

$$L_\pi x = r(\pi) + \alpha P(\pi)x \geq r(\pi) + \alpha P(\pi)y = L_\pi y, \tag{3.8}$$

[3] Given a vector norm $\|x\|$, the corresponding subordinate matrix norm for a square matrix A is defined by $\|A\| = max_{\{x \mid \|x\|=1\}}\|Ax\|$.

i.e. L_π is monotone. U is also monotone, since $Ux = max_f \ L_f x \geq L_{f_y} x \geq L_{f_y} y = Uy$. Furthermore, we obtain

$$\|L_\pi x - L_\pi y\|_\infty = \|\alpha P(\pi)(x-y)\|_\infty \leq \alpha \cdot \|P(\pi)\|_\infty \cdot \|x-y\|_\infty = \alpha \cdot \|x-y\|_\infty,$$

i.e. L_π is a contraction with contraction factor α. For the mapping U we have,

$$Ux - Uy = L_{f_x} x - L_{f_y} y \leq L_{f_x} x - L_{f_x} y = \alpha \cdot P(f_x)(x-y) \leq \alpha \cdot \|x-y\|_\infty \cdot e. \tag{3.9}$$

Interchanging x and y yields

$$Uy - Ux \leq \alpha \cdot \|x-y\|_\infty \cdot e. \tag{3.10}$$

From (3.9) and (3.10) it follows that $\|Ux - Uy\|_\infty \leq \alpha \cdot \|x-y\|_\infty$, i.e. U is a contraction with contraction factor α. □

The next theorem shows that for any randomized decision rule π, the total expected α-discounted reward of the policy π^∞ is the fixed-point of the mapping L_π.

Theorem 3.5
$v^\alpha(\pi^\infty)$ is the unique solution of the functional equation $L_\pi x = x$.

Proof
Theorem 3.1 and Theorem 3.4 imply that it is sufficient to show that $L_\pi v^\alpha(\pi^\infty) = v^\alpha(\pi^\infty)$.
We have

$$\begin{aligned} L_\pi v^\alpha(\pi^\infty) - v^\alpha(\pi^\infty) = & \ r(\pi) - \{I - \alpha P(\pi)\} v^\alpha(\pi^\infty) \\ = & \ r(\pi) - \{I - \alpha P(\pi)\}\{I - \alpha P(\pi)\}^{-1} r(\pi) = r(\pi) - r(\pi) = 0. \end{aligned}$$ □

Corollary 3.2
$v^\alpha(\pi^\infty) = lim_{n\to\infty} \ L_\pi^n x$ for any $x \in \mathbb{R}^N$.

The next theorem shows that the value vector v^∞ is the fixed-point of the mapping U. The proof of this result is more complicated than the proof of Theorem 3.5

Theorem 3.6
v^α is the unique solution of the functional equation $Ux = x$.

Proof
It is sufficient to show that $Uv^\alpha = v^\alpha$. Let $R = (\pi^1, \pi^2, \dots)$ be an arbitrary Markov policy. Then,

$$\begin{aligned} v^\alpha(R) = & \ r(\pi^1) + \textstyle\sum_{t=2}^\infty \ \alpha^{t-1} P(\pi^1)P(\pi^2)\cdots P(\pi^{t-1}) r(\pi^t) \\ = & \ r(\pi^1) + \alpha P(\pi^1) \textstyle\sum_{s=1}^\infty \ \alpha^{s-1} P(\pi^2)P(\pi^3)\cdots P(\pi^s) r(\pi^{s+1}) \\ = & \ r(\pi^1) + \alpha P(\pi^1) v^\alpha(R_2) = L_{\pi^1} v^\alpha(R_2), \end{aligned}$$

where $R_2 = (\pi^2, \pi^3, \dots)$. From the monotonicity of L_{π^1} and the definition of U, we obtain

$$v^\alpha(R) = L_{\pi^1} v^\alpha(R_2) \leq L_{\pi^1} v^\alpha \leq U v^\alpha,\ R \in C(M).$$

Hence, $v^\alpha = sup_{R \in C(M)}\, v^\alpha(R) \leq U v^\alpha$.

In order to show the reverse inequality $v^\alpha \geq U v^\alpha$, take any $\varepsilon > 0$. Since $v^\alpha = sup_{R \in C(M)}\, v^\alpha(R)$, for any $j \in S$ there exists a Markov policy $R_j^\varepsilon = (\pi^1(j), \pi^2(j), \dots)$ such that $v_j^\alpha(R_j^\varepsilon) \geq v_j^\alpha - \varepsilon$. Let $a_i \in A(i)$ be such that $r_i(a_i) + \alpha \sum_j\ p_{ij}(a_i) v_j^\alpha = max_a\ \{r_i(a) + \alpha \sum_j\ p_{ij}(a) v_j^\alpha\},\ i \in S$. Consider the policy $R^* = (\pi^1, \pi^2, \dots)$ defined by

$$\pi_{ia}^1 := \begin{cases} 1 & \text{if } a = a_i \\ 0 & \text{otherwise} \end{cases} \quad \text{and } \pi_{i_1 a_1 \cdots i_t a}^t := \pi_{i_t a}^{t-1}(i_2),\ a \in A(i_t),\ t \geq 2.$$

So, R^* is the policy that chooses a_i in state i at time point $t = 1$, and if the state at time $t = 2$ is i_2, then the policy follows $R_{i_2}^\varepsilon$, where the process is considered to be originating in state i_2. Therefore,

$$\begin{aligned} v_i^\alpha \ &\geq\ v_i^\alpha(R^*) = r_i(a_i) + \alpha \textstyle\sum_j\ p_{ij}(a_i) v_j^\alpha(R_j^\varepsilon)\ \geq\ r_i(a_i) + \alpha \sum_j\ p_{ij}(a_i)(v_j^\alpha - \varepsilon) \\ &=\ max_a\ \{r_i(a) + \alpha \textstyle\sum_j\ p_{ij}(a) v_j^\alpha\} - \alpha \cdot \varepsilon = (U v^\alpha)_i - \alpha \cdot \varepsilon,\ i \in S. \end{aligned}$$

Since $\varepsilon > 0$ is arbitrarily chosen, $v^\alpha \geq U v^\alpha$. □

Because $v^\alpha = U v^\alpha = L_{f_{v^\alpha}} v^\alpha$, it follows from Theorem 3.5 that $v^\alpha = v^\alpha(f_{v^\alpha}^\infty)$, i.e. $f_{v^\alpha}^\infty$ is an optimal policy. If $f^\infty \in C(D)$ satisfies

$$r_i(f) + \alpha \sum_j\ p_{ij}(f) v_j^\alpha = max_a\ \{r_i(a) + \alpha \sum_j\ p_{ij}(a) v_j^\alpha\},\ i \in S,$$

then f^∞ is called a *conserving policy*. Conserving policies f^∞ satisfy $L_f v^\alpha = U v^\alpha = v^\alpha$ and are optimal policies. Therefore, the equation $Ux = x$ is called the *optimality equation*.

Corollary 3.3

(1) There exists a deterministic α-discounted optimal policy.

(2) $v^\alpha = lim_{n \to \infty}\ U^n x$ for any $x \in \mathbb{R}^N$.

(3) Any conserving policy is α-discounted optimal.

As already mentioned, we will derive bounds for the value vector v^α. These bounds can be obtained by using Lemma 3.3. Notice that the mappings L_π and U satisfy, for any $x \in R^N$ and any scalar c, $L_f(x + c \cdot e) = L_f x + \alpha c \cdot e$ and $U(x + c \cdot e) = Ux + \alpha c \cdot e$.

Theorem 3.7

For any $x \in \mathbb{R}^N$, we have

(1) $x - (1-\alpha)^{-1} \|Ux - x\|_\infty \cdot e \leq Ux - \alpha(1-\alpha)^{-1}\|Ux - x\|_\infty \cdot e \leq v^\alpha(f_x^\infty) \leq v^\alpha \leq$
$Ux + \alpha(1-\alpha)^{-1}\|Ux - x\|_\infty \cdot e \leq x + (1-\alpha)^{-1}\|Ux - x\|_\infty \cdot e.$

(2) $\|v^\alpha - x\|_\infty \leq (1-\alpha)^{-1}\|Ux - x\|_\infty$.

(3) $\|v^\alpha - v^\alpha(f_x^\infty)\|_\infty \leq 2\alpha(1-\alpha)^{-1}\|Ux - x\|_\infty$.

Proof

Take an arbitrary $x \in \mathbb{R}^N$. By Lemma 3.3, for $a = -\|Ux - x\|_\infty,\ b = \|Ux - x\|_\infty$ and $B = L_{f_x}$, and by the definition of v^α, we obtain (notice that $Bx = L_{f_x}x = Ux$),

$$x - (1-\alpha)^{-1}\|Ux - x\|_\infty \cdot e \le Ux - \alpha(1-\alpha)^{-1}\|Ux - x\|_\infty \cdot e \le v^\alpha(f_x^\infty) \le v^\alpha.$$

Next, applying Lemma 3.3 with $B = U$, gives the remaining part of (1), i.e.

$$v^\alpha \le Ux + \alpha(1-\alpha)^{-1}\|Ux - x\|_\infty \cdot e \le x + (1-\alpha)^{-1}\|Ux - x\|_\infty \cdot e.$$

The parts (2) and (3) follow directly from part (1). □

The next theorem provides a stronger bound for $\|v^\alpha - v^\alpha(f_x^\infty)\|_\infty$. This theorem uses the *span* of a vector $y \in \mathbb{R}^N$, which is defined by $span\,(y) := max_i\, y_i - min_i\, y_i$.

Theorem 3.8

For any $x \in \mathbb{R}^N$, we have

(1) $x + (1-\alpha)^{-1}min_i\ (Ux - x)_i \cdot e \le Ux + \alpha(1-\alpha)^{-1}min_i\ (Ux - x)_i \cdot e \le v^\alpha(f_x^\infty) \le v^\alpha \le$
$Ux + \alpha(1-\alpha)^{-1}max_i\,(Ux - x)_i \cdot e \le x + (1-\alpha)^{-1}max_i\, Ux - x)_i \cdot e.$

(2) $\|v^\alpha - v^\alpha(f_x^\infty)\|_\infty \le \alpha(1-\alpha)^{-1}span\,(Ux - x).$

Proof

Note that $min_i\,(Ux - x)_i \cdot e \le Ux - x \le max_i\,(Ux - x)_i \cdot e$. It is easy to verify that for $a = min_i\,(Ux - x)_i$ and $b = max_i\,(Ux - x)_i$ the proof is similar to the proof of Theorem 3.7. □

Remark

Since $min_i\,(Ux - x)_i \le \|Ux - x\|_\infty$ and $max_i\,(Ux - x)_i \le \|Ux - x\|_\infty$, we have the inequality $span\,(Ux - x) \le 2 \cdot \|Ux - x\|_\infty$. Consequently, the bound given by Theorem 3.8 part (2) is stronger than the bound given by Theorem 3.7 part (3).

Next, we discuss the elimination of suboptimal actions. An action $a \in A(i)$ is called *suboptimal* if there doesn't exist an α-discounted optimal policy $f^\infty \in C(D)$ with $f(i) = a$. Because f^∞ is α-discounted optimal if and only if $v^\alpha(f^\infty) = v^\alpha$, and because $v^\alpha = Uv^\alpha$, an action $a \in A(i)$ is suboptimal if and only if

$$v_i^\alpha > r_i(a) + \alpha \sum_j p_{ij}(a)v_j^\alpha. \tag{3.11}$$

Suboptimal actions can be disregarded. Notice that formula (3.11) is in some sense useless, because v^α is unknown. However, by upper and lower bounds on v^α as given in Theorems 3.7 and 3.8, suboptimality tests can be derived, as illustrated in the following theorem.

Theorem 3.9

Suppose that $x \le v^\alpha \le y$. If $r_i(a) + \alpha \sum_j p_{ij}(a)y_j < (Ux)_i$, then action $a \in A(i)$ is suboptimal.

Proof

$v_i^\alpha = (Uv^\alpha)_i \geq (Ux)_i > r_i(a) + \alpha \sum_j p_{ij}(a)y_j \geq r_i(a) + \alpha \sum_j p_{ij}(a)v_j^\alpha$. The first inequality is justified by the monotonicity of U. □

Corollary 3.4

Suppose that for some scalars b and c, we have $x + b \cdot e \leq v^\alpha \leq x + c \cdot e$. If

$$r_i(a) + \alpha \sum_j p_{ij}(a)x_j < (Ux)_i - \alpha(c-b), \tag{3.12}$$

then action $a \in A(i)$ is suboptimal.

Proof

$r_i(a) + \alpha \sum_j p_{ij}(a)(x_j + c) = r_i(a) + \alpha \sum_j p_{ij}(a)x_j + \alpha \cdot c < (Ux)_i + \alpha \cdot b = \{U(x + b \cdot e)\}_i$. □

Applying Corollary 3.4 on the bounds of v^α, derived in the Theorems 3.7 and 3.8, gives the following tests for the elimination of a suboptimal action $a \in A(i)$:

$$r_i(a) + \alpha \sum_j p_{ij}(a)x_j < (Ux)_i - 2\alpha(1-\alpha)^{-1}\|Ux - x\|_\infty. \tag{3.13}$$

$$r_i(a) + \alpha \sum_j p_{ij}(a)(Ux)_j < (U^2x)_i - 2\alpha^2(1-\alpha)^{-1}\|Ux - x\|_\infty. \tag{3.14}$$

$$r_i(a) + \alpha \sum_j p_{ij}(a)x_j < (Ux)_i - \alpha(1-\alpha)^{-1} span\,(Ux - x). \tag{3.15}$$

$$r_i(a) + \alpha \sum_j p_{ij}(a)(Ux)_j < (U^2x)_i - \alpha^2(1-\alpha)^{-1} span\,(Ux - x). \tag{3.16}$$

A suboptimality test T_1 is said to be *stronger* than a suboptimality test T_2 if every action that is excluded as being suboptimal by test T_2 is also excluded as suboptimal by test T_1. The following theorem is intuitively obvious.

Theorem 3.10

Suboptimality tests based on stronger bounds yield stronger tests.

Proof

Suppose that $x^1 \leq x^2 \leq v^\alpha \leq y^2 \leq y^1$. Assume that an action $a \in A(i)$ is suboptimal by a test based on x^1 and y^1, i.e. $r_i(a) + \sum_j p_{ij}(a)y_j^1 < (Ux^1)_i$. Then,

$$r_i(a) + \sum_j p_{ij}(a)y_j^2 \leq r_i(a) + \sum_j p_{ij}(a)y_j^1 < (Ux^1)_i \leq (Ux^2)_i,$$

i.e. a is also suboptimal by the test based on x^2 and y^2. □

Corollary 3.5

Suboptimality test (3.16) is stronger than any other test; both the tests (3.15) and (3.14) are stronger than test (3.13), but are not mutually comparable.

Proof

Since $-\|Ux-x\|_\infty \leq min_i\,(Ux-x)_i \leq max_i\,(Ux-x)_i \leq \|Ux-x\|_\infty$, we have

$$\begin{array}{llll}(1) & x-(1-\alpha)^{-1}\|Ux-x\|_\infty \cdot e & \leq & x+(1-\alpha)^{-1}min_i\,(Ux-x)_i \cdot e \leq v^\alpha \\ & & \leq & x+(1-\alpha)^{-1}max_i\,(Ux-x)_i \cdot e \\ & & \leq & x+(1-\alpha)^{-1}\|Ux-x\|_\infty \cdot e,\end{array}$$

implying that suboptimality test (3.15) is stronger that test (3.13).

$$\begin{array}{llll}(2) & Ux-\alpha(1-\alpha)^{-1}\|Ux-x\|_\infty \cdot e & \leq & Ux+\alpha(1-\alpha)^{-1}min_i\,(Ux-x)_i \cdot e \leq v^\alpha \\ & & \leq & Ux+\alpha(1-\alpha)^{-1}max_i\,(Ux-x)_i \cdot e \\ & & \leq & Ux+\alpha(1-\alpha)^{-1}\|Ux-x\|_\infty \cdot e,\end{array}$$

implying that suboptimality test (3.16) is stronger that test (3.14).

$$\begin{array}{llll}(3) & x-(1-\alpha)^{-1}\|Ux-x\|_\infty \cdot e & \leq & Ux+\alpha(1-\alpha)^{-1}\|Ux-x\|_\infty \cdot e \leq v^\alpha \\ & & \leq & Ux+\alpha(1-\alpha)^{-1}\|Ux-x\|_\infty \cdot e \\ & & \leq & x+(1-\alpha)^{-1}\|Ux-x\|_\infty \cdot e,\end{array}$$

implying that suboptimality test (3.14) is stronger that test (3.13).

$$\begin{array}{llll}(4) & x+(1-\alpha)^{-1}min_i\,(Ux-x)_i \cdot e & \leq & Ux+\alpha(1-\alpha)^{-1}min_i\,(Ux-x)_i \cdot e \leq v^\alpha \\ & & \leq & Ux+\alpha(1-\alpha)^{-1}max_i\,(Ux-x)_i \cdot e \\ & & \leq & x+(1-\alpha)^{-1}max_i\,(Ux-x)_i \cdot e,\end{array}$$

implying that suboptimality test (3.16) is stronger that test (3.15). □

Remark

In order to apply the tests (3.14) and (3.16) we need U^2x. However, in that case it is better to use the tests (3.13) and (3.15) with Ux instead of x, since $\|U^2x-Ux\|_\infty \leq \alpha \cdot \|Ux-x\|_\infty$ and *span* $(U^2x-Ux) \leq \alpha \cdot$ *span* $(Ux-x)$ (see Exercise 3.9).

3.4 Policy iteration

In the method of *policy iteration* a sequence of deterministic policies $f_1^\infty, f_2^\infty, \ldots$ is constructed such that

$$v^\alpha(f_{k+1}^\infty) > v^\alpha(f_k^\infty) \text{ for } k=1,2,\ldots \tag{3.17}$$

where $x>y$, for $x,y \in \mathbb{R}^N$, means that $x_i \geq y_i$ for every i and $x_i > y_i$ for at least one i. Because $C(D)$ is finite, the method of policy iteration is also finite. We will show that the method gives an α-discounted optimal policy upon termination.

For every $i \in S$ and $f^\infty \in C(D)$, the action set $A(i,f)$ is defined by

$$A(i,f) := \{a \in A(i) \mid r_i(a) + \alpha \sum_j p_{ij}(a) v_j^\alpha(f^\infty) > v_i^\alpha(f^\infty)\}. \tag{3.18}$$

The intuitive idea of the policy iteration method is that if action $f(i)$ is replaced by an action $a \in A(i,f)$, the resulting policy improves the total α-discounted rewards. Therefore, $A(i,f)$ are called the set of *improving actions*. In the next theorem we show the correctness of this notion.

Theorem 3.11

(1) If $A(i,f) = \emptyset$ for every $i \in S$, then f^∞ is an α-discounted optimal policy.

(2) If $A(i,f) \neq \emptyset$ for some $i \in S$, then $v^\alpha(g^\infty) > v^\alpha(f^\infty)$ for any $g^\infty \in C(D)$ with $g \neq f$ and $g(i) \in A(i,f)$ when $g(i) \neq f(i)$.

Proof

(1) Since $A(i,f) = \emptyset$, we have for every $i \in S$, $L_g v^\alpha(f^\infty) = r(g) + \alpha P(g) v^\alpha(f^\infty) \leq v^\alpha(f^\infty)$ for every deterministic decision rule g. By Theorem 3.3, this implies that $v^\alpha(g^\infty) \leq L_g v^\alpha(f^\infty) \leq v^\alpha(f^\infty)$ for every $g^\infty \in C(D)$, i.e. f^∞ is optimal.

(2) Take any $g \neq f$ such that $g(i) \in A(i,f)$ if $g(i) \neq f(i)$. Then, if $g(i) \neq f(i)$,

$$r_i(g) + \alpha \sum_j p_{ij}(g) v_j^\alpha(f^\infty) > v_i^\alpha(f^\infty). \tag{3.19}$$

If $g(i) = f(i)$,

$$r_i(g) + \alpha \sum_j p_{ij}(g) v_j^\alpha(f^\infty) = r_i(f) + \alpha \sum_j p_{ij}(f) v_j^\alpha(f^\infty) = v_i^\alpha(f^\infty), \tag{3.20}$$

the last equation by Theorem 3.5. From (3.19) and (3.20) it follows that

$$L_g v^\alpha(f^\infty) = r(g) + \alpha P(g) v^\alpha(f^\infty) > v^\alpha(f^\infty).$$

Hence, again by Theorem 3.3, we have $v^\alpha(g^\infty) \geq L_g v^\alpha(f^\infty) > v^\alpha(f^\infty)$. □

Algorithm 3.1 *Policy iteration algorithm*

Input: Instance of a discounted MDP.

Output: Optimal deterministic policy f^∞ and the value vector v^α.

```
1. Start with any $f^\infty \in C(D)$
2. Compute $v^\alpha(f^\infty)$ as the unique solution of the linear system $L_f x = x$
3. a. Compute $s_{ia}(f) := r_i(a) + \alpha \sum_j p_{ij}(a) v_j^\alpha(f^\infty) - v_i^\alpha(f^\infty)$ for all $(i,a) \in S \times A$
   b. Determine $A(i,f) := \{a \in A(i) \mid s_{ia}(f) > 0\}$ for all $i \in S$
4. if $A(i,f) = \emptyset$ for all $i \in S$ then go to step 6
   otherwise take any $g \neq f$ with $g(i) \in A(i,f)$ when $g(i) \neq f(i)$.
5. $f := g$ and return to step 2.
6. $f^\infty$ is an $\alpha$-discounted optimal policy and $v^\alpha(f^\infty)$ is the value vector $v^\alpha$ (STOP).
```

Remark

There is some freedom for the choice of g in step 4. A usual choice is to take g such that

$$s_{ig(i)}(f) = max_a \; s_{ia}(f), \; i \in S. \tag{3.21}$$

Then, for each $i \in S$: $g(i) = f(i)$ when $A(i,f) = \emptyset$ and $g(i) \in A(i,f)$ when $A(i,f) \neq \emptyset$.

Example 3.1

$\alpha = \frac{1}{2};\ S = \{1,2,3\};\ A(1) = A(2) = A(3) = \{1,2,3\};$

$r_1(1) = 1,\ r_1(2) = 2,\ r_1(3) = 3;\ r_2(1) = 6,\ r_2(2) = 4,\ r_2(3) = 5;\ r_3(1) = 8,\ r_3(2) = 9,\ r_3(3) = 7;$

$p_{11}(1) = 1;\ p_{12}(1) = 0;\ p_{13}(1) = 0;\ p_{11}(2) = 0;\ p_{12}(2) = 1;\ p_{13}(2) = 0;$

$p_{11}(3) = 0;\ p_{12}(3) = 0;\ p_{13}(3) = 1;\ p_{21}(1) = 1;\ p_{22}(1) = 0;\ p_{23}(1) = 0;$

$p_{21}(2) = 0;\ p_{22}(2) = 1;\ p_{23}(2) = 0;\ p_{21}(3) = 0;\ p_{22}(3) = 0;\ p_{23}(3) = 1;$

$p_{31}(1) = 1;\ p_{32}(1) = 0;\ p_{33}(1) = 0;\ p_{31}(2) = 0;\ p_{32}(2) = 1;\ p_{33}(2) = 0;$

$p_{31}(3) = 0;\ p_{32}(3) = 0;\ p_{33}(3) = 1.$

Start with the policy f, with $f(1) = 3, f(2) = 2$ and $f(3) = 1$.

In step 4 of the algorithm we will take g such that $s_{ig(i)}(f) = max_a\, s_{ia}(f),\ i \in S$.

Iteration 1

The system $L_f\, x = x$ becomes:

$$\begin{array}{rcrcrcl} x_1 & & & - & \frac{1}{2}x_3 & = & 3 \\ & & \frac{1}{2}x_2 & & & = & 4 \\ -\frac{1}{2}x_1 & & & + & x_3 & = & 8 \end{array} \quad \longrightarrow \quad \begin{array}{l} \text{solution: } v^\alpha(f^\infty) = (\frac{28}{3}, 8, \frac{38}{3}). \\ s_{11}(f) = -\frac{11}{3},\ s_{12}(f) = -\frac{10}{3},\ s_{13}(f) = 0. \\ s_{21}(f) = \frac{8}{3},\ s_{22}(f) = 0,\ s_{23}(f) = \frac{10}{3}. \\ s_{31}(f) = 0,\ s_{32}(f) = \frac{1}{3},\ s_{33}(f) = \frac{2}{3}. \end{array}$$

$A(1,f) = \emptyset;\ A(2,f) = \{1,3\};\ A(3,f) = \{2,3\}.$

$g(1) = g(2) = g(3) = 3$, which becomes the new policy: $f(1) = f(2) = f(3) = 3$.

Iteration 2

The system $L_f\, x = x$ becomes:

$$\begin{array}{rcrcrcl} x_1 & & & - & \frac{1}{2}x_3 & = & 3 \\ & & x_2 & - & \frac{1}{2}x_3 & = & 5 \\ & & & & \frac{1}{2}x_3 & = & 7 \end{array} \quad \longrightarrow \quad \begin{array}{l} \text{solution: } v^\alpha(f^\infty) = (10, 12, 14). \\ s_{11}(f) = -4,\ s_{12}(f) = -2,\ s_{13}(f) = 0. \\ s_{21}(f) = -1,\ s_{22}(f) = -2,\ s_{23}(f) = 0. \\ s_{31}(f) = -1,\ s_{32}(f) = 1,\ s_{33}(f) = 0. \end{array}$$

$A(1,f) = \emptyset;\ A(2,f) = \emptyset;\ A(3,f) = \{2\}.$

$g(1) = g(2) = 3$ and $g(3) = 2$, which becomes the new policy: $f(1) = f(2) = 3$ and $f(3) = 2$.

Iteration 3

The system $L_f x = x$ becomes:

$$\begin{array}{rcrcrcl} x_1 & & & - & \frac{1}{2}x_3 & = & 3 \\ & & x_2 & - & \frac{1}{2}x_3 & = & 5 \\ & - & \frac{1}{2}x_2 & + & x_3 & = & 9 \end{array} \quad \longrightarrow \quad \begin{array}{l} \text{solution: } v^\alpha(f^\infty) = (\frac{32}{3}, \frac{38}{3}, \frac{46}{3}). \\ s_{11}(f) = -\frac{11}{3},\ s_{12}(f) = -\frac{7}{3},\ s_{13}(f) = 0. \\ s_{21}(f) = -\frac{4}{3},\ s_{22}(f) = -\frac{7}{3},\ s_{23}(f) = 0. \\ s_{31}(f) = -2,\ s_{32}(f) = 0,\ s_{33}(f) = -\frac{2}{3}. \end{array}$$

$A(1,f) = \emptyset;\ A(2,f) = \emptyset;\ A(3,f) = \emptyset\}.$

f^∞ with $f(1) = f(2) = 3$ and $f(3) = 2$ is an optimal policy and $v^\alpha(f^\infty) = v^\alpha = (\frac{32}{3}, \frac{38}{3}, \frac{46}{3})$ is the value vector.

We now discuss the elimination of suboptimal actions with test (3.15) and $x = v^\alpha(f^\infty)$. Since

$$(Ux - x)_i = max_a\, \{r_i(a) + \sum_j p_{ij}(a) v_j^\alpha(f^\infty)\} - v_i^\alpha(f^\infty) = max_a\, s_{ia}(f),\ i \in S,$$

and $span(Ux - x) = max_i\,\{max_a\, s_{ia}(f)\} - min_i\,\{max_a\, s_{ia}(f)\}$, (3.15) becomes

$$s_{ia}(f) < max_a\, s_{ia}(f) - \alpha(1-\alpha)^{-1}\big\{max_i\{max_a\, s_{ia}(f)\} - min_i\,\{max_a\, s_{ia}(f)\}\big\},$$

resulting in the following theorem.

Theorem 3.12 *(Suboptimality test)*
If $s_{ia_i}(f) < max_a\, s_{ia}(f) - \alpha(1-\alpha)^{-1}\big\{max_i\,\{max_a\, s_{ia}(f)\} - min_i\,\{max_a\, s_{ia}(f)\}\big\}$, then action $a_i \in A(i)$ is a suboptimal action.

Remark
Since $s_{if(i)}(f) = 0,\ i \in S$, we have $max_i\{max_a s_{ia}(f)\} \geq min_i\{max_a s_{ia}(f)\} \geq min_i s_{if(i)}(f) = 0$.

Algorithm 3.2 *Policy iteration algorithm with suboptimality test (3.15) and using (3.21)*
Input: Instance of a discounted MDP.
Output: Optimal deterministic policy f^∞ and the value vector v^α.

```
1. Start with any $f^\infty \in C(D)$.
2. Compute $v^\alpha(f^\infty)$ as the unique solution $x$ of the linear system $L_f x = x$.
3. a. Compute $s_{ia}(f) := r_i(a) + \alpha \sum_j\, p_{ij}(a) v_j^\alpha(f^\infty) - v_i^\alpha(f^\infty)$ for all $(i,a) \in S \times A$.
   b. Determine $A(i,f) := \{a \in A(i) \mid s_{ia}(f) > 0\}$ for every $i \in S$.
4. if $A(i,f) = \emptyset$ for all $i \in S$ then go to step 7.
   otherwise take $g$ such that $s_{ig(i)}(f) = max_a\, s_{ia}(f),\ i \in S$.
5. $A(i) := \big\{a \mid s_{ia}(f) \geq max_a\, s_{ia}(f) - \alpha(1-\alpha)^{-1}\big\{max_i\,\{max_a\, s_{ia}(f)\} - min_i\,\{max_a\, s_{ia}(f)\}\big\}$
   for all $i \in S$.
6. $f := g$ and return to step 2.
7. $f^\infty$ is an $\alpha$-discounted optimal policy and $v^\alpha(f^\infty)$ is the value vector $v^\alpha$ (STOP).
```

Example 3.1 (continued)

Start with the policy f, with $f(1) = 3, f(2) = 2$ and $f(3) = 1$.

Iteration 1

The system $L_f\, x = x$ becomes:

$$\begin{array}{rcccrcl} x_1 & & & - & \frac{1}{2}x_3 & = & 3 \\ & & \frac{1}{2}x_2 & & & = & 4 \\ -\frac{1}{2}x_1 & & & + & x_3 & = & 8 \end{array} \quad \rightarrow \quad \begin{array}{l} \text{solution: } v^\alpha(f^\infty) = (\frac{28}{3}, 8, \frac{38}{3}). \\ s_{11}(f) = -\frac{11}{3},\ s_{12}(f) = -\frac{10}{3},\ s_{13}(f) = 0. \\ s_{21}(f) = \frac{8}{3},\ s_{22}(f) = 0,\ s_{23}(f) = \frac{10}{3}. \\ s_{31}(f) = 0,\ s_{32}(f) = \frac{1}{3},\ s_{33}(f) = \frac{2}{3}. \end{array}$$

$A(1,f) = \emptyset;\ A(2,f) = \{1,3\};\ A(3,f) = \{2,3\}$.

$g(1) = g(2) = g(3) = 3$, which becomes the new policy: $f(1) = f(2) = f(3) = 3$.

$\alpha(1-\alpha)^{-1}\big\{max_i\,\{max_a\, s_{ia}(f)\} - min_i\,\{max_a\, s_{ia}(f)\}\big\} = \frac{10}{3}$.

In state 1, action 1 is excluded, because $-\frac{11}{3} = s_{11}(f) < max_a\; s_{1a}(f) - \frac{10}{3} = -\frac{10}{3}$.

$A(1) = \{2,3\};\ A(2) = \{1,2,3\};\ A(3) = \{1,2,3\}$.

Iteration 2

The system $L_f\, x = x$ becomes:

$$\begin{array}{ccccl} x_1 & & - & \frac{1}{2}x_3 = & 3 \\ & x_2 & - & \frac{1}{2}x_3 = & 5 \\ & & & \frac{1}{2}x_3 = & 7 \end{array} \quad \longrightarrow$$

solution: $v^\alpha(f^\infty) = (10, 12, 14)$.
$s_{12}(f) = -2,\ s_{13}(f) = 0$.
$s_{21}(f) = -1,\ s_{22}(f) = -2,\ s_{23}(f) = 0$.
$s_{31}(f) = -1,\ s_{32}(f) = 1,\ s_{33}(f) = 0$.

$A(1,f) = \emptyset;\ A(2,f) = \emptyset;\ A(3,f) = \{2\}$.

$g(1) = g(2) = 3$ and $g(3) = 2$, which becomes the new policy: $f(1) = f(2) = 3$ and $f(3) = 2$.

$\alpha(1-\alpha)^{-1}\big\{max_i\,\{max_a\, s_{ia}(f)\} - min_i\,\{max_a\, s_{ia}(f)\}\big\} = 1$.

In state 1, action 2 is excluded, because $-2 = s_{12}(f) < max_a\; s_{1a}(f) - 1 = -1$.

In state 2, action 2 is excluded, because $-2 = s_{22}(f) < max_a\; s_{2a}(f) - 1 = -1$.

In state 3, action 1 is excluded, because $-1 = s_{31}(f) < max_a\; s_{3a}(f) - 1 = 0$.

$A(1) = \{3\};\ A(2) = \{1,3\};\ A(3) = \{2,3\}$.

Iteration 3

The system $L_f x = x$ becomes:

$$\begin{array}{cccccl} x_1 & & & - & \frac{1}{2}x_3 = & 3 \\ & & x_2 & - & \frac{1}{2}x_3 = & 5 \\ & - & \frac{1}{2}x_2 & + & x_3 = & 9 \end{array} \quad \longrightarrow$$

solution: $v^\alpha(f^\infty) = (\frac{32}{3}, \frac{38}{3}, \frac{46}{3})$.
$s_{13}(f) = 0$.
$s_{21}(f) = -\frac{4}{3},\ s_{23}(f) = 0$.
$s_{32}(f) = 0,\ s_{33}(f) = -\frac{2}{3}$.

$A(1,f) = \emptyset;\ A(2,f) = \emptyset;\ A(3,f) = \emptyset$.

f^∞ with $f(1) = f(2) = 3$ and $f(3) = 2$ is an optimal policy and $v^\alpha(f^\infty) = v^\alpha = (\frac{32}{3}, \frac{38}{3}, \frac{46}{3})$ is the value vector.

Next, we show that the policy iteration algorithm with (3.21) for the following policy is equivalent to *Newton's method* for solving the optimality equation $Ux = x$. Furthermore, we can make a statement about the convergence rate.

The choice (3.21) implies that $r(g) + \alpha P(g)v^\alpha(f^\infty) - v^\alpha(f^\infty) = Uv^\alpha(f^\infty) - v^\alpha(f^\infty)$, i.e.

$$L_g v^\alpha(f^\infty) = Uv^\alpha(f^\infty) \text{ and } g = f_{v^\alpha(f^\infty)}. \tag{3.22}$$

Define the operator F by

$$F : \mathbb{R}^N \to \mathbb{R}^N \text{ by } Fx = Ux - x. \tag{3.23}$$

Hence, v^α is the unique solution of the equation $Fx = 0$. Since $L_{f_x}x = Ux$, it follows that

$$Fx = L_{f_x}x - x = r(f_x) + \alpha P(f_x)x - x. \tag{3.24}$$

Suppose that Newton's method is applied to solve the equation $Fx = 0$. This method works as follows: starting with the vector x^1, the successive values $x^2, x^3, \dots$ are computed by the formula

$$x^{n+1} = x^n - \{\nabla F x^n\}^{-1} F x^n, \tag{3.25}$$

where ∇F is the *Jacobian* of F, i.e. $\nabla F x^n$ is an $N \times N$ matrix defined by

$$\big\{\nabla F x^n\big\}_{ij} = \left\{\frac{\partial (Fx)_i}{\partial x_j}\right\}_{x=x^n}.$$

From (3.24) it follows that $\nabla F x^n = \alpha P(f_{x^n}) - I$, where we assume that $r(f_x)$ and $P(f_x)$ are constant in a small neighbourhood of x^n. Hence, (3.25) can be written as

$$x^{n+1} = x^n + \{I - \alpha P(f_{x^n})\}^{-1}\Big\{r(f_{x^n}) - \{I - \alpha P(f_{x^n})\}x^n\Big\} = x^n + v^\alpha(f_{x^n}^\infty) - x^n = v^\alpha(f_{x^n}^\infty). \tag{3.26}$$

Theorem 3.13

Suppose that $f_1^\infty, f_2^\infty, \dots, f_p^\infty$ are the policies obtained by the policy iteration algorithm with (3.21) for the following policy. On the other hand, suppose that Newton's method is applied in order to solve the equation $Fx = 0$ with starting vector $x^1 = v^\alpha(f_1^\infty)$. Then,

(1) $x^n = v^\alpha(f_n^\infty),\ n = 1, 2, \dots, p.$

(2) $\|v^\alpha - v^\alpha(f_{n+1}^\infty)\|_\infty \le 2\alpha(1-\alpha)^{-1}\|v^\alpha - v^\alpha(f_n^\infty)\|_\infty,\ n = 1, 2, \dots, p-1.$

Proof

(1) We apply induction on n (the result is obvious for $n = 1$). Suppose that $x^n = v^\alpha(f_n^\infty)$, then we have to show that $x^{n+1} = v^\alpha(f_{n+1}^\infty)$. By (3.26) and the induction hypothesis,

$$x^{n+1} = v^\alpha(f_{x^n}^\infty) = v^\alpha(f_{v^\alpha(f_n^\infty)}^\infty). \tag{3.27}$$

It follows from (3.22) that $f_{n+1} = f_{v^\alpha(f_n^\infty)}$. Hence, by (3.27),

$$x^{n+1} = v^\alpha(f_{v^\alpha(f_n^\infty)}^\infty) = v^\alpha(f_{n+1}^\infty). \tag{3.28}$$

(2) $0 \le v^\alpha - v^\alpha(f_{n+1}^\infty) = v^\alpha - x^{n+1} = v^\alpha - x^n - \{I - \alpha P(f_{x^n})\}^{-1} F x^n$, and

$$\begin{aligned} Fx^n &= Ux^n - x^n = Ux^n - x^n - Uv^\alpha + v^\alpha \ge L_{f_{v^\alpha}} x^n - x^n - Uv^\alpha + v^\alpha \\ &= L_{f_{v^\alpha}} x^n - x^n - L_{f_{v^\alpha}} v^\alpha + v^\alpha = \{I - \alpha P(f_{v^\alpha})\}(v^\alpha - x^n). \end{aligned}$$

Hence,

$$\begin{aligned} 0 &\le v^\alpha - v^\alpha(f_{n+1}^\infty) \le v^\alpha - x^n - \{I - \alpha P(f_{x^n})\}^{-1}\{I - \alpha P(f_{v^\alpha})\}(v^\alpha - x^n) \\ &= \{I - \alpha P(f_{x^n})\}^{-1}\{I - \alpha P(f_{x^n})\}(v^\alpha - x^n) - \{I - \alpha P(f_{x^n})\}^{-1}\{I - \alpha P(f_{v^\alpha})\}(v^\alpha - x^n) \\ &= \{I - \alpha P(f_{x^n})\}^{-1}\{\alpha P(f_{v^\alpha}) - \alpha P(f_{x^n})\}(v^\alpha - x^n) \\ &= \alpha \cdot \{I - \alpha P(f_{x^n})\}^{-1}\{P(f_{v^\alpha}) - P(f_{x^n})\}(v^\alpha - x^n). \end{aligned}$$

Consequently,

$$\begin{aligned}
\|v^\alpha - v^\alpha(f_{n+1}^\infty)\|_\infty &\leq \alpha \cdot \|\{I - \alpha P(f_{x^n})\}^{-1}\|_\infty \cdot \|P(f_{v^\alpha}) - P(f_{x^n})\|_\infty \cdot \|v^\alpha - v^\alpha(f_n^\infty)\|_\infty \\
&= \alpha \cdot \| \textstyle\sum_{t=0}^\infty \, [\alpha P(f_{x^n})]^t\|_\infty \cdot \|P(f_{v^\alpha}) - P(f_{x^n})\|_\infty \cdot \|v^\alpha - v^\alpha(f_n^\infty)\|_\infty \\
&\leq \alpha(1-\alpha)^{-1}\|P(f_{v^\alpha}) - P(f_{x^n})\|_\infty \cdot \|v^\alpha - v^\alpha(f_n^\infty)\|_\infty \\
&= 2\alpha(1-\alpha)^{-1}\|v^\alpha - v^\alpha(f_n^\infty)\|_\infty. \qquad \square
\end{aligned}$$

Remark

In the last line of the proof the inequality $\|P(f_{v^\alpha}) - P(f_{x^n})\|_\infty \leq 2$ is used. This is a theoretical bound. Usually, $\|P(f_{v^\alpha}) - P(f_{x^n})\|_\infty$ is much smaller and for large n this norm tends to zero.

In general, the solution of the linear system $L_f x = x$ by Gauss elimination in step 2 of the policy iteration algorithm needs $\mathcal{O}(N^3)$ operations (cf. Stoer and Bulirsch ([283] pp. 169-172). However, by applying the next theorem, we will show that this evaluation can be done in $\mathcal{O}(mN^2)$ operations, where m is the number of states i in which $g(i) \neq f(i)$ with g the decision rule in step 4 of the policy iteration algorithm 3.1 or 3.2.

Theorem 3.14

$(B + UV^t)^{-1} = B^{-1} - B^{-1}U(I + V^tB^{-1}U)^{-1}V^tB^{-1}$, *assuming each of the inverses exists and that the matrices have the appropriate dimensions. In this expression,* V^t *denotes the transpose of matrix* V.

Proof

Let $T = (I + V^tB^{-1}U)^{-1}$, then

$$\begin{aligned}
(B + UV^t)(B^{-1} - B^{-1}UTV^tB^{-1}) &= I - UTV^tB^{-1} + UV^tB^{-1} - UV^tB^{-1}UTV^tB^{-1} \\
&= I - UTV^tB^{-1} + UV^tB^{-1} - U(T^{-1} - I)TV^tB^{-1} \\
&= I - UTV^tB^{-1} + UV^tB^{-1} - UV^tB^{-1} + UTV^tB^{-1} \\
&= I. \qquad \square
\end{aligned}$$

Let $\{i \mid g(i) \neq f(i)\} = \{i_1, i_2, \dots, i_m\}$, U the $N \times m$ matrix $\{e_{i_1}, e_{i_2}, \dots, e_{i_m}\}$, where e_{i_k} is the i_k-th unit vector in $\mathbb{R}^N$, and V is the $N \times m$ matrix $\{v^1, v^2, \dots, v^m\}$, where $v_l^k = -\alpha\{p_{i_k l}(g) - p_{i_k l}(f)\}$, $1 \leq k \leq m$, $1 \leq l \leq N$. Then,

$$(UV^t)_{kj} = \textstyle\sum_l \, u_{kl}v_{jl} = \begin{cases} v_j^k = -\alpha\{p_{i_k j}(g) - p_{i_k j}(f)\} & k = i_1, i_2, \dots, i_m, \; j \in S; \\ 0 & k \neq i_1, i_2, \dots, i_m, \; j \in S. \end{cases}$$

Hence, $I - \alpha P(g) = I - \alpha P(f) + UV^t$. Applying Theorem 3.14 yields the next result.

Theorem 3.15

If $I + V^t\{I - \alpha P(f)\}^{-1}U$ *is nonsingular, then*

$$\{I - \alpha P(g)\}^{-1} = \{I - \alpha P(f)\}^{-1} - \{I - \alpha P(f)\}^{-1}U\{I + V^t\{I - \alpha P(f)\}^{-1}U\}^{-1}V^t\{I - \alpha P(f)\}^{-1}.$$

Corollary 3.6
If $\{I-\alpha P(f)\}^{-1}$ is known, then $\{I-\alpha P(g)\}^{-1}$ can be computed in $\mathcal{O}(mN^2)$ operations.

Proof
‘Given $\{I-\alpha P(f)\}^{-1}$, $\{I-\alpha P(g)\}^{-1}$ can be computed as follows:

1. $Y_1 = V^t\{I-\alpha P(f)\}^{-1}$: mN^2 operations;
2. $Y_2 = Y_1U$: m^2N operations;
3. $Y_3 = I + Y_2$: m operations;
4. $Y_4 = Y_3^{-1}$: m^3 operations;
5. $Y_5 = Y_4Y_1$: m^2N operations;
6. $Y_6 = \{I-\alpha P(f)\}^{-1}U$: mN^2 operations;
7. $Y_7 = Y_6Y_5$: mN^2 operations;
8. $Y_8 = \{I-\alpha P(f)\}^{-1} - Y_7$: N^2 operations.

Hence, the overall complexity is $\mathcal{O}(mN^2)$. □

Remark
By Corollary 3.6, the computation of $v^\alpha(g^\infty) = \{I-\alpha P(g)\}^{-1}r(g)$ (step 2 of the policy iteration algorithm) requires also $\mathcal{O}(mN^2)$ operations. Because the computation of one $s_{ia}(f)$ in step 3 of the algorithm requires $\mathcal{O}(N)$ operations, one iteration of the policy iteration algorithm has complexity $\mathcal{O}\big(N(mN+M)\big)$, where $M := max_i\,|A(i)|$.

3.5 Linear programming

The value vector v^α is the unique solution of the optimality equation (3.4), i.e.

$$v_i^\alpha = max_{a\in A(i)}\ \{r_i(a) + \alpha \textstyle\sum_j\ p_{ij}(a)v_j^\alpha\},\ i \in S.$$

Hence, v^α satisfies

$$v_i^\alpha \ \geq\ r_i(a) + \alpha \sum_j\ p_{ij}(a)v_j^\alpha \text{ for all } (i,a) \in S\times A. \tag{3.29}$$

Intuitively it is clear that v^α is the smallest vector satisfying (3.29). This is the key property for the linear programming approach towards computing the value vector. It turns out that an optimal policy can be obtained from the dual linear program. We also show a one-to-one correspondence between the stationary policies and the feasible solutions of the dual program, such that the extreme points correspond to deterministic policies. Furthermore, we show that the linear programming method for discounted MDPs can be considered as equivalent to the policy iteration method, and that exclusion of suboptimal actions can also be included in the linear programming method.

A vector $v \in \mathbb{R}^N$ is said to be *α-superharmonic* if

$$v_i \ \geq\ r_i(a) + \alpha \sum_j\ p_{ij}(a)v_j \text{ for all } (i,a) \in S\times A. \tag{3.30}$$

Theorem 3.16
v^α is the smallest α-superharmonic vector (componentwise).

Proof

Since

$$v_i^\alpha = max_{a \in A(i)} \{r_i(a) + \alpha \sum_j \; p_{ij}(a) v_j^\alpha\} \geq r_i(a) + \alpha \sum_j \; p_{ij}(a) v_j^\alpha \text{ for all } (i,a) \in S \times A,$$

v^α is α-superharmonic. Suppose that $v \in \mathbb{R}^N$ is also α-superharmonic. Then,

$$v \geq r(f) + \alpha P(f) v \text{ for every } f^\infty \in C(D),$$

which implies $\{I - \alpha P(f)\} v \geq r(f)$. Since $\{I - \alpha P(f)\}^{-1} = \sum_{t=0}^\infty \; \alpha^t P^t(f) \geq 0$, we obtain

$$v \geq \{I - \alpha P(f)\}^{-1} r(f) = v^\alpha(f^\infty) \text{ for all } f^\infty \in C(D).$$

Hence, $v_i^\alpha = max_f \; v^\alpha(f^\infty) \leq v$, i.e. v^α is the smallest α-superharmonic vector. □

Corollary 3.7

v^α is the unique optimal solution of the linear programming problem

$$min \left\{ \sum_j \; \beta_j v_j \;\middle|\; \sum_j \{\delta_{ij} - \alpha p_{ij}(a)\} v_j \geq r_i(a), \;\; (i,a) \in S \times A \right\} \tag{3.31}$$

where β_j is any strictly positive number for every $j \in S$.

Proof

From Theorem 3.16 it follows that v^α is a feasible solution of (3.31) and that $v^\alpha \leq v$ for every feasible solution v of (3.31). Hence, v^α is the unique optimal solution of (3.31). □

By Corollary 3.7, the value vector v^α can be found as optimal solution of the linear program (3.31). This program does not give an optimal policy. However, the next theorem verifies that an optimal policy can be obtained from the solution of the dual program, which is:

$$max \left\{ \sum_{(i,a)} r_i(a) x_i(a) \;\middle|\; \begin{array}{rcl} \sum_{(i,a)} \{\delta_{ij} - \alpha p_{ij}(a)\} x_i(a) & = & \beta_j, \; j \in S \\ x_i(a) & \geq & 0, \; (i,a) \in S \times A \end{array} \right\}. \tag{3.32}$$

Theorem 3.17

(1) Any feasible solution x of (3.32) satisfies $\sum_a \; x_j(a) > 0$ for all $j \in S$.

(2) The dual program (3.32) has a finite optimal solution, say x^*.

(3) Any $f_*^\infty \in C(D)$ with $x_i^*\big(f_*(i)\big) > 0$ for every $i \in S$ is an α-discounted optimal policy.

Proof

(1) Let x be a feasible solution of (3.32). From the constraints of (3.32) it follows that

$$\sum_a x_j(a) = \beta_j + \alpha \sum_{(i,a)} p_{ij}(a) x_i(a) \geq \beta_j > 0 \text{ for all } j \in S.$$

(2) Since the primal program (3.31) has a finite optimal solution, namely the value vector v^α,

it follows from the theory of linear programming that the dual program (3.32) also has a finite optimal solution.

(3) Take any $f_*^\infty \in C(D)$ with $x_i^*\big(f_*(i)\big) > 0$ for every $i \in S$ (such policy exists by part (1)). Because $x_i^*\big(f_*(i)\big) > 0,\ i \in S$, the complementary slackness property of linear programming implies

$$\sum_j \{\delta_{ij} - \alpha p_{ij}(f_*)\} v_j^\alpha = r_i(f_*),\ i \in S.$$

Hence, in vector notation,

$$\{I - \alpha P(f_*)\} v^\alpha = r(f_*), \text{ which implies } v^\alpha = \{I - \alpha P(f_*)\}^{-1} r(f_*) = v^\alpha(f_*^\infty),$$

i.e. f_*^∞ is an α-discounted optimal policy. □

Observe that the primal program has N columns and $\sum_i |A(i)|$ rows, while the dual program has $\sum_i |A(i)|$ rows and N columns.

If the simplex method is used, then the programs (3.31) and (3.32) are solved simultaneously. Hence, by the simplex method both the value vector v^α and an optimal policy are computed.

Next, we show the one-to-one correspondence between the feasible solutions of (3.32) and the set $C(S)$ of stationary policies. For $\pi^\infty \in C(S)$ the vector x^π with components $x_i^\pi(a),\ (i,a) \in S \times A$, is defined by

$$x_i^\pi(a) := \big\{\beta^T \{I - \alpha P(\pi)\}^{-1}\big\}_i \cdot \pi_{ia},\ (i,a) \in S \times A. \tag{3.33}$$

Define, for any $t \in \mathbb{N}$ and $(i,a) \in S \times A$, a random variable $n_{ia}^{(t)}$ by

$$n_{ia}^{(t)} := \begin{cases} 1 \text{ if } (X_t, Y_t) = (i,a) \\ 0 \text{ otherwise} \end{cases}$$

Then, the total discounted number of times that $(X_t, Y_t) = (i,a)$ equals $\sum_{t=1}^\infty \alpha^{t-1}\, n_{ia}^{(t)}$. The next lemma shows that $x_i^\pi(a)$ can be interpreted as the expected total discounted number of times that $(X_t, Y_t) = (i,a)$, given initial distribution β, i.e. $\mathbb{P}\{X_1 = j\} = \beta_j$ for every $j \in S$, and policy π^∞.

Lemma 3.4

Given initial distribution β and a stationary policy π^∞, $x_i^\pi(a)$ satisfies $x_i^\pi(a) = \mathbb{E}_{\beta,\pi}\big\{\sum_{t=1}^\infty \alpha^{t-1} n_{ia}^{(t)}\big\}$ for all $(i,a) \in S \times A$.

Proof

Since $\{I - \alpha P(\pi)\}^{-1} = \sum_{t=1}^\infty \alpha^{t-1} P^{t-1}(\pi)$, we have

$$\begin{aligned} x_i^\pi(a) &= \textstyle\sum_j \beta_j \cdot \{\sum_{t=1}^\infty \alpha^{t-1} P^{t-1}(\pi)\}_{ji} \cdot \pi_{ia} = \sum_{t=1}^\infty \alpha^{t-1} \big\{\sum_j \beta_j \cdot \mathbb{P}_\pi\{X_t = i \mid X_1 = j\}\big\} \cdot \pi_{ia} \\ &= \textstyle\sum_{t=1}^\infty \alpha^{t-1} \big\{\sum_j \beta_j \cdot \mathbb{P}_\pi\{X_t = i, Y_t = a \mid X_1 = j\}\big\} = \sum_{t=1}^\infty \alpha^{t-1} \cdot \mathbb{E}_{\beta,\pi}\big\{n_{ia}^{(t)}\big\} \\ &= \mathbb{E}_{\beta,\pi}\big\{\textstyle\sum_{t=1}^\infty \alpha^{t-1} n_{ia}^{(t)}\big\}. \end{aligned}$$

□

Conversely, for a feasible solution x of (3.32), define π^x with elements π^x_{ia} by

$$\pi^x_{ia} := \frac{x_i(a)}{\sum_a x_i(a)}, \ (i,a) \in S \times A. \tag{3.34}$$

Theorem 3.18

The mapping (3.33) is a one-to-one mapping of the set of stationary policies onto the set of feasible solutions of the dual program (3.32) with (3.34) as the inverse mapping; furthermore, the set of extreme feasible solutions of (3.32) corresponds to the set $C(D)$ of deterministic policies.

Proof

First, we show that x^π is a feasible solution of (3.32).

$$\begin{aligned}\sum_{(i,a)} \{\delta_{ij} - \alpha p_{ij}(a)\}x^\pi_i(a) &= \sum_{(i,a)} \{\delta_{ij} - \alpha p_{ij}(a)\}\{\beta^T\{I-\alpha P(\pi)\}^{-1}\}_i \cdot \pi_{ia} \\ &= \sum_i \{\beta^T\{I-\alpha P(\pi)\}^{-1}\}_i \cdot \sum_a \{\delta_{ij} - \alpha p_{ij}(a)\} \cdot \pi_{ia} \\ &= \sum_i \{\beta^T\{I-\alpha P(\pi)\}^{-1}\}_i \cdot \{I-\alpha P(\pi)\}_{ij} \\ &= \{\beta^T\{I-\alpha P(\pi)\}^{-1} \cdot \{I-\alpha P(\pi)\}\}_j = \beta_j, \ j \in S.\end{aligned}$$

Since $\{I-\alpha P(\pi)\}^{-1} = \sum_{t=0}^\infty \{\alpha P(\pi)\}^t \geq 0, \ x^\pi_i(a) \geq 0$ for all $(i,a) \in S \times A$.

Next, we prove the one-to-one correspondence. Let x be a feasible solution of (3.32). Then, (3.34) yields $x_i(a) = \pi^x_{ia} \cdot x_i$, where $x_i := \sum_a x_i(a), \ i \in S$. Therefore, we can write

$$\begin{aligned}\beta_j &= \sum_{(i,a)} \{\delta_{ij} - \alpha p_{ij}(a)\}x_i(a) = \sum_{(i,a)} \{\delta_{ij} - \alpha p_{ij}(a)\} \cdot \pi^x_{ia} \cdot x_i \\ &= \sum_i \{\delta_{ij} - \alpha p_{ij}(\pi^x)\} \cdot x_i, \ j \in S.\end{aligned}$$

Hence, in vector notation, $\beta^T = x^T\{I - \alpha P(\pi^x)\}$, i.e. $x^T = \beta^T\{I - \alpha P(\pi^x)\}^{-1}$, and consequently $x_i = x^{\pi^x}_i, \ i \in S$. This implies $x_i(a) = x_i \cdot \pi^x_{ia} = x^{\pi^x}_i \cdot \pi^x_{ia} = x^{\pi^x}_i(a)$ for all $(i,a) \in S \times A$.
Conversely,

$$\pi^{x^\pi}_{ia} = \frac{x^\pi_i(a)}{\sum_a x^\pi_i(a)} = \pi_{ia} \text{ for all } (i,a) \in S \times A. \tag{3.35}$$

Therefore, we have shown the one-to-one correspondence and the fact that (3.34) is the inverse of (3.33). Finally, we show the correspondence between the extreme points of (3.32) and the set $C(D)$. Let $f^\infty \in C(D)$. Then, for every $i \in S$,

$$x^f_i(a) = \begin{cases} \{\beta^T\{I-\alpha P(f)\}^{-1}\}_i & , \ a = f(i) \\ 0 & , \ a \neq f(i) \end{cases}$$

Suppose x^f is not an extreme feasible solution. Then, there exist feasible solutions x^1 and x^2 of (3.32) and a real number $\lambda \in (0,1)$ such that $x^1 \neq x^2$ and $x^f = \lambda x^1 + (1-\lambda)x^2$.

Since $x^f_i(a) = 0, a \neq f(i), i \in S$, we have $x^1_i(a) = x^2_i(a) = 0, \ a \neq f(i), i \in S$.

Hence, the N-vectors $x^1 := (x^1_i(f(i))$ and $x^2 := (x^2_i(f(i))$ are solutions of the linear system $x^T\{I - \alpha P(f)\} = \beta^T$. However, this linear system has a unique solution $x^T = \beta^T\{I-\alpha P(f)\}^{-1}$. This implies $x^1 = x^2 = \beta^T\{I - \alpha P(f)\}^{-1}$, which contradicts $x^1 \neq x^2$. Hence, we have shown that

x^f is an extreme solution.
Conversely, let x be an extreme feasible solution of program (3.32). Since (3.32) has N constraints, x has at most N positive components. On the other hand, Theorem 3.17, part (1), implies that in each state there is at least one positive component. Consequently, x has exactly one positive component in each state i, i.e. the corresponding stationary policy is deterministic. □

Algorithm 3.3 *Linear programming algorithm*
Input: Instance of a discounted MDP.
Output: Optimal deterministic policy f^∞ and the value vector v^α.

1. Take any vector β with $\beta_j > 0$ for every $j \in S$.

2. Use the simplex method to compute optimal solutions v^* and x^* of the dual pair of linear programs:

$$min \left\{ \sum_j \beta_j v_j \;\middle|\; \sum_j \{\delta_{ij} - \alpha p_{ij}(a)\} v_j \ge r_i(a),\ (i,a) \in S \times A \right\}$$

and

$$max \left\{ \sum_{(i,a)} r_i(a) x_i(a) \;\middle|\; \begin{array}{rl} \sum_{(i,a)} \{\delta_{ij} - \alpha p_{ij}(a)\} x_i(a) & = \beta_j,\ j \in S \\ x_i(a) & \ge 0,\ (i,a) \in S \times A \end{array} \right\}.$$

3. Take $f_*^\infty \in C(D)$ such that $x_i^*(f_*(i)) > 0$ for every $i \in S$.
v^* is the value vector v^α and f_*^∞ is an α-discounted optimal policy (STOP).

Example 3.2
Consider the model of Example 3.1 and let $\beta_1 = \beta_2 = \beta_3 = \frac{1}{3}$.
The dual linear program (3.32) becomes:

$max\ x_1(1) + 2x_1(2) + 3x_1(3) + 6x_2(1) + 4x_2(2) + 5x_2(3) + 8x_3(1) + 9x_3(2) + 7x_3(3)$
subject to

$$\begin{array}{lllllllllll} \frac{1}{2}x_1(1) + & x_1(2) + & x_1(3) & -\frac{1}{2}x_2(1) & & & -\frac{1}{2}x_3(1) & & & = & \frac{1}{3} \\ & -\frac{1}{2}x_1(2) & & +\ x_2(1) + & \frac{1}{2}x_2(2) + & x_2(3) & & -\frac{1}{2}x_3(2) & & = & \frac{1}{3} \\ & & -\frac{1}{2}x_1(3) & & & -\frac{1}{2}x_2(3) & +\ x_3(1) + & x_3(2) + & \frac{1}{2}x_3(3) & = & \frac{1}{3} \end{array}$$

$x_1(1), x_1(2), x_1(3), x_2(1), x_2(2), x_2(3), x_3(1), x_3(2), x_3(3) \ge 0$

We start with phase I of the simplex method to obtain a first feasible basic solution corresponding to policy f^∞ where $f(1) = 3$, $f(2) = 2$ and $f(3) = 1$. Therefore, we take the columns of $x_1(3), x_2(2)$ and $x_3(1)$ as pivot columns in the first three iterations. The pivot element is the bold number in the tableau. Next, in phase II, the usual choice of the pivot column is taken, i.e. the column with the most negative element in the transformed objective function (last row in the tableau, also called the row of the *reduced costs*). We write the linear programming tableaus in the so-called contracted form (cf. [341]).

Iteration 1

		$x_1(1)$	$x_1(2)$	$x_1(3)$	$x_2(1)$	$x_2(2)$	$x_2(3)$	$x_3(1)$	$x_3(2)$	$x_3(3)$
z_1	$\frac{1}{3}$	$\frac{1}{2}$	1	$\mathbf{1}$	$-\frac{1}{2}$	0	0	$-\frac{1}{2}$	0	0
z_2	$\frac{1}{3}$	0	$-\frac{1}{2}$	0	1	$\frac{1}{2}$	1	0	$-\frac{1}{2}$	0
z_3	$\frac{1}{3}$	0	0	$-\frac{1}{2}$	0	0	$-\frac{1}{2}$	1	1	$-\frac{1}{2}$
I	-1	$-\frac{1}{2}$	$-\frac{1}{2}$	$-\frac{1}{2}$	$-\frac{1}{2}$	$-\frac{1}{2}$	$-\frac{1}{2}$	$-\frac{1}{2}$	$-\frac{1}{2}$	$-\frac{1}{2}$
II	0	-1	-2	-3	-6	-4	-5	-8	-9	-7

Iteration 2

		$x_1(1)$	$x_1(2)$	z_1	$x_2(1)$	$x_2(2)$	$x_2(3)$	$x_3(1)$	$x_3(2)$	$x_3(3)$
$x_1(3)$	$\frac{1}{3}$	$\frac{1}{2}$	1	1	$-\frac{1}{2}$	0	0	$-\frac{1}{2}$	0	0
z_2	$\frac{1}{3}$	0	$-\frac{1}{2}$	0	1	$\mathbf{\frac{1}{2}}$	1	0	$-\frac{1}{2}$	0
z_3	$\frac{1}{2}$	$\frac{1}{4}$	$\frac{1}{2}$	$\frac{1}{2}$	$-\frac{1}{4}$	0	$-\frac{1}{2}$	$\frac{3}{4}$	1	$\frac{1}{2}$
I	$-\frac{5}{6}$	$-\frac{1}{4}$	0	$\frac{1}{2}$	$-\frac{3}{4}$	$-\frac{1}{2}$	$-\frac{1}{2}$	$-\frac{3}{4}$	$-\frac{1}{2}$	$-\frac{1}{2}$
II	1	$\frac{1}{2}$	1	3	$-\frac{15}{2}$	-4	-5	$-\frac{19}{2}$	-9	-7

Iteration 3

		$x_1(1)$	$x_1(2)$	z_1	$x_2(1)$	z_2	$x_2(3)$	$x_3(1)$	$x_3(2)$	$x_3(3)$
$x_1(3)$	$\frac{1}{3}$	$\frac{1}{2}$	1	1	$-\frac{1}{2}$	0	0	$-\frac{1}{2}$	0	0
$x_2(2)$	$\frac{2}{3}$	0	-1	0	2	2	2	0	-1	0
z_3	$\frac{1}{2}$	$\frac{1}{4}$	$\frac{1}{2}$	$\frac{1}{2}$	$-\frac{1}{4}$	0	$-\frac{1}{2}$	$\mathbf{\frac{3}{4}}$	1	$\frac{1}{2}$
I	$-\frac{1}{2}$	$-\frac{1}{4}$	$-\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{4}$	1	$\frac{1}{2}$	$-\frac{3}{4}$	-1	$-\frac{1}{2}$
II	$\frac{11}{3}$	$\frac{1}{2}$	-3	3	$\frac{1}{2}$	8	3	$-\frac{19}{2}$	-13	-7

Iteration 4

		$x_1(1)$	$x_1(2)$	z_1	$x_2(1)$	z_2	$x_2(3)$	z_3	$x_3(2)$	$x_3(3)$
$x_1(3)$	$\frac{2}{3}$	$\frac{2}{3}$	$\frac{4}{3}$	$\frac{4}{3}$	$-\frac{2}{3}$	0	$-\frac{1}{3}$	$\frac{2}{3}$	$\frac{2}{3}$	$\frac{1}{3}$
$x_2(2)$	$\frac{2}{3}$	0	-1	0	2	2	$\mathbf{2}$	0	-1	0
$x_3(1)$	$\frac{2}{3}$	$\frac{1}{3}$	$\frac{2}{3}$	$\frac{2}{3}$	$-\frac{1}{3}$	0	$-\frac{2}{3}$	$\frac{4}{3}$	$\frac{4}{3}$	$\frac{2}{3}$
I	0	0	0	1	0	1	0	1	0	0
II	10	$\frac{11}{3}$	$\frac{10}{3}$	$\frac{28}{3}$	$-\frac{8}{3}$	8	$-\frac{10}{3}$	$\frac{38}{3}$	$-\frac{1}{3}$	$-\frac{2}{3}$

Iteration 5

		$x_1(1)$	$x_1(2)$	z_1	$x_2(1)$	z_2	$x_2(2)$	z_3	$x_3(2)$	$x_3(3)$
$x_1(3)$	$\frac{7}{9}$	$\frac{2}{3}$	$\frac{7}{6}$	$\frac{4}{3}$	$-\frac{1}{3}$	$\frac{1}{3}$	$\frac{1}{6}$	$\frac{2}{3}$	$\frac{1}{2}$	$\frac{1}{3}$
$x_2(3)$	$\frac{1}{3}$	0	$-\frac{1}{2}$	0	1	1	$\frac{1}{2}$	0	$-\frac{1}{2}$	0
$x_3(1)$	$\frac{8}{9}$	$\frac{1}{3}$	$\frac{1}{3}$	$\frac{2}{3}$	$\frac{1}{3}$	$\frac{2}{3}$	$\frac{1}{3}$	$\frac{4}{3}$	$\mathbf{1}$	$\frac{2}{3}$
II	$\frac{100}{9}$	$\frac{11}{3}$	$\frac{5}{3}$	$\frac{28}{3}$	$\frac{2}{3}$	$\frac{34}{3}$	$\frac{5}{3}$	$\frac{38}{3}$	-2	$-\frac{2}{3}$

Iteration 6

		$x_1(1)$	$x_1(2)$	z_1	$x_2(1)$	z_2	$x_2(2)$	z_3	$x_3(1)$	$x_3(3)$
$x_1(3)$	$\frac{1}{3}$	$\frac{1}{2}$	1	1	$-\frac{1}{2}$	0	0	0	$-\frac{1}{2}$	0
$x_2(3)$	$\frac{7}{9}$	$\frac{1}{6}$	$-\frac{1}{3}$	$\frac{1}{3}$	$\frac{7}{6}$	$\frac{4}{3}$	$\frac{2}{3}$	$\frac{2}{3}$	$\frac{1}{2}$	$\frac{1}{3}$
$x_3(2)$	$\frac{8}{9}$	$\frac{1}{3}$	$\frac{1}{3}$	$\frac{2}{3}$	$\frac{1}{3}$	$\frac{2}{3}$	$\frac{1}{3}$	$\frac{4}{3}$	1	$\frac{2}{3}$
II	$\frac{116}{9}$	$\frac{13}{3}$	$\frac{7}{3}$	$\frac{32}{3}$	$\frac{4}{3}$	$\frac{38}{3}$	$\frac{7}{3}$	$\frac{46}{3}$	2	$\frac{2}{3}$

The last tableau is an optimal simplex tableau corresponding to the following optimal solution:
$x_1^*(1) = 0, x_1^*(2) = 0, x_1^*(3) = \frac{1}{3}$; $x_2^*(1) = 0, x_2^*(2) = 0, x_2^*(3) = \frac{7}{9}$; $x_3^*(1) = 0, x_3^*(2) = \frac{8}{9}, x_3^*(3) = 0$.
The optimal solution of the primal problem is: $v_1^* = \frac{32}{3}, v_2^* = \frac{38}{3}$ and $v_3^* = \frac{46}{3}$.
Hence, the value vector $v^\alpha = (\frac{32}{3}, \frac{38}{3}, \frac{46}{3})$ and the α-discounted optimal policy is f_*^∞ with
$f_*(1) = 3, f_*(2) = 3$ and $f_*(3) = 2$.

We now show the equivalence between the policy iteration method and the linear programming method. Consider a deterministic policy f^∞. We have seen that x^f is an extreme point of (3.32) and that $x_i^f(f(i)) > 0$ for every $i \in S$. By introducing slack variables $y_i(a),\ (i,a) \in S \times A$ in the primal problem (3.31), this program becomes

$$min \left\{ \sum_j \beta_j v_j \;\middle|\; \begin{array}{rcll} \sum_j \{\delta_{ij} - \alpha p_{ij}(a)\} v_j - y_i(a) & = & r_i(a), & (i,a) \in S \times A \\ y_i(a) & \geq & 0, & (i,a) \in S \times A \end{array} \right\}. \tag{3.36}$$

Let $\left(v^f, y^f\right)$ be the dual solution corresponding to x^f. Then, by the complementary slackness property of linear programming, we have

$$x_i^f(a) \cdot y_i^f(a) = 0 \text{ for every } (i,a) \in S \times A.$$

Since $x_i^f\big(f(i)\big) > 0$ for every $i \in S,\ y_i^f\big(f(i)\big) = 0$ for every $i \in S$. Hence, from the constraints of (3.36), we obtain in vector notation $\{I - \alpha P(f)\} v^f = r(f)$, implying

$$v^f = \{I - \alpha P(f)\}^{-1} r(f) = v^\alpha(f^\infty),$$

and

$$y_i^f(a) = \sum_j \{\delta_{ij} - \alpha p_{ij}(a)\} v_j^\alpha(f^\infty) - r_i(a) = -s_{ia}(f),\ (i,a) \in S \times A, \tag{3.37}$$

where $s_{ia}(f)$ is defined in the policy iteration algorithms 3.1 and 3.2.

In any simplex tableau, possible choices for the pivot column are those columns of nonbasic $x_i(a)$-variables which have negative reduced costs (also called *shadow prices*): $y_i^f(a) < 0$, i.e. $s_{ia}(f) > 0$. Hence, the possible pivot columns in state i are exactly the columns corresponding to the actions of $A(i,f)$, where $A(i,f)$ is defined in (3.18).

Consider in the policy iteration method two subsequent policies, say f^∞ and g^∞, and let $E(f,g) = \{i \in S \mid f(i) \neq g(i)\}$. If we exchange in an iteration of the simplex method the nonbasic variables $x_i^f(a)$ and $x_i^g(a)$ for every $i \in E(f,g)$, then we obtain a linear programming algorithm in which (in general) more than one pivot step is executed in one iteration, and in which subsequent basic solutions correspond to subsequent policies of the policy iteration method. An algorithm in which in one iteration more than one pivot step can be executed is called a *block-pivoting simplex algorithm* (cf. [48] p. 201).

On the other hand, suppose that the usual simplex algorithm is applied with only one pivot step in one iteration and that we choose as entering variable a nonbasic variable $x_i^f(a)$ corresponding to a variable $s_{ia}(f) > 0$, i.e. $a \in A(i,f)$. Since such a choice is allowed in the policy iteration method, the usual simplex method is a special implementation of the policy iteration method. We summarize the above statements in the following theorem.

Theorem 3.19

(1) Any policy iteration algorithm is equivalent to a particular block-pivoting simplex algorithm.

(2) Any simplex algorithm is equivalent to a particular policy iteration algorithm.

Example 3.2 (continued)

Start with the simplex tableau corresponding to the first feasible solution. Consider an iteration and let the basic solution corresponds to policy f^∞. Then, choose in each state i for which $min_a\, y_i^f(a) < 0$ as pivot column the column corresponding to $x_i^f\big(g(i)\big)$, where $g(i)$ is such that $y_i^f\big(g(i)\big) = min_a\, y_i^f(a)$. In subsequent tableaus we execute the block-pivoting algorithm where in each iteration the pivot steps correspond to the nonbasic variables $x_i^f\big(g(i)\big)$. The pivots are again indicated by bold numbers.

Iteration 1

		$x_1(1)$	$x_1(2)$	z_1	$x_2(1)$	z_2	$x_2(3)$	z_3	$x_3(2)$	$x_3(3)$
$x_1(3)$	$\frac{2}{3}$	$\frac{2}{3}$	$\frac{4}{3}$	$\frac{4}{3}$	$-\frac{2}{3}$	0	$-\frac{1}{3}$	$\frac{2}{3}$	$\frac{2}{3}$	$\frac{1}{3}$
$x_2(2)$	$\frac{2}{3}$	0	-1	0	2	2	$\mathbf{2}$	0	-1	0
$x_3(1)$	$\frac{2}{3}$	$\frac{1}{3}$	$\frac{2}{3}$	$\frac{2}{3}$	$-\frac{1}{3}$	0	$-\frac{2}{3}$	$\frac{4}{3}$	$\frac{4}{3}$	$\mathbf{\frac{2}{3}}$
	10	$\frac{11}{3}$	$\frac{10}{3}$	$\frac{28}{3}$	$-\frac{8}{3}$	8	$-\frac{10}{3}$	$\frac{38}{3}$	$-\frac{1}{3}$	$-\frac{2}{3}$

Iteration 2

		$x_1(1)$	$x_1(2)$	z_1	$x_2(1)$	z_2	$x_2(2)$	z_3	$x_3(2)$	$x_3(1)$
$x_1(3)$	$\frac{1}{3}$	$\frac{1}{2}$	1	1	$-\frac{1}{2}$	0	0	0	0	$-\frac{1}{2}$
$x_2(3)$	$\frac{1}{3}$	0	$-\frac{1}{2}$	0	1	1	$\frac{1}{2}$	0	$-\frac{1}{2}$	0
$x_3(3)$	$\frac{4}{3}$	$\frac{1}{2}$	$\frac{1}{2}$	1	$\frac{1}{2}$	1	$\frac{1}{2}$	2	$\mathbf{\frac{3}{2}}$	$\frac{3}{2}$
	$\frac{108}{9}$	4	2	10	1	12	2	14	-1	1

Iteration 3

		$x_1(1)$	$x_1(2)$	z_1	$x_2(1)$	z_2	$x_2(2)$	z_3	$x_3(3)$	$x_3(1)$
$x_1(3)$	$\frac{1}{3}$	$\frac{1}{2}$	1	1	$-\frac{1}{2}$	0	0	0	0	$-\frac{1}{2}$
$x_2(3)$	$\frac{7}{9}$	$\frac{1}{6}$	$-\frac{1}{3}$	$\frac{1}{3}$	$\frac{7}{6}$	$\frac{4}{3}$	$\frac{2}{3}$	$\frac{2}{3}$	$\frac{1}{3}$	$\frac{1}{2}$
$x_3(2)$	$\frac{8}{9}$	$\frac{1}{3}$	$\frac{1}{3}$	$\frac{2}{3}$	$\frac{1}{3}$	$\frac{2}{3}$	$\frac{1}{3}$	$\frac{4}{3}$	$\frac{2}{3}$	1
	$\frac{116}{9}$	$\frac{13}{3}$	$\frac{7}{3}$	$\frac{32}{3}$	$\frac{4}{3}$	$\frac{38}{3}$	$\frac{7}{3}$	$\frac{46}{3}$	$\frac{2}{3}$	2

This is the optimal tableau which gives the value vector $v^\alpha = (\frac{32}{3}, \frac{38}{3}, \frac{46}{3})$ and the α-discounted optimal policy f_*^∞ with $f_*(1) = 3, f_*(2) = 3$ and $f_*(3) = 2$.

Next, we discuss the *elimination of suboptimal actions*. Since the linear programming method is equivalent to the policy iteration method, we can copy the results of section 3.4, in particular Theorem 3.12. Instead of the numbers $s_{ia}(f)$, we use in linear programming the dual slack variables $y_i^f(a)$, where $y_i^f(a) = -s_{ia}(f),\ (i,a) \in S \times A$. Hence, we obtain following result.

Theorem 3.20
If $y_i^f(a_i) > min_a\ y_i^f(a) - \alpha(1-\alpha)^{-1}\{min_i min_a\ y_i^f(a) - max_i min_a\ y_i^f(a)\}$, *then action* $a_i \in A(i)$ *is suboptimal.*

Example 3.2 (continued)

We consider the usual simplex method without block-pivoting and start with the first feasible tableau (iteration 4).

Iteration 4

$\alpha(1-\alpha)^{-1}\{min_i min_a\ y_i^f(a) - max_i min_a\ y_i^f(a)\} = -\frac{10}{3}$.

In state 1, action 1 is excluded, because $\frac{11}{3} = y_1^f(1) > min_a\ y_1^f(a) + \frac{10}{3} = \frac{10}{3}$.

Iteration 5

$\alpha(1-\alpha)^{-1}\{min_i min_a\ y_i^f(a) - max_i min_a\ y_i^f(a)\} = -2$.

In this iteration, no suboptimal actions are found.

A second method for eliminating suboptimal actions is based on a general LP property, due to Cheng [39] and presented in Theorem 3.21. Consider the linear programming problem formulated as

$$max\{p^T x \mid Ax = b;\ x \geq 0\}, \tag{3.38}$$

where A is am $m \times n$ matrix. Assume that $rank\,(A) = m$ and that this LP has a finite optimal value z_0^*. The dual of (3.38) is

$$min\{b^T u \mid A^T u - v = p;\ v \geq 0\}. \tag{3.39}$$

Let B be a feasible basis matrix of (3.38) with corresponding basic solution $x_B = B^{-1}b \geq 0$, $x_N = 0$. The corresponding dual basic solution is $u^T = p^T B^{-1},\ v^T = p^T B^{-1} A - p^T = u^T A - p^T$, which indeed satisfies $A^T u - v = p$. Note that $v \geq 0$ is not required for the corresponding dual basic solution; $v \geq 0$ if and only if $x = (x_B, x_N)$ and (u, v) are optimal feasible solutions of (3.38) and (3.39), respectively. The corresponding value of the primal problem (and also of the dual problem) is $z_0 := p_B^T B^{-1} b$.

A basis matrix B and the corresponding basic solution x are called *nondegenerated* if we have $(B^{-1}b)_i > 0$ for all $1 \leq i \leq m$. Denote the jth column of A by A_j.

Theorem 3.21

Let B be a nondegenerate basis of the linear program (3.38) with corresponding basic solutions x and (u, v), respectively. Let x_j be a nonbasic variable with reduced cost $v_j = p_B^T B^{-1} A_j - p_j > 0$. Then, we have

(1) If $B^{-1}A_j \geq 0$, then $x_j^* = 0$ in any optimal basic solution x^*.

(2) If $B^{-1}A_j \not\geq 0$ and $v_j + \theta \cdot \{\overline{z} - z_0) > 0$, where $\theta := min_i \frac{\{B^{-1}A_j\}_i}{\{B^{-1}b\}_i}$ and $\overline{z}$ is an upper bound of the optimum, then $x_j^* = 0$ in any optimal basic solution x^*.

Proof

Let B^* be an optimal basis matrix with x^* and (u^*, v^*) as corresponding basic optimal solutions. Then, $v^* \geq 0$, implying $(u^*)^T A \geq p^T$ and consequently $(u^*)^T B \geq p_B^T$.

(1) $v_j^* = (u^*)^T A_j - p_j = \{(u^*)^T B]\}\{B^{-1}A_j\} - p_j \geq p_B^T B^{-1} A_j - p_j = u^T A_j - p_j = v_j > 0$.

where the inequality is verified by the property $B^{-1}A_j \geq 0$. From the complementary slackness property of linear programming we have $x_j^* \cdot v_j^* = 0$ for all j, implying $x_j^* = 0$.

(2) Note that θ is a well-defined and finite number by the property that B is a nondegenerate basis matrix. Since $B^{-1}A_j \not\geq 0$, we have $\theta < 0$. Furthermore, by the definition of θ, $B^{-1}A_j - \theta \cdot B^{-1}b \geq 0$. Hence, we can write

$$\begin{array}{rcl} v_j^* & = & (u^*)^T A_j - p_j = \{(u^*)^T B\}\{B^{-1}A_j - \theta \cdot B^{-1}b + \theta \cdot B^{-1}b\} - p_j \\ & = & \{(u^*)^T B\}\{B^{-1}A_j - \theta \cdot B^{-1}b\} + \theta \cdot (u^*)^T b - p_j \\ & \geq & p_B^T\{B^{-1}A_j - \theta \cdot B^{-1}b\} + \theta \cdot (u^*)^T b - p_j \\ & = & v_j + \theta \cdot \{(u^*)^T b - p_B^T B^{-1} b = v_j + \theta \cdot (z_0^* - z_0) \geq v_j + \theta \cdot (\overline{z} - z_0) > 0. \end{array}$$

As in part (1), from the complementary slackness property of linear programming, we have $x_j^* \cdot v_j^* = 0$, implying $x_j^* = 0$. □

Notice that, by part (1) of Theorem 3.17, the linear program (3.32) for discounted MDPs is nondegenerated. In order to apply Theorem 3.21, we need an (easily) computable upper bound for the optimum of program (3.32). Such a bound is provided by the next lemma.

Lemma 3.5

$v^\alpha(f^\infty) - (1-\alpha)^{-1} \cdot min_{(i,a)}\ y_i^f(a) \cdot e$ is an upper bound of the value vector v^α.

Proof

Take any deterministic policy g^∞. Since $y_i^f(a) = \sum_j \{\delta_{ij} - \alpha p_{ij}(a)\} v_j^\alpha(f^\infty) - r_i(a)$, $(i, a) \in S \times A$, we obtain

$$min_{(i,a)}\ y_i^f(a) \cdot e \leq \{I - \alpha P(g)\} v^\alpha(f^\infty) - r(g).$$

Hence,

$$\begin{array}{rcl} v^\alpha(f^\infty) & \geq & \{I - \alpha P(g)\}^{-1} r(g) + min_{(i,a)}\, y_i^f(a) \cdot \{I - \alpha P(g)\}^{-1} \cdot e \\ & = & v^\alpha(g^\infty) + min_{(i,a)}\, y_i^f(a) \cdot \sum_{t=0}^\infty\ \alpha^t P(g)^t \cdot e \\ & = & v^\alpha(g^\infty) + min_{(i,a)}\, y_i^f(a) \cdot \{\ \sum_{t=0}^\infty\ \alpha^t\} \cdot e \\ & = & v^\alpha(g^\infty) + min_{(i,a)}\, y_i^f(a) \cdot (1-\alpha)^{-1} \cdot e \end{array}$$

Let g^∞ be an optimal policy. Then, $v^\alpha = v^\alpha(g^\infty) \leq v^\alpha(f^\infty) - (1-\alpha)^{-1} \cdot min_{(i,a)}\, y_i^f(a) \cdot e$. □

Corollary 3.8

$\beta^T v^\alpha(f^\infty) - \{\beta^T e\} \cdot (1-\alpha)^{-1} \cdot min_{(i,a)}\ y_i^f(a)$ is an upper bound of the optimum of program (3.32).

Proof

The optimal value of (3.32) is equal to the optimum of program (3.31) which is $\beta^T v^\alpha$. By Lemma 3.5, $\beta^T v^\alpha \le \sum_k \beta_k \{v_k^\alpha(f^\infty) - (1-\alpha)^{-1} \cdot min_{(i,a)}\ y_i^f(a)\} = \beta^T v^\alpha(f^\infty) - \{\beta^T e\} \cdot (1-\alpha)^{-1} \cdot min_{(i,a)}\ y_i^f(a)$. □

The next theorem is a direct consequence of Corollary 3.8 and Theorem 3.21.

Theorem 3.22

Let $A_{j,a}(f)$ and $B(f)$ be the columns of the nonbasic variable $x_i^f(a)$ and the basis matrix, respectively, in the simplex tableau corresponding to policy f^∞, and let $y_i^f(a) > 0$.
Then, action $a \in A(j)$ is suboptimal if either one of the following conditions is satisfied:
(1) $B(f)^{-1}A_{j,a}(f) \ge 0$.
(2) $B(f)^{-1}A_{j,a}(f) \not\ge 0$ and $y_j^f(a) - min_i \frac{\{B(f)^{-1}A_{j,a(f)}\}_i}{\{B(f)^{-1}\beta\}_i} \cdot \{\beta^T e\} \cdot (1-\alpha)^{-1} \cdot min_{(i,a)}\ y_i^f(a) > 0$.

Example 3.2 (continued)

Again, we start with the first feasible tableau (iteration 4).
Iteration 4
$(1-\alpha)^{-1} \cdot min_{(i,a)}\ y_i^f(a) = -\frac{20}{3}$.
In state 1, action 1 is excluded, because condition (1) of Theorem 3.22 is satisfied.
Iteration 5
$(1-\alpha)^{-1} \cdot min_{(i,a)}\ y_i^f(a) = -4$.
In state 2, action 2 is excluded, because condition (1) of Theorem 3.22 is satisfied.

Let $B(f)$ be the basis matrix corresponding to policy f^∞. Then, we have $B(f) = \{I - \alpha P(f)\}^T$, implying $x(f)^T = \beta^T\{I - \alpha P(f)\}^{-1}$, where $x(f)$ is the N-dimensional vector with elements $x_i(f) := x_i^f\big(f(i)\big)$ for all $i \in S$. The next two lemmata present interesting formulas.

Lemma 3.6

$\beta^T\{v^\alpha(g^\infty) - v^\alpha(f^\infty)\} = x(g)^T\{r(g) - r(f) + \alpha[P(g) - P(f)]v^\alpha(f^\infty)\}$ for every pair $f^\infty, g^\infty \in C(D)$.

Proof

$x(g)^T\{r(g) - r(f) + \alpha[P(g) - P(f)]v^\alpha(f^\infty)\} = x(g)^T r(g) - x(g)^T\{r(f) - \alpha P(g)v^\alpha(f^\infty) + \alpha P(f)v^\alpha(f^\infty)\}$.
Since $r(f) + \alpha P(f)v^\alpha(f^\infty) = r(f) + \alpha P(f)\{\sum_{t=0}^\infty \big(\alpha P(f)\big)^t r(f)\} = \sum_{t=0}^\infty \big(\alpha P(f)\big)^t r(f) = v^\alpha(f^\infty)$ and $x(g)^T r(g) = \beta^T\{I - \alpha P(g)\}^{-1} r(g) = \beta^T v^\alpha(g^\infty)$, we obtain

$$\begin{aligned} x(g)^T\{r(g) - r(f) + \alpha[P(g) - P(f)]v^\alpha(f^\infty)\} &= \beta^T v^\alpha(g^\infty) - x(g)^T\{v^\alpha(f^\infty) - \alpha P(g)v^\alpha(f^\infty)\} \\ &= \beta^T v^\alpha(g^\infty) - x(g)^T\{[I - \alpha P(g)]v^\alpha(f^\infty)\} \\ &= \beta^T v^\alpha(g^\infty) - \beta^T\{I - \alpha P(g)\}^{-1}\{[I - \alpha P(g)]v^\alpha(f^\infty)\} \\ &= \beta^T\{v^\alpha(g^\infty) - v^\alpha(f^\infty)\}. \end{aligned}$$ □

Lemma 3.7

$x(f)^T = x(g)^T\{I - \alpha[P(g) - P(f)][I - \alpha P(f)]^{-1}\}$ for every pair $f^\infty, g^\infty \in C(D)$.

Proof

Since $x(f)$ is the unique solution of the linear system $x^TB(f)^T = \beta^T$, we have to show

$$x(g)^T\{I - \alpha[P(g) - P(f)][I - \alpha P(f)]^{-1}\}B(f)^T = \beta^T.$$

We can write

$$\begin{aligned}&x(g)^T\{I - \alpha[P(g) - P(f)][I - \alpha P(f)]^{-1}\}B(f)^T = \\ &x(g)^T\{I - \alpha[P(g) - P(f)][I - \alpha P(f)]^{-1}\}\{I - \alpha P(f)\} = \\ &x(g)^T\{I - \alpha P(f) - \alpha P(g) + \alpha P(f)\} = x(g)^T\{I - \alpha P(g)\} = \beta^T.\end{aligned}$$

□

Consider two subsequent simplex tableaus corresponding to the policies f^∞ and g^∞, respectively. Then, g^∞ is a policy which is the same as f^∞, except in one state, say state j, action $g(j) \neq f(j)$. Then, $[P(g) - P(f)][I - \alpha P(f)]^{-1}$ has zero rows, except for row j, which has as kth element

$$\sum_l \{p_{jl}(g) - p_{jl}(f)\}\{Y(f)\}_{lk},$$

where $Y(f) = [I - \alpha P(f)]^{-1} = [B(f)^T]^{-1}$. Hence, by Lemma 3.7,

$$x_k(f) = x_k(g) - x_j(g) \cdot \alpha \sum_l \{p_{jl}(g) - p_{jl}(f)\}\{Y(f)\}_{lk},\ k \in S.$$

Therefore,

If $k \neq j$, then $\frac{x_k(f)-x_k(g)}{x_j(g)} = \delta_{jk} + \alpha \cdot \sum_l \{p_{jl}(f) - p_{jl}(g)\}\{Y(f)\}_{lk}$.

If $k = j$, then $\frac{x_k(f)}{x_j(g)} = \delta_{jk} + \alpha \cdot \sum_l \{p_{jl}(f) - p_{jl}(g)\}\{Y(f)\}_{lk}$.

Lemma 3.8

$\{B(f)^{-1}A_{j,a}(f)\}_k = \delta_{jk} + \alpha \cdot \sum_l \{p_{jl}(f) - p_{jl}(g)\}\{Y(f)\}_{lk},\ k \in S.$

Proof

Let q be the N-dimensional vector with elements $p_{jl}(f) - p_{jl}(g),\ k \in S$. In vector notation, we have to show $(e^j)^T + \alpha \cdot q^TY(f) = \{B(f)^{-1}A_{j,a}(f)\}^T$, or equivalently, $(e^j)^TB(f)^T + \alpha \cdot q^TY(f)B(f)^T = \{A_{j,a}(f)\}^T$. Therefore, we have to show $\{B(f)\}_{kj} + \alpha \cdot \{p_{jk}(f) - p_{jk}(g)\} = \{A_{j,a}(f)\}_k$ for all $k \in S$. Since $\{B(f)\}_{kj} = \delta_{jk} - \alpha \cdot p_{jk}(f)$, we have $\{B(f)\}_{kj} + \alpha \cdot \{p_{jk}(f) - p_{jk}(g)\} = \delta_{jk} - \alpha \cdot p_{jk}(g) = \{A_{j,a}(f)\}_k$ for all $k \in S$.

□

The following theorem gives an interpretation of suboptimal actions in the sense that we have either $x(f) \geq x(g)$ or $x(f) \ngeq x(g)$ and $\beta^T\{v^\alpha(g^\infty) - v^\alpha(f^\infty)\} < \theta \cdot \{\overline{v} - x(f)^Tr(f)\}$, where θ is a negative scalar defined by $\theta := min_{k\neq j} \frac{x_k(f)-x_k(g)}{x_j(f)}$ and $\overline{v}$ is an upper bound of the value vector v^α.

Theorem 3.23

Let $A_{j,a}(f)$ and $B(f)$ be the columns of the nonbasic variable $x_i^f(a)$ and the basis matrix, respectively, in the simplex tableau corresponding to policy f^∞, and let $y_j^f(a) > 0$. Furthermore, let g^∞ be the policy which is the same as f^∞, except in state j, where $g(j) = a \neq f(j)$.

Then, action $g(j) \in A(j)$ is suboptimal if either one of the following conditions is satisfied:

(1) $x(f) \geq x(g)$.

(2) $x(f) \ngeq x(g)$ and $\beta^T\{v^\alpha(g^\infty) - v^\alpha(f^\infty)\} < \theta \cdot \{\overline{v} - x(f)^Tr(f)\}$.

Proof

(1) By Lemma 3.8, $\{B(f)^{-1}A_{j,a}(f)\}_k = \delta_{jk} + \alpha \cdot \sum_l \{p_{jl}(f) - p_{jl}(g)\}\{Y(f)\}_{lk} = \begin{cases} \frac{x_k(f)-x_k(g)}{x_j(g)} & \text{if } k \neq j \\ \frac{x_k(f)}{x_j(g)} > 0 & \text{if } k = j \end{cases}$

Hence, if $x(f) \geq x(g)$, we have $B(f)^{-1}A_{j,a}(f) \geq 0$ and by Theorem 3.22 part (1), action $g(j) \in A(j)$ is a suboptimal action.

(2) If $x(f) \ngeq x(g)$, $\theta := min_{k \neq j} \frac{x_k(f)-x_k(g)}{x_j(g)} < 0$. The value $y_j^f(a)$ satisfies (use for the first equality equation (3.37))

$$\begin{array}{rcl} y_j^f(a) & = & \sum_k \{\delta_{jk} - \alpha p_{jk}(g)\} v_k^\alpha(f^\infty) - r_j(g) \\ & = & \{[I - \alpha P(g)]v^\alpha(f^\infty) - r(g)\}_j \\ & = & \{[I - \alpha P(f)]v^\alpha(f^\infty) - r(g) + \alpha[P(f)] - P(g)]v^\alpha(f^\infty)\}_j \\ & = & \{[I - \alpha P(f)][I - \alpha P(f)]^{-1} r(f) - r(g) + \alpha[P(f)] - P(g)]v^\alpha(f^\infty)\}_j \\ & = & \{r(f) - r(g) + \alpha[P(f)] - P(g)]v^\alpha(f^\infty)\}_j \end{array}$$

Since $\{r(f) - r(g) + \alpha[P(f)] - P(g)]v^\alpha(f^\infty)\}_k = 0$ for $k \neq j$, we obtain

$$\begin{array}{rcl} x(g)^T\{r(f) - r(g) + \alpha[P(f)] - P(g)]v^\alpha(f^\infty)\} & = & x_j(g) \cdot \{r(f) - r(g) + \alpha[P(f)] - P(g)]v^\alpha(f^\infty)\}_j \\ & = & x_j(g) \cdot y_j^f(a). \end{array}$$

Since $\beta^T v^\alpha(f^\infty) = x(f)^T r(f)$, we have by Lemma 3.7

$$\begin{array}{rcl} \beta^T v^\alpha(f^\infty) & = & x(g)^T\{I - \alpha[P(g) - P(f)][I - \alpha P(f)]^{-1}\} r(f) \\ & = & x(g)^T r(f) - \alpha \cdot x(g)^T\{P(g) - P(f)\} v^\alpha(f^\infty) \end{array}$$

Hence, we have

$$\begin{array}{rcl} \beta^T\{v^\alpha(g^\infty) - v^\alpha(f^\infty)\} & = & x(g)^T r(g) - x(g)^T\{r(f) - \alpha \cdot [P(g) - P(f)]v^\alpha(f^\infty)\} \\ & = & x(g)^T\{r(g) - r(f) + \alpha \cdot [P(g) - P(f)]v^\alpha(f^\infty)\} \\ & = & -x_j(g) \cdot y_j^f(a). \end{array}$$

By Theorem 3.21 part (2), we have to show $y_j^f(a) + \theta^* \cdot \{\overline{v} - x(f)^T r(f)\} > 0$, where the scalar θ^* is defined by $\theta^* := min_k \frac{\{B(f)^{-1}A_{j,a}(f)\}_k}{\{B(f)^{-1}\beta\}_k}$. This is equivalent to
$-x_j(g) \cdot y_j^f(a) < x_j(g) \cdot \theta^* \cdot \{\overline{v} - x(f)^T r(f)\}$, i.e. $\beta^T\{v^\alpha(g^\infty) - v^\alpha(f^\infty)\} < x_j(g) \cdot \theta^* \cdot \{\overline{v} - x(f)^T r(f)\}$.
Because $x(g)^T = \beta^T\{I - \alpha P(g)\}^{-1} = \beta^T\{B(g)^T\}^{-1} = \{B(g)^{-1}\beta\}^T$, we have $x(g) = B(g)^{-1}\beta$.
Since $\{B(f)^{-1}A_{j,a}(f)\}_j = \frac{x_j(f)}{x_j(g)} > 0$ (see the proof of part (1)), also $\frac{\{B(f)^{-1}A_{j,a}(f)\}_j}{\{B(f)^{-1}\beta\}_j} > 0$.
Consequently, $\theta^* = min_k \frac{\{B(f)^{-1}A_{j,a}(f)\}_k}{\{B(f)^{-1}\beta\}_k} = min_{k \neq j} \frac{\{B(f)^{-1}A_{j,a}(f)\}_k}{\{B(f)^{-1}\beta\}_k} = min_{k \neq j} \frac{x_k(f)-x_k(g)}{x_j(g) \cdot x_j(f)} < 0$.
Therefore, $\theta = min_{k \neq j} \frac{x_k(f)-x_k(g)}{x_j(f)} = x_j(g) \cdot \theta^*$, implying that the conditions $x(f) \ngeq x(g)$ and $\beta^T\{v^\alpha(g^\infty) - v^\alpha(f^\infty)\} < \theta \cdot \{\overline{v} - x(f)^T r(f)\}$ are sufficient for the suboptimality of the action $g(j)$. □

3.6 Value iteration

In the method of *value iteration* the value vector v^α is successively approximated, starting with some guess v^1, by a sequence $\{v^n\}_{n=1}^\infty$ which converges to v^α. This method is also called *successive approximation*. In this method a *nearly optimal policy* is determined. When applying the policy iteration method (and also in principle in the linear programming method) one has to solve a system of N linear equations in each iteration. For a very large state space this might be prohibitive. The method of value iteration does not have this disadvantage. An iteration of this method is quite simple. In addition, sometimes this method can also be used to prove properties of the structure of optimal policies. On the other hand, especially for discount factors close to 1, the convergence can be very slow.

In this section we discuss the basic value iteration method including suboptimality tests. Most of the properties of the value iteration method are based on the theory of monotone contraction mappings and on the optimality equation (see the sections 3.2 and 3.3).

For $\delta > 0$ we call a vector $v \in \mathbb{R}^N$ a *δ-approximation* of v^α if $\|v^\alpha - v\|_\infty \leq \delta$; for $\varepsilon > 0$ a policy R is an *ε-optimal policy* if $\|v^\alpha - v^\alpha(R)\|_\infty \leq \varepsilon$.

From Corollary 3.3, part (2), it follows that $v^\alpha = lim_{n\to\infty} U^n x$ for every $x \in \mathbb{R}^N$. Define the sequence $v^1, v^2, \dots$ by

$$\begin{cases} v^1 \in \mathbb{R}^N & \text{arbitrarily chosen} \\ v^{n+1} := Uv^n & n = 1, 2, \dots \end{cases} \tag{3.40}$$

with corresponding sequence $f_1^\infty, f_2^\infty, \dots$ of policies, where $f_n = f_{v^n}$ for every $n \in \mathbb{N}$. Then, we have

$$v^{n+1} = Uv^n = L_{f_n} v^n = r(f_n) + \alpha P(f_n) v^n, \; n \in \mathbb{N}. \tag{3.41}$$

The next lemma shows that f_n^∞ is an ε-optimal policy for n sufficiently large.

Lemma 3.9
$\|v^\alpha(f_n^\infty) - v^\alpha\|_\infty \leq 2\alpha^n(1-\alpha)^{-1} \cdot \|v^2 - v^1\|_\infty, \; n \in \mathbb{N}$.

Proof
From Theorem 3.7, part (3), it follows that

$$\begin{aligned} \|v^\alpha(f_n^\infty) - v^\alpha\|_\infty & \leq 2\alpha(1-\alpha)^{-1} \cdot \|Uv^n - v^n\|_\infty = 2\alpha(1-\alpha)^{-1} \cdot \|Uv^n - Uv^{n-1}\|_\infty \\ & \leq 2\alpha^2(1-\alpha)^{-1} \cdot \|v^n - v^{n-1}\|_\infty \\ & \leq \cdots \leq 2\alpha^n(1-\alpha)^{-1} \cdot \|v^2 - v^1\|_\infty, \; n \in \mathbb{N}. \end{aligned}$$

□

Algorithm 3.4 *Value iteration (version 1)*
Input: Instance of a discounted MDP and some scalar $\varepsilon > 0$.
Output: An ε-optimal deterministic policy f^∞ and a $\frac{1}{2}\varepsilon$-approximation of the value vector v^α.

```
1. Select $x \in \mathbb{R}^N$.
2. a. Compute $y$ by $y_i := max_a\{r_i(a) + \alpha \sum_j p_{ij}(a)x_j\}, \; i \in S$.
   b. Choose $f(i) \in argmax_a\{r_i(a) + \alpha \sum_j p_{ij}(a)x_j\}, \; i \in S$.
3. if $\|y - x\|_\infty \leq \frac{1}{2}(1-\alpha)\alpha^{-1}\varepsilon$ then
         $f^\infty$ is an $\varepsilon$-optimal policy and $y$ is a $\frac{1}{2}\varepsilon$-approximation of the value vector $v^\alpha$ (STOP)
   else $x := y$ and return to step 2.
```

Theorem 3.24
Algorithm 3.4 is finite and correct.

Proof
Since the sequence $\{U^n x\}_{n=1}^\infty$ converges to v^α, the algorithm is finite. The algorithm terminates with some x, y and f, where $y = Ux$ and $f = f_x$. From the proof of Lemma 3.9 it follows that $\|v^\alpha(f^\infty) - v^\alpha\|_\infty \leq 2\alpha(1-\alpha)^{-1} \cdot \|y - x\|_\infty \leq \varepsilon$, i.e. f^∞ is an ε-optimal policy. Furthermore, $\|v^\alpha - y\|_\infty = \|Uv^\alpha - Ux\|_\infty \leq \alpha \cdot \|v^\alpha - x\|_\infty \leq \alpha(1-\alpha)^{-1} \cdot \|y - x\|_\infty \leq \frac{1}{2}\varepsilon$, the second last inequality by Theorem 3.7, part (2). □

Example 3.3
Consider the model of Example 3.1 and start with $x = (4, 4, 4)$ and $\varepsilon = 0.2$. The results of the computation are summarized below. The algorithm terminates as soon as the norm of the difference of two subsequent y-vectors is at most 0.1.

	Iteration						
	1	2	3	4	5	6	7
y_1	5.00	8.50	9.50	10.13	10.38	10.53	10.59
y_2	8.00	10.50	11.50	12.13	12.38	12.53	12.59
y_3	11.00	13.00	14.25	14.75	15.06	15.19	15.27
f_1	3	3	3	3	3	3	3
f_2	1	3	3	3	3	3	3
f_3	2	2	2	2	2	2	2

Hence, f^∞ with $f(1) = 3$, $f(2) = 3$ and $f(3) = 2$ is a 0.2-optimal policy and $(10.59, 12.59, 15.27)$ is a 0.1-approximation of v^α.

Remark
We see in the example that already after one iteration the optimal policy is found, although the approximation y is far away from v^α. This phenomenon occurs often when using the method of value iteration.

We now present an algorithm with a test for the *exclusion of suboptimal actions*, based on (3.15).

Algorithm 3.5 *Value iteration (version 2)*
Input: Instance of a discounted MDP and some scalar $\varepsilon > 0$.
Output: An ε-optimal deterministic policy f^∞ and a $\frac{1}{2}\varepsilon$-approximation of the value vector v^α.

```
1. Select $x \in \mathbb{R}^N$.
2. a. Compute $y$ by $y_i := max_a\, y_i(a)$, where $y_i(a) := r_i(a) + \alpha \sum_j p_{ij}(a)x_j,\ (i,a) \in S \times A$.
   b. Choose $f(i) \in argmax_a\, y_i(a),\ i \in S$.
3. if $\|y - x\|_\infty \leq \frac{1}{2}(1-\alpha)\alpha^{-1}\varepsilon$ then
         $f^\infty$ is an $\varepsilon$-optimal policy and $y$ is a $\frac{1}{2}\varepsilon$-approximation of the value vector $v^\alpha$ (STOP)
   else $x := y$ and go to to step 4.
4. a. Compute span by $span := max_i\,(y_i - x_i) - min_i\,(y_i - x_i)$.
   b. for all $(i,a) \in S \times A$ do
          if $y_i(a) < y_i - \alpha(1-\alpha)^{-1} \cdot span$, then $A(i) := A(i) - \{a\}$.
   c. if $\#A(i) = 1$ for every $i \in S$ then
          $f^\infty$ is an optimal policy and $v^\alpha := \{I - P(f)\}^{-1}r(f)$ is the value vector (STOP).
5. $x := y$ and return to step 2.
```

Theorem 3.25
Algorithm 3.5 is finite and correct.

Proof

Let $A^{(n)}(i)$ be the action set in state i in iteration n. Define the operator $U^{(n)} : \mathbb{R}^N \rightarrow \mathbb{R}^N$ by

$$\{U^{(n)}x\}_i = max_{a \in A^{(n)}(i)} \big\{r_i(a) + \alpha \sum_j p_{ij}(a)x_j\big\},\ i \in S.$$

Algorithm 3.5 computes the sequence $v^1, v^2, \dots$ where $v^{n+1} = U^{(n)}v^n$. Since the operator depends on n, we cannot simply use the general theory for contracting operators.

We first show the finiteness of the algorithm. Let the actions $b, c \in A(i)$ be such that

$b \in argmax_{a \in A^{(n)}(i)}\{r_i(a) + \alpha \sum_j p_{ij}(a)v_j^n\}$ and $c \in argmax_{a \in A^{(n-1)}(i)}\{r_i(a) + \alpha \sum_j p_{ij}(a)v_j^{n-1}\}$.

Since $A^{(n)}(i) \subseteq A^{(n-1)}(i)$, action $b \in A^{(n-1)}(i) \cap A^{(n)}(i)$ and we can write

$$\begin{aligned} v_i^{n+1} - v_i^n &\le \{r_i(b) + \alpha \textstyle\sum_j p_{ij}(b)v_j^n\} - \{r_i(b) + \alpha \textstyle\sum_j p_{ij}(b)v_j^{n-1}\} \\ &= \alpha \textstyle\sum_j p_{ij}(b)\{v_j^n - v_j^{n-1}\} \le \alpha \textstyle\sum_j p_{ij}(b) \cdot \|v^n - v^{n-1}\|_\infty = \alpha \cdot \|v^n - v^{n-1}\|_\infty. \end{aligned}$$

On the other hand, because $v_i^n = r_i(c) + \alpha \sum_j p_{ij}(c)v_j^{n-1}$, i.e. in the algorithm we have $y_i(c) = y_i$, action c is not excluded in step 4b of the algorithm. Hence, $c \in A^{(n)}(i)$ and we obtain

$$\begin{aligned} v_i^n - v_i^{n+1} &\le \{r_i(c) + \alpha \textstyle\sum_j p_{ij}(c)v_j^{n-1}\} - \{r_i(c) + \alpha \textstyle\sum_j p_{ij}(c)v_j^n\} \\ &= \alpha \textstyle\sum_j p_{ij}(c)\{v_j^{n-1} - v_j^n\} \le \alpha \textstyle\sum_j p_{ij}(c) \cdot \|v^n - v^{n-1}\|_\infty = \alpha \cdot \|v^n - v^{n-1}\|_\infty. \end{aligned}$$

Consequently, we have shown $\|v^{n+1} - v^n\|_\infty \le \alpha \cdot \|v^n - v^{n-1}\|_\infty \le \cdots \le \alpha^{n-1} \cdot \|v^2 - v^1\|_\infty$, i.e. the algorithm is finite.

Next, we show by induction on n that the suboptimality test is correct. The first iteration is correct. Suppose that the elimination is correct during the iterations $1, 2, \dots, n-1$ and consider iteration n. Above, it was shown that $U^{(n)}$ is a contraction with contraction factor α. Since no optimal actions are excluded, v^α is the fixed-point of $U^{(n)}$. Hence, by taking $U^{(n)}$ and $A^{(n)}(i)$ instead of U and $A(i)$, it follows from the general theory derived in Section 3.3 that the suboptimality test is correct.

Finally, we show that the algorithm terminates with an ε-optimal policy f^∞ and a $\frac{1}{2}\varepsilon$-approximation of v^α. Let m be the last iteration of the algorithm.

If $\#A(i) = 1$ for every $i \in S$, then obviously f^∞ is optimal. Otherwise, let f_x be such that $y = U^{(m)}x = L_{f_x}x$. Since v^α and $v^\alpha(f^\infty)$ are the fixed-points of $U^{(m)}$ and L_{f_x}, it follows (see Theorem 3.7) that

$$\|v^\alpha - v^\alpha(f_x^\infty)\|_\infty \le 2\alpha(1-\alpha)^{-1} \cdot \|U^{(m)}x - x\|_\infty = 2\alpha(1-\alpha)^{-1} \cdot \|y - x\|_\infty \le \varepsilon$$

and

$$\|v^\alpha - y\|_\infty = \|U^{(m)}v^\alpha - U^{(m)}x\|_\infty \le \alpha\|v^\alpha - x\|_\infty \le \alpha(1-\alpha)^{-1} \cdot \|y - x\|_\infty \le \tfrac{1}{2}\varepsilon.$$

□

Remarks

1. If the algorithm terminates in step 4c, then an optimal policy f^∞ is obtained, but the value vector v^α is unknown. Also it is unknown how good the approximation y is. In order to compute the exact value of v^α we have to solve the linear system $x = L_f x$.
2. It is not necessary to execute step 4 in each iteration; it can be done, for instance, periodically.

Example 3.3 (continued)

Iteration 1

$y_1(1) = 3, y_1(2) = 4, y_1(3) = 5 : y_1 = 5$. $y_2(1) = 8, y_2(2) = 6, y_2(3) = 7 : y_2 = 8$.
$y_3(1) = 10, y_3(2) = 11, y_3(3) = 9 : y_3 = 11$. $f(1) = 3, f(2) = 1, f(3) = 2$. $span = 6$.
No actions can be excluded.

Iteration 2

$y_1(1) = 3.5, y_1(2) = 6, y_1(3) = 8.5 : y_1 = 8.5$.
$y_2(1) = 7.5, y_2(2) = 8, y_2(3) = 10.5 : y_2 = 10.5$.
$y_3(1) = 10.5, y_3(2) = 13, y_3(3) = 12.5 : y_3 = 13$. $f(1) = 3, f(2) = 3, f(3) = 2$. $span = 1.5$.
In state 1 the actions 1 and 2 are excluded; in state 2 the actions 1 and 2 and in state 3 action 1.

Iteration 3

$y_1(3) = 9.5 : y_1 = 9.5$. $y_2(3) = 11.5 : y_2 = 11.5$. $y_3(2) = 14.25, y_3(3) = 13.5 : y_3 = 14.25$.
$f(1) = 3, f(2) = 3, f(3) = 2$. $span = 0.25$.
In state 3 actions 3 is excluded.
f^∞ with $f(1) = 3$, $f(2) = 3$ and $f(3) = 2$ is an optimal policy and $v^\alpha = (\frac{32}{3}, \frac{38}{3}, \frac{46}{3})$ is the value vector.

The method of value iteration is an iterative procedure to solve the functional equation $Ux = x$. In this section we discuss two variants of the standard procedure, the *Pre-Gauss-Seidel* and the *Gauss-Seidel* variant, respectively. These variants are based on contraction mappings with fixed-point v^α and with contraction factor at most α. Hence, they may be considered as accelerations of the basic algorithm.

Variant 1 (Pre-Gauss-Seidel)

In (3.40), v^{n+1} is computed from v^n by the formula

$$v_i^{n+1} := max_a\{r_i(a) + \alpha \sum_{j=1}^N p_{ij}(a)v_j^n\},\ i = 1, 2, \dots, N.$$

Since, in general, v^{n+1} is a better approximation of v^α than v^n, it seems favorable to use the values $v_1^{n+1}, v_2^{n+1}, \dots, v_{i-1}^{n+1}$ in the computation of v_i^{n+1} instead of $v_1^n, v_2^n, \dots, v_{i-1}^n$. So, the following formula is used:

$$v_i^{n+1} := max_a\{r_i(a) + \alpha \sum_{j=1}^{i-1} p_{ij}(a)v_j^{n+1} + \alpha \sum_{j=i}^N p_{ij}(a)v_j^n\},\ i = 1, 2, \dots, N. \tag{3.42}$$

This is the so-called *Pre-Gauss-Seidel* variant. Similar to the mappings L_π and U for the standard procedure, the Pre-Gauss-Seidel variant can be described by the operators $\overline{L}_\pi$ and $\overline{U}$, respectively, which are mappings from $\mathbb{R}^N$ to $\mathbb{R}^N$, defined by

$$\{\overline{L}_\pi x\}_i := r_i(\pi) + \alpha \sum_{j=1}^{i-1} p_{ij}(\pi)\{\overline{L}_\pi x\}_j + \alpha \sum_{j=i}^N p_{ij}(\pi)x_j,\ i = 1, 2, \dots, N, \tag{3.43}$$

and

$$\{\overline{U}x\}_i := max_a\{r_i(a) + \alpha \sum_{j=1}^{i-1} p_{ij}(a)\{\overline{U}x\}_j + \alpha \sum_{j=i}^N p_{ij}(a)x_j\},\ i = 1, 2, \dots, N. \tag{3.44}$$

For every $x \in \mathbb{R}^N$ the policy $\overline{f}_x^\infty$ is the policy that satisfies $\overline{L}_{\overline{f}_x} x = \overline{U}x$.

Theorem 3.26

The operators $\overline{L}_\pi$ and $\overline{U}$ are monotone contracting mappings with fixed-points $v^\alpha(\pi^\infty)$ and v^α, respectively, with contraction factor α.

Proof

We apply Lemma 3.1, part (2). Therefore, suppose that $x \leq y \leq d \cdot e$ for some scalar d. With induction on state i we will show that $\{\overline{L}_\pi x\}_i \leq \{\overline{L}_\pi y\}_i + \alpha \cdot |d|,\ i = 1, 2, \dots, N$.

For $i = 1$, we have

$$\begin{aligned}
\{\overline{L}_\pi x\}_1 &= \{L_\pi x\}_1 = r_1(\pi) + \alpha \textstyle\sum_{j=1}^N\ p_{1j}(\pi) x_j \\
&\leq r_1(\pi) + \alpha \textstyle\sum_{j=1}^N\ p_{1j}(\pi) y_j + \alpha \cdot |d| \textstyle\sum_{j=1}^N\ p_{1j}(\pi) = \{\overline{L}_\pi y\}_1 + \alpha \cdot |d|.
\end{aligned}$$

Suppose that $\{\overline{L}_\pi x\}_j \leq \{\overline{L}_\pi y\}_j + \alpha \cdot |d|$ for $j = 1, 2, \dots, i-1$. Then, we can write

$$\begin{aligned}
\{\overline{L}_\pi x\}_i &= r_i(\pi) + \alpha \textstyle\sum_{j=1}^{i-1}\ p_{ij}(\pi) \{\overline{L}_\pi x\}_j + \alpha \textstyle\sum_{j=i}^N\ p_{ij}(\pi) x_j \\
&\leq r_i(\pi) + \alpha \textstyle\sum_{j=1}^{i-1}\ p_{ij}(\pi) \{\overline{L}_\pi y\}_j + \alpha^2 \cdot |d| \textstyle\sum_{j=1}^{i-1}\ p_{ij}(\pi) \\
&\qquad\qquad + \alpha \textstyle\sum_{j=i}^N\ p_{ij}(\pi) y_j + \alpha \cdot |d| \textstyle\sum_{j=i}^N\ p_{ij}(\pi) \\
&= \{\overline{L}_\pi y\}_i + \alpha^2 \cdot |d| \textstyle\sum_{j=1}^{i-1}\ p_{ij}(\pi) + \alpha \cdot |d| \textstyle\sum_{j=i}^N\ p_{ij}(\pi) \\
&\leq \{\overline{L}_\pi y\}_i + \alpha \cdot |d| \textstyle\sum_{j=1}^N\ p_{ij}(\pi) = \{\overline{L}_\pi y\}_i + \alpha \cdot |d|.
\end{aligned}$$

Hence, by Lemma 3.1, part (2), $\overline{L}_\pi$ is a monotone contraction with contraction factor α. Again by induction on state i, one can easily show that $v^\alpha(\pi^\infty)$ satisfies (3.43), i.e. $v^\alpha(\pi^\infty)$ is the unique fixed-point of $\overline{L}_\pi$. The proof for $\overline{U}$ is similar, and is left to the reader. □

Lemma 3.10

(1) $\overline{U}x = sup_\pi\ \overline{L}_\pi x$ *for every* $x \in \mathbb{R}^N$.

(2) $\overline{f}_{v^\alpha}^\infty$ *is an* α*-discounted optimal policy.*

(3) $x - (1-\alpha)^{-1} \|\overline{U}x - x\|_\infty \cdot e \leq \overline{U}x - \alpha(1-\alpha)^{-1} \|\overline{U}x - x\|_\infty \cdot e \leq v^\alpha(\overline{f}_x^\infty) \leq v^\alpha \leq$
$\overline{U}x + \alpha(1-\alpha)^{-1} \|\overline{U}x - x\|_\infty \cdot e \leq x + (1-\alpha)^{-1} \|\overline{U}x - x\|_\infty \cdot e$.

(4) $\|v^\alpha - x\| \leq (1-\alpha)^{-1} \|\overline{U}x - x\|_\infty$.

(5) $\|v^\alpha(\overline{f}_x^\infty) - v^\alpha\| \leq 2\alpha(1-\alpha)^{-1} \|\overline{U}x - x\|_\infty$.

Proof

(1) By induction on i, we will show that $(\overline{L}_\pi x)_i \leq (\overline{U}x)_i$ for $i = 1, 2, \dots, N$.

For $i = 1$ the result is obvious and the induction step is

$$\begin{aligned}
\{\overline{L}_\pi x\}_i &= r_i(\pi) + \alpha \textstyle\sum_{j=1}^{i-1}\ p_{ij}(\pi) (\overline{L}_\pi x\}_j + \alpha \textstyle\sum_{j=i}^N\ p_{ij}(\pi) x_j \\
&\leq r_i(\pi) + \alpha \textstyle\sum_{j=1}^{i-1}\ p_{ij}(\pi) \{\overline{U}x\}_j + \alpha \textstyle\sum_{j=i}^N\ p_{ij}(\pi) x_j \\
&\leq max_a \Big\{ r_i(a) + \alpha \textstyle\sum_{j=1}^{i-1}\ p_{ij}(a) \{\overline{U}x\}_j + \alpha \textstyle\sum_{j=i}^N\ p_{ij}(a) x_j \Big\} = \{\overline{U}x\}_i.
\end{aligned}$$

Because $\overline{L}_{\overline{f}_x} = \overline{U}x$, it follows that $\overline{U}x = sup_\pi\ \overline{L}_\pi x$.

(2) Because $\overline{L}_{\overline{f}_{v^\alpha}} v^\alpha = \overline{U}v^\alpha = v^\alpha$, v^α is the fixed-point of $\overline{L}_{\overline{f}_{v^\alpha}}$, i.e. $v^\alpha = v^\alpha(\overline{f}_{v^\alpha}^\infty)$: Hence, $\overline{f}_{v^\alpha}^\infty$ is an α-discounted optimal policy.

The parts (3), (4) and (5) can be shown in a way analogously to the proof of Theorem 3.7. □

Lemma 3.11

(1) $\overline{U}(x + c \cdot e) \leq \overline{U}x + \alpha \cdot c \cdot e$ *for every* $x \in \mathbb{R}^N$ *and every* $c \geq 0$.

(2) $\overline{U}(x + c \cdot e) \geq \overline{U}x + \alpha \cdot c \cdot e$ *for every* $x \in \mathbb{R}^N$ *and every* $c \leq 0$.

Proof

Using induction on the state i, the proof is straightforward. □

Theorem 3.27
If $r_i(a) + \alpha \sum_{j=1}^{i-1} p_{ij}(a)\{\overline{U}x\}_j + \alpha \sum_{j=i}^{N} p_{ij}(a)x_j < \{\overline{U}x\}_i - 2\alpha(1-\alpha)^{-1}\|\overline{U}x - x\|_\infty$,
then action a *is suboptimal.*

Proof
By Lemma 3.10 part (3), we have

$$\begin{aligned} x - (1-\alpha)^{-1}\|\overline{U}x - x\|_\infty \cdot e &\le \overline{U}x - \alpha(1-\alpha)^{-1}\|\overline{U}x - x\|_\infty \cdot e \le v^\alpha \le \\ &\le \overline{U}x + \alpha(1-\alpha)^{-1}\|\overline{U}x - x\|_\infty \cdot e \le x + (1-\alpha)^{-1}\|\overline{U}x - x\|_\infty \cdot e. \end{aligned}$$

Therefore, we can write

$$\begin{aligned} v_i^\alpha &\ge \{\overline{U}x\}_i - \alpha(1-\alpha)^{-1}\|\overline{U}x - x\|_\infty \\ &> r_i(a) + \alpha \textstyle\sum_{j=1}^{i-1} p_{ij}(a)\{\overline{U}x\}_j + \alpha \sum_{j=i}^{N} p_{ij}(a)x_j + \alpha(1-\alpha)^{-1}\|\overline{U}x - x\|_\infty \\ &= r_i(a) + \alpha \textstyle\sum_{j=1}^{i-1} p_{ij}(a)\big\{\{\overline{U}x\}_j + (1-\alpha)^{-1}\|\overline{U}x - x\|_\infty\big\} + \alpha \sum_{j=i}^{N} p_{ij}(a)\big\{x_j + (1-\alpha)^{-1}\|\overline{U}x - x\|_\infty\big\} \\ &\ge r_i(a) + \alpha \textstyle\sum_{j=1}^{i-1} p_{ij}(a)\big\{\{\overline{U}x\}_j + \alpha(1-\alpha)^{-1}\|\overline{U}x - x\|_\infty\big\} + \alpha \sum_{j=i}^{N} p_{ij}(a)\big\{x_j + (1-\alpha)^{-1}\|\overline{U}x - x\|_\infty\big\} \\ &\ge r_i(a) + \alpha \textstyle\sum_{j=1}^{i-1} p_{ij}(a)v_j^\alpha + \alpha \sum_{j=i}^{N} p_{ij}(a)v_j^\alpha \\ &= r_i(a) + \alpha \textstyle\sum_{j=1}^{N} p_{ij}(a)v_j^\alpha \end{aligned}$$

□

From the previous results it follows that the following algorithm computes an ε-optimal policy whithin a finite number of iterations.

Algorithm 3.6 *Value iteration (Pre-Gauss-Seidel)*
Input: Instance of a discounted MDP and some scalar $\varepsilon > 0$.
Output: An ε-optimal deterministic policy f^∞ and a $\frac{1}{2}\varepsilon$-approximation of the value vector v^α.

```
1. Select $x \in \mathbb{R}^N$ arbitrary.

2. for $i = 1, 2, \ldots, N$ do
   begin
      $y_i := max_a\, y_i(a)$, where $y_i(a) := r_i(a) + \alpha\{\sum_{j=1}^{i-1} p_{ij}(a)y_j + \sum_{j=i}^{N} p_{ij}(a)x_j\},\ a \in A(i)$;
      choose $f(i) \in argmax_a\, y_i(a)$
   end

3. if $\|y - x\|_\infty \le \frac{1}{2}(1-\alpha)\alpha^{-1}\varepsilon$ then
         $f^\infty$ is an $\varepsilon$-optimal policy and $y$ is a $\frac{1}{2}\varepsilon$-approximation of the value vector $v^\alpha$ (STOP)
   else go to to step 4.

4. for all $(i, a) \in S \times A$ do
         if $y_i(a) < y_i - 2\alpha(1-\alpha)^{-1} \cdot \|y - x\|_\infty$, then $A(i) := A(i) - \{a\}$.

5. if $\#A(i) = 1$ for every $i \in S$ then
         $f^\infty$ is an optimal policy and $v^\alpha := \{I - P(f)\}^{-1}r(f)$ is the value vector (STOP).

6. $x := y$ and return to step 2.
```

Example 3.3 (continued)
Start with $x = (4, 4, 4)$. The computations can be represented by the following scheme:
$y_1(1) = 1 + \frac{1}{2}x_1;\ y_1(2) = 2 + \frac{1}{2}x_2;\ y_1(3) = 3 + \frac{1}{2}x_3;\ y_1 = max\{y_1(1), y_1(2), y_1(3)\}$.
$y_2(1) = 6 + \frac{1}{2}y_1;\ y_2(2) = 4 + \frac{1}{2}x_2;\ y_2(3) = 5 + \frac{1}{2}x_3;\ y_2 = max\{y_2(1), y_2(2), y_2(3)\}$.
$y_3(1) = 8 + \frac{1}{2}y_1;\ y_3(2) = 9 + \frac{1}{2}y_2;\ y_3(3) = 7 + \frac{1}{2}x_3;\ y_3 = max\{y_3(1), y_3(2), y_3(3)\}$.
Iteration 1
$y_1(1) = 3,\ y_1(2) = 4,\ y_1(3) = 5:\ y_1 = 5;\ f(1) = 3.$
$y_2(1) = 8.5,\ y_2(2) = 6,\ y_2(3) = 7:\ y_2 = 8.5;\ f(2) = 1.$
$y_3(1) = 10.5,\ y_3(2) = 13.25,\ y_3(3) = 9:\ y_3 = 13.25;\ f(3) = 2.$
$x = (5, 8.5, 13.25).$
Iteration 2
$y_1(1) = 3,\ y_1(2) = 6.25,\ y_1(3) = 9.61:\ y_1 = 9.61;\ f(1) = 3.$
$y_2(1) = 10.81,\ y_2(2) = 8.25,\ y_2(3) = 11.61:\ y_2 = 11.61;\ f(2) = 3.$
$y_3(1) = 12.81,\ y_3(2) = 14.81,\ y_3(3) = 13.61:\ y_3 = 14.81;\ f(3) = 2.$
$x = (9.61, 11.61, 14.81).$
Iteration 3
$y_1(1) = 5.81,\ y_1(2) = 7.81,\ y_1(3) = 10.41:\ y_1 = 10.41;\ f(1) = 3.$
$y_2(1) = 11.20,\ y_2(2) = 9.81,\ y_2(3) = 12.41:\ y_2 = 12.41;\ f(2) = 3.$
$y_3(1) = 13.20,\ y_3(2) = 15.20,\ y_3(3) = 14.41:\ y_3 = 15.20;\ f(3) = 2.$
$i = 1$: the actions 1 and 2 are excluded.
$i = 2$: action 2 is excluded.
$i = 3$: the action 1 is excluded.
$x = (10.41, 12.41, 15.20).$
Iteration 4
$y_1(1) = 10.60:\ y_1 = 10.60;\ f(1) = 3.$
$y_2(1) = 11.30,\ y_2(3) = 12.60:\ y_2 = 12.60;\ f(2) = 3.$
$y_3(2) = 15.30,\ y_3(3) = 14.60:\ y_3 = 15.30;\ f(3) = 2.$
$i = 2$: action 1 is excluded.
$i = 3$: action 3 is excluded.
f^∞ with $f(1) = 3,\ f(2) = 3$ and $f(3) = 2$ is an optimal policy and $v^\alpha = (\frac{32}{3}, \frac{38}{3}, \frac{46}{3})$ is the value vector.

Remarks

1. The convergence to the value vector is faster in the Pre-Gauss-Seidel variant than in the standard version. On the other side, the exclusion of suboptimal actions is, in general, not so successful.
2. The performance of the Pre-Gauss-Seidel variant depends on the ordering of the states. Therefore, it is worthwhile to apply the following scheme, in which the iterations are in pairs; the states are ordered in the usual way first and then reversed. Hence, a pair of iterations has the following scheme:

$$\begin{cases} y_i = max_a\{r_i(a) + \alpha \sum_{j=1}^{i-1} p_{ij}(a)y_j + \alpha \sum_{j=i}^{N} p_{ij}(a)x_j\},\ i = 1, 2, \dots, N; \\ z_i = max_a\{r_i(a) + \alpha \sum_{j=1}^{i} p_{ij}(a)y_j + \alpha \sum_{j=i+1}^{N} p_{ij}(a)z_j\},\ i = N, N-1, \dots, 1. \end{cases}$$

Variant 2 (Gauss-Seidel)
The idea of the Pre-Gauss-Seidel variant can be extended to the term with $j = i$. Then, formula (3.43) becomes (with L^* instead of $\overline{L}$):

$(L^*_\pi x)_i = r_i(\pi) + \alpha \sum_{j=1}^{i} p_{ij}(\pi)(L^*_\pi x)_j + \alpha \sum_{j=i+1}^{N} p_{ij}(\pi)x_j,\ i = 1,2,\dots,N,$
i.e.

$$(L^*_\pi x)_i = \{1-\alpha p_{ii}(\pi)\}^{-1}\Big\{r_i(\pi) + \alpha \sum_{j=1}^{i-1} p_{ij}(\pi)(L^*_\pi x)_j + \alpha \sum_{j=i+1}^{N} p_{ij}(\pi)x_j\Big\},\ i = 1,2,\dots,N. \tag{3.45}$$

The corresponding operator U^* and the maximizing decision rule f^*_x are defined by

$$(U^*x)_i = max_a \,\{1-\alpha p_{ii}(a)\}^{-1}\Big\{r_i(a) + \alpha \sum_{j=1}^{i-1} p_{ij}(a)(U^*x)_j + \alpha \sum_{j=i+1}^{N} p_{ij}(a)x_j\Big\},\ i = 1,2,\dots,N. \tag{3.46}$$

and $L^*_{f^*_x}x = U^*x$.

Theorem 3.28

(1) *The operator L^*_π is a monotone contraction with fixed-point $v^\alpha(\pi^\infty)$ and with contraction factor $\beta_\pi := \alpha \cdot max_i \frac{1-p_{ii}(\pi)}{1-\alpha p_{ii}(\pi)} \le \alpha$.*

(2) *The operator U^* is a monotone contraction with fixed-point v^α and with contraction factor $\beta := \alpha \cdot max_{i,a} \frac{1-p_{ii}(a)}{1-\alpha p_{ii}(a)}$.*

Proof

The proof is similar to the proof of Theorem 3.26 and is left to the reader (as Exercise 3.22).

Lemma 3.12

(1) $U^*x = sup_\pi\, L^*_\pi x$ *for every* $x \in \mathbb{R}^N$.

(2) $f^{*\infty}_{v^\alpha}$ *is an α-discounted optimal policy.*

(3) $x - (1-\beta)^{-1}\|U^*x - x\|_\infty \cdot e \le U^*x - \beta(1-\beta)^{-1}\|U^*x - x\|_\infty \cdot e \le v^\alpha(f^{*\infty}_x) \le v^\alpha \le U^*x + \beta(1-\beta)^{-1}\|U^*x - x\|_\infty \cdot e \le x + (1-\beta)^{-1}\|U^*x - x\|_\infty \cdot e.$

(4) $\|v^\alpha - x\|_\infty \le (1-\beta)^{-1}\|U^*x - x\|_\infty$.

(5) $\|v^\alpha(f^{*\infty}_x) - v^\alpha\|_\infty \le 2\beta(1-\beta)^{-1}\|U^*x - x\|_\infty$.

Proof

The proof is similar to the proof of Lemma 3.10. □

Lemma 3.13

(1) $U^*(x + c\cdot e) \le U^*x + \beta \cdot c \cdot e$ *for every* $x \in \mathbb{R}^N$ *and every* $c \ge 0$;

(2) $U^*(x + c\cdot e) \ge U^*x + \beta \cdot c \cdot e$ *for every* $x \in \mathbb{R}^N$ *and every* $c \le 0$.

Proof

The proof is similar to the proof of Lemma 3.11 □

Theorem 3.29

If $\{1-\alpha p_{ii}(a)\}^{-1}\Big\{r_i(a) + \alpha \sum_{j=1}^{i-1} p_{ij}(a)(U^*x)_j + \alpha \sum_{j=i+1}^{N} p_{ij}(a)x_j\Big\} < (U^*x)_i - 2\beta(1-\beta)^{-1}\cdot \|U^*x - x\|_\infty$, *then action a is suboptimal.*

Proof

The proof is similar to the proof of Theorem 3.27 and left to the reader (as Exercise 3.23). □

Algorithm 3.7 *Value iteration (Gauss-Seidel)*

Input: Instance of a discounted MDP and some scalar $\varepsilon > 0$.

Output: An ε-optimal deterministic policy f^∞ and a $\frac{1}{2}\varepsilon$-approximation of the value vector v^α.

```
1. Select $x \in \mathbb{R}^N$ arbitrary; $\beta := \alpha \cdot max_{i,a} \frac{1-p_{ii}(a)}{1-\alpha p_{ii}(a)}$.
2. for $i = 1, 2, \dots, N$ do
   begin
      $y_i := max_a\, y_i(a)$, where $y_i(a) := \{1-\alpha p_{ii}(a)\}^{-1}\Big\{r_i(a)+\alpha\sum_{j=1}^{i-1} p_{ij}(a)(U^*x)_j+\alpha\sum_{j=i+1}^{N} p_{ij}(a)x_j\Big\}$;
      choose $f(i) \in argmax_a\, y_i(a)$
   end
3. if $\|y-x\|_\infty \leq \frac{1}{2}(1-\beta)\beta^{-1}\varepsilon$ then
         $f^\infty$ is an $\varepsilon$-optimal policy and $y$ is a $\frac{1}{2}\varepsilon$-approximation of the value vector $v^\alpha$ (STOP)
   else go to to step 4.
4. for all $(i,a) \in S \times A$ do
         if $y_i(a) < y_i - 2\beta(1-\beta)^{-1} \cdot \|y-x\|_\infty$, then $A(i) := A(i) - \{a\}$.
5. if $\#A(i) = 1$ for every $i \in S$ then
         $f^\infty$ is an optimal policy and $v^\alpha := \{I - P(f)\}^{-1}r(f)$ is the value vector (STOP).
6. $x := y$ and return to step 2.
```

Example 3.3 (continued)

Start with $x = (4, 4, 4)$; $\beta = 0.5$. The computations are given by the following scheme:

$y_1(1) = 2;\ y_1(2) = 2 + \frac{1}{2}x_2;\ y_1(3) = 3 + \frac{1}{2}x_3;\ y_1 = max\{y_1(1), y_1(2), y_1(3)\};$
$y_2(1) = 6 + \frac{1}{2}y_1;\ y_2(2) = 8;\ y_2(3) = 5 + \frac{1}{2}x_3;\ y_2 = max\{y_2(1), y_2(2), y_2(3)\};$
$y_3(1) = 8 + \frac{1}{2}y_1;\ y_3(2) = 9 + \frac{1}{2}y_2;\ y_3(3) = 14;\ y_3 = max\{y_3(1), y_3(2), y_3(3)\};$

Iteration 1

$y_1(1) = 2,\ y_1(2) = 4,\ y_1(3) = 5:\ y_1 = 5;\ f(1) = 3.$
$y_2(1) = 8.5,\ y_2(2) = 8,\ y_2(3) = 7:\ y_2 = 8.5;\ f(2) = 1.$
$y_3(1) = 10.5,\ y_3(2) = 13.25,\ y_3(3) = 14:\ y_3 = 14;\ f(3) = 3.$
$x = (5, 8.5, 14).$

Iteration 2

$y_1(1) = 2,\ y_1(2) = 6.25,\ y_1(3) = 10:\ y_1 = 10;\ f(1) = 3.$
$y_2(1) = 11,\ y_2(2) = 8,\ y_2(3) = 12:\ y_2 = 12;\ f(2) = 3.$
$y_3(1) = 13,\ y_3(2) = 15,\ y_3(3) = 14:\ y_3 = 15;\ f(3) = 2.$
$x = (10, 12, 15).$

Iteration 3

$y_1(1) = 2,\ y_1(2) = 7,\ y_1(3) = 10.5:\ y_1 = 10.5;\ f(1) = 3.$
$y_2(1) = 11.25,\ y_2(2) = 8,\ y_2(3) = 12.5:\ y_2 = 12.5;\ f(2) = 3.$
$y_3(1) = 13.25,\ y_3(2) = 15.25,\ y_3(3) = 14:\ y_3 = 15.25;\ f(3) = 2.$
$i = 1$: the actions 1 and 2 are excluded.
$i = 2$: the actions 1 and 2 are excluded.
$i = 3$: the actions 1 and 3 are excluded.
$x = (10.41, 12.41, 15.20).$

f^∞ with $f(1) = 3,\ f(2) = 3$ and $f(3) = 2$ is an optimal policy and $v^\alpha = (\frac{32}{3}, \frac{38}{3}, \frac{46}{3})$ is the value vector.

Variant 3 (Relaxation and one-step look-ahead)
The idea in relaxation is to replace the iterand v^{n+1} by $\hat{v}^{n+1}$, where $\hat{v}^{n+1}$ is a linear combination of v^{n+1} and v^n:

$$\hat{v}^{n+1} := \omega v^{n+1} + (1-\omega)v^n = v^n + \omega(v^{n+1} - v^n) = v^n + \omega\delta^{n+1}, \tag{3.47}$$

where $\delta^{n+1} := v^{n+1} - v^n$; ω is called the *relaxation factor*. For $\omega = 1$, we obtain the standard value iteration algorithm, i.e. $\hat{v}^{n+1} = v^{n+1}$. Furthermore, we look one-step ahead and examine an *estimation* of v^{n+2}. This estimator, denoted by w^{n+1}, will replace v^{n+1} in the iteration scheme. Such an estimator has the prospect to be closer to v^{n+2} than v^{n+1}, and in this way to improve the speed of the convergence of the algorithm.

Hence, given the approximation v^n of v^α obtained in iteration n, the next iteration consists of three steps: (1) v^{n+1} is computed by one of the variants of value iteration; (2) the relaxation $\hat{v}^{n+1} := v^n + \omega\delta^{n+1}$, where $\delta^{n+1} := v^{n+1} - v^n$, is determined for some relaxation factor ω; (3) w^{n+1}, an estimator of v^{n+2}, is computed and this w^{n+1} is used in the subsequent iteration as the approximation v^{n+1} of v^α.

Case 1: The iteration scheme in step 1 is the standard value iteration.
In the standard value iteration we have $v^{n+1} = Uv^n = L_{f_n}v^n$. For w^{n+1} we take

$$w^{n+1} := r(f_n) + \alpha P(f_n)\hat{v}^{n+1} = r(f_n) + \alpha P(f_n)\{v^n + \omega\delta^{n+1}\} = v^{n+1} + \alpha\omega g^{n+1},$$

where $g^{n+1} := P(f_n)\delta^{n+1}$.

Case 2: The iteration scheme in step 1 is the Pre-Gauss-Seidel variant.
In this variant we have $v^{n+1} = \overline{U}v^n = \overline{L}_{f_n}v^n$. For w^{n+1} we take

$$w_i^{n+1} := r_i(f_n) + \alpha \textstyle\sum_{j=1}^{i-1} p_{ij}(f_n)w_j^{n+1} + \alpha \sum_{j=i}^N p_{ij}(f_n)\hat{v}_j^{n+1} \text{ for } i = 1,2,\dots,N.$$

Lemma 3.14
$w^{n+1} = v^{n+1} + \alpha\omega g^{n+1}$, where $g_i^{n+1} = \alpha \sum_{j=1}^{i-1} p_{ij}(f_n)g_j^{n+1} + \sum_{j=i}^N p_{ij}(f_n)\delta_j^{n+1}$ for $i = 1,2,\dots,N$.

Proof
We appy induction on the states. For $i = 1$, we have

$$\begin{aligned} w_1^{n+1} &= r_1(f_n) + \alpha \textstyle\sum_{j=1}^N p_{1j}(f_n)\hat{v}_j^{n+1} \\ &= r_1(f_n) + \alpha \textstyle\sum_{j=1}^N p_{1j}(f_n)(v_j^n + \omega\delta_j^{n+1}) \\ &= r_1(f_n) + \alpha \textstyle\sum_{j=1}^N p_{1j}(f_n)v_j^n + \alpha\omega \sum_{j=1}^N p_{1j}(f_n)\delta_j^{n+1} \\ &= v_1^{n+1} + \alpha\omega g_1^{n+1}. \end{aligned}$$

For the induction step, we obtain

$$\begin{aligned} w_i^{n+1} &= r_i(f_n) + \alpha \textstyle\sum_{j=1}^{i-1} p_{ij}(f_n)w_j^{n+1} + \alpha \sum_{j=i}^N p_{ij}(f_n)\hat{v}_j^{n+1} \\ &= r_i(f_n) + \alpha \textstyle\sum_{j=1}^{i-1} p_{ij}(f_n)\{v_j^{n+1} + \alpha\omega g_j^{n+1}\} + \alpha \sum_{j=i}^N p_{ij}(f_n)\{v_j^n + \omega\delta_j^{n+1}\} \\ &= r_i(f_n) + \alpha \textstyle\sum_{j=1}^{i-1} p_{ij}(f_n)v_j^{n+1} + \alpha \sum_{j=i}^N p_{ij}(f_n)v_j^n + \\ &\qquad \alpha\omega\{\alpha \textstyle\sum_{j=1}^{i-1} p_{ij}(f_n)g_j^{n+1} + \sum_{j=i}^N p_{ij}(f_n)\delta_j^{n+1}\} \\ &= v_i^{n+1} + \alpha\omega g_i^{n+1}. \end{aligned}$$

□

Case 3: The iteration scheme in step 1 is the Gauss-Seidel variant.
In this variant we have $v^{n+1} = U^*v^n = L_{f_n}^*v^n$. For w^{n+1} we take

$$w_i^{n+1} := \{1 - \alpha p_{ii}(f_n)\}^{-1}\{r_i(f_n) + \alpha \textstyle\sum_{j=1}^{i-1} p_{ij}(f_n)w_j^{n+1} + \alpha \sum_{j=i+1}^N p_{ij}(f_n)\hat{v}_j^{n+1} \text{ for } i = 1,2,\dots,N.$$

Lemma 3.15
$w^{n+1} = v^{n+1} + \alpha\omega g^{n+1}$, where $g_i^{n+1} = \{1 - \alpha p_{ii}(f_n)\}^{-1}\{\alpha \sum_{j=1}^{i-1} p_{ij}(f_n)g_j^{n+1} + \sum_{j=i+1}^{N} p_{ij}(f_n)\delta_j^{n+1}\}$ for $i = 1, 2, \dots, N$.

Proof
The proof is similar to the proof of Lemma 3.14. □

To summarize: in all three cases we have $w^{n+1} = v^{n+1} + \alpha\omega g^{n+1}$, where v^{n+1} and g^{n+1} are dependent of the method which is used. In addition, we want to choose ω so that the resulting algorithm has a considerably improved convergence. Since $\delta^{n+2} = v^{n+2} - v^{n+1}$, as estimator of δ^{n+2} we take $w^{n+1} - \hat{v}^{n+1}$ and we denote this estimator by $\hat{\delta}^{n+1}$. So,

$$\hat{\delta}^{n+1} := w^{n+1} - \hat{v}^{n+1} = (v^{n+1} + \alpha\omega g^{n+1}) - (v^n + \omega\delta^{n+1}) = \delta^{n+1} + \omega(\alpha g^{n+1} - \delta^{n+1}). \tag{3.48}$$

In order to find the best value of ω, we consider

$$D(\omega) := span\, \hat{\delta}^{n+2} = M(\omega) - m(\omega), \tag{3.49}$$

where

$$M(\omega) = max_i\, \hat{\delta}_i^{n+2} = max_i\, \{\delta_i^{n+1} + \omega h_i^{n+1}\} \text{ and } m(\omega) = min_i\, \hat{\delta}^{n+2} = min_i\, \{\delta_i^{n+1} + \omega h_i^{n+1}\}, \tag{3.50}$$

with $h^{n+1} := \alpha g^{n+1} - \delta^{n+1}$.

Since $M(\omega)$ is the maximum of a set of linear functions in ω, $M(\omega)$ is a piecewise linear convex function (the slopes of the line segments are nondecreasing in ω); similarly, $m(\omega)$ is a piecewise linear concave function (the slopes of the line segments are nonincreasing in ω). Hence, $D(\omega)$ is a piecewise linear, nonnegative function. It seems plausible to find ω such that $D(\omega)$ is minimized. Therefore, it is sufficient to examine $D(\omega)$ only at the endpoints of the segments of the linear functions. This implies that the optimal value ω is found in one of the breakpoints, either of $M(\omega)$ or of $m(\omega)$.

Below we present an algorithm for the computation of ω^*. In step 1 we start with the values $\omega^* = 0$, $M = M(\omega^*) = M(0)$ and H, the slope of the first line segment of $M(\omega)$. In step 2, we find the endpoint $\omega^* + \omega_1$ of the first line segment of $M(\omega)$ at the right side of ω^* and the slope h_k^{n+1} of the first line segment of $M(\omega)$ at the right side of $\omega^* + \omega_1$. In step 3 the slope h of the first line segment of $m(\omega)$ at the right side of $\omega^* + \omega_1$ is determined. Next, we consider the three possible situations: (1) if $H \leq h$ and $h_k^{n+1} \geq h$, then - since the slopes of the line segments of $M(\omega)$ and $m(\omega)$ are increasing and decreasing, respectively - $\omega^* + \omega_1$ is the value of ω that minimizes $D(\omega)$ and we stop (see step 4 in the algorithm); (2) if $H \leq h$ and $h_k^{n+1} < h$, the minimum of $D(\omega)$ is to the right and we continue the procedure, i.e. we take $\omega^* := \omega^* + \omega_1$ and update some values (see the steps 4 and 7 in the algorithm); (3) if $H > h$, then the minimum of $D(\omega)$ is in the interval $[\omega^*, \omega^* + \omega_1]$ and we consider $m(\omega)$ at the right side of $m(\omega^*)$, which is done in step 8 of the next algorithm. In this step 8, we use the fact that $m(\omega) = min_i\, \{\delta_i^{n+1} + \omega h_i^{n+1}\}$ is equivalent to $max_i\, \{-\delta_i^{n+1} + \omega(-h_i^{n+1})\}$. In the algorithm below we omit the iteration index, i.e. we denote δ and h instead of δ^{n+1} and h^{n+1}, respectively.

Algorithm 3.8 *Computation of the relaxation factor ω^**
Input: Two vectors $\delta,\ h \in \mathbb{R}^N$.
Output: *A scalar* ω^* such that $D(\omega^*) = min_\omega\, D(\omega)$, where $D(\omega) := M(\omega) - m(\omega)$ with $M(\omega) = max_i\, (\delta_i + \omega h)$ and $m(\omega) = min_i\, (\delta_i + \omega h)$.

1. $\omega^* := 0$; let k be such that $\delta_k = max_i\, \delta_i$ (if k is not unique, select under the candidates the state with the highest h_i); $M := \delta_k$ and $H := h_k$.

```
2. $\omega_1 := min_{\{i \mid h_i > H\}} \left\{\frac{M-\delta_i}{h_i-H}\right\}$ and let $k$ such that $\omega_1 = \frac{M-\delta_k}{h_k-H}$.
3. Let $r$ be such that $min_i\,\{\delta_i + \omega_1 h_i\} = \delta_r + \omega_1 h_r$; $h := h_r$.
4. if $H \leq h$ and $h_k \geq h$ then $\omega^* := \omega^* + \omega_1$ (STOP).
   else go to step 5.
5. if $H \leq h$ and $h_k < h$ then go to step 7.
   else go to step 6.
6. if $H > h$ then go to step 8.
7. $\omega^* := \omega^* + \omega_1$;  $\delta_i := \delta_i + \omega_1 h_i,\ i \in S$;  $M := \delta_k$;  $H := h_k$; return to step 2.
8. $\delta_i := -\delta_i,\ i \in S$;  $h_i := -h_i,\ i \in S$; let $k$ be such that $\delta_k = max_i\,\delta_i$; $M := \delta_k$;  $H := h_k$;
   return to step 2.
```

We now formulate an algorithm for the value iteration method with relaxation and one-step look-ahead. In each iteration one of the three variants of value iteration can be chosen.

Algorithm 3.9 *Value iteration method with relaxation and one-step look-ahead (3 variants)*
Input: Instance of a discounted MDP and some scalar $\varepsilon > 0$.
Output: A deterministic policy f^∞ and a vector w, which are approximations of the optimal policy f_*^∞ and the value vector v^α, respectively.

```
1. Select $x \in \mathbb{R}^N$ arbitrary.
2. Make the choice which variant is used in the next iteration: variant 1 (standard value iteration),
   variant 2 (Pre-Gauss-Seidel) or variant 3 (Gauss-Seidel).
3. if variant 1 is chosen in step 2 then
           begin compute $y$ and $f^\infty$ as in step 2 of Algorithm 3.4; go to step 4 end
   else
      begin if variant 2 is chosen in step 2 then
                 begin compute $y$ and $f^\infty$ as in step 2 of Algorithm 3.6; go to step 4 end
            else
               begin if variant 3 is chosen in step 2 then
                        begin compute $y$ and $f^\infty$ as in step 2 of Algorithm 3.7; go to step 4 end
               end
      end
4. $\delta := y - x$.
5. if variant 1 is chosen in step 2 then
         begin for $i = 1, 2, \ldots, N$ do $g_i := \sum_{j=1}^N p_{ij}(f)\delta_j$; go to step 6 end
   else
      begin if variant 2 is chosen in step 2 then
               begin for $i = 1, 2, \ldots, N$ do $g_i := \alpha \sum_{j=1}^{i-1} p_{ij}(f) g_j + \sum_{j=i}^N p_{ij}(f)\delta_j$; go to step 6 end
            else
            begin if variant 3 is chosen in step 2 then
                  begin for $i = 1, 2, \dots, N$ do
                        $g_i := \{1 - \alpha p_{ii}(f)\}^{-1}\{\alpha \sum_{j=1}^{i-1} p_{ij}(f)g_j + \sum_{j=i+1}^{N} p_{ij}(f)\delta_j\}$;
                      go to step 6
                  end
            end
      end
6. $h := \alpha g - \delta$; compute $\omega^*$ by Algorithm 3.8; $w := y + \alpha\omega^* g$.
7. if $\|w - x\| \leq \varepsilon$ then
      $f^\infty$ and $w$ are approximations of the optimal policy $f_*^\infty$ and the value vector $v^\alpha$ (STOP)
   else return to step 2.
```

Herzberg and Yechiali ([116]) have proposed Algorithm 3.9 and they have tested this algorithm on several problems. Their numerical results reveal considerable reductions in computation time when compared to other value iteration schemes.

3.7 Value iteration and bisection

The convergence of value iteration is very slow if the discount factor α is close to 1. In value iteration to improve the accuracy of v^n, the approximation of the value vector v^α in iteration n, by a factor 10, we need roughly $-\{log_{10}\,\alpha\}^{-1}$ additional iterations. If $\alpha = 0.999$ that means about 2300 additional iterations. For α close to 1 a rate of convergence independent of a is especially advantageous. The method of bisection is such a computational scheme. It need $log_2\, 10 \approx 3.32$ iterations to improve the accuracy by a factor 10.

The bisection method is based to the following principles. Assume that an upper and a lower bound, say $\overline{v}$ and $\underline{v}$ respectively, of the value vector v^α are known: $\underline{v} \leq v^\alpha \leq \overline{v}$. Then, the interval $[\underline{v}, \overline{v}]$ is intersected into two halves (the bisection step). The procedure will be repeated with that half that includes v^α. However, the bisection method is only applicable in completely ordered Banach spaces, whereas our problem operates in $\mathbb{R}^N$, which is only a partial ordered space. For this reason the bisection method is in this case not transferable in a straightforward manner. It can happen that v^α is neither situated in the first half nor completely in the second half.

Therefore, we must compound the method of bisection with the value iteration in a suitable way. Moreover, we have to use the monotonicity property of the operators L_f and U. This is achieved by the following method, which consists of five steps. In broad outlines these five steps are as follows.

1. Computation of the starting interval $[\underline{v}, \overline{v}]$ such that $\underline{v} \leq v^\alpha \leq \overline{v}$.
2. Bisection: compute $v := \frac{1}{2}(\underline{v} + \overline{v})$.
3. Termination criterion: if $\|\overline{v} - \underline{v}\|_\infty < \varepsilon$ then stop with policy f_v^∞ as an η-optimal policy, where $\eta := \frac{\alpha(1+\alpha)}{1-\alpha} \cdot \varepsilon$, and with v as $\frac{1}{2}\varepsilon$-approximation of the value vector v^α.
4. Test which of the following situations has happened:
   a. we can conclude $\underline{v} \leq v^\alpha \leq v$;
   b. we can conclude $v \leq v^\alpha \leq \overline{v}$;
   c. none of the cases a and b.
5. Perform some iterations of the value iteration method until $v^{n+l} \geq v^n$, in which case $v^\alpha \geq v^n$, or $v^{n+l} \leq v^n$, in which case $v^\alpha \leq v^n$.

In the sequel we first elaborate some of the above steps and the we formulate the complete algorithm

Step 1: Computation of the starting interval
a. Select some v^0.
b. Compute $v^1 := Uv^0,\ f_i := f_{v^0}$ and $\underline{v} := v^\alpha(f_1^\infty)$.
c. Compute $\overline{v} := v^1 + \frac{\alpha}{1-\alpha} \cdot \|v^1 - v^0\|_\infty \cdot e$.
Remark
Obviously $\underline{v} = v^\alpha(f_1^\infty) \le v^\alpha$. From Theorem 3.7 part (1) follows $v^\alpha \le v^1 + \frac{\alpha}{1-\alpha} \cdot \|v^1 - v^0\|_\infty \cdot e = \overline{v}$.

Step 3: Termination
Since $\|v^\alpha - v\|_\infty \le \frac{1}{2} \cdot \|\overline{v} - \underline{v}|_\infty \le \frac{1}{2}\varepsilon$, v is a $\frac{1}{2}\varepsilon$-approximation of the value vector v^α. By Theorem 3.7 part (3), we also have

$$\begin{aligned}
\|v^\alpha - v^\alpha(f_v^\infty)\|_\infty &\le 2\alpha(1-\alpha)^{-1} \cdot \|Uv - v\|_\infty \\
&\le 2\alpha(1-\alpha)^{-1} \cdot \{\|Uv - v^\alpha\|_\infty + \|v^\alpha - v\|_\infty\} \\
&\le 2\alpha(1-\alpha)^{-1} \cdot \{\alpha \cdot \|v - v^\alpha\|_\infty + \|v^\alpha - v\|_\infty\} \\
&= 2\alpha(1-\alpha)^{-1}(\alpha+1) \cdot \|v - v^\alpha\|_\infty \le \eta.
\end{aligned}$$

Therefore, f_v^∞ as an η-optimal policy, where $\eta := \frac{\alpha(1+\alpha)}{1-\alpha} \cdot \varepsilon$.

Step 4: The testing procedure
To test whether $\underline{v} \le v^\alpha \le v$ is not so easy as it looks, because v^α is unknown. We make use of the monotonicity property of the operators L_f and U. Starting from the bisection point $v = \frac{1}{2}(\underline{v} + \overline{v})$ we look in which direction the value iteration method would lead. If $Uv \le v$, then we can conclude $v^\alpha \le v$ and case a is verified. Similarly, if $Uv \ge v$, then we can conclude $v^\alpha \ge v$ and case b is verified.
Remark
In order to conclude $(Uv)_i \ge v_i$, it is not always necessary to compute $max_a\, \{r_i(a) + \alpha \sum_j p_{ij}(a)v_j\}$. For instance, as soon as $r_i(a) + \alpha \sum_j p_{ij}(a)v_j \ge v_i$ for some action $a \in A(i)$, we may conclude $(Uv)_i \ge v_i$. Therefore, we adapt the computation of Uv and the resulting vector will be denoted by Tv.
We now describe the computation of Tv. Let $s_i(a,v) := r_i(a) + \alpha \sum_j p_{ij}(a)v_j$ for all $(i,a) \in S \times A$. The first component $(Tv)_1$ is defined as follows. We computed $s_1(a,v)$ for each $a \in A(i)$, one at the time, until $s_1(a,v) \ge v_1$. As soon as $s_1(a,v) \ge v_1$ for some $a \in A(i)$, say for $a = a_1$, we define $(Tv)_1 := s_1(a,v)$. If there does not exist such a_1, we define $(Tv)_1 := max_a\, s_1(a,v) < v_1$.
Next, we define $(Tv)_2$ similarly. Then, there are the following 4 possibilities:
(1) $(Tv)_1 \ge v_1$ and $(Tv)_2 \ge v_2$;
(2) $(Tv)_1 \ge v_1$ and $(Tv)_2 = (Uv)_2 < v_2$;
(3) $(Tv)_1 = (Uv)_1 < v_1$ and $(Tv)_2 \ge v_2$;
(4) $(Tv)_1 = (Uv)_1 < v_1$ and $(Tv)_2 = (Uv)_2 < v_2$.
In the cases (2) and (3), we stop because bisection can not be applied. In the cases (1) and (4) we continue similarly for $i = 3, 4, \dots, N$. Hence, we end with the following possibilities.
Case 1: $(Tv)_i \ge v_i$ for all $i \in S$.
In this case we have actions $a_i \in A(i)$ such that $s_i(a_i, v) \ge v_i$ for all $i \in S$. Let policy f^∞ defined by $f(i) := a_i,\ i \in S$ Then, $Uv \ge L_f v \ge v$, implying $v^\alpha \ge v$, and consequently $v^\alpha \in [v, \overline{v}]$.
Case 2: $(Tv)_i \le v_i$ for all $i \in S$.
In this case $max_a\, s_i(a,v) \le v_i$ for all $i \in S$. Then, we have $Uv \le v$, implying $v^\alpha \le v$, and consequently $v^\alpha \in [\underline{v}, v]$.

Case 3: This case occurs when case 1 and 2 are not satisfied. Then the testing procedure was interrupted because bisection can not be applied.

Step 5: Value iteration
If we terminate the test procedure because we cannot use the bisection method, we execute the following procedure.

a. Compute w := Uv.
b. If $\|w - v\|_\infty \leq \frac{1-\alpha}{2\alpha} \cdot \varepsilon$, then stop with the ε-optimal policy $f^\infty := f_v^\infty$ and with w as $\frac{1}{2}\varepsilon$-approximation of the value vector v^α.
c. If $w \geq v$, then $v^\alpha \geq v$ and consequently $v^\alpha \in [v, \overline{v}]$ and continue with bisection.
If $w \leq v$, then $v^\alpha \leq v$ and consequently $v^\alpha \in [\underline{v}, v]$ and continue with bisection.
d. $v := w$ and repeat the value iteration (step 4).

Remark
If $\|w - v\|_\infty \leq \frac{1-\alpha}{2\alpha} \cdot \varepsilon$, then by Theorem 3.7 part (3) f_v^∞ is an ε-optimal policy. Furthermore, by Theorem 3.7 part (2), $\|v^\alpha - w\|_\infty \leq \alpha \cdot \|v^\alpha - v\|_\infty \leq \frac{1}{2}\varepsilon$, i.e. w is $\frac{1}{2}\varepsilon$-approximation of the value vector v^α.

Combining the above elements yields the following algorithm.

Algorithm 3.10 *Value iteration and bisection*
Input: Instance of a discounted MDP and some scalar $\varepsilon > 0$.
Output: A deterministic policy f^∞ and a vector w, which are approximations of the optimal policy f_*^∞ and the value vector v^α, respectively.

```
1. Initialization:
   $\eta := \frac{\alpha(1+\alpha)}{1-\alpha} \cdot \varepsilon$; select some $x \in \mathbb{R}^N$; $y := Ux$; $\underline{v} := v^\alpha(f_x^\infty)$; $\overline{v} := y + \frac{\alpha}{1-\alpha} \cdot \|y - x\|_\infty$.
2. Bisection
   $v := \frac{1}{2}(\underline{v} + \overline{v})$.
3. Termination criterion:
   if $\|\overline{v} - \underline{v}\|_\infty < \varepsilon$ then
      begin determine $f_v$ such that $Uv = L_{f_v} v$;
            $f_v^\infty$ is an $\eta$-optimal policy and $v$ is a $\frac{1}{2}\varepsilon$-approximation of the value vector $v^\alpha$
      end (STOP).
4. Testing procedure:
   (a) Computation of $(Tv)_1$:
       for all $a \in A(i)$ do
          begin $s_1(a, v) := r_1(a) + \alpha \sum_j p_{1j}(a) v_j$;
                if $s_1(a, v) \geq v_1$ then begin $a_1 := a$; $(Tv)_1 := s_1(a, v)$; go to step 4(b) end
          end
       $(Tv)_1 := max_a\, s_1(a, v)$
   (b) Computation of $(Tv)_i$, $i \geq 2$:
       if $(Tv)_1 \geq v_1$ then
       begin for $i = 2, 3, \ldots, N$ do
             begin $stop := 0$;
```

```
                for all $a \in A(i)$ do
                begin while $stop = 0$ do
                               begin $s_i(a,v) := r_i(a) + \alpha \sum_j p_{ij}(a)v_j$;
                                     if $s_i(a,v) \geq v_i$ then
                                           begin $a_i := a$; $(Tv)_i := s_i(a,v)$; $stop := 1$ end
                               end
                end
                if $stop = 0$ then go to step 5
          end
          $\underline{v} := v$; go to step 2
    end
    if $(Tv)_1 < v_1$ then
    begin for $i = 2, 3, \ldots, N$ do
          begin for all $a \in A(i)$ do
                  begin $s_i(a,v) := r_i(a) + \alpha \sum_j p_{ij}(a)v_j$;
                        if $s_i(a,v) > v_i$ then go to step 5
                  end
          end
          $\overline{v} := v$; go to step 2
    end
```

5. *Value iteration:*

   (a) $w := Uv$.

   (b) **if** $\|w - v\|_\infty \leq \frac{1-\alpha}{2\alpha} \cdot \varepsilon$ **then** f_v^∞ is an ε-optimal policy and w is a $\frac{1}{2}\varepsilon$-approximation of the value vector v^α (STOP).

   (c) **if** $w \geq v$ **then begin** $\underline{v} := w$; **go to** step 3 **end**

   (d) **if** $w \leq v$ **then begin** $\overline{v} := w$; **go to** step 3 **end**

   (e) $v := w$; **go to** step 5 (a).

Remark

Computational experiments show (see [10]) that this approach is very suitable for MDPs with a discount factor close to 1.

3.8 Modified Policy Iteration

In step 2 of the policy iteration method (see Algorithm 3.1) we determine $v^\alpha(f^\infty)$ as unique solution of the linear system $L_f\, x = x$, i.e.

$$\{I - \alpha P(f)\}x = r(f). \tag{3.51}$$

In a model with N states, this requires $\mathcal{O}(N^3)$ elementary operations (e.g. additions and multiplications). Hence, for large N, obtaining an exact solution of (3.51) may be computationally prohibitive. In section

3.4 we have shown (see (3.25)) that, for consecutive policies f^∞ and g^∞ in Algorithm 3.1, where g is chosen by rule (3.21), and with $x = v^\alpha(f^\infty)$ and $y = v^\alpha(g^\infty)$,

$$y = x + \{I - \alpha P(f)\}^{-1}\{Ux - x\} = x + \sum_{t=0}^{\infty} \{\alpha P(f)\}^t\{Ux - x\}. \tag{3.52}$$

In the *modified policy iteration* method the matrix $A := \sum_{t=0}^{\infty} \{\alpha P(f)\}^t$ is truncated by

$$A^{(k)} := \sum_{t=0}^{k-1} \{\alpha P(f)\}^t \text{ for some } 1 \leq k \leq \infty.$$

For $k = 1$, $A^{(1)} = I$, and formula (3.52) becomes $y = x + (Ux - x) = Ux$, i.e. the modified policy iteration method is value iteration; for $k = \infty$, $A^{(\infty)} = A$, and formula (3.52) is the policy iteration method. For $1 < k < \infty$, the modified policy iteration method may be considered as a combination of policy iteration and value iteration. Policy iteration may be viewed as Newton's method for the solution of the optimality equation $Ux = x$. Similarly, the modified policy iteration method can be considered as an inexact Newton method.

We allow different values of k to be chosen in each iteration and we denote by $k(n)$ the value of k in iteration n. Hence, we obtain

$$\begin{aligned}
x^{n+1} &= x^n + A^{(k(n))}\{Ux^n - x^n\} \\
&= x^n + \textstyle\sum_{t=0}^{k(n)-1} \{\alpha P(f_n)\}^t\{r(f_n) + \alpha P(f_n)x^n - x^n\} \\
&= r(f_n) + \alpha P(f_n)r(f_n) + \cdots + \big\{\alpha P(f_n)\big\}^{k(n)-1} r(f_n) + \big\{\alpha P(f_n)\big\}^{k(n)} x^n \\
&= \big\{L_{f_n}\big\}^{k(n)} x^n.
\end{aligned}$$

The modified policy iteration method is presented in the following algorithm. The correctness of this algorithm is shown in Theorem 3.31.

Algorithm 3.11 *Modified policy iteration*
Input: Instance of a discounted MDP and some scalar $\varepsilon > 0$.
Output: An ε-optimal deterministic policy f^∞.

```
1. Select $x \in \mathbb{R}^N$.
2. a. Choose any $k$ with $1 \leq k \leq \infty$.
   b. Determine $f$ such that $L_f x = Ux$.
   c. if $\|Ux - x\|_\infty \leq \frac{1}{2}(1-\alpha)\alpha^{-1}\varepsilon$ then $f^\infty$ is an $\varepsilon$-optimal policy (STOP).
3. a. $y := \{L_f\}^k x$.
   b. $x := y$ and return to step 2.
```

Example 3.4
Consider the model of Example 3.1, start with $x = (\frac{28}{3}, 8, \frac{28}{3})$, let $\varepsilon = 0.2$ and take $k = 2$ in each iteration. Notice that $\frac{1}{2}(1-\alpha)\alpha^{-1}\varepsilon = 0.1$.

Iteration 1
$Ux = (\frac{28}{3}, \frac{34}{3}, \frac{40}{3})$; $f(1) = f(2) = f(3) = 3$.
$y = (\frac{29}{3}, \frac{35}{3}, \frac{41}{3})$; $x = (\frac{29}{3}, \frac{35}{3}, \frac{41}{3})$.

Iteration 2
$Ux = (9.833, 11.833, 14.833)$; $f(1) = f(2) = 3$, $f(3) = 2$.
$y = (10.417, 12.417, 14.917)$; $x = (10.417, 12.417, 14.917)$.

Iteration 3
$Ux = (10.459, 12.459, 15.209)$; $f(1) = f(2) = 3$, $f(3) = 2$.
$y = (10.604, 12.604, 15.229)$; $x = (10.604, 12.604, 15.229)$.

Iteration 4
$Ux = (10.615, 12.615, 15.302)$; $f(1) = f(2) = 3$, $f(3) = 2$.
$y = (10.651, 12.651, 15.308)$; $x = (10.651, 12.651, 15.308)$.

Iteration 5
$Ux = (10.654, 12.654, 15.309)$; $f(1) = f(2) = 3$, $f(3) = 2$.
f^∞ is an ε-optimal policy.

Let $x^1, x^2, \ldots$ be subsequent approximations of v^α, obtained by Algorithm 3.11. Then,

$$x^{n+1} = \{L_{f_n}\}^{k(n)}\, x^n,\ n = 1, 2, \ldots. \tag{3.53}$$

Since the operator depends on n, it is not obvious from the general theory that this operator is monotone and/or contracting. The next example shows that, in general, the operator $\{L_{f_n}\}^{k(n)}$ is neither a contraction nor is it monotone.

Example 3.5
Let $S = \{1, 2\}$; $A(1) = A(2) = \{1, 2\}$; $\alpha = \frac{3}{4}$. $r_1(1) = 1,\ r_1(2) = 0,\ r_2(1) = 1,\ r_2(2) = 0$.
$p_{11}(1) = 0,\ p_{12}(1) = 1;\ p_{11}(2) = 1,\ p_{12}(2) = 0;\ p_{21}(1) = 0,\ p_{22}(1) = 1;\ p_{21}(2) = 1,\ p_{22}(2) = 0$.
In an iteration, x will be transformed to $\{L_{f_x}\}^k x$ for some k, where f_x satisfies $Ux = L_{f_x}\, x$.
Let $x = (3, 0)$, then $(Ux)_1 = max\{1 + \alpha \cdot 0, 0 + \alpha \cdot 3\} = \frac{9}{4}$, $(Ux)_2 = max\{1 + \alpha \cdot 0, 0 + \alpha \cdot 3\} = \frac{9}{4}$.
Hence, $f_x(1) = f_x(2) = 2$, and consequently, $r(f_x) = \binom{0}{0}$ and $P(f_x) = \begin{pmatrix} 1 & 0 \\ 1 & 0 \end{pmatrix}$, so $\{P(f_x)\}^t = \begin{pmatrix} 1 & 0 \\ 1 & 0 \end{pmatrix}$ for all $t \geq 1$. Therefore, $\{L_{f_x}\}^k\, x = \left(\frac{3}{4}\right)^k \{P(f_x)\}^k x = \left(\frac{3}{4}\right)^k \binom{3}{3}$.
Next, let $y = (0, 0)$, then $(Uy)_1 = max\{1 + \alpha \cdot 0, 0 + \alpha \cdot 0\} = 1$, $(Uy)_2 = max\{1 + \alpha \cdot 0, 0 + \alpha \cdot 0\} = 1$.
Hence, $f_y(1) = f_y(2) = 1$, and consequently, $r(f_y) = \binom{1}{1}$ and $P(f_y) = \begin{pmatrix} 0 & 1 \\ 0 & 1 \end{pmatrix}$, so $\{P(f_x)\}^t = \begin{pmatrix} 0 & 1 \\ 0 & 1 \end{pmatrix}$ for all $t \geq 1$. Therefore, $\{L_{f_y}\}^k\, y = \binom{1}{1} + \alpha\binom{1}{1} + \cdots + \alpha^{k-1}\binom{1}{1} + \alpha^k\binom{0}{0} = \frac{1-\alpha^k}{1-\alpha}\binom{1}{1} = \left\{1 - \left(\frac{3}{4}\right)^k\right\}\binom{4}{4}$.
Notice that $x = \binom{3}{0} \geq \binom{0}{0} = y$, but $\left(\frac{3}{4}\right)^k \binom{3}{3} \geq \left\{1 - \left(\frac{3}{4}\right)^k\right\}\binom{4}{4}$ is not valid for all k, since for $k \to \infty$, $\left(\frac{3}{4}\right)^k \binom{3}{3} \to \binom{0}{0}$ and $\left\{1 - \left(\frac{3}{4}\right)^k\right\}\binom{4}{4} \to \binom{4}{4}$, i.e. the mapping $\{L_{f_x}\}^k$ is not monotone in general.
Suppose that the operator $\{L_{f_x}\}^k$ is a contraction. Then, $\|\{L_{f_x}\}^k\, x - \{L_{f_y}\}^k\, y\|_\infty \leq \beta \cdot \|x - y\|_\infty$ for some $0 < \beta < 1$ and for all k. Since $\|\{L_{f_x}\}^\infty\, x - \{L_{f_y}\}^\infty\, y\|_\infty = 4 > 3 = \|x - y\|_\infty$, the operator is not a contraction.

Although the operator $\{L_{f_n}\}^{k(n)}$ is neither a contraction nor monotone, it can be shown that $\{L_{f_n}\}^{k(n)}\, x^n$ converges to the value vector v^α for any starting vector x^1. In order to prove this result we need the following lemma.

Lemma 3.16

Let $x^{n+1} := \{L_{f_n}\}^{k(n)}\, x^n$ *and* $\beta(n) := \alpha^{k(n)} \cdot \beta(n-1),\ n \in \mathbb{N},$ *with* $\beta(0) := 1$. *Assume that* $Ux^1 - b \cdot e \leq x^1 \leq Ux^1 + d \cdot e$ *for some* $b, d \geq 0$. *Then, for* $n = 0, 1, \ldots.$

(1) $x^{n+1} \leq Ux^{n+1} + \beta(n) \cdot d \cdot e.$

(2) $x^{n+1} \leq v^\alpha + \frac{\beta(n)}{1-\alpha} \cdot d \cdot e.$

(3) $x^{n+2} \geq Ux^{n+1} - \frac{\alpha\beta(n)}{1-\alpha} \cdot d \cdot e.$

(4) $x^{n+2} \geq v^\alpha - \frac{\alpha^{n+1}}{1-\alpha} \cdot \{(n+1)d + b\} \cdot e.$

Proof

(1) We prove this result by induction on n (for $n = 0$ the result is obvious).

Assume that $x^n \leq Ux^n + \beta(n-1) \cdot d \cdot e$. Since for any fixed $f^\infty \in C(D)$ and any fixed $k \in \mathbb{N}$ the operator $\{L_f\}^k$ is a monotone contraction with factor α^k and with the additional property that $\{L_f\}^k(x + c \cdot e) = \{L_f\}^k\, x + \alpha^k c \cdot e$ for any $x \in \mathbb{R}^N$ and any scalar c, we obtain

$$\begin{aligned}
x^{n+1} &= \{L_{f_n}\}^{k(n)}\, x^n \leq \{L_{f_n}\}^{k(n)}\, \big\{Ux^n + \beta(n-1) \cdot d \cdot e\big\} \\
&= \{L_{f_n}\}^{k(n)}\{Ux^n\} + \alpha^{k(n)} \cdot \beta(n-1) \cdot d \cdot e = \{L_{f_n}\}^{k(n)}\{L_{f_n} x^n\} + \beta(n) d \cdot e \\
&= \{L_{f_n}\}^{k(n)+1}\, x^n + \beta(n) \cdot d \cdot e = \{L_{f_n}\}\{L_{f_n}^{k(n)} x^n\} + \beta(n) \cdot d \cdot e \\
&= L_{f_n}\, x^{n+1} + \beta(n) \cdot d \cdot e \leq Ux^{n+1} + \beta(n) \cdot d \cdot e.
\end{aligned}$$

(2) Iterating the inequality of part (1) gives for any $m \geq 1$

$$\begin{aligned}
x^{n+1} &\leq Ux^{n+1} + \beta(n) \cdot d \cdot e \\
&\leq U\big\{Ux^{n+1} + \beta(n) \cdot d \cdot e\big\} + \beta(n) d \cdot e = U^2 x^{n+1} + \alpha\beta(n) \cdot d \cdot e + \beta(n) \cdot d \cdot e \\
&\leq \cdots \leq U^m x^{n+1} + (1 + \alpha + \cdots + \alpha^{m-1})\beta(n) \cdot d \cdot e.
\end{aligned}$$

Therefore, by letting $m \to \infty$, we obtain $x^{n+1} \leq v^\alpha + \frac{\beta(n)}{1-\alpha} \cdot d \cdot e$.

(3) Also by part (1), we obtain

$$\begin{aligned}
x^{n+2} &= \{L_{f_{n+1}}\}^{k(n+1)}\, x^{n+1} = \{L_{f_{n+1}}\}^{k(n+1)-1}\{L_{f_{n+1}}\, x^{n+1}\} = \{L_{f_{n+1}}\}^{k(n+1)-1}\{Ux^{n+1}\} \\
&\geq \{L_{f_{n+1}}\}^{k(n+1)-1}\{x^{n+1} - \beta(n) \cdot d \cdot e\} = \{L_{f_{n+1}}\}^{k(n+1)-1}\, x^{n+1} - \alpha^{k(n+1)-1}\beta(n) \cdot d \cdot e.
\end{aligned}$$

Iterating the inequality $\{L_{f_{n+1}}\}^{k(n+1)}\, x^{n+1} \geq \{L_{f_{n+1}}\}^{k(n+1)-1}\, x^{n+1} - \alpha^{k(n+1)-1}\beta(n) \cdot d \cdot e$, gives

$$\begin{aligned}
x^{n+2} &= \{L_{f_{n+1}}\}^{k(n+1)}\, x^{n+1} \geq L_{f_{n+1}} x^{n+1} - \{\alpha + \alpha^2 + \cdots + \alpha^{k(n+1)-1}\}\beta(n) \cdot d \cdot e \\
&\geq L_{f_{n+1}} x^{n+1} - \tfrac{\alpha\beta(n)}{1-\alpha} \cdot d \cdot e.
\end{aligned}$$

(4) Since $Ux^1 \leq x^1 + b \cdot e$, it follows that $U^{n+2}x^1 \leq U^{n+1}x^1 + \alpha^{n+1} b \cdot e,\ n = -1, 0, \ldots.$

Hence, by iterating, $U^m\{U^{n+2}x^1\} \leq U^{n+1}x^1 + (1 + \alpha + \cdots + \alpha^m)\alpha^{n+1} b \cdot e,\ m = 0, 1, \ldots.$

Therefore, $v^\alpha = lim_{m\to\infty} U^m\{U^{n+2}x^1\} \leq U^{n+1}x^1 + \frac{\alpha^{n+1}}{1-\alpha} b \cdot e.$

For part (4), it is sufficient to show that $U^{n+1}x^1 \leq x^{n+2} + \frac{(n+1)\alpha^{n+1}}{1-\alpha} \cdot d \cdot e.$

From part (3) it follows that for $j = 0, 1, \ldots$ we have

$$\begin{aligned}
U^{n+1-j}x^{j+1} &= U^{n-j}\{Ux^{j+1}\} \leq U^{n-j}\{x^{j+2} + \tfrac{\alpha\beta(j)}{1-\alpha} \cdot d \cdot e\} \\
&= U^{n-j}x^{j+2} + \tfrac{\alpha^{n+1-j}\beta(j)}{1-\alpha} \cdot d \cdot e.
\end{aligned}$$

Summing up the above inequality over $j = 0, 1, \ldots, n$ gives

$$U^{n+1}x^1 \leq x^{n+2} + (1-\alpha)^{-1}\big\{ \textstyle\sum_{j=0}^n \alpha^{n+1-j}\beta(j)\big\} d \cdot e.$$

Since $\beta(j) = \alpha^{k(j)+k(j-1)+\cdots+k(1)} \leq \alpha^j$ for $j = 0, 1, \ldots, n$, the above inequality implies that

$$U^{n+1}x^1 \leq x^{n+2} + \tfrac{(n+1)\alpha^{n+1}}{1-\alpha} \cdot d \cdot e.$$

□

Theorem 3.30
Let $x^{n+1} = \{L_{f_n}\}^{k(n)}\, x^n,\ = 1, 2, \dots.$ *Then,* $v^\alpha = lim_{n\to\infty}\, x^n$.

Proof
We apply Lemma 3.16 with $b = d = \|Ux^1 - x^1\|_\infty$. Since $lim_{n\to\infty}\, \beta(n) = 0$ and $lim_{n\to\infty}\, n\alpha^n = 0$, we obtain

$$\begin{array}{rcl} limsup_{n\to\infty}\, x^n & \leq & limsup_{n\to\infty}\, \{v^\alpha + \frac{\beta(n-1)}{1-\alpha}\cdot d \cdot e\} \\ & = & v^\alpha \\ & = & lim_{n\to\infty}\, \big\{v^\alpha - \alpha^{n-1}(1-\alpha)^{-1}\{(n-1)d + b\}\cdot e\big\} \\ & = & liminf_{n\to\infty}\, \big\{v^\alpha - \alpha^{n-1}(1-\alpha)^{-1}\{(n-1)d + b\}\cdot e\big\} \\ & \leq & liminf_{n\to\infty}\, x^n, \end{array}$$

i.e. $v^\alpha = lim_{n\to\infty}\, x^n$. □

Theorem 3.31
Algorithm 3.11 terminates in a finite number of iterations with an ε*-optimal policy.*

Proof
Since v^α is the fixed-point of U, we have

$$\begin{array}{rcl} \|Ux^n - x^n\|_\infty & \leq & \|Ux^n - Uv^\alpha\|_\infty + \|Uv^\alpha - x^n\|_\infty = \|Ux^n - Uv^\alpha\|_\infty + \|v^\alpha - x^n\|_\infty \\ & \leq & \alpha\cdot\|x^n - v^\alpha\|_\infty + \|v^\alpha - x^n\|_\infty = (1+\alpha)\cdot\|v^\alpha - x^n\|_\infty. \end{array}$$

Because $v^\alpha = lim_{n\to\infty}\, x^n$, the stop criterion of step 2c in Algorithm 3.11 is satisfied after a finite number of iterations. From Theorem 3.7 part (3) and the stop criterion of Algorithm 3.11 it follows that

$$\|v^\alpha - v^\alpha(f_x^\infty)\|_\infty \leq 2\alpha(1-\alpha)^{-1}\|Ux - x\|_\infty \leq \varepsilon,$$

i.e. Algorithm 3.11 terminates with an ε-optimal policy. □

Convergence rate
We may assume that $Ux^1 \geq x^1$, because for $x^1 := (1-\alpha)^{-1}min_i\{max_a\, r_i(a)\}\cdot e$ this property is satisfied, namely:

$$\begin{array}{rcl} \{Ux^1\}_i & = & max_a\{r_i(a) + \alpha\sum_j\, p_{ij}(a)x_j^1\} = max_a\, r_i(a) + \alpha(1-\alpha)^{-1}min_i\{max_a\, r_i(a)\} \\ & \geq & min_i\{max_a\, r_i(a)\} + \alpha(1-\alpha)^{-1}min_i\{max_a\, r_i(a)\} \\ & = & (1-\alpha)^{-1}min_i\{max_a\, r_i(a)\} = x_i^1,\ i\in S. \end{array}$$

We will show that the convergence of x^n to v^α is at least linear, i.e. for some $0 < c < 1$,

$$\|v^\alpha - x^{n+1}\|_\infty \leq c\cdot\|v^\alpha - x^n\|_\infty \text{ for } n = 0, 1, \dots.$$

Since the operator of the modified policy is neither a contraction nor is it monotone, we cannot rely on general theorems. Therefore, we present a special treatment for the proof of this property. Consider the related operator $U^{(k)} : \mathbb{R}^N \to \mathbb{R}^N$, defined by

$$U^{(k)}\, x := max_f\, L_f^k\, x. \tag{3.54}$$

Theorem 3.32
$U^{(k)}$ *is a monotone contraction with contraction factor* α^k *and with fixed-point* v^α.

Proof
Suppose that $x \geq y$. From the monotonicity of $L^k_f,\ f \in C(D)$, we obtain

$$U^{(k)}\, x = max_f\, L^k_f\, x \geq max_f\, L^k_f\, y = U^{(k)}\, y.$$

Consider a fixed state $i \in S$ and let $f_{x,i}$ be such that $\{U^{(k)}\, x\}_i = \{L^k_{f_{x,i}}\, x\}_i,\ x \in \mathbb{R}^N$. Then, for each $i \in S$, we have

$$\{U^{(k)}\, x - U^{(k)}\, y\}_i \leq \{L^k_{f_{x,i}}\, x - L^k_{f_{x,i}}\, y\}_i = \alpha^k\{P^k(f_{x,i})(x-y)\}_i \leq \alpha^k \cdot \|x-y\|_\infty$$

and

$$\{U^{(k)}\, y - U^{(k)}\, x\}_i \leq \{L^k_{f_{y,i}}\, y - L^k_{f_{y,i}}\, x\}_i = \alpha^k\{P^k(f_{y,i})(x-y)\}_i \leq \alpha^k \cdot \|x-y\|_\infty.$$

Hence, $\|U^{(k)}\, x - U^{(k)}\, y\|_\infty \leq \alpha^k \cdot \|x-y\|_\infty$, i.e. $U^{(k)}$ is a monotone contraction with contraction factor α^k. Finally, we show that v^α is the fixed-point. Let $f_* \in C(D)$ be an α-optimal policy. Since $v^\alpha = L^k_{f_*}\, v^\alpha \geq L^k_f\, v^\alpha$ for every $f \in C(D)$, we obtain

$$v^\alpha \geq max_f\, L^k_f\, v^\alpha = U^{(k)}\, v^\alpha \geq L^k_{f_*}\, v^\alpha = v^\alpha,$$

i.e. v^α is the fixed-point of $U^{(k)}$. □

Consider, besides the sequence $\{x^n\}_{n=1}^\infty$ defined by (3.53), the sequence $\{y^n\}_{n=1}^\infty$ and $\{z^n\}_{n=1}^\infty$, defined by

$$y^1 := z^1 := x^1;\ y^{n+1} := Uy^n,\ z^{n+1} := U^{(k(n))}z^n,\ n \in \mathbb{N}.$$

Lemma 3.17
Under the assumption $Ux^1 \geq x^1$, we have, $Ux^n \geq x^n$ and $v^\alpha \geq z^n \geq x^n \geq y^n$ for every $n \in \mathbb{N}$.

Proof
We apply induction on n. Since $Ux^1 \geq x^1$, we have $v^\alpha = lim_{n\to\infty}\, U^n x^1 \geq U\, x^1 \geq x^1 = y^1 = z^1$; so, the result is true for $n = 1$. Assume that $Ux^n \geq x^n$ and $v^\alpha \geq z^n \geq x^n \geq y^n$. Then,

$$\begin{aligned} Ux^{n+1} - x^{n+1} &= U\big\{\{L_{f_n}\}^{k(n)}\, x^n\big\} - \{L_{f_n}\}^{k(n)}\, x^n \geq \{L_{f_n}\}^{k(n)+1}\, x^n - \{L_{f_n}\}^{k(n)}\, x^n \\ &= \{L_{f_n}\}^{k(n)}\, \{Ux^n\} - \{L_{f_n}\}^{k(n)}\, x^n = \alpha^{k(n)}\big\{P(f_n)\big\}^{k(n)}\{Ux^n - x^n\} \geq 0. \end{aligned}$$

Furthermore, we have

$$v^\alpha = U^{(k(n))}v^\alpha \geq U^{(k(n))}z^n = z^{n+1} = max_f\, L^{k(n)}_f\, z^n \geq L^{k(n)}_{f_n}\, z^n \geq L^{k(n)}_{f_n}\, x^n = x^{n+1}.$$

Since $x^{n+1} = x^n + A^{(k(n))}\{Ux^n - x^n\} = x^n + \sum_{t=0}^{k(n)-1}\{\alpha P(f_n)\}^t\{Ux^n - x^n\}$, we obtain

$$x^{n+1} = Ux^n + \sum_{t=1}^{k(n)-1}\{\alpha P(f_n)\}^t\{Ux^n - x^n\} \geq Ux^n \geq Uy^n = y^{n+1}.$$

□

The next corollary shows that the convergence is geometric, i.e. $\|v^\alpha - x^{n+1}\|_\infty \leq \alpha\|v^\alpha - x^n\|_\infty$.

Corollary 3.9
Under the assumption $Ux^1 \geq x^1$, we have $\|v^\alpha - x^{n+1}\|_\infty \leq \alpha \cdot \|v^\alpha - x^n\|_\infty$.

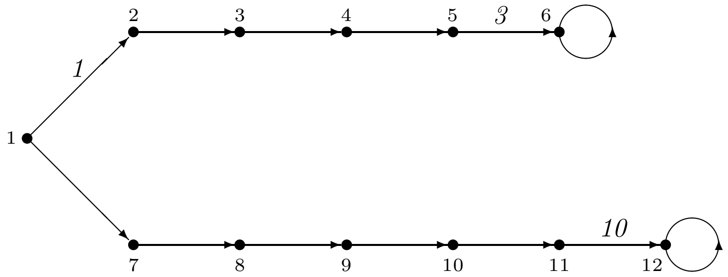
Proof
From the last line of the proof of Lemma 3.17 it follows that $x^{n+1} \geq Ux^n$. Hence, also by Lemma 3.17, we have $0 \leq v^\alpha - x^{n+1} \leq v^\alpha - Ux^n = Uv^\alpha - Ux^n$. Consequently, we obtain

$$\|v^\alpha - x^{n+1}\|_\infty \leq \|Uv^\alpha - Ux^n\|_\infty \leq \alpha \cdot \|v^\alpha - x^n\|_\infty.$$

□

Lemma 3.17 shows that, under the assumption $Ux^1 \geq x^1 = y^1$, the iterates x^n of modified policy iteration always exceed the iterates y^n of value iteration, which is modified policy iteration with $k = 1$. One might conjecture that the iterates of modified policy iteration with fixed order $k + m$ $(m \geq 1)$ always dominates those of modified policy iteration with fixed k. The following example shows that this conjecture is false.

Example 3.6
Let $S = \{1, 2, \dots, 12\}$; $A\{1\} = \{1, 2\}$, $A\{i\} = \{1\}$, $2 \leq i \leq 12$; $r_1(1) = 1$, $r_5(1) = 3$, $r_{11}(1) = 10$ (all other rewards are 0). The transitions are deterministic and from state i to state $i + 1$, $2 \leq i \leq 5$ and $7 \leq i \leq 11$. The states 6 and 12 are absorbing. In state 1, action 1 gives a transition to state 2 and action 2 to state 7. Let $\frac{1}{81} < \alpha < 1$. Below is a picture of this model.



Firstly, consider the modified policy iteration method with $k = 3$ and starting vector $x^1_i = 0$ for $1 \leq i \leq 12$.
$x^2 = (1, 0, 3\alpha^2, 3\alpha, 3, 0, 0, 0, 10\alpha^2, 10\alpha, 10, 0)$.
$x^3 = (1 + 10\alpha^4, 3\alpha^3, 3\alpha^2, 3\alpha, 3, 0, 10\alpha^4, 10\alpha^3, 10\alpha^2, 10\alpha, 10, 0)$.
Secondly, take $k = 4$ and obtain (call the iterates $\overline{x}^2$ and $\overline{x}^3$):
$\overline{x}^2 = (1, 3\alpha^3, 3\alpha^2, 3\alpha, 3, 0, 0, 10\alpha^3, 10\alpha^2, 10\alpha, 10, 0)$.
$\overline{x}^3 = (3\alpha^4, 3\alpha^3, 3\alpha^2, 3\alpha, 3, 0, 10\alpha^4, 10\alpha^3, 10\alpha^2, 10\alpha, 10, 0)$.
Notice that $\overline{x}^2 > x^2$ and $x^3 > \overline{x}^3$.

Exclusion of suboptimal actions

In order to exclude suboptimal actions we need bounds on the value vector v^α. The next theorem provides appropriate bounds.

Theorem 3.33

$$x^n + (1-\alpha)^{-1} min_i \, (Ux^n - x^n)_i \cdot e \leq Ux^n + \alpha(1-\alpha)^{-1} min_i \, (Ux^n - x^n)_i \cdot e \leq$$
$$x^{n+1} + \alpha^{k(n)}(1-\alpha)^{-1} min_i \, (Ux^n - x^n)_i \cdot e \leq v^\alpha \leq x^n + (1-\alpha)^{-1} max_i \, (Ux^n - x^n)_i \cdot e.$$

Proof
We start with the upper bound. Let $f^\infty = f^\infty_{v^\alpha}$. Then, we have

$$\begin{aligned} Ux^n - x^n &\geq L_f\, x^n - x^n = r(f) + \alpha P(f)x^n - x^n = r(f) + \alpha P(f)v^\alpha + \alpha P(f)(x^n - v^\alpha) - x^n \\ &= L_f\, v^\alpha + \alpha P(f)(x^n - v^\alpha) - x^n = (v^\alpha - x^n) + \alpha P(f)(x^n - v^\alpha) \\ &= \{I - \alpha P(f)\}(v^\alpha - x^n). \end{aligned}$$

Hence,

$$\begin{aligned} v^\alpha - x^n &\leq \{I - \alpha P(f)\}^{-1}\{Ux^n - x^n\} \leq \{I - \alpha P(f)\}^{-1} max_i \, (Ux^n - x^n)_i \cdot e \\ &= (1-\alpha)^{-1} max_i \, (Ux^n - x^n)_i \cdot e. \end{aligned}$$

For the lower bounds, we can write

$$\begin{aligned}
v^\alpha - x^n &= Uv^\alpha - x^n \geq L_{f_n} v^\alpha - x^n = r(f_n) + \alpha P(f_n) v^\alpha - x^n \\
&= r(f_n) + \alpha P(f_n) x^n - x^n + \alpha P(f_n)(v^\alpha - x^n) \\
&= L_{f_n} x^n - x^n + \alpha P(f_n)(v^\alpha - x^n),
\end{aligned}$$

implying $\{I - \alpha P(f_n)\}(v^\alpha - x^n) \geq L_{f_n} x^n - x^n$. Therefore,

$$\begin{aligned}
v^\alpha - x^n &\geq \{I - \alpha P(f_n)\}^{-1}\{L_{f_n} x^n - x^n\} = \{I - \alpha P(f_n)\}^{-1}\{U x^n - x^n\} \\
&= \textstyle\sum_{t=0}^\infty \{\alpha P(f_n)\}^t (Ux^n - x^n).
\end{aligned}$$

Since

$$x^{n+1} = x^n + \textstyle\sum_{t=0}^{k(n)-1} \{\alpha P(f_n)\}^t (Ux^n - x^n) = Ux^n + \sum_{t=1}^{k(n)-1} \{\alpha P(f_n)\}^t (Ux^n - x^n),$$

we obtain

$$\begin{aligned}
v^\alpha &\geq x^n + \textstyle\sum_{t=0}^{k(n)-1} \{\alpha P(f_n)\}^t (Ux^n - x^n) + \sum_{t=k(n)}^\infty \{\alpha P(f_n)\}^t (Ux^n - x^n) \\
&= x^{n+1} + \textstyle\sum_{t=k(n)}^\infty \{\alpha P(f_n)\}^t (Ux^n - x^n) \\
&\geq x^{n+1} + \alpha^{k(n)} (1-\alpha)^{-1} min_i \, (Ux^n - x^n)_i \cdot e \\
&= Ux^n + \textstyle\sum_{t=1}^{k(n)-1} \{\alpha P(f_n)\}^t (Ux^n - x^n) + \alpha^{k(n)} (1-\alpha)^{-1} min_i \, (Ux^n - x^n)_i \cdot e \\
&\geq Ux^n + \textstyle\sum_{t=1}^{k(n)-1} \alpha^t \, min_i \, (Ux^n - x^n)_i \cdot e + \alpha^{k(n)} (1-\alpha)^{-1} min_i \, (Ux^n - x^n)_i \cdot e \\
&= Ux^n + \alpha \cdot \{1 - \alpha^{k(n)-1}\}(1-\alpha)^{-1} min_i \, (Ux^n - x^n)_i \cdot e + \alpha^{k(n)} (1-\alpha)^{-1} min_i \, (Ux^n - x^n)_i \cdot e \\
&= Ux^n + \alpha (1-\alpha)^{-1} min_i \, (Ux^n - x^n)_i \cdot e \\
&= x^n + (Ux^n - x^n) + \alpha (1-\alpha)^{-1} min_i \, (Ux^n - x^n)_i \cdot e \\
&\geq x^n + min_i \, (Ux^n - x^n)_i \cdot e + \alpha (1-\alpha)^{-1} min_i \, (Ux^n - x^n)_i \cdot e \\
&= x^n + (1-\alpha)^{-1} min_i \, (Ux^n - x^n)_i \cdot e.
\end{aligned}$$

□

Theorem 3.34
If

$$r_i(a) + \alpha \sum_j p_{ij}(a) x_j^n < x_i^{n+1} + \alpha^{k(n)} (1-\alpha)^{-1} \, min_k \, (Ux^n - x^n)_k - \alpha (1-\alpha)^{-1} \, max_k \, (Ux^n - x^n)_k \quad (3.55)$$

then action a is suboptimal.

Proof

$$\begin{aligned}
r_i(a) + \alpha \textstyle\sum_j p_{ij}(a) v_j^\alpha &\leq r_i(a) + \alpha \textstyle\sum_j p_{ij}(a) \{x_j^n + (1-\alpha)^{-1} \, max_k \, (Ux^n - x^n)_k\} \\
&= r_i(a) + \alpha \textstyle\sum_j p_{ij}(a) x_j^n + \alpha (1-\alpha)^{-1} \, max_k \, (Ux^n - x^n)_k \\
&< x_i^{n+1} + \alpha^{k(n)} (1-\alpha)^{-1} \, min_k \, (Ux^n - x^n)_k \leq v_i^\alpha.
\end{aligned}$$

□

Algorithm 3.12 *Modified policy iteration with exclusion of suboptimal actions*
Input: Instance of a discounted MDP and some scalar $\varepsilon > 0$.
Output: An ε-optimal deterministic policy f^∞.

1. Select $x \in \mathbb{R}^N$.
2. a. Choose any k with $1 \leq k \leq \infty$.
   b. Compute $y_i(a) := r_i(a) + \alpha \sum_j p_{ij}(a) x_j, \ (i,a) \in S \times A$.
   c. Determine f such that $L_f \, x = Ux$.
   d. **if** $\|Ux - x\|_\infty \leq \frac{1}{2}(1-\alpha)\alpha^{-1}\varepsilon$ **then** f^∞ is an ε-optimal policy (STOP).
   e. $max := max_k \, (Ux - x)_k$ and $min := min_k \, (Ux - x)_k$.

3. a. $y := \{L_f\}^k x$.

   b. $A(i) := \{a \mid y_i(a) \geq y_i + \alpha^k(1-\alpha)^{-1}\,min - \alpha(1-\alpha)^{-1}\,max\},\ i \in S$.

   c. **if** $|Ai)| = 1$ **for all** $i \in S$ **then** f^∞ is an optimal policy (STOP).

   d. $x := y$ and **return to** step 2.

Example 3.4 (continued)

Iteration 1

$y_1(1) = \frac{17}{3},\ y_1(2) = 6,\ y_1(3) = \frac{28}{3};\ y_2(1) = \frac{32}{3},\ y_2(2) = 8,\ y_2(3) = \frac{34}{3};$
$y_3(1) = \frac{38}{3},\ y_3(2) = 13,\ y_3(3) = \frac{40}{3}.\ Ux = (\frac{28}{3}, \frac{34}{3}, \frac{40}{3});\ f(1) = f(2) = f(3) = 3.$
$max = 4,\ min = 0;\ y = (\frac{29}{3}, \frac{35}{3}, \frac{41}{3});\ A\{1\} = \{3\},\ A\{2\} = \{1,3\},\ A\{3\} = \{1,2,3\}.$
$x = (\frac{29}{3}, \frac{35}{3}, \frac{41}{3}).$

Iteration 2

$y_1(3) = 9.833;\ y_2(1) = 10.833,\ y_2(3) = 11.833;\ y_3(1) = 12.833,\ y_3(2) = 14.633,\ y_3(3) = 13.833.$
$Ux = (9.833, 11.833, 14.833);\ f(1) = f(2) = 3,\ f(3) = 2;\ max = 1.166,\ min = 0.166.$
$y = (10.417, 12.417, 14.917).\ A\{1\} = \{3\},\ A\{2\} = \{3\},\ A\{3\} = \{2,3\}.$
$x = (10.417, 12.417, 14.917).$

Iteration 3

$y_1(3) = 10.459;\ y_2(3) = 12.459;\ y_3(2) = 15.209,\ y_3(3) = 14.459;\ Ux = (10.459, 12.459, 15.209).$
$f(1) = f(2) = 3,\ f(3) = 2;\ max = 0.292,\ min = 0.042;\ y = (10.604, 12.604, 15.229).$
$A\{1\} = \{3\},\ A\{2\} = \{3\},\ A\{3\} = \{2\};\ f^\infty$ is an optimal policy.

3.9 Monotone optimal policies

In this section we study under which conditions optimal policies are monotone for discounted MDPs. A policy $f^\infty \in C(D)$ is a *monotone policy* if either $f(i+1) \geq f(i)$ for $1, 2, \dots, N-1$ (*nondecreasing*) or $f(i+1) \leq f(i)$ for $1, 2, \dots, N-1$ (*nonincreasing*). The analysis is similar to section 2.4 in which an MDP with finite horizon is studied. Also in the present section we assume that $A(i) = A = \{1, 2, \dots, M\},\ i \in S$, and that S and A be ordered in the usual way. On $S \times A$ we consider the functions r and $p(k)$ with components $r_i(a)$ and $\sum_{j=k}^N p_{ij}(a)$, respectively. We show the existence of optimal monotone policies under the following assumptions:

Assumption 3.1

(A1) $r_i(a)$ is nondecreasing in i for all a;

(A2) $\sum_{j=k}^N p_{ij}(a)$ is nondecreasing in i for all k and a.

(A3) $r_i(a)$ is supermodular on $S \times A$;

(A4) $\sum_{j=k}^N p_{ij}(a)$ is supermodular on $S \times A$ for all k.

The proof of the structure of the optimal policy is based on the value iteration scheme $v^0 = 0,\ v^n = Uv^{n-1}$, $n = 1, 2, \dots$ for which we know $\lim_{n\to\infty} v^n = v^\alpha$.

Lemma 3.18

Under the assumptions A1 and A2, v_i^n is nondecreasing in i for $n = 0, 1, \dots$.

Proof

Apply induction on n. For $n = 0$ we have $v_i^n = 0$ for all $i \in S$, so the result holds. Assume that the result holds for n. Since, by assumption $A2$, $\sum_{j=k}^N p_{i+1,j}(a) \geq \sum_{j=i}^N p_{ij}(a)$ for all k and a, and because $\sum_{j=1}^N p_{i+1,j}(a) = \sum_{j=1}^N p_{ij}(a) = 1$, we obtain from Lemma 2.2

$\sum_{j=1}^N p_{i+1,j}(a)v_j^n \geq \sum_{j=1}^N p_{i,j}(a)v_j^n$ for all $i = 1, 2, \dots, N-1$ and all $a = 1, 2, \dots, M$.

By assumption $A1$, we get

$r_{i+1}(a) + \alpha \cdot \sum_{j=1}^N p_{i+1,j}(a)v_j^n \geq r_i(a) + \alpha \cdot \sum_{j=1}^N p_{i,j}(a)v_j^n$

for all $i = 1, 2, \dots, N-1$ and all $a = 1, 2, \dots, M$.

Therefore,

$v_{i+1}^{n+1} = max_a \{r_{i+1}(a) + \alpha \cdot \sum_{j=1}^N p_{i+1,j}(a)v_j^n\} \geq max_a \{r_i(a) + \alpha \cdot \sum_{j=1}^N p_{i,j}(a)v_j^n\} = v_i^{n+1}$

for all $i = 1, 2, \dots, N-1$, i.e. v_i^{n+1} is nondecreasing in i. □

Corollary 3.10

Under the assumption A1 and A2, v_i^α is nondecreasing in i.

Proof

The result follows directly from Lemma 3.18 and the property $v_i^\alpha = \lim_{n\to\infty} v_i^n,\ i \in S$.

Theorem 3.35

Under the assumptions A1, A2, A3 and A4, there exists an optimal policy $f^\infty \in C(D)$, where $f(i)$ is nondecreasing in i.

Proof

We first prove that $s_i(a) := r_i(a) + \alpha \cdot \sum_{j=1}^N p_{i,j}(a)v_j^\alpha$ is supermodular on $S \times A$. Let $i_1 \geq i_2$ and $a_1 \geq a_2$. Define $y_j := p_{i_1 j}(a_1) + p_{i_2 j}(a_2)$ and $z_j := p_{i_1 j}(a_2) + p_{i_2 j}(a_1)$ for all $j \in S$. By Assumption $A4$, for all $1 \leq k \leq N$, we have $\sum_{j=k}^N y_j \geq \sum_{j=k}^N z_j$. Since $\sum_{j=1}^N y_j = \sum_{j=1}^N z_j = 2$, and because v_i^α is nondecreasing in i (see Corollary 3.10), applying Lemma 2.2 yields

$$\sum_{j=1}^N \{p_{i_1 j}(a_1) + p_{i_2 j}(a_2)\}v_j^\alpha \geq \sum_{j=1}^N \{p_{i_1 j}(a_2) + p_{i_2 j}(a_1)\}v_j^\alpha,$$

i.e. $\sum_{j=1}^N p_{ij}(a)v_j^\alpha$ is supermodular. Because the sum of supermodular functions is supermodular, $s_i(a)$ is also supermodular on $S \times A$. If an optimal action $f(i)$ in state i is not unique, take the largest optimal action. Then, applying Lemma 2.3 yields the result that $f(i)$ is nondecreasing in i. □

Next, we will present other assumptions under which an optimal nondecreasing policy exists. These assumptions are stated below.

Assumption 3.2

(B1) $r_i(a)$ is nonincreasing in i for all a;

(B2) $\sum_{j=k}^N p_{ij}(a)$ is nondecreasing in i for all k and a.

(B3) $r_i(a)$ is supermodular on $S \times A$;

(B4) $\sum_{j=k}^N p_{ij}(a)$ is submodular on $S \times A$ for all k.

Note
$B1$ and $B4$ are the 'reverse' versions of $A1$ and $A4$, and $B2$ and $B3$ are the same conditions as $A2$ and $A3$.

Lemma 3.19
Under the assumptions B1 and B2, v_i^n, for $n = 0, 1, \dots,$ and v_i^α are nonincreasing in i.

Proof We first prove v_i^n is nondecreasing in i for $n = 0, 1, \dots$. Apply induction on n. For $n = 0$ we have $v_i^n = 0$ for all $i \in S$, so the result holds. Assume that the result holds for n. Since, by assumption $B2$, $\sum_{j=k}^N p_{i+1,j}(a) \geq \sum_{j=i}^N p_{ij}(a)$ for all k and a, and because $\sum_{j=1}^N p_{i+1,j}(a) = \sum_{j=1}^N p_{ij}(a) = 1$, we obtain from Lemma 2.2, with $v = -v^n$,

$\sum_{j=1}^N p_{i+1,j}(a)v_j^n \leq \sum_{j=1}^N p_{i,j}(a)v_j^n$ for all $i = 1, 2, \dots, N-1$ and all $a = 1, 2, \dots, M$.

By assumption $B1$, we get

$r_{i+1}(a) + \alpha \cdot \sum_{j=1}^N p_{i+1,j}(a)v_j^n \leq r_i(a) + \alpha \cdot \sum_{j=1}^N p_{i,j}(a)v_j^n$ for all $i = 1, 2, \dots, N-1$ and all $a = 1, 2, \dots, M$.

Therefore,

$v_{i+1}^{n+1} = max_a \, \{r_{i+1}(a) + \alpha \cdot \sum_{j=1}^N p_{i+1,j}(a)v_j^n\} \leq max_a \, \{r_i(a) + \alpha \cdot \sum_{j=1}^N p_{i,j}(a)v_j^n\} = v_i^{n+1}$

for all $i = 1, 2, \dots, N-1$, i.e. v_i^{n+1} is nondecreasing in i.

Since $v_i^\alpha = \lim_{n\to\infty} v_i^n$ for all $i \in S$, v_i^α is also nonincreasing in i. □

Theorem 3.36
Under the assumptions B1, B2, B3 and B4, there exists an optimal policy $f^\infty \in C(D)$, where $f(i)$ is nondecreasing in i.

Proof
We first prove that $s_i(a) := r_i(a) + \alpha \cdot \sum_{j=1}^N p_{i,j}(a)v_j^\alpha$ is supermodular on $S \times A$. Let $i_1 \geq i_2$ and $a_1 \geq a_2$. Define $y_j := p_{i_1j}(a_2) + p_{i_2j}(a_1)$ and $z_j := p_{i_1j}(a_1) + p_{i_2j}(a_2)$ for all $j \in S$. By Assumption $B4$, for all $1 \leq k \leq N$, we have $\sum_{j=k}^N y_j \geq \sum_{j=k}^N z_j$. Since $\sum_{j=1}^N y_j = \sum_{j=1}^N z_j = 2$, and because $-v_i^\alpha$ is nondecreasing in i (see Lemma 3.19), applying Lemma 2.2 yields

$$\sum_{j=1}^N \{p_{i_1j}(a_2) + p_{i_2j}(a_1)\}(-v_j^\alpha) \geq \sum_{j=1}^N \{p_{i_1j}(a_1) + p_{i_2j}(a_2)\}(-v_j^\alpha),$$

i.e.

$$\sum_{j=1}^N \{p_{i_1j}(a_1) + p_{i_2j}(a_2)\}v_j^\alpha \geq \sum_{j=1}^N \{p_{i_1j}(a_2) + p_{i_2j}(a_1)\}v_j^\alpha.$$

i.e. $\sum_{j=1}^N p_{ij}(a)v_j^\alpha$ is supermodular. Because the sum of supermodular functions is supermodular, $s_i(a)$ is also supermodular on $S \times A$. If an optimal action $f(i)$ in state i is not unique, take the largest optimal action. Then, applying Lemma 2.3 yields the result that $f(i)$ is nondecreasing in i. □

Example 3.7
Consider a problem that is basically a machine replacement problem with 8 states (state 1 is the state of a new machine) and two actions (action 1 corresponds to continue and action 2 to replace). So, let $S := \{1, 2, \dots, 8\}$, $A := \{1, 2\}$ and let $\alpha := 0.9$. The rewards are:

$r_i(a)$	$r_1(a)$	$r_2(a)$	$r_3(a)$	$r_4(a)$	$r_5(a)$	$r_6(a)$	$r_7(a)$	$r_8(a)$
$a = 1$	0	0	0	0	0	0	-1	-5
$a = 2$	-2	-2	-2	-2	-2	-2	-2	-2

The transition probabilities for action 1 are:

$p_{ij}(1)$	$j=1$	$j=2$	$j=3$	$j=4$	$j=5$	$j=6$	$j=7$	$j=8$
$i=1$	0.03	0.07	0.05	0.1	0.1	0.2	0.2	0.25
$i=2$	0	0.02	0.03	0.1	0.1	0.2	0.2	0.35
$i=3$	0	0	0.05	0.05	0.1	0.1	0.2	0.5
$i=4$	0	0	0	0.05	0.05	0.1	0.2	0.6
$i=5$	0	0	0	0	0.02	0.08	0.1	0.8
$i=6$	0	0	0	0	0	0.05	0.1	0.85
$i=7$	0	0	0	0	0	0	0.1	0.9
$i=8$	0	0	0	0	0	0	0	1

For the second action (replacement), the transition probabilities are: $p_{i1}(2) = 1$ for $i = 1, 2, \dots, 8$. We leave it to the reader to verify that this model satisfies the assumptions $B1, B2, B3$ and $B4$. If we use the linear programming problem (3.32) with $\beta_j = \frac{1}{8}$ for $j = 1, 2, \dots, 8$, this program becomes:

$max\{-x_{71} - 5x_{81} - 2x_{12} - 2x_{22} - 2x_{32} - 2x_{42} - 2x_{52} - 2x_{62} - 2x_{72} - 2x_{82}$

subject to the constraints

$x_{11} + x_{12} = 0.125 + 0.9 \cdot \{0.03x_{11} + x_{12} + x_{22} + x_{32} + x_{42} + x_{52} + x_{62} + x_{72} + x_{82}\};$

$x_{21} + x_{22} = 0.125 + 0.9 \cdot \{0.07x_{11} + 0.02x_{21}\};$

$x_{31} + x_{32} = 0.125 + 0.9 \cdot \{0.05x_{11} + 0.03x_{21} + 0.05x_{31}\};$

$x_{41} + x_{42} = 0.125 + 0.9 \cdot \{0.1_{x11} + 0.1x_{21} + 0.05x_{31} + 0.05x_{41}\};$

$x_{51} + x_{52} = 0.125 + 0.9 \cdot \{0.1x_{11} + 0.1x_{21} + 0.1x_{31} + 0.05x_{41} + 0.02x_{51}\};$

$x_{61} + x_{62} = 0.125 + 0.9 \cdot \{0.2x_{11} + 0.2x_{21} + 0.1x_{31} + 0.1x_{41} + 0.08x_{51} + 0.05x_{61}\};$

$x_{71} + x_{72} = 0.125 + 0.9 \cdot \{0.2x_{11} + 0.2x_{21} + 0.2x_{31} + 0.2x_{41} + 0.1x_{51} + 0.1x_{61} + 0.1x_{71}\};$

$x_{81} + x_{82} = 0.125 + 0.9 \cdot \{0.25x_{11} + 0.35x_{21} + 0.5x_{31} + 0.6x_{41} + 0.8x_{51} + 0.85x_{61} + 0.9x_{71} + x_{81}\};$

$x_{11}, x_{21}, x_{31}, x_{41}, x_{51}, x_{61}, x_{71}, x_{81}, x_{12}, x_{22}, x_{32}, x_{42}, x_{52}, x_{62}, x_{72}, x_{82} \geq 0.$

An optimal solution of this program and the corresponding control-limit policy is presented in the next table (note that more optimal solutions are possible):

i	$x_i(1)$	$x_i(2)$	$f(i)$
$i=1$	3.5700	0	1
$i=2$	0.3563	0	1
$i=3$	0.3092	0	1
$i=4$	0.5155	0	1
$i=5$	0.5391	0	1
$i=6$	0.9893	0	1
$i=7$	0	1.1177	2
$i=8$	0	2.6029	2

We now provide an implementation of a policy iteration algorithm, which finds a monotone optimal policy. We assume that the assumptions $A1, A2, A3$ and $A4$ hold, so that a nondecreasing optimal policy exists. Therefore, the policy space is the set of nondecreasing policies.

Algorithm 3.13 *Determination of a nondecreasing optimal policy f_*^∞ for an MDP either under assumption 3.1 or under assumption 3.2.*
Input: Instance of a discounted MDP which satisfies either assumption 3.1 or assumption 3.2.
Output: A nondecreasing optimal deterministic policy f_*^∞.

```
1. Choose any nondecreasing policy $f_1^\infty \in C(D)$; $n := 1$.
2. Compute $v^n := v^\alpha(f_n^\infty)$ by solving $\{I - \alpha P(f_n)\}x = r(f_n)$.
3. $i := 1$; $A_i := \{1, 2, \dots, M\}$.
   (a) $A_i^* := argmax_{a \in A_i}\{r_i(a) + \alpha \cdot \sum_{j=1}^N p_{ij}(a)v_j^n\}$.
   (b) if $i = N$ then go to step 3d
       else $A_{i+1} := \big\{a \in A_i \mid a \geq max\{a^* \mid a^* \in A_i^*\}\big\}$.
   (c) $i := i + 1$; return to step 3a.
   (d) Choose $f_{n+1}$ such that $f_{n+1}(j) \in A_i^*$ for all $j \in S$, setting $f_{n+1}(j) := f_n(j)$ if possible.
4. if $f_{n+1} = f_n$ then begin $f_*^\infty := f_n$; STOP end
   else begin $n := n + 1$; return to step 2 end
```

The advantage of this algorithm is that the maximization can be carried out over action sets A_i which become smaller in the order of the states.

3.10 Bibliographic notes

The principle of optimality is credited to Bellman ([17]). Discounted models appear to have been first analyzed in generality by Howard ([134]). Blackwell ([29]) and Denardo ([56]) have provided fundamental theoretical papers on discounted models.

The proof of Theorem 3.6 is drawn from Ross ([236]). Shapiro ([266]) made the observation that Brouwer's fixed-point theorem can also be used to prove that the mapping U has a fixed-point (see Exercise 3.7). The concept of conserving policy was proposed by Dubins and Savage ([75]). Bather ([11]) was the first to use the span semi-norm in Markov decision processes. The idea to use bounds for the value vector in order to derive suboptimality tests is due to MacQueen ([188], [189]). These ideas were extended and improved by Grinold ([109]), Porteus ([220], [222]) Hordijk and Kallenberg ([129]) and others.

Policy iteration is usually attributed to Howard ([134]). We followed the more mathematically treatment of Blackwell ([29]). The equivalence between policy iteration and Newton's method was shown in Puterman and Brumelle ([228]). Hastings ([112]; see Exercise 3.12) has proposed a method to accelerate the policy iteration method. Theorem 3.14, Theorem 3.15 and Corollary 3.6 are based on Ng's note ([204]). Hartley, Lavercombe and Thomas ([110]) have evaluated by computational experiments several ideas which have been suggested for the implementation of of policy iteration.

The linear programming method for discounted MDPs was proposed by d'Epenoux ([67]). The equivalence between block-pivoting and policy iteration was mentioned by De Ghellinck ([51]). The one-to-one correspondence between the feasible solutions of the dual program and the set of stationary policies can be found in De Ghellinck and Eppen ([52]). For an extensive study of linear programming and Markov decision models see Kallenberg ([148]). In [281] Stein reports on numerical experience with the linear programming method. It turns out that this method is efficient - in comparison with value iteration, policy iteration and modified policy iteration - for problems with discount factor a close to 1 and for not too large state spaces. Furthermore, the experiments show that action elimination and block-pivoting cannot reduce the computation time of the standard LP method considerably. Bello and Riano ([19]) have built the package JMDP, an object-oriented framework to model and solve discounted and unichained average MDPs in Java. In this package LP-solvers Xpress-MP (see [49]) and QSopt (see [6]) are used.

The use of value iteration originates in the work of Shapley ([267]), who applied it in stochastic games. Hastings ([111], [112]) and Kushner and Kleinman ([173]) independently suggested the use of (pre)-Gauss-Seidel iteration to accelerate value iteration. The variant with *relaxation and one-step look-ahead* is due to Herzberg and Yechiali ([116]). Other accelerations were proposed by Kushner and Kleinman ([174]) and Reetz ([234]), both papers on *overrelaxation*, Wessels ([323]) and Van Nunen and Wessels ([303], [304]), these last papers based on *stopping times*, and by Porteus and Totten ([223], [224]). The method of value iteration and bisection of Section 3.7 is due to Bartmann ([10]).

The method of modified policy iteration was suggested in Morton [201] and formalized by Van Nunen ([301]), and by Puterman and Shin ([229], [230]). The example showing that the operator of this method is in general neither a contraction nor monotone is due to Van Nunen ([302]). The observation that modified policy iteration method can be viewed as an inexact Newton method to solve the optimality equation $Ux = x$ was made by Dembo and Haviv ([55]). The exclusion of suboptimal actions is developed by Puterman and Shin ([230]). Example 3.6 is due to Van der Wal and Van Nunen ([298]).

The development of monotone optimal policies is provided by the work of Serfozo ([263]) and Topkis ([289]). Other contributions are given e.g. by Ross ([239]), White ([328]) and Heyman and Sobel ([117]).

3.11 Exercises

Exercise 3.1
Let $\mathcal{M} = \big\{\mu \in \mathbb{R}^N \mid \mu_i > 0,\ i \in S \text{ and } \sum_j p_{ij}(a)\mu_j \leq \mu_i \text{ for all } (i,a) \in S \times A\big\}$.
Define $\|x\|_\mu = max_i\ \mu_i^{-1} \cdot |x_i|$.

a. Show that $\|x\|_\mu$ is a norm in $\mathbb{R}^N$.

b. Formulate the generalizations of Lemma 3.2 and Lemma 3.3 with respect to the norm $\|x\|_\mu$.

c. Give the key property by which the proofs of these new Lemmata follow from the versions with the supremum norm.

Exercise 3.2
Suppose that B is a monotone contraction with contraction factor β and fixed-point x^*.
Show that for any $n \in \mathbb{N}$ the mapping B^n is a monotone contraction with contraction factor β^n and fixed-point x^*.

Exercise 3.3
Let X be a Banach space and $B : X \to X$.
Suppose that B is *nonexpanding*, i.e. $\|Bx - By\| \leq \|x - y\|$ for every $x, y \in X$, and suppose furthermore that B^n is a contraction mapping with contraction factor β and fixed-point x^*, for some $n \in \mathbb{N}$. Then, show that

(1) x^* is the unique fixed-point of B.

(2) $\|x^* - x\| \leq n(1-\beta)^{-1} \cdot \|Bx - x\|$ for every $x \in X$.

Exercise 3.4
Let $B : \mathbb{R}^N \rightarrow \mathbb{R}^N$ be such that for all $x, y \in \mathbb{R}^N$. Assume that for some $\beta \in [0, 1)$ we have

$$max_i\,(Bx - By)_i \;\le \beta \cdot max_i\,(x - y)_i \text{ and } min_i\,(Bx - By)_i \ge \beta \cdot min_i\,(x - y)_i.$$

Show that:

(1) B is a monotone contraction mapping with contraction factor β.

(2) $x + (1 - \beta)^{-1} \cdot min_i\,(Bx - x)_i \cdot e \le x^* \le x + (1 - \beta)^{-1} \cdot max_i\,(Bx - x)_i \cdot e$, where x^* is the fixed-point of B and x is an arbitrary point of $\mathbb{R}^N$.

Exercise 3.5
Let $R = (\pi^1, \pi^2, \dots)$ be any Markov policy and let $x \in \mathbb{R}^N$. Show that $v^\alpha(R) = lim_{n\to\infty}\, L_{\pi^1} L_{\pi^2} \cdots L_{\pi^n}\, x$.

Exercise 3.6
Give the optimality equation of the following model:

$S = \{1, 2\};\ A(1) = A(2) = \{1, 2\};\ \alpha = 0.9;\ r_1(1) = 4, r_1(2) = 2, r_2(1) = 0, r_2(2) = 2;$
$p_{11}(1) = \frac{1}{3}, p_{12}(1) = \frac{2}{3}, p_{11}(2) = \frac{2}{3}, p_{12}(2) = \frac{1}{3}, p_{21}(1) = 1, p_{22}(1) = p_{21}(2) = 0, p_{22}(2) = 1.$

Exercise 3.7
Brouwer's fixed-point theorem is:
Suppose that G is a continuous function which maps a compact convex set $X \subseteq \mathbb{R}^N$ into itself. Then G has a fixed-point.
Show by Brouwer's theorem that U has a fixed-point.
Hint: Take $X = \{x \in \mathbb{R}^N \mid \|x\|_\infty \le (1 - \alpha)^{-1} \cdot M\}$, where $M := max_{(i,a)}\,|r_i(a)|$.

Exercise 3.8
For any $x \in \mathbb{R}^N$ and $\mu \in \mathcal{M}$, defined in Exercise 3.1, we define

$$b_1 := min_i\,\frac{(Ux-x)_i}{\mu_i};\ \beta_1 := \begin{cases} \alpha \cdot min_{i,a}\,\frac{1}{\mu_i}\sum_j\, p_{ij}(a)\mu_j \text{ if } b_1 > 0 \\ \alpha \cdot max_{i,a}\,\frac{1}{\mu_i}\sum_j\, p_{ij}(a)\mu_j \text{ if } b_1 \le 0 \end{cases}$$

$$b_2 := max_i\,\frac{(Ux-x)_i}{\mu_i};\ \beta_2 := \begin{cases} \alpha \cdot max_{i,a}\,\frac{1}{\mu_i}\sum_j\, p_{ij}(a)\mu_j \text{ if } b_2 > 0 \\ \alpha \cdot min_{i,a}\,\frac{1}{\mu_i}\sum_j\, p_{ij}(a)\mu_j \text{ if } b_2 \le 0 \end{cases}$$

Show that:

(1) $\beta_1 b_1 \cdot \mu_i \le \alpha \cdot b_1 \sum_j\, p_{ij}(a)\mu_j$ and $\beta_2 b_2 \cdot \mu_i \ge \alpha \cdot b_2 \sum_j\, p_{ij}(a)\mu_j$ for every $(i, a) \in S \times A$.

(2) $x + (1 - \beta_1)^{-1} b_1 \cdot \mu \le Ux + \beta_1(1 - \beta_1)^{-1} b_1 \cdot \mu \le v^\alpha(f_x^\infty) \le v^\alpha \le Ux + \beta_2(1 - \beta_2)^{-1} b_2 \cdot \mu \le$
$x + (1 - \beta_2)^{-1} b_2 \cdot \mu.$

(3) If $r_i(a) + \alpha \sum_j\, p_{ij}(a)x_j < (Ux)_i + \beta_1(1 - \beta_1)^{-1} b_1 \cdot \mu_i - \beta_2(1 - \beta_2)^{-1} b_2 \cdot \mu_i$, then action $a \in A(i)$ is suboptimal.

(4) If $r_i(a) + \alpha \sum_j\, p_{ij}(a)(Ux)_j < (Ux)_i + \beta_1(1 - \beta_1)^{-1} b_1 \cdot \mu_i - \beta_2^2(1 - \beta_2)^{-1} b_2 \cdot \mu_i$, then action $a \in A(i)$ is suboptimal.

(5) Test (4) is stronger than test (3).

Exercise 3.9
Show that $span\ (U^2x - Ux) \le \alpha \cdot (Ux - x)$.

Exercise 3.10

Consider the following MDP:

$S = \{1,2\};\ A(1) = A(2) = \{1,2\};\ \alpha = \frac{1}{2};\ r_1(1) = 1,\ r_1(2) = 0;\ r_2(1) = 2,\ r_2(2) = 2.$

$p_{11}(1) = \frac{1}{2},\ p_{12}(1) = \frac{1}{2};\ p_{11}(2) = \frac{1}{4},\ p_{12}(2) = \frac{3}{4}.$

$p_{21}(1) = \frac{2}{3},\ p_{22}(1) = \frac{1}{3};\ p_{21}(2) = \frac{1}{3},\ p_{22}(2) = \frac{2}{3}.$

Use the policy iteration algorithm 3.2 to find an α-discounted optimal policy for this model (start with $f(1) = f(2) = 1$).

Exercise 3.11

Show that $Fy \geq Fx$ implies that $y \leq x$, where F is defined by $Fx = Ux - x$.

Exercise 3.12

Consider the following modification of the policy iteration method:

```
1. Start with any $f \in C(D)$.
2. Compute $v^\alpha(f^\infty)$ as unique solution of the linear system $L_f x = x$.
3. for $i = 1$ to $N$ do
     begin
        $d_{ia}(f) := r_i(a) + \alpha \sum_{j=1}^{i-1} p_{ij}(a)x_j + \alpha \sum_{j=i}^{N} p_{ij}(a)v_j^\alpha(f^\infty),\ a \in A(i)$;
        if $d_{ia}(f) \leq v_i^\alpha(f^\infty)$ for every $a \in A(i)$ then begin $x_i := v_i^\alpha(f^\infty)$; $g(i) := f(i)$ end
        else begin $x_i := max_a\, d_{ia}(f)$; choose $g(i)$ such that $d_{ig(i)} = x_i$ end
     end
4. if $g(i) = f(i)$ for every $i \in S$ then go to step 6
5. $f := g$; return to step 2.
6. $f^\infty$ is an $\alpha$-discounted optimal policy (STOP).
```

Prove the correctness of this method by showing the following steps:

a. (i) $x \geq v^\alpha(f^\infty)$; (ii) $x = v^\alpha(f^\infty)$ if and only if $f = g$.

b. If $f = g$, then $f^\infty)$ is an α-discounted optimal policy.

c. If $f \neq g$, then $v^\alpha(g^\infty) \geq x \geq v^\alpha(f^\infty)$.

Exercise 3.13

Apply the method of Exercise 3.12 to the MDP model of Example 3.1.

Exercise 3.14

Show that for a given initial distribution β and a stationary policy π^∞, $\sum_j \beta_j v_j^\alpha(\pi^\infty) = \sum_{(i,a)} r_i(a) x_i^\pi(a)$.

Exercise 3.15

Use the linear programming method to compute the value vector and an optimal policy for the model of Exercise 3.10 (take $\beta_1 = \beta_2 = \frac{1}{2}$).

Exercise 3.16

Show the following optimality properties:

(1) If $\pi^\infty \in C(S)$ is an α-discounted optimal policy, then x^π is an optimal solution of (3.32).

(2) If x is an optimal solution of (3.32), then $\left(\pi^x\right)^\infty$ is an α-discounted optimal policy.

Exercise 3.17
Apply the suboptimality tests of the Theorems 3.20 and 3.22 to the model of Exercise 3.10 (take $\beta_1 = \beta_2 = \frac{1}{2}$).

Exercise 3.18
Use algorithm 3.4 to compute an ε-optimal policy for the model of Exercise 3.10. Take $\varepsilon = 0.2$ and start with $x = (2, 2)$.

Exercise 3.19
Use algorithm 3.5 to compute an ε-optimal policy for the model of Exercise 3.10. Take $\varepsilon = 0.2$ and start with $x = (2, 2)$.

Exercise 3.20
Use algorithm 3.6 to compute an ε-optimal policy for the model of Exercise 3.10. Take $\varepsilon = 0.2$ and start with $x = (2, 2)$.

Exercise 3.21
Use algorithm 3.7 to compute an ε-optimal policy for the model of Exercise 3.10. Take $\varepsilon = 0.2$ and start with $x = (2, 2)$.

Exercise 3.22
Prove Theorem 3.28

Exercise 3.23
Prove Theorem 3.29

Exercise 3.24
Use algorithm 3.11 to compute an ε-optimal policy for the model of Exercise 3.10. Take $\varepsilon = 0.2$, start with $x = (2, 2)$ and choose $k = 2$ in each iteration.

Exercise 3.25
Show that $\|v^\alpha - x^{n+1}\|_\infty \leq \beta \cdot \|v^\alpha - x^n\|_\infty$, where x^n is the x in iteration n of algorithm 3.11 and $\beta := min\{\alpha, \alpha^{k(n)} + (1-\alpha)^{-1}\big(\alpha - \alpha^{k(n)}\big)\|P(f_n) - P(f_\alpha)\|_\infty\}$, where $f_n := f_{x^n}$ and $f_\alpha := f_{v^\alpha}$, respectively. Assume that $Ux^1 \geq x^1$.

Exercise 3.26
Consider the following modified policy algorithm.

Algorithm 3.14
Input: Instance of a discounted MDP and some scalar $\varepsilon > 0$.
Output: An ε-optimal deterministic policy f^∞ and a $\frac{1}{2}\varepsilon$-approximation of the value vector v^α.

```
1. Select $x \in \mathbb{R}^N$ arbitrary; $\overline{y} := \infty$.
2. a. Choose any $k$ with $1 \leq k \leq \infty$.
   b. Determine $f$ such that $L_f\, x = Ux$.
   c. Let $min := min_i\, (Ux - x)_i$ and $max := max_i\, (Ux - x)_i$.
   d. $\underline{y} := x + (1-\alpha)^{-1}\, min \cdot e$; $\overline{y} = x + (1-\alpha)^{-1}\, max \cdot e$.
   c. if $\|\overline{y} - \underline{y}\|_\infty \leq \varepsilon$ then begin $y := \frac{1}{2}(\overline{y} - \underline{y})$; go to step 3 end
      else begin $y := \{L_f\}^k\, x$; $x := y$: return to step 2 end
```

3. y is a $\frac{1}{2}\varepsilon$-approximation of v^α and f^∞ is an ε-optimal policy (STOP).

(1) Apply this algorithm to the MDP model of exercise 3.24.

(2) Show, under the assumption $Ux^1 \geq x^1$, the following properties for this algorithm ($x^n, \ f_n, \ \underline{y}^n, \ \overline{y}^n$ are the values of $x, \ f, \ \underline{y}, \ \overline{y}$, respectively, in iteration n):

a. $x^n \leq Ux^n \leq x^{n+1} \leq v^\alpha(f_n^\infty)$.

b. $\underline{y}^n \leq v^\alpha(f_n^\infty) \leq v^\alpha \leq \overline{y}^n$.

c. $\underline{y}^n \uparrow v^\alpha$ and $\overline{y}^n \downarrow v^\alpha$.

d. $\|v^\alpha - y\|_\infty \leq \frac{1}{2}\varepsilon$ and $\|v^\alpha - v^\alpha(f^\infty)\|_\infty \leq \varepsilon$, when the algorithm terminates.

Exercise 3.27
Let x^n be the value of x in iteration n of Algorithm 3.11. Show that if

$$r_i(a) + \alpha \sum_j p_{ij}(a)x_j^n < x_i^n + (1-\alpha)^{-1} min_k\,(Ux^n - x^n)_k - \alpha(1-\alpha)^{-1}\, max_k(Ux^n - x^n)_k,$$

then action a is suboptimal.

Chapter 4

Total reward

4.1 Introduction

.

Alternatives to the expected total discounted reward criterion in infinite-horizon models include the total expected reward and the average expected reward criteria. This chapter deals with the total expected reward criterion. We have to make some assumptions on the rewards and/or the transition probabilities, without which the total expected reward may be unbounded or not even well defined. When these assumptions are not fulfilled, the average reward and more sensitive optimality criteria can be applied. These last models will be discussed in the chapters 5, 6 and 7.

We will generalize the concept of transition probability to the concept of *transition rate*. The numbers $p_{ij}(a)$ are required only to be nonnegative. Given that at time $t = 1$ the system is observed in state i with 'quantity' 1, we define for any policy R by $p^t_{ij}(R)$ and $p^t_{ij}(a, R)$ the expectation of the 'quantity' in state j at time t, and the expectation of the 'quantity' in state j at time t in conjunction with the probability that action a is chosen at time t, respectively.

Hence, the total expected reward in the first T periods, given initial state i and the use of policy R, is given by

$$v_i^T(R) := \sum_{t=1}^{T} \sum_{j,a} p_{ij}^t(a,R) \cdot r_j(a),\ i \in S. \tag{4.1}$$

We distinguish between the following models:

An MDP is called *stochastic* if $\sum_j p_{ij}(a) = 1$ for all $(i,a) \in S \times A$; it is called *substochastic* if $\sum_j p_{ij}(a) \leq 1$ for all $(i,a) \in S \times A$. A discounted MDP with discount factor α may be considered as an MDP with total rewards for which $\sum_j p_{ij}(a) = \alpha$ for all $(i,a) \in S \times A$ and for some $\alpha \in [0,1)$.

An MDP is said to be *contracting* if there exists a vector $\mu \in R^N$ with $\mu_i > 0$ for all $i \in S$, and a scalar $\alpha \in [0,1)$ such that $\sum_j p_{ij}(a)\mu_j \leq \alpha \cdot \mu_i$ for all $(i,a) \in S \times A$. An *excessive* MDP satisfies $\sum_j p_{ij}(a)\mu_j \leq \mu_i$ for all $(i,a) \in S \times A$ for some $\mu \in R^N$ with $\mu_i > 0$ for all $i \in S$.

In a *transient* MDP every policy R is transient, i.e. $\sum_{t=1}^{\infty} p_{ij}^t(a,R) < \infty$ for all $(i,a) \in S \times A$ and all $j \in S$. If every policy R is *normalized*, i.e. $\sum_{t=1}^{\infty} \alpha^{t-1} p_{ij}^t(a,R) < \infty$ for all $(i,a) \in S \times A$, all $j \in S$ and all $\alpha \in [0,1)$, then the MDP is called normalized.

It is obvious that any transient policy is normalized, but a normalized policy is not transient, in general. Notice that a substochastic MDP is normalized, but not necessarily transient. Furthermore, a transient MDP may be non-substochatic.

A policy R is said to be *regular* if $\lim_{T\to\infty} v_i^T(R)$ exists (possibly $+\infty$ or $-\infty$ for every $i \in S$. For a regular policy R, the *total expected reward* $v_i(R)$ over the infinite horizon is defined by

$$v_i(R) = \lim_{T\to\infty} v_i^T(R) = \sum_{t=1}^{\infty} \sum_{j,a} p_{ij}^t(a,R) \cdot r_j(a),\ i \in S. \tag{4.2}$$

Let

$$v_i^+(R) := \sum_{t=1}^{\infty} \sum_{j,a} \mathbb{P}_{i,R}\{X_t = j, Y_t = a\} \cdot r_j^+(a),\ i \in S, \tag{4.3}$$

where $r_j^+(a) := max\{0, r_j(a)\}$, and

$$v_i^-(R) := \sum_{t=1}^{\infty} \sum_{j,a} \mathbb{P}_{i,R}\{X_t = j, Y_t = a\} \cdot r_j^-(a),\ i \in S, \tag{4.4}$$

where $r_j^-(a) := max\{0, -r_j(a)\}$.

The *total expected reward* $v_i(R)$ is well defined, possibly $\pm\infty$, if $min\{v_i^+(R), v_i^-(R)\} < \infty$.

The *regular value vector* v, the *transient value vector* w and the *normalized value vector* z are defined by

$$\begin{aligned} v_i &= sup\{v_i(R) \mid R \text{ is a regular policy}\},\ i \in S; & (4.5)\\ w_i &= sup\{v_i(R) \mid R \text{ is a transient policy}\},\ i \in S; & (4.6)\\ z_i &= sup\{v_i(R) \mid R \text{ is a regular and normalized policy}\},\ i \in S. & (4.7) \end{aligned}$$

A policy R^* is said to be *regular optimal* if R^* is regular and $v(R^*) = v$; R^* is *transient optimal* if R^* is transient and $v(R^*) = w$; R^* is *normalized optimal* if R^* is regular and normalized and $v(R^*) = z$. Since any transient policy is regular and normalized, we have the inequalities

$$w_i \leq z_i \leq v_i,\ i \in S. \tag{4.8}$$

4.2 Square matrices, eigenvalues and spectral radius

Let $\mathcal{M}= \{\mu \in \mathbb{R}^N \mid \mu_i > 0,\ i = 1, 2, \ldots, N\}$. For any $\mu \in \mathcal{M}$ it can easily be verified (cf. Exercise 3.1) that

$$\|x\|_\mu := max_{1\le i\le N}\, \mu^{-1} \cdot |x_i|,\ x \in \mathbb{R}^N \tag{4.9}$$

is a norm in $\mathbb{R}^N$. Let $\mathcal{P}(N)$ be the set of nonnegative $N \times N$-matrices. It is well known that the adjoint matrix norm $\|P\|_\mu$ satisfies

$$\|P\|_\mu := \|P\mu\|_\mu = max_{1\le i\le N}\, \mu^{-1} \sum_{j=1}^{N} p_{ij}\mu_i \text{ for every } P \in \mathcal{P}(N). \tag{4.10}$$

Notice that the supremum norm $\|\cdot\|_\infty$ corresponds to $\mu = e$.

Theorem 4.1
For any $P \in \mathcal{P}(N)$ and any norm $\|\cdot\|$ on $\mathbb{R}^N$, we have $\lim_{n\to\infty} \|P^n\|^{1/n} = \inf_{n\ge1} \|P^n\|^{1/n}$.

Proof
Take any $P \in \mathcal{P}(N)$ and any norm $\|\cdot\|$ on $\mathbb{R}^N$. The theorem is trivial if $P^n = 0$ for some $n \in \mathbb{N}$. Therefore, we may assume that $P^n \neq 0$ for every $n \in \mathbb{N}$. Let $a_n = log_2 \|P^n\|$. Then, we have for every $k, n \in \mathbb{N}$

$$a_{n+k} = log_2 \|P^{n+k}\| \le log_2 \{\|P^n\| \cdot \|P^k\|\} = log_2 \|P^n\| + log_2 \|P^k\| = a_n + a_k.$$

For any arbitrary, fixed $m \in \mathbb{N}$, we can write $n = m \cdot q_n + r_n$ for every $n \in \mathbb{N}$, where q_n and r_n are nonnegative integers with $0 \le r_n < m$. Then, $a_n = a_{m\cdot q_n + r_n} \le a_m \cdot q_n + a_{r_n}$, and consequently,

$\frac{a_n}{n} \le \frac{a_m\cdot q_n}{m\cdot q_n + r_n} + \frac{a_{r_n}}{n} = \frac{a_m}{m+\frac{r_n}{q_n}} + \frac{a_{r_n}}{n} \le \frac{a_m}{m} + \frac{a_{r_n}}{n}$, implying $\limsup_{n\to\infty} \frac{a_n}{n} \le \frac{a_m}{m}$.

Since m is arbitrarily chosen, we get $\limsup_{n\to\infty} \frac{a_n}{n} \le \inf_{m\ge1} \frac{a_m}{m} \le \limsup_{m\to\infty} \frac{a_m}{m}$, which proves that $\lim_{n\to\infty} \frac{a_n}{n}$ exists and equals $\inf_{n\ge1} \frac{a_n}{n}$. Furthermore, we have

$$\begin{aligned} \lim_{n\to\infty} \|P^n\|^{1/n} &= \lim_{n\to\infty} 2^{\,a_n/n} = 2^{\,\lim_{n\to\infty} a_n/n} \\ &= 2^{\,\inf_{n\ge1} a_n/n} = \inf_{n\ge1} 2^{\,a_n/n} = \inf_{n\ge1} \|P^n\|^{1/n}. \end{aligned}$$

□

We say that two norms, $\|\cdot\|_A$ and $\|\cdot\|_B$ on $\mathbb{R}^N$ are *equivalent* if there exist $m, M > 0$ such that

$$m \cdot \|x\|_B \le \|x\|_A \le m \cdot \|x\|_B \text{ for every } x \in \mathbb{R}^N.$$

The L_1-norm $\|\cdot\|_1$ and the L_2-norm $\|\cdot\|_2$ on $\mathbb{R}^N$ are defined by

‘ $\|x\|_1 := \sum_{i=1}^N |x_i|$ and $\|x\|_2 := \sqrt{\sum_{i=1}^N |x_i|^2}$, respectively.

Lemma 4.1
The L_1-norm and L_2-norm are equivalent.

Proof
We first show that $\|x\|_2 \le \|x\|_1$, or equivalently, $\|x\|_2^2 = \sum_{i=1}^N |x_i|^2 \le \|x\|_1^2 = \left(\sum_{i=1}^N |x_i|\right)^2$.

We can write

$$\|x\|_1^2 = \left(\sum_{i=1}^N |x_i|\right)^2 = \left(\sum_{i=1}^N |x_i|\right) \cdot \left(\sum_{j=1}^N |x_j|\right) \ge \sum_{i=1}^N |x_i| \cdot |x_i| = \sum_{i=1}^N |x_i|^2 = \|x\|_2^2.$$

Next we show that $\|x\|_1 \le \sqrt{N} \cdot \|x\|_2$, or equivalently, $\|x\|_1^2 = \left(\sum_{i=1}^N |x_i|\right)^2 \le N \cdot \|x\|_2^2 = N \cdot \sum_{i=1}^N |x_i|^2$.

We apply induction on N. For $N = 1$ the inequality holds with equality.
Assume that $\big(\sum_{i=1}^{N-1} |x_i|\big)^2 \leq (N-1) \cdot \sum_{i=1}^{N-1} |x_i|^2$. Then, we have

$$\begin{array}{rcl} \big(\sum_{i=1}^{N} |x_i|\big)^2 & = & \big(\sum_{i=1}^{N-1} |x_i| + |x_N|\big)^2 \\ & = & \big(\sum_{i=1}^{N-1} |x_i|\big)^2 + 2 \cdot |x_N| \cdot \sum_{i=1}^{N-1} |x_i| + |x_N|^2 \\ & \leq & (N-1) \cdot \sum_{i=1}^{N-1} |x_i|^2 + 2 \cdot |x_N| \cdot \sum_{i=1}^{N-1} |x_i| + |x_N|^2 \end{array}$$

Hence, we have to show that $\sum_{i=1}^{N-1} |x_i|^2 + (N-1) \cdot |x_N|^2 - 2 \cdot |x_N| \cdot \sum_{i=1}^{N-1} |x_i| \geq 0$. Indeed, we obtain $\sum_{i=1}^{N-1} |x_i|^2 + (N-1) \cdot |x_N|^2 - 2 \cdot |x_N| \cdot \sum_{i=1}^{N-1} |x_i| = \sum_{i=1}^{N-1} \big(|x_i| - |x_N|\big)^2 \geq 0$. □

Theorem 4.2
All norms in R^N are equivalent.

Proof
We shall demonstrate that any norm $\|\cdot\|$ on R^N is equivalent to the L_2-norm. Since the relation 'equivalent norms' is an equivalence relation, it then follows that all norms in R^N are equivalent. Take any norm $\|\cdot\|$ on R^N and let $\{e^i\}_{i=1}^N$ be a basis for R^N. Then, any vector $x \in R^N$ has an expression as $x = \sum_{i=1}^N x_i \cdot e^i$. First, let us check that $\|\cdot\|$ is continuous with respect to the L_2-norm. For all pair $x, y \in \mathbb{R}^N$, we can write

$$\|x - y\| = \| \sum_{i=1}^N (x_i - y_i) \cdot e^i \| \leq \sum_{i=1}^N |x_i - y_i| \cdot \|e^i\| \leq M_1 \cdot \|x - y\|_1 \leq M_2 \cdot \|x - y\|_2,$$

where $M_1 := max_{1 \leq i \leq N} \|e^i\|$ and $M_1 \cdot \|x - y\|_1 \leq M_2 \cdot \|x - y\|_2$ for some $M_2 > 0$ because the L_1-norm and L_2-norm are equivalent (see Lemma 4.1). In other words, when x and y are 'nearby' with respect tot the L_2-norm, they are also 'nearby' with respect to any other norm.
Now, consider the unit sphere S with respect to the L_2-norm, i.e. $S = \{x \in \mathbb{R}^N \mid \|x\|_2 = 1\}$. This is a compact set. Therefore, the continuous function $\|\cdot\|$ attains maximum and minimum values on S, say m and M, respectively. Hence, $m \cdot \|x\|_2 \leq \|x\| \leq M \cdot \|x\|_2$ for any $x \in \mathbb{R}^N$ with $\|x\|_2 = 1$. Every $y \in \mathbb{R}^N$ can be expressed as $y = c \cdot x$ for some $x \in S$ and some $c \in \mathbb{R}$. Therefore, we also have $m \cdot \|y\|_2 \leq \|y\| \leq M \cdot \|y\|_2$ for any $y \in \mathbb{R}^N$. The scalars m and M are obviously nonnegative and $m \leq M$. Assume that $m = 0$. Then, there exists a point x on S for which $\|x\| = 0$. But then $x = 0$, which contradicts $\|x\|_2 = 1$. □

Let $P \in \mathcal{P}(N)$. It is easily seen that the *characteristic polynomial* $\phi_P(\lambda) := det\,(P - \lambda I)$ is a Nth degree polynomial of the form

$$\phi_P(\lambda) = (-1)^N \{\lambda^N + \alpha_{N-1}\lambda^{N-1} + \cdots + \alpha_1\lambda + \alpha_0\}. \tag{4.11}$$

The zeroes of $\phi_P(\lambda)$ are the *eigenvalues* of P. If $\lambda_1, \lambda_2, \dots, \lambda_k$ are the distinct eigenvalues, then $\phi_P(\lambda)$ can be represented in the form

$$\phi_P(\lambda) = (-1)^N (\lambda - \lambda_1)^{\sigma_1} (\lambda - \lambda_2)^{\sigma_2} \cdots (\lambda - \lambda_k)^{\sigma_k}, \tag{4.12}$$

where the integer σ_i, which is also denoted by $\sigma(\lambda_i)$, is called the *algebraic multiplicity* of the eigenvalue λ_i. A vector $x^i \neq 0$ such that $Px^i = \lambda_i\, x^i$ is an *eigenvector* of λ_i. The set $L(\lambda_i) := \{x^i \mid Px^i = \lambda_i\, x^i\}$ is a linear space of dimension $\rho(\lambda_i) = N - rank(P - \lambda_i I)$, which is the *geometric multiplicity* of the eigenvalue λ_i. The algebraic and geometric multiplicity of an eigenvalue can be different (see Example 4.3).

Let λ_i be an eigenvalue of P and let x^i be an eigenvector of λ_i with respect to P. Furthermore, let T be an arbitrary nonsingular $N \times N$ matrix. For $y^i := T^{-1}x^i \neq 0$ one can write, with $Q := T^{-1}PT$,

$$Qy^i = T^{-1}PTy^i = T^{-1}Px^i = \lambda_i T^{-1}x^i = \lambda_i y^i, \tag{4.13}$$

i.e. λ_i is an eigenvalue of Q and y^i is an eigenvector of λ_i with respect to Q. $Q = T^{-1}PT$ is called a *similarity transformation* of P. One easily shows that similarity of matrices is an equivalence relation. Similar matrices have not only the same eigenvalues, but also the same characteristic polynomial, namely:

$$\begin{aligned}\phi_Q(\lambda) &= det\,(Q-\lambda I) = det\,(T^{-1}PT-\lambda I) = det\,\{T^{-1}(P-\lambda I)T\} \\ &= det\,(T^{-1})\cdot det\,(P-\lambda I)\cdot det\,(T) = det\,(P-\lambda I) = \phi_P(\lambda).\end{aligned}$$

Moreover, the algebraic and geometric multiplicity of the eigenvalues, i.e. $\rho(\lambda_i)$ and $\sigma(\lambda_i)$ remain the same. For $\sigma(\lambda_i)$ this follows from the invariance of the characteristic polynomial, and for $\rho(\lambda_i)$ it follows from the fact that, T being nonsingular, the eigenvectors $x^{i,1}$, $x^{i,2}$, $\ldots$, $x^{i,\rho(\lambda_i)}$ are linearly independent if and only if the corresponding vectors $x^{i,1} = T^{-1}x^{i,1}$, $y^{i,2} = T^{-1}x^{i,2}$, $\ldots$, $y^{i,\rho(\lambda_i)} = T^{-1}x^{i,\rho(\lambda_i)}$ are linearly independent.

Lemma 4.2

Let $\lambda_1, \lambda_2, \ldots, \lambda_k$ be the distinct eigenvalues of $P \in \mathcal{P}(N)$. Then, $1 \le \rho(\lambda_i) \le \sigma(\lambda_i)$ for $i = 1, 2, \ldots, k$.

Proof

Take any $1 \le i \le k$. We prove only the nontrivial part $\rho(\lambda_i) \le \sigma(\lambda_i)$. Let $x^{i,1}$, $x^{i,2}$, $\ldots$, $x^{i,\rho(\lambda_i)}$ be the linearly independent independent eigenvalues associated with λ_i: $Px^{i,j} = \lambda_i\, x^{i,j}$ for $j = 1, 2, \ldots, \rho(\lambda_i)$. We select $N - \rho(\lambda_i)$ additional linearly independent vectors $x^{i,j}$ for $j = \rho(\lambda_i)+1, \rho(\lambda_i)+2, \ldots, N$, such that $x^{i,j}, j = 1, 2, \ldots, N$ form a basis in $\mathbb{R}^N$. Then, the $N \times N$ matrix T_i with columns $x^{i,j}, j = 1, 2, \ldots, N$ is nonsingular. In view of $T_i e^j = x^{i,j}$, we have

$$T_i^{-1}PT_ie^j = T_i^{-1}Px^{i,j} = \lambda_i T_i^{-1}x^{i,j} = \lambda_i e^j \text{ for } j = 1, 2, \ldots, \rho(\lambda_i).$$

Therefore, $T_i^{-1}PT_i$ has the form $T_i^{-1}PT_i = \begin{pmatrix} \lambda_i & B \\ 0 & C \end{pmatrix}$ and for the characteristic polynomial of P and $T_i^{-1}PT_i$, we obtain

$$\phi_P(\lambda) = det\,(P-\lambda I) = det\,(T_i^{-1}PT_i - \lambda I) = (\lambda_i - \lambda)^{\rho(\lambda_i)}\cdot det\,(C-\lambda I).$$

Hence, $\phi_P(\lambda)$ is divisible by $(\lambda_i-\lambda)^{\rho(\lambda_i)}$ i.e. λ_i is a zero of $\phi_P(\lambda)$ of multiplicity at least $\rho(\lambda_i)$, which implies $\rho(\lambda_i) \le \sigma(\lambda_i)$.

□

Example 4.1

Consider the matrix $J_i = \begin{pmatrix} \lambda_i & 1 & 0 & 0 & \dots & 0 \\ 0 & \lambda_i & 1 & 0 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & 0 & \lambda_i & 1 \\ 0 & 0 & 0 & 0 & 0 & \lambda_i \end{pmatrix}$. Such a matrix is called a *Jordan block*.

Then, $\phi_{J_i}(\lambda) = (\lambda_i - \lambda)^N$, so $\sigma(\lambda_i) = N$.

Since $J_i - \lambda I = \begin{pmatrix} 0 & 1 & 0 & 0 & \dots & 0 \\ 0 & 0 & 1 & 0 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$, we have $rank\,(J_i - \lambda I) = N-1$. Hence,

$\rho(\lambda_i) = N - rank\,(J_i - \lambda I) = N - (N-1) = 1$. Since $J_i e^1 = \lambda_i e^1$, the unique eigenvector (up to scalar multiples) of J_i is e^1 and consequently, $L(\lambda_i) = \{x \mid x = c\cdot e^1;\ c \in \mathbb{R}\}$.

A *Jordan normal form matrix* J is a block diagonal matrix whose blocks are all Jordan matrices, i.e.

$$J = \begin{pmatrix} J_1 & 0 & 0 & 0 & \cdots & 0 \\ 0 & J_2 & 1 & 0 & \cdots & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & 0 & 0 & J_{p-1} & 1 \\ 0 & 0 & 0 & 0 & 0 & J_p \end{pmatrix}, \text{ where } J_i = \begin{pmatrix} \lambda_i & 1 & 0 & 0 & \cdots & 0 \\ 0 & \lambda_i & 1 & 0 & \cdots & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & 0 & 0 & \lambda_i & 1 \\ 0 & 0 & 0 & 0 & 0 & \lambda_i \end{pmatrix}.$$

There is a fundamental theorem which we state without proof (the proof can be found in books on linear algebra, e.g. see p. 216 in [164]).

Theorem 4.3

Let A be an $N \times N$ matrix with $\lambda_1, \lambda_2, \cdots, \lambda_k$ as distinct eigenvalues with geometric and algebraic multiplicities $\rho(\lambda_i)$ and $\sigma(\lambda_i)$, respectively, $i = 1, 2, \ldots, k$. Then, for each of the eigenvalues λ_i there exist natural numbers ν_1^i, ν_2^i, $\nu_{\rho(\lambda_i)}^i$ such that

(1) $\sigma(\lambda_i) = \sum_{j=1}^{\rho(\lambda_i)} \nu_j^i$;

(2) there exists a nonsingular $N \times N$ matrix T such that $J = T^{-1}AT$ has the following Jordan normal form:

$$J = \begin{pmatrix} J_{\nu_1^1} & \cdots & 0 & 0 & \cdots & \cdots & \cdots & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ 0 & \cdots & J_{\nu^1_{\nu(\lambda_1)}} & 0\cdots & \cdots & \cdots & 0 & \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ 0 & \cdots & 0 & 0 & 0 & J_{\nu_1^k} & \cdots & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ 0 & \cdots & 0 & 0 & 0 & 0 & \cdots & J_{\nu^k_{\nu(\lambda_k)}} \end{pmatrix} \textit{ with } J_{\nu_j^i} \textit{ a } \nu_j^i \times \nu_j^i \textit{ Jordan matrix.}$$

Example 4.2

Consider the matrix $A = \begin{pmatrix} 2 & 4 & -8 \\ 0 & 0 & 4 \\ 0 & -1 & 4 \end{pmatrix}$. Then, it is easy to verify that this matrix has only $\lambda = 2$ as eigenvalue with algebraic multiplicity $\sigma(\lambda) = 3$. Since $A - 2I = \begin{pmatrix} 0 & 4 & -8 \\ 0 & -2 & 4 \\ 0 & -1 & 2 \end{pmatrix}$ has rank 1, $\rho(\lambda) = 2$. Then, $\sigma(\lambda) = 3 = \nu_1 + \nu_2$. Hence, the Jordan normal form has 2 blocks, one block 2×2 and one block 1×1 with in each block a Jordan matrix. Therefore, $A = \begin{pmatrix} 2 & 1 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 2 \end{pmatrix}$. The nonsingular matrix $T = \begin{pmatrix} 4 & 0 & 1 \\ -2 & 1 & 0 \\ -1 & 0 & 0 \end{pmatrix}$. It is also easy to verify that $J = T^{-1}AT$.

The *spectral radius* $\rho(P)$ of $P \in \mathcal{P}(N)$ is defined by

$$\rho(P) = max\,\{|\lambda| \mid \lambda \text{ is an eigenvalue of } P\}. \tag{4.14}$$

Let λ be any eigenvalue of $P \in \mathcal{P}(N)$ with eigenvector v. Then, we can write

$$|\lambda|^n \cdot \|v\| = \|\lambda^n v\| = \|P^n v\| \leq \|P^n\| \cdot \|v\| \text{ for all } n \in /N.$$

Since $v \neq 0$, $|\lambda| \leq \|P^n\|^{1/n}$ for all $n \in \mathbb{N}$. From the definition of $\rho(P)$ and Theorem 4.1 it follows that

$$\rho(P) \leq max\,\{|\lambda| \mid \lambda \text{ is an eigenvalue of } P\} \leq \lim_{n\to\infty} \|P^n\|^{1/n} = \inf_{n\geq 1} \|P^n\|^{1/n}.$$

We will show (see Theorem 4.4) that $\rho(P) = \lim_{n\to\infty} \|P^n\|^{1/n} = \inf_{n\geq 1} \|P^n\|^{1/n}$.

Lemma 4.3
Let $P \in \mathcal{P}(N)$. Then, $\lim_{n\to\infty} P^n = 0$ if and only if $\rho(P) < 1$.

Proof

$\Rightarrow$ Let λ be any eigenvalue of $P \in \mathcal{P}(N)$ with eigenvector v. Then, $P^n v = \lambda^n v$ and we have

$$0 = \lim_{n\to\infty} P^n v = \lim_{n\to\infty} \lambda^n v.$$

Since $v \neq 0$, $\lim_{n\to\infty} \lambda^n = 0$, which implies $|\lambda| < 1$. Since this must be true for any eigenvalue λ, we can conclude $\rho(P) < 1$.

$\Leftarrow$ Let P has $\lambda_1, \lambda_2, \ldots, \lambda_k$ as distinct eigenvalues with geometric and algebraic multiplicities $\rho(\lambda_i)$ and $\sigma(\lambda_{})$, respectively, for $i = 1, 2, \ldots, k$. From the Jordan Normal Form Theorem, we know that for each $i = 1, 2, \ldots, k$ there exist natural numbers ν_j for $j = 1, 2, \ldots, \rho(\lambda_i)$ with $\sigma(\lambda_i) = \sum_{j=1}^{\rho(\lambda_i)} \nu_j^i$, and there exists a nonsingular matrix T such that $J = T{-}1PT$ has the following Jordan normal form:

$$J = \begin{pmatrix} J_{\nu_1^1} & \cdots & 0 & 0 & \cdots & \cdots & \cdots & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ 0 & \cdots & J_{\nu^1_{\nu(\lambda_1)}} & 0\cdots & \cdots & \cdots & 0 & \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ 0 & \cdots & 0 & 0 & 0 & J_{\nu_1^k} & \cdots & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ 0 & \cdots & 0 & 0 & 0 & 0 & \cdots & J_{\nu^k_{\nu(\lambda_k)}} \end{pmatrix} \text{ with } J_{\nu_j^i} \text{ a } \nu_j^i \times \nu_j^i \text{ Jordan matrix.}$$

Note that $P^n = TJ^nT^{-1}$ for all $n \in \mathbb{N}$. Since J is a block-diagonal matrix, matrix J^n is also a block-diagonal matrix:

$$J^n = \begin{pmatrix} J^n_{\nu_1^1} & \cdots & 0 & 0 & \cdots & \cdots & \cdots & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ 0 & \cdots & J^n_{\nu^1_{\nu(\lambda_1)}} & 0\cdots & \cdots & \cdots & 0 & \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ 0 & \cdots & 0 & 0 & 0 & J^n_{\nu_1^k} & \cdots & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ 0 & \cdots & 0 & 0 & 0 & 0 & \cdots & J^n_{\nu^k_{\nu(\lambda_k)}} \end{pmatrix},$$

where

$$J^n_{\nu_j^i} = \begin{pmatrix} \lambda_i^n & \binom{n}{1}\lambda_i^{n-1} & \binom{n}{2}\lambda_i^{n-2} & \binom{n}{3}\lambda_i^{n-3} & \cdots & \binom{n}{\nu_j^i-1}\lambda_i^{n-\nu_j^i+1} \\ 0 & \lambda_i^n & \binom{n}{1}\lambda_i^{n-1} & \binom{n}{2}\lambda_i^{n-2} & \cdots & \binom{n}{\nu_j^i-2}\lambda_i^{n-\nu_j^i+1} \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & 0 & 0 & \lambda_i^n & \binom{n}{1}\lambda_i^{n-1} \\ 0 & 0 & 0 & 0 & 0 & \lambda_i^n \end{pmatrix}.$$

The proof that the above $J^n_{\nu^i_j}$ is correct can be given by induction on n. For $n = 1$ the formula holds, where $\binom{n}{m} := 0$ if $n < m$, because $J^n_{\nu^i_j}$ is a Jordan block for $n = 1$. Assuming the (k,l)th element of $J^n_{\nu^i_j}$ equals $\binom{n}{l-k}\lambda_i^{n-l+k}$ for $k \le l$, the (k,l)th element of $J^{n+1}_{\nu^i_j} = \sum_p \{J^n_{\nu^i_j}\}_{kp} \cdot \{J_{\nu^i_j}\}_{pk}$. Since $\{J_{\nu^i_j}\}_{pk} \neq 0$ only for $p = l$ or $p = l-1$, we obtain

$$\begin{aligned}\{J^{n+1}_{\nu^i_j}\}_{kl} &= \{J^n_{\nu^i_j}\}_{k,l-1} \cdot \{J_{\nu^i_j}\}_{l-1,k} + \{J^n_{\nu^i_j}\}_{kl} \cdot \{J_{\nu^i_j}\}_{lk} \\ &= \tbinom{n}{l-1-k}\lambda_i^{n-l+1+k} \cdot 1 + \tbinom{n}{l-k}\lambda_i^{n-l+k} \cdot \lambda_i \\ &= \{\tbinom{n}{l-1-k} + \tbinom{n}{l-k}\} \cdot \lambda_i^{n-l+1+k} = \tbinom{n+1}{l-k} \cdot \lambda_i^{(n+1)-l+k}.\end{aligned}$$

Since $\lambda_i < 1$ for all eigenvalues λ_i, $\lim_{n\to\infty} J^n = 0$ and consequently, $\lim_{n\to\infty} P^n = 0$. □

Lemma 4.4
Let $P \in \mathcal{P}(N)$. Then, if $\rho(P) > 1$, $\|P^n\|$ is not bounded for increasing n values.

Proof
Write, as in Lemma 4.3, $P^n = TJ^nT^{-1}$ with J the Jordan matrix and T a nonsingular matrix. Since $\lambda_i > 1$ for at least one eigenvalue λ_i, it follows that there is at least one element in J^n which does not remain bounded as n increases, so proving the statement of the lemma. □

Theorem 4.4 *Gelfand's formula (1941)*
Let $P \in \mathcal{P}(N)$. Then, $\rho(P) = \lim_{n\to\infty} \|P^n\|^{1/n} = \inf_{n\ge1} \|P^n\|^{1/n}$.

Proof
For any $\varepsilon > 0$, consider the matrix $P_1(\varepsilon) := \frac{1}{\rho(P)+\varepsilon} \cdot P$. Obviously, $\rho\big(P_1(\varepsilon)\big) = \frac{\rho(P)}{\rho(P)+\varepsilon} < 1$. Then, by Lemma 4.3, $\lim_{n\to\infty} P_1^n(\varepsilon) = 0$, implying the existence of an integer n_1 such that $\|P_1^n(\varepsilon)\| < 1$ for all $n \ge n_1$. This means that $\|P^n\| < \{\rho(P)+\varepsilon\}^n$ for all $n \ge n_1$, i.e. $\|P^n\|^{1/n} < \rho(P)+\varepsilon$ for all $n \ge n_1$.
On the other hand, consider the matrix $P_2(\varepsilon) := \frac{1}{\rho(P)-\varepsilon} \cdot P$. Since $\rho\big(P_2(\varepsilon)\big) = \frac{\rho(P)}{\rho(P)-\varepsilon} > 1$, by Lemma 4.4, there exists an integer n_2 such that $\|P_2^n(\varepsilon)\| > 1$ for all $n \ge n_2$. This implies $\|P^n\| > \{\rho(P)-\varepsilon\}^n$ for all $n \ge n_2$, and consequently $\|P^n\|^{1/n} > \rho(P)-\varepsilon$ for all $n \ge n_2$.
Taking $n_3 := max(n_1, n_2)$, we obtain $\rho(P)-\varepsilon < \|P^n\|^{1/n} < \rho(P)+\varepsilon$ for all $n \ge n_3$. Since ε was arbitrarily chosen, $\rho(P) = \lim_{n\to\infty} \|P^n\|^{1/n} = \inf_{n\ge1} \|P^n\|^{1/n}$, the last equality by Theorem 4.1. □

Theorem 4.5
For any $P \in \mathcal{P}(N)$ and any norm $\|\cdot\|$ in $\mathbb{R}^N$, the following five statements are equivalent:
(1) $|\lambda| < 1$ for every eigenvalue λ of P.
(2) $\rho(P) < 1$.
(3) $|P^n\| < 1$ for some $n \ge 1$.
(4) $I - P$ is nonsingular and $(I-P)^{-1} = \sum_{n=0}^\infty P^n$.
(5) $\lim_{n\to\infty} P^n = 0$.

Proof
(1) and (2) are equivalent by the definition of the spectral radius; (2) and (3) are equivalent by Theorem 4.4; (2) and (5) are equivalent by Lemma 4.3. Therefore, it is sufficient to show that (4) and (5) are equivalent.

Assume that (4) holds. Since $\sum_{n=0}^{\infty} P^n$ exists, obviously $\lim_{n\to\infty} P^n = 0$, so (5) holds. On the other hand, assume that (5) holds. Since $(I-P)(I+P+\cdots+P^{n-1}) = I - P^n$ and $\lim_{n\to\infty} P^n = 0$, we can write

$$det\,(I-P)\cdot det\,(I+P+\cdots+P^{n-1}) \to det\,(I) = 1 \text{ for } n\to\infty.$$

Therefore, $det\,(I-P) \neq 0$, implying $I-P$ is nonsingular and $(I-P)^{-1} = \sum_{n=0}^{\infty} P^n$. □

Theorem 4.6
Let $P \in \mathcal{P}(N)$, $I-P$ nonsingular and $\rho(P) \leq 1$. Then, $\rho(P) < 1$

Proof
For any $\alpha \in [0,1)$, we have $\rho(\alpha P) = \alpha\rho(P) < 1$. By Theorem 4.5, $(I-\alpha P)^{-1} = \sum_{n=0}^{\infty} \alpha^n P^n$. Since the matrix $I-P$ is nonsingular and the elements of $(I-\alpha P)^{-1}$ are rational functions in α with as numerator the determinant of $I-\alpha P$ (this is based on Cramers rule, see e.g. Karlin [155] p. 387), we have $(I-P)^{-1} = \lim_{\alpha\uparrow 1} \sum_{n=0}^{\infty} \alpha^n P^n$. Since P is nonnegative the monotone convergence theorem (cf. Loève [185] p. 124) implies

$$\lim_{\alpha\uparrow 1} \sum_{n=0}^{\infty} \alpha^n P^n = \sum_{n=0}^{\infty} \lim_{\alpha\uparrow 1} \alpha^n P^n = \sum_{n=0}^{\infty} P^n = (I-P)^{-1}.$$

Hence, by Theorem 4.5, $\rho(P) < 1$. □

4.3 The linear program

In this section we discuss some properties of a linear program that will be used in the sequel of this chapter. In particular, we establish the correspondence between the randomized stationary transient policies and the feasible solutions of this linear program. Furthermore, we show that the deterministic transient policies correspond to the extreme solutions of the program. Let $\beta \in \mathbb{R}^N$ with $\beta_j > 0$ for every $j \in S$. Then, consider the following linear program:

$$max \left\{ \sum_{(i,a)} r_i(a)x_i(a) \;\middle|\; \begin{array}{rcl} \sum_{(i,a)} \{\delta_{ij} - p_{ij}(a)\}x_i(a) & = & \beta_j,\ j \in S \\ x_i(a) & \geq & 0,\ (i,a) \in S\times A \end{array} \right\}. \tag{4.15}$$

For any feasible solution $x \in \mathbb{R}^{|S\times A|}$ of (4.15) we define a vector, also denoted by x and with $x \in R^N$ by $x_i := \sum_a x_i(a),\ i \in S$. Hence, we have $x_j = \sum_a x_j(a) = \beta_j + \sum_{(i,a)} p_{ij}(a)x_i(a) \geq \beta_j > 0$ for all $j \in S$. Define a stationary policy $\pi^\infty(x)$ by

$$\pi_{ia}(x) := \frac{x_i(a)}{x_i},\ (i,a) \in S\times A. \tag{4.16}$$

From (4.16), we obtain $x_i(a) = \pi_{ia}(x)\cdot x_i,\ (i,a)\in S\times A$. Therefore,

$$x_j = \beta_j + \sum_{(i,a)} p_{ij}(a)\pi_{ia}(x)\cdot x_i = \beta_j + \sum_i p_{ij}\big(\pi(x)\big)\cdot x_i,\ j \in S.$$

In vector notation, $x^T = \beta^T + x^T P\big(\pi(x)\big)$. Iterating this equality yields

$$x^T = \beta^T \sum_{t=1}^{n} P^{t-1}\big(\pi(x)\big) + x^T P^n\big(\pi(x)\big) \geq \beta^T \sum_{t=1}^{n} P^{t-1}\big(\pi(x)\big) \text{ for all } n \in \mathbb{N}.$$

Hence, $\sum_{t=1}^{\infty} P^{t-1}\big(\pi(x)\big)$ exists and has finite components: $\sum_{t=1}^{\infty} \{P^{t-1}\big(\pi(x)\big)\}_{ij} < \infty$ for all $i,j \in S$. Therefore, the stationary policy $\pi^\infty(x)$ is transient and satisfies

$$x^T = \beta^T \sum_{t=1}^{\infty} P^{t-1}\big(\pi(x)\big) = \beta^T\{I - P\big(\pi(x)\big)\}^{-1}. \tag{4.17}$$

Conversely, let π^∞ be an arbitrary transient stationary policy. Then, $P^n(\pi) \to 0$ if $n \to \infty$. Therefore, by Theorem 4.5, $\big(I - P(\pi)\big)^{-1}$ exists. We define the vector $x(\pi) \in \mathbb{R}^{|S \times A|}$ by

$$x_{ia}(\pi) := \{\beta^T \big(I - P(\pi)\big)^{-1}\}_i \cdot \pi_{ia},\ (i,a) \in S \times A. \tag{4.18}$$

Theorem 4.7
The mapping (4.18) is a bijection between the set of transient stationary policies and the set of feasible solutions of (4.15) with (4.16) as the inverse mapping. Furthermore, the set of transient deterministic policies corresponds to the set of extreme feasible solutions of (4.15).

Proof
First, we prove that $x(\pi)$, for any transient stationary policy π^∞, is a feasible solution of (4.15).

$$\begin{aligned}
\textstyle\sum_{(i,a)} \{\delta_{ij} - p_{ij}(a)\}x_{ia}(\pi) &= \textstyle\sum_a x_{ja}(\pi) - \sum_{(i,a)} p_{ij}(a)x_{ia}(\pi) \\
&= \{\beta^T \big(I - P(\pi)\big)^{-1}\}_j - \textstyle\sum_{(i,a)} p_{ij}(a)\{\beta^T \big(I - P(\pi)\big)^{-1}\}_i \cdot \pi_{ia} \\
&= \{\beta^T \big(I - P(\pi)\big)^{-1}\}_j - \textstyle\sum_i \{\beta^T \big(I - P(\pi)\big)^{-1}\}_i \cdot \{P(\pi)\}_{ij} \\
&= \{\beta^T \big(I - P(\pi)\big)^{-1}\}_j - \{\beta^T \big(I - P(\pi)\big)^{-1}\}\{P(\pi)\}_j \\
&= \{\beta^T \big(I - P(\pi)\big)^{-1}\}\{(I - P(\pi)\}_j = \beta_j,\ j \in S.
\end{aligned}$$

Furthermore, $x_{ia}(\pi) = \{\beta^T \big(I - P(\pi)\big)^{-1}\}_i \cdot \pi_{ia} = \{\beta^T \sum_{t=1}^\infty P^{t-1}(\pi)\}_i \cdot \pi_{ia} \geq 0$ for all $(i,a) \in S \times A$. Hence, $x(\pi)$ is a feasible solution of (4.15). The relations (4.16), (4.17) and (4.18) imply $x\big(\pi(x)\big) = x$ and $\pi\big(x(\pi)\big) = \pi$, i.e. the mapping (4.18) is a bijection between the set of transient stationary policies and the set of feasible solutions of (4.15) with (4.16) as the inverse mapping.

Let f^∞ be an arbitrary deterministic transient policy. Suppose that $x(f)$ is not an extreme feasible solution of (4.15). Then, there exist feasible solutions x^1 and x^2 of program (4.15) and a real number $\lambda \in (0,1)$ such that $x^1 \neq x^2$ and $x(f) = \lambda x^1 + (1-\lambda)x^2$. Since $x_{ia}(f) = 0$ for all $a \neq f(i),\ i \in S$, also $x^1_{ia} = x^2_{ia} = 0$ for all for all $a \neq f(i),\ i \in S$. Hence, the Nth-dimensional vectors x^1 and x^2 with components $x^1_i\big(f(i)\big),\ i \in S$ and $x^2_i\big(f(i)\big),\ i \in S$, respectively, are solutions of the linear system $x^T\{I - P(f)\} = \beta^T$. Since f^∞ is a transient policy, the matrix $\{I - P(f)\}$ is nonsingular and consequently, the system $x^T\{I - P(f)\} = \beta^T$ has a unique solution, namely $\beta^T\{I - P(f)\}^{-1}$. This implies $x^1 = x^2$, which yields a contradiction.

Conversely, let x be an arbitrary extreme feasible solution of (4.15). Since (4.15) has N constraints, x has at most N positive components. Since $\sum_a x_j(a) = \beta_j + \sum_{(i,a)} p_{ij}(a)x_{ia} \geq \beta_j > 0$ for all $j \in S$, x has precisely N positive components, for each state j exactly one. Hence, the corresponding policy $\pi^\infty(x)$ is a deterministic policy. □

4.4 Transient, contracting, excessive and normalized MDPs

We start this section with a lemma that shows that the total expected reward of a regular policy is the limit of the discounted expected reward when the discount factor α tends to 1.

Lemma 4.5
For any regular policy R the expected total reward satisfies: $v_i(R) = \lim_{\alpha \uparrow 1} v^\alpha_i(R),\ i \in S.$

Proof

Take any initial state i and any policy R. We distinguish the following cases.

Case 1: $-\infty < v_i(R) < +\infty$.

Let $v_i^{(t)}(R)$ be the expected reward in period t: $v_i^{(t)}(R) := \sum_{j,a} \mathbb{P}_{i,R}\{X_t = j, Y_t = a\} \cdot r_j(a)$.

Take any $\varepsilon > 0$. Then, there exists a T_* such that $|v_i(R) - \sum_{t=1}^T v_i^{(t)}(R)| < \varepsilon$ for every $T \geq T_*$.

Since $|v_i^{(t)}(R)|$ is bounded by $M := max_{(i,a)} |r_i(a)|$, the two power series $v_i^\alpha(R) := \sum_{t=1}^\infty \alpha^{t-1} v_i^{(t)}(R)$ and $\sum_{s=1}^\infty \alpha^{s-1}$ have radius of convergence (at least) 1. Hence, for any $\alpha \in [0,1)$, we may write

$$(1-\alpha)^{-1} v_i^\alpha(R) = \Big\{\sum_{s=1}^\infty \alpha^{s-1}\Big\}\Big\{\sum_{t=1}^\infty \alpha^{t-1} v_i^{(t)}(R)\Big\} = \sum_{t=1}^\infty \Big\{\sum_{s=1}^t v_i^{(s)}(R)\Big\} \cdot \alpha^{t-1}.$$

Therefore,

$$\begin{array}{l} |(1-\alpha)^{-1}\{v_i^\alpha(R) - v_i(R)\}| \;\leq\; \sum_{t=1}^\infty |\sum_{s=1}^t v_i^{(s)}(R) - v_i(R)| \cdot \alpha^{t-1} \;= \\ \sum_{t=1}^{T_*} |\sum_{s=1}^t v_i^{(s)}(R) - v_i(R)| \cdot \alpha^{t-1} + \sum_{t=T_*+1}^\infty |\sum_{s=1}^t v_i^{(s)}(R) - v_i(R)| \cdot \alpha^{t-1}. \end{array}$$

Let $A = max_{1 \leq t \leq T_*} |\sum_{s=1}^t v_i^{(s)}(R) - v_i(R)|$. Then, we obtain

$$\begin{array}{rcl} |(1-\alpha)^{-1}\{v_i^\alpha(R) - v_i(R)\}| & \leq & \sum_{t=1}^{T_*} A \cdot \alpha^{t-1} + \sum_{t=T_*+1}^\infty \varepsilon \cdot \alpha^{t-1} \\ & \leq & A \cdot \frac{1-\alpha^{T_*}}{1-\alpha} + \varepsilon \cdot \sum_{t=1}^\infty \alpha^{t-1} < 2\varepsilon(1-\alpha)^{-1} \end{array}$$

for α sufficiently close to 1. Hence, $|v_i^\alpha(R) - v_i(R)| < 2\varepsilon$ for α sufficiently close to 1. This implies $\lim_{\alpha\uparrow 1} v_i^\alpha(R) = v_i(R)$.

Case 2: $v_i(R) = +\infty$.

Choose $M > 0$ arbitrary. Then, there exists an integer T_* such that $\sum_{t=1}^T v_i^{(t)}(R) > M$ for all $T > T^*$. Similarly as in case 1 we can write

$$\begin{array}{rcl} (1-\alpha)^{-1} v_i^\alpha(R) & = & \sum_{t=1}^\infty \{\sum_{s=1}^t v_i^{(s)}(R)\}\alpha^{t-1} \\ & = & \sum_{t=1}^{T^*} \{\sum_{s=1}^t v_i^{(s)}(R)\}\alpha^{t-1} + \sum_{t=T^*+1}^\infty \{\sum_{s=1}^t v_i^{(s)}(R)\}\alpha^{t-1}. \end{array}$$

Let $m = min_{1 \leq t \leq T^*} \sum_{s=1}^t v_i^{(s)}(R)$, then $(1-\alpha)^{-1} v_i^\alpha(R) > m \cdot \frac{1-\alpha^{T^*}}{1-\alpha} + M \cdot \frac{\alpha^{T^*}}{1-\alpha}$, i.e. $v_i^\alpha(R) > m \cdot (1-\alpha^{T^*}) + M \cdot \alpha^{T^*}$. For $\alpha \uparrow 1$, we have $m \cdot (1-\alpha^{T^*}) + M \cdot \alpha^{T^*} \to M$.

Hence, since M was arbitrarily chosen, $\lim_{\alpha\uparrow 1} v_i^\alpha(R) = +\infty = v_i(R)$.

Case 3: $v_i(R) = -\infty$.

The proof is similar to the proof of case 2 and left to the reader (see Exercise 4.3). □

Theorem 4.8

The following seven statements are equivalent:

(1) Every policy is transient.

(2) Every stationary policy is transient.

(3) Every deterministic policy is transient.

(4) $\rho\big(P(\pi)\big) < 1$ *for every stationary policy* π^∞.

(5) $\rho\big(P(f)\big) < 1$ *for every deterministic policy* f^∞.

(6) The linear program (4.15) with $r_i(a) = 1,\ (i,a) \in S \times A$ *has a finite optimum.*

(7) The MDP is contracting.

Proof

Obviously, (1) implies (2), and (2) implies (3). Let π^∞ be a stationary transient policy. Then, we have

$$\sum_{t=1}^\infty \mathbb{P}^\infty_{i,\pi}\{X_t = j, Y_t = a\} = \sum_{t=1}^\infty \{P^{t-1}(\pi)\}_{ij} \cdot \pi_{ja} < \infty < \infty \text{ for all } i, j \in S \text{ and } a \in A(j).$$

Hence, $\sum_{t=1}^\infty P^{t-1}(\pi)$ is convergent, which implies $\lim_{t\to\infty} P^t(\pi) = 0$, and - by Theorem 4.5 - $\rho\big(P(\pi)\big) < 1$. Similarly, it can be shown that if $\rho\big(P(\pi)\big) < 1$, then $\sum_{t=1}^\infty P^{t-1}(\pi)$ is convergent and π^∞ is a transient policy. Therefore, (2) is equivalent to (4) and also (3) and (5) are equivalent. Hence, it is sufficient to show that (3) implies (6), (6) implies (7) and (7) implies (1).

The proof that (3) implies (6):

Since there are transient deterministic policies, program (4.15) is feasible. Suppose that it has an infinite solution. Then, we know from the theory of linear programming that there exists a basis solution, which corresponds by Theorem 4.7 to a deterministic policy f^∞, such that in the simplex tableau the column of some nonbasic variable $x_k(a_k)$ has only nonpositive values. On the other hand, Theorem 4.7 implies that the exchange of the nonbasic variable $x_k(a_k)$ with the basic variable $x_k\big(f(k)\big)$ provides a feasible simplex tableau, i.e. the column of the nonbasic variable $x_k(a_k)$ contains a positive element in the row of the basic variable $x_k\big(f(k)\big)$: contradiction, which proves that (3) implies (6).

The proof that (6) implies (7):

Consider the dual program of (4.15) with $r_i(a) = 1$ for all $(i, a) \in S \times A$, which is

$$min\Big\{\sum_j \beta_j\mu_j \;\Big|\; \sum_j \{\delta_{ij} - p_{ij}(a)\}\mu_j \geq 1,\ (i,a) \in S \times A\Big\}. \tag{4.19}$$

This program has also a finite optimal solution, say μ. Let f^∞ be any deterministic transient policy. Since $\lim_{n\to\infty} P^n(f) = 0$, by Theorem 4.5, $\{I - P(f)\}$ is nonsingular with inverse $\sum_{n=0}^\infty P^n(f)$. From the constraints of (4.19), we obtain $\{I - P(f)\} \geq e$. Hence, $\mu = \{I - P(f)\}^{-1}e \geq e$, the last inequality because $\{I - P(f)\}^{-1}e = \sum_{n=0}^\infty P^n(f)e \geq P^0(f)e = e$. Define $\alpha := 1 - \{max_k\, \mu_k\}^{-1} \in [0, 1)$. Then, we have

$$\textstyle\sum_j p_{ij}(a)\mu_j \leq \mu_i - 1 \leq \mu_i - \frac{\mu_i}{max_k\, \mu_k} = \alpha \cdot \mu_i,\ (i,a) \in S \times A,$$

i.e. the MDP is contracting.

The proof that (7) implies (1):

From Corollary 1.1 it follows that it is sufficient to show that $\sum_{t=1}^\infty \mathbb{P}_{i,R}\{X_t = j\} < \infty$ for all $i, j \in S$ and all $R \in C(M)$. Take any $i, j \in S$ and any $R = (\pi^1, \pi^2, \dots) \in C(M)$. Since the MDP is contracting, there exist an $\alpha \in [0, 1)$ and $\mu \in \mathbb{R}^N$ with $\mu_i > 0,\ i \in S$, such that $P(\pi^t)\mu \leq \alpha \cdot \mu$ for all Markov decision rules π^t. Therefore, $P(\pi^1)P(\pi^2)\cdots P(\pi^t) \leq \alpha^t \cdot \mu$ for all $t \in \mathbb{N}$. Hence, we obtain

$$\textstyle\sum_{t=1}^\infty P(\pi^1)P(\pi^2)\cdots P(\pi^{t-1})\mu \leq (1-\alpha)^{-1} \cdot \mu.$$

Since $\mu_i > 0$ for every $i \in S$ and because

$$\textstyle\sum_{t=1}^\infty \mathbb{P}_{i,R}\{X_t = j\} = \sum_{t=1}^\infty \{P(\pi^1)P(\pi^2)\cdots P(\pi^{t-1})\}_{ij} < \infty,$$

it follows that R is a transient policy. □

Characterization (6) provides an algorithm for checking the contraction property of a given MDP model. Below we present this algorithm.

Algorithm 4.1 *Checking the contracting property of a substochastic MDP (LP approach)*

Input: Instance of an MDP (without immediate rewards) and a vector $\beta \in \mathbb{R}^N$ with $\beta_j > 0,\ 1 \leq j \leq N$.

Output: Decision whether or not this MDP is contracting.

1. Solve the linear program

$$max \left\{ \sum_{(i,a)} x_i(a) \;\middle|\; \begin{array}{rcl} \sum_{(i,a)} \{\delta_{ij} - p_{ij}(a)\} x_i(a) & = & \beta_j, \; j \in S \\ x_i(a) & \geq & 0, \; (i,a) \in S \times A \end{array} \right\}.$$

2. **if** the linear has a finite optimum **then** the MDP is contracting (STOP)

   else the MDP is not contracting (STOP).

Remark 1

If it happens in Algorithm 4.1 that the model is contracting, we obtain - as shown in the proof of Theorem 4.8 - from the dual program (4.19) a μ as optimal solution and α, where $\alpha := 1 - \{max_k \, \mu_k\}^{-1} \in [0,1)$, such that $\mu_i > 0$ for all $i \in S$ and $\sum_j p_{ij}(a)\mu_j \leq \alpha \cdot \mu_i$ for all $(i,a) \in S \times A$.

Remark 2

A discounted model is a contracting model with transition probabilities $p'_{ij}(a) := \alpha p_{ij}(a)$ for all $i,j \in S$ and all $a \in A(i)$ and with $\mu_i = 1$ for all $i \in S$. In fact the discounting and the contracting models are equivalent: a contracting substochastic model can be transformed into a stochastic discounted model such that for any policy R the total expected reward in the original model differs a multiplicative factor with the total discounted reward in the transformed model.

To prove this equivalence, we introduce the following transformed model $(\overline{S}, \overline{A}, \overline{p}, \overline{r})$:

$$\overline{S} = S \,\cup\, \{0\}; \; \overline{A}(i) = \left\{ \begin{array}{ll} A(i) & i \neq 0 \\ \{1\} & i = 0 \end{array} \right. ; \; \overline{r}_i(a) = \left\{ \begin{array}{ll} \frac{1}{\mu_i} r_i(a) & i \neq 0, \; a \in \overline{A}(i) \\ 0 & i = 0, \; a \in \overline{A}(i) \end{array} \right.$$

$$\overline{p}_{ij}(a) = \left\{ \begin{array}{ll} \frac{1}{\alpha\mu_i} p_{ij}(a)\mu_j & i \neq 0, \; j \neq 0, \; a \in \overline{A}(i) \\ 1 - \frac{1}{\alpha\mu_i} \sum_{k \in S} p_{ik}(a)\mu_k & i \neq 0, \; j = 0, \; a \in \overline{A}(i) \\ 1 & i = 0, \; j = 0, \; a \in \overline{A}(i) \\ 0 & i = 0, \; j \neq 0, \; a \in \overline{A}(i) \end{array} \right.$$

$$\begin{array}{lrcl} \text{For } i = 0\text{:} & \sum_{j \in \overline{S}} \overline{p}_{ij}(a) & = & \sum_{j \in \overline{S}} p_{0j}(1) = 1. \\ \text{For } i \neq 0\text{:} & \sum_{j \in \overline{S}} \overline{p}_{ij}(a) & = & \sum_{j \in S} \overline{p}_{ij}(a) + \overline{p}_{i0}(a) \\ & & = & \sum_{j \in S} \frac{1}{\alpha\mu_i} p_{ij}(a)\mu_j + \big\{1 - \frac{1}{\alpha\mu_i} \sum_{k \in S} p_{ik}(a)\mu_k\big\} = 1 \text{ for all } a \in \overline{A}(i). \end{array}$$

Hence, the transformed model is stochastic. In order to analyze the expected total (discounted) rewards, we may restrict ourselves to Markov policies. Let $R = (\pi^1, \pi^2, \dots)$ be a Markov policy. Then, by induction on t, it is straightforward to show that

$$\{\overline{P}(\pi^1)\overline{P}(\pi^2) \cdots \overline{P}(\pi^t)\}_{ij} = \left(\tfrac{1}{\alpha}\right)^t \cdot \tfrac{1}{\mu_i} \cdot \{P(\pi^1)P(\pi^2) \cdots P(\pi^t)\}_{ij} \cdot \mu_j \text{ for all } i,j \in S \text{ and } t \in \mathbb{N}.$$

Therefore, we can write,

$$\begin{array}{rcl} \overline{v}_i^\alpha(R) & = & \sum_{t=1}^\infty \alpha^{t-1} \cdot \{\overline{P}(\pi^1)\overline{P}(\pi^2) \cdots \overline{P}(\pi^{t-1})\overline{r}(\pi^t)\}_i \\ & = & \sum_{t=1}^\infty \alpha^{t-1} \cdot \sum_{j \in \overline{S}} \{\overline{P}(\pi^1)\overline{P}(\pi^2) \cdots \overline{P}(\pi^{t-1})\}_{ij} \cdot \overline{r}_j(\pi^t) \\ & = & \sum_{t=1}^\infty \alpha^{t-1} \cdot \sum_{j \in S} \{\overline{P}(\pi^1)\overline{P}(\pi^2) \cdots \overline{P}(\pi^{t-1})\}_{ij} \cdot \overline{r}_j(\pi^t) \\ & = & \sum_{t=1}^\infty \sum_{j \in S} \frac{1}{\mu_i} \cdot \{P(\pi^1)P(\pi^2) \cdots P(\pi^{t-1})\}_{ij} \cdot \mu_j \cdot \frac{1}{\mu_j} r_j(\pi^t) \\ & = & \frac{1}{\mu_i} \cdot \sum_{t=1}^\infty \sum_{j \in S} \{P(\pi^1)P(\pi^2) \cdots P(\pi^{t-1})\}_{ij} \cdot r_j(\pi^t) \\ & = & \frac{1}{\mu_i} \cdot \sum_{t=1}^\infty \{P(\pi^1)P(\pi^2) \cdots P(\pi^{t-1}) r(\pi^t)\}_i, \; i \in S \\ & = & \frac{1}{\mu_i} \cdot v_i(R), \; i \in S. \end{array}$$

Remark 3

For any $\alpha \in [0,1)$ we define the transition rates $p^{\alpha}_{ij}(a)$ by $p^{\alpha}_{ij}(a) := \alpha \cdot p_{ij}(a)$ for all $i,j \in S$ and $a \in A(i)$. Then, a policy R is normalized with respect to the transition rates $p_{ij}(a)$ if and only if R is transient for all $\alpha \in [0,1)$ with respect to the transition rates $p^{\alpha}_{ij}(a)$. Hence, an MDP is normalized if there exists a vector $\mu \in \mathbb{R}^N$ with $\mu_i > 0$ for all $i \in S$ which satisfies $\sum_j p_{ij}(a)\mu_j \leq \mu_i$ for all $(i,a) \in S \times A$, i.e. if the MDP is excessive. The reverse statement, that a normalized MDP is excessive, is not true in general. The above observations transforms Theorem 4.8 into the following result.

Theorem 4.9

The following seven statements are equivalent:

(1a) Every policy is normalized.

(2a) Every stationary policy is normalized.

(3a) Every deterministic policy is normalized.

(4a) $\rho\big(P(\pi)\big) \leq 1$ *for every stationary policy* π^∞.

(5a) $\rho\big(P(f)\big) \leq 1$ *for every deterministic policy* f^∞.

(6a) For all $\alpha \in [0,1)$ *the linear program (4.15) with* $p^{\alpha}_{ij}(a)$ *instead of* $p_{ij}(a)$ *for all* $i,j \in S$ *and* $a \in A(i)$ *and with* $r_i(a) = 1$ *for all* $(i,a) \in S \times A$ *has a finite optimum.*

(7a) The MDP is normalized.

Introduce the vector $y^N \in \mathbb{R}^N$ inductively by

$$\begin{cases} y^0_i := 1, \ i \in S \\ y^t_i := max_a \sum_j p_{ij}(a) y^{t-1}_j, \ i \in S, \ t = 1,2,\dots,N \end{cases} \tag{4.20}$$

Lemma 4.6

$y^t_i = sup_R \sum_j \mathbb{P}_{i,R}\{X_{t+1} = j\}$ *for all* $i \in S$ *and* $t = 0,1,\dots,N$.

Proof

From Corollary 1.1 it follows that it is sufficient to show that $y^t_i = sup_{R \in C(M)} \sum_j \mathbb{P}_{i,R}\{X_{t+1} = j\}$.

We apply induction on t. For $t = 0$: $\sum_j \mathbb{P}_{i,R}\{X_1 = j\} = \mathbb{P}_{i,R}\{X_1 = i\} = 1 = y^0_i, \ i \in S$.

Suppose that the result is correct for some t. Take any $i \in S$ and any policy $R = (\pi^1, \pi^2, \pi^3, \dots) \in C(M)$.

$\sum_j \mathbb{P}_{i,R}\{X_{t+2} = j\} = \sum_j \big\{ \sum_k p_{ik}(\pi^1) \mathbb{P}_{k,R'}\{X_{t+1} = j\}\big\} = \sum_k p_{ik}(\pi^1) \big\{ \sum_j \mathbb{P}_{k,R'}\{X_{t+1} = j\}\big\}$,

where $R' = (\pi^2, \pi^3, \dots)$. Hence, by the induction hypothesis,

$$\sum_j \mathbb{P}_{i,R}\{X_{t+2} = j\} \leq \sum_k p_{ik}(\pi^1) y^t_k \leq max_a \sum_k p_{ik}(a) y^t_k = y^{t+1}_i.$$

Hence, $sup_R \sum_j \mathbb{P}_{i,R}\{X_{t+1} = j\} \leq y^t_i$ for all $i \in S$ and $t = 0,1,\dots,N$.

On the other hand, also by induction on t, we show that $\sum_j \mathbb{P}_{i,R_t}\{X_{t+1} = j\} = y^t_i$ for some deterministic Markov policy $R_t = (f_t, f_{t-1}, \dots, f_1, f_1, \dots)$.

For $t = 0$ we have already shown above that $\sum_j \mathbb{P}_{i,R}\{X_1 = j\} = y^0_i = 1$ for any policy R.

Suppose that $\sum_k \mathbb{P}_{j,R_{t-1}}\{X_t = k\} = y^{t-1}_j$ for some deterministic Markov policy R_{t-1}. Let f_t be such that $y^t_i = \sum_j p_{ij}\big(f_t(i)\big) y^{t-1}_j, \ i \in S$. Then, with $R_t := (f_t, R_{t-1})$, we obtain

$$\begin{aligned} y^t_i &= \sum_j p_{ij}\big(f_t(i)\big) y^{t-1}_j \ = \ \sum_j p_{ij}\big(f_t(i)\big) \big\{ \sum_k \mathbb{P}_{j,R_{t-1}}\{X_t = k\}\big\} \\ &= \sum_k \sum_j p_{ij}\big(f_t(i) \mathbb{P}_{j,R_{t-1}}\{X_t = k\} \ = \ \sum_k \mathbb{P}_{i,R_t}\{X_{t+1} = k\}. \end{aligned}$$

□

Theorem 4.10

Consider a substochasttic MDP. Then, the seven statements of Theorem 4.8 are also equivalent tot the following seven statements:

(8) $\|y^N\|_\infty = max_{i \in S}\, y^N_i < 1$.

(9) $\|P(\pi^1)P(\pi^2)\cdots P(\pi^N)\|_\infty < 1$ *for every* $(\pi^1,\pi^2,\dots) \in C(M)$.

(10) $\|P^N(\pi)\|_\infty < 1$ *for every* $\pi^\infty \in C(S)$.

(11) $\|P^N(f)\|_\infty < 1$ *for every* $f^\infty \in C(D)$.

(12) $sup_R\, \rho(R) < 1$*, where* $\rho(R) := \limsup_{n\to\infty} \|P^n(R)\|_\infty^{1/n}$ *and* $\{P^n(R)\}_{ij} := \mathbb{P}_{i,R}\{X_{n+1} = j\}$.

(13) $sup_{\pi^\infty}\, \rho\big(P(\pi)\big) < 1$.

(14) $sup_{f^\infty}\, \rho\big(P(f)\big) < 1$.

Proof

The proof that (1) implies (8):

From the proof of Lemma 4.6 it follows that $y^N_i = max_{R \in C(DM)} \sum_j \mathbb{P}_{i,R}\{X_{N+1} = j\}$ for all $i \in S$, where $C(DM)$ is the finite set of deterministic Markov policies $R = (f_1, f_2, \dots, f_N)$. Take any $i \in S$ and any deterministic Markov policy $R = (f_1, f_2, \dots, f_N)$. Then, it is sufficient to show $\sum_j \mathbb{P}_{i,R}\{X_{N+1} = j\} < 1$. Add a state 0 with $A(0) := \{1\}$ and let $p_{k0}(a) := 1 - \sum_{j=1}^N p_{kj}(a)$ for all $(k,a) \in S \times A$, and let $p_{0j}(1) := 0$ for all $1 \le j \le N$ and $p_{00}(1) := 1$. Consider the extended model with state space $S^* := S \cup \{0\}$ and let R^* be the policy in the extended model which corresponds to R.

Define the following subsets of S^*:

$$T_1 := \{i\} \text{ and } T_k := \{j \in S^* \mid \mathbb{P}_{i,R^*}\{X_k = j\} > 0\} \text{ for } k = 2,3,\dots.$$

We first show three propositions.

Proposition 1:

If, for all $1 \le n \le N$, $0 \notin \cup_{l=1}^n T_l$ implies $T_{n+1} \not\subseteq \cup_{l=1}^n T_l$, then statement (8) is true.

The proof of Proposition 1:

Suppose $0 \notin \cup_{l=1}^N T_l$. Since the state 0 is absorbing, this implies that $0 \notin \cup_{l=1}^n T_l$ for $n = 1,2,\dots,N$. Then, by the assumption of the proposition, $\cup_{l=1}^{n+1} T_l$ has at least one state more than $\cup_{l=1}^n T_l$ for all $n = 1,2,\dots,N$. Consequently, $\cup_{l=1}^{N+1} T_l = S^*$ and $0 \in T_{N+1}$, i.e. $\mathbb{P}_{i,R^*}\{X_{N+1} = 0\} > 0$,and therefore $\sum_{j \in S} \mathbb{P}_{i,R}\{X_{N+1} = j\} < 1$. □

Proposition 2:

Let, for some $1 \le n \le N$, $0 \notin \cup_{l=1}^n T_l$ and $T_{n+1} \subseteq \cup_{l=1}^n T_l$. Let the deterministic f_*^∞ be such that

$$f_*(j) := \begin{cases} f_k(j) & \text{if } j \in T_k \backslash \cup_{l=1}^{k-1} T_l; \\ \text{arbitrarily chosen} & \text{if } j \notin \cup_{l=1}^n T_l. \end{cases}$$

Define $T_1^* := \{i\}$ and $T_k^* := \{j \in S^* \mid \mathbb{P}_{i,f_*^\infty}\{X_k = j\} > 0\}$ for $k = 2,3,\dots$.

Then, $T_k^* \subseteq \cup_{l=1}^n T_l$ for all $k = 1,2,\dots$.

The proof of Proposition 2:

The proof is by induction on k. For $k = 1$: $T_1^* = T_1 \subseteq \cup_{l=1}^n T_l$ for all $n \ge 1$. Suppose that $T_k^* \subseteq \cup_{l=1}^n T_l$ for $k = 1,2,\dots,m$. Take any $j \in T_{m+1}$. Then, there exists a state $s \in T_m^*$ such that $p_{sj}\big(f_*(s)\big) > 0$. Since $s \in \cup_{l=1}^n T_l$, we have $f_*(s) = f_k(s)$ for some k satisfying $s \in T_k \backslash \cup_{l=1}^{k-1} T_l$. Since $s \in T_k$ and $f_*(s) = f_k(s)$, we obtain $\mathbb{P}_{i,R^*}\{X_{k+1} = j\} \ge \mathbb{P}_{i,R^*}\{X_k = s\} \cdot p_{sj}\big(f_*(s)\big) > 0$. Hence, $j \in T_{k+1} \subseteq \cup_{l=1}^{n+1} T_l = \cup_{l=1}^n T_l$, which completes the proof that $T_{m+1}^* \subseteq \cup_{l=1}^n T_l$. □

Proposition 3:

Suppose that we have the same assumptions as in Proposition 2. Let f^∞ the policy in the substochastic model corresponding with policy f_*^∞ of the extended model, defined in Proposition 2. Then, f^∞ is a nontransient policy.

The proof of Proposition 3:

Since $0 \notin \cup_{l=1}^n T_l$ and $T_k^* \subseteq \cup_{l=1}^n T_l$ for all $k \in \mathbb{N}$, we have $\mathbb{P}_{i,f_*^\infty}\{X_k = 0\}$, $k \in \mathbb{N}$, and consequently, $\sum_{j \in S} \mathbb{P}_{i,f^\infty}\{X_k = j\} = 1$, $k \in \mathbb{N}$. Hence, $\sum_{t=1}^\infty \sum_{j \in S} \mathbb{P}_{i,f^\infty}\{X_t = j\} = +\infty$, implying that f^∞ is nontransient. □

We can complete the proof of statement (8) as follows. Since every policy is transient (by property (1)), the assumption of Proposition 2 is not valid (by Proposition 3). Hence, $0 \notin \cup_{l=1}^n T_l$ and at the same time $T_{n+1} \subseteq \cup_{l=1}^n T_l$ is impossible for all $1 \leq n \leq N$. Hence, $0 \notin \cup_{l=1}^n T_l$ implies $T_{n+1} \not\subseteq \cup_{l=1}^n T_l$, $1 \leq n \leq N$. Then, by Proposition 1, statement (8) holds. □

The proof that (8) and (9) are equivalent:

From the proof of Lemma 4.6 if follows that

$$y_i^N = max_{R \in C(DM)} \sum_j \mathbb{P}_{i,R}\{X_{N+1} = j\} = sup_{R \in C(M)} \sum_j \mathbb{P}_{i,R}\{X_{N+1} = j\} \text{ for all } i \in S.$$

From this property we obtain directly that (8) and (9) are equivalent. □

Further, it is obvious that (14) and (5) are equivalent, that (9) implies (10), (10) implies (11), and (13) implies (14). Notice that, by Theorem 4.4, (12) implies (13). Assume that (11) implies (5). Then, (8) implies (9), (9) implies (10), (11) implies (5) and (5) implies (8), the last implication because (5) and (1) are equivalent. Hence, assuming (11) implies (5) give the equivalence between the seven statements of Theorem 4.8 and (8), (9), (10) and (11). Furthermore, assume that (8) implies (12). Then, (8) implies (12), (12) implies (13), (13) implies (14), (14) implies (5) and (5) implies (8). Hence, assuming (11) implies (5) and (8) implies (12), the seven statements of Theorem 4.8 and the seven statements of Theorem 4.10 are all equivalent. Consequently, it is sufficient to show that (11) implies (5) and (8) implies (12).

The proof that (11) implies (5):

Take any $f \in C(D)$. Since $\|P^N(f)\|_\infty$ ¡ 1. we have - by Theorem 4.5, $\rho\big(P(f)\big) < 1$. □

The proof that (8) implies (12):

In Lemma 4.6 we have shown that for every $t \in \mathbb{N}$, every $i \in S$ and any policy R, we have

$$\sum_j \mathbb{P}_{i,R}\{X_{t+1} = j\} \leq y_i^t = \sum_j \{P(f_t)P(f_{t-1}) \cdots P(f_1)\}_{ij} \text{ for some policy } (f_t, f_{t-1} \cdots f_1) \in C(DM). \tag{4.21}$$

Since we have already shown that (8) implies (9), (9) implies (10), (10) implies (11) and (11) implies (5) and because (5) is equivalent to (7), (8) implies that the MDP is contracting. Hence, there exists a vector $\mu \in R^N$ with $_i > 0$ for all $i \in S$, and a scalar $\alpha \in [0,1)$ such that $\sum_j p_{ij}(a)\mu_j = \alpha \cdot \mu_i$ for all $(i,a) \in S \times A$. Therefore, for every $t \in N$, we have $P(f_t)P(f_{t-1}) \cdots P(f_1)\mu \leq \alpha^t \cdot \mu$, and consequently,

$$\lim\sup_{t \to \infty} \|P(f_t)P(f_{t-1}) \cdots P(f_1)\|_\mu^{1/t} \leq \alpha^t < 1.$$

Since we can show (see Exercise 4.5) that

$$\lim\sup_{t \to \infty} \|P(f_t)P(f_{t-1}) \cdots P(f_1)\|_\mu^{1/t} = \lim\sup_{t \to \infty} \|P(f_t)P(f_{t-1}) \cdots P(f_1)\|_\infty^{1/t}$$

Hence, we obtain, also using (4.21),

$$sup_R\, \rho(R) = sup_R \big\{ \lim\sup_{n \to \infty} \|P^n(R)\|_\infty^{1/n} \big\} \leq \lim\sup_{t \to \infty} \|P(f_t)P(f_{t-1}) \cdots P(f_1)\|_\infty^{1/t} < 1.$$

□

Characterization (8) provides a finite algorithm for checking the contraction property of a given substochastic MDP model. Below we present this algorithm.

Algorithm 4.2 *Checking the contracting property of a substochastic MDP (iterative approach)*
Input: Instance of an MDP (without immediate rewards).
Output: Decision whether or not this MDP is contracting.

```
1. $y_i^0 := 1$ for every $i \in S$; $t := 1$.
2. $y_i^t := max_a \sum_j p_{ij}(a) y_j^{t-1}$ for every $i \in S$.
3. if $max_i \, y_i^t < 1$ then the MDP is contracting (STOP)
   else go to step 4.
4. if $t = N$ then the MDP is not contracting (STOP)
   else begin $t := t + 1$; return to step 2 end
```

Theorem 4.11
Consider an excessive MDP. Then, the seven statements of Theorem 4.8 are also equivalent to the following two statements:
(15) $sup_R \sum_j \{P^{N+1}(R)\}_{ij} \mu_j < \mu_i$ *for all* $i \in S$.
(16) $max_{f^\infty \in C(D)} \|P^{N+1}(f)\|_\mu < 1$.

Proof
We will show that (15) implies (16), (16) implies (5) and (5) implies (15).
<u>The proof that (15) implies (16):</u>
Take any $f^\infty \in C(D)$. From (15) it follows that $\sum_j \{P^{N+1}(f)\}_{ij} \mu_j < \mu_i$ for all $i \in S$. Hence, we have $\|P^{N+1}(f)\|_\mu = max_i \, \mu^{-1} \sum_j \{P^{N+1}(f)\}_{ij} \mu_j < 1$. □
<u>The proof that (16) implies (5):</u>
Take any $f^\infty \in C(D)$. Since $\|P^{N+1}(f)\|_\mu < 1$, we have - by Theorem 4.5 - $\rho\big(P(f)\big) < 1$. Therefore $\rho\big(P(f)\big) < 1$ for every $f^\infty \in C(D)$. □
<u>The proof that (5) implies (15):</u>
Similar as in the proof of Lemma 4.6 it can be shown that for all $i \in S$ and all $t \in \mathbb{N}$,

$$sup_R \sum_j \{P^t(R)\}_{ij} \mu_j = sup_{R \in C(DM)} \sum_j \{P^t(R)\}_{ij} \mu_j$$

Therefore, it is sufficient to show that $sup_{R \in C(DM)} \sum_j \{P^{N+1}(R)\}_{ij} \mu_j < \mu_i$ for all $i \in S$. Take any state $i \in S$ and any deterministic Markov policy $R = (f_1, f_2, \dots, f_N)$. Then, it is sufficient to show that $\sum_j \{P^{N+1}(R)\}_{ij} \mu_j < \mu_i$. Add an absorbing state 0 with $A(0) := \{1\}$ and let $p_{k0}(a) := \mu_k - \sum_{j=1}^N p_{kj}(a) \mu_j$ for all $(k, a) \in S \times A$, and let $p_{0j}(1) := 0$ for all $1 \leq j \leq N$ and $p_{00}(1) := 1$. Given state k and action $a \in A(k)$ is chosen by policy R, notice that we have a positive transition probability to state 0 if and only if $\sum_j p_{kj}(a) \mu_j < \mu_i$. Consider the extended model with state space $S^* := S \cup \{0\}$ and let R^* be the policy in the extended model which corresponds to R.
Define the following subsets of S^*:

$$T_1 := \{i\} \text{ and } T_k := \{j \in S^* \mid \mathbb{P}_{i,R^*}\{X_k = j\} > 0\} \text{ for } k = 2, 3, \dots.$$

Similarly to the proof of Theorem 4.10 the following three propositions can be shown.

<u>Proposition 1:</u>
If, for all $1 \leq n \leq N$, $0 \notin \cup_{l=1}^n T_l$ implies $T_{n+1} \nsubseteq \cup_{l=1}^n T_l$, then statement (15) is true.

Proposition 2:

Let, for some $1 \leq n \leq N$, $0 \notin \cup_{l=1}^n T_l$ and $T_{n+1} \subseteq \cup_{l=1}^n T_l$. Let the deterministic f_*^∞ be such that

$$f_*(j) := \begin{cases} f_k(j) & \text{if } j \in T_k \setminus \cup_{l=1}^{k-1} T_l; \\ \text{arbitrarily chosen} & \text{if } j \notin \cup_{l=1}^n T_l. \end{cases}$$

Define $T_1^* := \{i\}$ and $T_k^* := \{j \in S^* \mid \mathbb{P}_{i,f_*^\infty}\{X_k = j\} > 0\}$ for $k = 2, 3, \dots$.

Then, $T_k^* \subseteq \cup_{l=1}^n T_l$ for all $k = 1, 2, \dots$.

Proposition 3:

Suppose that we have the same assumptions as in Proposition 2. Let f^∞ the policy in the substochastic model corresponding with policy f_*^∞ of the extended model, defined in Proposition 2. Then, f^∞ is a nontransient policy.

We can complete the proof of property (15) as follows. Notice that (5) is equivalent to the property that every policy is transient (see Theorem 4.8). Hence, by Proposition 3, the assumption of Proposition 2 is not valid, i.e. $0 \notin \cup_{l=1}^n T_l$ and at the same time $T_{n+1} \subseteq \cup_{l=1}^n T_l$ is impossible for all $1 \leq n \leq N$. Therefore, $0 \notin \cup_{l=1}^n T_l$ implies $T_{n+1} \nsubseteq \cup_{l=1}^n T_l$, $1 \leq n \leq N$. Then, by Proposition 1, statement (15) holds. □

Remark 4:

Testing the excessivity of a given MDP can be done in a finite way by checking the feasibility of the following system of linear inequalities

$$\begin{cases} \sum_j p_{ij}(a)\mu_j & \leq & \mu_i & \text{for all } (i,a) \in S \times A \\ \mu_i & \geq & 1 & \text{for all } i \in S \end{cases} \tag{4.22}$$

This feasibility test can be executed by the so-called phase I of the simplex method.

From the results in this section we derive the following properties:

(a) A stochastic MDP is substochastic (trivial).

(b) A discounted MDP is substochastic and contracting (trivial).

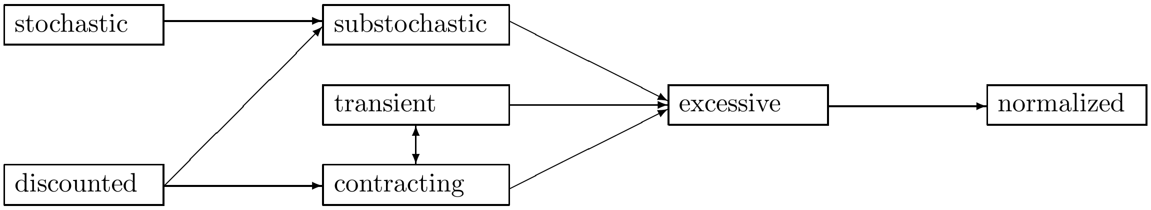
(c) A substochastic MDP is excessive (take $\mu_i = 1$ for all $i \in S$).

(d) Contracting and transient MDPs are equivalent (Theorem 4.8).

(e) A contracting and - by (d) also a transient - MDP are excessive (because $\sum_j p_{ij}(a)\mu_j \leq \alpha\mu_j \leq \mu_i$ for all $(i,a) \in S \times A$).

(f) An excessive MDP is normalized (see Remark 3).

The above properties are visualized in the following diagram:



4.5 The optimality equation

In this section we show that the regular value vector v satisfies, under certain conditions, the optimality equation

$$x_i = max_a \big\{r_i(a) + \sum_j p_{ij}(a)x_j\big\},\ i \in S. \tag{4.23}$$

Theorem 4.12

Suppose that the MDP is normalized and that every policy is regular. Then, there exists a regular optimal deterministic policy.

Proof

Since the MDP is normalized, we have - by Theorem 4.9 - $\rho\big(P(f)\big) \leq 1$ for every deterministic policy f^∞. Hence, $\rho\big(\alpha P(f)\big) \leq \alpha < 1$ for every $\alpha \in [0,1)$, and consequently (see Theorem 4.5) $I - \alpha P(f)$ is nonsingular and $v^\alpha(f^\infty) = \sum_{t=1}^\infty \alpha^{t-1}P^{t-1}(f)r(f) = \{I - \alpha P(f)\}^{-1}r(f)$ for every deterministic policy f^∞ and every $\alpha \in [0,1)$. Since, for any $f^\infty \in C(D)$, $v^\alpha(f^\infty)$ is the unique solution of the linear system $\{I - \alpha P(f)\}x = r(f)$, by Cramer's rule, $v_i^\alpha(f^\infty)$ is a rational function in α for each component i.

Suppose there is no deterministic Blackwell optimal policy, where Blackwell optimality means discount optimality for all discount factors $\alpha \in [\alpha_0, 1)$ for some α_0. In a normalized MDP, for each $\alpha \in [0,1)$ there exists a discounted optimal deterministic policy (the proof is similar the the proof given in Chapter 3 for stochastic MDPs). Hence, there is a sequence $\{\alpha_k,\ k = 1,2,\dots\}$ and a sequence $\{f_k,\ k = 1,2,\dots\}$ such that $\alpha_k \uparrow 1$ and $v^\alpha = v^\alpha(f_k^\infty) > v^\alpha(f_{k-1}^\infty)$ for $\alpha = \alpha_k,\ k = 2,3,\dots$.

Since $C(D)$ is finite, there are different policies f^∞ and g^∞ such that for any increasing subsequence $\alpha_{k_n},\ n = 1,2,\dots$ with $lim_{n\to\infty}\, \alpha_{k_n} = 1$, we have

$$\begin{cases} v^\alpha(f^\infty) > v^\alpha(g^\infty) & \text{for } \alpha = \alpha_{k_1}, \alpha_{k_3}, \dots \\ v^\alpha(f^\infty) < v^\alpha(g^\infty) & \text{for } \alpha = \alpha_{k_2}, \alpha_{k_4}, \dots \end{cases} \tag{4.24}$$

Let $h(\alpha) := v^\alpha(f^\infty) - v^\alpha(g^\infty)$. Then, for each $i \in S$ the function $h_i(\alpha)$ is a continuous rational function in α on $[0,1)$. From (4.24) it follows that $h_i(\alpha)$ has an infinite number of zeros, which contradicts the rationality of $h_i(\alpha)$, unless $h_i(\alpha) \equiv 0$. Hence, there exists a Blackwell optimal policy $f_0^\infty \in C(D)$. Furthermore, by Lemma 4.5,

$$v_i(f_0^\infty) = \lim_{\alpha\uparrow 1}\, v_i^\alpha(f_0^\infty) \geq \lim_{\alpha\uparrow 1}\, v_i^\alpha(R) = v_i(R)$$

for all regular policies R, i.e. f_0^∞ is a regular optimal deterministic policy. □

For any $c \in [-\infty, +\infty]$ we define $0 \cdot c := c \cdot 0 := 0$. We call an N-dimensional vector x with components $x_i \in [-\infty, +\infty]$ $p-summable$ if $\sum_j p_{ij}(a)x_j$ is well defined for all $(i,a) \in S \times A$, i.e. not both of the values $+\infty$ and $-\infty$ may occur in the summation. The next example shows that the solution of the optimality equation not p-summable, in general.

Example 4.3

$S = \{1,2,3\};\ A(1) = A(2) = A(3) = \{1\};\ p_{11}(1) = 0,\ p_{12}(1) = \frac{1}{2},\ p_{13}(1) = \frac{1}{2};\ p_{21}(1) = 0,\ p_{22}(1) = 1,\ p_{23}(1) = 0;\ p_{31}(1) = 0,\ p_{32}(1) = 0,\ p_{33}(1) = 1;\ r_1(1) = 0;\ r_2(1) = 2;\ r_3(1) = -1.$
This is a stochastic MDP. The optimality equation is:

$$v_1 = \tfrac{1}{2}v_2 + \tfrac{1}{2}v_3;\ v_2 = 2 + v_2;\ v_3 = -1 + v_3.$$

Hence, in the extended real space $[-\infty, +\infty]$, we obtain $v_2 = +\infty,\ v_3 = -\infty$, but v_1 is undefined because $\frac{1}{2} \cdot (+\infty) + \frac{1}{2} \cdot (-\infty)$ is not defined.

Theorem 4.13

If there exists a regular optimal deterministic policy and if the regular value vector v is p-summable, then v satisfies the optimality equation $x_i = max_a \left\{r_i(a) + \sum_j p_{ij}(a)x_j\right\},\ i \in S$.

Proof

Let f^∞ be a regular optimal deterministic policy, i.e. $v = v(f^\infty)$. Since v is p-summable, we may write

$$v_i = v_i(f^\infty) = r_i(f) + \sum_j p_{ij}(f)v_j(f^\infty) = max_a \left\{r_i(a) + \sum_j p_{ij}(a)v_j(f^\infty)\right\},\ i \in S. \tag{4.25}$$

Let $a_i \in A(i),\ i \in S$, be such that $r_i(a_i) + \sum_j p_{ij}(a_i)v_j = max_a \left\{r_i(a) + \sum_j p_{ij}(a)v_j\right\},\ i \in S$.

Take policy $R = (\pi^1, \pi^2, , \dots,) \in C(M)$ such that $\pi^1_{ia} = \begin{cases} 1,\ a = a_i \\ 0,\ a \neq a_i \end{cases}$; $\pi^t_{ia} = \begin{cases} 1,\ a = f(i) \\ 0,\ a \neq f(i) \end{cases}$ $t \geq 2$.

Since $v_i(R) = r_i(a_i) + \sum_j p_{ij}(a_i)v_j(f^\infty) = r_i(a_i) + \sum_j p_{ij}(a_i)v_j,\ i \in S$, R is a regular policy. Furthermore, we can write

$$v_i \geq v_i(R) = r_i(a_i) + \sum_j p_{ij}(a_i)v_j = max_a \left\{r_i(a) + \sum_j p_{ij}(a)v_j\right\},\ i \in S. \tag{4.26}$$

From (4.25), (4.26) and $v_i = v_i(f^\infty)$ it follows that $v_i = max_a \left\{r_i(a) + \sum_j p_{ij}(a)v_j\right\},\ i \in S$. □

Corollary 4.1

For any $g^\infty \in C(D)$ such that $v(g^\infty)$ is p-summable, the total expected reward $v(g^\infty)$ satisfies the equation $x = r(g) + P(g)x$.

Proof

Take any $g^\infty \in C(D)$ such that $v(g^\infty)$ is p-summable. Since $v(g^\infty)$ is p-summable, the total expected reward for any starting state i, which is denoted by $v_i(g^\infty)$, satisfies $v_i(g^\infty) = r_i(g) + \sum_j p_{ij}(g)v_j(g^\infty)$ and is well defined. Hence, g^∞ is a regular policy.

Consider the MDP model with in state i only action $g(i),\ i \in S$. Since this model has only one policy, namely g^∞, which is regular. this policy is a deterministic regular optimal policy. By applying Theorem 4.13 to this model, we obtain $v(g^\infty) = v = r(g) + P(g)v = r(g) + P(g)v(g^\infty)$. □

Remark:

Unfortunately, in contrast to discounted models, the solution of the optimality equation is not unique, in general. For instance, in a stochastic MDP, i.e. $\sum_j p_{ij}(a) = 1$ for all $(i,a) \in S \times A$, if x is a solution of the optimality equation also $x + c \cdot e$ is a solution for any scalar c. The reason for the nonuniqueness is that the mapping $L_f x := r(f) + P(f)x$ and the mapping $(Ux)_i := max_a\{r_i(a) + \sum_j p_{ij}(a)x_j\},\ i \in S$, are no contractions (the monotonicity holds for L_f and U and is easy to verify).

The next example shows that the functional equation does not have a unique solution in a normalized MDP which has a transient policy.

Example 4.4

$S = \{1\};\ A(1) = \{1,2\};\ p_{11}(1) = 1,\ p_{11}(2) = \frac{1}{2};\ r_1(1) = 0;\ r_1(2) = -1$.

This is a normalized and regular MDP, and f_1^∞ with $f_1(1) = 1$ is a deterministic normalized and regular optimal policy. Therefore, $v_1 = z_1 = 0$.

Notice that a policy is transient if and only if the action $a = 2$ is chosen infinitely often with posive probability, in which case the total expected reward is equal to -2. Hence, $w_1 = -2$.

The functional equation of this MDP is: $x_1 = max\{0 + x_1, -1 + \frac{1}{2}x_1\}$ with solution set $\{x_1 \mid x_1 \geq -2\}$. Hence, both $v = z = 0$ and $w = -2$ are solutions of the optimality equation.

A policy $f^\infty \in C(D)$ is called *conserving* if $r(f) + P(f)v = v$. An optimal policy is always conserving (see Corollary 4.1), but the reverse statement is not true as the next example shows.

Example 4.5
$S = \{1, 2\};\ A(1) = \{1, 2\};\ A(2) = \{1\};\ p_{11}(1) = 1,\ p_{12}(1) = 0;\ p_{11}(2) = 0,\ p_{12}(2) = 1;\ p_{21}(1) = 0,$ $p_{22}(1) = 0;\ r_1(1) = 0;\ r_1(2) = 2;\ r_2(1) = -1.$
This is a substochastic and regular MDP, and it is easy to verify that $v = (1, -1)$.
The policy f_1^∞ with $f_1(1) = 2$ and $f_1(2) = 1$ is a deterministic normalized and regular optimal policy, which is also a conserving policy. The policy f_2^∞ with $f_2(1) = 1$ and $f_2(2) = 1$ is conserving, because $r(f_2) + P(f_2)v = \binom{0}{-1} + \left(\begin{smallmatrix}1 & 0\\ 0 & 0\end{smallmatrix}\right)\binom{0}{-1} = \binom{1}{-1} = v$. However, $v(f_2^\infty) = \binom{0}{-1}$, so $v(f_2^\infty)$ is not optimal.

4.6 Optimal transient policies

In this section we discuss the problem of finding an optimal policy within the set of transient policies, i.e. a policy R_* such that

$$v_i(R_*) = sup\,\{v_i(R) \mid R \text{ is a transient policy}\} = w_i,\ i \in S. \tag{4.27}$$

Throughout this section we assume the following.

Assumption 4.1
(1) The model is normalized and every policy is regular.
(2) There exists at least one transient policy.

The next theorem shows how the existence of a transient policy can be verified.

Theorem 4.14
There exists a transient policy if and only if the linear program

$$max\left\{\sum_{(i,a)} x_i(a) \;\middle|\; \begin{array}{rcl} \sum_{(i,a)} \{\delta_{ij} - p_{ij}(a)\} x_i(a) & = & \beta_j,\ j \in S \\ x_i(a) & \geq & 0,\ (i,a) \in S \times A \end{array} \right\}. \tag{4.28}$$

where $\beta_j > 0,\ j \in S$, are some given numbers, has a feasible solution.

Proof
Let x be a feasible solution of the linear program (4.28). Then, by Theorem 4.7, $\pi^\infty(x)$ defined by (4.16) is a transient policy. Conversely, let R be a transient policy. Define $x(R)$ by

$$x_{ia}(R) := \sum_{t=1}^{\infty} \sum_k \beta_k \cdot \mathbb{P}_{k,R}\,\{X_t = i,\ Y_t = a\},\ (i,a) \in S \times A. \tag{4.29}$$

Since R is transient, $x_{ja}(R)$ is well defined and finite for all (j, a). Furthermore, by Corollary 1.1, we may assume that R is a Markov policy, say $R = (\pi_1, \pi_2, \ldots)$. Then, we obtain

$$\begin{aligned}
\textstyle\sum_{(i,a)} \{\delta_{ij} - p_{ij}(a)\} x_{ia}(R) &= \textstyle\sum_{(i,a)} \{\delta_{ij} - p_{ij}(a)\} \cdot \lim_{n\to\infty} \sum_{t=1}^n \sum_k \beta_k \cdot \mathbb{P}_{k,R}\,\{X_t = i,\ Y_t = a\} \\
&= \textstyle\sum_{(i,a)} \{\delta_{ij} - p_{ij}(a)\} \cdot \sum_k \beta_k \cdot \lim_{n\to\infty} \sum_{t=1}^n \left\{P(\pi^1) \cdots P(\pi^{t-1})\right\}_{ki} \cdot \pi_{ia}^t \\
&= \textstyle\sum_k \beta_k \cdot \lim_{n\to\infty} \sum_{t=1}^n \left\{P(\pi^1) \cdots P(\pi^{t-1})\right\}_{ki} \cdot \left\{I - P(\pi^t)\right\}_{ij} \\
&= \textstyle\sum_k \beta_k \cdot lim_{n\to\infty} \left\{I - P(\pi^1) \cdots P(\pi^n)\right\}_{kj} \\
&= \textstyle\sum_k \beta_k \cdot \delta_{kj} = \beta_j,\ j \in S.
\end{aligned}$$

□

Theorem 4.15

There exists a transient deterministic policy.

Proof

Since for the concept of a transient policy the rewards are not important, we may assume that $r_i(a) = -1$ for all $(i,a) \in S \times A$. Let R_* be a transient policy, i.e. $\sum_{t=1}^\infty \mathbb{P}_{i,R_*}\{X_t = j,\ Y_t = a\} < \infty$ for all $i \in S$ and all $(j,a) \in S \times A$. Hence,

$$0 \geq v_i(R_*) = \sum_{t=1}^\infty \sum_{j,a} \mathbb{P}_{i,R_*}\{X_t = j,\ Y_t = a\} \cdot (-1) > -\infty,\ i \in S.$$

Because $v_i = sup_R\, v_i(R),\ i \in S$, we also have $-\infty < v_i \leq 0,\ i \in S$. Since, by Theorem 4.12, there exists a deterministic policy f^∞ such that $v = v(f^\infty)$, we can write

$$-\infty < v_i(f^\infty) = \sum_{t=1}^\infty \sum_{j,a} \mathbb{P}_{i,f^\infty}\{X_t = j,\ Y_t = a\} \cdot (-1) \leq 0,\ i \in S.$$

Consequently,

$$\sum_{t=1}^\infty \mathbb{P}_{i,f^\infty}\{X_t = j,\ Y_t = a\} < +\infty \text{ for all } i \in S \text{ and all } (j,a) \in S \times A,$$

i.e. f^∞ is a transient deterministic policy. □

Although $v_i(R)$ is finite for every transient policy and for every $i \in S$, the transient value vector w has not necessarily finite components w_i for all $i \in S$, as the next example shows.

Example 4.6

$S = \{1,2\};\ A(1) = \{1,2\};\ A(2) = \{1\};\ p_{11}(1) = 1,\ p_{12}(1) = 0;\ p_{11}(2) = 0,\ p_{12}(2) = 1;\ p_{21}(1) = 0,$ $p_{22}(2) = \frac{1}{2};\ r_1(1) = 1;\ r_1(2) = 1;\ r_2(1) = 1.$

This is a normalized and regular MDP, and f^∞ with $f(1) = 2,\ f(2) = 1$ is a transient policy. Hence, this MDP satisfies Assumption 4.1.

Consider the sequence $\big\{\pi^\infty(n)\big\}_{n=1}^\infty$ of stationary policies, defined by $\pi_{ia}(n) := \begin{cases} 1 - \frac{1}{n} & \text{if } a = 1; \\ \frac{1}{n} & \text{if } a = 2. \end{cases}$

Since $v\big(\pi^\infty(n)\big)$ satisfies the equation $\begin{cases} x_1 &= 1 + (1 - \frac{1}{n})x_1 + \frac{1}{n}x_2 \\ x_2 &= 1 + \frac{1}{2}x_2 \end{cases}$, we obtain

$v_1\big(\pi^\infty(n)\big) = n + 2,\ v_2\big(\pi^\infty(n)\big) = 2$. Since $r_j(a) = 1$ for every $(j,a) \in S \times A$, we have

$$\begin{cases} n + 2 &= v_1\big(\pi^\infty(n)\big) &= \sum_{t=1}^\infty \sum_{j,a} \mathbb{P}_{i,\pi^\infty(n)}\{X_t = j,\ Y_t = a\} \\ 2 &= v_2\big(\pi^\infty(n)\big) &= \sum_{t=1}^\infty \sum_{j,a} \mathbb{P}_{i,\pi^\infty(n)}\{X_t = j,\ Y_t = a\} \end{cases}$$

Since these values are finite for every n, every policy $\pi^\infty(n)$ is transient. However, the transient value vector w has not finite components, because $w_1 \geq sup_n\, v_1\big(\pi^\infty(n)\big) = +\infty$.

Theorem 4.16

If all components of the transient value vector w are finite, then w is the solution of the functional equation $x_i = max_a\big\{r_i(a) + \sum_j p_{ij}(a)x_j\big\},\ i \in S$.

Proof

Let $R_1 := (\pi_1, \pi_2, \dots)$ be an arbitrary transient Markov policy. Then, $v(R_1) = r(\pi_1) + P(\pi_1)u(R_1)$, where $u_j(R_1)$ represents the expected total reward earned from time point 2, given that the state at time 2 is j. Let $R_2 := (\pi_2, \pi_3, \dots)$, then we have $u_j(R_1) = v_j(R_2)$ for any state j such that $p_{ij}(\pi_1) > 0$ for some $i \in S$.

If the stochastic process induced by R_2 starts in a state j for which $p_{ij}(\pi_1) > 0$ for some $i \in S$, then this process is also transient. Hence, we can write

$$\begin{aligned} v_i(R_1) &= r_i(\pi_1) + \textstyle\sum_j p_{ij}(\pi_1)u_j(R_1) \;=\; r_i(\pi_1) + \sum_j p_{ij}(\pi_1)v_j(R_2) \\ &\le r_i(\pi_1) + \textstyle\sum_j p_{ij}(\pi_1)w_j \le max_a \big\{r_i(a) + \sum_j p_{ij}(a)w_j\big\},\ i \in S. \end{aligned}$$

Therefore,

$$w_i \le max_a \big\{r_i(a) + \sum_j p_{ij}(a)w_j\big\},\ i \in S. \tag{4.30}$$

Take any $\varepsilon > 0$. Suppose that for every $j \in S$, the policy $R(j)$ is a transient policy which satisfies $v_j\big(R(j)\big) \ge w_j - \varepsilon$. Take $a_i \in A(i)$ such that $r_i(a_i) + \sum_j p_{ij}(a_i)w_j = max_a \big\{r_i(a) + \sum_j p_{ij}(a)w_j\big\},\ i \in S$. Let R_3 be the policy that chooses at time 1 action a_i for initial sate i, and then follows $R(j)$, when the state at time 2 is state j. The stochastic process induced by $R(j)$ is considered as starting in state j. Hence, policy R_3 is also transient and we obtain

$$\begin{aligned} w_i &\ge v_i(R_3) \;=\; r_i(a_i) + \textstyle\sum_j p_{ij}(a_i)v_j\big(R(j)\big) \;\ge\; r_i(a_i) + \sum_j p_{ij}(a_i)(w_j - \varepsilon) \\ &= max_a \big\{r_i(a) + \textstyle\sum_j p_{ij}(a)w_j\big\} - \varepsilon \cdot \sum_j p_{ij}(a_i) \\ &\ge max_a \big\{r_i(a) + \textstyle\sum_j p_{ij}(a)w_j\big\} - \varepsilon \cdot p, \end{aligned}$$

where $p := max_{(i,a)} \sum_j p_{ij}(a)$. Since ε is arbitrarily chosen, we have

$$w_i \ge max_a \big\{r_i(a) + \sum_j p_{ij}(a)w_j\big\},\ i \in S. \tag{4.31}$$

From (4.28) and (4.29), it follows that $w_i = max_a \big\{r_i(a) + \sum_j p_{ij}(a)w_j\big\},\ i \in S.$ □

In the context of this section we say that a (finite) vector $z \in \mathbb{R}^N$ is superharmonic if for all pairs $(i,a) \in S \times A$, we have $z_i \ge r_i(a) + \sum_j p_{ij}(a)z_j$.

Theorem 4.17
If the transient value vector w is finite, then w is the (componentwise) smallest superharmonic vector.

Proof
Theorem 4.16 implies that w is superharmonic. Suppose that z is also superharmonic. From Theorem 1.1 it follows that it is sufficient to show that $z \ge v(R)$ for every transient Markov policy R. Let $R = (\pi_1, \pi_2, \dots)$ be an arbitrary transient Markov policy. Since z is superharmonic, we have $z \ge r(\pi^t) + P(\pi^t)z$ for all $t \in \mathbb{N}$. By iterating this inequality, we obtain

$$z \ge \textstyle\sum_{t=1}^T P(\pi^1)P(\pi^2)\cdots P(\pi^{t-1})r(\pi^t) + P(\pi^1)P(\pi^2)\cdots P(\pi^T)z \text{ for all } T \in \mathbb{N}.$$

Because R is a transient policy, we have $P(\pi^1)P(\pi^2)\cdots P(\pi^T) \to 0$ for $T \to \infty$.
Since $v(R) = \lim_{T\to\infty} \sum_{t=1}^T P(\pi^1)P(\pi^2)\cdots P(\pi^{t-1})r(\pi^t)$, we obtain $z \ge v(R)$. □

Theorem 4.17 implies that, if the transient value vector w is finite, w is the unique optimal solution of the linear program

$$min\Big\{\sum_j \beta_j z_j \;\Big|\; \sum_j \{\delta_{ij} - p_{ij}(a)\}z_j \ge r_i(a),\ (i,a) \in S \times A\Big\}, \tag{4.32}$$

where β is any vector in $\mathbb{R}^N$ that satisfies $\beta_j > 0$ for every $j \in S$. The dual program of (4.32) is program (4.15). In Section 4.3, we have shown in Theorem 4.7 that the mapping (4.18) is a bijection between the set of transient stationary policies and the set of feasible solutions of (4.15) with (4.16) as the inverse mapping. Furthermore, the set of transient deterministic policies corresponds to the set of extreme feasible solutions of (4.15).

Theorem 4.18

(1) If program (4.15) is infeasible, then there does not exist a transient policy.

(2) If program (4.15) has an infinite solution, then there does not exist a transient optimal policy.

(3) If x^* is an extreme optimal solution of program (4.15), then the deterministic policy f_*^∞ such that $x_j^*\big(f_*(j)\big) > 0,\ j \in S$, is a transient optimal policy.

Proof

(1) This property is a consequence of Theorem 4.14.

(2) Suppose that there exist a transient optimal policy. Then, w is finite and, by Theorem 4.17, (4.32) has a finite optimal solution. Hence, (4.15) - the dual program of (4.32) - has also a finite optimal solution, which provides a contradiction.

(3) Let x^* be an extreme optimal solution of program (4.15). Then, x^* has at most N positive components. Since $\sum_a x_j^*(a) = \beta_j + \sum_{(i,a)} p_{ij}(a)x_i^*(a) \ge \beta_j > 0,\ j \in S$, for each $j \in S$, $x_j^*(a) > 0$ for exactly one action $a \in A(j)$. From Theorem 4.7 it follows that f_*^∞ is a transient policy. By the complementary slackness property of linear programming, we obtain $\{I - P(f_*)\}w = r(f_*)$. Hence, we can write $w = \{I - P(f_*)\}^{-1}r(f_*) = v(f_*^\infty)$, i.e. f_*^∞ is a transient optimal policy. □

The above results give rise to the following algorithm which determines an optimal policy in the set of transient policies, if such policy exists. Otherwise, the algorithm concludes either that no transient policy exists or that no optimal policy exists in the class of transient policies.

Algorithm 4.3 *Linear programming algorithm to find an optimal transient policy.*

Input: Instance of an MDP which satisfies Assumption 4.1.

Output: A policy f_*^∞ witch is optimal in the set of transient policies, if such policy exists, and the transient value vector w; otherwise, we find either that there exists no transient policy or that there exists no optimal policy in the set of teansient policies.

1. Take any vector β with $\beta_j > 0$ for every $j \in S$.

2. Use the simplex method to solve the dual pair of linear programs:

$$min \left\{ \sum_j \beta_j z_j \;\middle|\; \sum_j \{\delta_{ij} - p_{ij}(a)\}z_j \ge r_i(a),\ (i,a) \in S \times A \right\}$$

and

$$max \left\{ \sum_{(i,a)} r_i(a)x_i(a) \;\middle|\; \begin{array}{rl} \sum_{(i,a)} \{\delta_{ij} - p_{ij}(a)\}x_i(a) & = \beta_j,\ j \in S \\ x_i(a) & \ge 0,\ (i,a) \in S \times A \end{array} \right\}.$$

3. **if** the second linear program is infeasible **then** there exists no transient policy (STOP);

4. **if** the second linear program has an infinite solution **then** there exists no optimal policy in the set of transient policies (STOP);

5. **if** the second linear program has a finite optimal extreme solution, say x^*, **then** the first linear program has also a finite optimal solution, say z^*, and **go to** step 6.

6. $w := z^*$ is the transient vector value and $f_*^\infty \in C(D)$ such that $x_j^*(f_*(j)) > 0$ for every $j \in S$, is an optimal policy in the set of transient policies (STOP).

Since the simplex method is used to solve the linear program (4.15), we obtain an extreme optimal solution x^*. The next example shows that if in step 5 of the algorithm the property 'extreme' is necessary: if x^* is any optimal solution, then the corresponding policy is not necessarily optimal in the set of transient policies.

Example 4.7
$S = \{1,2\};\ A(1) = \{1,2\};\ A(2) = \{1\};\ p_{11}(1) = 1,\ p_{12}(1) = 0;\ p_{11}(2) = 0,\ p_{12}(2) = 1;\ p_{21}(1) = 0,$ $p_{22}(2) = \frac{1}{2};\ r_1(1) = 1;\ r_1(2) = 1;\ r_2(1) = 1$. Take $\beta_1 = \beta_2 = \frac{1}{2}$.
Notice that this MDP satisfies Assumption 4.1 and that the transient value vector $w = (3,2)$.
The linear program becomes:

$$max \left\{ x_1(2) + x_2(1) \;\middle|\; \begin{array}{rcrcl} x_1(2) & & & = & \frac{1}{2} \\ -x_1(2) & + & \frac{1}{2}x_2(1) & = & \frac{1}{2} \\ x_1(1), & x_1(2), & x_2(1) & \geq & 0 \end{array} \right\}.$$

The optimal solutions are $x_1(1) \geq 0,\ x_1(2) = \frac{1}{2},\ x_2(1) = 2$. The only extreme optimal solution of this program is $x_1^*(1) = 0,\ x_1^*(2) = \frac{1}{2},\ x_2^*(1) = 2$ with corresponding policy $f_*(1) = 2,\ f_*(2) = 1$, which is an optimal policy in the set of transient policies. Another optimal solution is $x_1(1) = 1,\ x_1(2) = \frac{1}{2},\ x_2(1) = 2$ with a possible corresponding policy $f(1) = 1,\ f(2) = 1$, which is not a transient policy.

The following example shows that a policy f_*^∞, obtained by Algorithm 4.3, is in general not optimal in the set of all policies.

Example 4.8
$S = \{1,2\};\ A(1) = A(2) = \{1,2\};\ p_{11}(1) = \frac{1}{2},\ p_{12}(1) = 0;\ p_{11}(2) = 0,\ p_{12}(2) = 1;\ p_{21}(1) = 0,\ p_{22}(2) = \frac{1}{2};$ $p_{21}(2) = 1,\ p_{22}(2) = 0;\ r_1(1) = -1;\ r_1(2) = 1;\ r_2(1) = 1$. Take $\beta_1 = \beta_2 = \frac{1}{2}$.
Notice that this MDP satisfies Assumption 4.1. The linear program becomes:

$$max \left\{ -x_1(1) - x_2(1) \;\middle|\; \begin{array}{rcrcrcrcl} \frac{1}{2}x_1(1) & + & x_1(2) & & & - & x_2(2) & = & \frac{1}{2} \\ & - & x_1(2) & + & \frac{1}{2}x_2(1) & + & x_2(2) & = & \frac{1}{2} \\ x_1(1), & & x_1(2), & & x_2(1), & & x_2(2) & \geq & 0 \end{array} \right\}.$$

An extreme optimal solution is: $x_1^*(1) = 0,\ x_1^*(2) = \frac{1}{2},\ x_2^*(1) = 2,\ x_2^*(2) = 0$. The corresponding policy is: $f_*(1) = 2,\ f_*(2) = 1$ with $v_1(f_*^\infty) = v_2(f_*^\infty) = w_1 = w_2 = -2$. This policy is not optimal in the set of all policies, because for the policy $f^\infty \in C(D)$ with f(1) = f(2) = 2, we have $v_1(f^\infty) = v_2(f^\infty) = 0$. This is a better, in fact an optimal, policy. However, this policy is not transient.

Theorem 4.19
Assume that the linear program (4.15) has a finite optimal solution. Then, the correspondence between the transient stationary policies and the feasible solutions of linear program (4.15) preserves the optimality property, i.e.

(1) *If π^∞ is a stationary transient policy which is optimal in the set of transient policies, then $x(\pi)$ is an optimal solution of the linear program (4.15).*

(2) *If x is an optimal solution of the linear program (4.15), then the stationary transient policy $\pi^\infty(x)$ is optimal in the set of transient policies.*

Proof

1. Since the transient value vector w is an optimal solution of (4.32) and $x(\pi)$ is feasible for (4.15), the dual program of (4.32), it follows that it is sufficient to verify that $\sum_{(i,a)} r_i(a)\, x_{ia}(\pi) = \sum_j \beta_j w_j$. Indeed, we can write

$$\begin{array}{rcl} \sum_{(i,a)} r_i(a)\, x_{ia}(\pi) & = & \sum_{(i,a)} r_i(a) \left\{\beta^T \big(I - P(\pi)\big)^{-1}\right\}_i \cdot \pi_{ia} \\ & = & \beta^T \big(I - P(\pi)\big)^{-1} r(\pi) = \beta^T v(\pi^\infty) = \beta^T w. \end{array}$$

2. We have,

$$\begin{array}{rcl} \beta^T v\big(\pi^\infty(x)\big) & = & \beta^T \big\{I - P\big(\pi(x)\big)\big\}^{-1} r\big(\pi(x)\big) \\ & = & \sum_{(i,a)} r_i(a)\, x_{ia}\big(\pi(x)\big) = \sum_{(i,a)} r_i(a)\, x_i(a) = \beta^T w. \end{array}$$

Since $\beta_j > 0,\ j \in S$, and $v\big(\pi^\infty(x)\big) \leq w$, it follows that $v\big(\pi^\infty(x)\big) = w$, i.e. $\pi^\infty(x)$ is optimal in the set of transient policies. □

4.7 The contracting model

Throughout this section we assume that the model is contracting, i.e. there exists a vector $\mu \in R^N$ with $\mu_i > 0$ for all $i \in S$, and a scalar $\alpha \in [0, 1)$ such that $\sum_j p_{ij}(a)\mu_j \leq \alpha \cdot \mu_i$ for all $(i, a) \in S \times A$. We have seen that contracting is equivalent to transient, and that any contracting MDP may be considered as a stochastic MDP with discounting. Therefore, results for discounted MDPs as the optimality equation, policy iteration, linear programming and value iteration can directly be applied to contracting MDPs. We will summarize this result in the following theorem and algorithms.

Theorem 4.20

(1) The value vector v is the unique solution of the optimality equation

$$x_i = max_a \{r_i(a) + \textstyle\sum_j p_{ij}(a)x_j\},\ i \in S.$$

(2) The value vector is the (componentwise) smallest vector which satisfies

$$x_i \geq r_i(a) + \textstyle\sum_j p_{ij}(a)x_j,\ (i, a) \in S \times A.$$

Algorithm 4.4 *Policy iteration algorithm*

Input: Instance of a contracting MDP.

Output: Optimal deterministic policy f^∞ and the value vector v.

```
1. Start with any $f^\infty \in C(D)$.

2. Compute $v(f^\infty)$ as the unique solution of the linear system $x = r(f) + P(f)x$.

3. a. Compute $s_{ia}(f) := r_i(a) + \sum_j p_{ij}(a)v_j(f^\infty) - v_i(f^\infty)$ for every $(i, a) \in S \times A$.

   b. Determine $A(i, f) := \{a \in A(i) \mid s_{ia}(f) > 0\}$ for every $i \in S$.

4. if $A(i, f) = \emptyset$ for every $i \in S$ then go to step 6.

   otherwise take $g$ such that $s_{ig(i)}(f) = max_a\, s_{ia}(f),\ i \in S$.

5. $f := g$ and return to step 2.

6. $v(f^\infty)$ is the value vector and $f^\infty$ an optimal policy (STOP).
```

Algorithm 4.5 *Linear programming algorithm*
Input: Instance of a contracting MDP.
Output: Optimal deterministic policy f^∞ and the value vector v.

1. Take any vector β with $\beta_j > 0,\ j \in S$.
2. Use the simplex method to compute optimal solutions v^* and x^* of the dual pair of linear programs:

$$min\Big\{\sum_j \beta_j v_j \;\Big|\; \sum_j \{\delta_{ij} - p_{ij}(a)\}v_j \ge r_i(a),\ (i,a) \in S \times A\Big\}$$

and

$$max\left\{\sum_{(i,a)} r_i(a)x_i(a) \;\left|\; \begin{array}{rl} \sum_{(i,a)} \{\delta_{ij} - p_{ij}(a)\}x_i(a) & = \beta_j,\ j \in S \\ x_i(a) & \ge 0,\ (i,a) \in S \times A \end{array}\right.\right\}.$$

3. Take $f_*^\infty \in C(D)$ such that $x_i^*(f_*(i)) > 0$ for every $i \in S$.
   v^* is the value vector and f_*^∞ an optimal policy (STOP).

As we have seen in the chapter on discounted MDPs, the policy iteration method is equivalent to the block-pivoting method for linear programming in the sense of the next theorem.

Theorem 4.21
(1) Any policy iteration algorithm is equivalent to a block-pivoting simplex algorithm.
(2) Any simplex algorithm is equivalent to a particular policy iteration algorithm.

Elimination of suboptimal actions
Also for contracting MDPs one can derive tests for the elimination of suboptimal actions. An action $a_i \in A(i)$ is *suboptimal* if $r_i(a_i) + \sum_j p_{ij}(a_i)v_j < v_i$. We shall derive a suboptimality test which can be used in the policy iteration and in the linear programming method. In both methods we have in each iteration a deterministic policy f^∞ for which we know the numbers $v_i(f^\infty),\ i \in S$, and also the numbers $s_{ia}(f)$ (in the linear programming method these numbers are obtained from the reduced costs), defined by $s_{ia}(f) := r_i(a) + \sum_j p_{ij}(a)v_j(f^\infty) - v_i(f^\infty)$ for every $(i,a) \in S \times A$.

Theorem 4.22
Let b be an upper bound of the value vector v. If $s_{ia_i}(f) < max_a\, s_{ia}(f) - \sum_j p_{ij}(a_i)\{b_j - v_j(f^\infty)\}$, then action $a_i \in A(i)$ is suboptimal.

Proof
We can write,

$$\begin{array}{rcl} r_i(a_i) + \sum_j p_{ij}(a_i)v_j & \le & r_i(a_i) + \sum_j p_{ij}(a_i)b_j \\ & = & s_{ia_i}(f) + v_i(f^\infty) + \sum_j p_{ij}(a_i)\{b_j - v_j(f^\infty)\} \\ & < & max_a\, s_{ia}(f) + +v_i(f^\infty) \\ & = & max_a\,\{r_i(a) + \sum_j p_{ij}(a)v_j(f^\infty)\} \\ & \le & max_a\,\{r_i(a) + \sum_j p_{ij}(a)v_j\} = v_i. \end{array}$$

□

Remark

The suboptimality test of Theorem 4.22 can also be used in Algorithm 4.3 with b an upper bound of the transient value vector w. The proof is similar to the proof of Theorem 4.22, namely:

$$\begin{array}{rcl} r_i(a_i) + \sum_j p_{ij}(a_i)w_j & \leq & r_i(a_i) + \sum_j p_{ij}(a_i)b_j \\ & = & s_{ia_i}(f) + v_i(f^\infty) + \sum_j p_{ij}(a_i)\{b_j - v_j(f^\infty)\} \\ & < & max_a\, s_{ia}(f) + +v_i(f^\infty) \\ & = & max_a\, \{r_i(a) + \sum_j p_{ij}(a)v_j(f^\infty)\} \\ & \leq & max_a\, \{r_i(a) + \sum_j p_{ij}(a)w_j\} = w_i. \end{array}$$

Let μ be an optimal solution of the linear program

$$min \left\{ \sum_j \mu_j \;\middle|\; \begin{array}{rcl} \sum_j \{\delta_{ij} - p_{ij}(a)\}\mu_j & \geq & 1,\ (i,a) \in S \times A \\ \mu_j & \geq & 0,\ j \in S \times A \end{array} \right\}. \tag{4.33}$$

We have seen in Remark 1 of Section 4.4 that this linear program has a finite optimal solution. The next theorem shows how an upper bound b of the value vector v can be derived from μ and from the data we have during an iteration.

Theorem 4.23

Let μ be an optimal solution of the linear program (4.33) and let $\alpha := 1 - \frac{1}{max_k\, \mu_k}$. Furthermore, let $b(f)$ be defined by $b(f) := v(f^\infty) + (1-\alpha)^{-1} \cdot max_{(i,a)} \frac{s_{ia}(f)}{\mu_i} \cdot \mu$. Then, $b(f)$ is an upper bound of the value vector v.

Proof

Let $M := max_{(i,a)} \frac{s_{ia}(f)}{\mu_i} \geq 0$ and let g^∞ be an optimal deterministic policy. Then, we have

$$M \geq \frac{s_{ig(i)}(f)}{\mu_i} = \frac{r_i(g) + \sum_j p_{ij}(g)v_j(f^\infty) - v_i(f^\infty)}{\mu_i},\ i \in S.$$

Consequently, $r(g) + P(g)v(f^\infty) - v(f^\infty) \leq M \cdot \mu$, i.e. $\{I - P(g)\}v(f^\infty) \geq r(g) - M \cdot \mu$, which implies

$$v(f^\infty) \geq \{I - P(g)\}^{-1}r(g) - M \cdot \{I - P(g)\}^{-1}\mu = v(g^\infty) - M \cdot \{I - P(g)\}^{-1}\mu.$$

From the definitions of μ and α, we obtain

$$\textstyle\sum_j p_{ij}(g)\mu_j \leq \mu_i - 1 \leq \mu_i - \frac{\mu_i}{max_k\, \mu_k} = \mu_i \cdot \left(1 - \frac{1}{max_k\, \mu_k}\right) = \alpha \cdot \mu_i,\ i \in S.$$

Hence, $P(g)\mu \leq \alpha \cdot \mu$, which implies $\{I - P(g)\}\mu \geq (1-\alpha) \cdot \mu$. Therefore, $(1-\alpha)^{-1}\mu \geq \{I - P(g)\}^{-1}\mu$.

Now, we can write $v = v(g^\infty) \leq v(f^\infty) + M \cdot \{I - P(g)\}^{-1}\mu \leq v(f^\infty) + M \cdot (1-\alpha)^{-1}\mu = b(f)$. □

In value iteration, we iterate: $v_i^{n+1} := (Uv^n)_i = max_a\, \{r_i(a) + \sum_j p_{ij}(a)v_j^n\},\ i \in S$ for $n = 1, 2, \ldots$. For the stop criterion we consider $\|v^{n+1} - v^n\|$ for some norm. In the contracting case we use the $\| \cdot \|_\mu$-norm, defined by $\|x\|_\mu := max_i\, \frac{1}{\mu_i} \cdot |x_i|$ for some μ with $\mu_i > 0,\ i \in S$ and $\sum_j p_{ij}(a)\mu_j \leq \alpha \cdot \mu_i$, where $\alpha \in [0, 1)$, for all $(i,a) \in S \times A$. Since the model is contracting, such μ and α exist (see also Exercise 3.1, in which the reader was asked to show that $\| \cdot \|_\mu$ is a correct norm). For this norm we obtain with operator L_f, defined by $L_f\, x = r(f) + P(f)x$,

$$\begin{array}{rcl} \|L_f\, x - L_f\, y\|_\mu & = & max_i\, \frac{1}{\mu_i} \cdot |\sum_j p_{ij}(f)(x_j - y_j)| \;\leq\; max_i\, \frac{1}{\mu_i} \cdot \sum_j p_{ij}(f) \cdot |x_j - y_j| \\ & = & max_i\, \frac{1}{\mu_i} \cdot \sum_j p_{ij}(f)\mu_j \cdot \frac{1}{\mu_j} \cdot |x_j - y_j| \\ & \leq & max_i\, \frac{1}{\mu_i} \cdot \sum_j p_{ij}(f)\mu_j \cdot \|x - y\|_\mu \leq \alpha \cdot \|x - y\|_\mu, \end{array}$$

i.e. L_f is a contraction with respect to the $\| \cdot \|_\mu$-norm with contraction factor α. Similarly, it can be shown that U also is a contraction with respect to the $\| \cdot \|_\mu$-norm with contraction factor α. The value iteration algorithm for contracting MDPs is similar to the value iteration algorithm for discounted MDPs. Below we formulate this algorithm.

Algorithm 4.6 *Value iteration*
Input: Instance of a contracting MDP and some scalar $\varepsilon > 0$.
Output: An ε-optimal deterministic policy f^∞ and a $\frac{1}{2}\varepsilon$-approximation of the value vector v.

```
1. Select $x \in \mathbb{R}^N$ arbitrary.
2. a. Compute $y$ by $y_i := max_a \{r_i(a) + \sum_j p_{ij}(a)x_j\},\ i \in S$.
   b. Let $f(i) \in argmax_a \{r_i(a) + \sum_j p_{ij}(a)x_j\},\ i \in S$.
3. if $\|y - x\|_\mu \leq \frac{1}{2}(1-\alpha)\alpha^{-1}\varepsilon$ then
       $f^\infty$ is an $\varepsilon$-optimal policy and $y$ is a $\frac{1}{2}\varepsilon$-approximation of the value vector $v$ (STOP)
   else $x := y$ and return to step 2.
```

Example 4.9
Consider the contracting model with $S = \{1, 2\}$; $A(1) = A(2) = \{1, 2\}$; $p_{11}(1) = \frac{1}{2},\ p_{12}(1) = 0$; $p_{11}(2) = 0$, $p_{12}(2) = \frac{1}{2}$; $p_{21}(1) = 0,\ p_{22}(1) = \frac{3}{4}$; $p_{21}(2) = \frac{1}{2},\ p_{22}(2) = 0$; $r_1(1) = 2,\ r_1(2) = 3$; $r_2(1) = 1,\ r_2(2) = 4$.
The optimality equation is:
$x_1 = max\{2 + \frac{1}{2}x_1, 3 + \frac{1}{2}x_2\},\ x_2 = max\{1 + \frac{3}{4}x_2, 4 + \frac{1}{2}x_1\}$ with solution $x_1 = \frac{20}{3},\ x_2 = \frac{22}{3}$.
If we apply policy iteration, stating with $f(1) = f(1) = 1$, we obtain:
Iteration 1:
$x_1 = 2 + \frac{1}{2}x_1,\ x_2 = 1 + \frac{3}{4}x_2 \ \rightarrow\ v_1(f^\infty) = 4,\ v_2(f^\infty) = 4$.
$s_{11}(f) = 0,\ s_{12}(f) = 1;\ s_{21}(f) = 0,\ s_{22}(f) = 2 \ \rightarrow\ A(1, f) = A(2, f) = \{2\}$.
$g(1) = g(2) = 2$.
Iteration 2:
$x_1 = 3 + \frac{1}{2}x_2,\ x_2 = 4 + \frac{1}{2}x_1 \ \rightarrow\ v_1(f^\infty) = \frac{20}{3},\ v_2(f^\infty) = \frac{22}{3}$.
$s_{11}(f) = -\frac{4}{3},\ s_{12}(f) = 0;\ s_{21}(f) = -\frac{5}{6},\ s_{22}(f) = 0 \ \rightarrow\ A(1, f) = A(2, f) = \emptyset$.
$(\frac{20}{3}, \frac{22}{3})$ is the value vector and f^∞ with $f(1) = f(2) = 2$ is an optimal policy.
The dual linear program with $\beta_1 = \beta_2 = \frac{1}{2}$ becomes:

$$max \left\{ 2x_1(1) + 3x_1(2) + x_2(1) + 4x_2(2) \;\middle|\; \begin{array}{rcrcrcrcl} \frac{1}{2}x_1(1) & + & x_1(2) & & & - & \frac{1}{2}x_2(2) & = & \frac{1}{2} \\ & & -\frac{1}{2}x_1(2) & + & \frac{1}{4}x_2(1) & + & x_2(2) & = & \frac{1}{2} \\ & & & & \multicolumn{3}{r}{x_1(1),\, x_1(2),\, x_2(1),\, x_2(2)} & \geq & 0 \end{array} \right\}.$$

The optimal solution of this dual program is: $x_1(1) = 0,\ x_1(2) = 1,\ x_2(1) = 0,\ x_2(2) = 1$.
The primal problem is:

$$min \left\{ \frac{1}{2}v_1 + \frac{1}{2}v_2 \;\middle|\; \begin{array}{rcrcl} \frac{1}{2}v_1 & & & \geq & 2; \\ v_1 & - & \frac{1}{2}v_2 & \geq & 0; \end{array} \begin{array}{rcrcl} & & \frac{1}{4}v_2 & \geq & 1 \\ -\frac{1}{2}v_1 & + & v_2 & \geq & 4 \end{array} \right\}.$$

This program has as optimal solution: $v_1 = \frac{20}{3},\ v_2 = \frac{22}{3}$. Hence, the value vector is $\left(\frac{20}{3}, \frac{22}{3}\right)$ and the optimal solution takes in both states action 2.
Finally, we present the value iteration method for this model with $v^0 = (4, 4)$ and $\varepsilon = 0.2$. The iteration scheme is: $y_1 = max\{2 + \frac{1}{2}x_1, 3 + \frac{1}{2}x_2\},\ y_2 = max\{1 + \frac{3}{4}x_2, 4 + \frac{1}{2}x_1\}$. The values of μ and α can be obtained as indicated in Remark 1 of section 4.4. The linear program (4.19) is;

$$min \left\{ \tfrac{1}{2}\mu_1 + \tfrac{1}{2}\mu_2 \;\middle|\; \begin{array}{lllll} \tfrac{1}{2}\mu_1 & & \geq & 1; & \tfrac{1}{4}\mu_2 \;\geq\; 1 \\ \mu_1 & - \;\tfrac{1}{2}\mu_2 & \geq & 1; & -\tfrac{1}{2}\mu_1 \;+\; \mu_2 \;\geq\; 1 \end{array} \right\}.$$

This program has as optimal solution: $\mu_1 = 3,\ \mu_2 = 4$. From μ we obtain $\alpha := 1 - \frac{1}{max_k\, \mu_k} = \frac{3}{4}$.
The algorithm terminates if the μ-norm of the difference of two subsequent y-vectors is at most $\frac{1}{6}\varepsilon = \frac{1}{30}$.
Since $\|y - x\|_\mu = max\{\frac{1}{3}|y_1 - x_1|, \frac{1}{4}|y_2 - x_2|\}$, the procedure is terminated as soon as $|y_1 - x_1| \leq \frac{1}{10}$ and $|y_2 - x_2| \leq \frac{2}{15}$. The results of the computation are summarized below.

	Iteration				
	1	2	3	4	5
y_1	5.00	6.00	6.25	6.50	6.57
y_2	6.00	6.50	7.00	7.13	7.25
f_1	2	2	2	2	2
f_2	2	2	2	2	2

Hence, f^∞ with $f(1) = f(2) = 2$ is a 0.2-optimal policy and $(6.57, 7.25)$ is a 0.1-approximation of the value vector.

4.8 Finite horizon and transient MPDs

In section 2.3 we have seen that any, nonstationary or stationary, finite horizon MDP can be transformed into a stationary transient MDP. Since the states $(j, T+1),\ j \in S$, are equal (the process terminates and there are no rewards), we may replaces these states by one state, say state $T+1$. The corresponding linear programs, according to Algorithm 4.5, are:

$$max \sum_{(i,a)} \sum_{t=1}^{T} r_i^t(a) x_{i,t}(a)$$

subject to

$$\begin{array}{rcllcl} \sum_a x_{j,1}(a) & & & & = & 1,\ j \in S \\ \sum_a x_{j,t}(a) & - & \sum_{(i,a)} p_{ij}^t(a) x_{i,t-1}(a) & & = & 1,\ j \in S,\ 2 \leq t \leq T \\ x_{T+1} & - & \sum_{(i,a)} x_{i,T}(a) & & = & 1 \\ & & & x_{i,t}(a) & \geq & 0,\ (i,a) \in S \times A,\ 1 \leq t \leq T \end{array}$$

and

$$min \sum_{j \in S} \sum_{t=1}^{T} v_j^t + v^{T+1}$$

subject to

$$\begin{array}{rcrcl} v_i^t & - & \sum_{j \in S} p_{ij}^t(a) v_j^{t+1} & \geq & r_i^t(a),\ (i,a) \in S \times A,\ 1 \leq T - 1 \\ v_i^T & - & v^{T+1} & \geq & r_i^T(a),\ (i,a) \in S \times A \\ & & v^{T+1} & = & 0 \end{array}$$

We shall show that a special version of the simplex method to solve the above linear programs is in fact the backward recursion Algorithm 2.1. Because $v^{T+1} = 0$, we can fill in this value in the minimization problem, and consequently the last equation of the maximization problem, i.e. the equation $x_{T+1} - \sum_{(i,a)} x_{i,T}(a) = 1$ can be deleted. For the maximization problem we introduce artificial variables $z_j^t,\ j \in S,\ 1 \leq t \leq T$ and for the minimization problem nonnegative slack variables $s_i^t(a),\ (i,a) \in S \times A,\ 1 \leq t \leq T$. Then, we obtain the following linear systems:

$$\begin{cases} z_j^1 &= 1 - \sum_a x_{j,1}(a), & j \in S \\ z_j^t &= 1 - \sum_a x_{j,t}(a) + \sum_{(i,a)} p_{ij}^t(a) x_{i,t-1}(a), & j \in S,\ 2 \le t \le T \end{cases}$$

$$\begin{cases} s_i^t(a) &= v_i^t - \sum_{j \in S} p_{ij}^t(a) v_j^{t+1} - r_i^t(a), & (i,a) \in S \times A,\ 1 \le T-1 \\ s_i^T(a) &= v_i^T - r_i^T(a), & (i,a) \in S \times A \end{cases}$$

We state the special version of the simplex method in the next algorithm.

Algorithm 4.7 *Determination of an optimal policy for a nonstationary MDP over T periods by linear programming (version 1).*
Input: Instance of a finite nonstationary MDP and the time horizon T.
Output: Optimal Markov policy $R_* = (f^1, f^2, \dots, f^T)$ and the value vector v^T.

1. $t := T + 1$ and select as basic variables the artificial variables $z_j^t,\ j \in S,\ 1 \le t \le T$.
2. $t := t - 1$.
3. Select for every $i \in S$ an action $f^t(i) \in A(i)$ such that $s_i^t(a)\big(f^t(i)\big) = min_a\, s_i^t(a)$.
4. Exchange for every $j \in S$ the basic variable z_j^t and the nonbasic variable $x_{j,t}\big(f^t(i)\big)$.
5. **if** $t \ge 2$ **then go to** step 2
   else begin $R_* := (f^1, f^2, \dots, f^T)$ is an optimal policy; $v^T := -s^1(f^1)$ is the value vector **end** (STOP).

This algorithm has the following properties:

1. Step 4 consists of N standard pivot iterations, which can be viewed as one block-pivoting iteration (cf. [48] p. 201).
2. By induction to the number of iterations, one can easily verify that during the algorithm:
   a. any pivot element is 1;
   b. in any column of $x_{j,t}(a)$ all elements in the first rows, i.e. the rows of $z_i^s,\ i \in S,\ 1 \le s \le t-1$ and $z_i^t,\ 1 \le i \le j-1$, are 0;
   c. in any row of z_j^t all elements in the first columns, i.e. the columns of $x_{i,s}(a),\ (i,a) \in S \times A$, $1 \le s \le t-2$, are 0.

   Hence, when in step 4 of the algorithm the basic variable z_j^t and the nonbasic variable $x_{j,t}\big(f^t(i)\big)$ are exchanged, then the first rows and first columns stay unchanged (first in the sense of 2b and 2c).
3. The only numbers that are of interest in any iteration (see step 3 of the algorithm) are the updated values of $s_i^t(a)$ for all $(i,a) \in S \times A$ and $t = 1, 2, \dots, T$. At the start of the algorithm, we have $s_i^T(a) = -r_i^T(a)$ for all $(i,a) \in S \times A$. We shall show by induction that at the start of the iteration for some t, we have $s_i^t(a) = -r_i^t(a) + \sum_j p_{ij}^t(a) s_j^{t+1}\big(f^{t+1}(j)\big)$, where $s_j^{T+1}\big(f^{T+1}(j)\big) = 0$ for all $j \in S$. By the properties mentioned in 2, the numbers $s_i^t(a),\ (i,a) \in S \times A$, do not change during the iterations $T,\ T-1, \dots, t+2$. In iteration with $t+1$, a block-pivoting step is executed with the pivot columns of the nonbasic variables $x_{j,t+1}\big(f^{t+1}(j)\big),\ j \in S$. Since the pivot elements are 1, during the single pivoting step for some $j \in S$, we have the assignment $s_i^t(a) := s_i^t(a) + p_{ij}^t(a) s_j^{t+1}\big(f^{t+1}(j)\big)$. Hence, after this block-pivoting step, we obtain the desired expression $s_i^t(a) = -r_i^t(a) + \sum_j p_{ij}^t(a) s_j^{t+1}\big(f^{t+1}(j)\big)$.
4. According to step 3 of the algorithm, we have

$$\begin{aligned} s_i^T\big(f^T(i)\big) &= min_a\, \{-r_i^T(a)\} = -max_a\, r_i^T(a),\ i \in S \\ s_i^t\big(f^t(i)\big) &= min_a\, \{-r_i^t(a) + \textstyle\sum_j p_{ij}^t(a) s_j^{t+1}\big(f^{t+1}(j)\big)\},\ i \in S,\ 1 \le t \le T-1 \\ & \quad -max_a\, \{r_i^t(a) - \textstyle\sum_j p_{ij}^t(a) s_j^{t+1}\big(f^{t+1}(j)\big)\},\ i \in S,\ 1 \le t \le T-1 \end{aligned}$$

Let $x_i^t := -s_i^t\big(f^t(i)\big),\ i \in S,\ 1 \le t \le T$. Then, with $x^{T+1} := 0$, we can write
$x_i^t = \{r(f^t) + P(f^t)x^{t+1}\}_i = max_a\,\{r_i^t(a) + \sum_j\, p_{ij}^t(a)x_j^{t+1})\},\ i \in S,\ 1 \le t \le T-1,$
which is the backward reduction of Theorem 2.1.

As a consequence of the above properties we obtain the following algorithm, which is equivalent to the backward induction Algorithm 2.1. The reason that we also have formulated Algorithm 4.7 is that this special simplex method can be used for problems with additional constraints (see section 9.2.5 of chapter 9). Algorithm 4.8 is not suitable for problems with additional constraints.

Algorithm 4.8 *Determination of an optimal policy for a nonstationary MDP over T periods by linear programming (version 2).*
Input: Instance of a finite nonstationary MDP and the time horizon T.
Output: Optimal Markov policy $R_* = (f^1, f^2, \dots, f^T)$ and the value vector v^T.

```
1. $t := T + 1$; $s_j^{T+1} := 0$ for all $j \in S$.
2. for $t = T, T-1, \dots, 1$ do
   begin
       $s_i^t(a) := -r_i^t(a) + \sum_j\, p_{ij}^t(a)s_j^{t+1}$ for all $(i, a) \in S \times A$;
       select for every $i \in S$ an action $f^t(i) \in A(i)$ such that $s_i^t\big(f^t(i)\big) = min_a\, s_i^t(a)$;
       $s_j^t := s_j^t\big(f^t(j)\big),\ j \in S$
   end
3. $R_* := (f^1, f^2, \dots, f^T)$ is an optimal policy; $v^T := -s^1(f^1)$ is the value vector (STOP).
```

Example 2.1 (continued)
The linear programs with the additional artificial and slack variables are:
$max\,\{x_{1,1}(1) + 2x_{1,1}(1) + 5x_{2,1}(2) + x_{1,2}(1) + 2x_{2,2}(1) + 5x_{2,2}(2) + x_{1,3}(1) + 2x_{2,3}(1) + 5x_{2,3}(2)\}$
subject to

$$\begin{array}{rcl}
z_1^1 & = & 1 - x_{1,1}(1) - x_{1,1}(2) \\
z_2^1 & = & 1 - x_{2,1}(1) - x_{2,1}(2) \\
z_1^2 & = & 1 + \frac{1}{2}x_{1,1}(1) + \frac{1}{4}x_{1,1}(2) + \frac{2}{3}x_{2,1}(1) + \frac{1}{3}x_{2,1}(2) - x_{1,2}(1) - x_{1,2}(2) \\
z_2^2 & = & 1 + \frac{1}{2}x_{1,2}(1) + \frac{3}{4}x_{1,2}(2) + \frac{1}{3}x_{2,2}(1) + \frac{2}{3}x_{2,2}(2) - x_{2,2}(1) - x_{2,2}(2) \\
z_1^3 & = & 1 + \frac{1}{2}x_{1,2}(1) + \frac{1}{4}x_{1,2}(2) + \frac{2}{3}x_{2,2}(1) + \frac{1}{3}x_{2,2}(2) - x_{1,3}(1) - x_{1,3}(2) \\
z_2^3 & = & 1 + \frac{1}{2}x_{1,2}(1) + \frac{3}{4}x_{1,2}(2) + \frac{1}{3}x_{2,2}(1) + \frac{2}{3}x_{2,2}(2) - x_{2,3}(1) - x_{2,3}(2)
\end{array}$$

respectively,
$min\,\{v_1^1 + v_2^1 + v_1^2 + v_2^2 + v_1^3 + v_2^3\}$
subject to

$$\begin{array}{rclrcl}
s_1^1(1) & = & v_1^1 - \frac{1}{2}v_1^2 - \frac{1}{2}v_2^2 - 1; & s_2^2(1) & = & v_2^2 - \frac{2}{3}v_1^3 - \frac{1}{3}v_2^3 - 2; \\
s_1^1(2) & = & v_1^1 - \frac{1}{4}v_1^2 - \frac{3}{4}v_2^2; & s_2^2(2) & = & v_2^2 - \frac{1}{3}v_1^3 - \frac{2}{3}v_2^3 - 5; \\
s_2^1(1) & = & v_2^1 - \frac{2}{3}v_1^2 - \frac{1}{3}v_2^2 - 2; & s_1^3(1) & = & v_1^3 - 1; \\
s_2^1(2) & = & v_2^1 - \frac{1}{3}v_1^2 - \frac{2}{3}v_2^2 - 5; & s_1^3(2) & = & v_1^3; \\
s_1^2(1) & = & v_1^2 - \frac{1}{2}v_1^3 - \frac{1}{2}v_2^3 - 1; & s_2^3(1) & = & v_2^3 - 2; \\
s_1^2(2) & = & v_1^2 - \frac{1}{4}v_1^3 - \frac{3}{4}v_2^3; & s_2^3(2) & = & v_2^3 - 5.
\end{array}$$

We first apply Algorithm 4.8. The simplex tableaus are stated below and the pivot elements are indicated by an asterisk.

Iteration 1

		$x_{1,1}(1)$	$x_{1,1}(2)$	$x_{2,1}(1)$	$x_{2,1}(2)$	$x_{1,2}(1)$	$x_{1,2}(2)$	$x_{2,2}(1)$	$x_{2,2}(2)$	$x_{1,3}(1)$	$x_{1,3}(2)$	$x_{2,3}(1)$	$x_{2,3}(2)$
z_1^1	1	1	1	0	0	0	0	0	0	0	0	0	0
z_2^1	1	0	0	1	1	0	0	0	0	0	0	0	0
z_1^2	1	$-\frac{1}{2}$	$-\frac{1}{4}$	$-\frac{2}{3}$	$-\frac{1}{3}$	1	1	0	0	0	0	0	0
z_2^2	1	$-\frac{1}{2}$	$-\frac{3}{4}$	$-\frac{1}{3}$	$-\frac{2}{3}$	0	0	1	1	0	0	0	0
z_1^3	1	0	0	0	0	$-\frac{1}{2}$	$-\frac{1}{4}$	$-\frac{2}{3}$	$-\frac{1}{3}$	^{*1}	1	0	0
z_2^3	1	0	0	0	0	$-\frac{1}{2}$	$-\frac{3}{4}$	$-\frac{1}{3}$	$-\frac{2}{3}$	0	0	1	^{*1}
	0	-1	0	-2	-5	-1	0	-2	-5	-1	0	-2	-5

Iteration 2

		$x_{1,1}(1)$	$x_{1,1}(2)$	$x_{2,1}(1)$	$x_{2,1}(2)$	$x_{1,2}(1)$	$x_{1,2}(2)$	$x_{2,2}(1)$	$x_{2,2}(2)$	z_1^3	$x_{1,3}(2)$	$x_{2,3}(1)$	z_2^3
z_1^1	1	1	1	0	0	0	0	0	0	0	0	0	0
z_2^1	1	0	0	1	1	0	0	0	0	0	0	0	0
z_1^2	1	$-\frac{1}{2}$	$-\frac{1}{4}$	$-\frac{2}{3}$	$-\frac{1}{3}$	^{*1}	1	0	0	0	0	0	0
z_2^2	1	$-\frac{1}{2}$	$-\frac{3}{4}$	$-\frac{1}{3}$	$-\frac{2}{3}$	0	0	1	^{*1}	0	0	0	0
$x_{1,3}(1)$	1	0	0	0	0	$-\frac{1}{2}$	$-\frac{1}{4}$	$-\frac{2}{3}$	$-\frac{1}{3}$	1	1	0	0
$x_{2,3}(2)$	1	0	0	0	0	$-\frac{1}{2}$	$-\frac{3}{4}$	$-\frac{1}{3}$	$-\frac{2}{3}$	0	0	1	1
	6	-1	0	-2	-5	-4	-4	$-\frac{13}{3}$	$-\frac{26}{3}$	1	1	3	5

Iteration 3

		$x_{1,1}(1)$	$x_{1,1}(2)$	$x_{2,1}(1)$	$x_{2,1}(2)$	z_1^2	$x_{1,2}(2)$	$x_{2,2}(1)$	z_2^2	z_1^3	$x_{1,3}(2)$	$x_{2,3}(1)$	z_2^3
z_1^1	1	1	^{*1}	0	0	0	0	0	0	0	0	0	0
z_2^1	1	0	0	1	^{*1}	0	0	0	0	0	0	0	0
$x_{1,2}(1)$	1	$-\frac{1}{2}$	$-\frac{1}{4}$	$-\frac{2}{3}$	$-\frac{1}{3}$	1	1	0	0	0	0	0	0
$x_{2,2}(2)$	1	$-\frac{1}{2}$	$-\frac{3}{4}$	$-\frac{1}{3}$	$-\frac{2}{3}$	0	0	1	1	0	0	0	0
$x_{1,3}(1)$	$\frac{11}{6}$	$-\frac{5}{12}$	$-\frac{3}{8}$	$-\frac{4}{9}$	$-\frac{7}{18}$	$\frac{1}{2}$	$\frac{1}{4}$	$-\frac{1}{3}$	$\frac{1}{3}$	1	1	0	0
$x_{2,3}(2)$	$\frac{13}{6}$	$-\frac{7}{12}$	$-\frac{5}{8}$	$-\frac{5}{9}$	$-\frac{11}{18}$	$\frac{1}{2}$	$-\frac{1}{4}$	$\frac{1}{3}$	$\frac{2}{3}$	0	0	1	1
	$\frac{56}{3}$	$-\frac{22}{3}$	$-\frac{15}{2}$	$-\frac{68}{9}$	$-\frac{109}{9}$	4	0	$\frac{13}{3}$	$\frac{26}{3}$	1	1	3	5

Iteration 4

		$x_{1,1}(1)$	z_1^1	$x_{2,1}(1)$	z_2^1	z_1^2	$x_{1,2}(2)$	$x_{2,2}(1)$	z_2^2	z_1^3	$x_{1,3}(2)$	$x_{2,3}(1)$	z_2^3
$x_{1,1}(2)$	1	1	1	0	0	0	0	0	0	0	0	0	0
$x_{2,1}(2)$	1	0	0	1	1	0	0	0	0	0	0	0	0
$x_{1,2}(1)$	$\frac{17}{12}$	$-\frac{1}{4}$	$\frac{1}{4}$	$-\frac{1}{3}$	$\frac{1}{3}$	1	1	0	0	0	0	0	0
$x_{2,2}(2)$	$\frac{29}{12}$	$\frac{1}{4}$	$\frac{3}{4}$	$\frac{1}{3}$	$\frac{2}{3}$	0	0	1	1	0	0	0	0
$x_{1,3}(1)$	$\frac{187}{72}$	$-\frac{1}{24}$	$\frac{3}{8}$	$-\frac{1}{18}$	$\frac{7}{18}$	$\frac{1}{2}$	$\frac{1}{4}$	$-\frac{1}{3}$	$\frac{1}{3}$	1	1	0	0
$x_{2,3}(2)$	$\frac{245}{72}$	$\frac{1}{24}$	$\frac{5}{8}$	$\frac{1}{18}$	$\frac{11}{18}$	$\frac{1}{2}$	$-\frac{1}{4}$	$\frac{1}{3}$	$\frac{2}{3}$	0	0	1	1
	$\frac{689}{18}$	$\frac{1}{6}$	$\frac{15}{2}$	$\frac{41}{9}$	$\frac{109}{9}$	4	0	$\frac{13}{3}$	$\frac{26}{3}$	1	1	3	5

From this optimal tableau it follows that the optimal policy is: $f^1(1) = 2,\ f^1(2) = 2,\ f^2(1) = 1,$ $f^2(2) = 2,\ f^3(1) = 1,\ f^3(2) = 2$. The value vector is $(\frac{15}{2}, \frac{109}{9})$.

Next, we apply Algorithm 4.8. We start with $t = 4$ and $s^4 = (0, 0)$.

Iteration 1: $t = 3$.

$s_1^3(1) = -1,\ s_2^3(2) = 0,\ s_2^3(1) = -2,\ s_2^3(2) = -5;\ f^3(1) = 1,\ f^3(2) = 2;\ s^3 = (-1, -5)$.

Iteration 2: $t = 2$.

$s_1^2(1) = -1 + \frac{1}{2}(-1) + \frac{1}{2}(-5) = -4,\ s_1^2(2) = 0 + \frac{1}{4}(-1) + \frac{3}{4}(-5) = -4,$

$s_2^2(1) = -2 + \frac{2}{3}(-1) + \frac{1}{3}(-5) = -\frac{13}{3},\ s_2^2(2) = -5 + \frac{1}{3}(-1) + \frac{2}{3}(-5) = -\frac{26}{3});$

$f^2(1) = 1$ (or 2), $f^2(2) = 2;\ s^2 = (-4, -\frac{26}{3}.$

Iteration 3: $t = 1$.

$s_1^1(1) = -1 + \frac{1}{2}(-4) + \frac{1}{2}(-\frac{26}{3}) = -\frac{22}{3},\ s_1^1(2) = 0 + \frac{1}{4}(-4) + \frac{3}{4}(-\frac{26}{3}) = -\frac{15}{2},$

$s^1_2(1) = -2 + \frac{2}{3}(-4) + \frac{1}{3}(-\frac{26}{3}) = -\frac{68}{9},\ s^1_2(2) = -5 + \frac{1}{3}(-4) + \frac{2}{3}(-\frac{26}{3}) = -\frac{109}{9}$;
$f^1(1) = 2,\ f^1(2) = 2;\ s^2 = (-\frac{15}{2}, -\frac{109}{9})$.
Hence, the optimal policy is: $f^1(1) = 2,\ f^1(2) = 2,\ f^2(1) = 1$ (or 2), $f^2(2) = 2,\ f^3(1) = 1,\ f^3(2) = 2$. The value vector $v^T = (\frac{15}{2}, \frac{109}{9})$.

4.9 Positive MDPs

Throughout this section we have the following assumption.

Assumption 4.2
(1) The model is substochastic.
(2) $r_i(a) \geq 0$ for all $(i, a) \in S \times A$.

A vector $x \in \mathbb{R}^N$ is said to be *superharmonic* if $v_i \geq r_i(a) + \sum_j p_{ij}(a)v_j$ for all $(i, a) \in S \times A$.

Theorem 4.24
The value vector v is the (componentwise) smallest nonnegative superharmonic vector.

Proof
From Assumption 4.2 it follows that every policy is regular and that the MDP is normalized. Hence, by Theorem 4.12, there exists a regular optimal deterministic policy. From Assumption 4.2 it also follows that v is p-summable. Then, by Theorem 4.13, it follows that v is a superharmonic vector. It is obvious that v is nonnegative. Suppose that x is also a nonnegative superharmonic vector. It is sufficient to show that $x \geq v(f^\infty)$ for every $f^\infty \in C(D)$. Take an arbitrary $f^\infty \in C(D)$. Then, the superharmonicity of x implies $x \geq r(f) + P(f)x$. By iterating this inequality, we obtain

$$x \geq \textstyle\sum_{t=1}^n P^{t-1}(f)r(f) + P^n(f)x \geq \sum_{t=1}^n P^{t-1}(f)r(f),\ n \in \mathbb{N}.$$

Hence, let $n \to \infty$, $x \geq \sum_{t=1}^\infty P^{t-1}(f)r(f) = v(f^\infty)$. □

Theorem 4.24 implies that the value vector v is the unique optimal solution of the linear program

$$min \left\{ \sum_j \beta_j x_j \;\middle|\; \begin{array}{rcl} \sum_j \{\delta_{ij} - p_{ij}(a)\}x_j & \geq & r_i(a),\quad (i,a) \in S \times A \\ x_j & \geq & 0,\quad j \in S \end{array} \right\}, \tag{4.34}$$

where $\beta_j > 0,\ j \in S$. The dual program is

$$max \left\{ \sum_{(i,a)} r_i(a)x_i(a) \;\middle|\; \begin{array}{rcl} \sum_{(i,a)} \{\delta_{ij} - p_{ij}(a)\}x_i(a) & \leq & \beta_j,\quad j \in S \\ x_i(a) & \geq & 0,\quad (i,a) \in S \times A \end{array} \right\}. \tag{4.35}$$

The dual program (4.35) is feasible ($x = 0$ is a feasible solution). Therefore, (4.35) either has a finite optimal solution or the there is an infinite solution. We consider both cases separately.

<u>Case 1:</u> (4.35) has a finite optimal solution.
In this case there is also an extreme finite optimal solution x^*, which is computed for instance by the simplex method. The next theorem shows how an optimal policy is obtained from x^*.

Theorem 4.25

Let x^* be an extreme optimal solution of (4.35). Then, any f_*^∞ such that $x_i^*\big(f_*(i)\big) > 0$ for each i with $\sum_a x_i^*(a) > 0$ is an optimal policy.

Proof

By introducing slack variables we can write the constraints of problem (4.35) as

$$\begin{cases} \sum_{(i,a)} \{\delta_{ij} - p_{ij}(a)\}x_i(a) + y_j & = & \beta_j, \ \ j \in S \\ x_i(a) & \geq & 0, \ \ (i,a) \in S \times A \\ y_j & \geq & 0, \ \ j \in S \end{cases}$$

It follows from the theory of linear programming that the optima of the pair of dual linear programs are equal, i.e. $\sum_j \beta_j v_j = \sum_{(i,a)} r_i(a)x_i^*(a)$. Since x^* is an extreme point and the dual program (4.35) has N constraints, the optimal extreme solution (x^*, y^*) has at most N positive components. Because

$$\sum_a x_j^*(a) + y_j^* = \beta_j + \sum_{(i,a)} p_{ij}(a)x_i^*(a) \geq \beta_j > 0, \ j \in S,$$

for each $j \in S$, either $\sum_a x_j^*(a) > 0$ or $y_j^* > 0$, implying that for each j with $\sum_a x_j^*(a) > 0$ there is exactly one action $f_*(j)$ for which $x_j^*\big(f_*(j)\big) > 0$.

Furthermore, we have $(x^*)^T = (\beta - y^*)^T + (x^*)^T P(f_*)$. By iterating this equality, we obtain

$$(x^*)^T = (\beta - y^*)^T \sum_{t=1}^n P^{t-1}(f_*) + (x^*)^T P^n(f_*) \text{ for all } n \in \mathbb{N}.$$

Consequently,

$$(x^*)^T r(f_*) = (\beta - y^*)^T \sum_{t=1}^n P^{t-1}(f_*)r(f_*) + (x^*)^T P^n(f_*)r(f_*) \text{ for all } n \in \mathbb{N}.$$

Since $v(f_*^\infty) = \sum_{t=1}^\infty P^{t-1}(f_*)r(f_*) \leq v$ and v is finite, we have $lim_{n\to\infty} P^n(f_*)r(f_*) = 0$.

Therefore, by letting $n \to \infty$,

$$\beta^T v = \sum_j \beta_j v_j = \sum_{(i,a)} r_i(a)x_i^*(a) = (x^*)^T r(f_*) = (\beta - y^*)^T v(f_*^\infty) \leq \beta^T v(f_*^\infty),$$

implying that f_*^∞ is an optimal policy. □

Remark

If the MDP is contracting, then the linear programs (4.34) and (4.35) have finite optimal solutions. The converse statement is not true, in general. Consider the MDP:

$S = \{1,2\}$; $A(1) = A(2) = \{1\}$; $p_{11}(1) = 0$, $p_{12}(1) = 1$, $p_{21}(1) = 0$, $p_{22}(1) = 1$: $r_1(1) = 1$, $r_2(1) = 0$. This MDP is not transient. However, program (4.35) with $\beta_1 = \beta_2 = \frac{1}{2}$ becomes:

$$max\big\{x_1(1) \bigm| x_1(1) \leq \tfrac{1}{2}; \ -x_1(1) \leq \tfrac{1}{2}; \ x_1(1) \geq 0; \ x_2(1) \geq 0\big\}.$$

This program has a finite optimal solution, namely: $x_1(1) = \frac{1}{2}$; $x_2(1) = 0$.

Case 2: (4.35) has an infinite optimal solution.

If we solve the problem by the simplex method, starting with the extreme feasible solution $x = 0$, we obtain after a finite number of iterations a simplex tableau corresponding to an extreme feasible

solution (x^*, y^*) in which one of the columns is nonpositive. In this column, the coefficient of the transformed objective function is strictly negative. This column provides a direction vector $s^* \neq 0$ such that
(1) $x^*(\lambda) := x^* + \lambda s^*$ is feasible for all $\lambda \geq 0$.
(2) $\sum_{(i,a)} r_i(a)x^*_i(a)(\lambda) \to +\infty$ for $\lambda \to +\infty$.
From (1) and (2) it follows that

$$\sum_{(i,a)} \{\delta_{ij} - p_{ij}(a)\}s^*_i(a) \quad \leq 0, \quad j \in S \tag{4.36}$$

$$s^*_i(a) \quad \geq 0, \quad (i,a) \in S \times A \tag{4.37}$$

$$\sum_{(i,a)} r_i(a)s^*_i(a) \quad > 0 \tag{4.38}$$

As we have seen in the proof of Theorem 4.25 the basis of the simplex tableau corresponding to (x^*, y^*) contains for each state $j \in S$ at most one positive $x^*_j(a)$. Let

$S_* := \{j \mid \sum_a x^*_j(a) > 0\}$ and let $a_j \in A(j)$ such that $x^*_j(a_j) > 0,\ j \in S_*$.

Lemma 4.7
If the nonpositive column corresponds to the nonbasic variable $x^*_k(a_*)$, then $k \notin S_*$.

Proof
Assume the contrary, i.e. $k \in S_*$. Let s^* be the direction vector that satisfies (4.36), (4.37) and (4.38). Let a^* be the nonpositive column of $x_k(a_*)$ and let i_j be the row index of the basic variable $x_j(a_j),\ j \in S_*$.

Then, the direction vector s^* satisfies $s^*_j(a) = \begin{cases} -a^*_{i_j} & j \in S_*,\ a = a_j; \\ 1 & j = k,\ a = a_*; \\ 0 & \text{elsewhere.} \end{cases}$

Let $\delta := \frac{-a^*_{i_k}}{1-a^*_{i_k}}$, so $0 \leq \delta < 1$. Define the stationary policy π^∞ by

$$\pi_{ia} := \begin{cases} 1 & i \in S_*,\ i \neq k,\ a = a_i \\ \delta & i \in S_*,\ i = k,\ a = a_i \\ 1-\delta & i \in S_*,\ i = k,\ a = a_* \\ 1 & i \notin S_*,\ \text{for some arbitrary action } a_i \in A(i) \end{cases} \tag{4.39}$$

Then, $s^*_i(a) = s^*_i \cdot \pi_{ia}$ for all $(i,a) \in S \times A$, where $s^*_i := \sum_a s^*_i(a),\ i \in S$. Let $S_+ := \{i \mid s^*_i > 0\}$. Furthermore, it can easily be verified that $P(\pi) = \delta \cdot P(f_1) + (1-\delta) \cdot P(f_2)$, where $f_1(i) := a_i,\ i \in S$ and $f_2(i) := \begin{cases} f_1(i) & i \neq k; \\ a_* & i = k. \end{cases}$
By (4.36), we obtain

$$0 \geq \sum_{(i,a)} \{\delta_{ij} - p_{ij}(a)\}s^*_i(a) = \sum_{(i,a)} \{\delta_{ij} - p_{ij}(a)\}\pi_{ia} \cdot s^*_i = s^*_j - \sum_i p_{ij}(\pi)s^*_i,\ j \in S, \tag{4.40}$$

implying $0 < \sum_j s^*_j \leq \sum_i\{\sum_j p_{ij}(\pi)\}s^*_i \leq \sum_i s^*_i$. Therefore,

$$(s^*)^T e = (s^*)^T P(\pi)e \text{ and } \textstyle\sum_j p_{ij}(\pi) = 1 \text{ for every } i \text{ with } s^*_i > 0. \tag{4.41}$$

Therefore, also $\sum_j p_{ij}(f_1) = \sum_j p_{ij}(f_2) = 1$ for every $i \in S_+$. From (4.40) and (4.41) it follows that $(s^*)^T \leq (s^*)^T P(\pi)$ and $(s^*)^T e = (s^*)^T P(\pi) e$. Consequently, $(s^*)^T = (s^*)^T P(\pi)$. From the theory of Markov chains, it is well known that $S_+ \subseteq R(\pi)$, where $R(\pi)$ is the set of states that are recurrent in the Markov chain induced by $P(\pi)$, and S_+ is closed under $P(\pi)$. Therefore,

$$S_+ \text{ is closed under } P(f_1) \text{ and } \sum_j p_{ij}^{(n)}(f_1) = \sum_{j \in S_+} p_{ij}^{(n)}(f_1) = 1,\ i \in S_+,\ n \in \mathbb{N}. \tag{4.42}$$

Since (x^*, y^*) is an extreme feasible solution and since $S_+ \subseteq S_*$, we also have

$$x_j^*(a_j) = \beta_j + \sum_{(i,a)} p_{ij}(a) x_i(a) \geq \beta_j + \sum_{i \in S_+} p_{ij}(f_1) x_i(a_i),\ j \in S_+. \tag{4.43}$$

Notice that S_+ is closed under $P(f_1)$ and define the vectors $\overline{x}, \overline{\beta}$ and the matrix $\overline{P}$ as the restrictions of the vectors x^*, β and the matrix $P(f_1)$ to the states of S_+. Then, (4.43) becomes in vector notation: $\overline{x}^T \geq \overline{\beta}^T + \overline{x}^T \overline{P}$. By iterating this inequality, we obtain

$$\overline{x}^T \geq \textstyle\sum_{t=1}^n \overline{\beta}^T \overline{P}^{t-1} + \overline{x}^T \overline{P}^n \text{ for all } n \in \mathbb{N}.$$

Consequently, since $\beta_j > 0$ for all j, we have $\sum_{t=1}^\infty p_{ij}^{(t-1)}(f_1) < \infty$ for all $i,\ j \in S_+$, implying $p_{ij}^{(n)}(f_1) \to 0$ for $n \to \infty$ for all $i, j \in S_+$. Hence, $\sum_{j \in S_+} p_{ij}^{(n)}(f_1) \to 0$ for $n \to \infty$ for all $i \in S_+$. This contradicts (4.42) and concludes the proof that if the nonpositive column corresponds to the nonbasic variable $x_k^*(a_*)$, then $k \notin S_*$. □

From the proof of Lemma 4.7 it follows that the direction vector s^* induces a deterministic policy f_*^∞ with

$$f_*(i) := \begin{cases} a_i & \text{with } x_i^*(a_i) > 0 & i \in S_* \\ a_k & \text{with } x_k^*(a_k) \text{ the nonbasic variable corresponding to the nonpositive column} & i = k \\ a_i & \text{an arbitrary action from } A(i) & i \notin S_* \text{ and } i \neq k \end{cases}$$

Lemma 4.8
$v_j(f_*^\infty) = +\infty$ for at least one state j.

Proof
Similar as in the proof of Lemma 4.7 we can derive that $S_+ \subseteq R(f_*^\infty)$ and that S_+ is closed under $P(f_*)$. From (4.38) it follows that $(s^*)^T r(f_*) > 0$. Hence, there is a state $j \in S_+$ such that $r_j(f_*) > 0$. For all states i in the same ergodic set as j, we have

$$v_i(f_*^\infty) = \textstyle\sum_{t=1}^\infty \{P^{t-1}(f_*) r(f_*)\}_i = \lim_{n \to \infty} n \cdot \frac{1}{n} \cdot \sum_{t=1}^n \{P^{t-1}(f_*) r(f_*)\}_i$$

and

$$\lim_{n \to \infty} \tfrac{1}{n} \cdot \textstyle\sum_{t=1}^n \{P^{t-1}(f_*) r(f_*)\}_i = \{P^*(f_*) r(f_*)\}_i \geq p_{ij}^*(f_*) r_j(f_*) > 0,$$

where $P^*(f_*)$ is the stationary matrix of the Markov chain $P(f_*)$ (for the definition of the stationary matrix and its properties we refer to Chapter 5). Hence, we can conclude that $v_j(f_*^\infty) = +\infty$. □

We construct in the following way an optimal policy f^∞. We first determine the ergodic sets in S_+ which have a state j such that $r_j(f_*) > 0$. For any state in these ergodic sets we define $f(i) := f_*(i)$. Outside these ergodic sets, we choose actions which lead to these ergodic sets, if possible. Then, f^∞ has for certain initial states, say the states $S_1 \subseteq S$, a total reward $+\infty$. The states $S \backslash S_1$ are closed under every policy and we repeat the same approach to the model of the states $S \backslash S_1$. The method is summarized in the following algorithm, which can be used for any positive MDP without knowing in advance whether or not program (4.35) has a finite optimum.

Algorithm 4.9 *Determination of an optimal policy for positive MDPs*
Input: Instance of a substochastic positive MDP.
Output: Optimal deterministic policy f^∞.

1. Take any vector β, where $\beta_j > 0,\ j \in S$.
2. Use the simplex method to solve the linear program
$$max \left\{ \sum_{(i,a)} r_i(a)x_i(a) \;\middle|\; \begin{array}{rcl} \sum_{(i,a)} \{\delta_{ij} - p_{ij}(a)\}x_i(a) & \leq & \beta_j,\ j \in S \\ x_i(a) & \geq & 0,\ (i,a) \in S \times A \end{array} \right\}.$$
3. **if** the linear program has a finite optimal solution x^* **then go to** step 4
else go to step 5.
4. Select any $f^\infty \in C(D)$ such that $x^*_i(f(i)) > 0$ for every i such that $\sum_a x^*_i(a) > 0$; **go to** step 12.
5. Let a^* be the nonpositive column in the simplex tableau in which the infinite optimum is discovered. Suppose that this column corresponds to the nonbasic variable $x_k(a_k)$. Let i_j be the row index of the basic variable $x_j(a_j),\ j \in S_* := \{j \mid \sum_a x^*_j(a) > 0\}$ and let the direction vector s^* be defined by $s^*_j(a) := \begin{cases} -a^*_{i_j} & j \in S_*,\ a = a_j; \\ 1 & j = k,\ a = a_k; \\ 0 & \text{elsewhere.} \end{cases}$
6. Take $f(i) \in C(D)$ such that $f(i) = a_i$ for $i \in S_* \cup \{k\}$.
7. Determine in the Markov chain induced by $P(f)$ the ergodic sets on $S_+ := \{j \mid \sum_a s^*_i(a) > 0\}$.
8. Determine S_1 as the union of the ergodic sets which contain a state j for which $r_j(f) > 0$.
9. **if** $S_1 = S$ **then go to** step 12.
else go to step 10.
10. **if** there is a triple $(i, a_i, , j)$ with $i \in S \backslash S_1,\ a_i \in A(i),\ j \in S_1$ and $p_{ij}(a_i) > 0$ **then begin** $f(i) := a_i;\ S_1 := S_1 \cup \{i\}$; **go to** step 9 **end**
else go to step 11.
11. $S := S \backslash S_1$; **return to** step 2.
12. f^∞ is an optimal deterministic policy (STOP).

Theorem 4.26
Algorithm 4.9 determines an optimal policy in a finite number of iterations.

Proof
If the linear program in step 2 of the algorithm has a finite optimal solution, then Theorem 4.25 implies that the policy f^∞, defined in step 4 of the algorithm, is optimal.
Next, suppose that the linear program has an infinite solution. Then, by Lemma 4.8, $v_j(f^\infty) = +\infty$ for every $j \in S_1$, where S_1 is the nonempty set defined in step 8. Hence, if $S_1 = S$, then the algorithm terminates in step 12 with an optimal policy f^∞. If $S_1 \neq S$, then in step 10 the policy f^∞ may be redefined in states $i \in S \backslash S_1$ with $p_{ij}(f) > 0$ for at least one state $j \in S_1$. Consequently, for these states i we also have $v_i(f^\infty) = +\infty$. Notice that, because S_1 is extended to $S_1 \cup \{i\}$, the property $v_j(f^\infty) = +\infty$ for all $j \in S_1$ is maintained.

If step 11 is reached, then $p_{ij}(a) = 0$ for all triples (i, a, j) such that $i \in S \backslash S_1,\ a \in A(i),\ j \in S_1$. Hence, the set $S \backslash S_1$ is closed under any policy. Therefore, we may return to step 2 and repeat the procedure in the state space $S \backslash S_1$. Since in each iteration $S_1 \neq \emptyset$, the algorithm has at most N iterations and, consequently, finite. □

Example 4.10

Consider the following substochastic positive MDP.
$S = \{1, 2, 3, 4, 5, 6, 7\};\ A(1) = A(2) = A(4) = A(7) = \{1, 2\};\ A(3) = A(5) = \{1, 2, 3\};\ A(6) = \{1\}$.
The nonzero transitions are: $p_{11}(1) = 1;\ p_{13}(2) = 1;\ p_{21}(1) = 1;\ p_{24}(2) = 1;\ p_{33}(1) = \frac{1}{2};\ p_{31}(2) = 1;$ $p_{37}(3) = 1;\ p_{43}(1) = 1;\ p_{42}(2) = 1;\ p_{54}(1) = \frac{1}{2};\ p_{53}(2) = \frac{1}{2};\ p_{56}(3) = 1;\ p_{67}(1) = \frac{1}{2};\ p_{77}(1) = \frac{1}{2};\ p_{76}(2) = 1.$
The rewards are: $r_1(1) = 0;\ r_1(2) = 0;\ r_2(1) = 0;\ r_2(2) = 1;\ r_3(1) = 1;\ r_3(2) = 1;\ r_3(3) = 1;\ r_4(1) = 1;$ $r_4(2) = 1;\ r_5(1) = 1;\ r_5(2) = 2;\ r_5(3) = 3;\ r_6(1) = 1;\ r_7(1) = 1;\ r_7(2) = 1.$
Taking $\beta_j = \frac{1}{7}$ for $j = 1, 2, \ldots, 7$, the linear program becomes (without the nonnegativity of the x-variables):

$max\ \{x_2(2) + x_3(1) + x_3(2) + x_3(3) + x_4(1) + x_4(2) + x_5(1) + 2x_5(2) + 3x_5(3) + x_6(1) + x_7(1) + x_7(2)\}$
subject to

$$\begin{array}{cccccccccccccccc}
x_1(2) & -x_2(1) & & & -x_3(2) & & & & & & & & & & \leq \frac{1}{7} \\
 & x_2(1) & +x_2(2) & & & & & -x_4(2) & & & & & & & \leq \frac{1}{7} \\
-x_1(2) & & & +\frac{1}{2}x_3(1) & +x_3(2) & +x_3(3) & -x_4(1) & & & -\frac{1}{2}x_5(2) & & & & & \leq \frac{1}{7} \\
 & & -x_2(2) & & & & +x_4(1) & +x_4(2) & -\frac{1}{2}x_5(1) & & & & & & \leq \frac{1}{7} \\
 & & & & & & & & x_5(1) & +x_5(2) & +x_5(3) & & & & \leq \frac{1}{7} \\
 & & & & & & & & & & -x_5(3) & +x_6(1) & & -x_7(2) & \leq \frac{1}{7} \\
 & & & & & -x_3(3) & & & & & -x_5(3) & -\frac{1}{2}x_6(1) & +\frac{1}{2}x_7(1) & +x_7(2) & \leq \frac{1}{7}
\end{array}$$

With slack variables $y_j,\ j = 1, 2, \ldots, 7$, we obtain the following first simplex tableau corresponding to the basic solution $x = 0$.

		$x_1(2)$	$x_2(1)$	$x_2(2)$	$x_3(1)$	$x_3(2)$	$x_3(3)$	$x_4(1)$	$x_4(2)$	$x_5(1)$	$x_5(2)$	$x_5(3)$	$x_6(1)$	$x_7(1)$	$x_7(2)$
y_1	$\frac{1}{7}$	1	−1	0	0	−1	0	0	0	0	0	0	0	0	0
y_2	$\frac{1}{7}$	0	1	1	1	0	0	0	−1	0	0	0	0	0	0
y_3	$\frac{1}{7}$	−1	0	0	$\frac{1}{2}$	1	1	−1	0	0	$-\frac{1}{2}$	0	0	0	0
y_4	$\frac{1}{7}$	0	0	−1	0	0	0	1	1	$-\frac{1}{2}$	0	0	0	0	0
y_5	$\frac{1}{7}$	0	0	0	0	0	0	0	0	1	1	1	0	0	0
y_6	$\frac{1}{7}$	0	0	0	0	0	0	0	0	0	0	−1	1	0	−1
y_7	$\frac{1}{7}$	0	0	0	0	0	−1	0	0	0	0	−1	$-\frac{1}{2}$	$\frac{1}{2}$	1
	0	0	0	−1	−1	−1	−1	−1	−1	−1	−2	−3	−1	−1	−1

As first pivot column we select the column of $x_2(2)$. After one pivot step we obtain as simplex tableau:

		$x_1(2)$	$x_2(1)$	y_2	$x_3(1)$	$x_3(2)$	$x_3(3)$	$x_4(1)$	$x_4(2)$	$x_5(1)$	$x_5(2)$	$x_5(3)$	$x_6(1)$	$x_7(1)$	$x_7(2)$
y_1	$\frac{1}{7}$	1	−1	0	0	−1	0	0	0	0	0	0	0	0	0
$x_2(2)$	$\frac{1}{7}$	0	1	1	1	0	0	0	−1	0	0	0	0	0	0
y_3	$\frac{1}{7}$	−1	0	0	$\frac{1}{2}$	1	1	−1	0	0	$-\frac{1}{2}$	0	0	0	0
y_4	$\frac{1}{7}$	0	1	1	0	0	0	1	0	$-\frac{1}{2}$	0	0	0	0	0
y_5	$\frac{1}{7}$	0	0	0	0	0	0	0	0	1	1	1	0	0	0
y_6	$\frac{1}{7}$	0	0	0	0	0	0	0	0	0	0	−1	1	0	−1
y_7	$\frac{1}{7}$	0	0	0	0	0	−1	0	0	0	0	−1	$-\frac{1}{2}$	$\frac{1}{2}$	1
	0	0	1	1	−1	−1	−1	−1	−1	−1	−2	−3	−1	−1	−1

In this tableau the column of $x_4(2)$ generates an infinite direction with $a^* = (0, -1, 0, 0, 0, 0, 0)^T$. $S_* = \{2\},\ k = 4$ and $a_k = 2$. The direction vector s^* is defined by $s^*_2(2) = 1,\ s^*_4(2) = 1,\ s^*_j(a) = 0$ for all $(j, a) \neq (2, 2)$ or $(4, 2)$. Take $f(1) = 1,\ f(2) = 2,\ f(3) = 1,\ f(4) = 2,\ f(5) = 1,\ f(6) = 1$

and $f(7) = 1$. $S_+ = \{2, 4\}$. $P(f)$ has on S_+ one ergodic set, namely $\{2, 4\}$. $S_1 = \{2, 4\}$. The triple $(i, a_i, j) = (5, 1, 4)$ satisfies $i \in S\backslash S_1,\ a_i \in A(i),\ j \in S_1$ and $p_{ij}(a_i) > 0$, so $f(4) := 1$ and $S_1 := \{2, 4, 5\}$. Then, there is no triple (i, a_i, j) which satisfies $i \in S\backslash S_1,\ a_i \in A(i),\ j \in S_1$ and $p_{ij}(a_i) > 0$. Hence, we repeat the procedure for $S = \{1, 3, 6, 7\}$.

The linear program is (without the nonnegativity of the x-variables):

$max\, \{x_3(1) + x_3(2) + x_3(3) + x_6(1) + x_7(1) + x_7(2)\}$
subject to

$$\begin{array}{rrrrrrrcl}
x_1(2) & & -x_3(2) & & & & & \leq & \frac{1}{7} \\
-x_1(2) & +\frac{1}{2}x_3(1) & +x_3(2) & +x_3(3) & & & & \leq & \frac{1}{7} \\
 & & & & +x_6(1) & & -x_7(2) & \leq & \frac{1}{7} \\
 & & & -x_3(3) & -\frac{1}{2}x_6(1) & +\frac{1}{2}x_7(1) & +x_7(2) & \leq & \frac{1}{7}
\end{array}$$

With slack variables $y_j,\ j = 1, 3, 6, 7$, we obtain the following first simplex tableau corresponding to the basic solution $x = 0$.

		$x_1(2)$	$x_3(1)$	$x_3(2)$	$x_3(3)$	$x_6(1)$	$x_7(1)$	$x_7(2)$
y_1	$\frac{1}{7}$	1	0	-1	0	0	0	0
y_3	$\frac{1}{7}$	-1	$\frac{1}{2}$	1	1	0	0	0
y_6	$\frac{1}{7}$	0	0	0	0	1	0	-1
y_7	$\frac{1}{7}$	0	0	0	-1	$-\frac{1}{2}$	$\frac{1}{2}$	1
	0	0	-1	-1	-1	-1	-1	-1

As first pivot column we select the column of $x_3(2)$. After one pivot step we obtain as simplex tableau:

		$x_1(2)$	$x_3(1)$	y_3	$x_3(3)$	$x_6(1)$	$x_7(1)$	$x_7(2)$
y_1	$\frac{2}{7}$	0	$\frac{1}{2}$	1	1	0	0	0
$x_3(2)$	$\frac{1}{7}$	-1	$\frac{1}{2}$	1	1	0	0	0
y_6	$\frac{1}{7}$	0	0	0	0	1	0	-1
y_7	$\frac{1}{7}$	0	0	0	-1	$-\frac{1}{2}$	$\frac{1}{2}$	1
	$\frac{1}{7}$	-1	$-\frac{1}{2}$	1	0	-1	-1	-1

In this tableau the column of $x_1(2)$ generates an infinite direction with $a^* = (0, -1, 0, 0)^T$. $S_* = \{3\},\ k = 1$ and $a_k = 2$. The direction vector s^* is defined by $s^*_3(2) = 1,\ s^*_1(2) = 1,\ s^*_j(a) = 0$ for all $(j, a) \neq (3, 2)$ or $(1, 2)$. Take $f(1) = 2,\ f(3) = 2,\ f(6) = 1$ and $f(7) = 1$. $S_+ = \{1, 3\}$. $P(f)$ has on S_+ one ergodic set, namely $\{1, 3\}$. $S_1 = \{1, 3\}$. There is no triple (i, a_i, j) which satisfies $i \in S\backslash S_1,\ a_i \in A(i),\ j \in S_1$ and $p_{ij}(a_i) > 0$. Hence, we repeat the procedure for $S = \{6, 7\}$.

The linear program is (without the nonnegativity of the x-variables):

$$max \left\{ x_6(1) + x_7(1) + x_7(2) \;\middle|\; \begin{array}{rrrcl} x_6(1) & & -x_7(2) & \leq & \frac{1}{7} \\ -\frac{1}{2}x_6(1) & +\frac{1}{2}x_7(1) & +x_7(2) & \leq & \frac{1}{7} \end{array} \right\}.$$

With slack variables $y_j,\ j = 6, 7$, we obtain the following first simplex tableau corresponding to the basic solution $x = 0$.

		$x_6(1)$	$x_7(1)$	$x_7(2)$
y_6	$\frac{1}{7}$	1	0	-1
y_7	$\frac{1}{7}$	$-\frac{1}{2}$	$\frac{1}{2}$	1
	0	-1	-1	-1

As first pivot column we select the column of $x_6(1)$. After one pivot step we obtain as simplex tableau:

		y_6	$x_7(1)$	$x_7(2)$
$x_6(1)$	$\frac{1}{7}$	1	0	-1
y_7	$\frac{3}{14}$	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$
	$\frac{1}{7}$	1	-1	-2

As second pivot column we select the column of $x_7(2)$. After one pivot step we obtain as simplex tableau:

		y_6	$x_7(1)$	y_7
$x_6(1)$	$\frac{4}{7}$	2	1	2
$x_7(2)$	$\frac{3}{7}$	1	1	2
	1	3	1	4

This is an optimal simplex tableau with optimal solution $x_6(1) = \frac{4}{7}, x_7(1) = 0, x_7(2) = \frac{3}{7}$. Define $f(6) = 1$ and $f(7) = 2$.

We have obtained an optimal deterministic policy f^∞ with $f(1) = 2, f(2) = 2, f(3) = 2, f(4) = 2$, $f(5) = 1,\ f(6) = 1$ and $f(7) = 2$. The value vector $v = (+\infty, +\infty, +\infty, +\infty, +\infty, +\infty, 3, 4)$.

4.10 Negative MDPs

Throughout this section we assume the following:

Assumption 4.3
(1) The model is substochastic.
(2) $r_i(a) \geq 0$ for all $(i, a) \in S \times A$.

In this case the total expected reward $v_i(R)$ exists for all $i \in S$ and all policies R, possibly $-\infty$. If there exists a transient policy R, then we have $-\infty < v_i(R) \leq v_i \leq 0$ for all $i \in S$. Theorem 4.14 shows how the existence of a transient policy can be verified.

Consider a policy $f^\infty \in C(D)$. It is intuitively clear that if $\phi_i(f^\infty)$, the average reward with starting state i, is strictly negative, the total reward $v_i(f^\infty) = -\infty$; if $\phi_i(f^\infty) = 0$ and state i is recurrent in the Markov chain induced by f^∞, then $v_i(f^\infty) = 0$. In the next theorem we show this property. In the proof we use some results from average reward MDPs, which are shown in the next chapter. In the model for the average reward we have stochastic MDPs ($\sum_j p_{ij}(a) = 1$ for all $(i, a) \in S \times A$). Therefore we have to use the extended model as introduced in the proof that (1) implies (8) in Theorem 4.10 and with $r_0(1) = 0$.

Theorem 4.27
Let f^∞ be an arbitrary stationary and deterministic policy.
(1) If $\phi_i(f^\infty) < 0$, then $v_i(f^\infty) = -\infty$.
(2) If $\phi_i(f^\infty) = 0$ and i is recurrent in the Markov chain induced by f^∞, then $v_i(f^\infty) = 0$.

Proof

(1) In the next chapter we show that $v^\alpha(f^\infty) = lim_{\alpha \uparrow 1} \left\{ \frac{\phi(f^\infty)}{1-\alpha} + u(f) \right\}$ for some vector $u(f)$. Hence, if $\phi_i(f^\infty) < 0$, then $v_i(f^\infty) = lim_{\alpha \uparrow 1}\, v^\alpha(f^\infty) = -\infty$.

(2) In the next chapter we also show that $\phi(f^\infty) = P^*(f)r(f)$, where $P^*(f)$ is the stationary matrix of $P(f)$. Let R_k be the ergodic set which contains state i. Then, since the rows of R_k $P^*(f)$ are identical for the states of R_k, $\phi_j(f^\infty) = 0$ for all $j \in R_k$.

Furthermore, we have $p^*_{ij}(f) > 0,\ j \in R_k,\ p^*_{ij}(f) = 0,\ j \notin R_k$, and $p^t_{ij}(f) = 0,\ j \notin R_k,\ t \in \mathbb{N}_0$.

From $0 = \phi_i(f^\infty) = \sum_j p^*_{ij}(f) r_j(f) = \sum_{j \in R_k} p^*_{ij}(f) r_j(f)$, we have $r_j(f) = 0,\ j \in R_k$.

Hence, $v_i(f^\infty) = \sum_{t=1}^\infty \sum_j p^{t-1}_{ij}(f) r_j(f) = \sum_{j \in R_k} p^{t-1}_{ij}(f) r_j(f) = 0$. □

Corollary 4.2
Let $f_1^\infty \in C(D)$ be an average optimal policy.
(1) If $\phi_i(f_1^\infty) < 0$, then $v_i = -\infty$.
(2) If $\phi_i(f_1^\infty) = 0$ and i is recurrent in the Markov chain induced by f_1^∞, then $v_i = 0$.

Proof
(1) Since $\phi_i(f^\infty) \leq \phi_i(f_1^\infty) < 0$ for every $f^\infty \in C(D)$, by Theorem 4.27 part (1), $v_i(f^\infty) = -\infty$ for every $f^\infty \in C(D)$, i.e. $v_i = -\infty$.
(2) From Theorem 4.27 it follows that $v_i(f_1^\infty) = 0$, implying $v_i = 0$. □

We can construct an optimal policy f_*^∞ for negative MDPs in the following way:
Firstly, we determine an average optimal policy, say f_1^∞. Let $S_0 := \{i \mid \phi_i(f_1^\infty) < 0\}$.
For $i \in S_0$: $v_i = -\infty,\ f_*(i) = f_1(i)$ is optimal in state i and remove state i from the model.
For $i \notin S_0$: if there are actions a such that $\sum_{j \in S_0} p_{ij}(a) > 0$: remove these actions from $A(i)$.
In the resulting model, we have $\phi_j(f_1^\infty) = 0$ for all states j, and there is at least one recurrent class. We determine the recurrent states $R(f_1)$ in the Markov chain of $P(f_1)$. From Corollary 4.2 we know that $f_*(i) = f_1(i)$ is optimal in the states $i \in R(f_1)$. If there are states left, then we try to find an ergodic set with respect to another average optimal policy, say f_2^∞. Therefore, we first change the model in the following way:

$$S := S \backslash R(f_1) \cup \{0\};\ A(i) := \begin{cases} A(i) & i \neq 0 \\ \{1\} & i = 0 \end{cases};\ r_i(a) := \begin{cases} r_i(a) & i \neq 0,\ a \in A(i) \\ -1 & i = 0,\ a \in A(i) \end{cases}$$

$$p_{ij}(a) := \begin{cases} p_{ij}(a) & i \neq 0,\ j \neq 0,\ a \in A(i); \\ \sum_{k \in R(f_1)} p_{ik}(a) & i \neq 0,\ j = 0,\ a \in A(i); \\ 1 & i = 0,\ j = 0,\ a \in A(i); \\ 0 & i = 0,\ j \neq 0,\ a \in A(i). \end{cases}$$

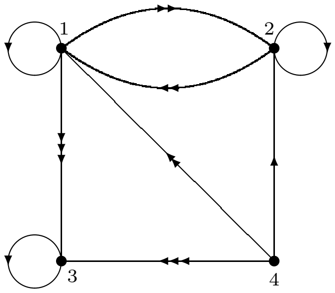
In this reduced model, we compute an average optimal policy, say f_2^∞. Then, there are two possible situations:
Case 1: $\phi_i(f_2^\infty) = 0$ for at least one state i.
We remove the set $\{i \mid \phi_i(f_2^\infty) < 0\}$, which contains at least the state 0. Determine in the remaining set $\{i \mid \phi_i(f_2^\infty) = 0\}$ the states which are recurrent under $P(f_2)$, say $R(f_2)$. Then, $v_i(f_2^\infty) = 0,\ i \in R(f_2)$, and consequently, $f_*(i) = f_2(i)$ are optimal actions for the states of $R(f_2)$. We remove the states of $R(f_2)$ and repeat this procedure for the model with the remaining states.
Case 2: $\phi_i(f_2^\infty) < 0$ for all states i.
In this case we redefine $r_0(1) := 0,\ p_{0j}(1) := 0$ for all j. For the remaining states together with the set S_1 of already removed states, there is an optimal transient policy and we compute such an optimal transient policy, e.g. by Algorithm 4.5.
Every time we encounter Case 1, the state space decreases with at least one state. Hence, after a finite number of iterations either we encounter Case 2 or we have an average optimal policy such that all states i for which $\phi_i(f_2^\infty) = 0$ are recurrent in the Markov chain induced by this policy. Below we present the algorithm.

Algorithm 4.10 *Negative MDPs*
Input: Instance of a substochastic negative MDP.
Output: Optimal deterministic policy f^∞.

```
1. if $\sum_j p_{ij}(a) < 1$ for at least one pair $(i,a) \in S \times A$ then construct the extended model:
   $S := S \cup \{0\};\ A(i) := \begin{cases} A(i) & i \neq 0 \\ \{1\} & i = 0 \end{cases};\ r_i(a) := \begin{cases} r_i(a) & i \neq 0,\ a \in A(i) \\ 0 & i = 0,\ a \in A(i) \end{cases}$
   $p_{ij}(a) := \begin{cases} p_{ij}(a) & i \neq 0,\ j \neq 0,\ a \in A(i); \\ 1 - \sum_{k \neq 0} p_{ik}(a) & i \neq 0,\ j = 0,\ a \in A(i); \\ 1 & i = 0,\ j = 0,\ a \in A(i); \\ 0 & i = 0,\ j \neq 0,\ a \in A(i). \end{cases}$
2. a. Compute an average optimal policy $f_1^\infty$ (see Chapter 5).
   b. Let $S_0 := \{i \mid \phi_i(f_1^\infty) < 0\};\ f_*(i) := f_1(i),\ i \in S_0$.
   c. if $S_0 = S$ then go to step 7;
   d. $S_1 := \emptyset$.
   e. for every $(i,a) \in (S \backslash S_0) \times A$ do
         if $\sum_{j \in S_0} p_{ij}(a) > 0$ then $A(i) := A(i) \backslash \{a\}$.
   f. $S := S \backslash S_0$.
3. a. Determine the set $R(f_1)$ of the recurrent states in $S$ in the Markov chain $P(f_1)$.
   b. $f_*(i) := f_1(i),\ i \in R(f_1)$.
   c. if $R(f_1) = S$ then go to step 4g.
   d. Construct the following reduced model:
      $S := S \backslash R(f_1) \cup \{0\};\ A(i) := \begin{cases} A(i) & i \neq 0 \\ \{1\} & i = 0 \end{cases};\ r_i(a) := \begin{cases} r_i(a) & i \neq 0 \\ -1 & i = 0 \end{cases}$
      $p_{ij}(a) := \begin{cases} p_{ij}(a) & i \neq 0,\ j \neq 0,\ a \in A(i); \\ \sum_{k \in R(f_1)} p_{ik}(a) & i \neq 0,\ j = 0,\ a \in A(i); \\ 1 & i = 0,\ j = 0,\ a \in A(i); \\ 0 & i = 0,\ j \neq 0,\ a \in A(i). \end{cases}$
4. a. Compute an average optimal policy $f_1^\infty$ in the reduced model (see Chapter 5).
   b. $S_2 := \{i \mid \phi_i(f_1^\infty) < 0\}$.
   c. if $S = S_2$ then begin $S_1 := S_1 \cup (S_2 \backslash \{0\})$; go to step 4g end
   d. $S_1 := S_1 \cup (S_2 \backslash \{0\})$.
   e. for every $(i,a) \in (S \backslash S_2) \times A$ do if $\sum_{j \in S_2} p_{ij}(a) > 0$ then $A(i) := A(i) \backslash \{a\}$.
   f. $S := S \backslash S_2$; return to step 3a.
   g. if $S_1 = \emptyset$ then go to step 7.
5. Construct the following transient model:
   $S := S_1;\ A(i) := A(i),\ i \in S_1;\ r_i(a) := r_i(a),\ i \in S_1,\ a \in A(i);\ p_{ij}(a) := p_{ij}(a),\ i,j \in S_1,\ a \in A(i)$.
6. Compute an optimal transient policy $f_*^\infty$, e.g. by Algorithm 4.5.
7. $f_*^\infty$ is an optimal policy (STOP).
```

Example 4.11

Let $S = \{1,2,3,4\}$; $A(1) = \{1,2,3\}$, $A(2) = \{1,2\}$, $A(3) = \{1\}$, $A(4) = \{1,2,3\}$.
The nonzero transition probabilities are:
$p_{11}(1) = 0.5$; $p_{12}(2) = 1$; $p_{13}(3) = 0$; $p_{22}(1) = 0.5$; $p_{21}(2) = 1$; $p_{33}(1) = 0.5$; $p_{42}(1) = 1$; $p_{41}(2) = 1$; $p_{43}(3) = 1$.
The rewards are:
$r_1(1) = -1$; $r_1(2) = 0$; $r_1(3) = 1$; $r_2(1) = -1$; $r_2(2) = 0$; $r_2(2) = 0$; $r_3(1) = -1$; $r_4(1) = 0$; $r_4(2) = -1$; $r_4(3) = 0$.
The graph at the right hand side presents the model (partly).



The algorithm has the following result:

1. The extended model becomes:
$S = \{0,1,2,3,4\}$; $A(0) = \{1\}$; $A(1) = \{1,2,3\}$, $A(2) = \{1,2\}$, $A(3) = \{1\}$, $A(4) = \{1,2,3\}$.
The nonzero probabilities are:
$p_{00}(1) = 1$; $p_{10}(1) = 0.5$; $p_{11}(1) = 0.5$; $p_{12}(2) = 1$; $p_{13}(3) = 1$; $p_{20}(1) = 0.5$; $p_{22}(1) = 0.5$; $p_{21}(2) = 1$; $p_{30}(1) = 0.5$; $p_{33}(1) = 0.5$; $p_{42}(1) = 1$; $p_{41}(2) = 1$; $p_{43}(3) = 1$.
The rewards are: $r_0(1) = 0$; $r_1(1) = -1$; $r_1(2) = 0$; $r_1(3) = 1$; $r_2(1) = -1$; $r_2(2) = 0$; $r_3(1) = -1$; $r_4(1) = 0$; $r_4(2) = -1$; $r_4(3) = 0$.

2. a. An average optimal policy f_1^∞ of the extended model is $f_1(0) = f_1(1) = f_1(2) = f_1(3) = f_1(4) = 1$ and has average rewards $\phi_0(f_1^\infty) = \phi_1(f_1^\infty) = \phi_2(f_1^\infty) = \phi_3(f_1^\infty) = \phi_4(f_1^\infty) = 0$.
b. $S_0 := \emptyset$.
d. $S_1 := \emptyset$.
f. $S := \{0,1,2,3,4\}$.

3. a. $R(f_1) = \{0\}$.
b. $f_*(0) := 1$.
d. The reduced model is the same as the first extended model, except that $r_0(1) := -1$.

4. a. An average optimal policy f_1^∞ in the reduced model is $f_1(0) = 1$, $f_1(1) = 2$, $f_1(2) = 2$, $f_1(3) = 1$, $f_1(4) = 1$ with average rewards $\phi_0(f_1^\infty) = -1$, $\phi_1(f_1^\infty) = \phi_2(f_1^\infty) = 0$, $\phi_3(f_1^\infty) = -1$, $\phi_4(f_1^\infty) = 0$.
b. $S_2 := \{0,3\}$.
d. $S_1 := \{3\}$.
e. $A(1) := \{2\}$; $A(2) := \{2\}$; $A(4) := \{1,2\}$.
f. $S := \{1,2,4\}$.

3. a. $R(f_1) = \{1,2\}$.
b. $f_*(1) := f_*(2) := 2$.
d. The reduced model is: $S := \{0,4\}$; $A(0) := \{1\}$; $A(4) := \{1,2\}$; $p_{00}(1) := 1$; $p_{40}(1) := 1$; $p_{40}(2) := 1$; $r_0(1) = -1$; $r_4(1) = 0$; $r_4(2) = -1$.

4. a. An average optimal policy f_1^∞ in the reduced model is $f_1(0) = f_1(4) = 1$ with average rewards $\phi_0(f^\infty) = \phi_4(f^\infty) = -1$.
b. $S_2 := \{0,4\} = S$.
c. $S_1 = \{3,4\}$.

5. The transient model becomes: $S := \{3,4\}$; $A(3) := \{1\}$; $A(4) := \{1,2\}$; $r_3(1) = -1$; $r_4(1) = 0$; $r_4(2) = -1$; $p_{33} = 0.5$ (the other transition probabilities are 0).

6. An optimal transient policy f_*^∞ is: $f_*(3) = f_*(4) = 1$ with $v_3(f_*^\infty) = -2$; $v_4(f_*^\infty) = 0$.

7. f_*^∞ with $f_*(1) = f_*(2) = 2$; $f_*(3) = f_*(4) = 1$ is an optimal policy with $v_1 = v_2 = 0$; $v_3 = -2$ and $v_4 = 0$.

Theorem 4.28
Algorithm 4.10 terminates with an optimal policy.

Proof
First, we consider the finiteness of the algorithm. The only loop may possibly occur in the steps 3 and 4. However, each time when we return in step 3d, the number of states in S decreases, namely:

> The model defined in step 3d has state 0 as absorbing state and $\phi_0(f_1^\infty) = -1$. Therefore, $0 \in S_2 := \{i \mid \phi_i(f_1^\infty) < 0\}$. If $S_2 = \{0\}$, then there are in the Markov chain induced by $P(f_1)$ no positive transactions from any state $i \neq 0$ to state 0. But then, $S\backslash\{0\}$ contains an ergodic set. So, $|S_2| \cup |R(f_1)| \geq 2$ and consequently the state space S defined in step 3d in some iteration of the algorithm has fewer states than the state space S in the previous iteration.

Consequently, Algorithm 4.10 determines a deterministic policy f_*^∞ in a finite number of iterations. This policy has the following properties:
(1) $v_i(f_*^\infty) = v_i = -\infty$ for all $i \in S_0$.
(2) $v_i(f_*^\infty) = v_i = 0$ for all $i \in S\backslash(S_0 \cup S_1)$.
(3) f_*^∞ is an optimal transient policy in the model defined in step 5.
Hence, it is sufficient to show that S_1 has an optimal transient policy and that f_*^∞ is optimal for the states in S_1. Firstly, suppose that there exists an optimal nontransient policy, say g^∞, in the model of step 5. Since g^∞ is nontransient, $R(g) \cap S_1 \neq \emptyset$. From the construction of S_1 (see the steps 4c and 4d) it follows that $\phi_i(g^\infty) < 0,\ i \in R(g)$, implying that $v_i(g^\infty) = -\infty,\ i \in R(g)$, which contradicts that g^∞ is optimal. Next, we will prove that f_*^∞ is an optimal policy. By the properties (1) and (2) it is sufficient to show that $v_i(f_*^\infty) \geq v_i(f^\infty)$ for $i \in S_1$ and for all policies f^∞. Since $v_j(f_*^\infty) = 0$ for all $j \in S\backslash(S_0 \cup S_1)$, we have $r_j(f_*) = 0$ for all $j \in S\backslash(S_0 \cup S_1)$. Hence, for $i \in S_1$, the total reward $v_i(f_*^\infty)$ is equal to the total reward in the transient model. □

4.11 Convergent MDPs

An MDP is *convergent* if $max\{v_i^+(R), v_i^-(R)\} < \infty$ for all policies R and all $i \in S$, i.e. the total absolute reward is finite for each policy. Hence, the value vector v is also finite. In this section we make the following assumption.

Assumption 4.4 *The MDP is convergent.*

A vector x has the property *anne* (short for asymptotic nonnegative expectation) if for every policy R, we have $\lim_{t\to\infty} \mathbb{P}_{i,R}\{X_t = j\} \cdot x_j^- = 0$ for all $i, j \in S$. Hence, any nonnegative vector has the property anne.

Theorem 4.29
The value vector v is the smallest superharmonic vector with the property anne.

Proof
Theorem 4.13 implies that v is a superharmonic vector. Let R be an arbitrary policy. Notice that $v_i^- = max\{-v_i, 0\} \leq max\{-v_i(R), 0\} = \{v_i(R)\}^-$. Since $r_j^-(a) = max\{0, -r_j(a)\} \geq -r_j(a)$, we have

$$\begin{array}{rcl} v_i^- & \leq & \{v_i(R)\}^- \ = \ max\Big\{ \sum_{t=1}^\infty \sum_{(j,a)} \mathbb{P}_{i,R}\{X_t = j, Y_t = a\} \cdot \{-r_j(a)\}, 0 \Big\} \\ & \leq & max\Big\{ \sum_{t=1}^\infty \sum_{(j,a)} \mathbb{P}_{i,R}\{X_t = j, Y_t = a\} \cdot r_j^-(a), 0 \Big\} \\ & = & \sum_{t=1}^\infty \sum_{(j,a)} \mathbb{P}_{i,R}\{X_t = j, Y_t = a\} \cdot r_j^-(a) = v_i^-(R) \text{ for all policies } R \text{ and all states } i. \end{array}$$

Let $R = (\pi^1, \pi^2, \dots)$ an arbitrary Markov policy with $v_i^-(R) < \infty,\ i \in S$, and let $R_t := (\pi^t, \pi^{t+1}, \dots)$ for $t \in \mathbb{N}$. Then, for any $t \in \mathbb{N}$ and any $i \in S$, we obtain

$$\begin{array}{rcl} \sum_j \mathbb{P}_{i,R}\{X_t = j\} \cdot v_j^- & \leq & \sum_j \mathbb{P}_{i,R}\{X_t = j\} \cdot v_j^-(R_t) \\ & = & \sum_j \mathbb{P}_{i,R}\{X_t = j\} \cdot \big\{ \sum_{s=1}^\infty \sum_{(k,a)} \mathbb{P}_{j,R_t}\{X_s = k, Y_s = a\} \cdot r_k^-(a) \big\} \\ & = & \sum_{s=1}^\infty \sum_{(k,a)} \sum_j \mathbb{P}_{i,R}\{X_t = j\} \cdot \mathbb{P}_{j,R_t}\{X_s = k, Y_s = a\} \cdot r_k^-(a) \\ & = & \sum_{s=1}^\infty \sum_{(k,a)} \mathbb{P}_{i,R}\{X_{t+s-1} = k, Y_{t+s-1} = a\} \cdot r_k^-(a) \\ & = & \sum_{m=t}^\infty \sum_{(k,a)} \mathbb{P}_{i,R}\{X_m = k, Y_m = a\} \cdot r_k^-(a). \end{array}$$

Let $A_t := \sum_{m=t}^\infty \sum_{(k,a)} \mathbb{P}_{i,R}\{X_m = k, Y_m = a\} \cdot r_k^-(a)$, for all $t \in \mathbb{N}$. Since, $v_i^-(R) < \infty$, we have $\lim_{t\to\infty} A_t = 0$, implying $\lim_{t\to\infty} \sum_j \mathbb{P}_{i,R}\{X_t = j\} \cdot v_j^- = 0$ for all $i \in S$ and for all Markov policy R. By Corollary 1.1, $\lim_{t\to\infty} \sum_j \mathbb{P}_{i,R}\{X_t = j\} \cdot v_j^- = 0$ for all policies. Hence, $\lim_{t\to\infty} \mathbb{P}_{i,R}\{X_t = j\} \cdot v_j^- = 0$ for all $i, j \in S$ and all policies R. Therefore, we have shown that v has the property anne.

Finally, suppose that w is also a superharmonic vector with the property anne. In order to show $v \leq w$, it is sufficient to show that $v(R) \leq w$ for all Markov policies R. Define by induction: $x_i^0 := w_i,\ i \in S$, and $x_i^{n+1} := max_a\{r_i(a) + \sum_j p_{ij}(a)x_j^n\},\ i \in S$. Since w is superharmonic it can easily be verified by induction on n that $x^n \leq w$ for $n = 0, 1, \dots$.

We first show that for any $i \in S$ and any Markov policy R, $v_i^t(R) + \sum_j \mathbb{P}_{i,R}\{X_{t+1} = j\} \cdot w_j \leq x_i^t,\ t \in \mathbb{N}$. Choose any Markov policy $R = (\pi^1, \pi^2, \dots)$ and any $i \in S$. The proof will be given by induction on t. For $t = 1$, we have

$v_i^1(R) + \sum_j \mathbb{P}_{i,R}\{X_2 = j\} \cdot w_j = \sum_a \pi_{ia}^1\{r_i(a) + \sum_j p_{ij}(a)w_j\} \leq max\{r_i(a) + \sum_j p_{ij}(a)w_j\} = x_i^1.$

Suppose that the inequality is valid for some t. Then, with Markov policy $R_* = (\pi^2, \pi^3, \dots)$, we can write

$$\begin{array}{rcl} v_i^{t+1}(R) + \sum_j \mathbb{P}_{i,R}\{X_{t+2} = j\} \cdot w_j & = & \sum_a \pi_{ia}^1\big\{r_i(a) + \sum_k p_{ik}(a)v_k^t(R_*) + \sum_{j,k} p_{ik}(a)\, \mathbb{P}_{k,R_*}\{X_{t+1} = j\} \cdot w_j\big\} \\ & \leq & max_a\big\{r_i(a) + \sum_k p_{ik}(a)\{v_k^t(R_*) + \sum_j \mathbb{P}_{k,R_*}\{X_{t+1} = j\} \cdot w_j\}\big\} \\ & \leq & max_a\{r_i(a) + \sum_k p_{ik}(a)x_k^t\} = x_i^{t+1}. \end{array}$$

Take any Markov policy R. Then, by the anne property of w, we have

$$\liminf_{t\to\infty} \sum_j \mathbb{P}_{i,R}\{X_t = j\} \cdot w_j \geq \liminf_{t\to\infty} \sum_j \mathbb{P}_{i,R}\{X_t = j\} \cdot (-w_j^-) = 0.$$

Hence, we obtain

$$v_i(R) = \lim_{t\to\infty} v_i^t(R) \leq limsup_{t\to\infty} \big\{v_i^t(R) + \sum_j \mathbb{P}_{i,R}\{X_{t+1} = j\} \cdot w_j\big\} \leq x_i^t \leq w_i,\ i \in S,$$

and consequently, $v_i = sup_{R \in C(M)}\, v_i(R) \leq w_i,\ i \in S.$ □

We have seen in Section 4.5 that an optimal policy f^∞ is conserving, i.e. $v = r(f) + P(f)v$, and that the reverse statement is not necessarily true. If the policy is also *equalizing*, i.e. $\lim_{t\to\infty} \sum_j p_{ij}^t(f)v_j^+ = 0$ for all $i \in S$, then the policy is optimal as the next theorem shows.

Theorem 4.30

A policy $f^\infty \in C(D)$ is optimal if and only if f^∞ is conserving and equalizing.

Proof

⇒ Let f^∞ be an optimal policy, i.e. $v(f^\infty) = v$. Policy f^∞ is conserving, because

$$v = v(f^\infty) = r(f) + P(f)\{\sum_{t=1}^\infty P^{t-1}(f)r(f)\} = r(f) + P(f)v(f^\infty) = r(f) + P(f)v.$$

Iterating the above equation gives $v = \sum_{t=1}^n P^{t-1}(f)r(f) + P^n(f)v,\ n \in \mathbb{N}$. Since v is finite and $v = \sum_{t=1}^\infty P^{t-1}(f)r(f)$, we have $\lim_{n\to\infty} P^n(f)v = 0$, i.e. $\sum_j p_{ij}^n(f)v_j = 0,\ i \in S$. Since v has the property anne, $lim_{n\to\infty} \sum_j \, p^n_{ij}(f)v^-_j = 0, \; i \in S$, implying that also $\lim_{n\to\infty} \sum_j \, p^n_{ij}(f)v^+_j = 0, \; i \in S$, i.e. f^∞ is conserving and equalizing.

$\Leftarrow$ Since f^∞ is conserving, $v = r(f) + P(f)v$, implying $v = \sum_{t=1}^n \, P^{t-1}(f)r(f) + P^n(f)v, \; n \in \mathbb{N}$. The equalizing property gives $\limsup_{n\to\infty} \sum_j \, p^n_{ij}(f)v_j \leq \lim_{n\to\infty} \sum_j \, p^n_{ij}(f)v^+_j = 0, \; i \in S$. Hence, we obtain

$v_i = \lim_{n\to\infty}\{\sum_{t=1}^n \sum_j \, p^{t-1}_{ij}(f)r_j(f) + \sum_j \, p^n_{ij}(f)v_j\} \leq \sum_{t=1}^\infty \sum_j \, p^{t-1}_{ij}(f)r_j(f) = v_i(f^\infty), \; i \in S$,

i.e. f^∞ is optimal. □

Define by induction:

$$v^0_i := 0 \text{ and } v^{n+1}_i := max_a \, \{r_i(a) + \sum_j \, p_{ij}(a)v^n_j\}, \; i \in S. \tag{4.44}$$

An MDP is *stable* if $\lim_{n\to\infty} \, v^n_i = v_i$ for all $i \in S$. Hence, in a stable MDP, the value vector can be approximated arbitrary close by value iteration. The next example shows that a convergent MDP is not necessarily stable.

Example 4.5 (continued)
$S = \{1,2\}; \; A(1) = \{1,2\}; \; A(1) = \{1\}; \; p_{11}(1) = 1, \; p_{12}(1) = 0; \; p_{11}(2) = 0, \; p_{12}(2) = 1; \; p_{21}(1) = 0$, $p_{22}(1) = 0; \; r_1(1) = 0; \; r_1(2) = 2; \; r_2(1) = -1$. It is easy to verify that this MDP is convergent. We have seen $v = (1,-1)$. It is also easy to verify that $v^n = (2,-1)$ for all $n \in \mathbb{N}$. Hence, this MDP is not stable.

Theorem 4.31
Positive and negative convergent MDPs are stable.

Proof
Firstly, assume that the MDP is positive. Then,

$v_i = max_a\{r_i(a) + \sum_j \, p_{ij}(a)v_j\} \geq max_a\{r_i(a) + \sum_j \, p_{ij}(a)v^0_j\} = v^1_i \geq v^0_i, \; i \in S.$

Suppose that $v_i \geq v^k_i \geq v^{k-1}_i, \; i \in S$ for some k. Then,

$$\begin{array}{rcl} v_i & = & max_a\{r_i(a) + \sum_j \, p_{ij}(a)v_j\} \geq max_a\{r_i(a) + \sum_j \, p_{ij}(a)v^k_j\} = v^{k+1}_i \\ & \geq & max_a\{r_i(a) + \sum_j \, p_{ij}(a)v^{k-1}_j\} = v^k_i, \; i \in S. \end{array}$$

Hence, by induction, we have $v_i \geq v^{n+1}_i \geq v^n_i, \; i \in S, \; n \in \mathbb{N}$. Since for each i the sequence $\{v^n_i\}$ is monotone nondecreasing and bounded by $v_i < \infty$, $\lim_{n\to\infty} \, v^n_i$ exists, say $v^\infty_i = \lim_{n\to\infty} \, v^n_i, \; i \in S$, and $v^\infty \leq v$. By taking the limit for $n \to \infty$, it follows from $v^{n+1}_i = max_a \, \{r_i(a) + \sum_j \, p_{ij}(a)v^n_j\}$ that $v^\infty_i = max_a \, \{r_i(a) + \sum_j \, p_{ij}(a)v^\infty_j\}, \; i \in S$. Hence, v^∞ is superharmonic and has the property anne, the last property because v^∞ is nonnegative. Since v is the smallest superharmonic vector with the property anne, $v^\infty \geq v$, and we have shown $v_i = v^\infty_i = \lim_{n\to\infty} \, v^n_i, i \in S$, i.e. positive convergent MDPs are stable.
Next, we assume that the MDP is negative. Analogously to the positive case it can be shown that the sequence $\{v^n_i\}$ is monotone non-increasing in n, bounded below by v_i, with limit v^∞_i. Therefore, $v^\infty \geq v$ and satisfies $v^\infty_i = max_a \, \{r_i(a)+\sum_j \, p_{ij}(a)v^\infty_j\}, \; i \in S$. Let $f^\infty \in C(D)$ be such that $v^\infty = r(f)+P(f)v^\infty$. Then, by induction on n,

$$v^\infty = \sum_{t=1}^n \, P^{t-1}(f)r(f) + P^n(f)v^\infty \leq \sum_{t=1}^n \, P^{t-1}(f)r(f), \; n \in \mathbb{N}.$$

Hence, $v \geq v(f^\infty) = \lim_{n\to\infty} \sum_{t=1}^n \, P^{t-1}(f)r(f) \geq v^\infty$, implying $v_i = v^\infty_i = \lim_{n\to\infty} \, v^n_i$, i.e. negative convergent MDPs are stable. □

Remark

Since in negative MDPs every policy is equalizing, f^∞ is optimal if and only if f^∞ is conserving, i.e. $r(f) + P(f)v = v$. Hence, policy f^∞_v is an optimal policy.

4.12 Special models

4.12.1 Red-black gambling

The red-black gambling model was introduced in section 1.3.1. The characteristics of this model are:

$S = \{0, 1, \dots, N\};\ A(0) = A(N) = \{0\},\ A(i) = \{1, 2, \dots, min(i, N-i)\},\ 1 \le i \le N-1.$

For $1 \le i \le N-1, a \in A(i):\ p_{ij}(a) = \begin{cases} p & , j = i+a \\ 1-p & , j = i-a \\ 0 & , j \neq i+a, i-a \end{cases}$ and $r_i(a) = 0$.

$p_{0j}(0) = p_{Nj}(0) = 0, j \in S;\ r_0(0) = 0, r_N(0) = 1.$

The case $p = \frac{1}{2}$ was the subject of Exercise 1.4 in which the reader was asked to show that any $f^\infty \in C(D)$ is an optimal policy. This current section deals with the cases $p > \frac{1}{2}$ and $p < \frac{1}{2}$. In the red-black gambling model every policy is transient (see Exercise 4.8). Hence, we may use the results of section 4.7, e.g. that $v(f^\infty)$ is the unique solution of the linear system $x = r(f) + P(f)x$. In the red-black gambling model this system becomes

$$x_0 = 0;\ x_N = 1;\ x_i = px_{i+f(i)} + (1-p)x_{i-f(i)},\ 1 \le i \le N-1. \tag{4.45}$$

Let f_1^∞ be the *timid policy*, i.e. $f(i) = 1$ for all i. Then it is easy to verify that $v_i(f_1^\infty) = \frac{1-r^i}{1-r^N},\ 0 \le i \le N$, satisfies (4.45).

<u>Case 1: $p > \frac{1}{2}$</u>

In this case we will show that the timid policy f_1^∞ is optimal. For this purpose, it is sufficient to show that $v_i(f_1^\infty) \ge pv_{i+a}(f_1^\infty) + (1-p)v_{i-a}(f_1^\infty),\ (i, a) \in S \times A$.

Because $v_i(f_1^\infty) = \frac{1-r^i}{1-r^N},\ 0 \le i \le N$, we have to show

$1 - r^i \ge p(1 - r^{i+a}) + q(1 - r^{i-a})$, i.e. $-r^i \ge -pr^{i+a} - qr^{i-a}$, which is the same as

$1 \le pr^a + qr^{-a}$. For $F(a) = pr^a + qr^{-a}$, we have $F(1) = p\frac{q}{p} + q\frac{p}{q} = q + p = 1$.

It is sufficient to show that $F(a+1) \ge F(a)$ for all a.

$F(a+1) \ge F(a) \Leftrightarrow pr^{a+1} + qr^{-a-1} \ge pr^a + qr^{-a} \Leftrightarrow pr^{2a+2} + q \ge pr^{2a+1} + qr \Leftrightarrow$

$pr^{2a+1}(r-1) \ge q(r-1) \Leftrightarrow r^{2a+1} \le r \Leftrightarrow r \le 1 \Leftrightarrow p \ge \frac{1}{2}$.

<u>Case 2: $p < \frac{1}{2}$</u>

We will show that in this case *bold play*, i.e. stake $min(i, N-i)$ in state i, is optimal. Therefore we show that in value iteration with starting vector 0, i.e.

$v_i^0 = 0,\ i \in S;\ v_i^{n+1} = max_a\,\{pv_{i+a}^n + (1-p)v_{i-a}^n\},\ 1 \le i \le N-1;\ v_0^{n+1} = 0;\ v_N^{n+1} = 1$ for $n = 0, 1, \dots$.

the bold policy f_b^∞ satisfies $v^{n+1} = L_{f_b}v^n$. Since $v^n \to v$, this implies $v = L_{f_b}v$, i.e. f_b^∞ is an optimal policy.

Let $q = 1 - p$ and let w_{ia}^n be the difference between the action $f_b(i)$ and a in the computation of v_i^{n+1}, i.e.

$$w_{ia}^n = \begin{cases} pv_{2i}^n - pv_{i+a}^n - qv_{i-a}^n & , 1 \le i \le N/2 \quad , \ a \in A(i) \\ p + qv_{2i-N}^n - pv_{i+a}^n - qv_{i-a}^n & , N/2 \le i \le N-1 \quad , \ a \in A(i) \end{cases} \tag{4.46}$$

We have to show that $w_{ia}^n \ge 0$ for all i, a and n. To this end, we show by induction on n:

(1) $w_{ia}^n \ge 0$ for all i, a;

(2) $v^n_{i+a} \geq v^n_i + v^n_a$ for all i, a;

(3) $v^n_N + v^n_j \geq v^n_{N-k} + v^n_{j+k}$ for all j, k;

(4) $v^{n+1}_i = \begin{cases} pv^n_{2i} & , 1 \leq i < N/2 \\ p + qv^n_{2i-N} & , N/2 \leq i \leq N-1 \end{cases}$

For $n = 0$ it is easy to verify that the properties hold. Suppose that the properties are shown for n and consider $n+1$. Because w^n_{ia} has different expressions for the states below and above $N/2$, we have to distinguish between different intervals of the states.

Proof for (1):

For $i + a < N/2$ and $2i < N/2$:

$$w^{n+1}_{ia} = pv^{n+1}_{2i} - pv^{n+1}_{i+a} - qpv^{n+1}_{i-a} = p\{pv^n_{4i} - pv^n_{2(i+a)} - qv^n_{2(i-a)}\} = pw^n_{2i,2a} \geq 0.$$

For $i + a < N/2$ and $2i \geq N/2$:

$$\begin{aligned} w^{n+1}_{ia} &= pv^{n+1}_{2i} - pv^{n+1}_{i+a} - qpv^{n+1}_{i-a} \\ &= p\{p + qv^n_{4i-N} - pv^n_{2(i+a)} - qv^n_{2(i-a)}\} = pw^n_{2i,2a} \geq 0 \end{aligned}$$

For $i + a \geq N/2$ and $i < N/2$:

$$\begin{aligned} w^{n+1}_{ia} &= pv^{n+1}_{2i} - pv^{n+1}_{i+a} - qpv^{n+1}_{i-a} \\ &= p\{p + qv^n_{4i-N} - p - qv^n_{2(i+a)-N} - qv^n_{2(i-a)}\} \\ &= pq\{v^n_{4i-N} - v^n_{2(i+a)-N} - v^n_{2(i-a)}\} \geq 0 \end{aligned}$$

(the nonnegativity by property (2).

For $i + a \geq N/2,\ i \geq N/2,\ i - a < N/2$ and $2i - N < N/2$:

$$\begin{aligned} w^{n+1}_{ia} &= p + qv^{n+1}_{2i-N} - pv^{n+1}_{i+a} - qv^{n+1}_{i-a} \\ &= p + qpv^n_{4i-2N} - p\{p + qv^n_{2(i+a)}\} - qpv^n_{2(i-a)} \\ &= pq\{1 + v^n_{4i-2N} - v^n_{2(i+a)} - v^n_{2(i-a)} \\ &= pq\{v^n_N + v^n_{4i-2N} - v^n_{2(i+a)} - v^n_{2(i-a)} \geq 0 \end{aligned}$$

(the nonnegativity by property (3) with $j = 4i - 2N$ and $k = N - 2(i - a)$).

For $i + a \geq N/2,\ i \geq N/2,\ i - a < N/2$ and $2i - N \geq N/2$:

$$\begin{aligned} w^{n+1}_{ia} &= p + qv^{n+1}_{2i-N} - pv^{n+1}_{i+a} - qv^{n+1}_{i-a} \\ &= p + q\{p + qv^n_{4i-3N}\} - p\{p + qv^n_{2(i+a)-N}\} - qpv^n_{2(i-a)} \\ &= 2pq + q\{qv^n_{4i-3N} - pv^n_{2(i+a)-N} - pv^n_{2(i-a)}\} \\ &\geq pq\{2 + v^n_{4i-3N} - v^n_{2(i+a)-N} - v^n_{2(i-a)}\} \geq 0 \end{aligned}$$

(the nonnegativity because $v^n_{2(i+a)-N} + v^n_{2(i-a)} \leq 2$).

For $i + a \geq N/2,\ i \geq N/2,\ i - a \geq N/2$ and $2i - N < N/2$:

$$\begin{aligned} w^{n+1}_{ia} &= p + qv^{n+1}_{2i-N} - pv^{n+1}_{i+a} - qv^{n+1}_{i-a} \\ &= p + qpv^n_{4i-2N} - p\{p + qv^n_{2(i+a)-N}\} - q\{p + v^n_{2(i-a)-N}\} \\ &= q\{pv^n_{4i-2N} - pv^n_{2(i+a)-N} - qv^n_{2(i-a)-N}\} \\ &\geq qw^n_{2i-N,2a} \geq 0 \end{aligned}$$

For $i + a \geq N/2,\ i \geq N/2,\ i - a \geq N/2$ and $2i - N \geq N/2$:

$$\begin{aligned} w^{n+1}_{ia} &= p + qv^{n+1}_{2i-N} - pv^{n+1}_{i+a} - qv^{n+1}_{i-a} \\ &= p + q\{p + qv^n_{4i-3N}\} - p\{p + qv^n_{2(i+a)-N}\} - q\{p + v^n_{2(i-a)-N}\} \\ &= q\{p + v^n_{4i-3N} - pv^n_{2(i+a)-N} - qv^n_{2(i-a)-N}\} \\ &\geq qw^n_{2i-N,2a} \geq 0 \end{aligned}$$

Proof for (2):

For $i + a < N/2$:

$$v_{i+a}^{n+1} \;=\; pv_{2(i+a)}^n \;\geq\; p\{v_{2i}^n + v_{2a}^n\} \;=\; v_i^{n+1} + v_a^{n+1}.$$

For $i + a \geq N/2$ and $i < N/2$:

$$\begin{aligned} v_{i+a}^{n+1} &= p + qv_{2(i+a)-N}^n \;\geq \text{(because } w_{i+a,i-a}^n \geq 0) \\ &\geq pv_{2i}^n + qv_{2a}^n \;\geq\; pv_{2i}^n + pv_{2a}^n \;=\; v_i^{n+1} + v_a^{n+1}. \end{aligned}$$

For $i + a \geq N/2$ and $i \geq N/2$:

$$\begin{aligned} v_{i+a}^{n+1} &= p + qv_{2(i+a)-N}^n \;\geq\; p + q\{v_{2i-N}^n + v_{2a}^n\} \\ &\geq p + qv_{2i-N}^n + pv_{2a}^n \;=\; v_i^{n+1} + v_a^{n+1}. \end{aligned}$$

Proof for (3):

If $j \geq N - k$: $v_N^n + v_j^n \geq v_N^n + v_{N-k}^n \geq v_{j+k}^n + v_{N-k}^n$.

If $j \geq N - k$ and $j + k \leq N - k$:

For $N/2 \leq j \leq j + k \leq N - k$:

$$v_N^{n+1} + v_j^{n+1} = 1 + p + qv_{2j-N}^n \text{ and } v_{j+k}^{n+1} + v_{N-k}^{n+1} = 2p + q\{v_{2(j+k)-N}^n + v_{2(N-k)-N}^n\}.$$

Hence,

$$v_N^{n+1} + v_j^{n+1} \geq v_{j+k}^n + v_{N-k}^n \;\leftrightarrow\; 1 + v_{2j-N}^n \geq v_{2(j+k)-N}^n + v_{2(N-k)-N}^n,$$

which is true by property (3) (take in (3) $2j - N$ for j and $2k$ for k).

For $j < N/2 \leq j + k \leq N - k$:

$$\begin{aligned} v_{j+k}^{n+1} + v_{N-k}^{n+1} &= 2p + q\{v_{2(j+k)-N}^n + v_{2(N-k)-N}^n\} \text{ (by property (2))} \\ &\leq 2p + qv_{2j}^n \;=\; 2p + (1-p)v_{2j}^n \;=\; 2p + qv_{2j}^n \;=\; 1 + pv_{2j}^n - (1-2p)(1 - v_{2j}^n) \\ &\leq 1 + pv_{2j}^n = v_N^{n+1} + v_j^{n+1}. \end{aligned}$$

For $j \leq j + k < N/2 \leq N - k$:

$$\begin{aligned} v_N^{n+1} + v_j^{n+1} &= 1 + pv_{2j}^n \geq q + p\{1 + v_{2j}^n\} \geq \text{ (take in (3) } 2j \text{ for } j \text{ and } 2k \text{ for } k) \\ &\geq q + p\{v_{2(j+k)}^n + v_{2(N-k)-N}^n\}. \\ v_{j+k}^{n+1} + v_{N-k}^{n+1} &= pv_{2(j+k)}^n + p + q\{v_{2(N-k)-N}^n \leq q + p\{v_{2(j+k)}^n + v_{2(N-k)-N}^n\}. \end{aligned}$$

Hence, $v_N^{n+1} + v_j^{n+1} \geq v_{N-k}^{n+1} + v_{j+k}^{n+1}$.

For $j \leq j + k \leq N - k < N/2$:

This case cannot occur, because $(j + k) + (N - k) = j + N \geq N$.

Proof for (4):

Since $w_{ia}^{n+1} \geq 0$ for all $(i, a) \in S \times A$, the bold actions maximize $pv_{i+a}^{n+1} + (1-p)v_{i-a}^{n+1},\ i \in S$.

Therefore, $v_i^{n+2} = \begin{cases} pv_{2i}^{n+1} & ,1 \leq i < N/2 \\ p + qv_{2i-N}^{n+1} & ,N/2 \leq i \leq N-1 \end{cases}$

4.12.2 Optimal stopping

The optimal stopping model was introduced in section 1.3.3. The characteristics of the model are:

$S = \{1, 2, \dots, N\};\ A(i) = \{1, 2\},\ i \in S;\ r_i(1) = r_i,\ i \in S;\ r_i(2) = -c_i,\ i \in S;$

$p_{ij}(1) = 0,\ i, j \in S;\ p_{ij}(2) = p_{ij},\ i, j \in S.$

In this section we assume that $r_i \geq 0$ and $c_i \geq 0$ for all $i \in S$. Hence, any optimal policy is a transient policy, implying $v = w$. We also have $0 \leq min_j\, r_j \leq v_i \leq max_j\, r_j < \infty$ for all $i \in S$. Therefore, the value vector is finite and, by Theorem 4.17, v is the smallest superharmonic vector, i.e. v is the unique optimal solution of the linear program

$$min \left\{ \sum_j v_j \;\middle|\; \begin{array}{ll} v_i \geq r_i & ,\, i \in S \\ v_i \geq -c_i + \sum_j p_{ij} v_j & ,\, i \in S \end{array} \right\}. \tag{4.47}$$

Consider also the dual linear program

$$max \left\{ \sum_i r_i x_i - \sum_i c_i y_i \;\middle|\; \begin{array}{rcl} x_j + y_j - \sum_i p_{ij} y_i & = & 1,\ j \in S \\ x_i,\, y_i & \geq & 0,\ i \in S \end{array} \right\}. \tag{4.48}$$

Theorem 4.32

Let (x^*, y^*) be an extreme optimal solution of the dual program (4.48). Then, the policy f_*^∞ such that $f_*(i) = \begin{cases} 1 & \text{if } x_i^* > 0 \\ 2 & \text{if } x_i^* = 0 \end{cases}$ is an optimal policy.

Proof

If $x_j^* = 0$, then it follows from $x_j^* + y_j^* = 1 + \sum_i p_{ij} y_i^* \geq 1 > 0$ that $y_j^* > 0$. Since (x^*, y^*) is an extreme point, there are at most N positive components. Hence, for each state $i \in S$, we have either $x_j^* > 0$ or $y_j^* > 0$. Furthermore, for $z^* = \begin{cases} x_i^* & \text{if } x_i^* > 0 \\ y_i^* & \text{if } x_i^* = 0 \end{cases}$ we obtain $z^* = e^T + (z^*)^T P(f_*)$.

Iterating this equality gives $z^* = \sum_{t=1}^n e^T P^{t-1}(f_*) + (z^*)^T P^n(f_*) \geq \sum_{t=1}^n e^T P^{n-1}(f_*)$ for all $n \in \mathbb{N}$. Hence, f_*^∞ is transient and $\{I - P(f_*)\}^{-1} = \sum_{t=1}^\infty P^{t-1}(f_*)$. The complementary slackness of linear programming gives: $v = r(f_*) + P(f_*)v$, i.e. $v = \{I - P(f_*)\}^{-1} r(f_*) = \sum_{t=1}^\infty P^{t-1}(f_*) r(f_*) = v(f_*^\infty)$, i.e. f_*^∞ is an optimal policy. □

Algorithm 4.11 *Linear programming algorithm for optimal stopping*
Input: Instance of an optimal stopping problem
Output: Optimal deterministic policy f^∞.

1. Use the simplex method to compute optimal solutions v^* and (x^*, y^*) of the dual pair of linear programs:

$$min \left\{ \sum_j v_j \;\middle|\; \begin{array}{ll} v_i \geq r_i & ,\, i \in S \\ v_i \geq -c_i + \sum_j p_{ij} v_j & ,\, i \in S \end{array} \right\}$$

   and

$$max \left\{ \sum_i r_i x_i - \sum_i c_i y_i \;\middle|\; \begin{array}{rcl} x_j + y_j - \sum_i p_{ij} y_i & = & 1,\ j \in S \\ x_i,\, y_i & \geq & 0,\ i \in S \end{array} \right\}.$$

2. Take $f_*^\infty \in C(D)$ such that $f_*(i) = \begin{cases} 1 & \text{if } x_i^* > 0; \\ 2 & \text{if } x_i^* = 0. \end{cases}$

   v^* is the value vector and f_*^∞ an optimal policy (STOP).

Remark

An optimal stopping problem may be considered as a special case of the replacement problem that is discussed in section 8.1.1. In that section it is shown that an $\mathcal{O}(N^3)$ of Algorithm 4.11 is possible.

Example 4.12

$S = \{1, 2, 3, 4, 5\}$. The stopping rewards are: $r_1 = 0,\ r_2 = 2,\ r_3 = 2,\ r_4 = 3$ and $r_5 = 0$ and there are no costs for the continuing action ($c_i = 0,\ 1 \leq i \leq 5$). The states 1 and 5 are absorbing; in state i $(2 \leq i \leq 4)$ there is a probability $\frac{1}{2}$ to go to state $i + 1$ and a probability $\frac{1}{2}$ to go to state $i - 1$. The dual LP program is:

$$\begin{array}{l}
max\ 2x_2 + 2x_3 + 3x_4 \\
\text{subject to:}
\end{array}$$

$$\begin{array}{ccccccccccccccc}
x_1 & & & & & - & \frac{1}{2}y_2 & & & & & = & 1 \\
 & x_2 & & & & + & y_2 & - & \frac{1}{2}y_3 & & & = & 1 \\
 & & x_3 & & & - & \frac{1}{2}y_2 & + & y_3 & - & \frac{1}{2}y_4 & = & 1 \\
 & & & x_4 & & & & - & \frac{1}{2}y_3 & + & y_4 & = & 1 \\
 & & & & x_5 & & & & & - & \frac{1}{2}y_4 & = & 1 \\
\multicolumn{11}{r}{x_1,\ x_2,\ x_3,\ x_4,\ x_5,\ y_2,\ y_3,\ y_4} & \geq & 0
\end{array}$$

The optimal solution of the problem is:
$x_1 = 1,\ x_2 = \frac{3}{2},\ x_3 = 0,\ x_4 = \frac{3}{2};\ x_5 = 1;\ y_2 = 0;\ y_3 = 1$ and $y_4 = 0$. Hence, the optimal policy is: continue in state 3 and stop in the other states. The expected total reward is: $v_1 = 0,\ v_2 = 2,\ v_3 = 2\frac{1}{2},\ v_4 = 3$ and $v_5 = 0$.

Let $S_0 = \{i \in S \mid r_i \geq -c_i + \sum_j p_{ij}r_j\}$, i.e. S_0 is the set of states in which immediate stopping is not worse than continuing for one period and than choose to stop. The set S_0 follows directly from the data of the model. An optimal stopping problem is *monotone* if $p_{ij} = 0$ for all $i \in S_0,\ j \notin S_0$.

Theorem 4.33

In a monotone optimal stopping problem a one-step look-ahead policy, i.e. a policy that stops in the states of S_0 and continues outside S_0, is an optimal policy.

Proof

Let v be the value vector of the optimal stopping problem. Define w by $w_i := \begin{cases} r_i, & i \in S_0; \\ v_i, & i \notin S_0. \end{cases}$

The value vector is the solution of the optimality equation: $v_i = max\{r_i, -c_i + \sum_j p_{ij}v_j\},\ i \in S$. Therefore, $w \leq v$. Furthermore, w is feasible for the LP problem, namely:

If $i \in S_0:\ w_i = r_i \geq -c_i + \sum_j p_{ij}r_j = -c_i + \sum_{j \in S_0} p_{ij}r_j = -c_i + \sum_{j \in S_0} p_{ij}w_j = -c_i + \sum_j p_{ij}w_j$.

If $i \notin S_0:\ w_i = v_i \geq -c_i + \sum_j p_{ij}v_j \geq -c_i + \sum_j p_{ij}w_j$.

It is obvious that $w_i \geq r_i,\ i \in S$. Because v is the smallest solution of the LP problem, we have $v = w$, i.e. $v_i = r_i,\ i \in S_0$, i.e. the stopping action is in S_0 optimal. If $i \notin S_0$, then we obtain, $r_i < -c_i + \sum_j p_{ij}r_j \leq -c_i + \sum_j p_{ij}v_j \leq v_i$: continue outside S_0 is optimal. □

Example 4.13

N different real numbers are drawn, one by one. The second number has a probability of $\frac{1}{2}$ to come on the right of the first number on the line of the real numbers (also a probability of $\frac{1}{2}$ to come on the left of the first number). The third number has a probability of $\frac{1}{3}$ to come in each of the three intervals on the right line, where the intervals are generated by the first two numbers. Etc. After each draw there are two possibilities: the last draw is the largest up to now or this is not the case. Only when the last number is the largest up to now we have the option to stop with as reward that largest number. If the last number is not the largest up to now or when we don't use the option to stop when the last number is the largest, we have to continue, unless all N numbers are drawn. Which policy maximizes the probability to stop with the largest of all N numbers?

We make the following model for this problem. Let $S = \{1, 2, \dots, N\}$, where state i means that the i-th draw is the largest up to now. $A(i) = \{1, 2\},\ 1 \le i \le N-1;\ A(N) = \{1\};\ c_i = 0,\ i \in S$. As r_i we take the probability that, given that the i-th draw gives the largest number of the first i numbers, it is the largest number of all N numbers. The probability that the $(i+1)$-th number is the largest number of the first $i+1$ numbers is $\frac{1}{i+1}$; the probability that the $(i+2)$-th number is the largest of the first $i+2$ numbers is $\frac{1}{i+2}$, etc. Hence,

$$r_i = \Big(1 - \tfrac{1}{i+1}\Big)\Big(1 - \tfrac{1}{i+2}\Big) \cdots \Big(1 - \tfrac{1}{N}\Big) = \tfrac{i}{N}.$$

The transition probabilities are:

$$\begin{aligned}
p_{ij} &= \text{the probability that the numbers } (i+1) \text{ up to and including number } (j-1) \text{ are} \\
&\ \text{smaller than number } i \text{ and number } j \text{ is larger that number } i,\ j \ge i+1. \\
&= \Big(1 - \tfrac{1}{i+1}\Big)\Big(1 - \tfrac{1}{i+2}\Big) \cdots \Big(1 - \tfrac{1}{j-1}\Big) \cdot \tfrac{1}{j} = \tfrac{i}{(j-1)j}. \\
S_0 &= \{i \in S \mid r_i \ge -c_i + \textstyle\sum_j p_{ij} r_j\} = \{i \in S \mid \tfrac{i}{N} \ge \textstyle\sum_{j=i+1}^N \tfrac{i}{(j-1)j} \cdot \tfrac{j}{N}\} \\
&= \{i \in S \mid \tfrac{1}{i} + \tfrac{1}{i+1} + \cdots + \tfrac{1}{N-1} \le 1\}.
\end{aligned}$$

Because $\frac{1}{i} + \frac{1}{i+1} + \cdots + \frac{1}{N-1}$ is monotone decreasing in i, we have $S_0 = \{i \in S \mid i \ge i_*\}$, where i^* is defined by $i_* := min\{i \mid \frac{1}{i} + \frac{1}{i+1} + \cdots + \frac{1}{N-1} \le 1\}$. Because obviously S_0 is closed, the problem is monotone and therefore the optimal policy chooses the stopping action as soon as i_* drawn are made and that draw results in the largest number up to now. The value vector can be computed as follows.

If $i \ge i_*$: $v_i = r_i = \frac{i}{N}$.

If $i < i_*$: $v_i = -c_i + \sum_j p_{ij} v_j = \sum_{j=i+1}^N \frac{i}{(j-1)j} \cdot v_j = i \cdot \sum_{j=i+1}^N \frac{1}{(j-1)j} \cdot v_j$.

For $2 \le i \le i_* - 1$, we have:

$$\begin{aligned}
v_{i-1} &= (i-1) \cdot \textstyle\sum_{j=i}^N \tfrac{1}{(j-1)j} \cdot v_j = (i-1) \cdot \big\{\tfrac{1}{i(i-1)} v_i + \textstyle\sum_{j=i+1}^N \tfrac{1}{(j-1)j} \cdot v_j\big\} \\
&= \tfrac{1}{i} v_i + \tfrac{i-1}{i} v_i = v_i.
\end{aligned}$$

Therefore, we obtain

$$\begin{aligned}
v_1 &= v_2 = \cdots = v_{i_*-1} = (i_* - 1) \cdot \textstyle\sum_{j=i_*}^N \tfrac{1}{(j-1)j} \cdot v_j \\
&= (i_* - 1) \cdot \textstyle\sum_{j=i_*}^N \tfrac{1}{(j-1)j} \cdot \tfrac{j}{N} = \tfrac{i_*-1}{N} \cdot \textstyle\sum_{j=i_*}^N \tfrac{1}{j-1}.
\end{aligned}$$

Example 4.14

Problems like searching a target can often be modeled as an optimal stopping problem. Suppose that we are searching for an object with value r and that there is an a priori probability p that the object is in the search area. If we search in this area, for each search there are searching costs c and there is a probability β that we find the object, if it is in this area. A maximum of N searches is allowed. Of course, when the object is found, then we stop; but if the object is not found, will we do another search?

Let $S = \{0, 1, \dots, N-1\}$, where state i means that i failured searches have been done.

In state i, we have the posteriori probability $p_i = \frac{p(1-\beta)^i}{p(1-\beta)^i + (1-p)}$ that the object is present.

Hence, we obtain $r_i = 0$ and $c_i = c - p_i \cdot \beta \cdot r,\ i \in S$, and $p_{ij} = \begin{cases} 1 - p_i \cdot \beta & ,\ i \in S,\ j = i+1; \\ 0 & ,\ i \in S,\ j \ne i+1. \end{cases}$

$S_0 = \{i \mid r_i \ge -c_i + \sum_j p_{ij} r_j\} = \{i \mid c_i \ge 0\} = \{i \mid p_i \le \frac{c}{\beta \cdot r}\}$.

It is easy to verify that $p_0 \geq p_1 \geq \cdots \geq p_{N-1}$. Hence, $S_0 = \{i \mid i \geq i_*\}$, where $i_* := min\{ \mid p_i \leq \frac{c}{\beta \cdot r}\}$. It is obvious that S_0 is closed. Therefore, to stop in S_0 and continue outside S_0 is optimal.

This result is intuitively clear: S_0 consists of the states where the expected netto costs (c_i) are nonnegative. A formula for the stopping states can also be given in the original data, namely

$$\begin{array}{rclcl} p_i \leq \frac{c}{\beta\cdot r} & \Leftrightarrow & \frac{p(1-\beta)^i}{p(1-\beta)^i+(1-p)} \leq \frac{c}{\beta\cdot r} & \Leftrightarrow & p(1-\beta)^i \leq \frac{(1-p)c}{\beta\cdot r-c} \\ & \Leftrightarrow & p(1-\beta)^i \leq \frac{(1-p)c}{\beta\cdot r-c} & \Leftrightarrow & (1-\beta)^i \leq \frac{1-p}{p}\cdot\frac{c}{\beta\cdot r-c} \\ & \Leftrightarrow & i \geq \frac{log\left\{\frac{1-p}{p}\cdot\frac{c}{\beta\cdot r-c}\right\}}{log\,(1-\beta)} & = & i_*. \end{array}$$

4.13 Bibliographic notes

The study of Markov decision models with the expected total reward criterion originated with the book *How to gamble if you must* by Dubins and Savage ([75]), with appeared in 1965. The name *contracting* was introduced by Van Nunen and Wessels, who have studied this model systematically (cf. [302] and [305]. The concept of *excessive* MDPs was introduced by Hordijk ([123], p. 5). Veinott ([311]) introduced *transient* policies. In [243] and [244] Rothblum has studied *normalized* MDPs. He has generalized the Miller-Veinott ([199]) policy iteration algorithm for finding a deterministic policy that maximizes the expected discounted reward for all discount factors close enough to 1.

Many properties on square matrices, eigenvalues and spectral radius can be found in books on linear algebra and linear operators (e.g. [163]). Books on nonnegative matrices and Markov (decision) chains are written by Seneta ([261]) and Rothblum ([247]).

The linear program (4.15) and the correspondence with the stationary and deterministic policies as given in Theorem 4.7 was derived by Kallenberg ([148]).

Lemma 4.5 is due to Hordijk and Tijms ([132]). The equivalence results presented in Theorem 4.8, Theorem 4.9, Theorem 4.10 and Theorem 4.11 are based on papers written by Veinott ([311]), Rothblum ([243] and [244]), and Hordijk and Kallenberg ([123], [148] and [129]). A related paper is [77]. Algorithm 4.1 and the observations mentioned in Remark 1 and Remark 2 are due to Kallenberg ([148]).

Theorem 4.12 is based on a fundamental paper written Blackwell in 1962 ([29]). Blackwell called such a policy *1-optimal*. Later, this property was called *Blackwell optimal* in honor to Blackwell. The concept p-summable and Theorem 4.13 have its roots in [148]. The proof of Theorem 4.13 follows the line of reasoning in the proof of Theorem 6.1 in [236]. The concept *conserving* was presented by Dubins and Savage ([75]). Example 4.5 appeared in [300].

The material of Section 4.6 with the computation of an optimal transient policy was developed by Hordijk and Kallenberg ([148], [129]). The results in Section 4.7 are contributed by Van Nunen and Wessels ([302], [305]), and by Hordijk and Kallenberg ([148], [129]).

The treatment in Section 4.8 of a finite horizon MDP as a transient MDP with the special simplex algorithms (Algorithms 4.7 and 4.8) was proposed by Kallenberg ([147]). A related paper is [124].

Seminal papers on positive and negative MDPs are [30] and [285]. The sections 4.9 and 4.10 deal with linear programming and follow Kallenberg ([148], section 3.5 and 3.6). For value iteration we refer to Van der Wal ([297]) and for policy iteration to Puterman ([227]). References for convergent MDPs are Hordijk ([121], [122], [123] and Van der Wal ([297]).

Gambling theory can be found in [75]. For the proof of the optimality of the timid policy (case $p > \frac{1}{2}$) we refer to [238]. The proof for the bold policy (case $p < \frac{1}{2}$) is based on unpublished work of Denardo [62].

4.14 Exercises

Exercise 4.1
Consider the following model.
$S = \{1,2\}$; $A(1) = \{1,2\}$, $A(2) = \{1\}$; $p_{11}(1) = 1$, $p_{12}(1) = 0$; $p_{11}(2) = 0$, $p_{12}(2) = 1$; $p_{21}(1) = 0$, $p_{22}(1) = 0.5$; $r_1(1) = r_1(2) = r_2(1) = 1$. Define the sequence of stationary Markov policies $\pi^\infty(n)$ by

$$\pi_{1a}(n) := \begin{cases} 1-\frac{1}{n} & a=1 \\ \frac{1}{n} & a=2 \end{cases} \quad \text{and } \pi_{21}(n) := 1,\ n = 1,2,\dots.$$

Prove that $v_1\big(\pi^\infty(n)\big) < \infty$ for $n = 1,2,\dots$ and $sup_n\, v_1\big(\pi^\infty(n)\big) = +\infty$.

Exercise 4.2
A policy R is called *stopping* if $\lim_{t\to\infty} \mathbb{P}_{i,R}\{X_t = j, Y_t = a\} = 0$ for all i, j and a.

a. Show that any transient policy is stopping.

b. Consider the model $S = \{1\}$; $A(1) = \{1,2\}$; $p_{11}(1) = 1$; $p_{11}(2) = 0.5$ with policy R that takes action 2 at the time points $t = 2^n$, $n = 1,2,3,\dots$. Show that R is stopping, but not transient.

c. Show that a stationary policy π^∞ is transient if and only if π^∞ is stopping.

Exercise 4.3
Prove case 3 of Lemma 4.5.

Exercise 4.4
Consider the linear program $max \left\{ \sum_{(i,a)} x_i(a) \;\middle|\; \begin{array}{rl} \sum_{(i,a)} \{\delta_{ij} - p_{ij}(a)\} x_i(a) & \le \beta,\ j \in S \\ x_i(a) & \ge 0,\ (i,a) \in S \times A \end{array} \right\}$,
where $\beta_j > 0$, $j \in S$, are arbitrarily chosen numbers.

a. Prove that if the MDP is contracting, then this linear program has a finite optimal solution.

b. Prove that if this linear program has a finite optimal solution, then the MDP is contracting.

c. Show that an MDP is contracting if and only if there exists a solution to the system

$$\begin{cases} \mu_i = max_a\, \{1 + \sum_j p_{ij}(a)\mu_j\} & ,\ i \in S \\ \mu_i \ge 0 & ,\ i \in S \end{cases}$$

Exercise 4.5
Consider a contracting MDP. An action $a \in A(i)$ is *suboptimal* if $r_i(a) + \sum_j p_{ij}(a)v_j < v_i$.
Consider the dual linear program in Algorithm 4.5 and let $f^\infty \in C(D)$ be the policy corresponding to some simplex tableau in which the x-variables have values $x_i^f(a)$, $(i,a) \in S \times A$.
Let $d_i^f(a)$ be the value of the dual variable which corresponds to $x_i^f(a)$, $(i,a) \in S \times A$.
Show the following properties:

a. If $d_i^f(a_i) > min_a\, d_i^f(a) + \sum_j p_{ij}(a_i)\{b_j - v_j(f^\infty)\}$, where b is an upper bound of the value vector v, then action a_i is an suboptimal action.

b. $b := v(f^\infty) - \frac{min_{(i,a)}\{d_i^f(a)/\mu_i\}}{1-\alpha} \cdot \mu$ is an upper bound of the value vector v, where α and μ are such that $\mu_i > 0$, $i \in S$, $\alpha \in [0,1)$ and $\sum_j p_{ij}(a)\mu_j \le \alpha \cdot \mu_i$ for all $(i,a) \in S \times A$.

Exercise 4.6
Consider the following model:
$S = \{1,2,3,4,5,6,7\}$; $A(1) = A(2) = \{1,2\}$, $A(3) = \{1,2,3\}$, $A(4) = \{1,2\}$, $A(5) = \{1,2,3\}$, $A(6) = \{1\}$, $A(7) = \{1,2\}$; $p_{11}(1) = 1$, $p_{13}(2) = 1$, $p_{21}(1) = 1$, $p_{24}(2) = 1$, $p_{33}(1) = 0.5$,

$p_{31}(2) = 1,\ p_{37}(3) = 1,\ p_{43}(1) = 1,\ p_{42}(1) = 1,\ p_{54}(1) = 0.5,\ p_{53}(2) = 1,\ p_{56}(1) = 1,$
$p_{67}(1) = 0.5,\ p_{77}(1) = 0.5,\ p_{76}(2) = 1$ (the other transition probabilities are zero);
$r_1(1) = 0,\ r_1(2) = 0,\ r_2(1) = 0,\ r_2(2) = 2,\ r_3(1) = 1,\ r_3(2) = 1,\ r_3(3) = 1,\ r_4(1) = 1,$
$r_4(2) = 1,\ r_5(1) = 1,\ r_5(2) = 2,\ r_5(3) = 3,\ r_6(1) = 1,\ r_7(1) = 1,\ r_7(2) = 1.$
Use Algorithm 4.9 to determine an optimal policy. Take $\beta_i = \frac{1}{7},\ i \in S$.

Exercise 4.7
Consider the following model:
$S = \{1, 2, 3\};\ A(1) = \{1, 2\},\ A(2) = A(3) = \{1\};\ p_{11}(1) = p_{12}(2) = p_{23}(1) = p_{33}(1) = 1$
(the other transition probabilities are zero); $r_1(1) = 0,\ r_1(2) = 2,\ r_2(1) = -1,\ r_3(1) = 0.$

a. Is this model convergent?
b. Determine the value vector v and show that v has the property anne.
c. Compute the value vector and an optimal policy by linear programming.
d. What happens in value iteration, given by (4.44)?
e. Is the problem stable?

Exercise 4.8
Show that the red-black gambling model is transient.

Exercise 4.9
Consider an optimal stopping problem with the data:
$S = \{1, 2, 3, 4\};\ r_1 = 0,\ r_2 = 1,\ r_3 = 2,\ r_4 = 2;\ c_i = 0,\ 1 \le i \le 4.$
$p_{11} = \frac{1}{2},\ p_{12} = \frac{1}{2},\ p_{13} = 0,\ p_{14} = 0;\ p_{21} = \frac{1}{8},\ p_{22} = \frac{1}{8},\ p_{23} = \frac{1}{2},\ p_{24} = \frac{1}{4};$
$p_{31} = \frac{1}{3},\ p_{32} = \frac{1}{3},\ p_{33} = \frac{1}{3},\ p_{34} = 0;\ p_{41} = \frac{1}{4},\ p_{42} = \frac{1}{8},\ p_{43} = \frac{1}{2},\ p_{44} = \frac{1}{8}.$
Determine an optimal policy for this problem.

Exercise 4.10
Every night a thief is going for robbery and he will capture an amount of k with probability p_k for $k = 0, 1, \dots, n$. The probability to be caught is equal to p, and if he is caught, he will loose the total captures of all previous nights and he must stop. At which captured amount, the thief will stop with robbery? Show that the solution of this problem is: the thief stops as soon as he has captured the amount $\frac{1-p}{p} \cdot \sum_{j=0}^{n} p_j \cdot j$ and gives an intuitive explanation of this result.
Hint:
Use as state space $S = \{0, 1, \dots\}$, where state i means that the thief has i as total amount and that he is not yet caught. Assume that the results of the optimal stopping problem are also true for this infinite state space.

Exercise 4.11 *Optimal stopping problem*
Consider a person who wants to sell an asset for which he is offered an amount of money at the beginning of each week. We assume that these offers are independent and that an offer of amount j will be made with probability $p_j,\ 0 \le j \le N$. He has to decide immediately either to accept or to reject the offer. If the offer is not accepted, the offer is lost and a cost c is occurred. Which policy will maximize the expected total income?

a. Formulate this problem as an optimal stopping problem.
b. Show that this problem has an optimal control-limit policy.

Exercise 4.12 *How to serve in tennis*

Consider the example *How to serve in tennis* as described in section 1.3.2. Let $v(i, j, s)$ be the probability of winning the next game in tennis when the score is (i, j) and s is the number of the service which is due to the server ($s = 1$ or $s = 2$); let $v(4) = 1$ and $v(5) = 0$. Since winning (loosing) a point will not decrease (increase) the probability of winning a game, we have the relations:
$v(i+1, j, 1) \geq v(i, j, 1)$; $v(i+1, j, 1) \geq v(i, j, 2)$; $v(i, j, 1) \geq v(i, j, 2)$; $v(i, j, 2) \geq v(i, j+1, 1)$.

a. Give the optimality equation for this model.

b. Show the optimality of the policy, which is mentioned in section 1.3.2 as the optimal policy.

Chapter 5

Average reward - general case

5.1 Introduction

When decisions are made frequently, so that the discount rate is very close to 1, or when performance criterion cannot easily be described in economic terms with discount factors, the decision maker may prefer to compare policies on the basis of their average expected reward instead of their expected total discounted reward. Consequently, the average reward criterion occupies a cornerstone of queueing control theory especially when applied to controlling computer systems and communication networks. In such systems, the controller makes frequent decisions and usually assesses system performance on the basis of throughput rate or the average time a job remains in the system. This optimality criterion may also be appropriate for inventory systems with frequent restocking decisions.

In the criterion of average reward the limiting behavior of $\frac{1}{T}\sum_{t=1}^{T} r_{X_t}(Y_t)$ is considered for $T \to \infty$. Since $lim_{T\to\infty}\frac{1}{T}\sum_{t=1}^{T} r_{X_t}(Y_t)$ may not exist and interchanging limit and expectation is not allowed in general, there are four different evaluation measures which can be considered:

1. Lower limit of the average expected reward:

   $\phi_i(R) = \liminf_{T\to\infty} \frac{1}{T}\sum_{t=1}^{T} \mathbb{E}_{i,R}\{r_{X_t}(Y_t)\},\ i \in S$, with value vector $\phi = sup_R\ \phi(R)$.

2. Upper limit of the average expected reward:

   $\overline{\phi}_i(R) = \limsup_{T\to\infty} \frac{1}{T}\sum_{t=1}^{T} \mathbb{E}_{i,R}\{r_{X_t}(Y_t)\},\ i \in S$, with value vector $\overline{\phi} = \sup_R\ \overline{\phi}(R)$.

3. Expectation of the lower limit of the average reward:

   $\psi_i(R) = \mathbb{E}_{i,R}\{\liminf_{T\to\infty} \frac{1}{T}\sum_{t=1}^{T}\ r_{X_t}(Y_t)\},\ i \in S$, with value vector $\psi = sup_R\ \psi(R)$.

4. Expectation of the upper limit of the average reward:

   $\overline{\psi}_i(R) = \mathbb{E}_{i,R}\{\limsup_{T\to\infty} \frac{1}{T}\sum_{t=1}^{T}\ r_{X_t}(Y_t)\},\ i \in S$, with value vector $\overline{\psi} = sup_R\ \overline{\psi}(R)$.

As already mentioned in Section 1.2.2, these four criteria are equivalent in the sense that an optimal deterministic policy for one criterion is also optimal for the other criteria. We will use criterion 1, the lower limit of the average expected reward.

In this chapter we start with the classification of MDP models on the basis of the chain structure. Because the average reward criterion depends on the limiting behaviour of the underlying stochastic processes, this structure is of interest. In the subsequent section the stationary matrix, the fundamental matrix and the deviation matrix of a Markov chain is discussed. These matrices play an important role in the average reward criterion and also in more sensitive criteria. The most sensitive criteron is Blackwell optimality. Laurent series expansion relates the average reward to the total discounted reward. This is the subject of section 5.5. The last sections of this chapter deal with the optimality equation and with the methods of policy iteration, linear programming and value iteration.

5.2 Classification of MDPs

5.2.1 Definitions

There are several ways to classify MDPs. The first one distinguishes between *communicating* and *non-communicating*. An MDP is communicating if for every $i, j \in S$ there exists a policy $f^\infty \in C(D)$, which may depend on i and j, such that in the Markov chain $P(f)$ state j is accessible from state i. An MDP is *weakly communicating* if $S = S_1 \cup S_2$, where $S_1 \cap S_2 = \emptyset$, S_1 is a closed communicating set under *some* policy $f^\infty \in C(D)$ and S_2 is a (possibly empty) set of states which are transient under *all* policies.

A second kind of classification concerns the ergodic structure. One distinguish between *irreducible, unichain* and *multichain* MDPs. An MDP is irreducible (also called *completely ergodic*) if the Markov chain $P(f)$ is irreducible for every $f^\infty \in C(D)$. An MDP is a unichain MDP if the Markov chain $P(f)$ is a unichain Markov chain (exactly one ergodic class plus a possibly empty set of transient states) for every $f^\infty \in C(D)$. An MDP is multichain if there exists a policy $f^\infty \in C(D)$ for which the Markov chain $P(f)$ has (at least) two ergodic classes.

The next result is obvious.

Lemma 5.1

An irreducible MDP is communicating and unichain.

5.2.2 Classification of Markov chains

For a single Markov chain it is easy to determine whether or not the Markov chain belangs to a certain class. Easy means polynomially solvable, i.e. the problem belongs in terms of the complexity theory to the class $\mathcal{P}$ of problems solvable in polynomial-time.
Consider a Markov chain with transition matrix P. The classification of the Markov chain can be executed in the *associated directed graph* $G(P) = \big(V(P), A(P)\big)$, where the nodes $V(P)$ are the states of the Markov chain and the arcs of $A(P)$ satisfy $A(P) = \{(i,j) \mid p_{ij} > 0\}$.
Since a strongly connected component of $G(P)$ is closed if and only if the corresponding states in the Markov chain are an ergodic class, the following algorithm determines the ergodic classes $E_1, E_2, \dots, E_m$ and the set T of transient states.

Algorithm 5.1 *Ergodic classes and transient states of a Markov chain*
Input: A Markov chain.
Output: The ergodic sets $E_1, E_2, \dots, E_m$ and the set T of transient states.

```
1. a. Determine the strongly connected components of $G(P)$, say $C_1, C_2, \dots, C_n$.
   b. $m; = 0;\ T := \emptyset$.
2. for $i = 1, 2, \dots, n$ do
      if $C_i$ is closed then begin $m := m + 1;\ E_m := C_i$ end
      else $T := T \cup C_i$
```

The determination of the strongly connected components of a graph can be done in $\mathcal{O}(p) = \mathcal{O}(N^2)$, where p is the number of arcs of the graph (see [287]). For the examination whether the strongly connected components are closed or open, it is also sufficient to consider the arcs of the graph. Therefore, Algorithm 5.1 has complexity $\mathcal{O}(N^2)$.

5.2.3 Classification of Markov decision chains

An MDP has $\prod_{i \in S} |A(i)|$ different deterministic policies and each policy induces a Markov chain. Therefore, MDPs are also called *Markov decision chains.* The approach to analyse all Markov chains separately is prohibitive. The problem to determine whether or not an MDP belongs to a certain class is a combinatorial problem. It turns out that all classification problems are easy, i.e. polynomially solvable, except one. Checking the unichain condition is an $\mathcal{NP}$-hard problem.

For the analysis of the chain structure we use two directed graphs, G_1 and G_2, both with as node set the states of the MDP. In $G_1 = (S, A_1)$ the arc set $A_1 = \{(i,j) \mid p_{ij}(a) > 0 \text{ for every } a \in A(i)\}$. Hence, a path from i to j in G_1 means that state j is accessible from state i under *every* policy. In $G_2 = (S, A_2)$ the arc set $A_2 = \{(i,j) \mid p_{ij}(a) > 0 \text{ for some } a \in A(i)\}$, and a path from i to j in G_2 means that state j is accessible from state i under *some* policy. Let $M := \sum_{i \in S} |A(i)|$, then the construction of the graphs G_1 and G_2 has complexity $\mathcal{O}\big(\sum_{j \in S}\{\sum_{i \in S} |A(i)|\}\big) = \mathcal{O}(M \cdot N)$.

Communicating

The question whether or not an MDP is communication is solved by the following lemma.

Lemma 5.2
An MDP is communicating if and only if the graph G_2 is strongly connected.

Proof

$\Rightarrow$ Suppose that G_2 is not strongly connected, i.e. there are nodes i and j such that in G_2 is no path from i to j. This implies that for every $f^\infty \in C(D)$ in the Markov chain $P(f)$ state j is not accessible from state i. Consequently, the MDP is not communicating.

$\Leftarrow$ Suppose that G_2 is strongly connected and take any pair $i, j \in S$. Since G_2 is strongly connected there is a path from i to j. Hence, j is accessible from i under some policy. This implies the property communicating. □

The above Lemma implies the following algorithm for checking the communicating property of an MDP. Since the construction of G_2 has complexity $\mathcal{O}(M \cdot N)$, and the determination of the strongly connected components is of order $N^2 \leq M \cdot N$, the total complexity is $\mathcal{O}(M \cdot N)$.

Algorithm 5.2 *Checking the communicating property of an MDP*
Input: A Markov decision problem.
Output: The property 'communicating' or the property 'noncommunicating'.

```
1. Construct the graph $G_2$.
2. Determine the strongly connected components of $G_2$, say $C_1, C_2, \dots, C_n$.
3. if $n = 1$ then the MDP is communicating (STOP)
   else the MDP is noncommunicating (STOP)
```

If the outcome of Algorithm 5.2 is 'noncommunicating' ($n \geq 2$) one may ask whether the MDP is perhaps weakly communicating. If two or more of the strongly connected components are closed, then the MDP is not weakly communicating, since in that case there are two disjunct sets of states which both are ergodic under all policies. If only one of the strongly connected components is closed, say C_1, one can try to find a state outside C_1, say state i, for which there is a positive transition probability to C_1 under all actions $a \in A(i)$. If such state does not exist, then there is a policy with the property that starting outside C_1 one never enters C_1. Hence, the MDP is not weakly communicating. Continuing in this way yields the following algorithm.

Algorithm 5.3 *Checking the communicating and weakly communicating property of an MDP*
Input: A Markov decision problem.
Output: The property 'communicating' and if the MPD is 'noncommunicating' we obtain either the property 'weakly communicating' or the property 'not weakly communicating'.

```
1. Construct the graph $G_2$.
2. a. Determine the strongly connected components of $G_2$, say $C_1, C_2, \dots, C_n$.
   b. $m := 0$; $T := \emptyset$.
3. for $i = 1, 2, \dots, n$ do
      if $C_i$ is closed then begin $m := m + 1$; $E_m := C_i$ end
      else $T := T \cup C_i$
```

```
4. **if** $m \geq 2$ **then** the MDP is not weakly communicating (STOP)
   **else if** $T = \emptyset$ **then** the MDP is communicating (STOP)
        **else go to** step 5
5. a. $S_1 := E_1,\ S_2 := \emptyset$.
   b. **repeat**
        $k := 0$;
        **for every** $i \in T$ **do**
            **if** $\sum_{j \in S_1 \cup S_2} p_{ij}(a) > 0$ **for every** $a \in A(i)$ **then begin** $S_2 := S_2 \cup \{i\};\ T := T \backslash \{i\};\ k := 1$
   **end**
        **until** $k = 0$
6. **if** $T = \emptyset$ **then** the MDP is weakly communicating (STOP)
   **else** the MDP is not weakly communicating.
```

For the complexity of Algorithm 5.3 we remark that the steps 1 until 4 are executed only once and have complexity $\mathcal{O}(M \cdot N)$. Step 5 is executed at most N times and each step has complexity of order $\sum_{i \in T} \sum_{a \in A(i)} |S_1 \cup S_2| \leq M \cdot N$. Hence, the complexity of step 5, and also the overall complexity of the algorithm is $\mathcal{O}(M \cdot N^2)$.

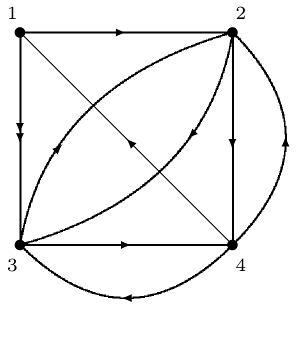
Irreducibility

For the irreducibility we use graph G_1. If G_1 is strongly connected, then the MDP is irreducible, because each pair of states communicates under every policy. If G_1 is not strongly connected we *condense* graph G_1 to graph G_1^c. The condensed graph G_1^c has a (compound) vertex for each strongly connected component of G_1. Let i_k and i_l be the compound vertices of G_1^c corresponding to the strongly connected components C_k and C_l, respectively, and let V_k and V_l be the vertex sets in G_1 of C_k and C_l, respectively. Then, (i_k, i_l) is an arc in G_1^c if every Markov chain in the MDP has a positive one-step transition from some state of V_k to some state of V_l, i.e. if $max_{r \in V_k} \{min_{a \in A(r)} \sum_{s \in V_l} p_{rs}(a)\} > 0$.

Since states in the same strongly connected component communicate under every policy, an arc (i_k, i_l) in G_1^c means that any $s \in V_l$ is accessible from any $r \in V_k$ under every policy. It is easy to verify that the construction of the condensed graph G_1^c has complexity $\mathcal{O}(M \cdot N)$. The operation 'condensation' can be repeated until there are no changes in the graph. Let $\{G_1^c\}^*$ be the finally, after repeated condensations, obtained graph.

Example 5.1

Let $S = \{1, 2, 3, 4\};\ A(1) = \{1, 2\},\ A(2) = \{1\},\ A(3) = \{1\},\ A(4) = \{1\}$.
$p_{12}(1) = 1;\ p_{13}(2) = 1;\ p_{23}(1) = p_{24}(1) = 0.5;\ p_{32}(1) = p_{34}(1) = 0.5;$
$p_{41}(1) = 0.5,\ p_{42}(1) = p_{43}(1) = 0.25$.
The graph at the right hand side presents the MDP model. Graph G_1 is the same, but without the arcs $(1, 2)$ and $(1, 3)$.
The strongly connected components of G_1 are: $C_1 = \{1\}$ and $C_2 = \{2, 3, 4\}$.
$G_1^c = (V_1^c, A_1^c)$ with $V_1^c = \{1^*, 2^*\}$, where 1^* corresponds to state 1 and 2^* to the states $2, 3$ and 4, and $A_1^c = \{(1^*, 2^*), (2^*, 1^*)\}$.
After condensing G_1^c, we obtain $\{G_1^c\}^*$, consisting of a single vertex.



The next lemma shows that irreducibility is equivalent to the property that $\{G_1^c\}^*$ consists of a single vertex.

Lemma 5.3
An MDP is irreducible if and only if the ultimate condensation $\{G_1^c\}^*$ consists of a single vertex.

Proof

$\Rightarrow$ Suppose that $\{G_1^c\}^*$ has at least two vertices. Then, there is a (compound) vertex, say i, without an incoming arc (if each vertex has an incoming arc, there is a circuit and the graph can be condensed). Therefore, in each state of the (compound) vertices $j \neq i$ an action can be chosen with transition probability 0 to the states of i. The Markov chain under such policy is not irreducible.

$\Leftarrow$ Let $\{G_1^c\}^*$ consists of a single vertex. From the definition of condensation it follows that each two states communicate under any policy, i.e. the Markov chain is irreducible. □

Algorithm 5.4 *Checking the irreducibility property of an MDP*
Input: A Markov decision problem.
Output: The property 'irreduclble' or 'not 'irreduclble'.

```
1. Construct the graph $G_1$; $G := G_1$.
2. Determine the strongly connected components of $G$, say $C_1, C_2, \dots, C_n$.
3. if all components consist of one vertex then go to step 4
   else begin construct the condensed graph of $G$, say $G^c$; $G := G^c$; return to step 2 end
4. if $n = 1$ then the MDP is irreducible (STOP)
   else the MPD is not irreducible (STOP)
```

The construction of G_1, the determination of the strongly connected components and the condensation operation have complexity of at most $\mathcal{O}(M \cdot N)$. In a new iteration the number of vertices of G decreases, so the number of iterations is at most N and the overall complexity of Algorithm 5.4 is $\mathcal{O}(M \cdot N^2)$.

The last classification question concerns the distinction between unichain and multichain. It turns out that this decision problem is $\mathcal{NP}$-complete, so there is no hope of a polynomial algorithm.

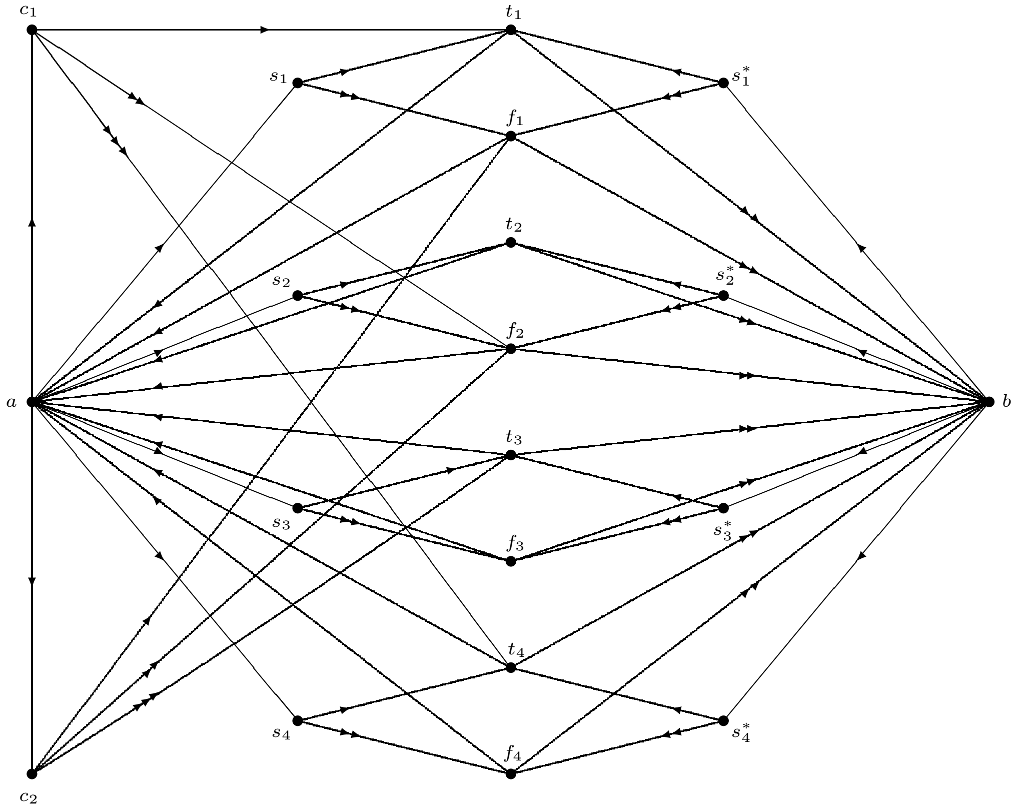
Suppose that there exists a policy that results in multiple ergodic classes. Such a policy serves as a *certificate* that the answer is "yes". Since the determination of the ergodic classes of a Markov chain is polynomially (see Algorithm 5.1), the problem is in $\mathcal{NP}$.

To prove that the problem is $\mathcal{NP}$-complete we use a reduction to the 3-satisfiability problem ($3SAT$). An instance of $3SAT$ consists of n Boolean variables $x_1, x_2, \dots, x_n$, and m clauses $C_1, C_2, \dots, C_m$, with three literals per clause. Each clause is the disjunction of three literals, where a literal is either a variable x_i or its negative $\overline{x}_i$, for example $C = x_2 \cup \overline{x}_4 \cup x_5$. The question is whether there is an assigment of values ("true" or "false") to the variables such that all clauses are satisfied.

Suppose that we are given an instance of $3SAT$, with n variables and m clauses.
We construct an MDP as follows:
(a) two special states a and b;
(b) $4n$ states $s_i, s_i^*, t_i, f_i,\ i = 1, 2, \dots, n$;
(c) m states $c_j,\ j = 1, 2, \dots, m$.

For the actions and the transition probabilities, we have:
$A(a) = \{1\}$ and $p_{as_i}(1) = \frac{1}{n+m},\ 1 \leq i \leq n;\ p_{ac_j}(1) = \frac{1}{n+m},\ 1 \leq j \leq m.$
$A(b) = \{1\}$ and $p_{bs_i^*}(1) = \frac{1}{n},\ 1 \leq i \leq n.$
$A(s_i) = \{1, 2\}$ and $p_{s_it_i}(1) = 1,\ 1 \leq i \leq n;\ p_{s_if_i}(2) = 1,\ 1 \leq i \leq n.$
$A(s_i^*) = \{1, 2\}$ and $p_{s_i^*t_i}(1) = 1,\ 1 \leq i \leq n;\ p_{s_i^*f_i}(2) = 1,\ 1 \leq i \leq n.$
$A(t_i) = \{1, 2\}$ and $p_{t_ia}(1) = 1,\ 1 \leq i \leq n;\ p_{t_ib}(2) = 1,\ 1 \leq i \leq n.$
$A(f_i) = \{1, 2\}$ and $p_{f_ia}(1) = 1,\ 1 \leq i \leq n;\ p_{f_ib}(2) = 1,\ 1 \leq i \leq n.$
$A(c_j) = \{1, 2, 3\}$ and action a corresponds to the a-th literal of clause C_j. In particilar, if the a-th literal in clause C_j is of the form x_i, then $p_{c_jt_i}(a) = 1$; if the a-th literal in clause C_j is of the form $\overline{x}_i$, then $p_{c_jf_i}(a) = 1$.

Example 5.2
Suppose that $n = 4,\ m = 2$ and $C_1 = x_1 \cup \overline{x}_2 \cup x_4,\ C_2 = \overline{x}_1 \cup \overline{x}_2 \cup x_3$. Below we draw the corresponding MDP. The transition probabilities are 1, except from a (the probabilities are $\frac{1}{6}$) and from b (the probabilities are $\frac{1}{4}$).



We claim that we have a "yes" instance of $3SAT$ if and only if the correponding MDP is multichain. Suppose that we have a "yes" instance of $3SAT$. Consider an assignment of the variables such that all clauses are satisfied. We define the following policy:

(1) At every state c_j, consider a literal in the clause which is "true". If that literal is unnegated, say x_k, pick in state c_j the action that moves to state t_k; if that literal is negated, say $\overline{x}_k$, pick in state c_j the action that moves to state f_k.

(2) At every state s_i, let the next state be t_i if x_i is "true", and f_i if x_i is "false".
(3) At every state s_i^*, let the next state be f_i if x_i is "true", and t_i if x_i is "false".
(4) At every state t_i, let the next state be a if x_i is "true", and b if x_i is "false".
(5) At every state f_i, let the next state be b if x_i is "true", and a if x_i is "false".
(6) In the states a and b is only one action.

First, look at state a as starting state of the Markov chain. At the next time point the Markov chain is in some state s_i or in some state c_j.
If the next state is s_i, then the following happens:
- if x_i is "true": the next state is t_i and return to state a;
- if x_i is "false": the next state is f_i and return to state a.
If the next state is c_j, then the following happens:
- if the chosen action corresponds to an unnegated variable x_k: the next state is t_k and return to state a;
- if the chosen action corresponds to a negated variable $\overline{x}_k$: the next state is f_k and return to state a.
We conclude that a is a recurrent state and, starting from a, state b is never visited.

Next, look at state b as starting state of the Markov chain. At the next time point the Markov chain is in some state s_i^* and the following happens:
- if x_i is "true": the next state is f_i and return to state b;
- if x_i is "false": the next state is t_i and return to state b.
We conclude that b is a recurrent state and, starting from b, state a is never visited. Therefore, the MDP is multichain.

For the converse, suppose that the MDP is multichain, and fix a policy that results in multiple ergodic classes. Given, the structure of the possible transitions, the state belongs to the set $\{a, b\}$ once every three transitions. Since we have multiple ergodic classes, it follows that a and b are both recurrent but do not belong to the same ergodic class. In particular, b is not accessible from a, and a is not accessible from b.

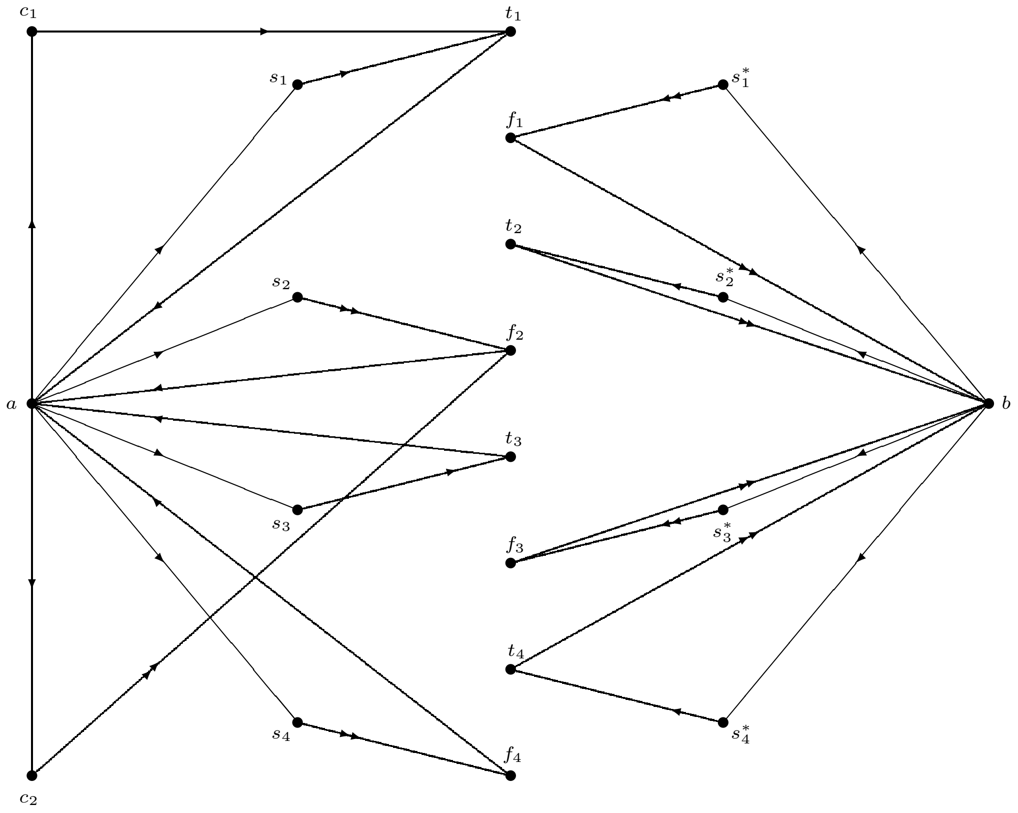
Consider the following assignment of the variables: if in state s_i action 1 is chosen, set x_i "true", and if in state s_i action 2 is chosen, set x_i "false". We need to show that with this assignment all clauses are satisfied.

Suppose that the transition out of s_i leads to t_i (i.e. $x_i = 1$). Since b is not accessible from a, it follows that b is not accessible from t_i, and therefore the action out of t_i leads back to a. Since a is not accessible from b, the transaction out of s_i^* leads to f_i and then back to b. Similarly, suppose that the transition out of s_i leads to f_i (i.e. $x_i = 0$). Since b is not accessible from a, it follows that b is not accessible from f_i, and therefore the action out of f_i leads back to a. Since a is not accessible from b, the transaction out of s_i^* leads to t_i and then back to b.
Consider now a clause C_j and suppose that the transition in state c_j leads to t_i, i.e. x_i is part of clause C_j. Since b is not accessible from a, it follows that t_i leads back to a. Using the remarks above, it follows that the transition out of s_i leads to t_i, and therefore x_i is set to "true", and the clause is satisfied.
Suppose that the transition in state c_j leads to f_i, i.e. $\overline{x}_i$ is part of clause C_j. Since b is not accessible from a, it follows that f_i leads back to a. Using the earlier remarks, it follows that the transition out of s_i leads to f_i, and therefore x_i is set to "false", and the clause is satisfied.

By the above arguments, we have shown the following theorem.

Theorem 5.1
The determination problem whether or not an MDP is unichain or multichain is $\mathcal{NP}$-complete.

Example 5.2 (continued)
Consider the assignment $x_1 = 1,\ x_2 = 0,\ x_3 = 1,\ x_4 = 0$. As corresponding policy we take action 1 in state c_1 and action 2 in state c_2. The Markov chain of this policy is presented in the figure below. It is easy to see that this chain is has two ergodic classes.



5.3 Stationary, fundamental and deviation matrix

5.3.1 The stationary matrix

Consider a policy $f^\infty \in C(D)$. In average reward MDPs the limiting behavior of $P^n(f)$ as n tends to infinity plays an important role. In general, $\lim_{n\to\infty} P^n(f)$ does not exist (a counterexample is left to the reader). Therefore, we consider other types of convergence.

Let $\{b_n\}_{n=0}^\infty$ be a sequence. This sequence is called *Cesaro convergent* with Cesaro limit b if

$$\lim_{n\to\infty} \frac{1}{n} \sum_{k=0}^{n-1} b_k \text{ exits and is equal to } b.$$

We denote this convergence by $lim_{n\to\infty}\, b_n =_c\ b$ or $b_n \to_c\ b$. The sequence is said to be *Abel convergent* with Abel limit b if

$$\lim_{\alpha\uparrow 1}(1-\alpha) \sum_{n=0}^{\infty} \alpha^n b_n \text{ exits and is equal to } b.$$

This convergence is denoted by $lim_{n\to\infty}\, b_n =_a\ b$ or $b_n \to_a\ b$. Ordinary convergence implies both Cesaro and Abel convergence, but the converse statements are not true in general (see Exercise 5.2). The next result is well known in the theory of the summability of series (e.g. Powell and Shah [225], p. 9).

Theorem 5.2

If the sequence $\{b_n\}_{n=0}^\infty$ is Cesaro convergent to b, then $\{b_n\}_{n=0}^\infty$ is also Abel convergent to b.

Remark

The converse statement of Theorem 5.2 is not true in general (see Exercise 5.3).

Theorem 5.3

Let P be any stochastic matrix, i.e. the matrix of a Markov chain. Then,
(1) $P^* := lim_{n\to\infty} \frac{1}{n}\sum_{k=0}^{n-1} P^k$ exits, i.e. $P^n \to_c P^*$.
(2) $P^*P = PP^* = P^*P^* = P^*$.

Proof

Let $B^{(n)} = \frac{1}{n}\sum_{k=0}^{n-1} P^k$. Since P^k is stochastic for every k, $B^{(n)}$ is also a stochastic matrix. Hence, the series $\left\{B^{(n)}\right\}_{n=1}^\infty$ is bounded. Therefore, each infinite subsequence of $\left\{B^{(n)}\right\}_{n=1}^\infty$ has a point of accumulation. Furthermore, we have

$$B^{(n)} + \frac{1}{n}\{P^n - I\} = B^{(n)}P = PB^{(n)},\ n \in \mathbb{N}. \tag{5.1}$$

Let $J = \lim_{k\to\infty} B^{(n_k)}$, where $\left\{B^{(n_k)}\right\}_{k=1}^\infty$ is a convergent subsequence of $\left\{B^{(n)}\right\}_{n=1}^\infty$. From (5.1) we obtain

$$J = JP = PJ. \tag{5.2}$$

Let $\left\{B^{(m_k)}\right\}_{k=1}^\infty$ also be a convergent subsequence of $\left\{B^{(n)}\right\}_{n=1}^\infty$ with limit matrix K. From (5.1) it also follows that

$$K = KP = PK. \tag{5.3}$$

Hence, $J = P^nJ = JP^n$ and $K = P^nK = KP^n$ for every n. Therefore, $J = B^{(n)}J = JB^{(n)}$ and $K = B^{(n)}K = KB^{(n)}$ for every n, implying that $J = KJ = JK$ and $K = JK = KJ$, i.e. $J = K$. The sequence $\left\{B^{(n)}\right\}_{n=1}^\infty$ has exactly one point of accumulation, i.e. $P^* := \lim_{n\to\infty} \frac{1}{n}\sum_{k=0}^{n-1} P^k$ exits and is the Cesaro limit of the sequence $\{P^n\}_{n=1}^\infty$. Hence, we have shown that $P^*P = PP^* = P^*P^* = P^*$. □

The matrix P^* is called the *stationary matrix* of the stochastic matrix P.

Corollary 5.1

$\lim_{\alpha\uparrow 1}\sum_{n=0}^\infty \alpha^n(P^n - P^*) = 0$.

Proof

Since P^n is Cesaro convergent to P^*, $P^n - P^*$ is Cesaro convergent to 0, and consequently Abel convergent to 0, i.e. $\lim_{\alpha\uparrow 1}\sum_{n=0}^\infty \alpha^n(P^n - P^*) = 0$. □

Let P be any stochastic matrix with ergodic classes $E_1, E_2, \dots, E_m$ and transient states T. By renumbering of the states the matrix can be written in the following so-called *standard form*:

$$P = \begin{pmatrix} P_1 & 0 & \cdot & \cdot & \cdot & \cdot & \cdot & 0 \\ 0 & P_2 & 0 & \cdot & \cdot & \cdot & \cdot & 0 \\ \cdot & \cdot & \cdot & \cdot & & & \cdot & 0 \\ \cdot & & \cdot & \cdot & \cdot & & \cdot & 0 \\ \cdot & & & \cdot & \cdot & \cdot & \cdot & 0 \\ 0 & \cdot & \cdot & \cdot & \cdot & 0 & P_m & 0 \\ A_1 & A_2 & \cdot & \cdot & \cdot & \cdot & A_m & Q \end{pmatrix}, \tag{5.4}$$

where the matrix P_k corresponds to the ergodic class $E_k,\ 1 \leq k \leq m$, and the matrix Q to the transient states. It is well known (e.g. Doob [74] p. 180) that $Q^n \to 0$ for $n \to \infty$. Since

$$(I - Q)(I + Q + \cdots + Q^{n-1}) = I - Q^n, \tag{5.5}$$

the right hand side of (5.5) tends to I, i.e. $I - Q$ is nonsingular and $(I - Q)^{-1} = \sum_{n=0}^{\infty} Q^n$.[1] From the theory of Markov chains it is also well known (see e.g. e.g. Chung [40] p. 33) that the stationary matrix of an ergodic class has strictly positive, identical rows, say π^k for P_k, and that π^k is the unique solution of the following system of linear equations

$$\left\{ \begin{array}{lcl} \sum_{i \in Ek} (\delta_{ij} - p_{ij})x_i & = & 0,\ j \in E_k \\ \sum_{i \in Ek} x_i & = & 1 \end{array} \right. \tag{5.6}$$

Since (5.6) is a system of $|E_k| + 1$ equations and $|E_k|$ variables, one of the equations, except the last normalization equation, can be deleted for the computation of π^k. The following results are also well known (see e.g. Feller [88]).

Lemma 5.4
Let a_i^k be the probability that, starting from state $i \in T$, the Markov chain will be absorbed in ergodic class $E_k,\ 1 \leq k \leq m$. Then, $a_i^k,\ i \in T$, is the unique solution of the linear system $(I - Q)x = b^k$, where $b^k = A_k e$.

Theorem 5.4
Let P be any stochastic matrix written in the standard form (5.4). Then,

$$P^* = \left(\begin{array}{cccccccc} P_1^* & 0 & \cdot & \cdot & \cdot & \cdot & \cdot & 0 \\ 0 & P_2^* & 0 & \cdot & \cdot & \cdot & \cdot & 0 \\ \cdot & \cdot & \cdot & \cdot & & & \cdot & \cdot \\ \cdot & & \cdot & \cdot & \cdot & & \cdot & \cdot \\ \cdot & & & \cdot & \cdot & \cdot & \cdot & \cdot \\ 0 & \cdot & \cdot & \cdot & \cdot & 0 & P_m^* & 0 \\ A_1^* & A_2^* & \cdot & \cdot & \cdot & \cdot & A_m^* & 0 \end{array} \right), \tag{5.7}$$

where P_k^* has identical rows π^k, which are the unique solution of (5.6) and $A_k^* = \{I - Q\}^{-1}\{A_k e\}\{\pi^k\}^T$, $1 \leq k \leq m$.

Algorithm 5.5 *Determination of the stationary matrix P^**
Input: A stochastic matrix P.
Output: The stationary matrix P^*.

```
1. Determine with Algorithm 5.1 the ergodic classes $E_1, E_2, \ldots, E_m$ and the transient states $T$.
2. Write $P$ in standard form (5.4).
3. for $k = 1, 2, \ldots, m$ do
     begin
         determine the unique solution $\pi_j^k,\ j \in E_k$, of the linear system
```

[1] A series $\sum_{n=0}^{\infty} A^n$ is a generalization of the geometric series and is often referred to as the *Neumann series.*

$$\begin{cases} \sum_{i \in E_k} (\delta_{ij} - p_{ij})x_i \quad = 0, \; j = 2, 3, \dots, |E_k| \\ \sum_{i \in E_k} x_i = 1; \end{cases}$$

determine the unique solution $a_i^k, \; i \in T$, of the linear sytem

$$\sum_{j \in T} (\delta_{ij} - p_{ij})x_j = \sum_{l \in E_k} p_{il}, \; i \in T.$$

end

4. $p^*_{ij} := \begin{cases} x_j^k & i \in E_k, \; j \in E_k, \; k = 1, 2, \dots, m \\ a_i^k x_j^k & i \in T, \; j \in E_k, \; k = 1, 2, \dots, m \\ 0 & \text{else} \end{cases}$

Example 5.3

Consider the Markov chain with transition matrix $P = \begin{pmatrix} 0.5 & 0 & 0 & 0.5 & 0 \\ 0 & 0 & 0 & 0.4 & 0.6 \\ 0 & 0.2 & 0.4 & 0 & 0.4 \\ 0.7 & 0 & 0 & 0.3 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix}$.

Using Algorithm 5.1 we obtain the ergodic classes $E_1 = \{1, 4\}, \; E_2 = \{5\}, \; T = \{2, 3\}$.

The standard form of the matrix is: $P = \begin{pmatrix} 0.5 & 0.5 & 0 & 0 & 0 \\ 0.7 & 0.3 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0.4 & 0.6 & 0 & 0 \\ 0 & 0 & 0.4 & 0.2 & 0.4 \end{pmatrix}$.

$k = 1$: π^1 is the unique solution of $\begin{cases} 0.5x_1 & - & 0.7x_2 & = & 0 \\ x_1 & + & x_2 & = & 1 \end{cases} \rightarrow \pi_1^1 = \frac{7}{12}, \; \pi_2^1 = \frac{5}{12}$.

a^1 is the unique solution of $\begin{cases} x_1 & & & = & 0.4 \\ -0.2x_1 & + & 0.6x_2 & = & 0 \end{cases} \rightarrow a_4^1 = \frac{2}{5}, \; a_5^1 = \frac{2}{15}$.

$k = 2$: $\pi^2 = 1$ (state 3 is an absorbing state) and a^2 is the unique solution of

$$\begin{cases} x_1 & & & = & 0.6 \\ -0.2x_1 & + & 0.6x_2 & = & 0.4 \end{cases} \rightarrow a_4^2 = \frac{3}{5}, \; a_5^2 = \frac{13}{15}.$$

The stationary matrix $P^* = \begin{pmatrix} \frac{7}{12} & \frac{5}{12} & 0 & 0 & 0 \\ \frac{7}{12} & \frac{5}{12} & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ \frac{7}{30} & \frac{5}{30} & \frac{9}{15} & 0 & 0 \\ \frac{7}{90} & \frac{5}{90} & \frac{13}{15} & 0 & 0 \end{pmatrix}$.

5.3.2 The fundamental matrix and the deviation matrix

Theorem 5.5

Let P be an arbitrary stochastic matrix. Then, $I - P + P^*$ is nonsingular and $Z := (I - P + P^*)^{-1}$ satisfies $Z = lim_{n \to \infty} \frac{1}{n} \sum_{i=1}^{n} \sum_{k=0}^{i-1} (P - P^*)^k$.

Proof
Since $P^*P = PP^* = P^*P^* = P^*$ (see Theorem 5.3) it follows, by induction on n, that $(P - P^*)^n = P^n - P^*,\ n \in \mathbb{N}$. Let $B := P - P^*$. Since

$$I - B^i = (I - B)(I + B + \cdots + B^{i-1}), \tag{5.8}$$

we have, by averaging (5.8),

$$I - \frac{1}{n}\sum_{i=1}^{n} B^i = (I - B) \cdot \frac{1}{n}\sum_{i=1}^{n}\sum_{k=0}^{i-1} B^k. \tag{5.9}$$

Since $\frac{1}{n}\sum_{i=1}^n B^i = \frac{1}{n}\sum_{i=1}^n (P^i - P^*) = \frac{1}{n}\sum_{i=1}^n P^i - P^*$, we obtain

$$\lim_{n\to\infty} \frac{1}{n}\sum_{i=1}^{n} B^i = \lim_{n\to\infty} \frac{1}{n}\sum_{i=1}^{n} P^i - P^* = P^* - P^* = 0,$$

i.e. $I - B = I - P + P^*$ is nonsingular and $Z = \lim_{n\to\infty} \frac{1}{n}\sum_{i=1}^n \sum_{k=0}^{i-1} (P - P^*)^k$. □

The matrix $Z = (I - P + P^*)^{-1}$ is called the *fundamental matrix* of P. The *deviation matrix* D is defined by $D := Z - P^* = \lim_{n\to\infty} \frac{1}{n}\sum_{i=1}^n \sum_{k=0}^{i-1} (P - P^*)^k - P^*$.

Theorem 5.6
The deviation matrix D satisfies
(1) $D = lim_{n\to\infty} \frac{1}{n}\sum_{i=1}^n \sum_{k=0}^{i-1} (P^k - P^*)$.
(2) $P^*D = DP^* = (I - P)D + P^* - I = D(I - P) + P^* - I = 0$.

Proof
(1) Since $(P - P^*)^k = (P^k - P^*)$ for $k = 1, 2, \ldots$, we obtain
$\sum_{i=1}^n \sum_{k=0}^{i-1} (P - P^*)^k = n \cdot I + \sum_{i=2}^n \sum_{k=1}^{i-1} (P - P^*)^k = n \cdot I + \sum_{i=2}^n \sum_{k=1}^{i-1} (P^k - P^*)$ and
$\sum_{i=1}^n \sum_{k=0}^{i-1} (P^k - P^*) = n \cdot (I - P^*) + \sum_{i=2}^n \sum_{k=1}^{i-1} (P^k - P^*)$.
Therefore,

$$\begin{aligned}\lim\nolimits_{n\to\infty} \tfrac{1}{n}\textstyle\sum_{i=1}^n \sum_{k=0}^{i-1} (P^k - P^*) &= \lim\nolimits_{n\to\infty} \tfrac{1}{n}\{n \cdot (I - P^*) + \textstyle\sum_{i=2}^n \sum_{k=1}^{i-1} (P - P^*)^k \\ &= Z - P^*.\end{aligned}$$

(2)

$$\begin{aligned}P^*D &= \lim\nolimits_{n\to\infty} \tfrac{1}{n}\textstyle\sum_{i=1}^n \sum_{k=0}^{i-1} P^*(P^k - P^*) = \lim_{n\to\infty} \tfrac{1}{n}\sum_{i=1}^n \sum_{k=0}^{i-1} (P^* - P^*) = 0. \\ (I - P)D &= \lim\nolimits_{n\to\infty} \tfrac{1}{n}\textstyle\sum_{i=1}^n \sum_{k=0}^{i-1} (I - P)(P^k - P^*) \\ &= \lim\nolimits_{n\to\infty} \tfrac{1}{n}\textstyle\sum_{i=1}^n \sum_{k=0}^{i-1} (P^k - P^{k+1}) \\ &= \lim\nolimits_{n\to\infty} \tfrac{1}{n}\textstyle\sum_{i=1}^n (I - P^k) = I - P^*.\end{aligned}$$

Similarly, it can be shown that $DP^* = 0$ and $D(I - P) = I - P^*$. □

The fundamental and the deviation matrix can be computed as follows. From (5.4) and (5.7) it follows that

$$I - P + P^* = \begin{pmatrix} C_1 & 0 & \cdot & \cdot & \cdot & \cdot & \cdot & 0 \\ 0 & C_2 & 0 & \cdot & \cdot & \cdot & \cdot & 0 \\ \cdot & \cdot & \cdot & \cdot & & & \cdot & \cdot \\ \cdot & & \cdot & \cdot & \cdot & & \cdot & \cdot \\ \cdot & & & \cdot & \cdot & \cdot & \cdot & \cdot \\ 0 & \cdot & \cdot & \cdot & \cdot & 0 & C_m & 0 \\ D_1 & D_2 & \cdot & \cdot & \cdot & \cdot & D_m & I - Q \end{pmatrix},$$

where $C_k := I - P_k + P_k^*$ and $D_k := -A_k + A_k^*,\ 1 \le k \le m$. Hence,

$$Z = (I - P + P^*)^{-1} = \begin{pmatrix} C_1^{-1} & 0 & \cdot & \cdot & \cdot & \cdot & \cdot & 0 \\ 0 & C_2^{-1} & 0 & \cdot & \cdot & \cdot & \cdot & 0 \\ \cdot & \cdot & \cdot & \cdot & & & \cdot & \cdot \\ \cdot & & \cdot & \cdot & \cdot & & \cdot & \cdot \\ \cdot & & & \cdot & \cdot & \cdot & \cdot & \cdot \\ 0 & \cdot & \cdot & \cdot & \cdot & 0 & C_m^{-1} & 0 \\ S_1 & S_2 & \cdot & \cdot & \cdot & \cdot & S_m & (I-Q)^{-1} \end{pmatrix},$$

where $S_k := -(I-Q)^{-1} D_k C_k^{-1},\ 1 \le k \le m$. The deviation matrix is $Z - P^*$.

Example 5.3 (continued)

$$I - P + P^* = \begin{pmatrix} \frac{13}{12} & -\frac{1}{12} & 0 & 0 & 0 \\ -\frac{7}{60} & \frac{67}{60} & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ \frac{7}{30} & -\frac{7}{30} & 0 & 1 & 0 \\ \frac{7}{90} & \frac{5}{90} & \frac{7}{15} & -\frac{1}{5} & \frac{3}{15} \end{pmatrix} \rightarrow C_1 = \begin{pmatrix} \frac{13}{12} & -\frac{1}{12} \\ -\frac{7}{60} & \frac{67}{60} \end{pmatrix} \text{ and } C_2 = (1).$$

By inverting, $C_1^{-1} = \begin{pmatrix} \frac{67}{72} & \frac{5}{72} \\ \frac{7}{72} & \frac{65}{72} \end{pmatrix}$ and $C_2^{-1} = (1)$. Since $I - Q = \begin{pmatrix} 1 & 0 \\ -\frac{1}{5} & \frac{3}{5} \end{pmatrix}$, by inverting, we obtain $(I-Q)^{-1} = \begin{pmatrix} 1 & 0 \\ \frac{1}{3} & \frac{5}{3} \end{pmatrix}$. Hence, $S_1 = -(I-Q)^{-1} D_1 C_1^{-1} = -\begin{pmatrix} 1 & 0 \\ \frac{1}{3} & \frac{5}{3} \end{pmatrix} \begin{pmatrix} \frac{7}{30} & -\frac{7}{30} \\ \frac{7}{90} & \frac{5}{90} \end{pmatrix} \begin{pmatrix} \frac{67}{72} & \frac{5}{72} \\ \frac{7}{72} & \frac{65}{72} \end{pmatrix} = \begin{pmatrix} -\frac{7}{36} & \frac{7}{36} \\ -\frac{7}{36} & -\frac{1}{36} \end{pmatrix}$ and $S_2 = -(I-Q)^{-1} D_2 C_2^{-1} = -\begin{pmatrix} 1 & 0 \\ \frac{1}{3} & \frac{5}{3} \end{pmatrix} \begin{pmatrix} 0 \\ \frac{7}{15} \end{pmatrix} (1) = \begin{pmatrix} 0 \\ -\frac{7}{9} \end{pmatrix}$. Therefore, we have computed

$$Z = \begin{pmatrix} \frac{67}{72} & \frac{5}{72} & 0 & 0 & 0 \\ \frac{7}{72} & \frac{65}{72} & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ -\frac{7}{36} & \frac{7}{36} & 0 & 1 & 0 \\ -\frac{7}{36} & -\frac{1}{36} & -\frac{7}{9} & \frac{1}{3} & \frac{5}{3} \end{pmatrix} \text{ and } D = \begin{pmatrix} \frac{25}{72} & -\frac{25}{72} & 0 & 0 & 0 \\ -\frac{35}{72} & \frac{35}{72} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ -\frac{77}{180} & \frac{1}{36} & -\frac{3}{5} & 1 & 0 \\ -\frac{49}{180} & -\frac{1}{12} & -\frac{74}{45} & \frac{1}{3} & \frac{5}{3} \end{pmatrix}.$$

In the theorems 5.5 and 5.6 the fundamental matrix Z and the deviation matrix D are expressed as Cesaro limits. These matrices can also be expressed in Abelian form as the next theorem shows.

Theorem 5.7
(1) $Z = \lim_{\alpha \uparrow 1} \sum_{n=0}^\infty \alpha^n (P - P^*)^n$.
(2) $D = \lim_{\alpha \uparrow 1} \sum_{n=0}^\infty \alpha^n (P^n - P^*)$.

Proof
(1) Similar as the proof that $(I-Q)^{-1} = \sum_{n=0}^\infty Q^n$, it can be shown that

$$H(\alpha) := \textstyle\sum_{n=0}^\infty \big\{\alpha(P - P^*)\big\}^n = \big\{I - \alpha(P - P^*)\big\}^{-1}.$$

Hence, $I = H(\alpha)\big\{I - \alpha(P - P^*)\big\} = H(\alpha)(I - P + P^*) + (1-\alpha)H(\alpha)(P - P^*)$. Since $P^n - P^*$ is Cesaro convergent to 0, $P^n - P^*$ is also Abel convergent to 0, i.e. $\lim_{\alpha \uparrow 1} (1-\alpha)H(\alpha) = 0$. Therefore, $Z = (I - P + P^*)^{-1} = \lim_{\alpha \uparrow 1} \sum_{n=0}^\infty \alpha^n (P - P^*)^n$.

(2) Because

$$\begin{aligned}\textstyle\sum_{n=0}^\infty \alpha^n(P^n - P^*) &= I - P^* + \textstyle\sum_{n=1}^\infty \alpha^n(P^n - P^*) = I - P^* + \textstyle\sum_{n=1}^\infty \alpha^n(P - P^*)^n \\ &= \textstyle\sum_{n=0}^\infty \alpha^n(P - P^*)^n - P^*,\end{aligned}$$

we obtain

$$\lim_{\alpha\uparrow 1} \textstyle\sum_{n=0}^\infty \alpha^n(P^n - P^*) = \lim_{\alpha\uparrow 1} \textstyle\sum_{n=0}^\infty \alpha^n(P - P^*)^n - P^* = Z - P^* = D.$$ □

The following theorem gives the relation between average rewards, discounted rewards (over an infinite horizon) and total rewards over a finite horizon.

Theorem 5.8

Let f^∞ be a deterministic policy. Then,

(1) $\phi(f^\infty) = P^*(f)r(f)$.

(2) $\phi(f^\infty) = \lim_{\alpha\uparrow 1} (1-\alpha)v^\alpha(f^\infty)$.

(3) $v^T(f^\infty) = T\phi(f^\infty) + D(f)r(f) - P^T(f)D(f)r(f)$ for any $T \in \mathbb{N}$.

Proof

(1) $\phi(f^\infty) = \liminf_{T\to\infty} \frac{1}{T}\sum_{t=1}^T P^t(f)r(f) = P^*(f)r(f)$.

(2) Since P^* is the Cesaro limit of P^t, it is also the Abel limit, i.e.

$\phi(f^\infty) = P^*(f)r(f) = \lim_{\alpha\uparrow 1} (1-\alpha)\sum_{t=0}^\infty \{\alpha P(f)\}^t r(f) = v^\alpha(f^\infty)$.

(3) We apply induction on T.

$T = 1$: $\phi(f^\infty) + D(f)r(f) - P(f)D(f)r(f) = \big\{P^*(f) + \{I - P(f)\}D(f)\big\}r(f) = r(f) = v^1(f)$,

using that $P^*(f) + \{I - P(f)\}D(f) = I$ (see Theorem 5.6, part (2)).

Suppose that the statement is true for T periods. Then, we can write

$(T+1)\phi(f^\infty) + D(f)r(f) - P^{T+1}(f)D(f)r(f) =$

$T\phi(f^\infty) + P^*(f)r(f) + D(f)r(r) - P^{T+1}(f)D(f)r(f) =$ (by the induction hypothesis)

$v^T(f^\infty) + P^T(f)D(f)r(f) + P^*(f)r(f) - P^{T+1}(f)D(f)r(f) =$

$v^T(f^\infty) + P^T(f)\big\{D(f) + P^*(f) - P(f)D(f)\big\}r(f) =$ (using Theorem 5.6, part (2))

$v^T(f^\infty) + P^T(f)r(f) = v^{T+1}(f^\infty)$. □

The regular case

A Markov chain P is called a *regular* Markov chain if the chain is irreducible and aperiodic. In that case it can be shown[2] that $P^* = \lim_{n\to\infty} P^n$. Since $(P - P^*)^n = P^n - P^*$ for $n = 1, 2, \dots$, we have $(P - P^*)^n \to 0$ if $n \to \infty$. Therefore,

$$Z = (I - P + P^*)^{-1} = \sum_{n=0}^\infty (P - P^*)^n.$$

Because $D = Z - P^*$ and $Z = I + \sum_{n=1}^\infty (P - P^*)^n = I + \sum_{n=1}^\infty (P^n - P^*)$, we obtain

$$D = \sum_{n=0}^\infty (P^n - P^*),$$

i.e. D represents the total deviation with respect to the stationary matrix. This explains the name *deviation matrix*.

[2] For the proof see e.g. J. Kemeny and L. Snell: *Finite Markov chains*, Van Nostrand, 1960. p. 70.

5.4 Extension of Blackwell's theorem

The next theorem shows that the interval $[0,1)$ can be partitioned in a finite number of subintervals such that in each subinterval there exists a deterministic policy which is optimal over the whole subinterval.

Theorem 5.9

There are numbers $\alpha_m, \alpha_{m-1}, \dots, \alpha_0, \alpha_{-1}$ and deterministic policies $f_m^\infty, f_{m-1}^\infty, \dots, f_0^\infty$ such that
(1) $0 = \alpha_m < \alpha_{m-1} < \cdots < \alpha_0 < \alpha_{-1} = 1$;
(2) $v^\alpha(f_j^\infty) = v^\alpha$ for all $\alpha \in [\alpha_j, \alpha_{j-1}),\ j = m, m-1, \dots, 0$.

Proof

For any deterministic policy f^∞, $v^\alpha(f^\infty)$ is the unique solution of the linear system $\{I - \alpha P(f)\}x = r(f)$. By Cramer's rule[3] $v_i^\alpha(f^\infty)$ is a rational function in α for each component i. Suppose that a deterministic Blackwell optimal policy does not exist. For any fixed α a deterministic α-discounted optimal policy exists. This implies a series $\{\alpha_k,\ k = 1, 2, \dots\}$ and a series $\{f_k,\ k = 1, 2, \dots\}$ such that

$$\alpha_1 \leq \alpha_2 \leq \cdots \text{ with } lim_{k\to\infty}\, \alpha_k = 1 \text{ and } v^\alpha = v^\alpha(f_k^\infty) > v^\alpha(f_{k-1}^\infty) \text{ for } \alpha = \alpha_k,\ k = 2, 3, \dots.$$

Since there are only a finite number of deterministic policies, there must be a couple of policies, say f^∞ and g^∞, such that for some nondecreasing subsequence $\alpha_{k_n}, n = 1, 2, \dots$ with $lim_{n\to\infty}\alpha_{k_n} = 1$,

$$\begin{cases} v^\alpha(f^\infty) > v^\alpha(g^\infty) & \text{for } \alpha = \alpha_{k_1}, \alpha_{k_3}, \dots \\ v^\alpha(f^\infty) < v^\alpha(g^\infty) & \text{for } \alpha = \alpha_{k_2}, \alpha_{k_4}, \dots \end{cases} \tag{5.10}$$

Let $h(\alpha) = v^\alpha(f^\infty) - v^\alpha(g^\infty)$, then $h_i(\alpha)$ is a continuous rational function in α on $[0,1)$ for each $i \in S$. From (5.10) it follows that $h_i(\alpha)$ has an infinite number of zeros, which is in contradiction with the rationality of $h_i(\alpha)$. Hence, there exists a deterministic Blackwell optimal policy, i.e. a policy f_0^∞ such that $v^\alpha(f_0^\infty) = v^\alpha$ for all $\alpha \in [\alpha_0, 1)$ for some $0 \leq \alpha_0 < 1$.

With similar arguments it can be shown that for each fixed $\alpha \in [0,1)$ there is a lower bound $L(\alpha) < \alpha$ and a deterministic policy $f_{L(\alpha)}^\infty$ such that $v^\alpha(f_{L(\alpha)}^\infty) = v^\alpha$ for all $\alpha \in \big(L(\alpha), \alpha\big)$. Similarly, for each fixed $\alpha \in (0,1]$ there is an upper bound $U(\alpha) > \alpha$ and a deterministic policy $f_{U(\alpha)}^\infty$ such that $v^\alpha(f_{U(\alpha)}^\infty) = v^\alpha$ for all $\alpha \in \big(\alpha, U(\alpha)\big)$.

The open intervals $\big(-1, U(0)\big)$, $\big\{\big(L(\alpha), U(\alpha)\big) \mid \alpha \in (0,1)\big\}$ and $\big(L(1), 2\big)$ are a covering of the compact set $[0,1]$. By the Heine-Borel-Lebesque covering theorem[4] the interval $[0,1]$ is covered by a finite number of intervals, say $\big(-1, U(0)\big)$, $\big\{\big(L(\alpha_j), U(\alpha_j)\big),\ j = m-1, m-2, \dots, 1\big\}$ and $\big(L(1), 2\big)$. We may assume that

$$\alpha_m := 0 < \alpha_{m-1} < \cdots < \alpha_0 < \alpha_{-1} := 1,\ L(\alpha_{m-1}) < U(0),\ L(1) < U(\alpha_1)$$

and

$$L(\alpha_j) < L(\alpha_{j-1}) < U(\alpha_j) < U(\alpha_{j-1}),\ j = m-1, m-2, \dots, 2.$$

Since the rational functions $v^\alpha(f_{L(\alpha_{j-1})}^\infty) = v^\alpha(f_{U(\alpha_j)}^\infty) = v^\alpha$ for all $\alpha \in \big(L(\alpha_{j-1}), U(\alpha_j)\big)$ we have

$$v^\alpha(f_{L(\alpha_{j-1})}^\infty) = v^\alpha(f_{U(\alpha_j)}^\infty),\ j = 0, 1, \dots, m.$$

Let $f_j = f_{U(\alpha_j)},\ j = 0, 1, \dots, m$. Then, $v^\alpha(f_j^\infty) = v^\alpha$ for all $\alpha \in (\alpha_j, \alpha_{j-1}),\ j = 0, 1, \dots, m$. Since $v^\alpha(f^\infty)$ is continuous in α, also $v^\alpha(f_j^\infty) = v^\alpha$ for $\alpha = \alpha_j,\ j = 0, 1, \dots, m$. □

[3] see e.g. J.B. Fraleigh and R.A. Beauregard: *Linear Algebra*, Addison Wesley, 1987, p. 214.

[4] See e.g. A.C. Zaanen: *Integration*, North Holland, 1967.

Corollary 5.2

For $\alpha \in [0,1)$, the value vector v^α is a continuous, piecewise rational function in α with no singular points.

Example 5.4

Let $S = \{1,2\};\ A(1) = \{1,2,3\},\ A(2) = \{1\},\ A(3) = \{1\};\ p_{11}(1) = 0,\ p_{12}(1) = 1;\ p_{11}(2) = 1,\ p_{12}(2) = 0;$ $p_{11}(3) = 0.5,\ p_{12}(3) = 0.5;\ r_1(1) = 1,\ r_1(2) = 0.5,\ r_1(3) = 0.75,\ r_2(1) = 0.$

There are three deterministic policies: $f_1^\infty, f_2^\infty, f_3^\infty$ with $f_1(1) = 1,\ f_2(1) = 2$ and $f_3(1) = 3$.

$v_1^\alpha(f_1^\infty) = 1,\ v_2^\alpha(f_1^\infty) = 0;\ v_1^\alpha(f_2^\infty) = \frac{0.5}{1-\alpha},\ v_2^\alpha(f_2^\infty) = 0;\ v_1^\alpha(f_3^\infty) = \frac{0.75}{1-0.5\alpha},\ v_2^\alpha(f_1^\infty) = 0.$

Hence, $v_1^\alpha = \begin{cases} 1 & \text{if } 0 \le \alpha \le 0.5; \\ \frac{0.5}{1-\alpha} & \text{if } 0.5 \le \alpha < 1. \end{cases}$

The value vector v^α is in the interval $[0,1)$ indeed a continuous, piecewise rational function in α with no singular points.

5.5 The Laurent series expansion

Theorem 5.8 part (2) shows a relation between discounted and average reward when the discount factor tends to 1. This relation is based on the Laurent expansion of $v^\alpha(f^\infty)$ close to $\alpha = 1$ as expressed in the next theorem.

Theorem 5.10

Let $u^k(f),\ k = -1, 0, \dots$ be defined by: $u^{-1}(f) := P^*(f)r(f),\ u^0(f) := D(f)r(f)$ and for $k \ge 1$ $u^k(f) := -D(f)u^{k-1}(f)$. Then, $\alpha v^\alpha(f^\infty) = \sum_{k=-1}^\infty \rho^k u^k(f)$ for $\alpha_0(f) < \alpha < 1$, where $\rho = \frac{1-\alpha}{\alpha}$ and $\alpha_0(f) = \frac{\|D(f)\|}{1+\|D(f)\|}$.

Proof

Let $x(f) := \frac{1}{\alpha} \cdot \sum_{k=-1}^\infty \rho^k u^k(f) = \frac{\phi(f^\infty)}{1-\alpha} + \frac{1}{\alpha} \cdot \sum_{k=0}^\infty \rho^k u^k(f)$. Since $u^k(f) = D(f)\{-D(f)\}^k r(f)$ for $k \ge 0$, the series $\sum_{k=0}^\infty \rho^k u^k(f)$ is well defined if $\|\rho D(f)\| < 1$, i.e. $\alpha \ge \frac{\|D(f)}{1+\|D(f)\|}$. Since $v^\alpha(f^\infty)$ is the unique solution of the linear system $\{I - \alpha P(f)\}x = r(f)$, it is sufficient to show that $\{I - \alpha P(f)\}x(f) = r(f)$, i.e. $y(f) := r(f) - \{I - \alpha P(f)\}x(f) = 0$. We can write,

$$\begin{aligned}
y(f) &= r(f) - \{I - \alpha P(f)\}\tfrac{P^*(f)r(f)}{1-\alpha} - \{I - \alpha P(f)\}\tfrac{D(f)}{\alpha}\textstyle\sum_{k=0}^\infty \{-\rho D(f)\}^k r(f) \\
&= r(f) - P^*(f)r(f) - \big\{\alpha\big(I - P(f)\big) + (1-\alpha)I\big\}\tfrac{D(f)}{\alpha}\textstyle\sum_{k=0}^\infty \{-\rho D(f)\}^k r(f) \\
&= \{I - P^*(f)\}r(f) - \{I - P(f)\}D(f)\textstyle\sum_{k=0}^\infty \{-\rho D(f)\}^k r(f) - \tfrac{1-\alpha}{\alpha}D(f)\textstyle\sum_{k=0}^\infty \{-\rho D(f)\}^k r(f) \\
&= \{I - P^*(f)\}r(f) - \{I - P^*(f)\}\textstyle\sum_{k=0}^\infty \{-\rho D(f)\}^k r(f) + \textstyle\sum_{k=0}^\infty \{-\rho D(f)\}^{k+1} r(f) \\
&= \{I - P^*(f)\}r(f) - \textstyle\sum_{k=0}^\infty \{-\rho D(f)\}^k r(f) + P^*(f)r(f) + \textstyle\sum_{k=1}^\infty \{-\rho D(f)\}^k r(f) \\
&= \{I - P^*(f)\}r(f) - r(f) - \textstyle\sum_{k=1}^\infty \{-\rho D(f)\}^k r(f) + P^*(f)r(f) + \textstyle\sum_{k=1}^\infty \{-\rho D(f)\}^k r(f) \\
&= 0.
\end{aligned}$$

□

Corollary 5.3

$v^\alpha(f^\infty) = \frac{\phi(f^\infty)}{1-\alpha} + u^0(f) + \varepsilon(\alpha)$, where $\varepsilon(\alpha)$ satisfies $lim_{\alpha\uparrow 1}\ \varepsilon(\alpha) = 0$.

Proof

From Theorem 5.10, we obtain $v^\alpha(f^\infty) = \frac{\phi(f)}{1-\alpha} + \frac{u^0(f)}{\alpha} + \sum_{k=1}^\infty \frac{(1-\alpha)^k}{\alpha^{k+1}} u^k(f)$. By the series expansion $\frac{1}{\alpha} = \frac{1}{1-(1-\alpha)} = 1 + (1-\alpha) + (1-\alpha)^2 + \cdots$, we may write $v^\alpha(f^\infty) = \frac{\phi(f)}{1-\alpha} + u^0(f) + \varepsilon(\alpha)$, where $\lim_{\alpha\uparrow 1}\ \varepsilon(\alpha) = 0$.

□

5.6 The optimality equation

In the discounted case, the value vector is the unique solution of an optimality equation. For the average reward criterion a similar result holds, but the equation is more complicated.

Theorem 5.11
Consider the system

$$\begin{cases} x_i = max_{a\in A(i)} \sum_j p_{ij}(a)x_j & , \ i \in S \\ x_i + y_i = max_{a\in A(i,x)}\{r_i(a) + \sum_j p_{ij}(a)y_j\} & , \ i \in S \end{cases} \tag{5.11}$$

where $A(i,x) := \{a \in A(i) \mid x_i = \sum_j p_{ij}(a)x_j\}, \ i \in S.$

This system has the following properties:

(1) $x = u^{-1}(f_0), \ y = u^0(f_0)$, *where* f_0^∞ *is a Blackwell optimal policy, satisfies (5.11).*

(2) If (x,y) *is a solution of (5.11), then* $x = \phi$, *the value vector.*

Proof
Since f_0^∞ is a Blackwell optimal policy, for α sufficiently close to 1, say $\alpha \in [\alpha_0, 1)$, one can write

$$v_i^\alpha(f_0^\infty) = v_i^\alpha = max_{a\in A(i)}\{r_i(a) + \alpha\sum_j p_{ij}(a)v_j^\alpha\} \geq r_i(a) + \alpha\sum_j p_{ij}(a)v_j^\alpha, \ (i,a) \in S \times A.$$

Combining this result with Corollary 5.3 gives for all $\alpha \in [\alpha_0, 1)$:

$$\begin{aligned} \tfrac{\phi_i(f_0^\infty)}{1-\alpha} + u_i^0(f_0) + \varepsilon_i(\alpha) &\geq r_i(a) + \{1-(1-\alpha)\}\textstyle\sum_j p_{ij}(a)\Big\{\tfrac{\phi_j(f_0^\infty)}{1-\alpha} + u_j^0(f_0) + \varepsilon_j(\alpha)\Big\} \\ &= r_i(a) + \textstyle\sum_j p_{ij}(a)\Big\{\tfrac{\phi_j(f_0^\infty)}{1-\alpha} + u_j^0(f_0) + \varepsilon_j(\alpha)\Big\} - \\ &\qquad (1-\alpha)\textstyle\sum_j p_{ij}(a)\Big\{\tfrac{\phi_j(f_0^\infty)}{1-\alpha} + u_j^0(f_0) + \varepsilon_j(\alpha)\Big\}, \ (i,a) \in S \times A, \end{aligned}$$

i.e.

$$\tfrac{1}{1-\alpha}\Big\{\phi_i(f_0^\infty) - \textstyle\sum_j p_{ij}(a)\phi_j(f_0^\infty)\Big\} + \Big\{u_i^0(f_0) - r_i(a) - \textstyle\sum_j p_{ij}(a)u_j^0(f_0) + \textstyle\sum_j p_{ij}(a)\phi_j(f_0^\infty)\Big\} + \varepsilon(\alpha) \geq 0.$$

Since this result holds for all $\alpha \in [\alpha_0, 1)$, the first term multiplied by $\frac{1}{1-\alpha}$ has to be nonnegative, i.e.

$$\phi_i(f_0^\infty) \geq \sum_j p_{ij}(a)\phi_j(f_0^\infty) \text{ for all } i \in S \text{ and } a \in A(i). \tag{5.12}$$

Furthermore, when $\phi_i(f_0^\infty) = \sum_j p_{ij}(a)\phi_j(f_0^\infty)$, the second term has to be nonnegative, i.e.

$$u_i^0(f_0) \geq r_i(a) + \sum_j p_{ij}(a)u_j^0(f_0) - \sum_j p_{ij}(a)\phi_j(f_0^\infty) = r_i(a) + \sum_j p_{ij}(a)u_j^0(f_0) - \phi_i(f_0^\infty). \tag{5.13}$$

For $a = f_0(i), \ i \in S$, the inequalities in (5.12) and (5.13) are equalities, because:

$$\phi(f_0^\infty) = P^*(f_0)r(f_0) = P(f_0)P^*(f_0)r(f_0) = P(f_0)\phi(f_0^\infty)$$

and

$$u^0(f_0) = D(f_0)r(f_0) = \{I - P^*(f_0) + P(f_0)D(f_0)\}r(f_0) = r(f_0) - \phi(f_0^\infty) + P(f_0)u^0(f_0).$$

By these results, part (1) is shown. For part (2), let (x,y) be a solution of (5.11). Then, for any $f^\infty \in C(D)$, $x \geq P(f)x$, implying that $x \geq P^n(f)x$ for all $n \in \mathbb{N}$, and consequently, $x \geq P^*(f)x$. Furthermore, since $0 = P^*(f)\{x - P(f)\}$ and all elements of $P^*(f)$ and $x - P(f)$ are nonnegative, $p_{ij}^*(f)\{x - P(f)x\}_j = 0$ for all $i,j \in S$, implying that $p_{ii}^*(f)\{x - P(f)x\}_i = 0$ for all $i, \in S$. For an ergodic state i, $p_{ii}^*(f) > 0$, and consequently $x_i - \sum_j p_{ij}(f)x_j = 0$, i.e. $f(i) \in A(i,x)$, and therefore, by (5.11) $x_i + y_i \geq r_i(f) + \sum_j p_{ij}(f)y_j$.

The columns of $P^*(f)$ corresponding to the transient states are zero, implying that $P^*(f)(x+y) \geq P^*(f)\{r(f)+P(f)y\} = \phi(f^\infty) + P^*(f)y$, i.e.

$$\phi(f^\infty) \leq P^*(f)x \leq x. \tag{5.14}$$

On the other hand, any solution of system (5.11) gives a policy g^∞ which satisfies $x = P(g)x$ and $x + y = r(g) + P(g)y$. Hence, $x = P^*(g)x$ and therefore,

$$\phi(g^\infty) = P^*(g)r(g) = P^*(g)\{x + y - P(g)y\} = x + P^*(g)\{y - P(g)y\} = x. \tag{5.15}$$

From (5.14) and (5.15) it follows that $x_i = max_{a \in A(i)} \sum_j p_{ij}(a)x_j = \phi_i, \; i \in S$. □

Remarks

1. Since the x-vector in (5.11) is unique, namely $x = \phi$, the set $A(i,x)$ is also unique for all $i \in S$.
2. If policy f^∞ satisfies $\phi = P(f)\phi$ and $\phi + y = r(f) + P(f)y$ for some vector y, then the policy is average optimal, namely $\phi = P^*(f)\phi = P^*(f)\{r(f) + P(f)y - y\} = \phi(f^\infty)$.
3. The proof suggests that a Blackwell optimal policy f_0^∞ is also average optimal, i.e. $\phi(f_0^\infty) \geq \phi(R)$ for every policy R. This result is shown below (Corollary 5.4).
4. If ϕ has identical components (e.g. if there is a unichain average optimal policy), then the first equation of (5.11) is superfluous and (5.11) can be replced by the single optimality equation

$$x + y_i = max_{a \in A(i)}\{r_i(a) + \sum_j p_{ij}(a)y_j\}, \; i \in S. \tag{5.16}$$

Theorem 5.12

$lim_{\alpha \uparrow 1}\, (1-\alpha)v^\alpha(R) \geq \phi(R)$ for all policies R.

Proof

Let R be an arbitrary policy, i any starting state and define $x_t := \sum_{(j,a)} \mathbb{P}_{i,R}\{X_t = j, \; Y_t = a\} \cdot r_j(a)$ for $t = 1, 2, \dots$. Since the sequence $\{x_t \mid t = 1, 2, \dots\}$ is bounded, we may write

$$(1-\alpha)^{-1}v_i^\alpha(R) = \big\{ \textstyle\sum_{t=1}^\infty \alpha^{t-1} \big\} \cdot \{\sum_{t=1}^\infty \alpha^{t-1}x_t\} = \sum_{t=1}^\infty \big\{ \sum_{s=1}^t x_s \big\} \cdot \alpha^{t-1}.$$

$(1-\alpha)^{-2} = \sum_{t=1}^\infty t\alpha^{t-1}$ for $\alpha \in (0,1)$, and therefore, $\phi_i(R) = \big\{ \sum_{t=1}^\infty t\alpha^{t-1} \big\} \cdot (1-\alpha)^2 \cdot \phi_i(R)$. Hence, $(1-\alpha)v_i^\alpha(R) - \phi_i(R) = (1-\alpha)^2 \cdot \sum_{t=1}^\infty \big\{ \frac{1}{t}\sum_{s=1}^t \, x_s - \phi_i(R) \big\} \cdot t\alpha^{t-1}$. Choose an arbitrary $\varepsilon > 0$. Since $\phi_i(R) = \liminf_{T\to\infty} \frac{1}{T}\sum_{t=1}^T \, x_t$, there exists T_ε such that $\phi_i(R) < \frac{1}{T}\sum_{t=1}^T \, x_t + \varepsilon$ for all $T > T_\varepsilon$. Therefore, we obtain

$(1-\alpha)^2 \sum_{t>T_\varepsilon} \big\{ \frac{1}{t}\sum_{s=1}^t x_s - \phi_i(R) \big\} t\alpha^{t-1} > -\varepsilon(1-\alpha)^2 \sum_{t>T_\varepsilon} t\alpha^{t-1} \geq -\varepsilon(1-\alpha)^2 \sum_{t=1}^\infty t\alpha^{t-1} = -\varepsilon$.

We also have,

$(1-\alpha)^2 \sum_{t \leq T_\varepsilon} \big\{ \frac{1}{t}\sum_{s=1}^t x_s - \phi_i(R) \big\} t\alpha^{t-1} \geq (1-\alpha)^2 min_{1 \leq t \leq T_\varepsilon} \big\{ \frac{1}{t}\sum_{s=1}^t x_s - \phi_i(R) \big\} \sum_{t \leq T_\varepsilon} t\alpha^{t-1} > -\varepsilon$

for α sufficiently close to 1. Hence, $(1-\alpha)v_i^\alpha(R) - \phi_i(R) \geq -2\varepsilon$ for α sufficiently close to 1, i.e. $\lim_{\alpha\uparrow 1}\, (1-\alpha)v^\alpha(R) \geq \phi(R)$. □

Corollary 5.4

A Blackwell optimal policy f_0^∞ is also average optimal and consequently there exists a deterministic optimal policy.

Proof

Let f_0^∞ be a Blackwell optimal policy and R an arbitrary policy. Then,

$$\phi(f_0^\infty) = \lim_{\alpha\uparrow 1}\, (1-\alpha)v^\alpha(f_0^\infty) = \lim_{\alpha\uparrow 1}\, (1-\alpha)v^\alpha \geq \lim_{\alpha\uparrow 1}\, (1-\alpha)v^\alpha(R) \geq \phi(R).$$ □

5.7 Policy iteration

In policy iteration a sequence of policies $f_1^\infty, f_2^\infty, \dots$ is constructed such that $\phi(f_{k+1}^\infty) \geq \phi(f_k^\infty)$ and $v^\alpha(f_{k+1}^\infty) > v^\alpha(f_k^\infty)$ for all $\alpha \in (\alpha_k, 1)$. Hence, each new policy in the sequence differs from the previous policies. Since $C(D)$ is finite, the policy iteration method terminates after a finite number of iteration.

Theorem 5.13
Consider the following system

$$\left\{\begin{array}{ccccccc} \{I-P(f)\}x & & & & & = & 0 \\ & x & + & \{I-P(f)\}y & & = & r(f) \\ & & & y & + \ \{I-P(f)\}z & = & 0 \end{array}\right. \tag{5.17}$$

Then, for every $f^\infty \in C(D)$, the system (5.17) has a solution $\big(x(f), y(f), z(f)\big)$, where $x(f)$ and $y(f)$ are unique with $x(f) = u^{-1}(f) = \phi(f^\infty)$ and $y(f) = u^0(f)$.

Proof
First, we will show that $x(f) = u^{-1}(f)$, $y(f) = u^0(f)$ and $z(f) = u^1(f)$ is a solution of (5.17). We use the properties $P^*(f)D(f) = 0$ and $(I - P(f))D(f) = I - P^*(f)$ (see Theorem 5.6).

$$\begin{aligned} \{I-P(f)\}x(f) &= \{I-P(f)\}u^{-1}(f) = \{I-P(f)\}P^*(f)r(f) = 0. \\ x(f) + \{I-P(f)\}y(f) &= P^*(f)r(f) + \{I-P(f)\}D(f)r(f) \\ &= \{P^*(f) + \big(I-P(f)\big)D(f)\}r(f) = r(f). \\ y(f) + \{I-P(f)\}z(f) &= D(f)r(f) - \{I-P(f)\}D^2(f)r(f) \\ &= \{I - \big(I-P(f)\big)D(f)\}D(f)r(f) = P^*(f)D(f)r(f) = 0. \end{aligned}$$

Next, we show the second part of the theorem. Let (x, y, z) be any solution of (5.17). Then, $x = P(f)x$ implies $x = P^*(f)x = P^*(f)\{r(f) - \big(I-P(f)\big)y\} = P^*(f)r(f) = u^{-1}(f)$. Since $y + \{I-P(f)\}z = 0$, we have $P^*(f)y = 0$, and consequently, $\{I - P(f) + P^*(f)\}y = \{I-P(f)\}y = r(f) - P^*(f)r(f)$, i.e.

$$\begin{aligned} y &= \{I - P(f) + P^*(f)\}^{-1}\{I - P^*(f)\}r(f) \\ &= Z(f)\{I-P^*(f)\}r(f) = \{D(f) + P^*(f)\}\{I-P^*(f)\}r(f) = D(f)r(f) = u^0(f). \end{aligned}$$ □

For every $i \in S$ and $f^\infty \in C(D)$, the action set $B(i,f)$ is defined by

$$B(i,f) = \left\{ a \in A(i) \;\middle|\; \begin{array}{l} \sum_j p_{ij}(a)\phi_j(f^\infty) > \phi_i(f^\infty) \text{ or} \\ \sum_j p_{ij}(a)\phi_j(f^\infty) = \phi_i(f^\infty) \text{ and } r_i(a) + \sum_j p_{ij}(a)u_j^0(f) > \phi_i(f^\infty) + u_i^0(f) \end{array} \right\}. \tag{5.18}$$

Theorem 5.14

(1) If $B(i,f) = \emptyset$ for every $i \in S$, then f^∞ is an average optimal policy.

(2) If $B(i,f) \neq \emptyset$ for at least one $i \in S$ and the policy $g^\infty \neq f^\infty$ satisfies $g(i) \in B(i,f)$ if $g(i) \neq f(i)$, then $\phi(g^\infty) \geq \phi(f^\infty)$ and $v^\alpha(g^\infty) > v^\alpha(f^\infty)$ for α sufficiently close to 1.

Proof
(1) Since $B(i,f) = \emptyset$ for every $i \in S$, we have, for any $h^\infty \in C(D)$, $\sum_j p_{ij}(h)\phi_j(f^\infty) \leq \phi_i(f^\infty)$ and $r_i(h) + \sum_j p_{ij}(h)u_j^0(f) \leq \phi_i(f^\infty) + u_i^0(f)$ if $\sum_j p_{ij}(h)\phi_j(f^\infty) = \phi_i(f^\infty)$.

Let $R = (h, f, f, \dots)$. Then, $v^\alpha(R) = r(h) + \alpha P(h)v^\alpha(f^\infty)$ and, by Theorem 5.10,

$$\begin{aligned}\alpha v^\alpha(f^\infty) &= \tfrac{\alpha}{1-\alpha}\phi(f^\infty) + u^0(f) + \varepsilon_1(\alpha) = \{1-(1-\alpha)\}\tfrac{\phi(f^\infty)}{1-\alpha} + u^0(f) + \varepsilon_1(\alpha)\cdot e \\ &= \tfrac{\phi(f^\infty)}{1-\alpha} + u^0(f) - \phi(f^\infty) + \varepsilon_1(\alpha)\cdot e,\end{aligned}$$

where $\varepsilon_1(\alpha)$ is such that $\lim_{\alpha\uparrow 1}\ \varepsilon_1(\alpha) = 0$. Furthermore, we have

$$\begin{aligned}v^\alpha(R) &= r(h) + P(h)\Big\{\tfrac{\phi(f^\infty)}{1-\alpha} + u^0(f) - \phi(f^\infty) + \varepsilon_1(\alpha)\cdot e\Big\} \\ &= \tfrac{P(h)\phi(f^\infty)}{1-\alpha} + r(h) + P(h)u^0(f) - P(h)\phi(f^\infty) + \varepsilon_1(\alpha)\cdot e.\end{aligned}$$

Since $v^\alpha(f^\infty) = \frac{\phi(f^\infty)}{1-\alpha} + u^0(f) + \varepsilon_2(\alpha)\cdot e$, where $\varepsilon_2(\alpha)$ is such that $\lim_{\alpha\uparrow 1}\ \varepsilon_2(\alpha) = 0$, we have

$$v^\alpha(f^\infty) - v^\alpha(R) = \frac{1}{1-\alpha}\{\phi(f^\infty) - P(h)\phi(f^\infty)\} + \{u^0(f) - r(h) - vP(h)u^0(f) + P(h)\phi(f^\infty)\} + \varepsilon_3(\alpha)\cdot e. \quad (5.19)$$

Since $\phi(f^\infty) - P(h)\phi(f^\infty) \geq 0$ and because, when $\{\phi(f^\infty) - P(h)\phi(f^\infty)\}_i = 0$, $\{u^0(f) - r(h) - P(h)u^0(f) + P(h)\phi(f^\infty)\}_i = \{u^0(f) - r(h) - P(h)u^0(f) + \phi(f^\infty)\}_i \geq 0$, we obtain $v^\alpha(f^\infty) - v^\alpha(R) \geq \varepsilon_3(\alpha)\cdot e$ for α sufficiently close to 1, where $\varepsilon_3(\alpha)$ is such that $\lim_{\alpha\uparrow 1}\ \varepsilon_3(\alpha) = 0$. Hence, $v^\alpha(f^\infty) \geq v^\alpha(R) + \varepsilon_3(\alpha)\cdot e = r(h) + \alpha P(h)v^\alpha(f^\infty) + \varepsilon_3(\alpha)\cdot e$, and consequently, $\{I - \alpha P(h)\}v^\alpha(f^\infty) \geq r(h) + \alpha P(h)v^\alpha(f^\infty) + \varepsilon_3(\alpha)\cdot e$. Therefore,

$$v^\alpha(f^\infty) \geq \{I - \alpha P(h)\}^{-1}\{r(h) + \varepsilon_3(\alpha)\cdot e\} = v^\alpha(h^\infty) + \tfrac{\varepsilon_3(\alpha)}{1-\alpha}\cdot e.$$

From the Laurent expansion it follows that $\phi(f^\infty) \geq \phi(h^\infty)$, i.e. f^∞ is an average optimal policy.

(2) Let $R = (g, f, f, \dots)$. Then,

if $g(i) = f(i)$, then the ith rows of $P(f)$ and $P(g)$ are identical and $r_i(f) = r_i(g)$, i.e.

$$v_i^\alpha(R) = \{r(g) + \alpha P(g)v^\alpha(f^\infty)\}_i = \{r(f) + \alpha P(f)v^\alpha(f^\infty)\}_i = v_i^\alpha(f^\infty).$$

if $g(i) \neq f(i)$, then $g(i) \in B(i,f)$, and because (5.19) holds for $h = g$, we have

$$v_i^\alpha(f^\infty) - v_i^\alpha(R) = \tfrac{1}{1-\alpha}\{\phi(f^\infty) - P(h)\phi(f^\infty)\}_i + \{u^0(f) - r(g) - P(g)u^0(f) + P(g)\phi(f^\infty)\}_i$$
$$+\ \varepsilon_3(\alpha)\cdot e \text{ for } \alpha \text{ sufficiently close to 1.}$$

Hence, for α sufficiently close to 1, $v^\alpha(R) = r(g) + \alpha P(g)v^\alpha(f^\infty) > v^\alpha(f^\infty)$, i.e.

$$\{I - \alpha P(g)\}v^\alpha(f^\infty) > r(g) \ \rightarrow\ v^\alpha(f^\infty) > \{I - \alpha P(g)\}^{-1}r(g) = v^\alpha(f^\infty).$$

Again, by the Laurent expansion, it follows that $\phi(g^\infty) \geq \phi(f^\infty)$. □

Algorithm 5.6 *Determination of an average optimal policy by policy iteration (version 1)*
Input: Instance of an MDP.
Output: An optimal deterministic policy f^∞ and the value vector ϕ.

1. Select an arbitrary $f^\infty \in C(D)$.

2. Determine $\phi(f^\infty)$ and $u^0(f)$ as unique (x,y)-part in a solution of the system

$$\begin{cases}\{I - P(f)\}x & & & = 0 \\ x & + \ \{I - P(f)\}y & & = r(f) \\ & y & + \ \{I - P(f)\}z & = 0\end{cases}$$

3. Determine for every $i \in S$

$$B(i,f) := \left\{ a \in A(i) \ \middle|\ \begin{array}{l}\sum_j p_{ij}(a)\phi_j(f^\infty) > \phi_i(f^\infty) \text{ or} \\ \sum_j p_{ij}(a)\phi_j(f^\infty) = \phi_i(f^\infty) \text{ and } r_i(a) + \sum_j p_{ij}(a)u_j^0(f) > \phi_i(f^\infty) + u_i^0(f)\end{array} \right\}.$$

```
4. if $B(i,f) = \emptyset$ for every $i \in S$ then
       begin $f^\infty$ is an average optimal policy; $\phi(f^\infty)$ is the value vector $\phi$; STOP end
   else begin take $g$ such that $g \neq f$ and $g(i) \in B(i,f)$ if $g(i) \neq f(i)$; $f := g$; return to step 2
       end
```

Example 5.5
Consider the MDP of Example 3.1. Start with the policy f^∞, where $f(1) = 3,\ f(2) = 2$ and $f(3) = 1$.
Iteration 1:
The solution of the linear system gives: $\phi(f^\infty) = \big(\frac{11}{2}, 4, \frac{11}{2}\big),\ u^0(f) = \big(-\frac{5}{4}, 0, \frac{5}{4}\big)$.
$B(1,f) = \emptyset,\ B(2,f) = \{1,3\};\ B(3,f) = \{3\}.\ g(1) = 3,\ g(2) = 3,\ g(3) = 3.\ f(1) = 3,\ f(2) = 3,\ f(3) = 3.$
Iteration 2:
The solution of the linear system gives: $\phi(f^\infty) = (7,7,7),\ u^0(f) = (-4,-2,0).\ B(1,f) = \emptyset,\ B(2,f) = \emptyset,$ $B(3,f) = \emptyset.$ f^∞ is an optimal policy and $\phi(f^\infty) = (7,7,7)$ is the value vector.

Modified algorithms
In this subsection we first show that the third part of the system in step 2 of Algorithm 5.6, i.e. the subsystem $y + I - P(f)z = 0$, cannot be deleted; otherwise, the policy iteration algorithm may cycle. Without the subsystem $y + I - P(f)z = 0$, the policy iteration algorithm becomes as follows.

Algorithm 5.7 *Determination of an average optimal policy by policy iteration (second version)*
Input: Instance of an MDP.
Output: An optimal deterministic policy f^∞ and the value vector ϕ.

1. Select an arbitrary $f^\infty \in C(D)$.
2. Determine $\big(x = \phi(f^\infty), y\big)$ as (x,y)-part of the system
$$\begin{cases} \{I - P(f)\}x & & = 0 \\ x & + \quad \{I - P(f)\}y & = r(f) \end{cases}$$
3. Determine for every $i \in S$
$$B(i,f) := \left\{ a \in A(i) \;\middle|\; \begin{array}{l} \sum_j p_{ij}(a)\phi_j(f^\infty) > \phi_i(f^\infty) \text{ or} \\ \sum_j p_{ij}(a)\phi_j(f^\infty) = \phi_i(f^\infty) \text{ and } r_i(a) + \sum_j p_{ij}(a)y_j > \phi_i(f^\infty) + y_i \end{array} \right\}.$$

```
4. if $B(i,f) = \emptyset$ for every $i \in S$ then
       begin $f^\infty$ is an average optimal policy; $\phi(f^\infty)$ is the value vector $\phi$; STOP end
   else begin take $g$ such that $g \neq f$ and $g(i) \in B(i,f)$ if $g(i) \neq f(i)$; $f := g$; return to step 2
       end
```

Example 5.6
$S = \{1,2,3\};\ A(1) = A(2) = \{1\},\ A(3) = \{1,2\};\ r_1(1) = r_2(1) = r_3(1) = r_3(2) = 0.$
$p_{11}(1) = 1,\ p_{12}(1) = p_{13}(1) = 0;\ p_{21}(1) = 0,\ p_{22}(1) = 1,\ p_{23}(1) = 0;\ p_{31}(1) = 1,\ p_{32}(1) = p_{33}(1) = 0;$
$p_{31}(2) = 0,\ p_{32}(2) = 1,\ p_{33}(2) = 0.$
This model has two deterministic policies: f_1^∞ and f_2^∞ with $f_1(3) = 1$ and $f_2(3) = 2$.
All policies are optimal and the value vector $\phi = (0,0,0,)$.
For policy f_1^∞ the linear system gives: $x_1 = x_2 = x_3 = 0;\ y_1 = y_3$ and y_2 can arbitrarily chosen.
For policy f_2^∞ the linear system gives: $x_1 = x_2 = x_3 = 0;\ y_2 = y_3$ and y_1 can arbitrarily chosen.

For policy f_1^∞ the sets $B(i, f_1)$ are: $B(1, f_1) = B(2, f_1) = \emptyset$; $B(3, f_1) = \begin{cases} \emptyset & \text{if } y_2 \leq y_1; \\ \{2\} & \text{if } y_2 > y_1. \end{cases}$

For policy f_2^∞ the sets $B(i, f_2)$ are: $B(1, f_2) = B(2, f_2) = \emptyset$; $B(3, f_2) = \begin{cases} \emptyset & \text{if } y_1 \leq y_2; \\ \{2\} & \text{if } y_1 > y_2. \end{cases}$

If we start with policy f_1^∞ and if we take $y_1 = 0,\ y_2 = 1,\ y_3 = 0$, then the next policy is policy f_2^∞. For policy f_2^∞, we can take $y_1 = 1,\ y_2 = 0,\ y_3 = 0$, which gives as next policy f_1^∞. So, we have a cycle.

We will present additional requirements to the solution y such that the policy iteration method has foolproof convergence to an average optimal policy. Therefore, we first analyze the set of solutions (x, y) of the linear system

$$\{I - P(f)\}x = 0;\ x + \{I - P(f)\}y = r(f). \tag{5.20}$$

Any solution of (5.20) satisfies $x = P^*(f)r(f)$ and consequently,

$$\{I - P(f)\}y = r(f) - \phi(f^\infty). \tag{5.21}$$

A solution y of equation (5.21) is called a *relative value vector*. It can easily be verified that a solution of equation (5.21) is given by $y^1 = Z(f)\{r(f) - \phi(f^\infty)\}$. Hence, for any solution y of (5.21), we have $\{I - P(f)\}(y - y^1) = 0$. Denote by $n(f)$ the number of subchains (i.e. closed, irreducible sets of states) of $P(f)$. Then, we know from the theory of Markov chains (cf. Section 5.3) that

$$\{P^*(f)\}_{ij} = \sum_{m=1}^{n(f)} a_i^m(f)\pi_j^m(f),\ i, j \in S. \tag{5.22}$$

where the row vector $\pi^m(f)$ is the unique stationary distribution on the mth subchain of $P(f)$ and $a_i^m(f)$ is the absorption probability in the mth subchain, starting from state i. Note that the column vectors $a^m(f)$ satisfy $\{I - P(f)\}a^m(f) = 0,\ 1 \leq m \leq n(f)$, and that the vectors $a^1(f), a^2(f), \dots, a^{n(f)}(f)$ are linearly independent. Since any solution of $\{I - P(f)\}x = 0$ implies $\{I - P^*(f)\}x = 0$ and because the rank of $\{I - P^*(f)\}$ is $N - n(f)$, it follows that any solution of $\{I - P(f)\}x = 0$ is given by

$$x = \sum_{m=1}^{n(f)} c_m a^m(f) \tag{5.23}$$

with $c_1, c_2, \dots, c_{n(f)}$ arbitrary scalars. Consequently, by (5.23), any solution y of (5.21) satisfies

$$y = Z(f)\{r(f) - \phi(f^\infty)\} + \sum_{m=1}^{n(f)} c_m a^m(f) \tag{5.24}$$

with $c_1, c_2, \dots, c_{n(f)}$ arbitrary scalars. We also have

$$\begin{aligned} Z(f)\{r(f) - \phi(f^\infty)\} &= \{D(f) + P^*(f)\}\{r(f) - \phi(f^\infty)\} \\ &= D(f)r(f) - D(f)P^*(f)r(f) + P^*(f)r(f) - P^*(f)P^*(f)r(f) \\ &= D(f)r(f) = u^0(f). \end{aligned}$$

Therefore, we have shown the following result.

Lemma 5.5
If (x, y) is a solution of (5.20), then $x = \phi(f^\infty)$ and $y = u^0(f) + \sum_{m=1}^{n(f)} c_m a^m(f)$, with $c_1, c_2, \dots, c_{n(f)}$ arbitrary scalars.

In order to prevent cycling in Algorithm 5.7, the following rules have been proposed in the literature.
Rule 1:
Set $y_i = 0$ for the smallest i within each subchain of $P(f)$.
Rule 2:
Choose y such that $P^*(f)y = 0$.
Rule 3:
For any two policies f_1^∞ and f_2^∞ that have a common subchain C and that select identical actions in all states belonging to C, the relative value vectors $y(f_1)$ and $y(f_2)$ are chosen such that $y_i(f_1) = y_i(f_2)$ for all states $i \in C$.

Notes

1. The requirement of rule 3 is feasible since it follows from Lemma 5.5 and the block structure of $Z(f)$ that the y-values of the states belonging to one subchain depend only upon the actions selected within that subchain.
2. Consider the choice of y according to rule 1. Then, $c_m = -u^0_{i(m)}(f),\ m = 1, 2, \dots, n(f)$, where $i(m)$ is the smallest i within subchain m of $P(f)$. In this case rule 3 is verified, because if on common subchains identical actions are chosen the same relative values are obtained. So, rule 1 is a special case of rule 3.
3. Consider the solution (x, y) chosen in Algorithm 5.6. Then, from the third system in step 2 of this algorithm we obtain $P^*(f)y = 0$, so in this algorithm rule 2 is satisfied. Because $P^*(f)y = 0$, we have $\{I - P(f) + P^*(f)\}y = r(f) - \phi(f^\infty)$, implying $y = Z(f)\{r(f) - \phi(f^\infty)\}$, i.e. in this version we choose the scalars $c_m = 0,\ m = 1, 2, \dots, n(f)$. So, Algorithm 5.6 satisfies rule 2 and rule 2 is a special case of rule 3.
4. Note that Example 5.6 does not satisfy rule 3. Namely, the two policies f_1^∞ and f_2^∞ have the subchains $\{1\}$ and $\{2\}$, and select in these subchains the only available action 1. Therefore, by rule 3, for both relative values vectors, say y^1 and y^2, we require $y^1_1 = y^2_1$ and $y^1_2 = y^2_2$, which is not the case in this example, because there we have chosen $y^1 = (0, 1, 0)$ and $y^2 = (1, 0, 0)$.
5. The rules 1 and 3 require the determination of the subchains for each policy generated by Algorithm 5.7. This needs $\mathcal{O}(N^2)$ additional operations in each iteration.

Next, we present a modified version of Algorithm 5.7 in which rule 1 is implemented. Moreover, we show the convergence of this algorithm.

Algorithm 5.8 *Determination of an average optimal policy by policy iteration (third version)*
Input: Instance of an MDP.
Output: An optimal deterministic policy f^∞ and the value vector ϕ.

1. Select an arbitrary $f^\infty \in C(D)$.
2. Determine the ergodic structure of $P(f)$ and let $n(f)$ be the number of subchains.
3. Determine $\big(x = \phi(f^\infty), y\big)$ as (x, y)-part of the system
$$\begin{cases} \{I - P(f)\}x & & = & 0 \\ x & + \ \{I - P(f)\}y & = & r(f) \end{cases}$$
with $y_i = 0$ for the smallest i within each subchain of $P(f)$.
4. (a) **for all** $(i, a) \in S \times A$ **do** $s_i(a) := \sum_j p_{ij}(a)\phi_j(f^\infty)$

   (b) **for all** $i \in S$ **do** $A_1(i) := \{a_1 \mid s_i(a_1) \geq s_i(a) \text{ for all } a \in A(i)\}$

```
        (c) **for all** $i \in S$ **do** choose $g(i) \in A_1(i)$, setting $g(i) = f(i)$ if possible
        (d) **if** $g(i) = f(i)$ **for all** $i \in S$ **then go to** step 5
            **otherwise begin for all** $i \in S$ **do** $f(i) := g(i)$; **go to** step 2 **end**
    5.  (a) **for all** $(i, a) \in S \times A$ **do** $t_i(a) := r_i(a) + \sum_j p_{ij}(a) y_j$
        (b) **for all** $i \in S$ **do** $A_2(i) := \{a_2 \in A_1(i) \mid t_i(a_2) \geq t_i(a_1)$ for all $a_1 \in A_1(i)\}$
        (c) **for all** $i \in S$ **do** choose $g(i) \in A_2(i)$, setting $g(i) = f(i)$ if possible
        (d) **if** $g(i) = f(i)$ **for all** $i \in S$ **then**
                **begin** $f^\infty$ is an average optimal policy; $x = \phi(f^\infty)$ is the value vector $\phi$; STOP **end**
            **otherwise begin for all** $i \in S$ **do** $f(i) := g(i)$; **go to** step 2 **end**
```

The improvement step of the algorithm consists of two phases. First, improvement is sought through the first optimality equation. If no strict improvement is possible, we seek an improvement decision rule through the second optimality equation. When none improvement is available, the algorithm terminates.

Example 5.7

$S = \{1, 2, 3\};\ A(1) = A(2) = \{1, 2\},\ A(3) = \{1\};\ r_1(1) = 3,\ r_1(2) = 1;\ r_2(1) = 0,\ r_2(2) = 1;\ r_3(1) = 2.$
$p_{11}(1) = 1,\ p_{12}(1) = p_{13}(1) = 0;\ p_{11}(2) = 0,\ p_{12}(1) = 1,\ p_{13}(1) = 0;\ p_{21}(1) = 0,\ p_{22}(1) = 1,\ p_{23}(1) = 0;$
$p_{21}(2) = p_{22}(2) = 0,\ p_{23}(2) = 1;\ p_{31}(1) = p_{32}(1) = 0,\ p_{33}(1) = 1.$
Start with the policy $f(1) = 2,\ f(2) = 1,\ f(3) = 1.$
Iteration 1:
$P(f)$ has two subchains with states $\{2\}$ and $\{3\}$, respectively. The linear system becomes:

$$\begin{array}{rcl}
x_1 - x_2 & = & 0 \\
x_1 + y_1 - y_2 & = & 1 \\
x_2 & = & 0 \\
x_3 & = & 2 \\
y_2 & = & 0 \\
y_3 & = & 0
\end{array}$$

The unique solution of this system is: $x = (0, 0, 2),\ y = (1, 0, 0).$
$s_1(1) = s_1(2) = s_2(1) = 0,\ s_2(2) = 2,\ s_3(1) = 2;\ A_1(1) = \{1, 2\},\ A_1(2) = \{2\},\ A_1(3) = \{1\}.$
$g(1) = 2,\ g(2) = 2,\ g(3) = 1;\ f(1) = 2,\ f(2) = 2,\ f(3) = 1.$
Iteration 2:
$P(f)$ has one subchain with state $\{3\}$. The linear system becomes:

$$\begin{array}{rcl}
x_1 - x_2 & = & 0 \\
x_2 - x_3 & = & 0 \\
x_1 + y_1 - y_2 & = & 1 \\
x_2 + y_2 - y_3 & = & 1 \\
x_3 & = & 2 \\
y_3 & = & 0
\end{array}$$

The unique solution of this system is: $x = (2, 2, 2),\ y = (-2, -1, 0).$
$s_1(1) = s_1(2) = s_2(1) = s_2(2) = s_3(1) = 2;\ A_1(1) = \{1, 2\},\ A_1(2) = \{1, 2\},\ A_1(3) = \{1\}.$
$g(1) = 1,\ g(2) = 2,\ g(3) = 1;\ f(1) = 1,\ f(2) = 2,\ f(3) = 1.$

Iteration 3:

$P(f)$ has two subchain with states $\{1\}$ and $\{3\}$, respectively. The linear system becomes:

$$\begin{array}{ccccccccccc}
 & & x_2 & - & x_3 & & & & & = & 0 \\
x_1 & & & & & & & & & = & 3 \\
 & & x_2 & & & + & y_2 & - & y_3 & = & 1 \\
 & & & & x_3 & & & & & = & 2 \\
 & & & & & y_1 & & & & = & 0 \\
 & & & & & & & & y_3 & = & 0
\end{array}$$

The unique solution of this system is: $x = (3, 2, 2),\ y = (0, -1, 0)$.

$s_1(1) = 3,\ s_1(2) = s_2(1) = s_2(2) = s_3(1) = 2;\ A_1(1) = \{1\},\ A_1(2) = \{1, 2\},\ A_1(3) = \{1\}$.

$g(1) = 1,\ g(2) = 2,\ g(3) = 1$.

$t_1(1) = 3,\ t_1(2) = 0,\ t_2(1) = -1,\ t_2(2) = 1,\ t_3(1) = 2;\ A_2(1) = \{1\},\ A_2(2) = \{2\},\ A_1(3) = \{1\}$.

$g(1) = 1,\ g(2) = 2,\ g(3) = 1$. Since $g = f$, policy f^∞ with $f(1) = 1,\ f(2) = 2,\ f(3) = 1$ is an average optimal policy with value vector $\phi = (3, 2, 2)$.

Denote by $y(f)$ an arbitrary y-solution to (5.20).

Lemma 5.6

Let $f^\infty, g^\infty \in C(D)$ and define the vectors $u,\ v,\ \Delta\,\phi$ and $\Delta\,y$ by:

$u := \{P(g)-I\}\phi(f^\infty),\ v := r(g)+\{P(g)-I\}y(f)-\phi(f^\infty),\ \Delta\,\phi := \phi(g^\infty)-\phi(f^\infty)$ and $\Delta\,y := y(g)-y(f)$.

Then, $u = \{I - P(g)\}\Delta\,\phi,\ v = \{I - P(g)\}\Delta\,y + \Delta\,\phi$ and $P^*(g)u = P^*(g)\{v - \Delta\,\phi\} = 0$.

Proof

$$\begin{aligned}
\{I - P(g)\}\Delta\,\phi &= \{I - P(g)\}\{\phi(g^\infty) - \phi(f^\infty)\} = 0 - \{I - P(g)\}\phi(f^\infty) = u. \\
\{I - P(g)\}\Delta\,y &= \{I - P(g)\}\{y(g) - y(f)\} = r(f) - \phi(f^\infty) - \{I - P(g)\}y(f) \\
&= v + \phi(f^\infty) - \phi(g^\infty) = v - \Delta\,\phi.
\end{aligned}$$

Since $P^*(g)\{I - P(g)\} = 0$, obviously, $P^*(g)u = 0$ and $P^*(g)\{v - \Delta\,\phi\} = 0$. □

Let $P(g)$ have m subchains. Then, $P(g)$ and $P^*(g)$ can, after renumbering of the states, be written in standard form as (cf. (5.4) and (5.7)):

$$P = \begin{pmatrix}
P_1(g) & 0 & \cdot & \cdot & \cdot & \cdot & \cdot & 0 \\
0 & P_2(g) & 0 & \cdot & \cdot & \cdot & \cdot & 0 \\
\cdot & \cdot & \cdot & \cdot & & & \cdot & 0 \\
\cdot & & \cdot & \cdot & \cdot & & \cdot & 0 \\
\cdot & & & \cdot & \cdot & \cdot & \cdot & 0 \\
0 & \cdot & \cdot & \cdot & \cdot & 0 & P_m(g) & 0 \\
A_1(g) & A_2(g) & \cdot & \cdot & \cdot & \cdot & A_m(g) & Q(g)
\end{pmatrix} \text{ and } P^* = \begin{pmatrix}
P_1^*(g) & 0 & \cdot & \cdot & \cdot & \cdot & \cdot & 0 \\
0 & P_2^*(g) & 0 & \cdot & \cdot & \cdot & \cdot & 0 \\
\cdot & \cdot & \cdot & \cdot & & & \cdot & \cdot \\
\cdot & & \cdot & \cdot & \cdot & & \cdot & \cdot \\
\cdot & & & \cdot & \cdot & \cdot & \cdot & \cdot \\
0 & \cdot & \cdot & \cdot & \cdot & 0 & P_m^*(g) & 0 \\
A_1^*(g) & A_2^*(g) & \cdot & \cdot & \cdot & \cdot & A_m^*(g) & 0
\end{pmatrix}.$$

Partition the vectors $\phi(f^\infty),\ \phi(g^\infty),\ y(f),\ y(g),\ \Delta\,\phi,\ \Delta\,y,\ u$ and v consistent with the above partition. Denote $u = (u^1, u^2, \dots, u^m, u^{m+1})$ and for the vectors $\phi(f^\infty),\ \phi(g^\infty),\ y(f),\ y(g),\ \Delta\,\phi,\ \Delta\,y$ and v, we use a similar partition.

Lemma 5.7

Suppose that $u^i = 0$ for $i = 1, 2, \dots, m$. Then, we have

(1) $(\Delta\,\phi)^i = P_i^*(g)v^i$ for $i = 1, 2, \dots, m$.

(2) $(\Delta\,\phi)^{m+1} = \{I - Q(g)\}^{-1}\{u^{m+1} + \sum_{j=1}^m A_j(g)(\Delta\,\phi)^j\}$.

(3) $(\Delta\,y)^{m+1} = \{I - Q(g)\}^{-1}\{v^{m+1} - (\Delta\,\phi)^{m+1} + \sum_{j=1}^m A_j(g)(\Delta\,y)^j\}$.

Proof

(1) Take any $1 \le i \le m$. Since, by Lemma 5.6, $\{I - P_i(g)\}(\Delta\,\phi)^i = u^i = 0$, we have $(\Delta\,\phi)^i = P_i(g)(\Delta\,\phi)^i$, implying $(\Delta\,\phi)^i = P_i^*(g)(\Delta\,\phi)^i$. Also from Lemma 5.6, we have $v^i = \{I - P_i(g)\}(\Delta\,y)^i + (\Delta\,\phi)^i$, and consequently, $P_i^*(g)v^i = 0 + P_i^*(g)(\Delta\,\phi)^i = (\Delta\,\phi)^i$.

(2) By Lemma 5.6, $u^{m+1} = \big[\{I - P(g)\}(\Delta\,\phi)\big]^{m+1} = -\sum_{j=1}^m A_j(g)(\Delta\,\phi)^j + \{I - Q(g)\}(\Delta\,\phi)^{m+1}$. Hence, $(\Delta\,\phi)^{m+1} = \{I - Q(g)\}^{-1}\{u^{m+1} + \sum_{j=1}^m A_j(g)(\Delta\,\phi)^j\}\}$.

(3) $v^{m+1} = \big[\{I - P(g)\}(\Delta\,y + \Delta\,\phi)\big]^{m+1} = -\sum_{j=1}^m A_j(g)(\Delta\,y)^j + \{I - Q(g)\}(\Delta\,y)^{m+1} + (\Delta\,\phi)^{m+1}$. Therefore, $(\Delta\,y)^{m+1} = \{I - Q(g)\}^{-1}\{v^{m+1} - (\Delta\,\phi)^{m+1} + \sum_{j=1}^m A_j(g)(\Delta\,y)^j\}$. □

Theorem 5.15

Let f^∞ and g^∞ be two subsequent policies in Algorithm 5.8. Suppose that $P(g)$ has m subchains $R_1, R_2, \dots, R_m$ and let T the set of transient states. Then,

(1) If $\{P(g)\phi(f^\infty)\}_j > \phi_j(f^\infty)$ for some stat j and if $g(i) = f(i)$ for all i with $\{P(g)\phi(f^\infty)\}_j = \phi_j(f^\infty)$, then $j \in T$ and $\phi_j(g^\infty) > \phi_j(f^\infty)$.

(2) If $P(g)\phi(f^\infty) = \phi(f^\infty)$ and $\big[r(g) - \phi(f^\infty) - \{I - P(g)\}y(f)\big]_j > 0$ for some state $j \in R_i$ and if $g(i) = f(i)$ for all i with $\big[r(g) - \phi(f^\infty) - \{I - P(g)\}y(f)\big]_i = 0$, then $\phi_k(g^\infty) > \phi_k(f^\infty)$ for all $k \in R_i$.

(3) If $P(g)\phi(f^\infty) = \phi(f^\infty)$ and $\big[r(g) - \phi(f^\infty) - \{I - P(g)\}y(f)\big]_j = 0$ for all states $j \in \cup_{i=1}^m R_i$, and furthermore, $\big[r(g) - \phi(f^\infty) - \{I - P(g)\}y(f)\big]_k > 0$ for some $k \in T$, then $\phi(g^\infty) = \phi(f^\infty)$ and $y_k(g) > y_k(f)$.

Proof

Let $u, v, \Delta\,\phi$ and $\Delta\,y$ be defined as in Lemma 5.6. Then, by the determination of g in step 4 of Algorithm 5.8, we have $P(g)\phi(f^\infty) \ge \phi(f^\infty)$, i.e. $u \ge 0$.

(1) Since $P^*(g)u = 0$ (see Lemma 5.6) and because all elements of $P_i^*(g)$ are strictly positive, $u^i = 0$ for $i = 1, 2, \dots, m$. Therefore, if $u_j > 0$, i.e. $\{P(g)\phi(f^\infty)\}_j > \phi_j(f^\infty)$, then state j is transient and $u_j^{m+1} > 0$. Because $g(i) = f(i)$ for all states i that are recurrent in the Markov chain $P(g)$, the submatrices of $P(g)$ and $P(f)$ with respect to $\cup_{i=1}^m R_i$ are identical. Consequently, we obtain

$$\begin{array}{rcl} (\Delta\,\phi)^i & = & P_i^*(g)v^i = P_i^*(g)\big\{r^i(g) + \{P_i(g) - I\}y^i(g) - \phi^i(g^\infty)\big\} \\ & = & P_i^*(g)r^i(g) - P_i^*(g)\phi^i(g^\infty)\} = \phi^i(g^\infty)\} - \phi^i(g^\infty)\} = 0,\ i = 1, 2, \dots, m. \end{array}.$$

Noting that $\{I - Q(g)\}^{-1} \ge I$ and applying Lemma 5.7 part (2), we have $(\Delta\,\phi)^{m+1} = \{I - Q(g)\}^{-1}\{u^{m+1} + \sum_{i=1}^m A_i(g)(\Delta\,\phi)^i\} \ge u^{m+1}$ and consequently, $(\Delta\,\phi)_j^{m+1} \ge u_j^{m+1} > 0$, i.e. $\phi_j(g^\infty) > \phi_j(f^\infty)$.

(2) Since $u = 0$ and $v_j^i > 0$ for some $j \in R_i$, by Lemma 5.7 part (1), we have $(\Delta\,\phi)_k^i = \{P_i^*(g)v^i\}_k > 0$ for all $k \in R_i$, i.e. $\phi_k(g^\infty) > \phi_k(f^\infty)$ for all $k \in R_i$.

(3) Since $u = 0$ and $v^i = 0$ for $i = 1, 2, \dots, m$, part (1) and (2) of Lemma 5.7 imply that $(\Delta\,\phi)^i = 0$ for $i = 1, 2, \dots, m+1$, i.e. $\phi(g^\infty) = \phi(f^\infty)$. Therefore, by Lemma 5.6, $\{I - P_i(g)\}(\Delta\,y)^i = v^i = 0$ for $i = 1, 2, \dots, m$. Consequently, $(\Delta\,y)^i = P_i^*(g)\}(\Delta\,y)^i$, so the vector $(\Delta\,y)^i$ has identical components for $i = 1, 2, \dots, m$. Hence, by Rule 1, $(\Delta\,y)^i = 0$ for $i = 1, 2, \dots, m$. By Lemma 5.7 part (3), we have $(\Delta\,y)^{m+1} = \{I - Q(g)\}^{-1}\{v^{m+1} - (\Delta\,\phi)^{m+1} + \sum_{i=1}^m A_i(g)(\Delta\,\phi)^i\} = \{I - Q(g)\}^{-1}v^{m+1} \geq v^{m+1}$. Since $v_k^{m+1} > 0$ for some $k \in T$, we obtain $(\Delta\,y)_k^{m+1} \geq u_k^{m+1} > 0$, i.e. $y_k(g) > y_k(f)$. □

Theorem 5.16
Algorithm 5.8 terminates in a finite number of iterations with an average optimal policy and also with the value vector.

Proof
From Theorem 5.15 it follows that Algorithm 5.8 has the following properties:

a. The average reward vectors of successive policies are monotone nondecreasing.

b. If improvement occurs for state j in step 4 (c) of the algorithm, then j is transient under $P(g)$ and $\phi_j(g^\infty) > \phi_j(f^\infty)$. Note that $\phi_k(g^\infty) > \phi_k(f^\infty)$ may hold for other states which are transient under $P(g)$.

c. If no improvement occurs in step 4 (c) and it occurs in step 5 (c) of the algorithm for state j, where j in recurrent under $P(g)$, then $\phi_k(g^\infty) > \phi_k(f^\infty)$ for all states in the same recurrent class as j, and possibly for other states which are transient under $P(g)$.

d. If no improvement occurs in step 4 (c) and it occurs in step 5 (c) of the algorithm for state k, where k in transient under $P(g)$, then $y_k(g) > y_k(f)$.

By the above observations, Algorithm 5.8 terminates in a finite number of iterations. At termination the policy f^∞ satisfies:

$$\begin{cases} \phi(f^\infty) & \geq & P(h)\phi(f^\infty) & \text{for all } h^\infty \in C(D) \\ r(f) + P(f)y(f) & \geq & r(h) + P(h)y(f) & \text{for all } h^\infty \in C(D) \end{cases}$$

Furthermore, we have, by step 3 of Algorithm 5.8, the relation $\phi(f^\infty) + \{I - P(f)\}y(f) = r(f)$. Hence, $\phi(f^\infty) \geq P^*(h)\phi(f^\infty)$ and $\phi(f^\infty) + y(f) \geq r(h) + P(h)y(f)$ for all $h^\infty \in C(D)$. The last inequality implies $P^*(h)\phi(f^\infty) \geq P^*(h)r(h) = \phi(h^\infty)$ for all $h^\infty \in C(D)$. Therefore, $\phi(f^\infty) \geq P^*(h)\phi(f^\infty) = \phi(h^\infty)$ for all $h^\infty \in C(D)$, i.e. f^∞ is an average optimal policy. Furthermore, Algorithm 5.8 provides in step 3 the value vector $x = \phi(f^\infty)$. □

5.8 Linear programming

To obtain the value vector and an average optimal policy by linear programming, we need a property for which the value vector is an extreme element. Such property, called superharmonicity, can be derived from the optimality equation. In the context of average reward, a vector $v \in \mathbb{R}^N$ is *superharmonic* if there exists a vector $u \in \mathbb{R}^N$ such that the pair (u, v) satisfies the following system of inequalities

$$\begin{cases} v_i & \geq & \sum_j p_{ij}(a)v_j & \text{for every } (i,a) \in S \times A \\ v_i + u_i & \geq & r_i(a) + \sum_j p_{ij}(a)u_j & \text{for every } (i,a) \in S \times A \end{cases} \qquad (5.25)$$

Theorem 5.17
The value vector ϕ is the (componentswise) smallest superharmonic vector.

Proof

Let f_0^∞ be a Blackwell optimal policy. From Theorem 5.11 it follows that

$$\left\{ \begin{array}{lll} \phi_i & \geq & \sum_j p_{ij}(a)\phi_j \qquad\qquad\qquad \text{for every } i \in S,\ a \in A(i) \\ \phi_i + u_i^0(f_0) & \geq & r_i(a) + \sum_j p_{ij}(a)u_j^0(f_0) \quad \text{for every } i \in S,\ a \in A(i,\phi) \end{array} \right. \tag{5.26}$$

where $A(i,\phi) := \{a \in A(i) \mid \phi_i = \sum_j p_{ij}(a)\phi_j\},\ i \in S.$

Let $A^*(i) := \{a \in A(i) \mid \phi_i + u_i^0(f_0) < r_i(a) + \sum_j p_{ij}(a)u_j^0(f_0)\},\ i \in S$, and define

$$s_i(a) := \phi_i - \textstyle\sum_j p_{ij}(a)\phi_j;\ t_i(a) := \phi_i + u_i^0(f_0) - r_i(a) - \sum_j p_{ij}(a)u_j^0(f_0),\ (i,a) \in S \times A,$$

$$u := u^0(f_0) - M \cdot \phi. \text{ where } M := \left\{ \begin{array}{ll} min\{\frac{t_i(a)}{s_i(a)} \,\Big|\, a \in A^*(i),\ i \in S\} & \text{if } \cup_{i\in S}\ A^*(i) \neq \emptyset \\ \qquad\qquad 0 & \text{if } \cup_{i\in S}\ A^*(i) = \emptyset \end{array} \right.$$

Note that $M \leq 0$.

For $a \in A(i,\phi)$, we have

$$\left\{ \begin{array}{l} \phi_i = \sum_j p_{ij}(a)\phi_j; \\ \phi_i + u_i = \phi_i + u_i^0(f_0) - M \cdot \phi_i \geq r_i(a) + \sum_j p_{ij}(a)\{u_j^0(f_0) - M \cdot \phi_j\} = r_i(a) + \sum_j p_{ij}(a)u_j. \end{array} \right.$$

For $a \in A^*(i)$, we have

$$\left\{ \begin{array}{lll} \phi_i > \sum_j p_{ij}(a)\phi_j; & & \\ \quad \phi_i + u_i & = & \phi_i + u_i^0(f_0) - M \cdot \{s_i(a) + \sum_j p_{ij}(a)\phi_j\} \\ & = & t_i(a) + r_i(a) + \sum_j p_{ij}(a)u_j^0(f_0) - M \cdot s_i(a) - M \cdot \sum_j p_{ij}(a)\phi_j \geq r_i(a) + \sum_j p_{ij}(a)u_j. \end{array} \right.$$

For $a \notin \{a \in A(i,\phi) \cup A^*(i)\}$, we have

$$\left\{ \begin{array}{lll} \phi_i > \sum_j p_{ij}(a)\phi_j; & & \\ \quad \phi_i + u_i & = & \phi_i + u_i^0(f_0) - M \cdot \phi_i \geq t_i(a) + r_i(a) + \sum_j p_{ij}(a)\{u_j^0(f_0) - M \cdot \phi_j\} \\ & = & t_i(a) + r_i(a) + \sum_j p_{ij}(a)u_j \geq r_i(a) + \sum_j p_{ij}(a)u_j. \end{array} \right.$$

Hence, the value vector ϕ is superharmonic.

Suppose that y is also superharmonic with corresponding vector x, Then, $y \geq P(f_0)y$, implying that $y \geq P^*(f_0)y \geq P^*(f_0)\{r(f_0) + \big(P(f_0) - I\big)x\} = P^*(f_0)r(f_0) = \phi(f_0^\infty) = \phi$, i.e. ϕ is the smallest superharmonic vector. □

Corollary 5.5

From the proof of Theorem 5.17 it follows that there exists a solution of the modified optimality equation

$$\left\{ \begin{array}{lll} x_i & = & max_{a\in A(i)} \sum_j p_{ij}(a)x_j \qquad\qquad , \ i \in S \\ x_i + y_i & = & max_{a\in A(i)}\{r_i(a) + \sum_j p_{ij}(a)y_j\} \quad , \ i \in S \end{array} \right. \tag{5.27}$$

with $x = \phi$ as unique x-vector in this solution.

Example 5.8

$S = \{1,2,3\};\ A(1) = A(2) = \{1,2\},\ A(3) = \{1\};\ r_1(1) = 3,\ r_1(2) = 1,\ r_2(1) = 0,\ r_2(2) = 1;\ r_3(1) = 2.$
$p_{11}(1) = 1, p_{12}(1) = p_{13}(1) = 0;\ p_{11}(2) = 0, p_{12}(2) = 1, p_{13}(2) = 0;\ p_{21}(1) = 0, p_{22}(1) = 1, p_{23}(1) = 0;$ $p_{21}(2) = p_{22}(2) = 0,\ p_{23}(1) = 1;\ p_{31}(1) = p_{32}(1) = 0, p_{33}(1) = 1.$

The modified optimality equation for this model is:

$x_1 = max\{x_1, x_2\};\ x_2 = max\{x_2\};\ x_3 = max\{x_3\}.$

$x_1 + y_1 = max\{3 + y_1, 1 + y_2\};\ x_2 + y_2 = max\{0 + y_2, 1 + y_3\};\ x_3 + y_3 = max\{2 + y_3\}.$

This equation has as solution $x = (3,2,2)$ and $y = (a, b-1, b)$ for any a and b with $3 + a \geq b$.

The original optimality equation is considerably more complex, because the equations in the second part depend on the values of x_1, x_2 and x_3.

Corollary 5.6
The value vector ϕ is the unique v-part of an optimal solution (u,v) of the linear program

$$min \left\{ \sum_j \beta_j v_j \;\middle|\; \begin{array}{rcll} \sum_j \{\delta_{ij} - p_{ij}(a)\} v_j & \geq & 0 & \text{for every } (i,a) \in S \times A \\ v_i + \sum_j \big(\delta_{ij} - p_{ij}(a)\big) u_j & \geq & r_i(a) & \text{for every } (i,a) \in S \times A \end{array} \right\}, \tag{5.28}$$

where $\beta_j > 0,\ j \in S$, is arbitrarily chosen.

The dual linear program of (5.28) is

$$max \left\{ \sum_{(i,a)} r_i(a) x_i(a) \;\middle|\; \begin{array}{lrcl} \sum_{(i,a)} \{\delta_{ij} - p_{ij}(a)\} x_i(a) & & = & 0,\ j \in S \\ \sum_a x_j(a) & + \sum_{(i,a)} \{\delta_{ij} - p_{ij}(a)\} y_i(a) & = & \beta_j,\ j \in S \\ & x_i(a), y_i(a) & \geq & 0,\ (i,a) \in S \times A \end{array} \right\}. \tag{5.29}$$

Theorem 5.18
Let (x,y) be an extreme optimal solution of (5.29). Then, any $f^\infty \in C(D)$, where $x_i\big(f(i)\big) > 0$ if $\sum_a x_i(a) > 0$ and $y_i\big(f(i)\big) > 0$ if $\sum_a x_i(a) = 0$, is an average optimal policy.

Proof
First, notice that f^∞ is well defined, because for every $j \in S$,

$$\textstyle\sum_a x_j(a) + \sum_a y_j(a) = \sum_{(i,a)} p_{ij}(a) y_i(a) + \beta_j > 0,\ j \in S.$$

Let $S_x := \big\{i \in S \mid \sum_a x_i(a) > 0\big\}$. Since $x_i\big(f(i)\big) > 0,\ i \in S_x$, and $y_i\big(f(i)\big) > 0,\ i \notin S_x$, it follows from the complementary slackness property of linear programming that

$$\phi_i + \sum_j \{\delta_{ij} - p_{ij}\big(f(i)\big)\} u_j = r_i\big(f(i)\big),\ i \in S_x \tag{5.30}$$

and

$$\sum_j \{\delta_{ij} - p_{ij}\big(f(i)\big)\} \phi_j = 0,\ i \notin S_x. \tag{5.31}$$

The primal program (5.28) implies $\sum_j \{\delta_{ij} - p_{ij}(a)\}\phi_j \geq 0,\ (i,a) \in S \times A$. Suppose that for some $k \in S_x$, $\sum_j \{\delta_{kj} - p_{kj}(f(k))\}\phi_j > 0$. Since $x_k\big(f(k)\big) > 0$, this implies that $\sum_j \{\delta_{kj} - p_{kj}(f(k))\}\phi_j \cdot x_k\big(f(k)\big) > 0$. Furthermore, $\sum_j \{\delta_{ij} - p_{ij}(a)\}\phi_j \cdot x_i(a) \geq 0,\ (i,a) \in S \times A$. Hence,

$$\sum_{(i,a)} \sum_j \{\delta_{ij} - p_{ij}(a)\}\phi_j \cdot x_i(a) > 0.$$

On the other hand, this result is contradictory to the constraints of the dual program (5.29) from which follows that

$$\sum_{(i,a)} \sum_j \{\delta_{ij} - p_{ij}(a)\}\phi_j \cdot x_i(a) = \sum_j \Big\{ \sum_{(i,a)} \big(\delta_{ij} - p_{ij}(a)\big) x_i(a) \Big\} \phi_j = 0.$$

This contradiction implies that

$$\sum_j \{\delta_{ij} - p_{ij}\big(f(i)\big)\} \phi_j = 0,\ i \in S_x. \tag{5.32}$$

From (5.31) and (5.32) it follows that

$$\sum_j \{\delta_{ij} - p_{ij}\big(f(i)\big)\} \phi_j = 0,\ i \in S. \tag{5.33}$$

Next, we show that S_x is closed under $P(f)$, i.e. $p_{ij}\big(f(i)\big) = 0,\ i \in S_x,\ j \notin S_x$. Suppose that $p_{kl}\big(f(k)\big) > 0$ for some $k \in S_x,\ l \notin S_x$. From the constraints of dual program (5.29) it follows that

$$0 = \sum_a x_l(a) = \sum_{(i,a)} p_{il}(a)x_i(a) \geq p_{kl}\big(f(k)\big)x_k\big(f(k)\big) > 0, \tag{5.34}$$

implying a contradiction. We now show that the states of $S \backslash S_x$ are transient in the Markov chain induced by $P(f)$. Suppose that $S \backslash S_x$ has an ergodic state. Since S_x is closed, the set $S \backslash S_x$ contains an ergodic class, say $J = \{j_1, j_2, \ldots, j_m\}$. Since (x, y) is an extreme solution and $y_j\big(f(j)\big) > 0,\ j \in J$, the corresponding columns in (5.29) are linearly independent. Because these columns have zeros in the first N rows, the second parts of these vectors are also independent vectors. Since $\delta_{jk} - p_{jk}\big(f(j)\big) = 0 - 0 = 0$ for $j \in J$ and $k \notin J$, the vectors $b^i,\ 1 \leq i \leq m$, where b^i has components $\delta_{j_ik} - p_{j_ik}\big(f(j_i)\big),\ k \in J$, are also linear independent. However, $\sum_{k=1}^m b^i_k = \sum_{k=1}^m \{\delta_{j_ij_k} - p_{j_ij_k}\big(f(j_i)\big)\} = 1 - 1 = 0,\ i = 1, 2, \ldots, m$, which contradicts the independence of $b^1, b^2, \ldots, b^m$.
We finish the proof as follows. From relation (5.32) it follows that $\phi = P(f)\phi$, and consequently we have $\phi = P^*(f)\phi$. Since that states of $S \backslash S_x$ are transient in the Markov chain induced by $P(f)$, the columns of $P^*(f)$ corresponding to $S \backslash S_x$ are zero-vectors. Hence, by (5.30),

$$\phi(f^\infty) = P^*(f)r(f) = P^*(f)\Big\{\phi + \{I - P(f)\}u\Big\} = P^*(f)\phi = \phi,$$

i.e. f^∞ is an average optimal policy. □

Algorithm 5.9 *Determination of an average optimal policy by linear programming*
Input: Instance of an MDP.
Output: An optimal deterministic policy f^∞ and the value vector ϕ.

1. Select any vector β, where $\beta_j > 0,\ j \in S$.

2. Use the simplex method to compute optimal solutions (u, v) and (x, y) of the linear programs

$$min\left\{\sum_j \beta_j v_j \;\middle|\; \begin{array}{rcll} \sum_j \{\delta_{ij} - p_{ij}(a)\}v_j & \geq & 0 & \text{for every } (i,j) \in S \times A \\ v_i + \sum_j \big(\delta_{ij} - p_{ij}(a)\big)u_j & \geq & r_i(a) & \text{for every } (i,j) \in S \times A \end{array}\right\}$$

and

$$max\left\{\sum_{(i,a)} r_i(a)x_i(a) \;\middle|\; \begin{array}{lll} \sum_{(i,a)}\{\delta_{ij} - p_{ij}(a)\}x_i(a) & & = 0,\ j \in S \\ \sum_a x_j(a) & + \sum_{(i,a)}\{\delta_{ij} - p_{ij}(a)\}y_i(a) & = \beta_j,\ j \in S \\ & x_i(a), y_i(a) & \geq 0,\ (i,a) \in S \times A \end{array}\right\}.$$

3. Take $f^\infty \in C(D)$ such that $x_i\big(f(i)\big) > 0$ if $\sum_a x_i(a) > 0$ and $y_i\big(f(i)\big) > 0$ if $\sum_a x_i(a) = 0$.

4. f^∞ is an average optimal policy and v is the value vector (STOP).

The next example shows an optimal solution (x, y) of the dual program (5.29) which has in some state i more than one positive $x_i(a)$ or $y_i(a)$ variable.

Example 5.9
Consider the MDP of Example 3.1. The dual linear program is:

$$max\{x_1(1) + 2x_1(2) + 3x_1(3) + 6x_2(1) + 4x_2(2) + 5x_2(3) + 8x_3(1) + 9x_3(2) + 7x_3(3)\}$$

subject to the constraints

$$\begin{array}{l}
x_1(2)+x_1(3)-x_2(1)-x_3(1)=0\\
x_2(1)+x_2(3)-x_1(2)-x_3(2)=0\\
x_3(1)+x_3(2)-x_1(3)-x_2(3)=0\\
x_1(1)+x_1(2)+x_1(3)+y_1(2)+y_1(3)-y_2(1)-y_3(1)=\frac{1}{3}\\
x_2(1)+x_2(2)+x_2(3)+y_2(1)+y_2(3)-y_1(2)-y_3(2)=\frac{1}{3}\\
x_3(1)+x_3(2)+x_3(3)+y_3(1)+y_3(2)-y_1(3)-y_2(3)=\frac{1}{3}\\
x_1(1),x_1(2),x_1(3),x_2(1),x_2(2),x_2(3),x_3(1),x_3(2),x_3(3)\geq 0
\end{array}$$

If we solve this linear program, we obtain:
$x_1(1)=x_1(2)=x_1(3)=x_2(1)=x_2(2)=0,\ x_2(3)=\frac{1}{2},\ x_3(1)=0,\ x_3(2)=\frac{1}{2},\ x_3(3)=0.$
$y_1(1)=0,\ y_1(2)=\frac{1}{6},\ y_1(3)=\frac{1}{6},\ y_2(1)=y_2(2)=y_2(3)=y_3(1)=y_3(2)=y_3(3)=0.$
Because $x_1(1)=x_1(2)=x_1(3)$ and $y_1(1)=0,\ y_1(2)=\frac{1}{6},\ y_1(3)=\frac{1}{6}$, we can take in state 1 both action 2 and 3 for an optimal action.

In the average reward case there is in general no one-to-one correspondence between the feasible solutions of the dual program (5.29) and the set of stationary policies. The natural formula for mapping feasible solutions (x,y) to the set of stationary policies is:

$$\pi_{ia}^{x,y}:=\begin{cases}\frac{x_i(a)}{\sum_a x_i(a)},\ a\in A(i),\ i\in S_x;\\[4pt] \frac{y_i(a)}{\sum_a y_i(a)},\ a\in A(i),\ i\in S\backslash S_x.\end{cases}$$

In the next example two different solutions are mapped on the same deterministic policy.

Example 5.10
Consider the MDP with $S=\{1,2,3,4\};\ A(1)=\{1\},\ A(2)=A(3)=\{1,2\},\ A(4)\{1\};$
$r_1(1)=r_2(1)=r_2(2)=r_3(1)=r_3(2)=r_4(1)=1;$
$p_{11}(1)=0,\ p_{12}(1)=1,\ p_{13}(1)=p_{14}(1)=0;\ p_{21}(1)=p_{22}(1)=0,\ p_{23}(1)=1,\ p_{24}(1)=0;$
$p_{21}(2)=p_{22}(2)=p_{23}(2)=0,\ p_{24}(2)=1;\ p_{31}(1)=p_{32}(1)=0,\ p_{33}(1)=1,\ p_{34}(1)=0;$
$p_{31}(2)=1,\ p_{32}(2)=p_{33}(2)=p_{34}(2)=0;\ p_{41}(1)=p_{42}(1)=p_{43}(1)=0,\ p_{44}(1)=1.$
The dual linear program becomes (take $\beta_1=\beta_2=\beta_3=\beta_4=\frac{1}{4}$).
$max\{x_1(1)+x_2(1)+x_2(2)+x_3(1)+x_3(2)+x_4(1)\}$
subject to the constraints

$$\begin{array}{rrrrrrrrrrcl}
x_1(1) & & & & -\ x_3(2) & & & & & & = & 0\\
-\ x_1(1) & +\ x_2(1) & +\ x_2(2) & & & & & & & & = & 0\\
 & -\ x_2(1) & & & +\ x_3(2) & & & & & & = & 0\\
 & & -\ x_2(2) & & & & & & & & = & 0\\
x_1(1) & & & & & & +\ y_1(1) & & & -\ y_3(2) & = & \frac{1}{4}\\
 & x_2(1) & +\ x_2(2) & & & & -\ y_1(1) & +\ y_2(1) & +\ y_2(2) & & = & \frac{1}{4}\\
 & & & x_3(1) & +\ x_3(2) & & & -\ y_2(1) & & +\ y_3(2) & = & \frac{1}{4}\\
 & & & & & x_4(1) & & & -\ y_2(2) & & = & \frac{1}{4}
\end{array}$$

$x_1(1),x_2(1),x_2(2),x_3(1),x_3(2),x_4(1),y_1(1),y_2(1),y_2(2),y_3(2)\geq 0.$

The following two feasible solutions (x^1,y^1) and (x^2,y^2) are mapped on the same deterministic policy f^∞, where $f(1)=f(2)=1,\ f(3)=2$ and $f(4)=1:$
$x_1^1(1)=x_2^1(1)=\frac{1}{4},\ x_2^1(2)=x_3^1(1)=0,\ x_3^1(2)=x_4^1(1)=\frac{1}{4};\ y_1^1(1)=y_2^1(1)=y_2^1(2)=y_3^1(2)=0$ and
$x_1^2(1)=x_2^2(1)=\frac{1}{6},\ x_2^2(2)=x_3^2(1)=0,\ x_3^2(2)=\frac{1}{6},\ x_4^2(1)=\frac{1}{2};\ y_1^2(1)=\frac{1}{6},\ y_2^2(1)=y_2^2(2)=\frac{1}{4},\ y_3^2(2)=\frac{1}{12}.$

For any $\pi^\infty \in C(S)$ we can define a feasible solution (x^π, y^π) of the dual program as follows. Consider the Markov chain induced by $P(\pi)$ and suppose that this Markov chain has m recurrent sets, say $S_1, S_2, \dots, S_m$, and let T be the set of transient states. Define (x^π, y^π) by

$$\begin{aligned} x_i^\pi(a) &:= \{\beta^T P^*(\pi)\}_i \cdot \pi_{ia},\ (i,a) \in S \times A & (5.35)\\ y_i^\pi(a) &:= \{\beta^T D(\pi) + \gamma^T P^*(\pi)\}_i \cdot \pi_{ia},\ (i,a) \in S \times A, & (5.36)\end{aligned}$$

where $\gamma_i := \begin{cases} 0 & i \in T; \\ max_{l\in S_j} \Big\{ - \frac{\sum_{k\in S} \beta_k d_{kl}(\pi)}{\sum_{k\in S_j} p^*_{kl}(\pi)} \Big\} & i \in S_j,\ 1 \le j \le m. \end{cases}$

Notice that γ is constant on every ergodic set.

Theorem 5.19

(x^π, y^π), defined by (5.35) and (5.36), is a feasible solution of the dual program (5.29).

Proof

$\sum_{(i,a)} \{\delta_{ij} - p_{ij}(a)\} x_i^\pi(a) = \sum_j x_j^\pi(a) - \sum_{(i,a)} p_{ij}(a) x_i^\pi(a) = \{\beta^T P^*(\pi)\}_j - \{\beta^T P^*(\pi)P(\pi)\}_j = 0.$

$\sum_j x_j^\pi(a) + \sum_{(i,a)} \{\delta_{ij} - p_{ij}(a)\} y_i^\pi(a)$

$$\begin{aligned} &= \{\beta^T P^*(\pi)\}_j + \{\beta^T D(\pi) + \gamma^T P^*(\pi)\}_j - \{\beta^T D(\pi)P(\pi) + \gamma^T P^*(\pi)P(\pi)\}_j \\ &= \big\{\beta^T\{P^*(\pi) + D(\pi)(I - P(\pi)\}\big\}_j + \{\gamma^T P^*(\pi)(I - P(\pi))\}_j \\ &= \big\{\beta^T\{P^*(\pi) + I - P^*(\pi)\}\big\}_j = \beta_j. \end{aligned}$$

The nonnegativity of $x_i^\pi(a)$ is obvious. For the nonnegativity of $y_i^\pi(a)$ we distinguish between $i \in T$ and $i \in S_j$ for some $1 \le j \le m$. Notice that $y_i^\pi(a) = \{\sum_k \beta_k d_{ki}(\pi) + \sum_k \gamma_k p^*_{ki}(\pi)\} \cdot \pi_{ia}$.

If $i \in T$:

$p^*_{ki}(\pi) = 0$ for all k and therefore, by Theorem 5.7, $d_{ki}(\pi) = \sum_{t=0}^\infty \{P^t(\pi)\}_{ki}$.

Hence, $y_i^\pi(a) = \{\sum_k \beta_k \cdot (\sum_{t=0}^\infty \{P^t(\pi)\}_{ki}) \cdot \pi_{ia} \ge 0$.

If $i \in S_j$:

$p^*_{ki}(\pi) = 0$ for all $k \notin (S_j \cup T)$. Hence,

$$\begin{aligned} y_i^\pi(a) &= \{\textstyle\sum_{k\in S} \beta_k d_{ki}(\pi) + \sum_{k\in S_j} \gamma_k p^*_{ki}(\pi)\} \cdot \pi_{ia} \\ &\ge \Big\{\textstyle\sum_{k\in S} \beta_k d_{ki}(\pi) + \sum_{k\in S_j} \big\{ - \frac{\sum_{k\in S} \beta_k d_{ki}(\pi)}{\sum_{k\in S_j} p^*_{ki}(\pi)} \big\} p^*_{ki}(\pi)\Big\} \cdot \pi_{ia} \\ &= \textstyle\sum_{k\in S} \beta_k d_{ki}(\pi) - \sum_{k\in S} \beta_k d_{ki}(\pi) = 0. \end{aligned}$$

□

Theorem 5.20

The correspondence between the stationary policies and the feasible solutions of program (5.29) preserves the optimality property, i.e.

(1) If π^∞ is an average optimal policy, then (x^π, y^π) is an optimal solution of (5.29).

(2) If (x, y) is an optimal solution of (5.29), then the stationary policy $\pi^{x,y}$ is an average optimal policy.

Proof

(1) Since (x^π, y^π) is feasible for (5.29) it is sufficient to show that $\sum_{(i,a)} r_i(a) x_i^\pi(a) = \sum \beta_j \phi_j$.

$\sum_{(i,a)} r_i(a) x_i^\pi(a) = \sum_{(i,a)} r_i(a) \{\beta^T P^*(\pi)\}_i \cdot \pi_{ia} = \{\beta^T P^*(\pi)\}_i r_i(\pi) = \beta^T \phi(\pi^\infty) = \beta^T \phi.$

(2) The proof of this part has the same structure as the proof of Theorem 5.18.

Suppose that $(v = \phi, u)$ is an optimal solution of the primal program (5.28).

Let $S_x := \big\{i \in S \mid \sum_a x_i(a) > 0\big\}$ and $A^+(i) := \big\{a \in A(i) \mid \pi^{x,y}_{ia} > 0\big\},\ i \in S$. Since $x_i(a) > 0,\ i \in S_x,\ a \in A^+(i)$ and $y_i(a) > 0,\ i \notin S_x,\ a \in A^+(i)$, it follows from the complementary slackness property of linear programming that

$$\phi_i + \sum_j \{\delta_{ij} - p_{ij}(a)\}u_j = r_i(a),\ i \in S_x,\ a \in A^+(i) \tag{5.37}$$

and

$$\sum_j \{\delta_{ij} - p_{ij}(a)\}\phi_j = 0,\ i \notin S_x,\ a \in A^+(i). \tag{5.38}$$

The primal program (5.28) implies $\sum_j \{\delta_{ij} - p_{ij}(a)\}\phi_j \geq 0,\ (i,a) \in S \times A$. Suppose that for some $k \in S_x$ and some $a_k \in A^+(k)$, $\sum_j \{\delta_{kj} - p_{kj}(a_k)\}\phi_j > 0$. Since $\pi^{x,y}_{ka_k} > 0$, we also have $x_{ka_k} > 0$, and $\sum_j \{\delta_{kj} - p_{kj}(a_k)\}\phi_j \cdot x_k(a_k) > 0$. Furthermore, $\sum_j \{\delta_{ij} - p_{ij}(a)\}\phi_j \cdot x_i(a) \geq 0,\ (i,a) \in S \times A$. Hence,

$$\textstyle\sum_{(i,a)} \sum_j \{\delta_{ij} - p_{ij}(a)\}\phi_j \cdot x_i(a) > 0.$$

On the other hand, this result is contradictory to the constraints of the dual program (5.29) from which follows that

$$\textstyle\sum_{(i,a)} \sum_j \{\delta_{ij} - p_{ij}(a)\}\phi_j \cdot x_i(a) = \sum_j \Big\{ \sum_{(i,a)} \big(\delta_{ij} - p_{ij}(a)\big)x_i(a)\Big\}\phi_j = 0.$$

This contradiction implies that

$$\sum_j \{\delta_{ij} - p_{ij}(a)\}\phi_j = 0,\ i \in S_x,\ a \in A^+(i). \tag{5.39}$$

From (5.38) and (5.39) it follows that

$$\sum_j \{\delta_{ij} - p_{ij}(a)\}\phi_j = 0,\ i \in S,\ a \in A^+(i). \tag{5.40}$$

Next, we show that S_x is closed under $P(\pi^{x,y})$, i.e. $p_{ij}(\pi^{x,y}) = 0,\ i \in S_x,\ j \notin S_x$. Suppose that $p_{kl}(\pi^{x,y}) > 0$ for some $k \in S_x,\ l \notin S_x$. Since $p_{kl}(\pi^{x,y}) = \sum_a p_{kl}(a)\pi^{x,y}_{ka}$, there exists an action a_k such that $p_{kl}(a_k) > 0$ and $\pi^{x,y}_{ka_k} > 0$. From the constraints of dual program (5.29) it follows that

$$0 = \sum_a x_l(a) = \sum_{(i,a)} p_{il}(a)x_i(a) \geq p_{kl}(a_k)x_k(a_k) > 0, \tag{5.41}$$

implying a contradiction.

Next, we show that the states of S_x are the recurrent states of the Markov chain induced by $P(\pi^{x,y})$. Let $x_i = \sum_a x_i(a),\ i \in S$. Since $x_i(a) = \pi^{x,y}_{ia} \cdot x_i$ for all (i,a), the constraints of (5.29) imply $x^T = x^T P(\pi^{x,y})$, and consequently, $x^T = x^T P^*(\pi^{x,y})$. Because, for $i \in T$, $x_i = \sum_j x_j p^*_{ji}(\pi^{x,y}) = 0$, we have $T \subseteq S\backslash S_x$. Suppose that $T \neq S\backslash S_x$. Since S_x is closed under $P(\pi^{x,y})$, there exists an ergodic set $S_1 \subseteq S\backslash S_x$. Hence, $0 = \sum_{j \notin S_1} \sum_{i \in S_1} p_{ij}(\pi^{x,y})$, implying $0 = \sum_{j \notin S_1} \sum_{i \in S_1} \sum_a p_{ij}(a)y_i(a)$. We also have, denoting $\sum_a y_i(a)$ by $y_i,\ i \in S$,

$$\begin{aligned}
0 &< \textstyle\sum_{j\in S_1} \beta_j = \sum_{j\in S_1} y_j - \sum_{j\in S_1}\sum_{(i,a)} p_{ij}(a)y_i(a)\\
&= \textstyle\sum_{j\in S_1} y_j - \sum_{j\in S}\sum_{(i,a)} p_{ij}(a)y_i(a) + \sum_{j\notin S_1}\sum_{(i,a)} p_{ij}(a)y_i(a)\\
&= \textstyle\sum_{j\in S_1} y_j - \sum_{j\in S}\sum_{i\in S_1}\sum_a p_{ij}(a)y_i(a) - \sum_{j\in S}\sum_{i\notin S_1}\sum_a p_{ij}(a)y_i(a)\\
&\qquad\qquad \textstyle+\sum_{j\notin S_1}\sum_{i\in S_1}\sum_a p_{ij}(a)y_i(a) + \sum_{j\notin S_1}\sum_{i\notin S_1}\sum_a p_{ij}(a)y_i(a)\\
&= \textstyle\sum_{j\in S_1} y_j - \sum_{j\in S}\sum_{i\in S_1}\sum_a p_{ij}(a)y_i(a) - \sum_{j\in S}\sum_{i\notin S_1}\sum_a p_{ij}(a)y_i(a)\\
&\qquad\qquad\qquad\qquad \textstyle+\sum_{j\notin S_1}\sum_{i\notin S_1}\sum_a p_{ij}(a)y_i(a)\\
&= \textstyle\sum_{j\in S_1} y_j - \sum_{i\in S_1} y_i - \sum_{j\in S}\sum_{i\notin S_1}\sum_a p_{ij}(a)y_i(a) + \sum_{j\notin S_1}\sum_{i\notin S_1}\sum_a p_{ij}(a)y_i(a)\\
&= \textstyle-\sum_{j\in S}\sum_{i\notin S_1}\sum_a p_{ij}(a)y_i(a) + \sum_{j\notin S_1}\sum_{i\notin S_1}\sum_a p_{ij}(a)y_i(a)\\
&= \textstyle-\sum_{j\in S_1}\sum_{i\notin S_1}\sum_a p_{ij}(a)y_i(a) \le 0,
\end{aligned}$$

implying a contraction. So, S_x is the set of the recurrent states in the Markov chain $P(\pi^{x,y})$.

We finish the proof as follows. From (5.40) it follows that

$$\phi_i = \textstyle\sum_j p_{ij}(a)\phi_j,\ i\in S,\ a\in A^+(i) \ = \ \sum_j\sum_a p_{ij}(a)\pi^{x,y}_{ia}\phi_j \ = \ \sum_j p_{ij}(\pi^{x,y})\phi_j,\ i\in S.$$

or, in vector notation, $\phi = P(\pi^{x,y})\phi$, implying $\phi = P^*(\pi^{x,y})\phi$. Since $S\backslash S_x$ is the set of transient states, we have $p^*_{ij}(\pi^{x,y}) = 0,\ j\in S\backslash S_x$. Therefore, we can write using (5.37),

$$\phi(\pi^{x,y}) = P^*(\pi^{x,y})r(\pi^{x,y}) = P^*(\pi^{x,y})\{\phi + \big(I - P(\pi^{x,y})\big)u\} = P^*(\pi^{x,y})\phi = \phi,$$

implying that policy $\pi^{x,y}$ is an average optimal policy. □

We have seen in the proof of Theorem 5.20 that S_x is the set of states that are recurrent in the Markov chain induced by $P(\pi^{x,y})$. In the proof of Theorem 5.18 it was shown that the states of $S\backslash S_x$ are transient in the Markov chain induced by $P(f)$. In the last case S_x may contain also transient states as the next example shows.

Example 5.11

Consider the MDP with $S=\{1,2,3\}$; $A(1)=A(2)=\{1\},\ A(3)=\{1,2\}$; $r_1(1)=1,\ r_2(2)=2,\ r_3(1)=4,$ $r_3(2)=3$. $p_{11}(1)=p_{12}(1)=0,\ p_{13}(1)=1;\ p_{21}(1)=p_{22}(1)=0,\ p_{23}(1)=1;\ p_{31}(1)=1,\ p_{32}(1)=0,$ $p_{33}(1)=0;\ p_{31}(2)=0,\ p_{32}(2)=1,\ p_{33}(3)=0.$

The dual linear program becomes (take $\beta_1=\beta_2=\frac14,\ \beta_3=\frac12$).

$$max\{x_1(1)+2x_2(1)+4x_3(1)+3x_3(2)\}$$

subject to the constraints

$$\begin{array}{rrrrrrrrrrrrrrrrcl}
 & x_1(1) & & & - & x_3(1) & & & & & & & & & & & = & 0\\
 & & & x_2(1) & & & - & x_3(2) & & & & & & & & & = & 0\\
 & x_1(1) & - & x_2(1) & + & x_3(1) & + & x_3(2) & & & & & & & & & = & 0\\
- & x_1(1) & & & & & & & + & y_1(1) & & & - & y_3(1) & & & = & \frac14\\
 & & & x_2(1) & & & & & & & + & y_2(1) & & & - & y_3(2) & = & \frac14\\
 & & & & & x_3(1) & + & x_3(2) & - & y_1(1) & - & y_2(1) & + & y_3(1) & + & y_3(2) & = & \frac12
\end{array}$$

$$x_1(1),\ x_2(1),\ x_3(1),\ x_3(2),\ y_1(1),\ y_2(1),\ y_3(1),\ y_3(2) \ge 0.$$

The solution (x,y) with $x_1(1)=x_2(1)=x_3(1)=x_3(2)=\frac14,\ y_1(1)=y_2(1)=y_3(1)=y_3(2)=0$ is an extreme optimal solution. The two deterministic policies of this model are both optimal policies with $S_x = S = \{1,2,3\}$, and both Markov chains have a transient state in S_x.

If π^∞ is an optimal stationary policy and if (x,y) is a feasible solution of program (5.29) with $\pi^{x,y}=\pi$, then in general (x,y) is not an optimal solution of (5.29). Below we give an example of this phenomenon.

Example 5.12
Consider the MDP with $S = \{1, 2, 3\};\ A(1) = A(2) = \{1, 2\},\ A(3) = \{1\};\ r_1(1) = 1,\ r_1(2) = 0,\ r_2(1) = 0,$ $r_2(2) = 0,\ r_3(1) = 0.\ p_{11}(1) = 0,\ p_{12}(1) = 1,\ p_{13}(1) = 0;\ p_{11}(2) = p_{12}(2) = 0,\ p_{13}(2) = 1;$
$p_{21}(1) = 0,\ p_{22}(1) = 1,\ p_{23}(1) = 0;\ p_{21}(2) = 1,\ p_{22}(2) = p_{23}(2) = 0;\ p_{31}(1) = p_{32}(1) = 0,\ p_{33}(1) = 1.$
The dual linear program becomes (take $\beta_1 = \beta_2 = \beta_3 = \frac{1}{3}$).

$$max\ x_1(1)$$

subject to the constraints

$$\begin{array}{rrrrrrrrrrrrrrrrcl}
 & x_1(1) & + & x_1(2) & & & - & x_2(2) & & & & & & & & & = & 0 \\
- & x_1(1) & & & & & + & x_2(2) & & & & & & & & & = & 0 \\
 & & - & x_1(2) & & & & & & & & & & & & & = & 0 \\
 & x_1(1) & + & x_1(2) & & & & & & & + & y_1(1) & + & y_2(1) & - & y_2(2) & = & \frac{1}{3} \\
 & & & & & x_2(1) & + & x_2(2) & & & - & y_1(1) & & & + & y_2(2) & = & \frac{1}{3} \\
 & & & & & & & & & x_3(1) & & & - & y_2(1) & & & = & \frac{1}{3}
\end{array}$$

$$x_1(1),\ x_1(2),\ x_2(1),\ x_2(2),\ x_3(1),\ y_1(1),\ y_2(1),\ y_2(2) \geq 0.$$

The deterministic policy f^∞ with $f(1) = 1,\ f(2) = 2,\ f(3) = 1$ is an optimal policy. The solution (x, y) with $x_1(1) = \frac{1}{6},\ x_1(2) = 0,\ x_2(1) = 0,\ x_2(2) = \frac{1}{6},\ x_3(1) = \frac{2}{3},\ y_1(1) = 0,\ y_1(2) = \frac{1}{3},\ y_2(2) = \frac{1}{6}$ is a feasible solution with $\pi^{x,y} = f$. However, However, (x, y) is not an optimal solution of the linear program (5.29), because the optimal solution of (5.29) is (x^*, y^*) with $x_1^*(1) = \frac{1}{3},\ x_1^*(2) = 0,\ x_2^*(1) = 0,\ x_2^*(2) = \frac{1}{3},$ $x_3^*(1) = \frac{1}{3},\ y_1^*(1) = 0,\ y_2^*(1) = 0,\ y_3^*(1) = 0.$

Theorem 5.21
Let f^∞ be a deterministic policy. Then, the corresponding feasible solution (x^f, y^f) is an extreme feasible solution of (5.29).

Proof
Suppose that (x^f, y^f) is not an extreme feasible solution of (5.29). Then, there exist feasible solutions (x^1, y^1) and (x^2, y^2) of (5.29) such that $(x^f, y^f) = \lambda \cdot (x^1, y^1) + (1 - \lambda) \cdot (x^2, y^2)$ for some $\lambda \in (0, 1)$ and $(x^1, y^1) \neq (x^2, y^2)$. Since $x_i^f(a) = y_i^f(a) = 0$ for $a \neq f(i), i \in S$, we have $x_i^1(a) = x_i^2(a) = y_i^1(a) = y_i^2(a) = 0$ for $a \neq f(i),\ i \in S$.

Consider the N-dimensional vectors also denoted as x^f, y^f, x^1, y^1, x^2 and y^2 with the components $x_i^f = x_i^f\big(f(i)\big),\ y_i^f = y_i^f\big(f(i)\big),\ x_i^1 = x_i^1\big(f(i)\big),\ x_i^2 = x_i^2\big(f(i)\big),\ y_i^1 = y_i^1\big(f(i)\big),\ y_i^2 = y_i^2\big(f(i)\big)$ for all $i \in S$. Then, these vectors are solutions of the linear system

$$\left\{ \begin{array}{lcrcl} x^T\{I - P(f)\} & & & = & 0 \\ x^T & + & y^T\{I - P(f)\} & = & \beta^T \end{array} \right. \tag{5.42}$$

We shall show that this system has a unique solution, which yields the desired contradiction. From (5.42) it follows that $x^T = x^T P(f)$, and consequently, $x^T = x^T P^*(f) = \beta^T P^*(f)$, implying that the x-part of (5.42) is unique. Furthermore, we obtain from (5.42)

$$y^T\{I - P(f) + P^*(f)\} = \beta^T\{I - P^*(f)\} + y^T P^*(f).$$

Since the matrix $I - P(f) + P^*(f)$ is nonsingular with inverse $Z(f) = D(f) + P^*(f)$ (see Theorem 5.5), we have

$$y^T = \beta^T\{I - P^*(f)\}\{D(f) + P^*(f)\} + y^T P^*(f)\{D(f) + P^*(f)\} = \beta^T D(f) + y^T P^*(f).$$

Consider the Markov chain induced by $P(f)$. Suppose that there are m ergodic sets, say $S_1, S_2, \dots, S_m$, and let T be the set of transient states. Since the columns of $P^*(f)$ corresponding with the transient states T are zero, y is unique on T, namely $y_i = \{\beta^T D(f)\}_i,\ i \in T$. By the definition of γ, which is part of the definition of y^f (see 5.36), in each ergodic set S_k there is a state, say state i_k, such that $y^f_{i_k} = 0$. Then also $y^1_{i_k} = y^2_{i_k} = 0$ for $k = 1, 2, \dots, m$. Define the vector z by $z_i := y^1_i - y^2_i,\ i \in S_k$, and the Markov matrix R by $r_{ij} := \{P(f)\}_{ij}$ for $i, j \in S_k$. Then, the equation (5.42), the uniqueness of x and the property $z_{i_k} = 0$ imply that $z^T = z^T R$, and consequently $z^T = z^T R^*$. Because S_k is an ergodic set, R^* has strictly positive elements and identical rows. Hence, we obtain

$$z_i = \sum_{j \in S_k} z_j r^*_{ji} = r^*_{ii} \cdot \sum_{j \in S_k} z_j,\ i \in S_k. \tag{5.43}$$

Hence, $0 = z_{i_k} = r^*_{i_k i_k} \cdot \sum_{j \in S_k} z_j$. Because $r^*_{i_k i_k} > 0$, we have $\sum_{j \in S_k} z_j = 0$, which implies by (5.43) that $z_i = 0$ for all $i \in S_k$. Therefore, we have shown that $y^1 = y^2$, which implies that also the y-part of (5.42) is unique.

□

Remark

Let f^∞ be the deterministic policy obtained in iteration k of the policy iteration Algorithm 5.6. This policy corresponds, by Theorem 5.21, to an extreme feasible solution of (5.29). Furthermore, by Theorem 5.14, $\phi(f_{k+1}) \geq \phi(f_k)$ for $k = 1, 2, \dots$. The value of the objective function satisfies

$$\textstyle\sum_{(i,a)} r_i(a) x^{f_k}_i(a) = \sum_i r_i(f_k)\{\beta^T P^*(f_k)\}_i = \beta^T P^*(f_k) r(f_k) = \beta^T \phi(f^\infty_k),$$

which is nondecreasing in k. Hence, we have the following result.

Theorem 5.22

The policy iteration Algorithm 5.6 is equivalent to a block-pivoting simplex algorithm.

Example 5.5 (continued)

Since program (5.29) has $2N$ equalities, we introduce artificial variables w_j and z_j for all $j \in S$, where w_j corresponds to $\sum_{(i,a)} \{\delta_{ij} - p_{ij}(a)\} x_i(a) = 0$ and z_j to $\sum_a x_j(a) + \sum_{(i,a)} \{\delta_{ij} - p_{ij}(a)\} y_i(a) = \beta_j$ for all $j \in S$. In this example we take $\beta_1 = \beta_2 = \beta_3 = \frac{1}{3}$ and we start with the policy f^∞, where $f_1(1) = 3,\ f_1(2) = 2$ and $f_1(3) = 1$. Note that in the Markov chain induced by $P(f_1)$ all states are recurrent. Below we state a corresponding simplex tableau with the variables $x_1(3), x_2(2), x_3(1)$ in the basis. They are exchanged with the artificial variables w_1, z_2 and z_1, respectively.

		$x_1(1)$	$x_1(2)$	w_1	$x_2(1)$	z_2	$x_2(3)$	z_1	$x_3(2)$	$x_3(3)$	$y_1(2)$	$y_1(3)$	$y_2(1)$	$y_2(3)$	$y_3(1)$	$y_3(2)$
$x_1(3)$	$\frac{1}{3}$	1	1	0	0	0	0	1	0	0	1	**1**	−1	0	−1	0
w_2	0	0	−1	0	1	0	1	0	−1	0	0	0	0	0	0	0
w_3	0	0	1	1	−1	0	−1	0	1	0	0	0	0	0	0	0
$x_3(1)$	$\frac{1}{3}$	1	0	−1	1	0	0	1	0	0	1	1	−1	0	−1	0
$x_2(2)$	$\frac{1}{3}$	0	0	0	1	1	1	0	0	0	−1	0	1	**1**	0	−1
z_3	0	−1	0	1	−1	0	0	−1	1	**1**	−1	−2	1	−1	2	1
	5	10	1	−8	6	4	−1	11	−9	−7	7	11	−7	4	−11	−4

In the second iteration of the policy iteration method we have the policy f^∞, where $f_2(1) = 3,\ f_2(2) = 3$ and $f_3(3) = 3$. Note that in the Markov chain induced by $P(f_2)$ state 3 is recurrent and the states 1 and 2 are transient. The tableau below is obtained from the previous one by exchanging $x_1(3)$ with $y_1(3)$, $x_2(2)$ with $y_2(3)$ and z_3 with $x_3(3)$ (the pivot elements are bold in the previous tableau).

		$x_1(1)$	$x_1(2)$	w_1	$x_2(1)$	z_2	$x_2(3)$	z_1	$x_3(2)$	z_3	$y_1(2)$	$x_1(3)$	$y_2(1)$	$x_2(2)$	$y_3(1)$	$y_3(2)$
$y_1(3)$	$\frac{1}{3}$	1	1	0	0	0	0	1	0	0	1	1	-1	0	-1	0
w_2	0	0	-1	1	1	0	1	0	-1	0	0	0	0	0	0	0
w_3	0	0	1	-1	-1	0	-1	0	**1**	0	0	0	0	0	0	0
$x_3(1)$	0	0	-1	1	1	0	0	0	0	0	0	-1	0	0	0	0
$y_2(3)$	$\frac{1}{3}$	0	0	1	1	1	1	0	0	0	-1	0	1	1	0	-1
$x_3(2)$	1	1	2	1	0	0	1	1	1	1	0	2	0	1	0	0
	7	6	4	-5	2	7	-2	7	-2	7	0	3	0	3	0	0

This tableau corresponds to the optimal policy f^∞, although it is not an optimal simplex tableau, because the reduced costs of $x_3(2)$ is negative, namely -2. To obtain an optimal simplex tableau we have to exchange $x_3(2)$ with w_3. Below we have this optimal simplex tableau. Notice that it also gives the optimal solution of the primal problem (5.28): $\phi_1 = \phi_2 = \phi_3 = 7$. (the shadow prices of z_1, z_2 and z_3) and $u_1 = -7,\ u_2 = 0,\ u_3 = 2$ (the shadow prices of w_1, w_2 and w_3).

		$x_1(1)$	$x_1(2)$	w_1	$x_2(1)$	z_2	$x_2(3)$	z_1	w_3	z_3	$y_1(2)$	$x_1(3)$	$y_2(1)$	$x_2(2)$	$y_3(1)$	$y_3(2)$
$y_1(3)$	$\frac{1}{3}$	1	1	0	0	0	0	1	0	0	1	1	-1	0	-1	0
w_2	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
$x_3(2)$	0	0	1	-1	-1	0	0	0	1	0	0	0	0	0	0	0
$x_3(1)$	0	0	-1	1	1	0	0	0	0	0	0	-1	0	0	0	0
$y_2(3)$	$\frac{1}{3}$	0	0	1	1	1	1	0	0	0	-1	0	1	1	0	-1
$x_3(2)$	1	1	1	2	1	1	1	1	-1	1	0	2	0	1	0	0
	7	6	6	-7	0	7	0	7	2	7	0	3	0	3	0	0

5.9 Value iteration

For the method of value iteration the following scheme is used:

$$\begin{cases} v_i^{n+1} := max_a\{r_i(a) + \sum_j p_{ij}(a)v_j^n\},\ i \in S,\ n = 0, 1, \dots \\ v_i^0 \text{ arbitrarily chosen},\ i \in S \end{cases} \tag{5.44}$$

Let $f_{n+1}^\infty \in C(D)$ be such that

$$v^{n+1} = r(f_{n+1}) + P(f_{n+1})v^n,\ n = 0, 1, \dots. \tag{5.45}$$

However, in general, neither the sequence $\{v^n\}_{n=0}^\infty$ nor the sequence $\{v^{n+1} - v^n\}_{n=0}^\infty$ is convergent as the next example shows.

Example 5.13
Let $S = \{1, 2\},\ A(1) = A(2) = \{1\},\ p_{11}(1) = 0,\ p_{12}(1) = 1,\ p_{21}(1) = 1,\ p_{22}(1) = 0,\ r_1(1) = 2,\ r_2(1) = 0$, and let $v^0 = (0, 0)$. Then, $v^{2n} = (2n, 2n)$ and $v^{2n+1} = (2n + 2, 2n),\ n = 0, 1, \dots$. Hence, no convergence for the sequence $\{v^n\}_{n=0}^\infty$ nor for $\{v^{n+1} - v^n\}_{n=0}^\infty$.

Remark
We will show that $\phi = lim_{n\to\infty} \frac{1}{n} v^n$. However, this is - in general - numerically an instable computation scheme if v^n tends to infinity. Fortunately, we may also use the property $\phi = lim_{n\to\infty} \frac{1}{n} \sum_{k=1}^n \{v^{k+1} - v^k\}$. These properties can be shown by using the sequence $\{e^n\}_{n=0}^\infty$, where

$$e^n := v^n - n \cdot \phi - u, \tag{5.46}$$

with u defined as in Theorem 5.17, i.e. $u = u^0(f_0) - M \cdot \phi$ for some M and with f_0^∞ a Blackwell optimal policy. In case the Markov chains $P(f)$ are aperiodic for all $f^\infty \in C(D)$, which we may assume without loss of the generality (see below in Lemma 5.10), we can show that $\phi = lim_{n\to\infty}\{v^{n+1} - v^n\}$.
Let f_0^∞ be a Blackwell optimal policy. From the proof of Theorem 5.17 it follows that (ϕ, u) satisfies for any $f^\infty \in C(D)$

$$\begin{cases} \phi & \geq \; P(f)\phi \\ \phi + u & \geq \; r(f) + P(f)u \end{cases} \tag{5.47}$$

Furthermore, we define $F_0 := \{f \mid \phi = P(f)\phi;\ \phi + u = r(f) + P(f)u\}$. Notice that $f_0 \in F_0$ and that every $f \in F_0$ is an average optimal policy, since $\phi(f^\infty) = P^*(f)r(f) = P^*(f)\{\phi + u - P(f)u\} = P^*(f)\phi = \phi$.

Lemma 5.8

(1) If $f \in F_0$, then $P(f)e^n \leq e^{n+1} \leq P(f_{n+1})e^n,\ n = 0, 1, \dots$.

(2) $\{e^n\}_{n=0}^\infty$ is bounded.

(3) $\phi = lim_{n\to\infty}\frac{1}{n}v^n$.

(4) $\phi = lim_{n\to\infty}\frac{1}{n}\sum_{k=1}^n\{v^k - v^{k-1}\}$.

Proof

(1) Let $n \in \mathbb{N}_0$ and $f \in F_0$ be arbitrarily chosen. Then,

$$\begin{aligned} P(f)e^n &= P(f)\{v^n - n\cdot\phi - u\} = P(f)v^n - n\cdot P(f)\phi - P(f)u \\ &= \{P(f)v^n + r(f)\} - n\cdot P(f)\phi - \{P(f)u + r(f)\} \\ &\leq v^{n+1} - (n+1)\cdot\phi - u = e^{n+1}. \\ P(f_{n+1})e^n &= P(f_{n+1})v^n - n\cdot P(f_{n+1})\phi - P(f_{n+1})u \\ &\geq P(f_{n+1})v^n - n\cdot\phi - \{u + \phi - r(f_{n+1})\} = v^{n+1} - (n+1)\cdot\phi - u = e^{n+1}. \end{aligned}$$

(2) From part (1), we obtain

$$\begin{aligned} P^n(f_0)(v^0 - u) &= P^n(f_0)e^0 \leq P^{n-1}(f_0)e^1 \leq \dots \leq P^0(f_0)e^n = e^n \leq P(f_n)e^{n-1} \\ &\leq P(f_n)P(f_{n-1})e^{n-2} \leq \dots \leq P(f_n)P(f_{n-1})\cdots P(f_1)e^0 \\ &= P(f_n)P(f_{n-1})\cdots P(f_1)(v^0 - u), \end{aligned}$$

implying that $min_i\,(v_i^0 - u_i)\cdot e \leq e^n \leq max_i\,(v_i^0 - u_i)$.

(3) Since $\phi = \frac{1}{n}\{v^n - e^n - u\}$ and $\{e^n\}_{n=0}^\infty$ is bounded, we have $\phi = lim_{n\to\infty}\frac{1}{n}v^n$.

(4) From $\frac{1}{n}\sum_{k=1}^n\{v^k - v^{k-1}\} = \frac{1}{n}(v^n - v^0)$ and part (3), we obtain $\phi = lim_{n\to\infty}\frac{1}{n}\sum_{k=1}^n\{v^k - v^{k-1}\}$. □

Lemma 5.9

Let, for all $i \in S$, $A_n(i) := \Big\{a \in A(i) \;\Big|\; max_b\{r_i(b) + \sum_j p_{ij}(b)v_j^{n-1}\} = r_i(a) + \sum_j p_{ij}(a)v_j^{n-1}\Big\}$ and $A_*(i) := \{a \in A(i) \mid \phi_i = \sum_j p_{ij}(a)\phi_j\}$. Then, for n sufficiently large, $A_n(i) \subseteq A_*(i),\ i \in S$.

Proof

Suppose the contrary. Then, there exists a pair $(i, a) \in S \times A$ and a sequence $\{n_k\},\ k = 1, 2, \dots$ such that $a \in A_{n_k}(i),\ k = 1, 2, \dots$ and $a \notin A_*(i)$. Since $\frac{1}{n_k}v_i^{n_k} = \frac{1}{n_k}\{r_i(a) + \sum_j p_{ij}(a)v_i^{n_{k-1}}\}$, and by part (3) of Lemma 5.8, we obtain $\phi_i = \sum_j p_{ij}(a)\phi_j$, i.e. $a \in A_*(i)$: contradiction. □

Next, we show that we may assume that for every $f^\infty \in C(D)$ the Markov chain $P(f)$ is aperiodic. In that case we have $P^*(f) = lim_{n\to\infty}P^n(f)$.[5] Consider for an arbitrary $\lambda \in (0, 1)$ the transition probabilities

$$p_{ij}(a)(\lambda) := \lambda\delta_{ij} + (1 - \lambda)p_{ij}(a),\ (i, a) \in S \times A,\ j \in S. \tag{5.48}$$

[5] See e.g. H.M. Taylor and S. Karlin: *An introduction to stochastic modeling*, 3rd edition, 1998, chapter 4.

Since $p_{ii}(a)(\lambda) \geq \lambda > 0,\ i \in S$, the transition matrix is aperiodic. Let $\phi_\lambda(f^\infty)$ be the average reward of policy f^∞ with respect to the transitions $p_{ij}(a)(\lambda)$. The following lemma shows that $\phi_\lambda(f^\infty) = \phi(f^\infty)$ for all $f^\infty \in C(D)$.

Lemma 5.10
$\phi_\lambda(f^\infty) = \phi(f^\infty)$ for all $f^\infty \in C(D)$.

Proof
$P_\lambda(f)\phi(f^\infty) = \{\lambda I + (1-\lambda)P(f)\}\phi(f^\infty) = \lambda\phi(f^\infty) + (1-\lambda)\phi(f^\infty) = \phi(f^\infty)$, and consequently, $P^*_\lambda(f)\phi(f^\infty) = \phi(f^\infty)$. We also have

$$\begin{aligned} r(f) + P_\lambda(f)D(f)r(f) - D(f)r(f) &= r(f) + \{\lambda I + (1-\lambda)P(f)\}D(f)r(f) - D(f)r(f) \\ &= r(f) + (\lambda - 1)\{I - P(f)\}D(f)r(f) \\ &= r(f) + (\lambda - 1)\{I - P^*(f)\}r(f) \\ &= \lambda r(f) + (1-\lambda)\phi(f^\infty). \end{aligned}$$

Hence, $(1-\lambda)r(f) + \{P_\lambda(f) - I\}D(f)r(f) = (1-\lambda)\phi(f^\infty)$. Multiplying this equality by $P^*_\lambda(f)$ gives $(1-\lambda)P^*_\lambda(f)r(f) = (1-\lambda)P^*_\lambda(f)\phi(f^\infty)\} = (1-\lambda)\phi(f^\infty)$, i.e. $\phi_\lambda(f^\infty) = \phi_(f^\infty)$. □

Theorem 5.23
*Let $m_i := \liminf_{n\to\infty} e^n_i,\ M_i := \limsup_{n\to\infty} e^n_i$, and $A_*(i) := \{a \in A(i) \mid \phi_i = \sum_j p_{ij}(a)\phi_j\}, i \in S$. Furthermore, let $s_i(a) := r_i(a) - \phi_i + \sum_j p_{ij}(a)u_j - u_i$ for all $(i,a) \in S \times A$.
Then, $max_{a\in A_*(i)}\big\{s_i(a) + \sum_j p_{ij}(a)m_j\big\} \leq m_i \leq M_i \leq max_{a\in A_*(i)}\big\{s_i(a) + \sum_j p_{ij}(a)M_j\big\}$ for all $i \in S$.*

Proof
For n sufficiently large, we obtain by Lemma 5.9

$$\begin{aligned} max_{a\in A_*(i)}\big\{s_i(a) + \textstyle\sum_j p_{ij}(a)e^n_j\big\} &= max_{a\in A_*(i)}\big\{r_i(a) - \phi_i + \textstyle\sum_j p_{ij}(a)u_j - u_i + \sum_j p_{ij}(a)e^n_j\big\} \\ &= max_{a\in A_*(i)}\big\{r_i(a) - \phi_i + \textstyle\sum_j p_{ij}(a)(u_j + e^n_j) - u_i\big\} \\ &= max_{a\in A_*(i)}\big\{r_i(a) - \phi_i + \textstyle\sum_j p_{ij}(a)(v^n_j - n\cdot\phi_j) - u_i\big\} \\ &= max_{a\in A_*(i)}\big\{r_i(a) - (n+1)\phi_i + \textstyle\sum_j p_{ij}(a)v^n_j - u_i\big\} \\ &= max_{a\in A_*(i)}\big\{r_i(a) + \textstyle\sum_j p_{ij}(a)v^n_j\big\} - (n+1)\phi_i - u_i \\ &= v^{n+1}_i - (n+1)\phi_i - u_i = e^{n+1}_i. \end{aligned}$$

Hence,

$$\begin{aligned} m_i &= \liminf_{n\to\infty} e^{n+1}_i = \liminf_{n\to\infty} max_{a\in A_*(i)}\big\{s_i(a) + \textstyle\sum_j p_{ij}(a)e^n_j\big\} \\ &\geq max_{a\in A_*(i)}\big\{s_i(a) + \textstyle\sum_j p_{ij}(a)\big(\liminf_{n\to\infty} e^n_j\big)\big\} = max_{a\in A_*(i)}\big\{s_i(a) + \sum_j p_{ij}(a)m_j\big\} \end{aligned}$$

and

$$\begin{aligned} M_i &= \limsup_{n\to\infty} e^{n+1}_i = \limsup_{n\to\infty} max_{a\in A_*(i)}\big\{s_i(a) + \textstyle\sum_j p_{ij}(a)e^n_j\big\} \\ &\leq max_{a\in A_*(i)}\big\{s_i(a) + \textstyle\sum_j p_{ij}(a)\big(\limsup_{n\to\infty} e^n_j\big)\big\} = max_{a\in A_*(i)}\big\{s_i(a) + \sum_j p_{ij}(a)M_j\big\}. \end{aligned}$$

□

Theorem 5.24
Under the aperiodicity assumption, the sequence $\{e^n\}_{n=0}^\infty$ is convergent.

Proof
Suppose that $m_j = \lim_{k\to\infty} e^{p_k}_j$ and $M_j = \lim_{k\to\infty} e^{q_k}_j,\ j \in S$, for some subsequences $\{p_k\}$ and $\{q_k\}$ of $\{0, 1, 2, \dots\}$. Choose for every $k \in \mathbb{N}$ an integer $h(k)$ such that $r_k := q_{h(k)} - p_k \geq k$. From Lemma 5.8 part (1), we obtain $e^{q_{h(k)}} = e^{r_k + p_k} \geq \{P(f)\}^{r_k} \cdot e^{p_k}$ for any $f \in F_0$ and $k \in \mathbb{N}$. Hence, for any $f \in F_0$,

$$\begin{aligned} M &= \lim_{k\to\infty} e^{q_k} = \lim_{k\to\infty} e^{q_{h(k)}} \geq \ lim_{k\to\infty} \{P(f)\}^{r_k} \cdot e^{p_k} \\ &= \lim_{k\to\infty} \{P(f)\}^k \cdot e^{p_k} = \big\{ \lim_{k\to\infty} \{P(f)\}^k \big\} \cdot \big\{ \lim_{k\to\infty} e^{p_k} \big\} = P^*(f)m. \end{aligned}$$

Similarly, we obtain $m \geq P^*(f)M$.

Since $m \geq P^*(f)M \geq P^*(f)P^*(f)m = P^*(f)m$ and $P^*(f)\{m - P^*(f)m\} = 0$, we have $m_j = \{P^*(f)m\}_j$, and similarly $M_j = \{P^*(f)M\}_j$, for every state recurrent under $P(f)$. Therefore, $m_j \geq \{P^*(f)M\}_j = M_j$ for every recurrent state, i.e. $m_j = M_j$ for every state which is recurrent under $P(f)$ for some $f \in F_0$.

Let f_* satisfy $f_*(i) \in A_*(i)$ and $\{s(f_*) + P(f_*)M\}_i = max_{a\in A_*(i)} \{s_i(a) + \sum_j p_{ij}(a)M_j\},\ i \in S$. By Theorem 5.23, $s(f_*) + P(f_*)M \geq M$, implying $P^*(f_*)s(f_*) \geq 0$. Since (ϕ, u) is superharmonic, $s(f_*) \leq 0$ and therefore $P^*(f_*)s(f_*) \leq 0$. Consequently, $P^*(f_*)s(f_*) = 0$. Hence, $s_j(f_*) = 0$ for j recurrent under f_*. Take policy f^∞ equal to f_*^∞ in the states which are recurrent under $P(f_*)$, and equal to a policy f_0^∞, where $f_0 \in F_0$, in the transient states of $P(f_*)$. Then, $f^\infty \in F_0$ and the states which are recurrent under $P(f_*)$ are a subset of the states which are recurrent under $P(f)$. Hence, $m_j = M_j$ for the states j recurrent under $P(f_*)$. By Theorem 5.23, we obtain $s(f_*) + P(f_*)m \leq m \leq M \leq s(f_*) + P(f_*)M$, i.e. $P(f_*)(M - m) \geq M - m \geq 0$, and consequently,

$$P^*(f_*)(M - m) \geq M - m \geq 0. \tag{5.49}$$

Since $m_j = M_j$ for the states which are recurrent under $P(f_*)$, we have $P^*(f_*)(M - m) = 0$, and by (5.49), $M = m$. □

Corollary 5.7
$\phi = lim_{n\to\infty} (v^{n+1} - v^n)$.

Proof
$\phi = (v^{n+1} - v^n) - (e^{n+1} - e^n)$. Since the sequence $\{e^n\}_{n=0}^\infty$ converges, $lim_{n\to\infty} (e^{n+1} - e^n) = 0$, and consequently, $\phi = lim_{n\to\infty} (v^{n+1} - v^n)$. □

We will close this section by an algorithm to compute an ε-optimal policy under the following assumption.

Assumption 5.1
Every Markov chain $P(f)$ is aperiodic and the value vector is constant, i.e. $\phi = \phi_0 \cdot e$.

Theorem 5.25
Let $l_n = min_i\, (v_i^n - v_i^{n-1})$ and $u_n = max_i\, (v_i^n - v_i^{n-1})$. Then,

(1) $l_n \uparrow \phi_0$ and $u_n \downarrow \phi_0$.

(2) $l_n \cdot e \leq \phi(f_n^\infty) \leq \phi_0 \cdot e \leq u_n \cdot e$ for every $n \geq 1$.

Proof
(1) $$\begin{aligned} v^{n+1} - v^n &\geq \{r(f_n) + P(f_n)v^n\} - \{r(f_n) + P(f_n)v^{n-1}\} = P(f_n)\{v^n - v^{n-1}\} \\ &\geq P(f_n) \cdot min_i\, \{v^n - v^{n-1}\}_i \cdot e = l_n \cdot e, \text{ implying } l_{n+1} \geq l_n. \end{aligned}$$

Similarly, it can be shown that $u_{n+1} \leq u_n$. Hence, by Corollary 5.7, we obtain $l_n \uparrow \phi_0$ and $u_n \downarrow \phi_0$.

(2) For any $n \geq 1$, we have $u_n \geq u_{n+1} \geq \cdots \geq lim_{k\to\infty} u_{n+k} = \phi_0$ and

$$\begin{aligned} \phi(f_n^\infty) &= P^*(f_n)r(f_n) = P^*(f_n)\{v^n - P(f_n)v^{n-1}\} = P^*(f_n)\{v^n - v^{n-1}\} \\ &\geq P^*(f_n) \cdot min_i\, \{v^n - v^{n-1}\}_i \cdot e = min_i\, \{v^n - v^{n-1}\}_i \cdot e = l_n \cdot e. \end{aligned}$$ □

From the above theorem we can derive an algorithm. However, since $\phi = lim_{n\to\infty} \frac{1}{n} v^n$ (see Lemma 5.8 part (3)), v^n grows linearly in n, which may cause numerical difficulties. To overcome these difficulties, we use the following transformation. Let $w_i^n := v_i^n - v_N^n,\ i \in S,\ n \geq 0$ and $g^n := v_N^n - v_N^{n-1},\ n \geq 1$. Then, we have $w_i^n = \{e_i^n + n\phi_0 + u_i\} - \{e_N^n + n\phi_0 + u_N\} = \{e_i^n - e_N^n\} + \{u_i - u_N\}$, which is a bounded sequence, and $g^n = \{e_N^n + n\phi_0 + u_N\} - \{e_N^{n-1} + (n-1)\phi_0 + u_N\} = \{e_N^n - e_N^{n-1}\} + \phi_0$, which is also a bounded sequence. Furthermore, the recurrence relations become

$$\begin{aligned} g^{n+1} &= v_N^{n+1} - v_N^n = max_{a \in A(N)}\, \{r_N(a) + \textstyle\sum_j p_{Nj}(a)(v_j^n - v_N^n)\} \\ &= max_{a \in A(N)}\, \{r_N(a) + \textstyle\sum_j p_{Nj}(a) w_j^n\}, \end{aligned}$$

and

$$\begin{aligned} w_i^{n+1} &= v_i^{n+1} - v_N^{n+1} = max_{a \in A(i)}\, \{r_i(a) + \textstyle\sum_j p_{ij}(a)(v_j^n - v_N^n)\} + (v_N^n - v_N^{n+1}) \\ &= max_{a \in A(i)}\, \{r_i(a) + \textstyle\sum_j p_{ij}(a) w_j^n\} - g^{n+1},\ i \in S. \end{aligned}$$

For the bounds l_n and u_n, we obtain

$$l_n = min_i\,(v_i^n - v_i^{n-1}) = min_i\,\{(w_i^n + v_N^n) - (w_i^{n-1} + v_N^{n-1})\} = min_i\,(w_i^n - w_i^{n-1}) + g^n$$

and

$$u_n = max_i\,(v_i^n - v_i^{n-1}) = max_i\,\{(w_i^n + v_N^n) - (w_i^{n-1} + v_N^{n-1})\} = max_i\,(w_i^n - w_i^{n-1}) + g^n.$$

In step 2 of algorithm 5.10 (see below) we use v for w^n, w for w^{n+1}, g for g^{n+1}, u for $u_{n+1} - g^{n+1}$ and l for $l_{n+1} - g^{n+1}$.

Algorithm 5.10 *Value iteration (aperiodicity and constant value vector case)*
Input: Instance of an MDP and some scalar $\varepsilon > 0$.
Output: An ε-optimal deterministic policy f^∞ and a $\frac{1}{2}\varepsilon$-approximation of the value ϕ_0.

```
1. Select $v \in \mathbb{R}^N$ arbitrarily; $v_N := 0$.
2. a. $y_i(a) := r_i(a) + \sum p_{ij}(a) v_j$ for all $(i,a) \in S \times A$.
   b. $g := max_{a \in A(N)}\, y_N(a)$.
   c. $w_i := max_{a \in A(i)}\, y_i(a) - g$ for all $i \in S$.
   d. Take $f$ such that $w = r(f) + P(f)v - g \cdot e$.
   e. $u := max_i\,(w_i - v_i)$; $l := min_i\,(w_i - v_i)$.
3. if $u - l \leq \varepsilon$ then
         begin $f^\infty$ is an $\varepsilon$-optimal policy; $\frac{1}{2}(u+l) + g$ is an $\frac{1}{2}\varepsilon$-approximation of $\phi_0$ (STOP) end
   else begin $v := w$; return to step 2 end.
```

Example 5.14
Consider the MDP of Example 3.1. The value vector is constant ($\phi_0 = 7$). Although the requirement of aperiodicity is not fulfilled, the algorithm works, as one can see below. Select $\varepsilon = 0.1$ and $v^0 = (0,0,0)$.
Iteration 1:
$y_1(1) = 1,\ y_1(2) = 2,\ y_1(3) = 3;\ y_2(1) = 6,\ y_2(2) = 4,\ y_2(3) = 5;\ y_3(1) = 8,\ y_3(2) = 9,\ y_3(3) = 7.$
$g = 9;\ w = (-6,-3,0);\ f(1) = 3,\ f(2) = 1,\ f(3) = 2;\ u = 0,\ l = -6;\ v = (-6,-3,0).$
Iteration 2:
$y_1(1) = -5,\ y_1(2) = -1,\ y_1(3) = 0;\ y_2(1) = 0,\ y_2(2) = 1,\ y_2(3) = 5;\ y_3(1) = 2,\ y_3(2) = 6,\ y_3(3) = 7.$
$g = 7;\ w = (-7,-2,0);\ f(1) = 3,\ f(2) = 3,\ f(3) = 3;\ u = 0,\ l = -1;\ v = (-7,-2,0).$
Iteration 3:
$y_1(1) = -6,\ y_1(2) = 0,\ y_1(3) = 3;\ y_2(1) = -1,\ y_2(2) = 2,\ y_2(3) = 5;\ y_3(1) = 1,\ y_3(2) = 7,\ y_3(3) = 7.$
$g = 7;\ w = (-4,-2,0);\ f(1) = 3,\ f(2) = 3,\ f(3) = 3;\ u = 0,\ l = 0$:
f^∞ with $f(1) = 3,\ f(2) = 3,\ f(3) = 3$ is an optimal policy and $\phi_0 = 7$.

Relaxation and one-step look-ahead

The standard value iteration algorithm is given by

$$v_i^{n+1} := max_a \{r_i(a) + \sum_j p_{ij}(a)v_j^n\},\ i \in S,\ n = 0, 1, \dots \tag{5.50}$$

In this subsection we make the following assumptions.

Assumption 5.2
(1) Every Markov chain $P(f)$ is aperiodic.
(2) The one-step rewards $r_i(a),\ (i,a) \in S \times A$, are strictly positive.
(3) The value vector ϕ is constant and this constant is denoted by ϕ_0.

The assumptions (1) and (2) are without loss of generality: for (1) see Lemma 5.10 and for (2) note that adding a constant to every $r_i(a)$ adds the same constant to the average rewards. Consider the differences $\delta_i^{n+1} := v_i^{n+1} - v_i^n,\ i \in S$. Then, we have $\phi_0 = \lim_{n\to\infty} \delta_i^n$ for every $i \in S$ (see Corollary 5.7). Furthermore, we have by Theorem 5.25 part (1), $min_i\, \delta_i^n \uparrow \phi_0$ and $max_i\, \delta_i^n \downarrow \phi_0$. The algorithm stops at a certain iteration n when the values $max_i\, \delta_i^n$ and $min_i\, \delta_i^n$ are close enough to each other. Note that if $v^0 := 0$, then $v_i^1 = max_a\, r_i(a) > 0,\ i \in S$. Furthermore, $v^n \uparrow$ (easily demonstrated by induction), so $min_i\, \delta_i^n > 0$ for all $n \in \mathbb{N}$. We shall use the relative accuracy criterion

$$\frac{max_i\, \delta_i^n}{min_i\, \delta_i^n} \le 1 + \varepsilon \tag{5.51}$$

for some $\varepsilon > 0$. By Theorem 5.25 part (2), this stopping criterion implies that, at termination in iteration n, we have $\|\phi_0 - \phi(f_n^\infty)\|_\infty \le u_n - l_n \le \varepsilon \cdot l_n$.

The idea in relaxation is to replace the iterand v^{n+1} by $\hat{v}^{n+1}$, formed as a linear combination of v^{n+1} and v^n:

$$\hat{v}_i^{n+1} := \omega \cdot v^{n+1} + (1-\omega) \cdot v^n = v^n + \omega \cdot (v^{n+1} - v^n) = v^n + \omega \cdot \delta^{n+1}, \tag{5.52}$$

where ω is called an *adaptive relaxation factor*; adaptive means that the relaxation factor may depend on the iteration index. For $\omega = 1$, we obtain the standard value iteration algorithm, i.e. $\hat{v}^{n+1} = v^{n+1}$ for all $n = 0, 1, \dots$.

In order to explore the effect of replacing the original v^{n+1} with the modified $\hat{v}^{n+1}$, we perform a one-step ahead analysis. Therefore, we consider an *estimation* of v^{n+2}. This estimator, denoted by w^{n+1}, will replace v^{n+1} in the iteration scheme. Such an estimator has the prospect to be closer to v^{n+2} than v^{n+1}, and in this way to improve the speed of the convergence of the algorithm. Hence, given the approximation v^n obtained in iteration n, the next iteration in which the approximation v^{n+1} is computed consists of three steps:
(1) a 'first' v^{n+1} is computed by (5.50) and let f_n such that $v^{n+1} = r(f_n) + P(f_n)v^n$;
(2) the relaxation $\hat{v}_i^{n+1} := v^n + \omega \cdot \delta^{n+1}$ is computed for some relaxation factor ω;
(3) w^{n+2} is computed using the policy f_n^∞ and the relaxation $\hat{v}_i^{n+1}$ and set $v^{n+1} := w^{n+2}$.

The vector w^{n+2} is defined by

$$w^{n+2} := r(f_n) + P(f_n)\hat{v}^{n+1} = r(f_n) + P(f_n)\{v^n + \omega \cdot \delta^{n+1}\} = v^{n+1} + \omega \cdot g^{n+1}, \tag{5.53}$$

where $g^{n+1} := P(f_n)\delta^{n+1}$.

Note that g_i^{n+1} will represent the value of δ_i^{n+2} for the standard algorithm ($\omega = 1$), if there is no change in the decision for state i at the $(n+2)$th iteration, i.e. $p_{ij}(f_n) = p_{ij}(f_{n+1}),\ j \in S$ and $r_i(f_n) = r_i(f_{n+1})$, namely

$$\begin{aligned} g_i^{n+1} &= \textstyle\sum_j p_{ij}(f_n)\{v_j^{n+1} - v_j^n\} = \sum_j p_{ij}(f_{n+1})v_j^{n+1} - \sum_j p_{ij}(f_n)v_j^n \\ &= \{v_i^{n+2} - r_i(f_{n+1})\} - \{v_i^{n+1} - r_i(f_n)\} = v_i^{n+2} - v_i^{n+1} = \delta_i^{n+2}. \end{aligned}$$

Next, we will argue what are good values for the relaxation factor ω. One way for finding a good ω is to choose ω such that the values $max_i\ \delta_i^{n+2}$ and $min_i\ \delta_i^{n+2}$ are close to each other. However, δ^{n+2} is unknown, since v^{n+2} is unknown. But we can use an estimator of δ^{n+2}, namely:

$$\hat\delta^{n+2} := w^{n+2} - \hat v^{n+1} = \{v^{n+1} + \omega \cdot g^{n+1}\} - \{v^n + \omega \cdot \delta^{n+1}\} = \delta^{n+1} + \omega \cdot (g^{n+1} - \delta^{n+1}).$$

Now, we solve - as function of ω - the equation

$$\hat\delta_{max}^{n+2} = \hat\delta_{min}^{n+2}, \tag{5.54}$$

where max and min are the states such that $\hat\delta_{max}^{n+2} = max_i\ \hat\delta_i^{n+2}$ and $\hat\delta_{min}^{n+2} = min_i\ \hat\delta_i^{n+2}$. The solution of (5.53) yields the value ω_{n+1}, where

$$\omega_{n+1} := \frac{\delta_{max}^{n+1} - \delta_{min}^{n+1}}{\{\delta_{max}^{n+1} - \delta_{min}^{n+1}\} - \{g_{max}^{n+1} - g_{min}^{n+1}\}} \tag{5.55}$$

Below we present an algorithm that applies the above elements.

Algorithm 5.11 *Value iteration with adaptive relaxation and one-step look-ahead*
Input: Instance of an MDP and some scalar $\varepsilon > 0$.
Output: A nearly optimal deterministic policy f^∞ and an approximation of the value vector ϕ_0.

```
1. $v := 0$.
2. for all $i \in S$ do begin $y_i := max_a\{r_i(a) + \sum p_{ij}(a)v_j\}$; $\delta_i := y_i - v_i$ end
3. for all $i \in S$ do determine $f(i)$ such that $y_i = r_i\big(f(i)\big) + \sum_j p_{ij}\big(f(i)\big)v_j$.
4. for all $i \in S$ do $g_i := \sum_j p_{ij}\big(f(i)\big)\delta_j$.
5. Determine $max$ such that $\delta_{max} = max_i\,\delta_i$; determine $min$ such that $\delta_{min} = min_i\,\delta_i$.
6. $\omega := \frac{\delta_{max}-\delta_{min}}{\{\delta_{max}-\delta_{min}\}-\{g_{max}-g_{min}\}}$.
7. for all $i \in S$ do $w_i := y_i + \omega \cdot g_i$.
8. $u := max_i\,(w_i - v_i)$; $l := min_i\,(w_i - v_i)$.
9. if $\frac{u}{l} \le 1 + \varepsilon$ then
       begin $f^\infty$ is an approximation of the optimal policy; $\frac{1}{2}(u+l)$ is an approximation of $\phi_0$ (STOP)
       end
   else begin $v := w$; return to step 2 end.
```

However, finding the adaptive relaxation factor ω by considering only the states max and min neglects to take account the influence of all other states and this might not be effective in certain iterations. Other approaches, which we shall discuss below, may also seem quite worthwhile, in particular for cases where the number of states is high, which is characteristic for MDPs. Such other approach is the *minimum ration criterion* which is described below.

Inequality (5.51) defines the stopping criterion. If this condition has not been satisfied, it seems plausible for the next iteration to find an ω that will minimize or at least reduce the term $D(\omega) := \frac{M(\omega)}{m(\omega)}$, where $M(\omega) := max_i\, \hat{\delta}_i^{n+2} = max_i\, \{\delta_i^{n+1} + \omega \cdot h_i^{n+1}\}$ and $m(\omega) := min_i\, \hat{\delta}_i^{n+2} = min_i\, \{\delta_i^{n+1} + \omega \cdot h_i^{n+1}\}$ with $h^{n+1} := g^{n+1} - \delta^{n+1}$. This criterion is meaningful if we take ω such that $m(\omega) > 0$.

Since $M(\omega)$ is the maximum of a set of linear functions, $M(\omega)$ is a piecewise linear convex function (the slopes of the line segments are nondecreasing in ω); similarly, $m(\omega)$ is a piecewise linear concave function (the slopes of the line segments are nonincreasing in ω). Moreover, $D(\omega)$ is piecewise *affine linear*, i.e. $D(\omega) = \frac{A+B\omega}{C+D\omega}$ and $D(\omega)$ is *monotone* on each segment, because on each segment $D'(\omega) = \frac{BC-AD}{(C+D\omega)^2}$. Therefore, one is only interested in the endpoints of any segment, which are the breakpoints for $M(\omega)$ and $m(\omega)$.

Denote the breakpoints of $M(\omega)$ as $0 = x_0 < x_1 < x_2 < \cdots < x_K$ and let $A_k + B_k\omega$ be the line segment over the range $x_{k-1} \leq \omega \leq x_k$ $(k = 1, 2, \dots, K)$. From the properties of $M(\omega)$ it follows that $\{A_k\}$ is a decreasing sequence, while the sequence $\{B_k\}$ is increasing. Similarly, let $0 = y_0 < y_1 < y_2 < \cdots < y_L$ denote the breakpoints of $m(\omega)$, while $C_l + D_l\omega$ is the line segment over the range $y_{l-1} \leq \omega \leq y_l$ $(l = 1, 2, \dots, L)$. Then, $\{C_l\}$ is an increasing sequence and $\{D_l\}$ an decreasing sequence.

Let $Z := \{x_k,\ 0 \leq k \leq K\} \cup \{y_l,\ 0 \leq l \leq L\} = \{z_j,\ 0 \leq j \leq J\}$, where the $\{z_j,\ 0 \leq j \leq J\}$ is such arranged that $0 = z_0 < z_1 < z_2 < \cdots < z_J$. Then, the minimum of $D(\omega)$ must occur at one of the breakpoints $\{z_0, z_1, \dots, z_J\}$. Hence, a reasonable heuristic is a one-pass scan along the z_j s. In each z_j we compute $D(z_j) = \frac{M(z_j)}{m(z_j)}$ and we stop with $\omega^* = z_j$ when we for first time discover that $D(z_j) \leq D(z_{j+1})$.

Another approach might be to try to retrieve *quickly* a 'good' starting point for the optimal ω and then continue the search from this starting point for the optimal ω^*. For this purpose we define the value ω_1^* such that $M(\omega_1^*) = min_\omega\, M(\omega) = min_\omega\, \{max_i\, (\delta_i^{n+1} + \omega h_i^{n+1})\}$; similarly let the value ω_2^* be such that $m(\omega_2^*) = max_\omega\, m(\omega) = max_\omega\, \{min_i\, (\delta_i^{n+1} + \omega h_i^{n+1})\}$.

A good starting point for the search procedure could be either ω_1^* or ω_2^*. However, there is a danger that by devoting to much effort to finding an optimal ω^* in each iteration, we may lose on overall efficiency. It was found empirically (reported in [115]) that the selection of ω^* according to the simple rule

$$\omega^* = \begin{cases} \omega_1^* & \text{if } D(\omega_1^*) \leq D(\omega_2^*) \\ \omega_2^* & \text{otherwise} \end{cases} \tag{5.56}$$

usually yields a near optimal relaxation factor. Hence, it is desirable to develop an efficient procedure for finding ω_1^* and ω_2^*. We can use a linear program to solve the *minmax* problem for finding ω_1^*. Notice that $M(\omega) = max_i\, \{\delta_i^{n+1} + \omega \cdot h_i^{n+1}\}$, which is the same as the minimum value of ω_0 when ω_0 is restricted to the value that are at least $\delta_i^{n+1} + \omega \cdot h_i^{n+1}$ for all $i \in S$. Hence, if (ω_0^*, ω_1^*) is the optimal value of the linear program

$$min \left\{ \omega_0 \;\middle|\; \begin{array}{lcl} \omega_0 & \geq & \delta_i^{n+1} + \omega_1 \cdot h_i^{n+1},\ i \in S \\ \omega_1 & \geq & 0 \end{array} \right\}. \tag{5.57}$$

The corresponding dual linear program is

$$max \left\{ \sum_{i \in S} \delta_i^{n+1} u_i \;\middle|\; \begin{array}{lcl} \sum_{i \in S} u_i & = & 1 \\ \sum_{i \in S} h_i^{n+1} u_i & \geq & 0 \\ u_i & \geq & 0,\ i \in S \end{array} \right\}. \tag{5.58}$$

Since (5.57) has a finite optimal solution, also program (5.58) has a finite optimal solution. Let (ω_0^*, ω_1^*) and u^* be the optimal basic solutions of the linear programs (5.58) and (5.59), respectively. From the complementary slackness property of linear programming we obtain

$$\omega_1^* \cdot \Big\{ \sum_{i \in S} h_i^{n+1} u_i \Big\} = 0,\ i \in S. \tag{5.59}$$

As there are only two constraints for the dual, a basic feasible solution will have at most two positive u_i^*s, say u_p^* and u_q*, and by the first constraint of (5.58), we have $u_p^* + u_q^* = 1$. If $\omega_1^* = 0$, then then $max_i \, \{\delta_i^{n+1} + \omega \cdot h_i^{n+1}\} \geq max_i \, \delta_i^{n+1}$ for all $\omega \geq 0$, i.e. $h_i^{n+1} \geq 0$ for all $i \in S$. This is an exceptional case, therefore we assume that $\omega_1^* > 0$. By property (5.59), we have $h_p^{n+1} u_p^* + h_q^{n+1} u_q^* = 0$. By the nonnegativity of u^*, we obtain for the slopes h_p^{n+1} and h_q^{n+1} that $h_p^{n+1} \leq 0$ and $h_q^{n+1} > 0$. The intuitive explanation for the opposite sings of h_p^{n+1} and h_q^{n+1} is that the optimum ω_1^* occurs where a downward slopping-line and an upward slopping-line cross.

Since $M(\omega)$ is piecewise convex, ω_1^* is the breakpoint where a change in sign of the h^{n+1}s occurs. We use this property in developing a greedy procedure for finding the value of ω_1^*. This procedure has the advantage that it skips some of the x_k values. In this procedure we omit the iteration index, i.e. we denote δ and h instead of δ^{n+1} and h^{n+1}, respectively.

Algorithm 5.12 *Procedure for the computation of* ω_1^*
Input: The vectors δ and h.
Output: The value ω_1^*

```
1. Set $\omega_1^* := 0$
2. Let $M$ and $k$ be such that $M = max_i \, \delta_i = \delta_k$ (if $k$ is not unique, select under the candidates that
   state with the highest $h$-value); set $H := h_k$.
3. Find $\omega_1$ such that $\omega_1 = min_{\{i \,|\, h_i > 0\}} \{ \frac{M - \delta_i}{h_i - H} \}$.
4. Let $r$ be such that $m = max_{\{i \,|\, h_i \leq 0\}} \{\delta_i + \omega_1 \cdot h_i\} = \delta_r + \omega_1 \cdot h_r$.
5. $\omega_1^* := \omega_1^* + \omega_1$.
6. if $m = M + \omega_1 \cdot H$ then STOP
   else begin $M := m;\ H := h_r;\ \delta_i := \delta_i + \omega_1 \cdot h_i,\ i \in S$; return to step 3 end
```

The formulation of the *maxmin* problem for finding the value ω_2^* is as follows, where the variable ω_0 represents in this case $min_i \, \hat{\delta}^{n+1}$:

$$max \left\{ \omega_0 \;\middle|\; \begin{array}{lcl} \omega_0 & \leq & \delta_i^{n+1} + \omega_2 \cdot h_i^{n+1}, \ i \in S \\ \omega_2 & \geq & 0 \end{array} \right\}. \tag{5.60}$$

which is equivalent to

$$min \left\{ -\omega_0 \;\middle|\; \begin{array}{lcl} -\omega_0 & \geq & -\delta_i^{n+1} - \omega_2 \cdot h_i^{n+1}, \ i \in S \\ \omega_2 & \geq & 0 \end{array} \right\}. \tag{5.61}$$

Thus, by substituting the negative values for δ_i^{n+1} and h_i^{n+1}, $i \in S$, the algorithm of finding the relaxation value ω_2^* is identical to that of finding ω_1^*.

<u>Remark</u>

Herzberg and Yechiali [115] have tested this relaxation with one-step look-ahead analysis. They have compared their approach with the standard value iteration algorithm. As expected, it appears that their method slightly increases the work per iteration, but significantly decreases the total number of iterations. The advantage of their approach rises with the increase of the dimensions (states and actions) of the problem. The reduction of the computation time is about a factor 2 or 3.

5.10 Bibliographic notes

The concept 'communicating' was introduced by Bather [12]. Platzman [217] introduced the notion 'weakly communicating' under the name *simply connected*. In Kallenberg [153] the classification of MDPs is discussed and the question whether checking the unichain condition can be done in polynomial time was raised in that paper. Tsitsiklis [293] has solved this problem by proving Theorem 5.1. From the paper McCuaig [196] it follows that for *deterministic MDPs* (each transition probability in $\{0, 1\}$) this problem is solvable in polynomial time. Feinberg and Yang have shown ([87]) that other special cases (the so-called *recurrent* and *absorbing* cases) are also polynomially solvable.

Cesaro published in 1890 his idea concerning the convergence of averages ([37]). Theorem 5.3 can be found in many textbooks on Markov chains, e.g. Kemeny and Snell [167]. The proof of this theorem and also the theorems 5.5 and 5.6 follows Veinott [312]. Theorem 5.7 is due to Blackwell [29].

Blackwell [29] provided a theoretical framework for analyzing multichain models. His observation that the average reward model may be viewed as a limit of expected discounted reard models, in which the discount rate approaches 1, stimulated extensive research on average reward models. He introduced the concept of the so-called *1-optimality*, which later was renamed to Blackwell optimality. He also showed that the partial Laurent series expansion, given in Corollary 5.3, provided a link between these two models. The complete Laurent expansion, as presented in Theorem 5.10, is due to Miller and Veinott [199].

The average reward optimality equation (5.11) appears implicitly in Blackwell [29]; an explicit statement appears in Derman's book [69]. This optimality equation is extensively investigated by Schweitzer and Federgruen [258]. Theorem 5.12 is a Tauberian result which can be found in Hordijk [120].

Howard [134] presented the policy iteration algorithm. However, he did not show that the algoritm terminates in finitely many steps. Veinott [308] completed this analysis by establishing that the algorithm cannot cycle. The anticycling rule 1 was proposed by Denardo and Fox [64], rule 2 by Blackwell [29] and rule 3 by Schweitzer and Federgruen [259]. Further, Federgruen and Spreen ([82]) have proposed a modification of Algorithm 5.7 which prevents cycling and avoids parsing the matrices $P(f)$ into their subchains. Spreen ([280]) has presented a choice rule for the relative value vector y which guarantees the convergence and is weaker than the rules 1, 2 and 3. Moreover, the computational complexity is of this variant is smaller.

The linear programming approach for the average reward criterion was independently introduced by De Ghellinck [51] and Manne [193] for the completely ergodic case. The first analysis for the multichain case has been presented in Denardo and Fox [64] who proved Theorem 5.17. Denardo [58] and Derman [69] improved these results slightly. Hordijk and Kallenberg [126] have solved the remaining problems and proved Theorem 5.18. Kallenberg [148] provides a comprehensive analysis of all aspects of linear programming for MDP models. Altman and Spieksma ([5]) have presented a stochastic interpretation of the decision variables that appear in the linear programs.

The value iteration scheme (5.44) was proposed by Bellman [17] and Howard [134]. Lemma 5.8 is due to Brown [34]. The data transformation (5.48) to assure aperiodicity was proposed by Schweitzer [256]. Bounds on the value vector, as given in Theorem 5.25, can be found in Hastings [113]. Denardo [61] proved the convergence of the sequence $\{e^n\}_{n=0}^{\infty}$ under the unichain and aperiodicity assumption. This result was generalized to the multichain case, as shown in Theorem 5.24, by Schweitzer and Federgruen ([257], [260]). The basic idea for Algorithm 5.10, called the *relative value iteration*, is due to White [325], who presented a convergent algorithm. The present implementation with monotone upper and lower bounds u_n and l_n, respectively, was proposed by Odoni ([207]). The relaxation with one-step look-ahead approach is analyzed by Herzberg and Yechiali ([115]) and based on ideas from Popyack, Brown and White ([219]).

5.11 Exercises

Exercise 5.1

Consider the following model:
$S = \{1,2\}$; $A(1) = A(2) = \{1,2\}$; $p_{11}(1) = 1,\ p_{12}(1) = 0$; $p_{11}(2) = 0,\ p_{12}(2) = 1$; $p_{21}(1) = 0,\ p_{22}(1) = 1$; $p_{21}(2) = 1,\ p_{22}(2) = 0$; $r_1(1) = 2,\ r_1(2) = 2$; $r_2(1) = -2,\ r_2(1) = -2$.
In each state one can choose to stay in that state (action 1) or to move to the other state (action 2). Consider the nonstationary policy R which, starting in state 1 at $t = 1$, takes action 2 and moves to state 2 and remains there for until $t = 1 + 3 = 4$ and then returns to state 1, and remains there until $t = 4 + 3^2 = 13$ periods, proceeds to state 2 and remains there until $t = 13 + 3^3 = 40$, etc. Compute for this policy $\phi_1(R) = \liminf_{T\to\infty} \frac{1}{T}\sum_{t=1}^{T} \mathbb{E}_{i,R}\{r_{X_t}(Y_t)\}$ and $\overline{\phi}_1(R) = \limsup_{T\to\infty} \frac{1}{T}\sum_{t=1}^{T} \mathbb{E}_{1,R}\{r_{X_t}(Y_t)\}$.

Exercise 5.2

a. Show that ordinary convergence implies Cesaro convergence.

b. Show, without making use of Theorem 5.2, that ordinary convergence implies Abel convergence.

c. Give a counterexample that Cesaro convergence does not imply ordinary convergence.

d. Give a counterexample that Abel convergence does not imply ordinary convergence.

Exercise 5.3

Give a counterexample that Abel convergence does not imply Cesaro convergence.

Exercise 5.4

Consider the stochastic matrix

$$P = \begin{pmatrix} 0 & 0 & 0 & 0.5 & 0 & 0 & 0 & 0 & 0.5 \\ 0 & 0.25 & 0.25 & 0 & 0.25 & 0 & 0 & 0 & 0.25 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.5 & 0 & 0 & 0 & 0.25 & 0.25 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.5 & 0 & 0 & 0 & 0 & 0.5 \end{pmatrix}.$$

a. Determine the ergodic sets and the transient states; write the matrix in standard form.

b. Determine the stationary matrix P^*.

Exercise 5.5

Consider the stochastic matrix

$$P = \begin{pmatrix} 0.5 & 0.5 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0.25 & 0 & 0 & 0.25 & 0.5 \end{pmatrix}.$$

Determine the stationary matrix, the fundamental matrix and the deviation matrix.

Exercise 5.6

Show that the deviation matrix D satisfies $De = 0$, i.e. the rows sum up to 0.

Exercise 5.7

Let P be an irreducible double stochastic matrix, i.e. an irreducible stochastic matrix with $\sum_{i=1}^n p_{ij} = 1$ for $j = 1, 2, \dots, n$: both the rows and the columns sum up to 1. Determine the stationary matrix P^*.

Exercise 5.8

Consider the following MDP:
$S = \{1,2\};\ A(1) = \{1,2,3\},\ A(2) = \{1\};\ r_1(1) = 1,\ r_1(2) = \frac{3}{4},\ r_1(3) = \frac{1}{2};\ r_2(1) = 0;$
$p_{11}(1) = 0,\ p_{12}(1) = 1;\ p_{11}(2) = \frac{1}{2},\ p_{12}(2) = \frac{1}{2};\ p_{11}(3) = 1,\ p_{12}(3) = 0;,\ p_{21}(1) = 0,\ p_{22}(1) = 1.$
Determine for the deterministic policy f^∞ with $f(1) = 2,\ f(2) = 1$:

a. On which subinterval of $[0, 1)$ is f^∞ an α-discounted optimal policy.
b. $u^k(f)$ for $k = -1, 0, 1, \dots$.
c. $v^T(f^\infty)$ for all $T = 1, 2, \dots$.
d. The Laurent expansion for $v^\alpha(f^\infty)$ and $\alpha_0(f) = \frac{\|D(f)\|}{1+\|D(f)\|}$.

Exercise 5.9

Suppose that the MDP is irreducible. Then the value vector has identical components, say ϕ.
Show the following properties:

(1) $(x, y) = (\phi, u^0(f_0))$ is a solution of the equation
$x + y_i = max_{a\in A(i)}\{r_i(a) + \sum_j p_{ij}(a)y_j\}, i \in S$, where f_0^∞ is a Blackwell optimal policy.

(2) If (x, y) is a solution of the above equation, then $x = \phi$ and $y = u^0(f_0) + c \cdot e$ for some $c \in \mathbb{R}$.

(3) Consider the following MDP, which is obviously not an irreducible model, but multichain.
$S = \{1,2,3\},\ A(1) = A(2) = \{1,2\},\ A(3) = 1.\ r_1(1) = 3,\ r_1(2) = 1,\ r_2(1) = 0,\ r_2(2) = 1,\ r_3(1) = 2.$
$p_{11}(1) = 1, p_{12}(1) = p_{13}(1) = 0;\ p_{11}(2) = 0, p_{12}(2) = 1, p_{13}(2) = 0;\ p_{21}(1) = 0, p_{22}(1) = 1, p_{23}(1) = 0;$
$p_{21}(2) = p_{22}(2) = 0,\ p_{23}(2) = 1;\ p_{31}(1) = p_{32}(1) = 0,\ p_{33}(1) = 1.$
Show that the optimality equation of part (1) doesn't has a solution for this multichain model.

Exercise 5.10

Suppose that the MDP is unichained. Then, for every f^∞, the average reward vector $\phi(f^\infty)$ has identical components, also denoted by $\phi(f^\infty)$.
Show the following properties:

(1) The linear system $\left\{ \begin{array}{rcl} x \cdot e + \{I - P(f)\}y & = & r(f) \\ y_1 & = & 0 \end{array} \right.$ has a unique solution $x = \phi(f^\infty)$ and
$y = u^0(f) - u_1^0(f) \cdot e$.

(2) Show that the set $B(i, f)$, defined in (5.13), in the unichain case can be simplified to
$$B(i, f) = \{a \in A(i) \mid r_i(a) + \textstyle\sum_j p_{ij}(a)y_j > x + y_i\},$$
where x and y are the solution of the system of part (1) of this exercise.

(3) Formulate the policy iteration algorithm for the unichain case.

(4) Consider the following MDP:
$S = \{1,2\};\ A(1) = A(2) = \{1,2\};\ r_1(1) = 4,\ r_1(2) = 2,\ r_2(1) = 3,\ r_2(2) = 1.$
$p_{11}(1) = \frac{1}{3}, p_{12}(1) = \frac{2}{3}; p_{11}(2) = \frac{2}{3}, p_{12}(2) = \frac{1}{3}; p_{21}(1) = \frac{1}{2}, p_{22}(1) = \frac{1}{2}; p_{21}(2) = \frac{1}{2}, p_{22}(2) = \frac{1}{2}.$
Show that the model is a unichain MDP and compute an average optimal policy by the algorithm of part (3), starting with the policy f^∞, where $f(1) = f(2) = 2$.

Exercise 5.11

Show the following properties for an MDP with $\rho := min_{i,j,a}\, p_{ij}(a) > 0$:
(1) $P(f_n)y^n \leq y^{n+1} \leq P(f_{n+1})y^n,\ n \in \mathbb{N}$, where $y^n = v^n - v^{n-1}$.
(2) $span\, y^{n+1} \leq (1 - N\rho) \cdot span\, y^n,\ n \in \mathbb{N}$.
(3) Algorithm 5.10 terminates in at most T iterations with $T = \frac{log\{\frac{\varepsilon}{u_0 - l_0}\}}{log(1-N\rho)}$.

Exercise 5.12

Consider the operators L_f and U defined on $\mathbb{R}^N$ by:

$$L_f\, x = r(f) + P(f)x \text{ and } (Ux)_i = max_{a \in A(i)}\, \{r_i(a) + \textstyle\sum_j\, p_{ij}(a)x_j\},\ i \in S.$$

Show that for any $x \in \mathbb{R}^N$ and $f^\infty \in C(D)$ (without any assumption about the chain structure):
(1) If $L_f\, x \leq y$, then $\phi(f) \leq max_i\, (y - x)_i \cdot e$.
(2) If $L_f\, x \geq y$, then $\phi(f) \geq min_i\, (y - x)_i \cdot e$.
(3) $min_i\, (Ux - x)_i \cdot e \leq \phi(f_x^\infty) \leq \phi \leq max_i\, (Ux - x)_i \cdot e$, where f_x satisfies $Ux = L_{f_x}\, x$.

Chapter 6

Average reward - special cases

6.1 The irreducible case

In this section we impose the following assumption:

Assumption 6.1
The Markov chain $P(f)$ is irreducible for every $f^\infty \in C(D)$.

We have seen in section 5.2.3 that checking the irreducibility property can be done in polynomial time, namely in $\mathcal{O}(M \cdot N^2)$, where $M := \sum_{i \in S} |A(i)|$. In this irreducible case, for every policy f^∞ the stationary matrix has identical and strictly positive rows, and consequently the vector $\phi(f)$ has identical components. Therefore we may $\phi(f)$, and also the value vector ϕ, consider as a scalar. The irreducible case looks like the discounted case. Most results are similar and can be obtained by the property that the stationary matrix has identical rows with strictly positive elements.

6.1.1 Optimality equation

Theorem 6.1

Consider the optimality equation

$$x + y_i = max_{a \in A(i)} \big\{ r_i(a) + \sum_j p_{ij}(a) y_j \big\},\ i \in S. \tag{6.1}$$

Then, we have

(1) $(x, y) = \big(\phi, u^0(f_0)\big)$*, where* ϕ *is the value and* f_0^∞ *a Blackwell optimal policy, is a solution of the optimality equation.*

(2) *If* (x, y) *is a solution of the optimality equation, then* $x = \phi$ *and* $y = u^0(f_0) + c \cdot e$ *for some* $c \in \mathbb{R}$.

Proof

(1) We have seen in Theorem 5.11 that $\big(\phi, u^0(f_0)\big)$ is a solution of (5.11). Since the vector ϕ has identical components, $A(i, \phi) = A(i)$ for all i. Hence, $\big(\phi, u^0(f_0)\big)$ is a solution of (6.1).

(2) It also follows from Theorem 5.11 that if (x, y) is a solution of (5.11), then $x = \phi$. From the optimality equation, we obtain $\phi + y \geq r(f_0) + P(f_0)y$. Furthermore, the property $\{I - P(f_0)\}D(f_0) = I - P^*(f_0)$ implies, $\phi + u^0(f_0) = r(f_0) + P(f_0)u^0(f_0)$. Let $z := y - u^0(f_0)$, then $z \geq P(f_0)z$, i.e. $z - P(f_0)z \geq 0$. Since $P^*(f_0)\{z - P(f_0)z\} = 0$ and because $P^*(f_0)$ has strictly positive elements, we have $z = P(f_0)z$. Consequently, $z = P^*(f_0)z = c \cdot e$ for some $c \in \mathbb{R}$, (the last equality because $P^*(f_0)$ has identical rows). Hence, $y = u^0(f_0) + c \cdot e$. □

Example 6.1

The following model, which was also used in Example 5.8, does not satisfy the irreducibility assumption. We show that in that case the optimality equation (6.1) cannot be used.

$S = \{1, 2, 3\};\ A(1) = A(2) = \{1, 2\},\ A(3) = 1;\ r_1(1) = 3,\ r_1(2) = 1,\ r_2(1) = 0,\ r_2(2) = 1,\ r_3(1) = 2.$

$p_{11}(1) = 1,\ p_{12}(1) = p_{13}(1) = 0;\ p_{11}(2) = 0,\ p_{12}(2) = 1,\ p_{13}(2) = 0;\ p_{21}(1) = 0,\ p_{22}(1) = 1,\ p_{23}(1) = 0;$
$p_{21}(2) = p_{22}(2) = 0,\ p_{23}(2) = 1;\ p_{31}(1) = p_{32}(1) = 0,\ p_{33}(1) = 1.$

The optimality equation (6.1) becomes:

$$x + y_1 = max\{3 + y_1, 1 + y_2\};\ x + y_2 = max\{0 + y_2, 1 + y_3\};\ x + y_3 = 2 + y_3.$$

The third equation gives $x = 2$. If we use this value in the first equation, we obtain:
$2 + y_1 = max\{3 + y_1, 1 + y_2\} \geq 3 + y_1$, implying that the system is infeasible.

Remark

We give a heuristic derivation that this optimality equation can be derived from the optimality equation for the discounted reward when the discount factor α tends to 1. First, we write the the optimality equation for the discounted reward as $0 = max_{a \in A(i)} \big\{ r_i(a) + \alpha \sum_j p_{ij}(a) v_j^\alpha - v_i^\alpha \big\},\ i \in S.$

Then, we use the first terms of the Laurent expansion: $v^\alpha = \frac{\phi \cdot e}{1-\alpha} + u^0 + \varepsilon(\alpha)$. So, we obtain

$$\begin{aligned} 0 &= max_{a \in A(i)} \big\{ r_i(a) + \alpha \textstyle\sum_j p_{ij}(a) \{ \frac{\phi}{1-\alpha} + u_j^0 \} - \{ \frac{\phi}{1-\alpha} + u_i^0 \} + \varepsilon(\alpha) \big\},\ i \in S. \\ 0 &= max_{a \in A(i)} \big\{ r_i(a) + \alpha \textstyle\sum_j p_{ij}(a) u_j^0 - \phi - u_i^0 + \varepsilon(\alpha) \big\},\ i \in S. \end{aligned}$$

Denote ϕ by x and u^0 by y and let α tends to 1. Then, we have

$0 = max_{a \in A(i)} \big\{ r_i(a) + \sum_j p_{ij}(a) y_j - x - y_i \big\},\ i \in S$, i.e. $x + y_i = max_{a \in A(i)} \Big\{ r_i(a) + \sum_j p_{ij}(a) y_j \Big\},\ i \in S.$

6.1.2 Policy iteration

Theorem 6.2
The linear system $\left\{ \begin{array}{rcl} x \cdot e + \{I - P(f)\}y & = & r(f) \\ y_1 & = & 0 \end{array} \right.$ *has* $x = \phi(f^\infty)$ *and* $y = u^0(f) - u^0_1(f) \cdot e$*. as unique solution.*

Proof
Multiply the first equality by $P^*(f)$:
$x \cdot P^*(f)e + P^*(f)\{I - P(f)\}y = P^*(f)r(f) = \phi(f^\infty) \cdot e \ \rightarrow \ x \cdot e = \phi(f^\infty) \cdot e$, i.e. $x = \phi(f^\infty)$.
Since $x = \phi(f^\infty) = P^*(f)r(f)$, the first equation can be written as:
$\{I - P(f) + P^*(f)\}y = r(f) - P^*(f)r(f) + P^*(f)y$, implying

$$\begin{array}{rcl} y & = & \{I - P(f) + P^*(f)\}^{-1}\big\{\big(I - P^*(f)\big)r(f) + P^*(f)y\big\} \\ & = & \{D(f) + P^*(f)\}\big\{\big(I - P^*(f)\big)r(f) + P^*(f)y\big\} = D(f)r(f) + P^*(f)y = u^0(f) + c \cdot e. \end{array}$$

Because $y_1 = 0$, we have, $c = -u^0_1(f)$. Hence, $y = u^0(f) - u^0_1(f) \cdot e$. □
For every $i \in S$ and $f^\infty \in C(D)$, we define the action set $B(i, f)$ by

$$B(i, f) := \big\{a \in A(i) \mid r_i(a) + \textstyle\sum_j p_{ij}(a)u^0_j(f) > \phi(f^\infty) + u^0_i(f)\big\}.$$

Since $u^0(f)$ and the solution y of the system in Theorem 6.1 differ a constant, we also have

$$B(i, f) = \big\{a \in A(i) \mid r_i(a) + \sum_j p_{ij}(a)y_j > \phi(f^\infty) + y_i\big\}. \tag{6.2}$$

Theorem 6.3
(1) *If* $B(i, f) = \emptyset$ *for every* $i \in S$*, then* f^∞ *is an average optimal policy.*
(2) *If* $B(i, f) \neq \emptyset$ *for at least one* $i \in S$ *and the policy* $g^\infty g \neq f^\infty$ *satisfies* $g(i) \in B(i, f)$ *if* $g(i) \neq f(i)$*, then* $\phi(g^\infty) > \phi(f^\infty)$*.*

Proof
(1) If $B(i, f) = \emptyset$ for every $i \in S$, then $r(g) + P(g)u^0(f) \leq \phi(f^\infty) \cdot e + u^0(f)$ for all $g^\infty \in C(D)$.
Hence, $P^*(g)r(g) + P^*(g)P(g)u^0(f) \leq \phi(f^\infty) \cdot P^*(g)e + P^*(g)u^0(f)$ for all $g^\infty \in C(D)$, i.e. $\phi(g^\infty) \cdot e + P^*(g)u^0(f) \leq \phi(f^\infty) \cdot e + P^*(g)u^0(f)$. Therefore, $\phi(g^\infty) \leq \phi(f^\infty)$ for all $g^\infty \in C(D)$: f^∞ is average optimal.

(2) If $g(i) = f(i)$, then row i of $P(f)$ is identical to row i of $P(g)$, and also $r_i(f) = r_i(g)$. Hence, $\{r(g) + P(g)u^0(f)\}_i = \{r(f) + P(f)u^0(f)\}_i = \{P^*(f)r(f) + u^0(f\}_i = \phi(f^\infty) + u^0_i(f)$, the last but one equality because $I + P(f)D(f) = P^*(f) + D(f)$.
If $g(i) \neq f(i)$, then $g(i) \in B(i, f)$ and $\{r(g) + P(g)u^0(f)\}_i > \phi(f^\infty) + u^0_i(f)$. Therefore, we have $r(g) + P(g)u^0(f) > \phi(f^\infty) \cdot e + u^0(f)$. Since the elements of $P^*(g)$ are strictly positive, we obtain $P^*(g)r(g) + P^*(g)u^0(f) > \phi(f^\infty) \cdot e + P^*(g)u^0(f)$: $\phi(g^\infty) > \phi(f^\infty)$. □

Algorithm 6.1 *Determination of an average optimal policy by policy iteration (irreducible case)*
Input: Instance of an irreducible MDP.
Output: An optimal deterministic policy f^∞ and the value ϕ.

1. Select an arbitrary $f^\infty \in C(D)$.

2. Determine the unique solution $\big(x = \phi(f^\infty), y\big)$ of the linear system
$$\begin{cases} x \cdot e + \{I - P(f)\}y &= r(f) \\ y_1 &= 0 \end{cases}$$
3. **for all** $i \in S$ **do** $B(i,f) := \{a \in A(i) \mid r_i(a) + \sum_j p_{ij}(a)y_j > \phi(f^\infty) + y_i\}$.
4. **if** $B(i,f) = \emptyset$ for every $i \in S$ **then**

```
        begin $f^\infty$ is an average optimal policy; $\phi(f^\infty)$ is the value $\phi$ (STOP) end
else    begin select $g \neq f$ such that $g(i) \in B(i,f)$ if $g(i) \neq f(i)$; $f := g$; return to step 2
        end
```

Example 6.2
Apply Algorithm 6.1 to the following model (easy to check that the model is irreducible).
$S = \{1,2\};\ A(1) = A(2) = \{1,2\};\ r_1(1) = 4,\ r_1(2) = 2,\ r_2(1) = 3,\ r_2(2) = 1;$
$p_{11}(1) = \frac{1}{3}, p_{12}(1) = \frac{2}{3};\ p_{11}(2) = \frac{2}{3}, p_{12}(2) = \frac{1}{3};\ p_{21}(1) = \frac{1}{2}, p_{22}(1) = \frac{1}{2};\ p_{21}(2) = \frac{1}{2}, p_{22}(2) = \frac{1}{2}.$
Start with f^∞, where $f(1) = 2,\ f(2) = 1$.

Iteration 1:

The system is $\begin{cases} x + \frac{1}{3}y_1 - \frac{1}{3}y_2 = 2 \\ x - \frac{1}{2}y_1 + \frac{1}{2}y_2 = 1 \\ y_1 = 0 \end{cases}$ with solution $x = \frac{8}{5},\ y_1 = 0,\ y_2 = -\frac{6}{5}$.

$B(1,f) = B(2,f) = \{1\}$. Select $g(1) = g(2) = 1$. Then, $f(1) = f(2) = 1$

Iteration 2:

The system is $\begin{cases} x + \frac{2}{3}y_1 - \frac{2}{3}y_2 = 4 \\ x - \frac{1}{2}y_1 + \frac{1}{2}y_2 = 3 \\ y_1 = 0 \end{cases}$ with solution $x = \frac{24}{7},\ y_1 = 0,\ y_2 = -\frac{6}{7}$.

$B(1,f) = B(2,f) = \emptyset$: f^∞, where $f(1) = f(2) = 1$, is an optimal policy and the value is $\frac{24}{7}$.

6.1.3 Linear programming

Since the value vector is the smallest superharmonic vector (cf. Theorem 5.17), in the case where ϕ is a constant, ϕ is the unique x-solution of the linear program

$$min \left\{ x \;\middle|\; x + \sum_j \{\delta_{ij} - p_{ij}(a)\}y_j \geq r_i(a),\ i \in S,\ a \in A(i) \right\}. \tag{6.3}$$

The dual of (6.3) is:

$$max \left\{ \sum_{i,a} r_i(a)x_i(a) \;\middle|\; \begin{array}{lcl} \sum_{i,a}\{\delta_{ij} - p_{ij}(a)\}x_i(a) & = & 0,\ j \in S \\ \sum_{i,a} x_i(a) & = & 1 \\ x_i(a) \geq 0,\ i \in S,\ a \in A(i) & & \end{array} \right\}. \tag{6.4}$$

Remark
We show that the dual linear program (6.4) can be considered as the dual linear program (3.32) for the discounted reward in which the discount factor tends to 1. First, we remark that if we summing up the constraints of (3.32), we obtain $\sum_{i,a} (1-\alpha)x_i(a) = \sum_j \beta_j$. If we take β_j such that $\sum_j \beta_j = 1$, add the

(redundant) constraint $\sum_{i,a}(1-\alpha)x_i(a)=1$ and multiply both the objective function as the constraints of (3.32) with $(1-\alpha)$, the program becomes:

$$max\left\{\sum_{i,a} r_i(a)(1-\alpha)x_i(a) \;\middle|\; \begin{array}{lcl} \sum_{i,a}\{\delta_{ij}-\alpha p_{ij}(a)\}(1-\alpha)x_i(a) & = & (1-\alpha)\beta_j,\ j\in S \\ \sum_{i,a}(1-\alpha)x_i(a) & = & 1 \\ (1-\alpha)x_i(a)\geq 0,\ i\in S,\ a\in A(i) & & \end{array}\right\}. \tag{6.5}$$

Let $\alpha\uparrow 1$ and denote $\lim_{\alpha\uparrow 1}(1-\alpha)x_i(a)$ again by $x_i(a)$ for all $(i,a)\in S\times A$, then we get (6.4).

Theorem 6.4

Let (ϕ, y^*) and x^* be optimal solutions of (6.3) and (6.4), respectively. Let $f_*^\infty\in C(D)$ be such that $x_i^*\big(f_*(i)\big)>0,\ i\in S$. Then, f_*^∞ is well-defined and an optimal policy.

Proof

Let x a feasible solution of (6.4) (notice that program (6.4) is feasible, because (6.3) has a finite optimal solution) and let $x_i:=\sum_a x_i(a),\ i\in S$. Let $\pi^\infty\in C(S)$ defined by $\pi_i(a):=\begin{cases}\frac{x_i(a)}{x_i} & \text{if } x_i>0,\ i\in S;\\ \text{arbitrary} & \text{if } x_i=0,\ i\in S.\end{cases}$

Hence, $x_i(a)=\pi_i(a)\cdot x_i,\ (i,a)\in S\times A$ and $\sum_{i,a}\{\delta_{ij}-p_{ij}(a)\}\pi_i(a)\cdot x_i=0,\ j\in S$. Therefore, $x^T\{I-P(\pi)\}=0$, where $\{P(\pi\}_{ij}=\sum_a p_{ij}(a)\pi_i(a)$ for all $i,j\ \in S$, i.e. x is a stationary distribution of the Markov chain $P(\pi)$. Since the chain is irreducible, we have $x_i>0,\ i\in S$. Therefore, $x_i^*=\sum_a x_i^*(a)>0$ for all $i\in S$, i.e. f_*^∞ is a well-defined policy. From the orthogonality property of linear programming it follows that $x_i^*(a)\cdot\big\{\phi+\sum_j\{\delta_{ij}-p_{ij}(a)\}y_j^*-r_i(a)\big\}=0$ for all $(i,a)\in S\times A$. Hence, we have $\phi\cdot e+\{I-P(f_*)\}y^*=r(f_*)$. Multiply the last equality by $P^*(f_*)$: $\phi\cdot e=P^*(f_*)r(f_*)=\phi(f_*^\infty)\cdot e$, implying that f_*^∞ is an optimal policy. □

Algorithm 6.2

Determination of an average optimal policy by linear programming (irreducible case)

Input: Instance of an irreducible MDP.

Output: An optimal deterministic policy f_*^∞ and the value ϕ.

1. Determine an optimal solution of the linear program (6.4).
2. Select $f_*^\infty\in C(D)$ such that $x_i^*\big(f_*(i)\big)>0$ for every $i\in S$.
3. The value ϕ is the optimum value of program (6.4); f_*^∞ is an optimal policy (STOP).

Example 6.3

Apply Algorithm 6.2 to the following model (easy to check that the model is irreducible).
$S=\{1,2\};\ A(1)=A(2)=\{1,2\};\ r_1(1)=1,\ r_1(2)=0,\ r_2(1)=2,\ r_2(2)=5.$
$p_{11}(1)=\frac{1}{2},p_{12}(1)=\frac{1}{2};\ p_{11}(2)=\frac{1}{4},p_{12}(2)=\frac{3}{4};\ p_{21}(1)=\frac{2}{3},p_{22}(1)=\frac{1}{3};\ p_{21}(2)=\frac{1}{3},p_{22}(2)=\frac{2}{3}.$
The linear program (6.4) is:

$$max\{1\cdot x_1(1)+0\cdot x_1(2)+2\cdot x_2(1)+5\cdot x_2(2)\}$$

subject to

$$x_1(1)+x_1(2)=\tfrac{1}{2}x_1(1)+\tfrac{1}{4}x_1(2)+\tfrac{2}{3}x_2(1)+\tfrac{1}{3}x_2(2);$$

$$x_2(1)+x_2(2)=\tfrac{1}{2}x_1(1)+\tfrac{3}{4}x_1(2)+\tfrac{1}{3}x_2(1)+\tfrac{2}{3}x_2(2);$$

$$x_1(1)+x_1(2)+x_2(1)+x_2(2)=1;$$

$$x_1(1),\ x_1(2),\ x_2(1),\ x_2(2)\geq 0.$$

The optimal optimal solution is: $x_1^*(1)=0,\ x_1^*(2)=\frac{4}{13},\ x_2^*(1)=0,\ x_2^*(2)=\frac{9}{13};\ optimum=\frac{45}{13}$.
Therefore, the optimal policy is: $f_*(1)=2,\ f_*(2)=2$ and the value $\phi=\frac{45}{13}$.

As in the discounted case, there is a bijection between the feasible solutions of the dual program (6.5) and the set $C(S)$ of stationary policies. Let π^∞ be a stationary policy and let $x(\pi)$ be the stationary distribution of $P(\pi)$. Define x^π by

$$x_i^\pi(a) := x_i(\pi) \cdot \pi_{ia},\ (i,a) \in S \times A. \tag{6.6}$$

Reversely, let x be a feasible solution of (6.5). Define π^x by

$$\pi_{ia}^x := \frac{x_i(a)}{\sum_a x_i(a)},\ (i,a) \in S \times A. \tag{6.7}$$

Theorem 6.5

The mapping (6.6) is a bijection between the feasible solutions of the dual program (6.5) and the set $C(S)$ with (6.7) as the reverse mapping. Furthermore, the extreme solutions of (6.5) correspond to the set $C(D)$ of deterministic policies.

Proof

Let π^∞ be any stationary policy. Then, x^π, defined by (6.6), satisfies

$$\begin{cases} \sum_{i,a}\{\delta_{ij} - p_{ij}(a)\}x_i^\pi(a) = \sum_i\{\delta_{ij} - p_{ij}(\pi)\}x_i(\pi) = \big\{\{x(\pi)\}^T\{I - P(\pi)\}\big\}_j = 0,\ j \in S; \\ \sum_{i,a} x_i^\pi(a) = \sum_i x_i(\pi) = 1 \text{ and } x_i^\pi(a) \geq 0,\ (i,a) \in S \times A. \end{cases}$$

Hence, x^π is a feasible solution of (6.5).

Conversely, let x be a feasible solution of (6.5). From the proof of Theorem 6.4 it follows that $\sum_a x_i(a) > 0$ for all $i \in S$; so, the policy π^x is well-defined. Since $\pi_{ia}^{x^\pi} = \pi_{ia}$ for all $(i,a) \in S \times A$, (6.6) is a bijection with (6.7) as the reverse mapping.

Let $f^\infty \in C(D)$ and suppose x^f is not an extreme point, i.e. $x^f = \lambda x^1 + (1-\lambda)x^2$, where $\lambda \in (0,1)$, $x^1 \neq x^2$ and x^1, x^2 are feasible solutions of (6.5). Since for all $i \in S$, we have $x_i^f(a) = 0$ for $a \neq f(i)$, also for all $i \in S$, we have $x_i^1(a) = x_i^2(a) = 0$ for $a \neq f(i)$. Therefore, both x^1 and x^2 are solutions of the same linear system $x^T\{I - P(f)\} = 0,\ x^Te = 1$, which has a unique solution: $x^1 = x^2$, implying a contradiction, showing that x^f is an extreme point.

Finally, let x be an extreme point of (6.5). Since the sum of the first N components is zero in every column, the rank of the whole system ($N+1$ equations) is at most N. Therefore, any extreme solution has at most N positive components. Since, $\sum_a x_i(a) > 0$ for all $i \in S$, x has in each state exactly one positive component. Hence, the corresponding policy is deterministic. □

Next, we show the equivalence between linear programming and policy iteration. Consider a deterministic policy f^∞. We have seen that x^f is an extreme point of (6.5) and that $x_i^f(f(i)) > 0$ for every $i \in S$. In the simplex tableau corresponding to x^f, the column of a nonbasic variable $x_i(a)$ has as reduced costs (the transformed objective function value) $d_i(a) = x + \sum_j \{\delta_{ij} - p_{ij}(a)\}y_j - r_i(a)$.

Since $x_i^f(f(i)) > 0$ for all $i \in S$, it follows from the complementary slackness property of linear programming that $d_i(f(i)) = 0$ for all $i \in S$. This property implies $x \cdot e + \{I - P(f)\}y = r(f)$, and consequently $x \cdot e = P^*(f)r(f) = \phi(f^\infty) \cdot e$. Hence, we have $x = \phi(f^\infty)$ and $\phi(f^\infty) \cdot e + \{I - P(f)\}y = r(f)$. Since we also have $\phi(f^\infty) \cdot e + \{I - P(f)\}u^0(f) = r(f)$, we obtain $\{I - P(f)\}\{y - u^0(f)\} = 0$, i.e. $y - u^0(f) = P(f)\{y - u^0(f)\}$. Hence, $y - u^0(f) = P^*(f)\{y - u^0(f)\} = c \cdot e$ for some scalar c.

This implies that the reduced costs satisfy $d_i(a) = \phi(f^\infty) + \sum_j \{\delta_{ij} - p_{ij}(a)\}u^0(f) - r_i(a)$. Since $a \in B(i,f)$ if and only if $d_i(a) < 0$, it follows that the set of actions from which $g(i)$ may be chosen in policy iteration corresponds to the possible choices for the pivot column in the simplex method, which yieds the following theorem.

Theorem 6.6

(1) Any policy iteration algorithm is equivalent to a block-pivoting simplex algorithm.

(2) Any simplex algorithm is equivalent to a particular policy iteration algorithm.

Detecting nonoptimal actions

In this section we present conditions to detect nonoptimal actions in irreducible MDPs with average rewards. They can be implemented in both the policy iteration and the linear programming algorithm. Also an implementation in the so-called 'new policy iteration scheme' (see Algorithm 6.5 in section 6.2.2) is discussed.

The dual program for irreducible MDPs is (6.4). Since the equalities $\sum_{i,a}\{\delta_{ij}-p_{ij}(a)\}x_i(a)=0,\ j\in S$, imply (by adding these N equalities) the trial identity $\sum_{j,a} x_j(a)=\sum_{i,a} x_i(a)$, the linear program (6.4) is degenerated. From Theorem 6.5 it follows that any basic solution x of (6.4) corresponds to a deterministic policy f^∞, i.e. $x=x^f$ for some $f^\infty\in C(D)$. Since $x^f_i>0$ for all $i\in S$ (see the proof of Theorem 6.5), any matrix $I-P(f)$ has rank $N-1$. Hence, program (6.4) is equivalent to the nondegenerated linear program

$$max\left\{\sum_{i,a} r_i(a)x_i(a) \;\middle|\; \begin{array}{lcl} \sum_{i,a}\{\delta_{ij}-p_{ij}(a)+\frac{1}{N}\}x_i(a) & = & \frac{1}{N},\ j\in S \\ x_i(a)\geq 0,\ i\in S,\ a\in A(i) & & \end{array}\right\}. \tag{6.8}$$

For nondegenerated linear programs, the matrix A of the constraints and the basic solution x - corresponding to $f^\infty\in C(D)$ - can be partitioned into $A=(B,C)$ and $x=(x_B,x_C)$, where $x_C=0$ and $(x_B)_i=x^f_i,\ i\in S$. Furthermore, we have $Bx_B=\{I-P^T(f)+\frac{1}{N}ee^T\}x_B=\frac{e}{N}$. In the next lemma we show relation between the matrix B and the fundamental matrix $Z(f)$.

Lemma 6.1

$B^{-1}=\{(I-\frac{1}{N}ee^T)Z(f)+e\pi(f)^T\}^T$, *where* $\pi(f)$ *is the stationary vector of the irreducible matrix* $P(f)$.

Proof

We have to show that $B^T\{(I-\frac{1}{N}ee^T)Z(f)+e\pi(f)^T\}=I$. Let $X=B^T\{(I-\frac{1}{N}ee^T)Z(f)+e\pi(f)^T\}$, then we can write

$$\begin{aligned} X &= B^T\{(I-\tfrac{1}{N}ee^T)Z(f)+e\pi(f)^T\}=\{I-P(f)+\tfrac{1}{N}ee^T\}\{(I-\tfrac{1}{N}ee^T)Z(f)+e\pi(f)^T\} \\ &= \{Z(f)-\tfrac{1}{N}ee^TZ(f)+e\pi(f)^T\}-\{P(f)Z(f)-\tfrac{1}{N}P(f)ee^TZ(f)+e\pi(f)^T\} \\ &\qquad\qquad\qquad\qquad\qquad\qquad +\{\tfrac{1}{N}ee^TZ(f)-\tfrac{1}{N}ee^TZ(f)+e\pi(f)^T\} \\ &= Z(f)-P(f)Z(f)+e\pi(f)^T=\{I-P(f)\}Z(f)+P^*(f) \\ &= \{I-P(f)\}\{D(f)+P^*(f)\}+P^*(f)=I-P^*(f)+P^*(f)=I. \end{aligned}$$

□

Given the current simplex vertex (x_B,x_C), the simplex method works as follow:

1. Compute the minimum shadow price of the nonbasic variables x_C. For the nonbasic variable $x_j(a)$ this shadow price equals $z_j(a)-r_j(a)=r^T_BB^{-1}A_{j,a}-r_j(a)$, where $r_B=r(f)$ and the column vector $A_{j,a}$ has elements $\delta_{jk}-p_{jk}(a)+\frac{1}{N},\ k\in S$. Notice that if all shadow prices are nonnegative the algorithm terminates with an average optimal policy f^∞.
2. Select the index (j,a) of the minimum, i.e. most negative, shadow price. Then, $x_j(a)$ enters the basis and x^f_i leaves the basis.

From Lemma 6.1, we obtain $\{r^T_BB^{-1}\}^T=\{(I-\frac{1}{N}ee^T)Z(f)+e\pi(f)^T\}r(f)=(I-\frac{1}{N}ee^T)Z(f)r(f)+\phi(f^\infty)\cdot e$. Hence,

$$\begin{array}{rcl}
z_j(a) & = & \{r_B^TB^{-1}A_{j,a}\}^T = \sum_k \{A_{j,a}\}_k \cdot \{(B^{-1})^Tr_B\}_k \\
 & = & \sum_k \{\delta_{jk} - p_{jk}(a) + \frac{1}{N}\} \cdot \big\{(I - \frac{1}{N}ee^T)Z(f)r(f) + \phi(f) \cdot e\big\}_k \\
 & = & \sum_k \{\delta_{jk} - p_{jk}(a) + \frac{1}{N}\} \cdot \{[Z(f)r(f)]_k - \frac{1}{N}\sum_i [Z(f)r(f)]_i + \phi(f^\infty)\} \\
 & = & [Z(f)r(f)]_j - \sum_k p_{jk}(a)[Z(f)r(f)]_k + \frac{1}{N}\sum_k [Z(f)r(f)]_k - \frac{1}{N}\sum_i [Z(f)r(f)]_i + \phi(f^\infty) \\
 & = & [Z(f)r(f)]_j - \sum_k p_{jk}(a)[Z(f)r(f)]_k + \phi(f^\infty).
\end{array}$$

Therefore, the shadow prices satisfy $z_j(a) - r_j(a) = [Z(f)r(f)]_j - \sum_k p_{jk}(a)[Z(f)r(f)]_k + \phi(f^\infty) - r_j(a)$. Since the shadow prices are zero for all basic variables, we have $\phi(f^\infty) \cdot e + \{I - P(f)\}Z(f)r(f) = r(f)$, i.e. $Z(f)r(f)$ is a solution of the linear system

$$\phi(f^\infty) \cdot e + \{I - P(f)\}y = r(f). \tag{6.9}$$

However, this system has a solution which is unique up to a constant. Hence, the shadow prices also satisfy $z_j(a) - r_j(a) = y_j - \sum_k p_{jk}(a)y_k + \phi(f^\infty) - r_j(a)$, where y is any solution to equation (6.9). Using (6.9) the shadow prices can also be written as

$$z_j(a) - r_j(a) = \{r_j(f) - r_j(a)\} + \sum_k \{p_{jk}(f) - p_{jk}(a)\}y_k, \tag{6.10}$$

where y is any solution to equation (6.9).

Theorem 6.7

Let f^∞ be the policy of the current simplex tableau and let $x_j(a)$ be a nonbasic variable variable with reduced costs $z_j(a) - r_j(a)$.

If either $\delta_{jk}+\sum_l \{p_{jl}(f)-p_{jl}(a)\}\{Z(f)\}_{lk} \geq 0$ for all $k \in S$ or $\delta_{jk}+\sum_l \{p_{jl}(f)-p_{jl}(a)\}\{Z(f)\}_{lk} < 0$ for at least one k and $z_j(a)-r_j(a)+\theta\cdot\{\overline{\phi}-\phi(f^\infty)\} > 0$, where $\overline{\phi} \geq \phi(f^\infty)$ and $\theta := min_k \frac{\delta_{jk}+\sum_l \{p_{jl}(f)-p_{jl}(a)\}\{Z(f)\}_{lk}}{\pi_k(f)}$, then action a is nonoptimal and can be deleted from $A(j)$.

Proof

We will show that this theorem follows from Theorem 3.21. Therefore, it is sufficient to show that $\{B^{-1}A_{j,a}\}_k = \delta_{jk} + \sum_l \{p_{jl}(f) - p_{jl}(a)\}\{Z(f)\}_{lk}$ for all $k \in S$. We have

$$\begin{array}{rcl}
\{B^{-1}A_{j,a}\}_k & = & \sum_l \{B^{-1}\}_{lk} \cdot \{A_{j,a}\}_l \\
 & = & \sum_l \{(I - \frac{1}{N}ee^T)Z(f) + e\pi(f)^T\}_{lk} \cdot \{\delta_{jl} - p_{jl}(a) + \frac{1}{N}\} \\
 & = & \sum_l \big\{[Z(f)]_{lk} - \frac{1}{N}\sum_i [Z(f)]_{ik} + \pi_k(f)\big\} \cdot \{\delta_{jl} - p_{jl}(a) + \frac{1}{N}\} \\
 & = & \big\{[Z(f)]_{jk} - \frac{1}{N}\sum_i [Z(f)]_{ik} + \pi_k(f)\big\} - \big\{\sum_l [Z(f)]_{lk} \cdot p_{jl}(a) - \frac{1}{N}\sum_i [Z(f)]_{ik}\pi_k(f)\big\} \\
 & & \qquad + \big\{\pi_k(f) - \pi_k(f) + \pi_k(f)\big\} \\
 & = & \{Z(f)\}_{jk} + \pi_k(f) - \sum_l \{Z(f)\}_{lk} \cdot p_{jl}(a) \\
 & = & \pi_k(f) + \sum_l \{\delta_{jl} - p_{jl}(a)\} \cdot \{Z(f)\}_{lk},\ k \in S.
\end{array}$$

Hence, we have to show that $\pi_k(f) = \delta_{jk} + \sum_l p_{jl}(f)\} \cdot \{Z(f)\}_{lk} - \{Z(f)\}_{jl}$ for all $k \in S$. Since $I - \{I - P(f)\}Z(f) = I - \{I - P(f)\}\{D(f) + P^*(f)\} = P^*(f)$, this is true. □

Remark

For the implementation of this suboptimality test, we need the values $\sum_l \{p_{jl}(f) - p_{jl}(a)\}\{Z(f)\}_{lk}$ for all $k \in S$. However, we do not nee to compute the matrix $Z(f)$ explicitly, but it suffices to update for each (j, a) the vectors $\sum_l \{p_{jl}(f) - p_{jl}(a)\}\{Z(f)\}_{lk},\ k \in S$. This can be done by a computation scheme of order $\sum_i |A(i)|$. For the details, we refer to [176].

The next two lemmata present interesting formulas, which hold even for unichain MDPs. The are related to the lemmata 3.6 and 3.6 for discounted MDPs.

Lemma 6.2
Let $f^\infty, g^\infty \in C(D)$. Then, $\phi(g^\infty) - \phi(f^\infty) = \pi(g)^T\{r(g) - r(f) + [P(g) - P(f)]y(f)\}$, where $y(f)$ is a relative value vector, i.e. $y(f)$ satisfies $\{I - P(f)\}y = r(f) - \phi(f^\infty) \cdot e$.

Proof
We can write

$$\begin{aligned}\pi(g)^T\{r(g) - r(f) + [P(g) - P(f)]y(f)\} &= \phi(g^\infty) - \pi(g)^T\{r(f) - P(g)y(f) + P(f)y(f)\}\\ &= \phi(g^\infty) - \pi(g)^T\{\phi(f^\infty)\cdot e + [I - P(g)]y(f)\}\\ &= \phi(g^\infty) - \phi(f^\infty).\end{aligned}$$

□

Lemma 6.3
Let $f^\infty, g^\infty \in C(D)$. Then, $\pi(f)^T = \pi(g)^T\big\{I - [P(g) - P(f)]Z(f)\big\}$.

Proof
We can write, using $\pi(g)^TP(g) = \pi(g)^T, D(f)P(f) = P(f)D(f)$ and $D(f)[I - P(f)] = I - P^*(f)$,

$$\pi(g)^T\big\{I - [P(g) - P(f)]Z(f)\big\}P(f) = \pi(g)^T\big\{P(f) - [P(g) - P(f)]\{D(f)P(f) + P^*(f)\}\big\} =$$
$$\pi(g)^T\big\{P(f) - P(g)D(f)P(f) - P(g)P^*(f) + P(f)D(f)P(f) + P^*(f)\big\} =$$
$$\pi(g)^T\big\{P(f) - D(f)P(f) - P^*(f) + P(f)D(f)P(f) + P^*(f)\big\} =$$
$$\pi(g)^T\big\{P(f)\{I - D(f) + D(f)P(f)\}\big\} = \pi(g)^T\big\{P(f)\{I - D(f)[I - P(f)]\}\big\} =$$
$$\pi(g)^T\big\{P(f)\{I - I + P^*(f)]\}\big\} = \pi(g)^TP^*(f) = \pi(g)^Te\,\pi(f)^T = \pi(f)^T.$$

□

The following theorem gives an interpretation of suboptimal actions in the sense that either $\pi(f) \geq \pi(g)$ or $\pi(f) \ngeq \pi(g)$ and $\phi(g^\infty) - \phi(f^\infty) < \theta \cdot \{\overline{\phi} - \phi(f^\infty)\}$ with ϕ an upper bound of the value and θ some negative coefficient, defined by $\theta := min_{k \neq j}\, \frac{\pi_k(f) - \pi_k(g)}{\pi_k(f)}$.

Theorem 6.8
Let f^∞ be the policy of the current simplex tableau, let $x_j(a)$ be a nonbasic variable with shadow price $z_j(a) - r_j(a) > 0$, and let g^∞ be the policy with $f(i) = g(i),\ i \neq j$ and $g(j) = a$. Then, action $a \in A(j)$ is suboptimal if either $\pi(f) \geq \pi(g)$ or $\pi(f) \ngeq \pi(g)$ and $\phi(g^\infty) - \phi(f^\infty) < \theta \cdot \{\overline{\phi} - \phi(f^\infty)\}$ with ϕ an upper bound of the value and θ some negative coefficient, defined by $\theta := min_{k\neq j}\, \frac{\pi_k(f) - \pi_k(g)}{\pi_k(f)}$.

Proof
We have, by Lemma 6.3, $\pi(f)^T = \pi(g)^T\big\{I - [P(g) - P(f)]Z(f)\big\}$ for all $f^\infty, g^\infty \in C(D)$. By the special case of g^∞ in this theorem, the matrix $[P(g) - P(f)]Z(f)$ has zero rows, except for row j which has the elements $\sum_l \{p_{jl}(g) - p_{jl}(f)\}\{Z(f)\}_{lk}$ for $k \in S$. Hence, $\pi_k(f) = \pi_k(g) - \pi_j(g) \cdot \sum_l \{p_{jl}(g) - p_{jl}(f)\}\{Z(f)\}_{lk}$ for $k \in S$. Therefore,

If $k \neq j$, then $\frac{\pi_k(f) - \pi_k(g)}{\pi_j(g)} = \delta_{jk} + \sum_l \{p_{jl}(f) - p_{jl}(g)\}\{Z(f)\}_{lk}$.

If $k = j$, then $0 \leq \frac{\pi_j(f)}{\pi_j(g)} = \delta_{jk} + \sum_l \{p_{jl}(f) - p_{jl}(g)\}\{Z(f)\}_{lk}$.

<u>Case 1:</u> $\pi(f) \geq \pi(g)$

Then, $\delta_{jk} + \sum_l \{p_{jl}(f) - p_{jl}(g)\}\{Z(f)\}_{lk} \geq 0$ for all $k \in S$ and the result follows from Theorem 6.7.

Case 2: $\pi(f) \not\geq \pi(g)$

In this case $\delta_{jk} + \sum_l \{p_{jl}(f) - p_{jl}(g)\}\{Z(f)\}_{lk} < 0$ for at least one $k \in S$. By Theorem 6.7 it is sufficient to show that $z_j(a) - r_j(a) + min_k \frac{\delta_{jk} + \sum_l \{p_{jl}(f) - p_{jl}(a)\}\{Z(f)\}_{lk}}{\pi_k(f)} \cdot \{\overline{\phi} - \phi(f^\infty)\} > 0$.

From Lemma 6.2 and the property that element k and row k of $r(g) - r(f)$ and $P(g) - P(f)$ are zero for $k \neq j$, we obtain

$\phi(g^\infty) - \phi(f^\infty) = \pi_j(g) \cdot \{r_j(g) - r_j(f) + \sum_l \{p_{jl}(g) - p_{jl}(f)\}y_j(f) = -\pi_j(g) \cdot \{z_j(a) - r_j(a)\}$,

the last equality by relation (6.10). From the above lines, it follows that we have to show that $\phi(f^\infty) - \phi(g^\infty) + \frac{\pi_k(f) - \pi_k(g)}{\pi_k(f)} \cdot \{\overline{\phi} - \phi(f^\infty)\} > 0$ for all $k \neq j$, which is true by the assumption in the formulation of this theorem. □

6.1.4 Value iteration

In section 5.9 we presented Algorithm 5.10 for value iteration under the assumption that the value vector is constant and the Markov chains $P(f),\ f^\infty \in C(D)$, are aperiodic. The last part of this assumption is not a serious restriction: by a data transformation, the original model can be transformed into a model in which every Markov chain $P(f),\ f^\infty \in C(D)$, is aperiodic and has the same average reward as the original Markov chain. In case of irreducibility no better algorithm than Algorithm 5.10 is known.

6.1.5 Modified policy iteration

In average reward models, value iteration may converge very slowly, and policy iteration may be inefficient in models with many states because of the need to solve large linear systems of equations. As in discounted models, modified policy iteration provides a compromise between these two algorithms. It avoids many value iterations and it avoids solving the linear system. We also can develop a modified policy iteration algorithm for the average reward criterion in the case that all MDPs are irreducible. Let the operators T and T_f, for $f^\infty \in C(D)$, be defined by

$$(Tx)_i := max_a\{r_i(a) + \sum_j p_{ij}(a)x_j\},\ i \in S;\ T_f\, x := r(f) + P(f)x. \tag{6.11}$$

Notice that for $k \in \mathbb{N}$, $T_f^k\, x = r(f) + P(f)r(f) + \cdots + P^{k-1}(f)r(f) + P^k(f)x = v^k(f^\infty) + P^k(f)x$, where $v^k(f^\infty)$ is the total reward over k periods when policy f^∞ is used.

Algorithm 6.3 *Modified value iteration (irreducible case)*
Input: Instance of an irreducible MDP and some scalar $\varepsilon > 0$.
Output: An ε-optimal deterministic policy f^∞ and a $\frac{1}{2}\varepsilon$-approximation of the value ϕ.

```
1. Select $x \in \mathbb{R}^N$ and $k \in \mathbb{N}$ arbitrary; determine $f$ such that $T_f\, x = T\, x$.

2. $l := min_i\,(Tx - x);\ u := max_i\,(Tx - x)$.

3. if $u - l \leq \varepsilon$ then
   begin $f^\infty$ is an $\varepsilon$-optimal policy and $\frac{1}{2}(u + l)$ is a $\frac{1}{2}\varepsilon$-approximation of the value $\phi$ (STOP) end
   else begin $x := T_f^k\, x$; return to step 2 end
```

We work in the remaining part of this subsection under the following *strong aperiodicity* assumption.

Assumption 6.2
$p_{ii}(a) > 0$ for all $(i,a) \in S \times A$.

We have seen in section 5.9 that the data transformation (5.48) gives strong aperiodicity without changing the average reward. So, this assumption does not give an essential restriction.

If $k = 1$, the method becomes the standard value iteration method. We will also argue that policy iteration corresponds to $k = \infty$. Let $\{x^n\}$, $\{f_n^\infty\}$ and $\{k_n\}$ be the values of x, f and k in iteration $n+1$ of Algorithm 6.3, i.e. $T_{f_n}\, x^n = T\, x^n$ and $x^{n+1} = T_{f_n}^{k_n}\, x^n$. Since, by Theorem 5.8, $v^k(f^\infty) = k \cdot \phi(f^\infty) + u^0(f) - P^k(f)u^0(f)$ for every policy f^∞, we obtain

$$x^{n+1} = T_{f_n}^{k_n}\, x^n = v^{k_n}(f_n^\infty) + P^{k_n}(f_n)x^n = k_n \cdot \phi(f_n^\infty) + u^0(f_n) - P^{k_n}(f_n)\{u^0(f_n) - x^n\}.$$

If $k_n \to \infty$, then we have $P^{k_n}(f_n) \to P^*(f_n)$ and therefore $P^{k_n}(f_n)\{u^0(f_n) - x^n\}$ converges to a constant vector. Hence, x^{n+1} and $u^0(f_n)$ differ a constant vector for a large k_n. In policy iteration with best improving actions, a new policy in state i is obtained by maximizing $r_i(a) + \sum_j\, p_{ij}(a)u_j^0(f_n)$, which gives the same policy as maximizing $r_i(a) + \sum_j\, p_{ij}(a)x_j^{n+1} = \big(Tx^{n+1}\big)_i$, which is the determination of the policy in step 1 of Algorithm 6.3.

Lemma 6.4
Let $g^n := Tx^n - x^n$, $l_n := min_i\, g_i^n$, and $u_n := max_i\, g_i^n$ for all $i \in S$ and $n \in \mathbb{N}$. Then,
$l_n \le \phi(f_n^\infty) \le \phi \le u_n$ for all $f^\infty \in C(D)$ and all $n \in \mathbb{N}$.

Proof
For all $f^\infty \in C(D)$, we have $P^*(f)\{r(f) + P(f)x^n - x^n\} = \phi(f^\infty) \cdot e$. So, with $f = f_n$, we can write

$$\phi(f_n^\infty) \cdot e \quad = \quad P^*(f_n)\{r(f_n) + P(f_n)x^n - x^n\} = P^*(f_n)\{Tx^n - x^n\} \ge P^*(f_n)l_n \cdot e = l_n \cdot e.$$

Clearly, $\phi(f_n^\infty) \le \phi$, and with $f = f_*$, where f_*^∞ is an average optimal policy, we obtain

$$\phi \cdot e = \phi(f_*^\infty) \cdot e = P^*(f_*)\{r(f_*) + P(f_*)x^n - x^n\} \le P^*(f_*)\{Tx^n - x^n\} \ge P^*(f_*)u_n \cdot e = u_n \cdot e.$$ □

Lemma 6.5
The sequence $\{l_n,\ n = 0, 1, \ldots\}$ is monotonically nondecreasing.

Proof
For $n = 0, 1, \ldots$, we have

$$\begin{aligned} T\, x^{n+1} - x^{n+1} \quad &\ge \quad T_{f_n}\, x^{n+1} - x^{n+1} = T_{f_n}^{k_n+1}\, x^n - T_{f_n}^{k_n}\, x^n \\ &= \quad \{r(f_n) + P(f_n)r(f_n) + \cdots + P^{k_n}(f_n)r(f_n)) + P^{k_n+1}(f_n)x^n\} - \\ &\qquad\qquad \{r(f_n) + P(f_n)r(f_n) + \cdots + P^{k_n-1}(f_n)r(f_n) + P^{k_n}(f_n)x^n\} \\ &= \quad P^{k_n}(f_n)\{T_{f_n}\, x^n - x^n\} = P^{k_n}(f_n)\{T\, x^n - x^n\} \ge l_n \cdot P^{k_n}(f_n) \cdot = l_n \cdot e. \end{aligned}$$

Hence, $min_i\, (Tx^{n+1} - x^{n+1})_i = l_{n+1} \ge l_n$ for $n = 0, 1, \ldots$. □

In the special case $k = 1$ (value iteration), also the sequence $\{u_n,\ n = 0, 1, \ldots\}$ is monotone, actually nonincreasing (see Theorem 5.25). However, this is not the case if $k \ge 2$, as the next example shows.

Example 6.4
$S = \{1,2\};\; A(1) = \{1\},\; A(2) = \{1,2\};\; r_1(1) = 100,\; r_2(1) = 0,\; r_2(2) = 10.$
$p_{11}(1) = 1, p_{12}(1) = 0;\; p_{21}(1) = 0.9,\; p_{22}(1) = 0.1;\; p_{21}(2) = 0.1,\; p_{22}(2) = 0.9.$
Start with $x^0 = (0,0)$ and take $k = 2$.
Iteration 1:
$(T\,x^0)_1 = 100 + 1 \times 0 = 100;\; (T\,x^0)_2 = max\{0 + 0.9 \times 0 + 0.1 \times 0, 10 + 0.1 \times 0 + 0.9 \times 0\} = 10.$
$f_0(1) = 1,\; f_0(2) = 2;\; l_0 = 10,\; u_0 = 100.$
$x^1 = T_{f_0}^2\,x^0 = T_{f_0}\{T\,x^0\} = (100 + 100, 10 + 0.1 \times 0, 10 + 0.1 \times 100 + 0.9 \times 10) = (200, 29).$
Iteration 2:
$(T\,x^1)_1 = 100 + 1 \times 200 = 300;$
$(T\,x^1)_2 = max\{0 + 0.9 \times 200 + 0.1 \times 29, 10 + 0.1 \times 200 + 0.9 \times 29\} = 182.9.$
$f_1(1) = 1,\; f_1(2) = 1;\; l_1 = 100,\; u_1 = 153.9.$ Hence, $u_1 > u_0$.

In the next lemma we show that the aperiodicity and irreducibility assumptions together imply that $\gamma > 0$, where

$$\gamma := min_{i,j \in S}\, min_{h_1,h_2,\dots,h_{N-1}}\, \{P(h_1)P(h_2)\cdots P(h_{N-1})\}_{ij}. \tag{6.12}$$

Lemma 6.6
$\gamma > 0$, where γ is defined in (6.12).

Proof
It is sufficient to show that $\{P(h_1)P(h_2)\cdots P(h_{N-1})\}_{ij} > 0$ for all $h_1, h_2, \dots, h_{N-1}$ and for all $i, j \in S$. Let $h_1, h_2, \dots, h_{N-1}$ be arbitrary decision rules. We define for $n = 0, 1, \dots, N-1$ and for $i \in S$:

$$S(i,0) := \{i\}; \text{ and } S(i,n) = \{j \in S \mid \{P(h_1)P(h_2) \cdot P(h_n)\}_{ij} > 0\},\; n = 1, 2, \dots, N-1.$$

Then, it has to be shown that $S(i, N-1) = S$ for all $i \in S$. We first show $S(i,n) \subseteq S(i,n+1)$. Let $j \in S(i,n)$, i.e. $\{P(h_1)P(h_2) \cdot P(h_n)\}_{ij} > 0$. Then, by the strong aperiodicity property, we have

$$\begin{array}{rcl} \{P(h_1)P(h_2)\cdots P(h_{n+1})\}_{ij} & = & \sum_k \{P(h_1)P(h_2)\cdots P(h_n)\}_{ik} P(h_{n+1})_{kj} \\ & \geq & \{P(h_1)P(h_2) \cdot P(h_n)\}_{ij} P(h_{n+1})_{jj} > 0. \end{array}$$

Hence, it remains to show that the sets $S(i,n)$ are strictly increasing in n as long as $S(i,n) \neq S$. Suppose $S(i,n+1) = S(i,n) \neq S$. Then, we have for all $j \in S(i,n)$ and $k \notin S(i,n)$ that $P(h_{n+1})_{jk} = 0$. Therefore, $S(i,n)$ is closed under $P(h_{n+1})$, which contradicts the irreducibility of the Markov chain $P(h_{n+1})$. □

The following lemma implies that the sequence l_n converges to the value ϕ exponentially fast.

Lemma 6.7
If $n, m \in \mathbb{N}$ satisfy $\sum_{i=0}^{m-1} k_{n+i} \geq N-1$, then $\phi - l_{n+m} \leq (1-\gamma)(\phi - l_n)$.

Proof
It follows from the proof of Lemma 6.5 that $g^{n+1} = Tx^{n+1} - x^{n+1} \geq P^{k_n}(f_n)\{Tx^n - x^n\} = P^{k_n}(f_n)g^n$. Consequently, for all $m = 1, 2, \dots,$ we have

$$g^{n+m} \geq P^{k_{n+m-1}}(f_{n+m-1})P^{k_{n+m-2}}(f_{n+m-2})\cdots P^{k_n}(f_n)g^n. \tag{6.13}$$

Let j_0 such that $u_n = g_{j_0}^n$. Then, for all $i \in S$ and all $h_1, h_2, \dots, h_{N-1}$, we can write

$$\begin{aligned}
\{P(h_1)P(h_2)\cdots P(h_{N-1})g^n\}_i &= \textstyle\sum_j \{P(h_1)P(h_2)\cdots P(h_{N-})\}_{ij}g_j^n \\
&= \textstyle\sum_{j\neq j_0} \{P(h_1)P(h_2)\cdots P(h_{N-1})\}_{ij}g_j^n + \{P(h_1)P(h_2)\cdots P(h_{N-1})\}_{ij_0}g_{j_0}^n \\
&= \textstyle\sum_{j\neq j_0} \{P(h_1)P(h_2)\cdots P(h_{N-1})\}_{ij}g_j^n + \{P(h_1)P(h_2)\cdots P(h_{N-1})\}_{ij_0}u_n \\
&\geq \textstyle\sum_{j\neq j_0} \{P(h_1)P(h_2)\cdots P(h_{N-1})\}_{ij}l_n + \{P(h_1)P(h_2)\cdots P(h_{N-1})\}_{ij_0}u_n \\
&= \big\{1-\{P(h_1)P(h_2)\cdots P(h_{N-1})\}_{ij_0}\big\}l_n + \{P(h_1)P(h_2)\cdots P(h_{N-1})\}_{ij_0}u_n \\
&= l_n + \{P(h_1)P(h_2)\cdots P(h_{N-1})\}_{ij_0}(u_n-l_n) \\
&\geq l_n + \gamma(u_n-l_n) = (1-\gamma)l_n + \gamma u_n \geq (1-\gamma)l_n + \gamma\phi,
\end{aligned}$$

the last inequality by Lemma 6.4. So, $P(h_1)P(h_2)\cdots P(h_{N-1})g^n \geq \{(1-\gamma)l_n+\gamma\phi\}\cdot e$.
Then, also for $k > N-1$ and all $h_1, h_2, \dots, h_k$,

$$\begin{aligned}
P(h_1)P(h_2)\cdots P(h_k)g^n &\geq \{P(h_1)P(h_2)\cdots P(h_{k-N+1})\}\{P(h_{k-N+2})P(h_{k-N+3})\cdots P(h_k)g^n\} \\
&\geq \{P(h_1)P(h_2)\cdots P(h_{k-N+1})\}\{(1-\gamma)l_n+\gamma\phi\}\cdot e \\
&= \{(1-\gamma)l_n+\gamma\phi\}\cdot e.
\end{aligned}$$

Hence, with (6.13), for all n, m such that $\sum_{i=0}^{m-1} k_{n+i} \geq N-1$,

$$g^{n+m} \geq P^{k_{n+m-1}}(f_{n+m-1})P^{k_{n+m-2}}(f_{n+m-2})\cdots P^{k_n}(f_n)g^n \geq \{(1-\gamma)l_n+\gamma\phi\}\cdot e.$$

Therefore, $l_{n+m} \geq (1-\gamma)l_n + \gamma\phi$, i.e. $\phi - l_{n+m} \leq (1-\gamma)(\phi - l_n)$. □

Since l_n converges to ϕ exponentially fast, f_n^∞ is ε-optimal for n sufficiently large. The problem, however, is to recognize how large n has to be. We next show that $u_n - l_n$ converges to 0, which provides the finiteness of Algorithm 6.3. Define δ by

$$\delta := min_{i,j\in S}\, min_{f^\infty\in C(D)}\, \{P^*(f)\}_{ij} > 0. \tag{6.14}$$

Lemma 6.8
$u_n - l_n \leq \frac{1}{\delta}(\phi - l_n)$ for all $n \in \mathbb{N}$.

Proof
$\phi\cdot e \geq \phi(f_n^\infty)\cdot e = P^*(f_n)\{rf_n) + P(f_n)x^n - x^n\} = P^*(f_n)\{Tx^n - x^n\} = P^*(f_n)g^n$.
Let j_0 such that $u_n = g_{j_0}^n$. Then, for all $i \in S$, we have

$$\begin{aligned}
\{P^*(f_n)g^n\}_i &= \textstyle\sum_{j\neq j_0} p_{ij}^*(f_n)g_j^n + p_{ij_0}^*(f_n)g_{j_0}^n \geq \sum_{j\neq j_0} p_{ij}^*(f_n)l_n + p_{ij_0}^*u_n \\
&= \{1-p_{ij_0}^*(f_n)\}l_n + p_{ij_0}^*u_n = l_n + p_{ij_0}^*(f_n)(u_n-l_n) \\
&\geq l_n + \delta(u_n-l_n) = (1-\delta)l_n + \delta u_n.
\end{aligned}$$

Hence, $P^*(f_n)g^n \geq \{(1-\delta)l_n+\delta u_n\}\cdot e$, and therefore, $\phi \geq (1-\delta)l_n + \delta u_n$, i.e. $u_n - l_n \leq \frac{1}{\delta}(\phi - l_n)$. □

Corollary 6.1
$u_n - l_n \to 0$ for $n \to \infty$.

Theorem 6.9
Algorithm 6.3 is correct.

Proof
Since $u_n - l_n$ converges to 0, the algorithm terminates. By Lemma 6.4, $l_n \leq \phi(f_n^\infty) \leq \phi \leq u_n$. Hence, if $u_n - l_n < \varepsilon$, then f_n^∞ is an ε-optimal policy. Furthermore, $|\phi - \frac{1}{2}(u_n + l_n)| < \frac{1}{2}\varepsilon$, i.e. $\frac{1}{2}(u_n + l_n)$ is a $\frac{1}{2}\varepsilon$-approximation of ϕ. □

6.2 The unichain case

Assumption 6.3
For every $f^\infty \in C(D)$ the Markov chain $P(f)$ has exactly one ergodic class plus a possibly empty set of transient states.

We have seen in section 5.2.3 that checking the unichain property is, in general, $\mathcal{NP}$-complete. Also in the unichain case, for every policy f^∞ the stationary matrix has identical rows, and consequently the vector $\phi(f)$ has identical components and we may $\phi(f)$ and the value vector ϕ consider as a scalar.

6.2.1 Optimality equation

We first argue that the result of Theorem 6.1 also holds in the unichain case. Following the proof of Theorem 6.1, we obtain part (1) and part (2) up to and including the statements $z - P(f_0)z \geq 0$ and $P^*(f_0)\{z - P(f_0)z\} = 0$. We have to give another proof that $z = P^*(f_0)z$, because the property that $P^*(f_0)$ has strictly positive elements doesn't hold in the unichain case. The columns of $P^*(f_0)$ are zero for the transient states and therefore the vector $P^*(f_0)z$ doesn't depend on the values of z_i for transient states i. Hence, we have to show that $z_j = \{P^*(f_0)z\}_j$ for the ergodic states j. But the states in this (only) ergodic class generate an irreducible Markov chain and the proof is similar as in Theorem 6.1.

Example 6.5
$S = \{1,2\};\ A(1) = \{1,2\},\ A(2) = \{1\};\ r_1(1) = 5,\ r_1(2) = 10,\ r_2(1) = -1.$
$p_{11}(1) = 0.5,\ p_{12}(1) = 0.5;\ p_{11}(2) = 0,\ p_{12}(2) = 1;\ p_{21}(1) = 0,\ p_{22}(1) = 1.$
It is obvious that this MDP is unichain and not irreducible (state 2 is absorbing and state 1 transient under all policies).

The optimality equation is: $\begin{cases} x + y_1 = max\{5 + 0.5y_1 + 0.5y_2, 10 + y_2\} \\ x + y_2 = -1 + y_2 \end{cases}$

From the second equation we obtain $x = -1$. For $y_2 = 0$ (y is unique up to a constant), the first equation becomes $-1 + y_1 = max\{5 + 0.5y_1, 10\}$. This equation has the solution $y_1 = 12$.

This model has two deterministic stationary policies: f_1^∞ and f_2^∞ with $f_1(1) = 1$ and $f_2(1) = 2$. Both policies are average optimal ($\phi_1(f_1^\infty) = \phi_1(f_2^\infty) = \phi_2(f_1^\infty) = \phi_2(f_2^\infty) = -1$). Furthermore, it is easy to verify that $u_1^0(f_1) = 12,\ u_1^0(f_2) = 11,\ u_2^0(f_1) = u_2^0(f_2) = 0$. Hence, $\big(\phi(f_1^\infty), u^0(f_1)\big)$ satisfies the optimality equation, but $\big(\phi(f_2^\infty), u^0(f_2)\big)$ does not.

For $y \in \mathbb{R}^N$ a decision rule g is called *y-improving* if $r(g) + P(g)y = max_f\ \{r(f) + P(f)y\}$.

Lemma 6.9
Let (ϕ, y) be a solution of the optimality equation (6.1) and let the decision rule g be y-improving. Then, g^∞ is an optimal policy.

Proof
From the optimality equation and the y-improving property of g it follows that

$$r(g) + P(g)y = max_f\ \{r(f) + P(f)y\} = \phi \cdot e + y.$$

By multiplying this equality by $P^*(g)$, we obtain $\phi(g^\infty) \cdot e = \phi \cdot e$, i.e. g^∞ is an optimal policy. □

From Example 6.5 it follows that the reverse statement (if g^∞ is an optimal policy, then g is y-improving) need not hold. The policy f_2^∞ is average optimal, but not y-maximizing, because we have $y = (12, 0)$, $r_1(f_2) + \sum_j p_{1j}(f_2)y = 10$ and $max_f\ \{r_1(f) + \sum_j p_{1j}(f)y\} = 11$.

6.2.2 Policy iteration

In this section, we also assume that every $P(f)$ is aperiodic. We have seen (Lemma 5.10) that this assumption is not an essential restriction. We will first derive some properties for a unichain and aperiodic Markov matrix P. Because of the aperiodicity, we have $P^* = \lim_{n\to\infty} P^N$. Let $B := P - P^*$. Then, using $P^* = PP^* = P^*P$, we obtain $B^n = P^n - P^*$ for $n \in \mathbb{N}$. Since $I - B^n = (I-B)(I + B + \cdots + B^{n-1})$ and $B^n \to 0$, it follows that $I - P + P^* = I - B$ is nonsingular and $Z := (I - P + P^*)^{-1} = \sum_{n=0}^{\infty} B^n$. Furthermore, we can write for $n \in \mathbb{N}$, using $Z = D + P^*,\ DP^* = 0$ and $DP^n = P^nD$,

$$\begin{aligned} \textstyle\sum_{i=0}^{n-1} P^n &= \textstyle\sum_{i=0}^{n-1} B^n + (n-1)P^* \\ &= (I-B)^{-1}(I-B^n) + (n-1)P^* = Z(I-B^n) + (n-1)P^* \\ &= (D+P^*)(I-P^n+P^*) + (n-1)P^* = D - P^nD + nP^*. \end{aligned}$$

Hence, we obtain for the total reward after n periods for any $f^\infty \in C(D)$,

$$v^n(f^\infty) = \textstyle\sum_{i=0}^{n-1} P^n(f)r(f) = n \cdot \phi(f^\infty) + u^0(f) - P^n(f)u^0(f).$$

Note that $P^n(f)u^0(f) \to P^*(f)u^0(f) = P^*(f)D(f)r(f) = 0$ as $n \to \infty$.

Next, we will show that the computation of P^* and D can be simplified if P is an aperiodic unichain Markov chain. P^* has identical rows, denoted by π, which is the unique solution of the linear system $\begin{cases} x^Te &= 1; \\ x^T(I-P) &= 0 \end{cases}$ This system has $N+1$ equations and N unknowns. Since it specifies x uniquely, it must contain exactly one redundancy. Since the columns of $I-P$ sum up to zero, i.e. $(I-P)e = 0$, this redundancy is in any of the last N rows of the system. We will delete the second row of this system, which corresponds to state 1, i.e. we delete the equation $x^Ty = 0$, where y is the first column of $I-P$. Let C be the $N \times N$-matrix obtained by replacing the first column of $I-P$ by e. Then, the system becomes $x^TC = e^1$, where e^1 is the first unit vector, Hence, $\pi = (e^1)^TC^{-1}$, i.e. π is the first row of C^{-1}.

The deviation matrix D can be found by inverting the matrix $(I - P + P^*)$. We will show that the inversion is superfluous. Consider the columns of the matrix $F = DC$. The first column of C is e. Hence, the first column of F is $De = 0$, i.e. the first column of F is the zero column. The other columns of C are the columns 2 through N of $I-P$. Hence, the columns 2 through N of F are the corresponding columns of $D(I-P) = I - P^*$. Therefore, D is the $N \times N$-matrix obtained from $I - P^*$ by replacing the first column by the zero column. Since $D = FC^{-1}$, after the computation of C^{-1} and $\pi := (e^1)C^{-1}$, the deviation matrix D can be obtained without a new inversion.

Example 6.6

Let $P = \begin{pmatrix} \frac{1}{2} & \frac{1}{2} & 0 \\ 1 & 0 & 0 \\ 1 & 0 & 0 \end{pmatrix}$. Then, $C = \begin{pmatrix} 1 & -\frac{1}{2} & 0 \\ 1 & 1 & 0 \\ 1 & 0 & 1 \end{pmatrix}$ with inverse matrix $C^{-1} = \begin{pmatrix} \frac{2}{3} & \frac{1}{3} & 0 \\ -\frac{2}{3} & \frac{2}{3} & 0 \\ -\frac{2}{3} & -\frac{1}{3} & 1 \end{pmatrix}$.

Hence, $\pi = (\frac{2}{3}, \frac{1}{3}, 0)^T$.

$$I - P^* = \begin{pmatrix} \frac{1}{3} & -\frac{1}{3} & 0 \\ -\frac{2}{3} & \frac{2}{3} & 0 \\ -\frac{2}{3} & -\frac{1}{3} & 1 \end{pmatrix} \ \to\ D = FC^{-1} = \begin{pmatrix} 0 & -\frac{1}{3} & 0 \\ 0 & \frac{2}{3} & 0 \\ 0 & -\frac{1}{3} & 1 \end{pmatrix} \begin{pmatrix} \frac{2}{3} & \frac{1}{3} & 0 \\ -\frac{2}{3} & \frac{2}{3} & 0 \\ -\frac{2}{3} & -\frac{1}{3} & 1 \end{pmatrix} = \begin{pmatrix} \frac{2}{9} & -\frac{2}{9} & 0 \\ -\frac{4}{9} & \frac{4}{9} & 0 \\ -\frac{4}{9} & -\frac{5}{9} & 1 \end{pmatrix}.$$

Having obtained $P^*(f)$ and $D(f)$ by the above computation scheme, we can determine $\phi(f^\infty$ and $u^0(f)$ by postmultiplication these matrices with $r(f)$. We also shall develop a separate equation from which $\phi(f^\infty$ and $u^0(f)$ can be computed. Note that $\phi(f^\infty)$ may be considered as a scalar. Consider the following system of N equation in $N+1$ unknowns:

$$x \cdot e + \{I - P(f)\}y = r(f). \tag{6.15}$$

We have seen in the proof of Theorem 6.2 that if (x, y) is a solution of (6.15), then $x = \phi(f^\infty)$ and y satisfies

$$y = u^0(f) + P^*(f)y = u^0(f) + c \cdot e, \tag{6.16}$$

with c any scalar. Two solutions of (6.15) are of particular interest. The solution in with $c = -u^0_1(f)$, i.e. $y_1 = 0$, reduces (6.15) to C(f)z = r(f), where $z_1 = x$ and $z_i = y_i,\ 2 \leq i \leq N$. Hence, $z = \{C(f)\}^{-1}r(f)$. For the second particular solution we choose $c = 0$, i.e. $y = u^0(f) = D(f)r(f)$. This is the unique solution of the system $\left\{ \begin{array}{rcl} x \cdot e + \{I - P(f)\}y & = & r(f); \\ \pi(f)^T y & = & 0. \end{array} \right.$.

Example 6.6 (continued)
Let $r = (0, 3, 4)^T$.
For the first particular solution: $z = \{C(f)\}^{-1}r(f) = (1, 2, 3)^T$, i.e. $\phi = 1$ and $y_1 = 0,\ y_2 = 2,\ y_3 = 3$.

For the second particular solution, we solve $\begin{cases} x + \frac{1}{2}y_1 - \frac{1}{2}y_2 = 0 \\ x - y_1 + y_2 = 3 \\ x - y_1 + y_3 = 4 \\ \frac{2}{3}y_1 + \frac{1}{3}y_2 = 0 \end{cases}$

The unique solution of this system is: $x = \phi = 1$ and $y_1 = -\frac{2}{3},\ y_2 = \frac{4}{3},\ y_3 = \frac{7}{3}$.

The policy iteration method for unichain MDPs is similar to the policy iteration method in the irreducible case, but the proof of finiteness is different. In the irreducible case for subsequent policies the average reward increases strictly. This is not true in the unichain case, where we have increasing in the following lexicographic sense: either the average reward increases strictly or there is no decrease in the average reward, but there is a strict increase in the bias term u^0. We will discuss the version of Algorithm 6.1 in which the 'best' improving actions are taken.

Algorithm 6.4 *Determination of an average optimal policy by policy iteration (unichain case)*
Input: Instance of a unichain MDP.
Output: An average optimal deterministic policy f^∞ and the value ϕ.

```
1. Select an arbitrary $f^\infty \in C(D)$.
2. Determine a solution $(x = \phi(f^\infty), y)$ of the system $x \cdot e + \{I - P(f)\}y = r(f)$.
3. for every $i \in S$ do $B(i, f) := \{a \in A(i) \mid r_i(a) + \sum_j p_{ij}(a)y_j > \phi(f^\infty) + y_i\}$.
4. if $B(i, f) = \emptyset$ for every $i \in S$ then
       begin $f^\infty$ is an average optimal policy; $\phi(f^\infty)$ is the value $\phi$ (STOP) end
   else begin select $g$ such that $r_i(g) + \sum_j p_{ij}(g)y_j = max_a\{r_i(a) + \sum_j p_{ij}(a)y_j\},\ i \in S$,
           taking $g(i) = f(i)$ if possible; $f := g$; return to step 2
       end
```

Remark
The determination of the solution $(x = \phi(f^\infty), y)$ in step 2 of Algorithm 6.4 can be found by inverting $C(f)$ and computing $C^{-1}(f)r(f)$. In successive iterations the matrices requiring inversion may differ by only one or a few rows. When this occurs, it will be more efficient to update the old inversion than to reinvert the whole matrix.

Theorem 6.10

Let g^∞ be the policy obtained in step 4 of Algorithm 6.4 and let $t := r(g) - \{I - P(g)\}y - \phi(f^\infty) \cdot e$. Then,
(1) $\phi(g^\infty) \geq \phi(f^\infty)$.
(2) $\phi(g^\infty) = \phi(f^\infty)$ if and only if $\pi(g)^T t = 0$.
(3) If $\pi(g)^T t = 0$, then $\pi(g) = \pi(f)$ and $u^0(g) \geq u^0(f) + t > u^0(f)$.

Proof

(1) Since, by the definition of g^∞ in step 4 of Algorithm 6.4, $t \geq 0$, we can write

$0 \leq \pi(g)^T t = \pi(g)^T\{r(g) - \{I - P(g)\}y - \phi(f^\infty) \cdot e\} = \phi(g^\infty) - \phi(f^\infty)$, i.e. $\phi(g^\infty) \geq \phi(f^\infty)$.

(2) From the proof of part (1) it follows that $\phi(g^\infty) = \phi(f^\infty)$ if and only if $\pi(g)^T t = 0$.

(3) If $g(i) = f(i)$, then $p_{ij}(g) = p_{ij}(f)$ for all $j \in S$, implying $\pi_i(g)\{p_{ij}(g) - p_{ij}(f)\} = 0$ for all $j \in S$. If $g(i) \neq f(i)$, then $t_i > 0$. Because $\pi(g)^T t = 0$, $\pi_i(g) t_i = 0$ for all $i \in S$. Hence, if $g(i) \neq f(i)$, then $\pi_i(g) = 0$, and consequently $\pi_i(g)\{p_{ij}(g) - p_{ij}(f)\} = 0$ for all $j \in S$. Therefore, $\pi(g)\{P(g) - P(f)\} = 0$. Hence, $\pi(g)^T = \pi(g)^T P(g) = \pi(g)^T P(f)$ and $\pi(g)^T e = 1$, so $\pi(g)$ is the stationary distribution of $P(f)$, i.e. $\pi(g) = \pi(f)$. Furthermore, we obtain

$$\begin{aligned} D(g)t &= D(g)\big\{r(g) - \{I - P(g)\}y - \phi(f^\infty) \cdot e\big\} = u^0(g) - \{I - P^*(g)\}y - \phi(f^\infty) \cdot D(g)e \\ &= u^0(g) - \{I - P^*(f)\}y = u^0(g) - u^0(f), \end{aligned}$$

the last equality by (6.16). Since, $D(g) = \sum_{n=0}^\infty \{P^n(g) - P^*(g)\}$ and $\pi(g)^t = 0$, we can write $D(g)t = \sum_{n=0}^\infty \{P^n(g) - P^*(g)\}t = \sum_{n=0}^\infty P^n(g)t \geq t > 0$.

Hence, we have $u^0(g) - u^0(f) = D(g)t > t > 0$. □

Theorem 6.11

Algorithm 6.4 terminates in finitely many iteration with an average optimal policy and the value.

Proof

Theorem 6.10 guarantees that the average reward is increasing in each iteration in which an action change occurs in state i that is recurrent under the new policy g^∞, namely: in that case $\pi_i(g) > 0$ and $t_i > 0$, so by part (1) and (2) of Theorem 6.10, we have $\phi(g^\infty) > \phi(f^\infty)$.

If action changes only occur in states that are transient under the new policy g^∞, then $\pi_i(g) = 0$ for all i with $t_i > 0$, and consequently, $\pi(g)^T t = 0$. Then, by part (3) of Theorem 6.10, we obtain $\pi(g) = \pi(f)$ and $u^0(g) \geq u^0(f) + t > u^0(f)$. Hence, we have shown that Algorithm 6.4 terminates in finitely many iteration.

At termination, $B(i, f) = \emptyset$ for every $i \in S$. Therefore, $r(h) + P(h)y \leq \phi(f^\infty) \cdot e + y$ for every $h^\infty \in C(D)$. Hence, we have $\phi(h^\infty) \cdot e = P^*(h)r(h) \leq \phi(f^\infty) \cdot e$ for every $h^\infty \in C(D)$, i.e. f^∞ is an average optimal policy and $\phi(f^\infty)$ is the value ϕ. □

Example 6.5 (continued)

We apply Algorithm 6.4 to this model starting with $f(1) = 2,\ f(2) = 1$. In step 2 we will choose y such that $y_1 = 0$.

Iteration 1

Consider the system $\left\{ \begin{array}{lllllll} x & + & y_1 & - & y_2 & = & 10 \\ x & & & & & = & -1 \\ & & y_1 & & & = & 0 \end{array} \right. \rightarrow x = \phi(f^\infty) = -1,\ y_1 = 0,\ y_2 = -11.$

$B(1, f) = \{1\},\ B(2, f\} = \emptyset.\ g(1) = 1,\ g(2) = 1;\ f(1) = 1,\ f(2) = 1.$

Iteration 2

Consider the system $\left\{\begin{array}{ccccccc} x & + & 0.5y_1 & - & 0.5y_2 & = & 5 \\ x & & & & & = & -1 \\ & & y_1 & & & = & 0 \end{array}\right. \rightarrow x = \phi(f^\infty) = -1,\ y_1 = 0,\ y_2 = -12.$

$B(1,f) = B(2,f\} = \emptyset : f^\infty$ is an average optimal policy and -1 is the value.

Remark

The above example shows that after determining a policy with optimal average reward (iteration 1), policy iteration goes on to find one with optimal bias (iteration 2 ends with $u^0(f)^T = (12,0)$ which is the optimal bias term). One might conjecture that Algorithm 6.4 always find a bias-optimal policy. The following example shows that this supposition is false.

Example 6.7

$S = \{1,2\};\ A(1) = \{1,2\},\ A(2) = \{1\};\ r_1(1) = 4,\ r_1(2) = 0,\ r_2(1) = 8.$

$p_{11}(1) = 1,\ p_{12}(1) = 0;\ p_{11}(2) = 0,\ p_{12}(2) = 1;\ p_{21}(1) = 1,\ p_{22}(1) = 0.$

It is obvious that this MDP is unichain and not irreducible.

We apply Algorithm 6.4 to this model starting with $f(1) = 2,\ f(2) = 1$. In step 2 we will choose y such that $y_1 = 0$.

Iteration 1

Consider the system $\left\{\begin{array}{ccccccc} x & + & y_1 & - & y_2 & = & 0 \\ x & - & y_1 & + & y_2 & = & 8 \\ & & y_1 & & & = & 0 \end{array}\right. \rightarrow x = \phi(f^\infty) = 4,\ y_1 = 0,\ y_2 = 4.$

$B(1,f) = B(2,f\} = \emptyset$. Hence, the algorithm terminates with f^∞ as average optimal policy and with 4 as the value. It is easily to verify that $u^0(f)^T = (-2,2)$.

Let g denote the decision rule which uses action 1 in state 1. Then, $\phi(g^\infty) = 4$ and $u^0(g)^T = (0,4)$, so g^∞ and not f^∞ is a bias-optimal policy.

A new policy iteration scheme

We will provide a new rule for generating a policy g^∞ from f^∞ such that in each iteration:

(1) $g(i) \neq f(i)$ only in one state i, say $i = j$;

(2) $\phi(g^\infty) - \phi(f^\infty)$ is as positive as possible, subject to condition (1).

This approach may be viewed as a kind of linear programming where we make the change of basis not on the *rate* change of the objective function, but rather on the *entire* change, i.e. (rate of change) × (step size). Start with policy f^∞ and fix a pair (j,a) such that $a \neq f(j)$. Let g^∞ be the policy with $g(i) := \left\{\begin{array}{ll} f(i) & i \neq j; \\ a & i = j. \end{array}\right.$ Notice that $r_i(g) = r_i(f)$ and $p_{ik}(g) = p_{ik}(f),\ k \in S$ for every $i \neq j$. We wish to choose the pair (j,a) to maximize $\phi(g^\infty) - \phi(f^\infty)$. From Lemma 6.2, we obtain

$$\phi(g^\infty)-\phi(f^\infty) = \pi(g)^T\{r(g)-r(f)+[P(g)-P(f)]y(f)\} = \pi_j(g)\{r_j(g)-r_j(f)+\sum_k [p_{jk}(g)-p_{jk}(f)]y_k(f)\}, \tag{6.17}$$

where $y(f)$ is any solution of $\{I - P(f)\}y = r(f) - \phi(f^\infty)\cdot e$. Let $B^{j,a}(f) := \{P(g) - P(f)\}Z(f)$.

Then, by Lemma 6.3, $\pi(f)^T = \pi(g)^T\big\{I - B^{j,a}(f)\big\}$, and consequently, $\pi_j(f) = \pi_j(g)\big\{1 - [B^{j,a}(f)]_{jj}\big\}$, i.e. $\pi_j(g) = \pi_j(f)\big\{1 - [B^{j,a}(f)]_{jj}\big\}^{-1}$. Therefore, we have

$$\phi(g^\infty) - \phi(f^\infty) = \pi_j(f)\big\{1 - [B^{j,a}(f)]_{jj}\big\}^{-1}\big\{r_j(g) - r_j(f) + \sum_k [p_{jk}(g) - p_{jk}(f)]y_k(f)\big\}. \tag{6.18}$$

Note that $Z(f)r(f) = D(f)r(f) + P^*(f)r(f) = u^0(f) + \phi(f^\infty) \cdot e = y(f) + c \cdot e$ for some scalar c. Hence, we have

$$\phi(g^\infty) - \phi(f^\infty) = \pi_j(f)\big\{1 - [B^{j,a}(f)]_{jj}\big\}^{-1}\big\{r_j(g) - r_j(f) + \sum_k [p_{jk}(g) - p_{jk}(f)]\{Z(f)r(f)\}_k\big\}. \quad (6.19)$$

Thus, maximizing $\phi(g^\infty) - \phi(f^\infty)$ means choosing the pair (j, a) to maximize the right-hand-side of (6.18) rather than maximizing the term $r_j(g) - r_j(f) + \sum_k [p_{jk}(g) - p_{jk}(f)]\{Z(f)r(f)\}_k$ as is usually done in normal pivoting in the simplex method. If the matrix $Z(f)$ is available, this is a minor increase in computational effort. Updating $Z(f)$ can be done efficiently by using the following lemma.

Lemma 6.10
$Z(g) = \big\{I - [P^*(g) - P^*(f)]\big\}Z(f)\big\{I - [P(g) - P(f)]Z(f)\big\}^{-1}$.

Proof
We will show that $Z(g)\big\{I - [P(g) - P(f)]Z(f)\big\} = \big\{I - [P^*(g) - P^*(f)]\big\}Z(f)$. Using the properties of the fundamental, the stationary and the deviation matrix ($Z = D + P^*,\ DP = PD = D + P^* - I,\ P^*Z = P^*$ and $PZ = ZP = D + 2P^* - I$), we can present the following deduction:

$$Z(g)\big\{I - [P(g) - P(f)]Z(f)\big\} = Z(g) - Z(g)P(g)Z(f) + Z(g)P(f)Z(f) =$$
$$\{D(g) + P^*(g)\} - \{D(g) + 2P^*(g) - I\}\{D(f) + P^*(f)\} + \{D(g) + P^*(g)\}\{D(f) + 2P^*(f) - I\} =$$
$$D(g) + P^*(g) - D(g)D(f) - D(g)P^*(f) - 2P^*(g)D(f) - 2P^*(g)P^*(f) + D(f) + P^*(f) + D(g)D(f) +$$
$$2D(g)P^*(f) - D(g) + P^*(g)D(f) + 2P^*(g)P^*(f) - P^*(g) =$$
$$D(g)P^*(f) - P^*(g)D(f) + D(f) + P^*(f) = -P^*(g)D(f) + D(f) + P^*(f),$$

the last equality because $D(g)P^*(f) = D(g)e\pi(f)^T = 0$. On the other hand, we have

$$\big\{I - [P^*(g) - P^*(f)]\big\}Z(f) = \big\{I - [P^*(g) - P^*(f)]\big\}\{D(f) + P^*(f)\} =$$
$$\{D(f) + P^*(f)\} - P^*(g)\{D(f) + P^*(f)\} + P^*(f)\{D(f) + P^*(f)\} =$$
$$D(f) + P^*(f) - P^*(g)D(f) - P^*(g)P^*(f) + P^*(f)D(f) + P^*(f)P^*(f) =$$
$$D(f) + P^*(f) - P^*(g)D(f) - P^*(g)P^*(f) + P^*(f) = D(f) + P^*(f) - P^*(g)D(f),$$

the last equality because $P^*(g)P^*(f) = e\pi(g)^T e\pi(f)^T = e\pi(f)^T = P^*(f)$.

Hence, we have completed the proof that $Z(g) = \big\{I - [P^*(g) - P^*(f)]\big\}Z(f)\big\{I - [P(g) - P(f)]Z(f)\big\}^{-1}$.□

Algorithm 6.5 *Determination of an average optimal policy by policy iteration (version 2; unichain case)*
Input: Instance of a unichain MDP.
Output: An optimal deterministic policy f^∞ and the value ϕ.

```
1. Select an arbitrary $f^\infty \in C(D)$; compute $\pi(f)$; compute $Z(f)$; compute $s(f) := Z(f)r(f)$.

2. for every pair $(j, a) \in S \times A$ with $a \neq f(j)$ do
   begin for every $i \in S$ do $g(i) := \begin{cases} f(i) & i \neq j \\ a & i = j \end{cases}$;
       $B := [P(g) - P(f)]Z(f);\ \pi(g)^T := \pi(f)^T\{I - B\}^{-1}$;
       $\Delta(j, a) := \pi_j(f)\{1 - b_{jj}\}^{-1}\{r_j(g) - r_j(f) + \sum_k [p_{jk}(g) - p_{jk}(f)]s_k(f)\}$
   end
```

```
3. if $\Delta(j,a) \leq 0$ for all $(j,a)$ then
        begin $f^\infty$ is an average optimal policy; $\pi(f)^T r(f)$ is the value (STOP) end
   else begin determine $(j_*, a_*)$ such that $\Delta(j_*, a_*) = max_{(j,a)}\, \Delta(j,a)$;
            for every $i \in S$ do $g(i) := \begin{cases} f(i) & i \neq j_* \\ a_* & i = j_* \end{cases}$;
            $B := [P(g) - P(f)]Z(f);\ \pi(g)^T := \pi(f)^T\{I - B\}^{-1}$;
            $Z(g) = \big\{I - [P^*(g) - P^*(f)]\big\}Z(f)\{I - B\}^{-1};\ s(g) := Z(g)r(g)$;
            $f := g$; return to step 2
        end
```

Remarks

1. In step 3, in the case there is not yet termination, the computation of B and $\pi(g)$ is in fact superfluously. We have already computed these values in step 2. Hence, in step 2 we can keep these value for the most positive $\Delta(j,a)$.
2. We can evaluate the computational complexity involved in each iteration of Algorithm 6.5. Assume that $|A(i)| \leq M$ for every $i \in S$.
   (a) In each iteration we have to update $P(f), \pi(f), Z(f)$ and $s(f)$ once, which have complexity N^2 at most.
   (b) For each pair (j,a) we have to compute B and $\{I - B\}^{-1}$. Since every row of $P(g) - P(f)$, except row j, is the zero row, the computation of B and $\{I - B\}^{-1}$ are of order N.
   (c) For each pair (j,a) we have to compute $\Delta(j,a)$ which is also of order N.
   Hence, the computational complexity of each iteration is of order $N^2 \cdot M$. Notice that one iteration of the linear program (6.3) also needs order $N^2 \cdot M$ elementary operations.

Example 6.8
$S = \{1,2\};\ A(1) = \{1,2\},\ A(2) = \{1\};\ r_1(1) = 1,\ r_1(2) = 4,\ r_2(1) = 0.$
$p_{11}(1) = 1,\ p_{12}(1) = 0;\ p_{11}(2) = 0,\ p_{12}(2) = 1;\ p_{21}(1) = 1,\ p_{22}(1) = 0.$
This MDP is unichain and not irreducible. We apply Algorithm 6.5 and start with $f(1) = f(2) = 1$.
Iteration 1
1. $P(f) = \left(\begin{smallmatrix} 1 & 0 \\ 1 & 0 \end{smallmatrix}\right);\ \pi(f)^T = (1,0);\ Z(f) = \{I - P(f) + P^*(f)\}^{-1} = I;\ s(f) = (1,0)^T.$
2. $(j,a) = (1,2);\ P(g) = \left(\begin{smallmatrix} 0 & 1 \\ 1 & 0 \end{smallmatrix}\right);\ B = \left(\begin{smallmatrix} -1 & 1 \\ 0 & 0 \end{smallmatrix}\right);\ \{I - B\}^{-1} = \left(\begin{smallmatrix} 0.5 & 0.5 \\ 0 & 1 \end{smallmatrix}\right);\ \pi(g)^T = (0.5, 0.5);\ \Delta(j,a) = 1.$
3. $(j_*, a_*) = (1,2);\ Z(g) = \big\{I - [P^*(g) - P^*(f)]\big\}Z(f)\{I - B\}^{-1} = \left(\begin{smallmatrix} 1.5 & -0.5 \\ 0.5 & 0.5 \end{smallmatrix}\right) \cdot I \cdot \left(\begin{smallmatrix} 0.5 & 0.5 \\ 0 & 1 \end{smallmatrix}\right) = \left(\begin{smallmatrix} 0.75 & 0.25 \\ 0.25 & 0.75 \end{smallmatrix}\right);$
   $f(1) = 2,\ f(2) = 1.$

Iteration 2
2. $(j,a) = (1,1);\ P(g) = \left(\begin{smallmatrix} 1 & 0 \\ 1 & 0 \end{smallmatrix}\right);\ B = \left(\begin{smallmatrix} 0.5 & -0.5 \\ 0 & 0 \end{smallmatrix}\right);\ \{I - B\}^{-1} = \left(\begin{smallmatrix} 2 & -1 \\ 0 & 1 \end{smallmatrix}\right);\ \pi(g)^T = (1,0);\ \Delta(j,a) = -1.$
3. f^∞ with $f(1) = 2$ and $f(2) = 1$ is an optimal policy and $\pi(f)^T r(f) = 2$ is the value.

6.2.3 Linear programming

In the unichain case the same linear program can be used as in the irreducible case, but the result is slightly different. The value ϕ is again the unique x-part of program (6.3), but we loose the property that every feasible solution x of the dual program (6.4) satisfies $\sum_a x_i(a) > 0,\ i \in S$. It turns out that this property can only be shown for recurrent states. However, since there is only one recurrent set, the actions in transient states doesn't influence the average reward for that states. The following theorem shows the result.

Theorem 6.12

Let (ϕ, y^*) and x^* be optimal solutions of the linear programs (6.3) and (6.4), respectively. Define f_*^∞ such that for every $i \in S$ $x_i^*\big(f_*(i)\big) > 0$ if $\sum_a x_i^*(a) > 0$ and $f_*(i)$ is an arbitrary action if $\sum_a x_i^*(a) = 0$. Then, f_*^∞ is an average optimal policy.

Proof

Suppose that $\sum_a x_j^*(a) = 0$ for some j. The constraints of (6.4) imply $0 = \sum_a x_j^*(a) = \sum_{i,a} p_{ij}(a)x_i^*(a)$. Hence, $p_{ij}(a)x_i^*(a) = 0$ for all $(i,a) \in S \times A$. Therefore, in states i with $\sum_a x_i^*(a) > 0$ we have $p_{ij}(f_*) = 0$, i.e. the set $S_{x^*} := \{i \mid \sum_a x_i^*(a) > 0\}$ is closed in the Markov chain $P(f_*)$. Since this Markov chain has only one ergodic set, the states $S \backslash S_{x^*}$ are transient under $P(f_*)$. From the orthogonality property of linear programming it follows that:

$$x_i^*(a) \cdot \big\{\phi + \textstyle\sum_j \{\delta_{ij} - p_{ij}(a)\}y_j^* - r_i(a)\big\} = 0 \text{ for all } (i,a) \in S \times A.$$

Hence, $\phi + \{\big(I - P(f_*)\big)y^*\}_i - r_i(f_*) = 0$ for all $i \in S_{x^*}$. Multiply $\phi \cdot e + \{I - P(f_*)y^*\} - r(f_*)$ by $P^*(f_*)$ and notice that the columns of $P^*(f_*)$ are zeros for the states in $S \backslash S_{x^*}$, because these states are transient. Then, we obtain $0 = \phi \cdot e - P^*(f_*)r(f_*) = \phi \cdot e - \phi(f_*^\infty) \cdot e$, implying that f_*^∞ is an optimal policy. □

Algorithm 6.6

Determination of an average optimal policy by linear programming (unichain case)

Input: Instance of a unichain MDP.

Output: An optimal deterministic policy f^∞ and the value ϕ.

1. Determine an optimal solution x^* of the linear program (6.4).
2. $S_{x^*} := \{i \mid \sum_a x_i^*(a) > 0\}$.
3. Select any $f_*^\infty \in C(D)$ such that $x_i^*\big(f_*(i)\big) > 0$ for every $i \in S_{x^*}$.
4. The value ϕ is the optimum value of program (6.4) and f_*^∞ is an optimal policy (STOP).

Example 6.5 (continued)

The linear program (6.4) for this model becomes

$max\{5x_1(1) + 10x_1(2) - x_2(1)\}$

subject to

$x_1(1) + x_1(2) = \frac{1}{2}x_1(1);\ x_2(1) = \frac{1}{2}x_1(1) + x_1(2) + x_2(1);$

$x_1(1) + x_1(2) + x_2(1) = 1;\ x_1(1), x_1(2), x_2(1) \geq 0.$

The optimal optimal solution is $x_1^*(1) = x_1^*(2) = 0,\ x_2^*(1) = 1$; optimum = -1. Therefore, in state 1 any action can be chosen and the two deterministic policies are both optimal. From this examples it also follows that in the unichain case there is no one-to-one correspondence between policy iteration and linear programming. Furthermore, that there is no one-to-one correspondence between the feasible solutions of the dual program and stationary policies: the optimal solution x^* corresponds to the two deterministic policies.

Solving the general case by a sequence of unichain linear programs

The linear programs (6.3) and (6.4) can also be used to find an average optimal policy regardless the chain structure. The decision maker needs not to know in advance whether the MDP is irreducible, unichained or multichained. After solving the linear programs (6.3) and (6.4), we need some additional procedures in the multichain case, without knowing in advance that the model is multichained. So, assume that the MDP has an arbitrary chain structure and let x be a basic feasible solution of (6.4). Define $S_x := \{i \mid \sum_a x_i(a) > 0\}$ and $S_x \times A_x := \{(i,a) \mid x_i(a) > 0\}$. We say that x *identifies a unique ergodic chain* if the sets S_x and $S_x \times A_x$ have the same number of elements, i.e. for any $i \in S_x,\ x_i(a) > 0$ for exactly one action $a \in A(i)$, say $a = f_x(i)$, and if under $P(f_x)$ the set S_x is closed and all states in S_x communicate. The next lemma shows that any basic feasible solution x of (6.4) identifies a unique ergodic chain.

Lemma 6.11

Let x be a basic feasible solution of linear program (6.4). Then, x identifies a unique ergodic chain and $\sum_{i,a} r_i(a)x_i(a)$ is the average reward of this chain.

Proof

Choose any policy f^∞ such that $x_i\big(f(i)\big) > 0,\ i \in S_x$. First, we show that S_x is closed under $P(f)$. Suppose $p_{kl}(f) > 0$ for some $k \in S_x$ and some $l \notin S_x$. From the constraints of (6.4) it follows that $0 = \sum_a x_l(a) = \sum_{i,a} p_{il}(a)x_i(a) \geq p_{kl}(f)x_k\big(f(k)\big) > 0$, implying a contradiction. We also have,

$$\begin{cases} 0 &= \sum_{i,a} \{\delta_{ij} - p_{ij}(a)\}x_i(a) = \sum_{(i,a)\in S_x\times A_x} \{\delta_{ij} - p_{ij}(a)\}x_i(a),\ j \in S \\ 1 &= \sum_{i,a} x_i(a) = \sum_{(i,a)\in S_x\times A_x} x_i(a) \end{cases} \tag{6.20}$$

Since S_x is closed under $P(f)$, S_x contains at least one ergodic set $S_1 \subseteq S_x$. Let z be the stationary distribution of $P(f)$, restricted to the states of S_1. Then,

$$\begin{cases} 0 &= \sum_{i\in S_1} \{\delta_{ij} - p_{ij}(f)\}z_i,\ j \in S_1 \\ 1 &= \sum_{i\in S_1} z_i \end{cases} \tag{6.21}$$

Let $S_1 \times A_1 := \{(i,a) \mid i \in S_1,\ a = f(i)\}$. Subtracting (6.21) from (6.21) yields

$$\begin{cases} 0 &= \sum_{(i,a)\in(S_x\times A_x)\backslash(S_1\times A_1)} \{\delta_{ij} - p_{ij}(a)\}x_i(a) + \sum_{((i,a)\in S_1\times A_1} \{\delta_{ij} - p_{ij}(a)\}\{x_i(a) - z_i\},\ j \in S \\ 0 &= \sum_{(i,a)\in(S_x\times A_x)\backslash(S_1\times A_1)} x_i(a) + \sum_{((i,a)\in S_1\times A_1} \{x_i(a) - z_i\} \end{cases} \tag{6.22}$$

Since x is a basic solution of (6.4), the columns of (6.4), corresponding to the positive x-variables, i.e. the $N+1$-dimensional columns $\begin{pmatrix} \delta_{ij} - p_{ij}(a),\ j \in S \\ 1 \end{pmatrix}$, $(i,a) \in S_x \times A_x$, are linear independent. Hence, the corresponding coefficients in (6.22) are zero. So, $S_x \times A_x = S_1 \times A_1$ and $x_i(a) = z_i,\ (i,a) \in S_1 \times A_1$. Consequently, the sets $S_x = S_1$ and $S_x \times A_x = S_1 \times A_1$ have the same number of elements and, under $P(f)$, $S_x = S_1$ is an ergodic set. Furthermore, we have $\sum_{i,a} r_i(a)x_i(a) = \sum_{i\in S_1} r_i(f)z_i = \phi(f^\infty)$, the average reward of the chain. □

Theorem 6.13

Let x^* be an extreme optimal solution of linear program (6.4). Then, x^* identifies a unique ergodic chain and $\sum_{i,a} r_i(a)x^*_i(a)$ is the maximum average reward of all chain of the MDP.

Proof

Consider the chain $C \subseteq S$ with the maximum reward $r(C)$ of all chains. Let f^∞ a deterministic policy that takes actions according to this chain and let x such that x satisfies on C the system $\begin{cases} x^TP(f) &= x^T \\ x^Te &= 1 \end{cases}$

Setting all other variables zero produces a feasible solution to program (6.4) that has $r(C)$ as value of the objective function. Hence, $r(C)$ is at most the optimum value of program (6.4).

Conversely, by Lemma 6.11, every basic feasible solution x to program (6.4) identifies a unique ergodic chain and $\sum_{i,a} r_i(a)x_i(a)$ is the average reward of this chain. Since program (6.4) has an optimal solution in one of its extreme feasible solutions, the theorem is proven. □

Example 5.9 (continued)
The linear program (6.4) for this model becomes
$max\{x_1(1) + 2x_1(2) + 3x_1(3) + 6x_2(1) + 4x_2(2) + 5x_2(3) + 8x_3(1) + 9x_3(2) + 7x_3(3) +\}$
subject to the constraints (without the nonnegativity constraints)

$$\begin{array}{rrrrrrrrrcl}
 & x_1(2) & + \ x_1(3) & - \ x_2(1) & & & - \ x_3(1) & & & = & 0 \\
 & - \ x_1(2) & & + \ x_2(1) & & + \ x_2(3) & & - \ x_3(2) & & = & 0 \\
 & & - \ x_1(3) & + \ x_2(1) & & - \ x_2(3) & - \ x_3(1) & + \ x_3(2) & & = & 0 \\
x_1(1) & + \ x_1(2) & + \ x_1(3) & + \ x_2(1) & + \ x_2(2) & + \ x_2(3) & + \ x_3(1) & + \ x_3(2) & + \ x_3(3) & = & 1
\end{array}$$

An extreme optimal solution of this program is x with $x_1(1) = x_1(2) = x_1(3) = x_2(1) = x_2(2) = 0$, $x_2(3) = \frac{1}{2}$, $x_3(1) = 0$, $x_3(2) = \frac{1}{2}$, $x_3(3) = 0$. Note that $S_x = \{2, 3\}$ and $S_x \times A_x = \{(2, 3), (3, 2)\}$, which identifies an ergodic Markov chain on S_x with $p_{22} = 0$, $p_{23} = 1$, $p_{32} = 1$ and $p_{33} = 0$. This chain has reward $r_2(3)x_2(3) + r_3(2)x_3(2) = 7$.

Program (6.4) is the primary tool in a procedure for computing an average optimal policy. This procedure consists of successive application up $2n$ linear programs, where n is the number of ergodic chains in the average optimal policy identified. The status of the procedure at each step is described by the pair (U, T), where U can be thought of as the set of states about which nothing is known, and T as a set of states that can be rendered transient.

The procedure is initialized with $U = S$ and $T = \emptyset$. Apply program (6.4). Let x be an optimal basic solution, so that the states S_x comprise the ergodic set with the maximum average reward, identified by this solution. Set $f(i) := a$ whenever $x_i(a) > 0$. If $S_x = U$, then this procedure completely specifies an average optimal policy f^∞. On the other hand, suppose $S_x \neq U$, so that the average reward maximizing ergodic chain does not encompass all the states. The search routine given below either succeeds in rendering all states $U\backslash S_x$ transient or identifies a subproblem that can be treated by program (6.4) for a smaller MDP.

Search routine

```
1. $V := U\backslash S_x$.
2. Search for a pair $(i, a) \in V \times A$ such that $\sum_{j \notin V} p_{ij}(a) > 0$.
     if no such pair exists then go to step 4
     else begin $f(i) := a$; $V := V\backslash\{i\}$ end
3. if $V = \emptyset$ then STOP
     else return to step 2
4. $T := T \cup \{U\backslash(S_x \cup V)\}$; STOP.
```

If the search procedure terminates with $V = \emptyset$, then all states in $U\backslash S_x$ are rendered transient by policy f^∞ and the average reward for these states is that of the ergodic chain with the maximum average reward, implying $\phi(f^\infty) = \phi$.

Suppose the search routine stops with $V \neq \emptyset$. From step 2 of the routine it follows that V is closed under any policy. In other words, the restriction of the original MDP to V gives a smaller MDP. The states in T are rendered transient by the policy identified in the search routine and will be treated later.

If $V \neq \emptyset$, replace U by V and reapply program (6.4) to find an ergodic chain with maximum average reward for this subproblem. If this chain fails to exhaust the states, the search routine can be reapplied, and doing so either renders all remaining states transient or identifies a further subproblem to which program (6.4) can be applied.

The entire procedure, including the augmentation of T is to be applied iteratively until it terminates, which occurs when an ergodic chain in program (6.4) exhausts the remaining states or when the search routine renders all remaining states transient. With each application, program (6.4) contributes to policy f^∞ an ergodic chain with maximum average reward for the current subproblem.

The iterative procedure just described exhausts all the states and completely specifies policy f^∞. lt is devised so that $\phi_i(f^\infty) = \phi_i$ for all $i \in S\backslash T$. The set T was added to at application of the search routine in such a way that $P(f)$ renders all states in T transient. Unfortunately, it can occur that $\phi_i(f^\infty) < \phi_i$ for some state(s) in T, in which case $f(i)$ will be modified. Since $S\backslash T$ is closed under $P(f)$, any policy that results from such a modification will remain optimal for all states in $S\backslash T$.

Suppose $T \neq \emptyset$. Let $R_i(a) := \sum_{j \notin T} p_{ij}(a)\phi_j$ for all $i \in T$ and $a \in A(i)$. Consider the linear program

$$min\Big\{ \sum_{j\in T} w_j \;\Big|\; \sum_{j \in T} \{\delta_{ij} - p_{ij}(a)w_j \geq R_i(a),\ (i,a) \in T \times A\Big\}. \tag{6.23}$$

Program (6.23) is the linear program for a substochastic MDP on the states T, which computes an optimal transient policy if (6.23) has a finite optimal solution (see program (4.32) and Theorem 4.18).

Lemma 6.12
Program (6.23) has a finite optimal solution.

Proof
We first show that the constant vector w defined by $w_j := max_{k \notin T}\, \phi_k, j \in T$, is a feasible solution of (6.23). Therefore, we have to show that $w_i \geq \sum_{j\in T} p_{ij}(a)w_j + \sum_{j \notin T} p_{ij}(a)\phi_j$ for all $i \in T$ and $a \in A(i)$. Indeed, we can write for any $(i,a) \in T \times A$,

$$\begin{aligned} \textstyle\sum_{j\in T} p_{ij}(a)w_j + \sum_{j\notin T} p_{ij}(a)\phi_j &\leq \textstyle\sum_{j \in T} p_{ij}(a) \cdot (max_{k\notin T}\, \phi_k) + \sum_{j\notin T} p_{ij}(a)\cdot (max_{k\notin T}\,\phi_k) \\ &= (max_{k \notin T}\,\phi_k)\cdot\{\textstyle\sum_{j\in T} p_{ij}(a) + \sum_{j\notin T} p_{ij}(a)\} = max_{k\notin T} = w_i. \end{aligned}$$

Next, assume that (6.23) has a infinite optimal solution. Then, the dual of (6.23) is infeasible. Since all states in T are transient under $P(f)$ in the original MDP, f^∞ is a transient policy in the MDP with states T. Then, by Theorem 4.14, the dual of (6.23) is feasible. This provides a contraction. Hence, program (6.23) has a finite optimal solution. □

Since the optimal solution w^* of program (6.23) is the transient value vector of the corresponding substochastic MDP, w^* is unique. An optimal transient policy follows from the dual program of (6.23), as shown in Theorem 4.14. The next example shows that we can still have $\phi_i(f^\infty) < \phi_i$ for some $i \in T$. This situation will arise if and only if an ergodic chain exists completely in T and has a higher average reward then the present $\phi_i(f^\infty)$ for these states.

Example 6.9
$S = \{1,2,3,4\};\ A(1) = A(2) = A(3) = \{1\},\ A(4) = \{1,2,3\};\ r_1(1) = 8, r_2(1) = 6, r_3(1) = 2, r_4(1) = 10,$ $r_4(2) = 5, r_4(3) = 7;\ p_{11}(1) = 1,\ p_{12}(1) = p_{13}(1) = p_{14}(1) = 0;\ p_{21}(1) = 0, p_{22}(2) = 1, p_{23}(1) = p_{24}(1) = 0;$ $p_{31}(1) = p_{32}(1) = 0,\ p_{33}(1) = 1,\ p_{34}(1) = 0;\ p_{41}(1) = \frac{1}{2},\ p_{42}(1) = 0,\ p_{43}(1) = \frac{1}{2},\ p_{44}(1) = 0;\ p_{41}(2) = 0,$ $p_{42}(2) = 1,\ p_{43}(2) = p_{44}(2) = 0;\ p_{41}(3) = p_{42}(3) = p_{43}(3) = 0,\ p_{44}(3) = 1.$

Iteration 1

The linear program (6.4) of this example becomes:

$$max\{8x_1(1) + 6x_2(1) + 2x_3(1) + 10x_4(1) + 5x_4(2) + 7x_4(3)\}$$

subject to the constraints (without the nonnegativity constraints)

$$\begin{array}{ccccccccccccc}
 & & & & & - & \frac{1}{2}x_4(1) & & & & & = & 0 \\
 & & & & & & & - & x_4(2) & & & = & 0 \\
 & & & & & - & \frac{1}{2}x_4(1) & & & & & = & 0 \\
 & & & & & & x_4(1) & + & x_4(2) & & & = & 0 \\
x_1(1) & + & x_2(1) & + & x_3(1) & + & x_4(1) & + & x_4(2) & + & x_4(3) & = & 1
\end{array}$$

An extreme optimal solution is x with $x_1(1) = 1,\ x_2(1) = x_3(1) = x_4(1 = x_4(2) = x_4(3) = 0$. Since $S_x \times A_x = \{(1,1)\}$, we have $f(1) = 1$ and this solution identifies an ergodic Markov chain on S_x with average reward $\phi_1 = \phi_1(f^\infty) = r_1(1)x_1(1) = 8$.

Next, we apply the search routine, starting with $U = \{1,2,3,4\}$ and $T = \emptyset$:
$V := \{2,3,4\};\ (i,a) = (4,1);\ f(4) := 1;\ V := \{2,3\};\ T := \{4\}$.

Iteration 2

The linear program (6.4) on $\{2,3\}$ becomes

$$max\{6x_2(1) + 2x_3(1) \mid x_2(1) + x_3(1) = 1;\ x_2(1), x_3(1) \geq 0\}.$$

An extreme optimal solution of this linear program is $x_2(1) = 1,\ x_3(1) = 0$. Since $S_x \times A_x = \{(2,1)\}$, we have $f(2) = 1$ and this solution identifies an ergodic Markov chain on S_x with the average reward $\phi_2 = \phi_2(f^\infty) = r_2(1)x_2(1) = 6$. The search routine gives: $V = \{3\},\ T = \{4\}$.

Iteration 3

The linear program (6.4) on $\{3\}$ becomes $max\{2x_3(1) \mid x_3(1) = 1;\ x_3(1) \geq 0\}$ with only one feasible solution, namely $x_3(1) = 1$. Since $S_x \times A_x = \{(3,1)\}$, we have $f(3) = 1$ and this solution identifies an ergodic Markov chain on S_x with the average reward $\phi_3 = \phi_3(f^\infty) = r_3(1)x_3(1) = 2$. The search routine gives: $V = \{3\},\ T = \{4\}$. Now, all states are identified. We have three ergodic sets: $\{l\}, \{2\}$ and $\{3\}$, and one transient state $\{4\}$. Notice that $\phi_4(f^\infty) = \frac{1}{2}(\phi_1 + \phi_3) = 5$ and $\phi_4 = 7$.

Iteration 4

In this iteration we solve program (6.23) with $R_4(1) = 5,\ R_4(2) = 6,\ R_4(3) = 0$. This linear program becomes $min\{w_4 \mid w_4 \geq 5;\ w_4 \geq 6;\ w_4 \geq 0\}$ with optimal solution $w_4^* = 6$. This solution provides a new action in state 4, namely $f(4) = 2$. Note that still $\phi_4(f^\infty) = 6 < \phi_4 = 7$.

To test whether $\phi_i(f^\infty) < \phi_i$ for some $i \in T$ and, if it does, to find an average optimal chain in T, we apply the following procedure:

1. Strip off the transient decisions (see below for this strip off routine).
2. Apply program (6.4) restricted to the remaining states and actions in T and with immediate rewards $s_i(a) := r_i(a) - \phi_i(f^\infty)$.

Strip off routine

1. Search for a pair $(i,a) \in T \times A$ such that $\sum_{j \notin T} p_{ij}(a) > 0$.
   If no such pair exists: stop this routine.
2. Delete action a from $A(i)$.
   if $A(i) = \emptyset$ **then** $T := T\backslash\{i\}$.
3. **if** $T = \emptyset$ **then begin** f^∞ is an average optimal policy; STOP **end**
   else go to step 1.

Consider program (6.4) restricted to the remaining states and actions of T after the stripping off routine. This program determines the ergodic chain in T which maximizes the maximal average reward with respect to the rewards $s_i(a) := r_i(a) - \phi_i(f^\infty)$. If program (6.4) has a nonpositive optimum, then $\phi(f^\infty) = \phi$. So, suppose that x is a basic optimal solution of (6.4) with a positive objective. Then, the ergodic chain on $T_x := \{i \mid \sum_a x_i(a) > 0\}$ identifies an ergodic chain with maximal average reward with respect to the rewards $s_i(a)$ which is higher then $\phi(f^\infty)$. Revise f^∞ by setting $f(i) := a$ for some a with $x_i(a) > 0$. Necessarily, for the revised policy f^∞, we obtain $\phi_i(f^\infty) = \phi_i,\ i \in T_x$. If $T_x = T$, then we have determined an average optimal policy. On the other hand, if $T_x \neq T$, then delete T_x from T, redefine $R_i(a)$ accordingly and return to program (6.23). This procedure iterates, identifies an ergodic chain in T each time and terminates finitely with an optima] policy.

Algorithm 6.7

Determination of an average optimal policy by linear programming by a sequence of linear programs

Input: Instance of a general MDP.

Output: An optimal deterministic policy f^∞.

```
1. $T := \emptyset$.
2. determine the optimum $v$ and an optimal solution $x$ of the linear program (6.4).
3. $S_x := \{i \mid \sum_a x_i(a) > 0\}$;.
4. for all $i \in S_x$ do begin select $f(i)$ such that $x_i\big(f(i)\big) > 0$; $\phi_i := v$ end
5. if $S_x = S$ then go to step 7.
   else go to step 6.
6. (a) $U := S$; $V := U \backslash S_x$.
   (b) for all $(i,a) \in V \times A$ do
           begin if $\sum_{j \notin V} p_{ij}(a) > 0$ then begin $f(i) := a$; $V := V \backslash \{i\}$ end
   (c) if $V = \emptyset$ then go to step 7
       else begin $T := T \cup \{U \backslash (S_x \cup V)\}$; $S := V$; go to step 2.
7. if $T = \emptyset$ then go to step 13
   else go to step 8
8. (a) for all $(i,a) \in T \times A$ do $R_i(a) := \sum_{j \notin T} p_{ij}(a)\phi_j$
   (b) determine an optimal transient policy $f^\infty$ for the substochatic MDP on the states of $T$ with
       rewards $R_i(a)$, $(i,a) \in T \times A$.
9. (a) for all $(i,a) \in T \times A$ do
           begin if $\sum_{j \notin T} p_{ij}(a) > 0$ then
               begin $A(i) := A(i) \backslash \{a\}$; if $A(i) = \emptyset$ then $T := T \backslash \{i\}$ end
           end
   (b) if $T = \emptyset$ then go to step 13
       else go to step 10
10. for all $(i,a) \in T \times A$ do $s_i(a) := r_i(a) - \phi_i(f^\infty)$
11. determine the optimum $v$ and an optimal solution $x$ of the linear program (6.4) on state space $T$,
    action sets $A(i)$, $i \in T$ and rewards $s_i(a)$.
```

```
12. **if** $v \leq 0$ **then go to** step 13
        **else begin** $f(i) := a$ for some $a$ with $x_i(a) > 0,\ i \in T_x$;
                **if** $T_x = T$ **then go to** step 13
                **else begin** $T := T \backslash T_x$; **go to** step 8 **end**
13. $f^\infty$ is an optimal policy (STOP).
```

Example 6.9 (continued)
We have already seen that we have computed in the first iterations:
Iteration 1: $f(1) = 1,\ \phi_1 = 8,\ T = \{4\}$.
Iteration 2: $f(2) = 1,\ \phi_2 = 6,\ T = \{4\}$.
Iteration 3: $f(3) = 1,\ \phi_3 = 2,\ T = \{4\}$.
Iteration 4: $R_4(1) = 5,\ R_4(2) = 6,\ R_4(3) = 0;\ f(4) = 2$.
Then, we continue in step 9 of Algorithm 6.7: $A(4) = \{2,3\};\ A(4) = \{3\};\ s_4(3) = 7 - 6 = 1$;
Iteration 5: $v = 1;\ x_4(3) = 1;\ f(4) = 3;\ f^\infty$ with $f(1) = 1,\ f(2) = 1,\ f(3) = 1,\ f(4) = 3$ is an optimal policy.

Relation between linear programming and policy iteration

For discounted and irreducible MDPs with the average reward criterion there is a one-to-one correspondence between the basic feasible solutions of the linear program and the deterministic policies. For the unichain average reward criterion we shall show that every deterministic policy corresponds to a basic feasible solution. The reverse statement is not true as the next example shows.

Example 6.10
$S = \{1,2,3\};\ A(1) = A(2) = \{1\},\ A(3) = \{1,2\};\ r_1(1) = 0,\ r_2(1) = -1,\ r_3(1) = -1,\ r_3(2) = 0.$
$p_{11}(1) = 1,\ p_{12}(1) = p_{13}(1) = 0;\ p_{21}(1) = p_{22}(2) = p_{23}(1) = \frac{1}{3};\ p_{31}(1) = p_{32}(1) = p_{33}(1) = \frac{1}{3};$
$p_{31}(2) = 1,\ p_{32}(2) = p_{33}(2) = 0.$
The linear program (6.4) of this example becomes (without the nonnegativity constraints):

$$max \left\{ -x_2(1) - x_3(1) \;\middle|\; \begin{array}{ccccccccc} & - & \frac{1}{3}x_2(1) & - & \frac{1}{3}x_3(1) & - & x_3(2) & = & 0 \\ & & \frac{2}{3}x_2(1) & - & \frac{1}{3}x_3(1) & & & = & 0 \\ & - & \frac{1}{3}x_2(1) & + & \frac{2}{3}x_3(1) & + & x_3(2) & = & 0 \\ x_1(1) & + & x_2(1) & + & x_3(1) & + & x_3(2) & = & 1 \end{array} \right\}.$$

The solution is x with $x_1(1) = 1,\ x_2(1) = x_3(1) = x_3(2) = 0$ is feasible and basic (even optimal). This solution does not correspond to a deterministic policy, because it is unclear which action should be chosen in state 3.

A slightly different version of program (6.4) is

$$max \left\{ \sum_{i,a} r_i(a)x_i(a) \;\middle|\; \begin{array}{ll} \sum_{i,a} x_i(a) & = \ 1 \\ \sum_{i,a}\{\delta_{ij} - p_{ij}(a)\}x_i(a) & = \ 0,\ j = 2,3,\dots,N \\ x_i(a) \geq 0,\ i \in S,\ a \in A(i) & \end{array} \right\}. \tag{6.24}$$

Since the rows $\sum_{i,a}\{\delta_{ij} - p_{ij}(a)\}x_i(a) = 0,\ j = 1,2,\dots,N$ are dependent (the sum of these N constraints is zero), one of these rows, say row 1, can be deleted. So, (6.24) is an equivalent linear program.

Let f^∞ be any deterministic policy and set $x_i(a) := 0$ whenever $a \neq f(i),\ i \in S$. Let the matrix $C(f)$ be obtained from $\{I - P(f)\}$ by replacing the first column by the 1-vector e. The nonzero variables of any feasible solution x of (6.24) with $x_i(a) := 0$ whenever $a \neq f(i),\ i \in S$, present an N-vector x and the constraints are transformed to $x^TC(f) = (1, 0, 0, \dots, 0)$, i.e. x is the stationary distribution $\pi(f)$ and $C(f)$ is invertible, because $P(f)$ is unichained, so must be a basis matrix. Moreover, the value of the objective function associated with this basis is $\pi(f)^Tr(f) = \phi(f^\infty)$.

Every deterministic policy corresponds in this way to a basic feasible solution to program (6.24). One might then hope that, if the simplex method for program (6.24) is initiated with a basis corresponding to a deterministic policy, it executes a series of pivot steps with each successive basis corresponding to a deterministic policy. To see what happens, we first rewrite the constraints. Let $C_i(a)$ be the column in (6.24) corresponding to $x_i(a)$. Then, the constraints of (6.24) can be written as $\sum_{i,a} x_i(a)C_i(a) = e^1$. With $C(f)$ as the current basis, the variable chosen to enter the basis is in the usual simplex method the one for which $r(f)^T\{C(f)^T\}^{-1}C_i(a) - r_i(a)$ is the most negative. Notice that $C(f)z = r(f)$ has a solution for which $z_1 = (1, 0, \dots, 0)^T\{C(f)\}^{-1}r(f) = \pi(f)r(f) = \phi(f^\infty)$. Consequently, because e is the first column of $C(f)$, the other columns of $C(f)$ are the corresponding columns of $\{I - P(f)\}$ and because $z_1 = \phi(f^\infty)$, we have $z_i = y_i,\ 2 \leq i \leq N$, where $y = (y_1, y_2, \dots, y_N)^T$ is a solution of the linear system $\{I - P(f)\}y = r(f) - \phi(f^\infty) \cdot e$. Hence, we can write $r(f)^T\{C(f)^T\}^{-1}C_i(a) - r_i(a) = z^TC_i(a) - r_i(a)$. Since $\{C_i(a)\}_j = \begin{cases} 1 & j = 1 \\ \delta_{ij} - p_{ij}(a) & j = 2, 3, \dots, N \end{cases}$ we obtain $z^TC_i(a) = \phi(f^\infty) + \sum_{j=2}^N \{\delta_{ij} - p_{ij}(a)\}y_j$, where y is a solution of $\{I - P(f)\}y = r(f) - \phi(f^\infty) \cdot e$. Since this system is unique up to a constant, we have $r(f)^T\{C(f)^T\}^{-1}C_i(a) - r_i(a) = \phi(f^\infty) + \sum_j \{\delta_{ij} - p_{ij}(a)\}y_j - r_i(a)$, where y is any solution of $\{I - P(f)\}y = r(f) - \phi(f^\infty) \cdot e$.

Hence, the reduced cost $r(f)^T\{C(f)^T\}^{-1}C_i(a) - r_i(a)$ is negative if and only if $a \in B(i, f)$. Let $t_i(a) := r_i(a) - \phi(f^\infty) - \sum_j \{\delta_{ij} - p_{ij}(a)\}y_j$. The usual version of the simplex method pivots in the column of $x_i(a)$ for which $t_i(a)$ is the most positive. This is the same choice as made in the usual version of the policy iteration algorithm.

On the other hand, the standard simplex method does not necessarily call for removal $x_i\big(f(i)\big)$ from the basis variables. However, we know that a basic solution results by exchanging $x_i(a)$ and $x_i\big(f(i)\big)$. Furthermore, by Theorem 6.10, cycling is precluded. Therefore we apply the simplex method with the following modified pivot rule.

Modified pivot rule:
Suppose $C(f)$ is the current basis. Let $g(i)$ be such that $t_i\big(g(i)\big) = max_a\, t_i(a) > 0$, then exchange the nonbasic variable $x_i\big(g(i)\big)$ with the basic variable $x_i\big(f(i)\big)$.

The above observations are summarized in the following theorem.

Theorem 6.14
The following procedures make the same sequence of policies:

(1) The simplex routine applied to linear program (6.24), initiated with a basis $C(f)$, and using the modified pivot rule.

(2) The policy iteration algorithm, initiated with policy f^∞, and in each iteration changing only the one decision for which $t_i(a)$ is most positive.

Example 6.10 (continued)
For this MDP the linear program (6.24) becomes (without the nonnegativity constraints):

$$max \left\{ -x_2(1) - x_3(1) \;\middle|\; \begin{array}{rcrcrcrcl} x_1(1) & + & x_2(1) & + & x_3(1) & + & x_3(2) & = & 1 \\ & & \frac{2}{3}x_2(1) & - & \frac{1}{3}x_3(1) & & & = & 0 \\ & - & \frac{1}{3}x_2(1) & + & \frac{2}{3}x_3(1) & + & x_3(2) & = & 0 \end{array} \right\}.$$

If we start with f^∞, then $C(f)^T = \begin{pmatrix} 1 & 1 & 1 \\ 0 & \frac{2}{3} & -\frac{1}{3} \\ 0 & -\frac{1}{3} & \frac{2}{3} \end{pmatrix}$ with $\{C(f)^T\}^{-1} = \begin{pmatrix} 1 & -3 & -3 \\ 0 & 2 & 1 \\ 0 & 1 & 2 \end{pmatrix}$.

The first simplex tableaus corresponding to the initial basis are (the pivot are the bold numbers):

		$x_1(1)$	$x_2(1)$	$x_3(1)$	$x_3(2)$
z_1	1	**1**	1	1	1
z_2	0	0	$\frac{2}{3}$	$-\frac{1}{3}$	0
z_3	0	0	$-\frac{1}{3}$	$\frac{2}{3}$	1
	0	0	1	1	0

		z_1	$x_2(1)$	$x_3(1)$	$x_3(2)$
$x_1(1)$	1	1	1	1	1
z_2	0	0	$\boldsymbol{\frac{2}{3}}$	$-\frac{1}{3}$	0
z_3	0	0	$-\frac{1}{3}$	$\frac{2}{3}$	1
	0	0	1	1	0

		z_1	z_2	$x_3(1)$	$x_3(2)$
$x_1(1)$	1	1	$-\frac{3}{2}$	$\frac{3}{2}$	1
$x_2(1)$	0	0	$\frac{3}{2}$	$-\frac{1}{2}$	0
z_3	0	0	$\frac{1}{2}$	$\boldsymbol{\frac{1}{2}}$	1
	0	0	$-\frac{3}{2}$	$\frac{3}{2}$	0

		z_1	z_2	z_3	$x_3(2)$
$x_1(1)$	1	1	-3	-3	-1
$x_2(1)$	0	0	2	1	1
$x_3(1)$	0	0	1	2	**2**
	0	0	-3	-3	-3

Notice that $r(f)^T\{C(f)^T\}^{-1}C_3(2) - r_3(2) = (0,-1,-1)\begin{pmatrix} 1 & -3 & -3 \\ 0 & 2 & 1 \\ 0 & 1 & 2 \end{pmatrix}\begin{pmatrix} 1 \\ 0 \\ 1 \end{pmatrix} - 0 = -3$, which is the reduced cost of the nonbasic variable $x_3(2)$. Next, we exchange $x_3(1)$ and $x_3(1)$, which gives the following optimal simplex tableau.

		z_1	z_2	z_3	$x_3(1)$
$x_1(1)$	1	1	$-\frac{5}{2}$	-2	$\frac{1}{2}$
$x_2(1)$	0	0	$\frac{3}{2}$	0	$-\frac{1}{2}$
$x_3(2)$	0	0	$\frac{1}{2}$	1	$\frac{1}{2}$
	0	0	$-\frac{3}{2}$	0	$\frac{3}{2}$

6.2.4 Value iteration

In this section we present another algorithm than the relative value Algorithm 5.10. The algorithm in this section is based on the value iteration algorithm for discounted MDPs and is stated below (see also Algorithm 6.3 with $k = 1$, which is a similar algorithm). When certain extra conditions are met, then the algorithm terminates in a finite number of iterations. In that case, the algorithm provides a deterministic ε-optimal policy and a $\frac{1}{2}\varepsilon$-approximation of the value. This last result is based on Lemma 6.4.

Algorithm 6.8 *Value iteration (unichain case)*
Input: Instance of a unichain MDP and some scalar $\varepsilon > 0$.
Output: An ε-optimal deterministic policy f^∞ and a $\frac{1}{2}\varepsilon$-approximation of the value ϕ.

```
1. Select $x \in \mathbb{R}^N$ arbitrary
2. Determine $f$ such that $T_f\, x = T\, x$.
3. $l := min_i\, (Tx - x);\ u := max_i\, (Tx - x)$.
4. if $u - l \leq \varepsilon$ then
   begin $f^\infty$ is an $\varepsilon$-optimal policy and $\frac{1}{2}(u + l)$ is a $\frac{1}{2}\varepsilon$-approximation of the value $\phi$ (STOP) end
   else begin $x := T_f\, x$; return to step 2 end
```

We will provide conditions under which the stopping criterion of Algorithm 6.8 holds in a finite number of iterations. Given the starting point $v^1 \in \mathbb{R}^N$, let $v^{n+l} := Tv^n$ for $n = 1, 2, \dots$. Let f_n^∞ be the policy for which $v^{n+1} = r(f_n) + P(f_n)v^n = Tv^n = T_{f_n}v^n \geq T_f v^n$ for all deterministic policies f^∞ and all $n \in \mathbb{N}$.

Theorem 6.15
span $(v^{n+2} - v^{n+1}) \leq \gamma \cdot span\,(v^{n+1} - v^n)$, *where* $\gamma := max_{i \in S,\, a \in A(i),\, j \in S,\, b \in A(j)} \{1 - \sum_k p(i,a,j,b,k)\}$ *with* $p(i,a,j,b,k) := min\{p_{ik}(a), p_{jk}(b)\}$.

Proof
We have

$$v^{n+2} - v^{n+1} \leq T_{f_{n+1}}v^{n+1} - T_{f_{n+1}}v^n = P(f_{n+1})(v^{n+1} - v^n) \leq max_i\,\{P(f_{n+1})(v^{n+1} - v^n\}_i \cdot e$$

and

$$v^{n+2} - v^{n+1} \geq T_{f_n}v^{n+1} - T_{f_n}v^n = P(f_n)(v^{n+1} - v^n) \geq min_j\,\{P(f_n)(v^{n+1} - v^n\}_j \cdot e.$$

Hence, we can write

$$\begin{aligned}
span\,(v^{n+2} - v^{n+1}) &\leq max_i\,\{P(f_{n+1})(v^{n+1} - v^n)\}_i - min_j\,\{P(f_n)(v^{n+1} - v^n)\}_j \\
&= max_{i,j}\,\Big\{\{P(f_{n+1})(v^{n+1} - v^n)\}_i - \{P(f_n)(v^{n+1} - v^n)\}_j\Big\} \\
&= max_{i,j}\,\{\textstyle\sum_k p_{ik}(f_{n+1})(v^{n+1} - v^n)_k - \sum_k p_{jk}(f_n)(v^{n+1} - v^n)_k\} \\
&\leq max_{i,j}\{max_{a \in A(i)} \textstyle\sum_k p_{ik}(a)(v^{n+1} - v^n)_k - min_{b \in A(i)} \sum_k p_{jk}(b)(v^{n+1} - v^n)_k\} \\
&= max_{i,j,a,b}\{\textstyle\sum_k p_{ik}(a)(v^{n+1} - v^n)_k - \sum_k p_{jk}(b)(v^{n+1} - v^n)_k\} \\
&= max_{i,j,a,b} \textstyle\sum_k \{p_{ik}(a) - p_{jk}(b)\}(v^{n+1} - v^n)_k \\
&= max_{i,j,a,b} \textstyle\sum_k \{[p_{ik}(a) - p(i,a,j,b,k)] + [p(i,a,j,b,k) - p_{jk}(b)]\}(v^{n+1} - v^n)_k \\
&\leq max_{i,j,a,b}\big\{\textstyle\sum_k \{[p_{ik}(a) - p(i,a,j,b,k)] \cdot max_k\,(v^{n+1} - v^n)_k \\
&\qquad\qquad - \textstyle\sum_k\, [p_{jk}(b) - p(i,a,j,b,k)] \cdot min_k\,(v^{n+1} - v^n)_k\big\} \\
&= max_{i,j,a,b}\big\{[1 - \textstyle\sum_k\, p(i,a,j,b,k)] \cdot max_k\,(v^{n+1} - v^n)_k \\
&\qquad\qquad - [1 - \textstyle\sum_k\, p(i,a,j,b,k)] \cdot min_k\,(v^{n+1} - v^n)_k\big\} \\
&= max_{i,j,a,b}\{1 - \textstyle\sum_k\, p(i,a,j,b,k)\} \cdot span\,(v^{n+1} - v^n) \\
&= \gamma \cdot span\,(v^{n+1} - v^n).
\end{aligned}$$

□

Note that $0 \leq \gamma \leq 1$. Furthermore, if $\gamma < 1$, then Theorem 6.15 ensures that in a finite number of iterations the stopping criterion of Algorithm 6.8 will be satisfied. The following example shows that, however $\gamma = 1$ and the model is unichain but not irreducible, the value iteration algorithm 6.8 may still convergence.

Example 6.11
$S = \{1,2,3\}$; $A(1) = \{1,2\}$, $A(2) = A(3) = \{1\}$; $r_1(1) = 2$, $r_1(2) = 1$, $r_2(1) = 2$, $r_3(1) = 3$.
$p_{11}(1) = p_{12}(1) = 0$, $p_{13}(1) = 1$; $p_{11}(2) = 0$, $p_{12}(2) = 1$, $p_{13}(2) = 0$; $p_{21}(1) = 1$, $p_{22}(1) = p_{23}(1) = 0$; $p_{31}(1) = p_{32}(1) = p_{33}(1) = \frac{1}{3}$.
It is easy to verify that this model unichain but not irreducible. Furthermore, we have:
$p_{11}(1) = p_{11}(2) = 0 \;\rightarrow\; min\{p_{11}(1), p_{11}(2)\} = 0$; $p_{12}(1) = 0$, $p_{12}(2) = 1 \;\rightarrow\; min\{p_{12}(1), p_{12}(2)\} = 0$; $p_{13}(1) = 1$, $p_{13}(2) = 0 \;\rightarrow\; min\{p_{13}(1), p_{13}(2)\} = 0$. Hence, $p(1,1,1,2,k) = 0$ for all k, and consequently $\gamma = 1$. The next tabular presents the results of with $x = v^1 = (0,0,0)$ and $\varepsilon = 0.01$.

n	v_1^n	v_2^n	v_3^n	$v_1^n - v_1^{n-1}$	$v_2^n - v_2^{n-1}$	$v_3^n - v_3^{n-1}$	$span\,(v^n - v^{n-1})$	f^∞
1	0.000	0.000	0.000					(1,1,1)
2	2.000	2.000	3.000	2.000	2.000	3.000	1.000	(1,1,1)
3	5.000	4.000	5.333	3.000	2.000	2.333	1.000	(1,1,1)
4	7.333	7.000	7.778	2.333	3.000	2.445	0.667	(1,1,1)
5	9.778	9.333	10.370	2.445	2.333	2.592	0.259	(1,1,1)
6	12.370	11.778	12.827	2.592	2.445	2.457	0.147	(1,1,1)
7	14.827	14.370	15.325	2.457	2.592	2.498	0.135	(1,1,1)
8	17.325	16.827	17.841	2.498	2.457	2.516	0.059	(1,1,1)
9	19.841	19.325	20.331	2.516	2.492	2.490	0.026	(1,1,1)
10	22.331	21.841	22.832	2.490	2.516	2.501	0.026	(1,1,1)
11	24.832	24.331	25.335	2.501	2.490	2.503	0.013	(1,1,1)
12	27.335	26.832	27.833	2.503	2.501	2.498	0.005	

Observe that Algorithm 6.8 terminates for $n = 12$ and identifies an ε-optimal policy f^∞ with $f(1) = f(2) = f(3) = 1$. A $\frac{1}{2}\varepsilon$-approximation of the value ϕ is $\frac{1}{2}(2.503 + 2.498) = 2.501$.

The above example shows that the value iteration algorithm may convergence with respect to the span seminorm even for $\gamma = 1$. Our approach for showing that this algorithm terminates after a finite number of iterations relies on the concept of an M-stage span contraction.

We say that an operator $B : \mathbb{R}^N \to \mathbb{R}^N$ is an *M-stage span contraction* if there exists a $0 \le \beta < 1$ and a nonnegative integer M for which $span\,(B^M x - B^M y) \le \beta \cdot span\,(x - y)$ for all $x, y \in \mathbb{R}^N$. A vector $v^* \in \mathbb{R}^N$ is a *span fixed point* of B if $span\,(Bv^* - v^*) = 0$, i.e. $B^* - v^* = c \cdot e$ for some scalar c. If the operator T is an M-stage span contraction with contraction factor β, we have

$$span\,(v^{M+2} - v^{M+1}) = span\,\{T^M(Tv^1) - T^M v^1\} \le \beta \cdot span\,(Tv^1 - v^1) = \beta \cdot span\,(v^2 - v^1).$$

Using the techniques of the theory of contraction mappings, the following theorem can be shown similar to the proof of Theorem 3.1 (we leave the details of the proof to the reader).

Theorem 6.16

Let T be an M-stage span contraction with contraction factor β. Then,
(1) There exists a span fixed point v^* of T;
(2) For any $v^1 \in \mathbb{R}^N$, the sequence $v^{n+1} := Tv^n$ $(n = 1, 2, \dots)$ satisfies $\lim_{n\to\infty} span\,(v^n - v^*) = 0$.

For any $f^\infty \in C(D)$ and any $n \in \mathbb{N}$, the (i, j)th element of the matrix $P^n(f)$ is denoted by $p^n_{ij}(f)$. For any pair $f_1^\infty, f_2^\infty \in C(D)$ and any $M \in \mathbb{N}$, the constant $\gamma(f_1^\infty, f_2^\infty, M)$ is defined by

$$\gamma(f_1^\infty, f_2^\infty, M) := 1 - min_{i,j \in S} \sum_k p(i, f_1^\infty, j, f_2^\infty, M), \tag{6.25}$$

where $p(i, f_1^\infty, j, f_2^\infty, M) := min\,\{p^M_{ik}(f_1), p^M_{jk}(f_2)\}$. The following result is a generalization of Theorem 6.15 and can be shown straightforward.

Theorem 6.17

Suppose there exists an integer $M \ge 1$ such that $\gamma(f_1^\infty, f_2^\infty, M) < 1$ for every pair $f_1^\infty, f_2^\infty \in C(D)$, Then,

(1) T is an M-stage span contraction with factor $\gamma^*(M) := max_{f_1^\infty, f_2^\infty \in C(D)}\, \gamma(f_1^\infty, f_2^\infty, M)$;

(2) For any $v^1 \in \mathbb{R}^N$ and any $\varepsilon > 0$, for the sequence $\{v^n\}_{n=1}^\infty$ with $v^{n+1} := Tv^n$, $n = 1, 2, \dots$, there exists an integer n_0 such that $span\,(v^{nM+2} - v^{nM+1}) < \varepsilon$ for all $n \ge n_0$.

Remark

The condition $\gamma(f_1^\infty, f_2^\infty, M) < 1$ 1 means that starting in any pair of distinct states i and j, the policies f_1^∞ and f_2^∞ both reach after exactly M transitions at least one identical state k with positive probability. Hence, $\gamma^*(M) < 1$ implies that all policies must be unichain and aperiodic. Clearly, it is not easy to verify the condition $\gamma^*(M) < 1$ directly. The following theorem provides conditions which are easier to check and imply $\gamma^*(M) < 1$.

Theorem 6.18

Suppose either

(a) $0 \leq \gamma < 1$, *where γ is is defined in Theorem 6.15;*

(b) *there exists a state l and an integer $M \geq 1$ such that for any f^∞), we have $p_{il}^M(f) > 0$ for all $i \in S$;*

(c) *all policies are unichain and $p_{ii}(a) > 0$ for all $(i, a) \in S \times A$.*

Then, the condition of Theorem 6.17 holds, so the conclusion of Theorem 6.17 follows.

Proof

Assume that condition (a) holds. Take $M = 1$ and select any $f_1^\infty, f_2^\infty \in C(D)$. Then, we have

$$\begin{aligned} 1 &> max_{i\in S,\, a\in A(i),\, j\in S,\, b\in A(j)}\,\{1 - \textstyle\sum_k p(i,a,j,b,k)\} \\ &\geq max_{i,j\in S}\,\{1 - \textstyle\sum_k p(i, f_1(i), j, f_2(j), k)\} \\ &= 1 - min_{i,j\in S} \textstyle\sum_k p(i, f_1^\infty, j, f_2^\infty, M) = \gamma(f_1, f_2, M). \end{aligned}$$

Hence, the condition of Theorem 6.17 holds.

Assume that condition (b) holds. Then, $p(i, f_1, j, f_2, l, M) = min\,(p_{il}^M(f_1), p_{il}^M(f_2) > 0$ for every $i, j \in S$ and every $f_1^\infty, f_2^\infty \in C(D)$. Also in this case the condition of Theorem 6.17 holds, because

$$\gamma(f_1, f_2, M) = 1 - min_{i,j\in S} \textstyle\sum_k p(i, f_1, j, f_2, k, M) \leq 1 - min_{i,j\in S}\, p(i, f_1, j, f_2, l, M) < 1.$$

Assume that condition (c) holds. Select two different states, say i_1 and i_2, and two different deterministic policies, say f_1^∞ and f_2^∞. Let $X_1(n) := \{j \in S \mid p_{i_1 j}^n(f_1) > 0\}$ and $X_2(n) := \{j \in S \mid p_{i_2 j}^n(f_1) > 0\}$ for $n \in \mathbb{N}$. We show by contradiction that $X_1(N) \cap X_2(N) \neq \emptyset$. Suppose $X_1(N) \cap X_2(N) = \emptyset$. Since $p_{ii}(a) > 0$ for all $(i, a) \in S \times A$, we have $X_i(n) \subseteq X_i(n+1)$ for $i = 1, 2$ and for all $n \in \mathbb{N}$, so that $X_i(n) \cap X_2(n) = \emptyset$ for all $1 \leq n \leq N$. Consequently, for some $1 \leq m < N$, we have $X_1(m) = X_1(m+1)$ and $X_2(m) = X_2(m+1)$. This means that $X_1(m)$ is closed under f_1^∞ and $X_2(m)$ is closed under f_2^∞. However, this contradicts the unichain assumption. Hence, we have shown $X_1(N) \cap X_2(N) \neq \emptyset$, i.e. there exists a state l for which $min\,(p_{i_1 l}^N(f_1), p_{i_2 l}^N(f_2) > 0$. Hence, $p(i_1, f_1, i_2, f_2, l, N) > 0$. Since $i_1, i_2, f_1^\infty, f_2^\infty$ are arbitrarily chosen, $\gamma(f_1, f_2, M) > 0$ for $M = N$ and for all pairs $f_1^\infty, f_2^\infty \in C(D)$. Therefore, the condition of Theorem 6.17 holds. □

Remarks

1. Under any of the conditions (a), (b) and (c) of Theorem 6.18, the value iteration algorithm 6.8 terminates.
2. In condition (c) only the unichain assumption is essential. By the data transformation (5.48), any MDP can be transformed into an equivalent MDP with $p_{ii}(a) > 0$ for all $(i, a) \in S \times A$.

Example 6.11 (continued)

Inspection of this example reveals that, when starting the algorithm with $v^1 = (0, 0, 0)$, action 1 always achieves the maximum in state 1, so that value iteration corresponds to iterating the policy f_1^∞ with $f_1(1) = 1$. We have already seen that $\gamma = 1$, so condition (a) is not satisfied. It is obvious that also condition (c) is not satisfied. Suppose that condition (b) holds. Let $f_2^\infty \in C(D)$ be the policy with

$f_2(1) = 2$. It is easy to verify that there does not exists a state l and an integer $M \geq 1$ such that $p^M_{il}(f_2) > 0$ for all $i \in S$. Hence, this example does not satisfy any of the conditions of Theorem 6.18, but nevertheless, algorithm 6.8 terminates.

6.2.5 Modified policy iteration

We discuss the modified policy iteration for unichain MDPs under the same strong aperiodicity Assumption 6.2 as in the irreducible case. We also use the same algorithm (Algorithm 6.3), but for notation convenience we take in each iteration the same k. However, the proof of its correctness is more complicated. For the unichain case Lemma 6.6 no longer holds and the constant δ, defined in (6.14), may be zero, so Lemma 6.8 can no longer be used. Notice that Lemma 6.4, Lemma 6.5 and relation (6.13) hold also in the unichain case.

First, we will derive a similar result as in Lemma 6.6. This result enables us to show the boundedness of $span(x^n) := max\, x^n_i - min\, x^n_i$. Next, it is shown that the boundedness of $span(x^n)$ implies that l_n converges to ϕ. Finally, we show that there exists a subsequence of $\{u_n\}$ which converges to ϕ.

Define

$$\eta := min_{i,j \in S}\, min_{h_1,h_2,\dots,h_{N-1}} \sum_k min\Big\{\{P(h_1)P(h_2)\cdots P(h_{N-1})\}_{ik}, \{P(h_1)P(h_2)\cdots P(h_{N-1})\}_{jk}\Big\}. \tag{6.26}$$

Then, the unichain condition and the strong aperiodicity assumption yield the following result, which states that any two states i and j have a common successor after $N-1$ transitions.

Lemma 6.13
$\eta > 0$.

Proof
Let $h_1, h_2, \dots, h_{N-1}$ be an arbitrary sequence of deterministic decision rules and select $i, j \in S$ arbitrarily. Define $S(i,n)$ for $n = 0, 1, \dots, N-1$ as in the proof of Lemma 6.6. Clearly $S(i,n) \subseteq S(i,n+1)$ and if $S(i,n) = S(i,n+1)$, then $S(i,n)$ is closed under $P(h_{n+1})$. We have to shown $S(i,N-1) \cap S(j,N-1) \neq \emptyset$. Suppose $S(i,N-1) \cap S(j,N-1) = \emptyset$. Then $S(i,N-1)$ and $S(j,N-1)$ are both proper subsets of S, so there exists $0 \leq m,n \leq N-2$ such that $S(i,m) = S(i,m+1)$ and $S(j,n) = S(j,n+1)$. This implies that $S(i,m)$ is closed under $P(h_{m+1})$ and that $S(j,n)$ is closed under $P(h_{n+1})$. Since $S(i,N-1) \cap S(j,N-1) = \emptyset$, $S(i,m) \cap S(j,n)$ is also empty. Let f^∞ be the policy with $f(s) := h_{m+1}$ for $s \in S(i,m)$ and $f(s) := h_{n+1}$ for $s \in S(j,n)$ (outside $S(i,m) \cup S(j,n)$ choose $f(s)$ arbitrary). Then, $P(f)$ has two disjunct, nonempty closed subsets, namely $S(i,m)$ and $S(j,n)$, which contradict the unichain condition. □

Lemma 6.14
For all $x \in \mathbb{R}^N$ and all deterministic decision rules $h_1, h_2, \dots, h_{N-1}$, we have $span(Qx) \leq (1-\eta) \cdot span(x)$, with $Q := P(h_1)P(h_2)\cdots P(h_{N-1})$.

Proof
Let i and j such that $span(Qx) = (Qx)_i - (Qx)_j$. Then,

$$\begin{aligned} span(Qx) &= \textstyle\sum_k \{q_{ik} - q_{jk}\} x_k \\ &= \textstyle\sum_k \big\{q_{ik} - min(q_{ik}, q_{jk})\big\} x_k - \sum_k \big\{q_{jk} - min(q_{ik}, q_{jk})\big\} x_k \\ &\leq \textstyle\sum_k \big\{q_{ik} - min(q_{ik}, q_{jk})\big\} \cdot max_k\, x_k - \sum_k \big\{q_{jk} - min(q_{ik}, q_{jk})\big\} \cdot min_k\, x_k \\ &= span(x) - \textstyle\sum_k min(q_{ik}, q_{jk}) \cdot \{max_k\, x_k - min_k\, x_k\} \leq (1-\eta) \cdot span(x). \end{aligned}$$

□

Define K by $K = max_{i,a}\, r_i(a) - min_{i,a}\, r_i(a)$. Then, $span\big(r(f)\big) \leq K$ for all $f^\infty \in C(D)$.

Lemma 6.15
$span(T_{h_1}T_{h_2}\cdots T_{h_{N-1}}x) \leq (N-1)\cdot K + (1-\eta)\cdot span(x)$ for all $x \in \mathbb{R}^N$ and all deterministic decision rules $h_1, h_2, \dots, h_{N-1}$.

Proof
Since $span(y+z) \leq span(y) + span(z)$ for all y, z and $span\{P(f)y\} \leq span(y)$ for all decision rules f and all y, we obtain

$$\begin{array}{rcl} span(T_{h_1}T_{h_2}\cdots T_{h_{N-1}}x) & = & span\big\{r(h_1) + P(h_1)r(h_2) + \cdots + \\ & & \qquad P(h_1)P(h_2)\cdots P(h_{N-2})r(h_{N-1}) + P(h_1)P(h_2)\cdots P(h_{N-1})x\big\} \\ & \leq & span\{r(h_1)\} + span\{r(h_2)\} + \cdots + \\ & & \qquad span\{r(h_{N-1})\} + span\{P(h_1)P(h_2)\cdots P(h_{N-1})x\} \\ & \leq & (N-1)\cdot K + span\{P(h_1)P(h_2)\cdots P(h_{N-1})x\} \\ & \leq & (N-1)\cdot K + (1-\eta)\cdot span(x), \end{array}$$

the last inequality by Lemma 6.14. □

In order to prove that $span(x^n)$ is bounded, we introduce the following notation for a fixed $k \in \mathbb{N}$:

$$w^{nk+p} := T^p_{f_n}\, x^n, \text{ for } n = 0, 1, \dots \text{ and } p = 0, 1, \dots, k-1.$$

Then, $w^{nk} = T^0_{f_n}\, x^n = x^n$, and consequently, $w^{nk+p} = T^p_{f_n}\, w^{nk},\ n = 0, 1, \dots;\ p = 0, 1, \dots, k-1$.

Theorem 6.19
$span\,(x^m) \leq \frac{1}{\eta}\cdot(N-1)\cdot K + span(x^0)$ for $m = 0, 1, \dots$.

Proof
It follows from Lemma 6.15, with $k = N-1$, that for all $n = 0, 1, \dots$ and all $p = 0, 1, \dots, N-2$, we have

$$\begin{array}{rcl} span\{w^{n(N-1)+p}\} & \leq & (N-1)\cdot K + (1-\eta)\cdot span\{w^{(n-1)(N-1)+p}\} \\ & \leq & (N-1)\cdot K + (1-\eta)\cdot(N-1)\cdot K + (1-\eta)^2\cdot span\{w^{(n-2)(N-1)+p}\} \\ & \leq & \cdots \ \leq \ \cdots \\ & \leq & (N-1)\cdot K + (1-\eta)\cdot(N-1)\cdot K + \cdots + (1-\eta)^{n-1}(N-1)K + (1-\eta)^n span\{w^p\}. \end{array}$$

Furthermore, it follows from the proof of Lemma 6.15 that

$$span\{w^p\} = T^p_{f_0}\, x^0 \leq p\cdot K + span(x^0) \leq (N-1)\cdot K + span(x^0) \text{ for } p = 0, 1, \dots, N-2.$$

Hence,

$$span\{w^{n(N-1)+q}\} \leq \textstyle\sum_{j=0}^n (1-\eta)^j\cdot(N-1)\cdot K + (1-\eta)^n span(x^0) \leq \frac{1}{\eta}\cdot(N-1)\cdot K + span(x^0),$$

implying that for any $m = 0, 1, \dots$, we have $span(w^m) \leq \frac{1}{\eta}\cdot(N-1)\cdot K + span(x^0)$. Since $w^{mk} = x^m$ for all $m \geq 0$, the theorem is proven. □

Before we can prove the convergence of the modified policy iteration, we first have to derive some other results.

$$\begin{array}{rcl} x^{n+1}-x^n & = & T^k_{f_n}\, x^n - x^n \\ & = & r(f_n)+P(f_n)r(f_n)+\cdots+P^{k-1}(f_n)r(f_n)+P^k(f_n)x^n-x^n \\ & = & \{I+P(f_n)+\cdots+P^{k-1}(f_n)\}\{r(f_n)+P(f_n)x^n-x^n\} \\ & = & \{I+P(f_n)+\cdots+P^{k-1}(f_n)\}\{T_{f_n}x^n-x^n\} \\ & = & \{I+P(f_n)+\cdots+P^{k-1}(f_n)\}\{Tx^n-x^n\} \\ & = & \{I+P(f_n)+\cdots+P^{k-1}(f_n)\}g^n. \end{array}$$

So, we obtain for $n=0,1,\dots$ and $m=1,2,\dots$

$$x^{n+m}-x^n=\sum_{l=n}^{n+m-1}\{I+P(f_l)+\cdots+P^{k-1}(f_l)\}g^l. \tag{6.27}$$

Consider the iterates $x^n,x^{n+1},\dots,x^{n+m-1}$. Since $u_l\geq\phi$ for all l, there has to be a state $j_0\in S$ with $g^l_{j_o}\geq\phi$ for at least $\frac{m}{N}$ of the m indices $l=n,n{+}1,\dots,n{+}m{-}1$. Using (6.27), where $x^{n+m}{-}x^n$ is expressed as a sum of km terms, and using the property (see Lemma 6.4 and Lemma 6.5) that $g^l\geq l_l\cdot e\geq l_n\cdot e$ for $l=n,n+1,\dots,n+m-1$, we obtain

$$x^{n+m}-x^n\geq\Big\{\frac{m}{N}\cdot\phi+\big\{km-\frac{m}{N}\big\}\cdot l_n\Big\}\cdot e=\big\{km\cdot l_n+\frac{m}{N}\cdot(\phi-l_n)\big\}\cdot e. \tag{6.28}$$

From (6.13) it follows that $g^{n+m}\geq P^k(f_{n+m-1})P^k(f_{n+m-2})\cdots P^k(f_n)g^n$, i.e. g^{n+m} is computed from g^n by premultiplication of km transition matrices. Hence, by the strong aperiodicity condition and with $\alpha:=min_{i,a}\,p_{ii}(a)\in(0,1]$, we obtain

$$\begin{array}{rcl} g^{n+m}_i & = & \sum_j\,\{P^k(f_{n+m-1})P^k(f_{n+m-2})\cdots P^k(f_n)\}_{ij}g^n_j \\ & = & \{P^k(f_{n+m-1})P^k(f_{n+m-2})\cdots P^k(f_n)\}_{ii}g^n_i+\sum_{j\neq i}\,\{P^k(f_{n+m-1})P^k(f_{n+m-2})\cdots P^k(f_n)\}_{ij}g^n_j \\ & \geq & \alpha^{km}g^n_i+(1-\alpha^{km})l_n,\ i\in S. \end{array}$$

For $p=0,1,\dots,k-1$, we obtain similarly

$$\begin{array}{rcl} g^{n+m}_i & \geq & \alpha^{km-p}\cdot\{P^p(f_n)g^n\}_i+(1-\alpha^{km-p})\cdot min_j\,\{P^p(f_n)g^n\}_j \\ & \geq & \alpha^{km-p}\cdot\{P^p(f_n)g^n\}_i+(1-\alpha^{km-p})\cdot l_n \\ & \geq & \alpha^{km}\cdot\{P^p(f_n)g^n\}_i+(1-\alpha^{km})\cdot l_n,\ i\in S, \end{array}$$

the last inequality because $\alpha^{-p}\cdot\big\{\{P^p(f_n)g^n\}_i-l_n\big\}\geq\{P^p(f_n)g^n\}_i-l_n,\ i\in S.$

Let $l_*:=\lim_{n\to\infty}l_n$ and let $i_0\in S$ satisfy $g^{n+m}_{i_0}=l_{n+m}$. Then, for $p=0,1,\dots,k-1$,

$$\begin{array}{rcl} \{P^p(f_n)g^n\}_{i_0} & \leq & \alpha^{-km}\{g^{n+m}_{i_0}-(1-\alpha^{km})l_n\}\ =\ l_n+\alpha^{-km}\{g^{n+m}_{i_0}-l_n\} \\ & = & l_n+\alpha^{-km}\{l_{n+m}-l_n\}\ \leq\ l_n+\alpha^{-km}\{l_*-l_n\}. \end{array}$$

Hence, $\big\{\{I+P(f_n)+\cdots+P^{k-1}(f_n)\}g^n\big\}_{i_0}\leq k\cdot l_n+k\cdot\alpha^{-km}\{l_*-l_n\}$ for all n. So, because the sequence l_n is nondecreasing, we have for $l=n,n+1,\dots,n+m-1$

$$\begin{array}{rcl} \big\{\{I+P(f_l)+\cdots+P^{k-1}(f_l)\}g^l\big\}_{i_0} & \leq & k\cdot l_l+k\cdot\alpha^{-km}\{l_*-l_l\} \\ & = & k\cdot\alpha^{-km}l_*+k\cdot l_l(1-\alpha^{-km}) \\ & \leq & k\cdot\alpha^{-km}l_*+k\cdot l_n(1-\alpha^{-km}) \\ & = & k\cdot l_n+k\cdot\alpha^{-km}(l_*-l_n), \end{array}$$

the last inequality because $\alpha<1$ implies $1-\alpha^{-km}<0$ and because the sequence l_n is nondecreasing. Therefore, we can write, using equation (6.27),

$$(x^{n+m}-x^n)_{i_0}=\Big\{\sum_{l=n}^{n+m-1}\{I+P(f_l)+\cdots+P^{k-1}(f_l)\}g^l\Big\}_{i_0}\leq km\cdot l_n+km\cdot\alpha^{-km}(l_*-l_n). \tag{6.29}$$

It follows from (6.28) and (6.29) that

$$span\,(x^{n+m}-x^n) \geq \big\{km\cdot l_n+\frac{m}{N}(\phi-l_n)\big\}-\big\{km\cdot l_n+km\cdot\alpha^{-km}(l_*-l_n)\big\} = \frac{m}{N}(\phi-l_n)-km\cdot\alpha^{-km}(l_*-l_n). \tag{6.30}$$

Theorem 6.20

$l_n \uparrow \phi$.

Proof

From Lemma 6.5 and Lemma 6.4 it follows that $l_n \uparrow$ and $l_n \leq \phi$ for all n. So, $l_* := \lim_{n\to\infty} l_n \leq \phi$. Suppose that $l_* < \phi$. Since, by Theorem 6.19, $span\,(x^n)$ is bounded, there exists a positive constant K_1 such that $span(x^n) \leq K_1$ for all n. Select m_* such that $\frac{m_*}{N}\cdot(\phi-l_n) \geq 2K_1+K_2$ for all n, where K_2 is some positive constant. Next, select n_* such that $km_*\cdot\alpha^{-km_*}(l_*-l_{n_*}) < K_2$. Then, it follows from (6.30) that $span\,(x^{n_*+m_*}-x^n_*) > (2K_1+K_2)-K_2 = 2K_1$. Since $span\,(x) \geq span\,(x-y)-span\,(y)$ for every x and y (see Exercise 6.1), we obtain $K_1 \geq span\,(x^{n_*+m_*}) \geq span\,(x^{n_*+m_*}-x^n_*)-span\,(x^n_*) > 2K_1-K_1 = K_1$, implying a contradiction. □

We now know that l_n converges to ϕ and, by Lemma 6.4, that f_n^∞ is ε-optimal for n sufficiently large. In order to be able to recognize that n is sufficiently large one needs the following result.

Theorem 6.21

ϕ is the smallest limit point of the sequence $\{u_n\}$.

Proof

We know from Lemma 6.4 that $u_n \geq \phi$ for all n. From the proof of Theorem 6.19 it follows that the sequence $\{span\,(w^n)\}$ is bounded. Since,

$$u_n-l_n = span\,(g^n) = span\,(Tx^n-x^n) = span\,(w^{nk+1}-w^{nk}) \leq span\,(w^{nk+1})+span\,(w^{nk}),$$

the sequence $\{u_n-l_n\}$ is bounded. Because $l_n \uparrow \phi$, also the sequence $\{u_n\}$ is bounded. Let u_* be the smallest limit point of $\{u_n\}$ and suppose that $u_* > \phi$. Then one may construct, similar as in the proof of Theorem 6.20 where we supposed that $l_* < \phi$ (we also need a similar expression as (6.30)), a contradiction. Hence, ϕ is the smallest limit point of the sequence $\{u_n\}$. □

Finally, we show that $span\,(g^n)$ converges to zero geometrically fast. Since $\{span\,(x^n)\}$ is bounded, also $\{span\,(x^n-x^n_N\cdot e)\}$ is bounded. Furthermore, $\phi\cdot e$ is a limit point of $\{g^n\}$. Because there are only a finite number of policies, there exists a subsequence of $\{x^n\}$ and $\{g^n\}$ with $g^{n_m}\to\phi\cdot e$, $f^\infty_{n_m} = f$ and $x^{n_m}-x^{n_m}_N\cdot e\to x$ for some $f^\infty\in C(D)$ and some $x\in\mathbb{R}^N$.

Then, for all m, $max_{g^\infty\in C(D)}\,T_g\,x^{n_m}-x^{n_m} = T_{f_{n_m}}\,x^{n_m}-x^{n_m} = T_f\,x^{n_m}-x^{n_m} = g^{n_m}$. Letting m tends to infinity and using the property $T_g\,x^{n_m}-x^{n_m} = T_g\,\{x^{n_m}-x^{n_m}_N\cdot e\}-\{x^{n_m}-x^{n_m}_N\cdot e\}$, we obtain $max_{g^\infty\in C(D)}\,T_g\,x-x = T_f\,x-x = \phi\cdot e$. Consequently, $T^k_f x = x+k\cdot\phi$.

Lemma 6.16

If $span\,(x^n-x) \leq \varepsilon$ and $T_{f_n}\,x = x+\phi\cdot e$, then $span\,(x^{n+1}-x) \leq \varepsilon$ and $T_{f_{n+1}}\,x \geq x+\phi\cdot e-\varepsilon\cdot e$.

Proof

Since $x^{n+1} = T^k_{f_n}\,x^n = T^k_{f_n}\,x+P^k(f_n)(x^n-x) = x+k\cdot\phi\cdot e+P^k(f_n)(x^n-x)$, we obtain the property $span\,(x^{n+1}-x) = span\,\{P^k(f_n)(x^n-x)\} \leq span\,(x^n-x) \leq \varepsilon$. Furthermore, we have

$$\begin{aligned} T_{f_{n+1}}\,x - x &= T_{f_{n+1}}\,x^{n+1} + P(f_{n+1})(x - x^{n+1}) - x \;\geq\; T_{f_n}\,x^{n+1} + P(f_{n+1})(x - x^{n+1}) - x \\ &= T_{f_n}\,x - x + P(f_n)(x^{n+1} - x) - P(f_{n+1})(x^{n+1} - x) \\ &\geq \phi \cdot e + min_i\,(x^{n+1} - x)_i \cdot e - max_i\,(x^{n+1} - x)_i \cdot e \\ &= \phi \cdot e - span\,(x^{n+1} - x) \cdot e \;\geq\; \phi \cdot e - \varepsilon \cdot e. \end{aligned}$$ □

Remark

Since $C(D)$ has a finite number of policies g^∞, there is also only a finite number of vectors $T_g\,x - x$ and at least one of them, namely $T_f\,x - x$, equals $\phi \cdot e$. Hence, the finiteness of $C(D)$ implies that there exists an $\varepsilon > 0$ such that $T_g\,x - x \geq \phi \cdot e - \varepsilon \cdot e$ if and only if $T_g\,x - x = \phi \cdot e$. If $\varepsilon > 0$ is taken in this way, Lemma 6.16 gives the following result.

Corollary 6.2

If $span\,(x^n - x) \leq \varepsilon$ and $T_{f_n}\,x = x + \phi \cdot e$, then $span\,(x^{n+1} - x) \leq \varepsilon$ and $T_{f_{n+1}}\,x = x + \phi \cdot e$, where $\varepsilon > 0$ is taken as in the above remark.

We have seen that $x^{n_m} - x^{n_m}_N \cdot e - x \to 0$ if $m \to \infty$. So, also $span(x^{n_m} - x) = span(x^{n_m} - x^{n_m}_N \cdot e - x) \to 0$ if $m \to \infty$. Furthermore, since $f^\infty_{n_m} = f$ for all m, $T_{f_{n_m}}\,x = x + \phi \cdot e$ for all m. Hence, there exists a number n_* and an $\varepsilon_* > 0$ such that $span\,(x^{n_*} - x) \leq \varepsilon$ and $T_{f_{n_*}}\,x - x = \phi \cdot e$. Then, by inductively applying Corollary 6.2, $span(x^{n_*+m} - x) \leq \varepsilon$ and $T_{f_{n_*+m}}\,x - x = \phi \cdot e$ for all $m = 0, 1, \ldots$.

Furthermore,

$$\begin{aligned} x^{n_*+m} &= T^k_{f_{n_*}+m-1} T^k_{f_{n_*}+m-2} \cdots T^k_{f_{n_*}}\,x^{n_*} \\ &= T^k_{f_{n_*+m-1}} T^k_{f_{n_*+m-2}} \cdots T^k_{f_{n_*}}\,x + P^k(f_{n_*+m-1})P^k(f_{n_*+m-2}) \cdots P^k(f_{n_*})(x^{n_*} - x) \\ &= x + mk \cdot \phi \cdot e + P^k(f_{n_*+m-1})P^k(f_{n_*+m-2}) \cdots P^k(f_{n_*})(x^{n_*} - x). \end{aligned}$$

Hence, $span\,(x^{n_*+m} - x) = span\,\{P^k(f_{n_*+m-1})P^k(f_{n_*+m-2}) \cdots P^k(f_{n_*})(x^{n_*} - x)\}$, and by Lemma 6.14 $span\,(x^{n_*+m} - x)$ decreases exponentially fast to zero as $m \to \infty$. Then, also g^{n_*+m} converges exponentially fast to $\phi \cdot e$, since

$$\begin{aligned} g^{n_*+m} &= T_{f_{n_*+m}}\,x^{n_*+m} - x^{n_*+m} = T_{f_{n_*+m}}\,x^{n_*+m} - x^{n_*+m} - T_{f_{n_*+m}}\,x + x + \phi \cdot e \\ &= \{P(f_{n_*+m}) - I\}(x^{n_*+m} - x) + \phi \cdot e. \end{aligned}$$

Therefore, the convergence of the modified policy iteration method is exponentially fast.

6.3 The communicating case

In this section we make the following assumption.

Assumption 6.4

For every $i, j \in S$ there exists a policy $f^\infty \in C(D)$, which may depend on i and j, such that in the Markov chain $P(f)$ state j is accessible from state i.

Clearly, this assumption is equivalent to the property that every *completely mixed stationary policy* π^∞, i.e. $\pi_{ia} > 0$ for every $(i, a) \in S \times A$, is irreducible. We have seen in section 5.2.3 that checking the communicating property can be done in polynomial time (polynomial in $M := \sum_{i \in S} |A(i)|$). In this case, policies with two or more recurrent sets are possible, but there is an optimal policy which has only one recurrent set. Hence, the value vector ϕ has identical components and there exists a unichain optimal policy.

6.3.1 Optimality equation

In the communicating case the value vector ϕ is a constant vector. Hence, by Theorem 5.11, ϕ is the unique x-part in the optimality equation (6.1). The next example shows that in the communicating case the property that the y-vector is unique up to a constant does not hold.

Example 6.12
$S = \{1,2\};\ A(1) = A(2) = \{1,2\};\ r_1(1) = 1,\ r_1(2) = 0;\ r_2(1) = 1,\ r_2(2) = 0;\ p_{11}(1) = 1,\ p_{12}(1) = 0;$ $p_{11}(2) = 0,\ p_{12}(2) = 1;\ p_{21}(1) = 0,\ p_{22}(1) = 1;\ p_{21}(2) = 1,\ p_{22}(2) = 0.$
This is a multichain, but communicating model. The optimality equation (6.1) becomes:

$$x + y_1 = max\{1 + y_1, 0 + y_2\};\ x + y_2 = max\{1 + y_2, 0 + y_1\}.$$

Two different solutions are: $x = 1,\ y_1 = 0,\ y_2 = 1$ and $x = 1,\ y_1 = 1,\ y_2 = 0$. The difference between the y-vectors is the non-constant vector $(-1, 1)$.

6.3.2 Policy iteration

Since there exists a unichain optimal policy one might conjecture that we can solve such a problem using Algorithm 6.4. The following example shows that this is not true, even when we start with a unichain policy.

Example 6.13
$S = \{1,2,3\};\ A(1) = \{1,2\},\ A(2) = \{1,2,3\},\ A(3) = \{1,2\}.$
$r_1(1) = 0,\ r_1(2) = 2;\ r_2(1) = 1,\ r_2(2) = 1,\ r_2(3) = 3;\ r_3(1) = 2;\ r_3(2) = 4.$
$p_{12}(1) = p_{11}(2) = p_{23}(1) = p_{21}(2) = p_{22}(3) = p_{32}(1) = p_{33}(2) = 1$ (other transitions are 0).
This is a multichain and communicating model.
Algorithm 6.4 with starting policy $f(1) = f(2) = 2,\ f(3) = 1$ and taking $y_1 = 0$ in (6.1) gives:
Iteration 1:

Consider the system $$\begin{cases} x + y_1 - y_1 = 2 \\ x + y_2 - y_1 = 1 \\ x + y_3 - y_2 = 2 \\ y_1 = 0 \end{cases} \rightarrow\ x = \phi(f^\infty) = 2,\ y_1 = 0,\ y_2 = y_3 = -1.$$

$B(1, f) = \emptyset,\ B(2, f) = \{3\},\ B(3, f) = \{2\}.$
$g(1) = 2,\ g(2) = 3,\ g(3) = 2.\ f(1) = 2,\ f(2) = 3,\ f(3) = 2.$

Iteration 2:

Consider the system $$\begin{cases} x + y_1 - y_1 = 2 \\ x + y_2 - y_2 = 3 \\ x + y_3 - y_3 = 4 \\ y_1 = 0 \end{cases} \rightarrow\ \text{inconsistent system (multichain policy).}$$

Below we state the following modification of the multichain policy iteration algorithm (Algorithm 5.6), which exploits the communication structure by finding a 'unichain improvement' which indicates whether or not the current policy is known to be unichain.

Algorithm 6.9 *Determination of an average optimal policy by policy iteration (communicating case)*
Input: Instance of an MDP.
Output: An optimal deterministic policy f^∞ and the value ϕ.

```
1. Select any $f^\infty \in C(D)$; go to step 2 (b).
2. (a) if $constant = 0$ then go to step 2 (b) else go to step 2 (c);
   (b) determine $\phi(f^\infty)$ and $y = u^0(f)$ as unique $(x, y)$-part in a solution of the system
           $\begin{cases} \{I - P(f)\}x & & & = 0 \\ x & + \{I - P(f)\}y & & = r(f) \\ & y & + \{I - P(f)\}z & = 0 \end{cases}$
       go to step 3.
   (c) determine $\phi(f^\infty)$ and $y = u^0(f)$ as unique $(x, y)$-part in a solution of the system
           $\begin{cases} x \cdot e & + \{I - P(f)\}y & & = r(f) \\ & y & + \{I - P(f)\}z & = 0 \end{cases}$
       go to step 3.
3. (a) if $\phi(f^\infty)$ is constant then go to step 3 (g) else go to step 3 (b);
   (b) $S_0 := \{i \in S \mid \phi_i(f^\infty) = max_k \, \phi_k(f^\infty)\}$; $g(i) := f(i),\ i \in S$; $T := S \backslash S_0$; $W := S_0$;
   (c) if $T = \emptyset$ then go to step 3 (e).
   (d) select $j \in T$ and $a_j \in A(j)$ such that $\sum_{k \in W} p_{jk}(a_j) > 0$; $T := T \backslash \{j\}$; $W := W \cup \{j\}$;
       $g(j) := a_j$; return to step 3 (c).
   (e) $constant := 1$; $f := g$; return to step 2.
   (f) for all $i \in S$ do $B(i, f) := \{a \in A(i) \mid r_i(a) + \sum_j p_{ij}(a)u^0_j(f) > \phi(f^\infty) + u^0_i(f)\}$;
   (g) if $B(i, f) = \emptyset$ for every $i \in S$ then $f^\infty$ is an average optimal policy (STOP)
       else begin select $g$ such that $r_i(g) + \sum_j p_{ij}(g)u^0_j(f) = max_a\{r_i(a) + \sum_j p_{ij}(a)u^0_j(f)\},\ i \in S$;
               $f := g$; $constant := 0$; return to step 2
          end
```

Example 6.13 (continued)

We apply Algorithm 6.8 to the model of Example 6.13, starting with $f(1) = f(2) = 2,\ f(3) = 1$.

Iteration 1:

step 2 (b): $\phi(f^\infty) = (2, 2, 2);\ u^0(f) = (0, -1, -1)$.

step 3 (c): $B(1, f) = \emptyset,\ B(2, f) = \{3\},\ B(3, f) = \{2\}$; $g(1) = 2,\ g(2) = 3,\ g(3) = 2$;
$f(1) = 2,\ f(2) = 3,\ f(3) = 2$; $constant := 0$.

Iteration 2:

step 2 (b): $\phi(f^\infty) = (2, 3, 4);\ u^0(f) = (0, 0, 0)$.

step 3 (b): $S_0 = \{3\}$; $g(1) = 2,\ g(2) = 3,\ g(3) = 2;\ T = \{1, 2\},\ W = \{3\}$.
$j = 2,\ a_j = 1;\ T = \{1\},\ W = \{2, 3\},\ g(2) = 1$.
$j = 1,\ a_j = 1;\ T = \emptyset,\ W = \{1, 2, 3\},\ g(1) = 1$.

step 3 (e): $constant = 1$; $f(1) = f(2) = 1,\ f(3) = 2$.

Iteration 3:

step 2 (c): $\phi(f^\infty) = (4, 4, 4);\ u^0(f) = (-7, -3, 0)$.

step 3 (c): $B(1, f) = B(2, f) = B(3, f) = \emptyset$; f^∞ is an optimal policy.

Theorem 6.22

Algorithm 6.8 terminates in a finite number of iterations with an optimal policy.

Proof

Case 1: $\phi(f^\infty)$ is not constant.

We will show that in such iteration a policy g^∞ is found with $\phi(g^\infty) > \phi(f^\infty)$, which implies that this case can occur only in a finite number of iterations.

Since f^∞ is not a constant vector, step 3 (b) is executed. During step 3 (b) we have $S = T \cup W$ and $T \cap W = \emptyset$. At the start of this step $T \neq \emptyset$, since otherwise $\phi(f^\infty)$ is a constant vector. The communicating assumption guarantees that there exists at least one pair of states $k \in W$ and $j \in T$ with $a_j \in A(j)$ where $p_{jk}(a_j) > 0$. Hence, after $|T|$ subiterations of step 3 (b) the set T is empty and for the policy g^∞ the average reward $\phi(g^\infty)$ is constant. By the definition of S_0, we have $\phi_i(g^\infty) > \phi_i(f^\infty),\ i \notin S_0$.

Case 2: $\phi(f^\infty)$ is constant.

In this case step 3 (g) is executed. This step is the same as one iteration in the multichain case (Algorithm 5.6), because for a constant $\phi(f^\infty)$ the action set $B(i,f)$ of (5.18) becomes $B(i,f) = \{a \in A(i) \mid r_i(a) + \sum_j p_{ij}(a)u_j^0(f) > \phi(f^\infty) + u_i^0(f)\}$. From Theorem 5.14 it follows that if $B(i,f) = \emptyset,\ i \in S$, then f^∞ is an average optimal policy, and if $B(i,f) \neq \emptyset$ for at least one $i \in S$, then g^∞ is 'better' than f^∞.

Hence, we have shown that all policies are different, so the algorithm terminates, and at termination the last policy is optimal. □

6.3.3 Linear programming

Since the value vector ϕ is constant in communicating models, we would expect some simplification in the linear programming approach. The property that ϕ is the smallest superharmonic vector implies in this case that ϕ is the unique v-part of an optimal solution (x,y) of the linear program

$$min \left\{ x \;\middle|\; x + \sum_j \{\delta_{ij} - p_{ij}(a)\}y_j \geq r_i(a),\ i \in S,\ a \in A(i) \right\}. \tag{6.31}$$

The dual of (6.31) is

$$max \left\{ \sum_{i,a} r_i(a)x_i(a) \;\middle|\; \begin{array}{lcl} \sum_{i,a}\{\delta_{ij} - p_{ij}(a)\}x_i(a) & = & 0,\ j \in S \\ \sum_{i,a} x_i(a) & = & 1 \\ x_i(a) \geq 0,\ i \in S,\ a \in A(i) & & \end{array} \right\}. \tag{6.32}$$

The next example shows that - in contrast with the irreducible and the unichain case - in the communicating case the optimal solution of the dual program doesn't provide an optimal policy, in general.

Example 6.13 (continued)

The dual linear program (6.32) of this model is (without the nonnegativity of the variables)

maximize $2x_1(2) + x_2(1) + x_2(2) + 3x_2(3) + 2x_3(1) + 4x_3(2)$

subject to

$$\begin{array}{rrrrrrrrrrrrrrrr}
 & x_1(1) & & & & & - & x_2(2) & & & & & & & = & 0 \\
- & x_1(1) & & & & x_2(1) & + & x_2(2) & & & - & x_3(1) & & & = & 0 \\
 & & & & - & x_2(1) & & & & & + & x_3(1) & & & = & 0 \\
 & x_1(1) & + & x_1(2) & + & x_2(1) & + & x_2(2) & + & x_2(3) & + & x_3(1) & + & x_3(2) & = & 1
\end{array}$$

The optimal solution is: $x_1(1) = x_1(2) = x_2(1) = x_2(2) = x_2(3) = x_3(1) = 0;\ x_3(2) = 1$. The objective function value equals 4. Proceeding as if this were a unichain model, we choose arbitrary actions in the states 1 and 2. Clearly, this approach could generate a nonoptimal policy, e.g. $f(1) = 2,\ f(2) = 3$.

Theorem 6.23

Let x^* be an extreme optimal solution of (6.32) and let $S_* := \{i \mid \sum_a x_i^*(a) > 0\}$. Select any policy f_*^∞ such that $x_i^*\big(f_*(i)\big) > 0,\ i \in S_*$. Then, $\phi_j(f_*^\infty) = \phi,\ j \in S_*$.

Proof

The proof follows directly from Lemma 6.11. □

Theorem 6.24

An MDP is communicating if and only if for every $b \in \mathbb{R}^N$ such that $\sum_i b_i = 0$ there exists a $y \in \mathbb{R}^{|S \times A|}$ such that $y_i(a) \geq 0$ for $(i,a) \in S \times A$ and $\sum_{i,a} \{\delta_{ij} - p_{ij}(a)\} y_i(a) = b_j,\ j \in S$.

Proof

First, assume that we have a communicating MDP. Let π^∞ be a completely mixed stationary policy. Then, $P(\pi)$ is an irreducible Markov chain. Let x be the (strictly positive) stationary distribution of $P(\pi)$ and let $Z(\pi) := \{I - P(\pi) + P^*(\pi)\}$ be the fundamental matrix of $P(\pi)$. Choose any $b \in \mathbb{R}^N$ such that $\sum_i b_i = 0$. Define $d \in \mathbb{R}^N$ by $d^T := b^T Z(\pi) + c \cdot x^T$ with $c \geq 0$ sufficiently large to assure $d \geq 0$. Take $y_i(a) := d_i \cdot \pi_i(a),\ (i,a) \in S \times A$. Then, $y_i(a) \geq 0,\ (i,a) \in S \times A$. Notice that

$$\begin{array}{llllll} \sum_{i,a} \{\delta_{ij} - p_{ij}(a)\} y_i(a) = b_j,\ j \in S & \Leftrightarrow & \sum_i \{\delta_{ij} - p_{ij}(\pi)\} d_i = b_j,\ j \in S & \Leftrightarrow & d^T \{I - P(\pi)\} = b^T \\ & \Leftrightarrow & \{b^T Z(\pi) + c \cdot x^T\}\{I - P(\pi)\} = b^T & \Leftrightarrow & b^T Z(\pi)\{I - P(\pi)\} = b^T \\ & \Leftrightarrow & b^T\{I - P^*(\pi)\} = b^T & \Leftrightarrow & b^T P^*(\pi) = 0. \end{array}$$

Since $P^*(\pi)$ has identical rows, we obtain $\sum_i b_i p_{ij}^*(\pi) = p_{jj}^* \sum_i b_i = 0$ for all $j \in S$, i.e. $b^T P^*(\pi) = 0$. Hence, we have $y \in \mathbb{R}^{|S \times A|}$ such that $y_i(a) \geq 0$ for $(i,a) \in S \times A$ and $\sum_{i,a} \{\delta_{ij} - p_{ij}(a)\} y_i(a) = b_j,\ j \in S$.

Next, assume that for every $b \in \mathbb{R}^N$ with $\sum_i b_i = 0$ there exists a $y \in \mathbb{R}^{|S \times A|}$ such that $y_i(a) \geq 0$ for $(i,a) \in S \times A$ and $\sum_{i,a} \{\delta_{ij} - p_{ij}(a)\} y_i(a) = b_j,\ j \in S$. Suppose that the MDP is not communicating. Then, there exists a pair of states (k,l) such that $\{P^t(f)\}_{kl} = 0$ for all $f^\infty \in C(D)$ and all $t \geq 1$. Define $S_l := \{i \in S \mid \{P^t(f)\}_{il} > 0$ for some $f^\infty \in C(D)$ and some $t \geq 1\}$. Suppose that $S_l = \emptyset$. Then, for any b with $b_l < 0$ and $\sum_i b_i = 0$ with corresponding y such that $y_i(a) \geq 0$ for $(i,a) \in S \times A$ and $\sum_{i,a} \{\delta_{ij} - p_{ij}(a)\} y_i(a) = b_j,\ j \in S$, we have

$$0 > b_l = \textstyle\sum_{i,a} \{\delta_{il} - p_{il}(a)\} y_i(a) = \sum_{i,a} \delta_{il} y_i(a) = \sum_a y_l(a) \geq 0,$$

which gives a contradiction. Hence, $S_l \neq \emptyset$. Select any b such that $b_j < 0,\ j \in S_l,\ b_j > 0,\ j \notin S_l$ and $\sum_i b_i = 0$ with corresponding y, i.e. $y_i(a) \geq 0$ for $(i,a) \in S \times A$ and $\sum_{i,a} \{\delta_{ij} - p_{ij}(a)\} y_i(a) = b_j,\ j \in S$. Define $y \in \mathbb{R}^N$ by $y_i := \sum_a y_i(a),\ i \in S$ and a stationary policy π^∞ by $\pi_{ia} := \begin{cases} \frac{y_i(a)}{y_i} & \text{if } y_i > 0; \\ \text{arbitrary} & \text{if } y_i = 0. \end{cases}$

Then, $y_i(a) = y_i \cdot \pi_{ia},\ (i,a) \in S \times A$ and $y_j - \sum_i p_{ij}(\pi) y_i = b_j,\ j \in S$. Note that $p_{ij}(\pi) = 0$ for $j \in S_l$ and $i \notin S_l$. Hence, we obtain $y_j - \sum_{i \in S_l} p_{ij}(\pi) y_i = b_j,\ j \in S_l$. Summing up this equation over the states of S_l gives:

$$\textstyle\sum_{j \in S_l} y_j = \sum_{j \in S_l} b_j + \sum_{i \in S_l} \{\sum_{j \in S_l} p_{ij}(\pi)\} y_i \leq \sum_{j \in S_l} b_j + \sum_{i \in S_l} y_i.$$

Hence, $\sum_{j \in S_l} b_j \geq 0$, which gives the desired contradiction. □

In a unichain model, we can choose arbitrary actions in transient states because under any action the system eventually reaches the single recurrent class and achieves the maximal average reward. In a communicating model, such an approach can result in nonoptimal policies because it could keep the system outside of S_* indefinitely. Either one of the following approaches solves this problem.

1. Search procedure
Obtain an optimal solution x^* of program (6.32). For $i \in S_* := \{i \mid \sum_a x^*_i(a) > 0\}$, take for $f_*(i)$ the action which satisfies $x^*_i\big(f_*(i)\big) > 0$. For the remaining states use the following search procedure.

```
  while $S_* \neq S$ do
      begin select $i \notin S_*,\ j \in S_*,\ f_*(i) \in A(i)$ satisfying $p_{ij}\big(f_*(i)\big) > 0$; $S_* := S_* \cup \{i\}$ end
```

By the communicating property this search procedure will find in each state of $S \backslash S_*$ an action which drives the system to S_* with positive probability.

2. Determination of the y variables
Obtain an optimal solution x^* of program (6.32). Select any $\beta \in \mathbb{R}^N$ with $\beta_j > 0,\ j \in S$ and $\sum_j \beta_j = 1$. Let $b_j := \beta_j - \sum_a x^*_j(a),\ j \in S$. Then, $\sum_j b_j = \sum_j \beta_j - \sum_{j,a} x^*_j(a) = 1 - 1 = 0$. Because the model is communicating, by Theorem 6.24 there exists a y^* such that $y^*_i(a) \geq 0$ for $(i,a) \in S \times A$ and $\sum_{i,a} \{\delta_{ij} - p_{ij}(a)\} y^*_i(a) = b_j,\ j \in S$. Notice that (x^*, y^*) is an optimal solution of the dual linear program (5.29). If $i \in S_*$, take for $f_*(i)$ the action which satisfies $x^*_i\big(f_*(i)\big) > 0$; if $i \notin S_*$, take for $f_*(i)$ an action which satisfies $y^*_i\big(f_*(i)\big) > 0$. In Theorem 5.18 is shown that f_* is an optimal policy.

Algorithm 6.10
Determination of an average optimal policy by linear programming (communicating case)
Input: Instance of an MDP.
Output: An optimal deterministic policy f^∞ and the value ϕ.

1. Determine an extreme optimal solution x^* of the linear program

$$max \left\{ \sum_{i,a} r_i(a) x_i(a) \;\middle|\; \begin{array}{lcl} \sum_{i,a} \{\delta_{ij} - p_{ij}(a)\} x_i(a) & = & 0,\ j \in S \\ \sum_{i,a} x_i(a) & = & 1 \\ x_i(a) \geq 0,\ i \in S,\ a \in A(i) & & \end{array} \right\}.$$

2. Select $f_*(i)$ such that $x^*_i\big(f_*(i)\big) > 0,\ i \in S_* := \{i \mid \sum_a x^*_i(a) > 0\}$.

3. **either go to** step 4 (search procedure) **or go to** step 6 (determination y variables).

4. **while** $S_* \neq S$ **do**

   begin select $i \notin S_*,\ j \in S_*,\ f_*(i) \in A(i)$ satisfying $p_{ij}\big(f_*(i)\big) > 0$; $S_* := S_* \cup \{i\}$ **end**

5. **go to** step 7.

6. (a) Select any $\beta \in \mathbb{R}^N$ with $\beta_j > 0,\ j \in S$ and $\sum_j \beta_j = 1$;

   (b) $b_j := \beta_j - \sum_a x^*_j(a),\ j \in S$;

   (c) Determine y^* such that $y^*_i(a) \geq 0,\ (i,a) \in S \times A$ and $\sum_{i,a} \{\delta_{ij} - p_{ij}(a)\} y^*_i(a) = b_j,\ j \in S$;

   (d) Select $f_*(i)$ such that $y^*_i\big(f_*(i)\big) > 0,\ i \in S \backslash S_*$;

   (e) **go to** step 7.

7. f^∞_* is an average optimal policy and $\phi = \sum_{i,a} r_i(a) x^*_i(a)$. (STOP).

Example 6.13 (continued)
We apply Algorithm 6.10 to the model of Example 6.13 with $\beta_j = \frac{1}{3},\ j = 1,2,3$.
We execute both step 4 (search procedure) as step 6 (determination y variables).

Step 1:
We have already seen that the linear program (6.32) has as optimal solution x^* satisfying:
$x_1^*(1) = x_1^*(2) = x_2^*(1) = x_2^*(2) = x_2^*(3) = x_3^*(1) = 0;\ x_3^*(2) = 1$ with objective function value 4.
Step 2:
$S_* = \{3\};\ f_*(3) = 2.$
Step 4 (search procedure):
$i = 2;\ j = 3;\ f_*(2) = 1;\ S_* = \{2, 3\}.$
$i = 1;\ j = 2;\ f_*(1) = 1;\ S_* = \{1, 2, 3\}.$
Step 6 (determination y variables):
$b_1 = \frac{1}{3},\ b_2 = \frac{1}{3},\ b_3 = -\frac{2}{3}$. The system becomes (without the nonnegativity of the variables):

$$\begin{array}{rrrrrrrrrl} & y_1(1) & & & - & y_2(2) & & & = & \frac{1}{3} \\ - & y_1(1) & + & y_2(1) & + & y_2(2) & - & y_3(1) & = & \frac{1}{3} \\ & & - & y_2(1) & & & + & y_3(1) & = & -\frac{2}{3} \end{array}$$

with feasible solution $y_1^*(1) = \frac{1}{3},\ y_2^*(1) = \frac{2}{3},\ y_2^*(2) = y_3^*(1) = 0$. Hence, $f_*(1) = f_*(2) = 1$.
Step 6:
The optimal policy is $f_*(1) = f_*(2) = 1$ and $f_*(3) = 2$; the value $\phi = 4$.

Remarks

1. It turns out that Algorithm 6.10 with the search procedure can also be used in the so-called *optimal unichain case*, i.e. for all optimal stationary policies f^∞ the associated Markov chain $P(f)$ is unichain (cf. Exercise 6.8).
2. This approach can also be used in the so-called *weak unichain case* , i.e. if for all optimal stationary policies f^∞ and all ergodic sets $E(f)$ of $P(f)$ with $\phi_i(f^\infty) = max_j\ \phi_j(f^\infty)$ for all $i \in E(f)$, there exists a policy g^∞ such that the states of $S \backslash E(f)$ are transient in the Markov chain induced by $P(g)$ (cf. Exercise 6.9).
3. Below we present an example for which Algorithm 6.10 with the search procedure fails. This example shows that the optimal unichain case needs the requirement for *all* policies, and the weak unichain case needs the requirement for *all* policies and *all* ergodic sets $E(f)$.

Example 6.14
$S = \{1, 2, 3\};\ A(1) = \{1, 2\}, A(2) = \{1\}, A(3) = \{1\};\ r_1(1) = 1, r_1(2) = 0, r_2(1) = 0, r_3(1) = 1, r_3(2) = 0;$
$p_{11}(1) = 1,\ p_{12}(1) = p_{13}(1) = 0;\ p_{11}(2) = 0,\ p_{12}(2) = 1,\ p_{13}(2) = 0;\ p_{21}(1) = 1,\ p_{22}(1) = p_{23}(1) = 0;$
$p_{31}(1) = p_{32}(1) = 0,\ p_{33}(1) = 1;\ p_{31}(2) = 1,\ p_{32}(1) = p_{33}(1) = 0.$
There are two optimal policies: f_1^∞ with $f_1(1) = f_1(2) = f_1(3) = 1$ and f_2^∞ with $f_2(1) = f_2(2) = 1$ and $f_2(3) = 2$. The policy $P(f_1)$ has two ergodic sets, so this MDP is not optimal unichain. Furthermore, for $E(f_1) := \{3\}$ we have $\phi_i(f_1^\infty) = max_j\ \phi_j(f_1^\infty) = 1$ for all $i \in E(f_1)$, but there does not exists a policy g^∞ such that the states of $S \backslash E(f_1) = \{1, 2\}$ are transient in the Markov chain induced by $P(g)$. Hence, this MDP is not weakly unichain. However, for the optimal policy f_2^∞ we have $P(f_2)$ is unichain with one ergodic set $E(f_2) := \{1\}$ and the states of $S \backslash E(f_2) = \{2, 3\}$ are transient under $P(f_2)$.
We will apply Algorithm 6.10 with the search procedure. The linear program for this model is

$$max \left\{ x_1(1) + x_3(1) \left| \begin{array}{rrrrrrrrrrl} & & & x_1(2) & - & x_2(1) & & & - & x_3(2) & = & 0 \\ & & - & x_1(2) & + & x_2(1) & & & & & = & 0 \\ & x_1(1) & + & x_1(2) & + & x_2(1) & + & x_3(1) & + & x_3(2) & = & 1 \end{array} \right. \right\}.$$

An extreme optimal solution is x^* with $x_1^*(1) = x_1^*(2) = x_2^*(1) = x_3^*(2) = 0$ and $x_3^*(1) = 1$. Since $S_* = \{3\}$ and $S \backslash S^*$ is closed, Algorithm 6.10 with the search procedure fails: no optimal decision can be found in state 1.

For models with the (weakly) unichain property not for all, but for some optimal policy, an optimal policy can be found by repeatedly applying Algorithm 6.10 with the search procedure until in all states an action has been determined. For the model in this example, after selecting $f_*(3) = 1$ and finding that the states of $S\backslash S^*$ are closed, we solve the MDP on the states of $S\backslash S^* = \{1, 2\}$. The linear program becomes

$$max \left\{ x_1(1) \;\middle|\; \begin{array}{ccccccc} & & x_1(2) & - & x_2(1) & = & 0 \\ & - & x_1(2) & + & x_2(1) & = & 0 \\ x_1(1) & + & x_1(2) & + & x_2(1) & = & 1 \end{array} \right\}.$$

The unique extreme optimal solution of this problem is x^* with $x_1^*(1) = 1,\ x_1^*(2) = x_2^*(1) = 0$. This provides $f_*(1) = 1$. Finally, the search procedure gives $f_*(2) = 1$.

6.3.4 Value iteration

In section 5.9 we presented an algorithm for value iteration under the assumption that the value vector is constant and the Markov chains $P(f),\ f^\infty \in C(D)$, are aperiodic. The last part of this assumption is not a serious restriction: by a data transformation the original model can be transformed into a model in which every Markov chain $P(f),\ f^\infty \in C(D)$, is aperiodic and has the same average reward as the original Markov chain. In case of unichain models no better algorithm than Algorithm 5.10 is known.

6.3.5 Modified value iteration

Algorithm 6.3 is again used as the modified value iteration method for communicating MDPs. In Section 6.2.5 the convergence proof for the unichain case has been given in two stages. First, the unichain assumption and the strong aperiodicity assumption were used to prove that $span(x^n)$ is bounded (Theorem 6.19). In the second stage we used the boundedness of $span(x^n)$ and the property $u_n \geq \phi$ for all n to prove that $l_n \uparrow \phi$ and that ϕ is a limit point of the sequence $\{u_n\}$ (Theorems 6.20 and 6.21). From these proofs it is clear that the modified policy iteration method will convergence whenever $span(x^n)$ is bounded and the value vector ϕ is independent of the initial state (if the strong aperiodicity assumption holds), which is the case for communicating MDPs. Therefore, we have to show that the sequence $\{span(x^n)\}$ is also bounded in the communicating case. Define $M,\ L_n,\ U_n$ and θ by:

$$M := max_{i,a}\,|r_i(a)|;\ L_n := min_i\, x_i^n;\ U_n := max_i\, x_i^n;\ \theta := min_{i,j,a}\,\{p_{ij}(a) \mid p_{ij}(a) > 0\}.$$

Lemma 6.17

For all $n = 0, 1, \dots,$ we have $L_{n+1} \geq L_n - k \cdot M$ and $U_{n+1} \leq U_n + k \cdot M$.

Proof

$$\begin{array}{rcl} x^{n+1} & = & T_{f_n}^k\, x^n = r(f_n) + P(f_n)r(f_n) + \cdots + P^{k-1}(f_n)r(f_n) + P^k(f_n)x^n \\ & \geq & -M \cdot e - M \cdot e - \cdots - M \cdot e + P^k(f_n)x^n \geq -k \cdot M \cdot e + L_n \cdot e. \end{array}$$

Hence, $L_{n+1} \geq L_n - k \cdot M$. Similarly it can be shown that $U_{n+1} \leq U_n + k \cdot M$. □

Lemma 6.18

If $span(x^{n+m-1}) \geq span(x^n)$, then for all l with $n \leq l \leq n+m-2$, $L_{l+1} - L_l \leq (2m-3) \cdot k \cdot M$.

Proof

From Lemma 6.17 we obtain

$$\begin{aligned}
span(x^{n+m-1}) &= U_{n+m-1} - L_{n+m-1} \\
&= \textstyle\sum_{j=n}^{n+m-2}\{(U_{j+1}-U_j)-(L_{j+1}-L_j)\} + U_n - L_n \\
&= \textstyle\sum_{j=n}^{n+m-2}(U_{j+1}-U_j) - \sum_{j=n,\ j\neq l}^{n+m-2}(L_{j+1}-L_j) - (L_{l+1}-L_l) + span(x^n) \\
&\leq (m-1)\cdot k\cdot M + (m-2)\cdot k\cdot M - (L_{l+1}-L_l) + span(x^{n+m-1}).
\end{aligned}$$

Hence, $L_{l+1} - L_l \leq (2m-3)\cdot k\cdot M$. □

Lemma 6.19

If $span(x^{n+m-1}) \geq span(x^n)$ and $x_i^{l+1} \leq c + L_{l+1}$ for some $i \in S$, some $n \leq l \leq n+m-2$ and some $c \in \mathbb{R}$, then $x_j^l \leq L_l + \lambda^{1-k}\cdot\theta^{-1}\cdot\{c + 2k\cdot M\cdot(m-1)\}$ for all $j \in S$ for which an action $a \in A(i)$ with $p_{ij}(a) > 0$ exists, where $\lambda \in (0,1)$ is the constant in the strong aperiodicity assumption (5.48).

Proof

Since $(Ux^l)_i = max_a\,\{r_i(a) + \sum_j p_{ij}(a)x_j^l\} \geq -M + max_a\,\{\sum_j p_{ij}(a)x_j^l\}$, for all $i \in S$ and because $x^{l+1} = T_{f_l}^k x^l = T_{f_l}^{k-1}(T_{f_l}x^l) = T_{f_l}^{k-1}(Ux^l)$, we have

$$\begin{aligned}
x^{l+1} &\geq -(k-1)\cdot M\cdot e + P^{k-1}(f_l)(Ux^l) \\
&\geq -(k-1)\cdot M\cdot e + P^{k-1}(f_l)\{-M\cdot e + max_f\,P(f)x^l\} \\
&= -k\cdot M\cdot e + P^{k-1}(f_l)\cdot max_f\,P(f)x^l\}.
\end{aligned}$$

Notice that

$$\begin{aligned}
P^{k-1}(f_l)\cdot max_f\,P(f)x^l &= P^{k-1}(f_l)\cdot\{max_f\,P(f)x^l + L_l\cdot e - L_l\cdot e)\} \\
&= L_l\cdot e + P^{k-1}(f_l)\cdot\{max_f\,P(f)x^l - L_l\cdot e\} \\
&\geq L_l\cdot e + \lambda^{k-1}\cdot\{max_f\,P(f)x^l - L_l e\} \\
&= (1-\lambda^{k-1})\cdot L_l\cdot e + \lambda^{k-1}\cdot max_f\,P(f)x^l.
\end{aligned}$$

Hence, $c + L_{l+1} \geq x_i^{l+1} \geq -k\cdot M + (1-\lambda^{k-1})\cdot L_l + \lambda^{k-1}\cdot max_a\,\sum_j p_{ij}(a)x_j^l$.

Then, by Lemma 6.18,

$$c + L_l + (2m-3)\cdot k\cdot M \geq c + L_{l+1} \geq -k\cdot M + (1-\lambda^{k-1})\cdot L_l + \lambda^{k-1}\cdot max_a\,\textstyle\sum_j p_{ij}(a)x_j^l,$$

i.e. $max_a\,\sum_j p_{ij}(a)\{x_j^l - L_l\} \leq \lambda^{1-k}\cdot\{c + 2(m-1)\cdot k\cdot M\}$.

Take any $j \in S$ for which an action $a \in A(i)$ with $p_{ij}(a) > 0$ exists. Then, we have $p_{ij}(a) \geq \theta$ and $\theta\cdot\{x_j^l - L_l\} \leq p_{ij}(a)\cdot\{x_j^l - L_l\} \leq \lambda^{1-k}\cdot\{c+2(m-1)\cdot k\cdot M\}$, implying $x_j^l \leq L_l + \lambda^{1-k}\cdot\theta^{-1}\cdot\{c+2(m-1)\cdot k\cdot M\}$. □

Define $c_0 := 0$ and $c_n := \lambda^{1-k}\cdot\theta^{-1}\cdot\{c_{n-1} + 2k\cdot M\cdot(N-1)\},\ n = 1,2,\dots,N-1$.

Lemma 6.20

If $span(x^{n+N-1}) \geq span(x^n)$, then $span(x^n) \leq c_{N-1}$.

Proof

Let $i \in S$ such that $x_i^{n+N-1} = L_{n+N-1}$ and define the sets $S(t),\ t = 0,1,\dots,N-1$ by

$$S(0) := \{i\};\ S(t+1) := \{j \in S \mid \exists\ k \in S(t) \text{ and } a \in A(k) \text{ such that } p_{kj}(a) > 0\},\ t = 0,1,\dots,N-2.$$

From $p_{jj}(a) \geq \lambda > 0,\ (j,a) \in S \times A$, it follows that $S(t) \subseteq S(t+1)$. Furthermore, it follows from the communicatingness that $S(N-1) = S$. Then, Lemma 6.19 with $c := c_0 = 0,\ m := N$ and $l := n+N-2$ implies that $x_j^{n+N-2} - L_{n+N-2} \leq c_1$ for all $j \in S(1)$. Next, again by Lemma 6.19 with $c = c_1,\ m = N$

and $l = n + N - 3$, we obtain $x_j^{n+N-3} - L_{n+N-3} \leq c_2$ for all $j \in S(2)$. Continuing in this way, we get $x_j^n - L_n \leq c_{N-1}$ for all $j \in S(N-1) = S$. Hence, $span(x^n) = max\, x_j^n - min\, x_j^n = max\, x_j^n - L_n \leq c_{N-1}$.

□

Finally, we prove in the next theorem that the sequence $\{span(x^n)\}$ is bounded.

Theorem 6.25
$span(x^n) \leq max\{span(x^0) + 2k(N-2)M,\ c_{N-1} + 2k(N-1)M\},\ n = 0, 1, \ldots.$

Proof
By Lemma 6.17, we have
$span(x^n) \leq span(x^{n-1}) + 2kM \leq \cdots \leq span(x^0) + 2knM \leq span(x^0) + 2k(N-2)M,\ n = 0, 1, \ldots, N-2.$
Consequently,

$$span(x^n) \leq max\{span(x^0) + 2k(N-2)M,\ c_{N-1} + 2k(N-1)M\} \text{ for } n = 0, 1, \ldots, N-2. \tag{6.33}$$

Furthermore, also by Lemma 6.17, we obtain

$$\begin{aligned} span(x^{n+N-1}) &= U_{n+N-1} - L_{n+N-1} \\ &\leq (U_{n+N-2} + kM) - (L_{n+N-2} - kM) = U_{n+N-2} - L_{n+N-2} + 2k \\ &\leq (U_{n+N-3} + kM) - (L_{n+N-3} - kM) + 2kM = U_{n+N-3} - L_{n+N-3} + 4kM \\ &\leq \cdots \leq U_n - L_n + 2k(N-1)M = span(x^n) + 2k(N-1)M. \end{aligned}$$

If $span(x^{n+N-1}) \geq span(x^n)$, then by Lemma 6.20 $span(x^n) \leq c_{N-1}$, and consequently $span(x^{n+N-1}) \leq c_{N-1} + 2k(N-1) \cdot M$, implying

$$span(x^{n+N-1}) \leq max\{span(x^n),\ c_{N-1} + 2k(N-1)M\},\ n = 0, 1, \ldots. \tag{6.34}$$

For any $n \geq N-1$, we write $n = q(N-1) + p$ for some $q \geq 1$ and some $0 \leq p \leq N-2$.

Then, by (6.34) and (6.33), we obtain

$$\begin{aligned} span(x^n) &\leq max\{span(x^p),\ c_{N-1} + 2k(N-1)M\} \\ &\leq max\{span(x^0) + 2k(N-2)M,\ c_{N-1} + 2k(N-1)M\}. \end{aligned}$$

□

The proofs for the modified policy iteration method in the unichain and the communicating case depend heavily on the strong aperiodicity assumption. One might wonder whether only aperiodicity, as in the standard value iteration method, would not suffice. The following example demonstrates one of the problems one can encounter under the weaker assumption that all Markov chains $P(f),\ f^\infty \in C(D)$, are aperiodic and unichain.

Example 6.15
$S = \{1, 2, 3, 4, 5, 6, 7\}$; $A(1) = \{1\}$; $A(2) = \{1\}$; $A(3) = \{1, 2\}$; $A(4) = \{1, 2\}$; $A(5) = \{1\}$; $A(6) = \{1, 2\}$; $A(7) = \{1\}$. There are 8 different policies (only in the states 3, 4 and 6 there are two choices).
The transition probabilities are (we give only the strictly positive probabilities):
$p_{12}(1) = p_{13}(1) = \frac{1}{2}$; $p_{23}(1) = 1$; $p_{33}(1) = 1$; $p_{34}(2) = 1$; $p_{41}(1) = 1$; $p_{45}(2) = p_{46}(2) = \frac{1}{2}$; $p_{56}(1) = 1$; $p_{67}(1) = 1$; $p_{63}(2) = 1$; $p_{73}(1) = 1$. It can easily be verified that all policies are unichain and aperiodic.
The rewards are: $r_1(1) = 2$; $r_2(1) = 4$; $r_3(1) = 4$; $r_3(2) = 6$; $r_4(1) = 4$; $r_4(2) = 6$; $r_5(1) = 6$; $r_6(1) = 2$; $r_6(2) = 0$; $r_7(1) = 0$.
Let f_1^∞ be the policy that takes in state 3 action 1, in state 4 action 2 and in state 6 action 1;

let f_2^∞ be the policy that takes in state 3 action 2, in state 4 action 1 and in state 6 action 2;
let f_3^∞ be the policy that takes in state 3 action 2, in state 4 action 2 and in state 6 action 2.
Then, it can be verified that $\phi(f_1^\infty) = \phi(f_2^\infty) = 4$ and $\phi(f_3^\infty) = \frac{30}{7}$ (this is the optimal policy).
Choose $x^0 = (1, 4, 2, 0, 0, 0, 0)$ and take $k = 2$.
Iteration 1:
$Tx^0 = T_{f_1}x^0 = (5, 6, 6, 6, 6, 2, 2)$; $l = 2$; $u = 6$; $x^1 = T_{f_1}^2 x^0 = (8, 10, 10, 10, 8, 4, 6)$.
Iteration 2:
$Tx^1 = T_{f_2}x^1 = (12, 14, 16, 12, 10, 10, 10)$; $l = 2$; $u = 6$; $x^2 = T_{f_2}^2 x^1 = (17, 20, 18, 16, 16, 16, 16)$.
Since $x^2 = x^0 + 16e$ cycling will occur between the two nonoptimal policies f_1^∞ and f_2^∞.

6.4 Bibliographic notes

In the irreducible and unichain case the solution of the optimality equation (6.1) can also be exhibited as the fixed-point of an N-step contraction (cf. Federgruen, Schweitzer and Tijms ([81]).

The policy iteration algorithm 6.1 was introduced by Howard ([134]), where he demonstrated finite convergence under the irreducibility assumption. Various treatments of policy iteration in special cases can also be found in Schweitzer ([255]), Denardo ([61]) and Lassere ([178]). Haviv and Puterman ([114]) have discussed the communicating policy algorithm 6.9.

The pioneering work in solving undiscounted MDPs by linear programming was made by De Ghellinkck ([51]) and Manne ([193]), who independently formulated the linear programs (6.3) and (6.4) for the irreducible case. The relation between the discounted and undiscounted linear programs is described in Nazareth and Kulkarni ([203]. For the unichain case we refer to Denardo and Fox ([64], Denardo ([58]), Derman ([69]) and Kallenberg ([148]). The iterative procedure, with a sequence of unichain linear programs, to determine an average optimal policy regardless the chain structure, was developed by Denardo ([58]). In the irreducible and unichain case also a suboptimality test can be implemented (cf. Hastings ([113]) and Lasserre ([177]). Bello and Raino (citeBello and Riano 06) have built the package JMDP, an object-oriented framework to model and solve discounted and unichained average MDPs in Java. In this package LP-solvers Xpress-MP (see [49]) and QSopt (see [6]) are used. Kallenberg ([148]) and Filar and Schultz ([97]) have developed the linear programming approach for communicating models.

The section on value iteration is taken from Puterman ([227]). Related work is done by Hu and Wu ([136]). Van der Wal ([296] and [297]) analyzed the modified policy iteration method for the irreducible, unichain and communicating case.

6.5 Exercises

Exercise 6.1
Show that $span\,(x) \geq span\,(x - y) - span\,(y)$ for every x and y.

Exercise 6.2
Let P be a unichain Markov chain with stationary distribution π and let $x \geq Px$ or $x \leq Px$.
Prove that $x_i = \pi^T x$ for every recurrent state i.

Exercise 6.3

Consider the following model: $S = \{1, 2, 3\},\ A(1) = A(2) = A(3) = \{1, 2\}$.
$r_1(1) = 1,\ r_1(2) = 0,\ r_2(1) = 2,\ r_2(2) = 1,\ r_3(1) = 1,\ r_3(2) = 2.$ $p_{11}(1) = \frac{1}{2},\ p_{12}(1) = \frac{1}{4},\ p_{13}(1) = \frac{1}{4};$
$p_{11}(2) = \frac{1}{4}, p_{12}(2) = \frac{1}{4}, p_{13}(2) = \frac{1}{2};\ p_{21}(1) = \frac{1}{2}, p_{22}(1) = \frac{1}{2}, p_{23}(1) = 0;\ p_{21}(2) = 0, p_{22}(2) = \frac{1}{2}, p_{23}(2) = \frac{1}{2};$
$p_{31}(1) = \frac{1}{2},\ p_{32}(1) = 0,\ p_{33}(1) = \frac{1}{2};\ p_{31}(2) = 0,\ p_{32}(2) = \frac{1}{2},\ p_{33}(2) = \frac{1}{2}.$

a. Show that this model is unichain, communicating, but not irreducible.

b. Formulate the optimality equation (6.1).

c. Determine an optimal policy by the Policy Iteration Algorithm 6.4, starting with policy f^∞ for which $f(1) = f(2) = f(3) = 1$.

Exercise 6.4

Consider the model of Exercise 6.3. Formulate the primal and dual linear program to solve this unichain model. Apply Algorithm 6.6 to determine the value and an optimal policy.

Exercise 6.5

Consider the model of Exercise 6.3. Execute three iterations of the modified policy iteration algorithm 6.3 with $k = 2$ and $x^0 = (1, 1, 1)$.

Exercise 6.6

For the standard method of value iteration (see section 5.9) the series $\{x^n - n \cdot \phi\}$ is bounded (see Lemma 5.8), even if all policies are periodic. One might wonder whether in the modified policy iteration the series $\{x^n - nk \cdot \phi\}$ is bounded.
Let $S = \{1, 2\};\ A(1) = \{1, 2\},\ A(2) = \{1\};\ r_1(1) = 4,\ r_1(2) = 3,\ r_2(1) = 0;$
$p_{11}(1) = 0,\ p_{12}(1) = 1;\ p_{11}(2) = 1,\ p_{12}(2) = 0;\ p_{21}(1) = 1,\ p_{22}(2) = 0.$
Consider the modified policy iteration algorithm with $k = 2$ and $x^0 = (0, 0)$.
Show that $x^n = (4n, 4n)$ and that $\{x^n - nk \cdot \phi\}$ is unbounded.

Exercise 6.7

Assume there is a state, say state 0, and a scalar $\alpha \in (0, 1)$ such that $p_{i0}(a) \geq 1 - \alpha$ for all $(i, a) \in S \times A$. So, the model is unichain. Consider a new decision process with identical state and action spaces and identical rewards but with transition probabilities given by

$$\overline{p}_{ij}(a) := \begin{cases} \frac{1}{\alpha}\, p_{ij}(a) & j \neq 0; \\ \frac{1}{\alpha}\{p_{i0}(a) - (1 - \alpha)\} & j = 0. \end{cases}$$

Show that the optimality equation of the α-discounted new process is equivalent to the optimality equation (6.1).

Exercise 6.8

Consider an MDP with the *optimal unichain assumption*, i.e. for each optimal stationary policy f^∞ the associated Markov chain $P(f)$ is unichain. For this model the following algorithm is proposed.

1. Determine an optimal solution x^* of the dual linear program

$$max \left\{ \sum_{i,a} r_i(a) x_i(a) \;\middle|\; \begin{array}{lcl} \sum_{i,a} \{\delta_{ij} - p_{ij}(a)\} x_i(a) & = & 0,\ j \in S \\ \sum_{i,a} x_i(a) & = & 1 \\ x_i(a) \geq 0,\ i \in S,\ a \in A(i) & & \end{array} \right\}.$$

2. Choose $f_*(i)$ such that $x^*_i\big(f_*(i)\big) > 0,\ i \in S_* := \{i \mid \sum_a x^*_i(a) > 0\}$.

3. **while** $S_* \neq S$ **do**

   begin select $i \notin S_*,\ j \in S_*,\ f_*(i) \in A(i)$ such that $p_{ij}\big(f_*(i)\big) > 0$; $S_* := S_* \cup \{i\}$ **end**

4. f_*^∞ is an average optimal policy and $\phi = \sum_{i,a} r_i(a)x_i^*(a)$ (STOP).

Prove the correctness of this algorithm.

Exercise 6.9

Consider an MDP with the *weak unichain assumption*, i.e. if all for optimal stationary policies f^∞ and all ergodic sets $E(f)$ of $P(f)$ with $\phi_i(f^\infty) = max_j\, \phi_j(f^\infty)$ for all $i \in E(f)$, there exists a policy g^∞ such that the states of $S \backslash E(f)$ are transient in the Markov chain induced by $P(g)$. Show that for this model the following algorithm is correct.

1. Determine an optimal solution x^* of the dual linear program

$$max \left\{ \sum_{i,a} r_i(a)x_i(a) \;\middle|\; \begin{array}{lll} \sum_{i,a}\{\delta_{ij} - p_{ij}(a)\}x_i(a) & = & 0,\ j \in S \\ \sum_{i,a}\ x_i(a) & = & 1 \\ x_i(a) \geq 0,\ i \in S,\ a \in A(i) & & \end{array} \right\}.$$

2. Choose $f_*(i)$ such that $x_i^*\big(f_*(i)\big) > 0,\ i \in S_* := \{i \mid \sum_a x_i^*(a) > 0\}$.

3. **while** $S_* \neq S$ **do**

   begin select $i \notin S_*,\ j \in S_*,\ f_*(i) \in A(i)$ such that $p_{ij}\big(f_*(i)\big) > 0$; $S_* := S_* \cup \{i\}$ **end**

4. f_*^∞ is an average optimal policy and $\phi = \sum_{i,a} r_i(a)x_i^*(a)$ (STOP).

Chapter 7

More sensitive optimality criteria

7.1 Introduction

In the two previous chapters we have considered the long-run average reward criterion. This criterion ignores transient rewards. The following examples shows why this is (sometimes) undesirable.

Example 7.1

$S = \{1, 2\};\ A(1) = \{1, 2\},\ A(2) = \{1\};\ r_1(1) = 1000,\ r_1(2) = 0,\ r_2(1) = 0;$

$p_{11}(1) = 0,\ p_{12}(1) = 1;\ p_{11}(2) = 0,\ p_{12}(1) = 1;\ p_{21}(1) = 0,\ p_{22}(1) = 1.$

This model has two deterministic policies and both policies are average optimal with average reward 0. However, the policy which chooses in state 1 the first action has a total reward of 1000 and would be preferred. The average reward criterion ignores this distinction.

We address this deficiency of the average reward criterion by more sensitive optimality criteria, the so-called n-discount optimality and the n-average optimality. We may restrict ourselves in this chapter to policies $f^\infty \in C(D)$ by the following arguments:

1. We have shown the existence of a deterministic Blackwell optimal policy.
2. We will show in this chapter (see Lemma 7.2) that a Blackwell optimal policy is n-discount optimal for any $n \geq -1$.
3. It can also be shown that n-discount optimality is equivalent to n-average optimality for all $n \geq -1$.

In section 1.2.2, the concept of *n-discount optimality* for $n = -1, 0, 1, \dots$ is defined as

$$\lim_{\alpha \uparrow 1} (1-\alpha)^{-n} \{v^\alpha(f_*^\infty) - v^\alpha\} = 0. \tag{7.1}$$

By the definition of n-discount optimality the following lemma is obvious.

Lemma 7.1
If a policy is n-discount optimal, then it is m-discount optimal for $m = -1, 0, \dots, n$.

Lemma 7.2
Suppose that f_0^∞ is a Blackwell optimal policy. Then, f_0^∞ is n-discount optimal for any $n \geq -1$.

Proof
Take any $n \geq -1$. Since f_0^∞ is a Blackwell optimal policy, we have $v^\alpha(f_0^\infty) = v^\alpha$ for some $0 < \alpha_0 < 1$ and for all $\alpha \in [\alpha_0, 1)$. Hence, $(1-\alpha)^{-n}\{v^\alpha(f_0^\infty) - v^\alpha\} = 0$ for all $\alpha \in [\alpha_0, 1)$. Therefore, we have $\lim_{\alpha \uparrow 1} (1-\alpha)^{-n}\{v^\alpha(f_0^\infty) - v^\alpha\} = 0$, i.e. f_0^∞ is n-discount optimal. □

Also in section 1.2.2, the concept of *n-average optimality* for $n = -1, 0, 1, \dots$ is defined: a policy R_*, is n-average optimal if

$$\liminf_{T \to \infty} \frac{1}{T} \{v^{n,T}(R_*) - v^{n,T}(R)\} \geq 0 \text{ for every policy } R, \tag{7.2}$$

where the vector $v^{n,T}(R)$ is defined by

$$v^{n,T}(R) := \begin{cases} v^T(R) & \text{for } n = -1 \\ \sum_{t=1}^T v^{n-1,t}(R) & \text{for } n = 0, 1, \dots \end{cases} \tag{7.3}$$

So, (-1)-average optimality is the same as average optimality.

Lemma 7.3
If a policy is n-average optimal, then it is m-average optimal for $m = -1, 0, \dots, n$.

Proof
The proof is left to the reader (see Exercise 7.1).

7.2 Equivalence between n-discount and n-average optimality

Sladky ([273]) has shown that a policy R_* is n-average optimal policy if and only if R_* is n-discount optimal. We will show in this section this result only for $n = -1$ and $n = 0$ in which we restrict ourselves to deterministic policies. In the case of arbitrary policies the notation will be more complicated; for $n \geq 1$ the analysis is much more sophisticated.

For $n = -1$, the criteria (-1)-discount optimality and (-1)-average optimality become

$$\lim_{\alpha\uparrow 1}(1-\alpha)\{v^\alpha(f_*^\infty) - v^\alpha\} = 0 \text{ and } \phi(f_*^\infty) \geq \phi(f^\infty) \text{ for every } f^\infty \in C(D),$$

respectively. The following theorem shows that average optimality is equivalent to (-1)-discount optimality.

Theorem 7.1
Average optimality is equivalent to (-1)*-discount optimality.*

Proof
In Theorem 5.8, part (2), is shown that $\phi(f^\infty) = \lim_{\alpha\uparrow 1}(1-\alpha)v^\alpha(f^\infty)$ for all $f^\infty \in C(D)$. For a Blackwell optimal policy f_0^∞, we obtain $\phi = \lim_{\alpha\uparrow 1}(1-\alpha)v^\alpha$. Let f_*^∞ be (-1)-discount optimal, then $0 = \lim_{\alpha\uparrow 1}(1-\alpha)\{v^\alpha(f_*^\infty) - v^\alpha\} = \phi(f_*^\infty) - \phi$, i.e. $\phi(f_*^\infty)$ is average optimal.
Conversely, let f_*^∞ be an average optimal policy, and let f_0^∞ be Blackwell optimal. Then, we can write

$$0 \geq \lim_{\alpha\uparrow 1}(1-\alpha)\{v^\alpha(f_*^\infty) - v^\alpha\} = \lim_{\alpha\uparrow 1}(1-\alpha)\{v^\alpha(f_*^\infty) - v^\alpha(f_0^\infty)\} = \phi(f_*^\infty) - \phi(f_0^\infty) \geq 0,$$

i.e. $\lim_{\alpha\uparrow 1}(1-\alpha)\{v^\alpha(f_*^\infty) - v^\alpha\} = 0$: f_*^∞ is (-1)-discount optimal. □

For $n = 0$, f_*^∞ is 0-discount optimal and 0-average optimal if

$$\lim_{\alpha\uparrow 1}\{v^\alpha(f_*^\infty) - v^\alpha\} = 0 \text{ and } \liminf_{T\to\infty} \tfrac{1}{T}\textstyle\sum_{t=1}^T\{v^t(f_*) - v^t(f)\} \geq 0 \text{ for all } f^\infty \in C(D),$$

respectively. The following theorem shows that 0-average optimality is equivalent to 0-discount optimality. This criterion is also called *bias optimality.*

Theorem 7.2
0*-average optimality is equivalent to* 0*-discount optimality.*

Proof
Suppose that f_*^∞ is 0-average optimal, and let f_0^∞ be a Blackwell optimal policy. Then, both f_*^∞ and f_0^∞ are average optimal policies. In Theorem 5.8 part (3) we have shown

$$v^t(f^\infty) = \textstyle\sum_{s=1}^t P^{s-1}(f)r(f) = t\cdot\phi(f^\infty) + u^0(f) - P^t(f)D(f)r(f),\ \ f^\infty \in C(D).$$

Hence,

$$\begin{aligned}
0 &\leq \liminf_{T\to\infty} \tfrac{1}{T}\textstyle\sum_{t=1}^T\{v^t(f_*) - v^t(f)\} \\
&= \liminf_{T\to\infty} \tfrac{1}{T}\textstyle\sum_{t=1}^T\big\{\{t\cdot\phi(f_*^\infty) + u^0(f_*) - P^t(f_*)D(f_*)r(f_*)\} - \\
&\qquad\qquad \{t\cdot\phi(f_0^\infty) + u^0(f_0) - P^t(f_0)D(f_0)r(f_0)\}\big\} \\
&= \liminf_{T\to\infty}\big\{\tfrac{1}{2}(T+1)\cdot\{\phi(f_*^\infty) - \phi(f_0^\infty)\} + \{u^0(f_*) - u^0(f_0)\} - \\
&\qquad\qquad \tfrac{1}{T}\textstyle\sum_{t=1}^T\big\{P^t(f_*)D(f_*)r(f_*) - P^t(f_0)D(f_0)r(f_0)\big\}\big\} \\
&= \{u^0(f_*) - u^0(f_0)\} + \liminf_{T\to\infty} \tfrac{1}{T}\textstyle\sum_{t=1}^T\big\{P^t(f_0)D(f_0)r(f_0) - P^t(f_*)D(f_*)r(f_*)\big\}.
\end{aligned}$$

Since $\lim_{T\to\infty}\frac{1}{T}\sum_{t=1}^T P^t(f)D(f) = P^*(f)D(f) = 0$ for every $f^\infty \in C(D)$, we obtain $0 \leq u^0(f_*) - u^0(f_0)$. Then, by the Laurent expansion, we can write

$$\begin{aligned}
0 &\geq \lim_{\alpha\uparrow 1}\{v^\alpha(f_*^\infty) - v^\alpha\} = \lim_{\alpha\uparrow 1})\{v^\alpha(f_*^\infty) - v^\alpha(f_0^\infty)\} \\
&= \lim_{\alpha\uparrow 1}(1-\alpha)^{-1}\{\phi(f_*^\infty) - \phi(f_0^\infty)\} + \{u^0(f_*) - u^0(f_0)\} = u^0(f_*) - u^0(f_0) \geq 0.
\end{aligned}$$

Hence, $\lim_{\alpha\uparrow 1}\{v^\alpha(f_*^\infty) - v^\alpha\} = 0$, i.e. f_*^∞ is 0-discount optimal.
Conversely, suppose that $lim_{\alpha\uparrow 1}\{v^\alpha(f_*^\infty) - v^\alpha\} = 0$. Take any $f^\infty \in C(D)$. Then, by the Laurent expansion, $\phi(f_*^\infty) \geq \phi(f^\infty)$ and if $\phi_i(f_*^\infty) = \phi_i(f^\infty)$ for some $i \in S$, then $u_i^0(f_*) \geq u_i^0(f)$. Hence, we can write

$$\begin{aligned}
&\liminf_{T\to\infty} \tfrac{1}{T}\textstyle\sum_{t=1}^T \{v^t(f_*^\infty) - v^t(f^\infty)\} = \\
&\liminf_{T\to\infty} \Big\{\tfrac{1}{2}(T+1)\{\phi(f_*^\infty) - \phi(f^\infty)\} + \{u^0(f_*) - u^0(f_0)\} - \\
&\qquad\qquad\qquad\qquad \tfrac{1}{T}\textstyle\sum_{t=1}^T \{P^t(f_*)D(f_*)r(f_*) - P^t(f_0)D(f_0)r(f_0)\}\Big\} = \\
&\liminf_{T\to\infty} \Big\{\tfrac{1}{2}(T+1)\{\phi(f_*^\infty) - \phi(f^\infty)\} + \{u^0(f_*) - u^0(f_0)\}\Big\} \geq 0,
\end{aligned}$$

i.e. f_*^∞ is 0-average optimal. □

Example 7.2
$S = \{1,2,3\}$; $A(1) = \{1,2\}$, $A(2) = \{1\}$, $A(3) = \{1\}$; $r_1(1) = 1,\ r_1(2) = 2,\ r_2(1) = 1,\ r_3(1) = 0$;
$p_{11}(1) = 0,\ p_{12}(1) = 1,\ p_{13}(1) = 0$; $p_{11}(2) = p_{12}(2) = 0,\ p_{13}(2) = 1$; $p_{21}(1) = p_{22}(1) = 0,\ p_{23}(1) = 1$; $p_{31}(1) = p_{32}(1) = 0,\ p_{33}(1) = 1$.
This model has two deterministic policies. If we look at the discounted reward, for f_1^∞ with $f_1(1) = 1$, we have $v_1^\alpha(f_1^\infty) = 1+\alpha,\ v_2^\alpha(f_1^\infty) = 1$ and $v_3^\alpha(f_1^\infty) = 0$; for f_2^∞ with $f_2(1) = 2$, we obtain $v_1^\alpha(f_2^\infty) = 2$, $v_2^\alpha(f_1^\infty) = 1$ and $v_3^\alpha(f_1^\infty) = 0$. Hence, $v^\alpha = (2,1,0)$ and f_2^∞ is α-discounted optimal for all discount factors $\alpha \in [0,1)$. For f_1^∞, we have $\lim_{\alpha\uparrow 1}\ (1-\alpha)^{-n}\{v^\alpha(f_1^\infty) - v^\alpha\} = \lim_{\alpha\uparrow 1}\ (1-\alpha)^{-n}(\alpha-1,0,0)$.
For $n = -1$ and $n = 0$ this is equal to $(0,0,0)$, for $n = 1$ the limit is $(-1,0,0)$ and for $n \geq 1$, this limit is the vector $(-\infty,0,0)$. Therefore, f_2^∞ is n-discount optimal for all $n = -1,0,1,\dots$, and f_1^∞ is n-discount optimal only for $n = 0$ and $n = 1$.

7.3 Stationary optimal policies and optimality equations

In this section we use the Laurent series expansion to interpret the n-discount optimality within the class of stationary policies. Furthermore, we provide a system of optimality equations which characterize a stationary n-discount optimal policy. First, we present a lemma which shows that n-discount optimality as defined in (7.1) is equivalent to

$$\liminf_{\alpha\uparrow 1}\ (1-\alpha)^{-n}\{v^\alpha(f_*^\infty) - v^\alpha(f^\infty\} \geq 0 \text{ for all } f^\infty \in C(D). \tag{7.4}$$

Lemma 7.4
The definitions (7.1) and (7.4) for n-discount optimality are equivalent.

Proof
Assume that f_*^∞ is n-discount optimal in the sense of (7.1). Take an arbitrary policy $f^\infty \in C(D)$. Since $v^\alpha(f_*^\infty) \leq v^\alpha$, we have $(1-\alpha)^{-n}\{v^\alpha(f_*^\infty) - v^\alpha(f^\infty\} \geq (1-\alpha)^{-n}\{v^\alpha(f_*^\infty) - v^\alpha\},\ \alpha \in (0,1)$. Hence,

$$\liminf_{\alpha\uparrow 1}\ (1-\alpha)^{-n}\{v^\alpha(f_*^\infty) - v^\alpha(f^\infty\} \geq \liminf_{\alpha\uparrow 1}\ (1-\alpha)^{-n}\{v^\alpha(f_*^\infty) - v^\alpha\} = 0.$$

Conversely, suppose that f_*^∞ satisfies (7.4). Let f_0^∞ be a Blackwell optimal policy. Then, we can write

$$0 \leq \liminf_{\alpha\uparrow 1}\ (1-\alpha)^{-n}\{v^\alpha(f_*^\infty) - v^\alpha(f_0^\infty\} = \liminf_{\alpha\uparrow 1}\ (1-\alpha)^{-n}\{v^\alpha(f_*^\infty) - v^\alpha\} \leq 0.$$

Therefore, $\lim_{\alpha\uparrow 1}\ (1-\alpha)^{-n}\{v^\alpha(f_*^\infty) - v^\alpha\} = 0$. □

Instead of the discount factor α we can also use the interest rate ρ, where the relation between the two are given by $\alpha := \frac{1}{1+\rho}$ or $\rho := \frac{1-\alpha}{\alpha}$. Since $\lim_{\alpha\uparrow 1} (1-\alpha)^{-n}\, v^\alpha(f^\infty) = \lim_{\alpha\uparrow 1} (\frac{1-\alpha}{\alpha})^{-n}\, v^\alpha(f^\infty)$ and $\alpha \uparrow 1$ if and only if $\rho \downarrow 0$, the concept of n-discount optimality is equivalent to

$$\liminf_{\rho\downarrow 0}\ \rho^{-n}\{v^\rho(f_*^\infty) - v^\rho(f^\infty)\} \geq 0 \text{ for all } f^\infty \in C(D).$$

In Theorem 5.10 we have shown that $\alpha v^\alpha(f^\infty) = \sum_{k=-1}^\infty \rho^k u^k(f)$ for $0 < \rho \leq \|D(f)\|^{-1}$. Hence, as function of the interest rate ρ, the total expected discounted reward is written as

$$v^\rho(f^\infty) = (1+\rho) \cdot \sum_{k=-1}^\infty \rho^k u^k(f). \tag{7.5}$$

Let $F_\infty := \{f_0^\infty \mid f_0^\infty \text{ is Blackwell optimal}\}$ and $F_n := \{f_*^\infty \mid f_*^\infty \text{ is } n\text{-discount-optimal}\}$ for $n \geq -1$. We have seen that $F_{n+1} \subseteq F_n$ for $n = -1, 0, 1, \dots$ and that $f_*^\infty \in F_{-1}$ if and only if $u^{-1}(f_*) \geq u^{-1}(f)$ for all $f^\infty \in C(D)$. In the next theorem is shown that $F_n = \{f_*^\infty \in F_{n-1} \mid u^n(f_*) \geq u^n(f),\ f^\infty \in F_{n-1}\}$ for all $n \geq 0$ and that $F_\infty = \cap_{n=-1}^\infty F_n$.

Theorem 7.3

$F_n = \{f_*^\infty \in F_{n-1} \mid u^n(f_*) \geq u^n(f)$ for all $f^\infty \in F_{n-1}\}$ for all $n \geq 0$ and $F_\infty = \cap_{n=-1}^\infty F_n$.

Proof

We use induction on n. Let $f_*^\infty \in F_0$, i.e. $\liminf_{\rho\downarrow 0} \{v^\rho(f_*^\infty) - v^\rho(f^\infty)\} \geq 0$ for all $f^\infty \in C(D)$. Take an arbitrary policy $f^\infty \in F_{-1}$. Then, f_*^∞ and f^∞ are both average optimal policies, i.e. $u^{-1}(f_*) = u^{-1}(f)$. From (7.5) it follows that

$$v^\rho(f_*^\infty) - v^\rho(f^\infty) = (1+\rho) \cdot \big\{\{u^0(f_*) - u^0(f)\} + \sum_{k=1}^\infty \rho^k\{u^k(f_*) - u^k(f)\}\big\}. \tag{7.6}$$

Hence, $0 \leq \liminf_{\rho\downarrow 0} \{v^\rho(f_*^\infty) - v^\rho(f^\infty)\} = u^0(f_*) - u^0(f)$. Consequently, $u^0(f_*) \geq u^0(f)$.

Conversely, suppose that $f_*^\infty \in F_{-1}$ and that $u^0(f_*) \geq u^0(f)$ for all $f^\infty \in F_{-1}$. Notice that

$$\liminf_{\rho\downarrow 0} \{v^\rho(f_*^\infty) - v^\rho(f^\infty)\} = \liminf_{\rho\downarrow 0} \ \big\{\tfrac{1}{\rho}\{u^{-1}(f_*) - u^{-1}(f)\} + \{u^0(f_*) - u^0(f)\}\big\}.$$

Since $u^{-1}(f_*) \geq u^{-1}(f),\ f^\infty \in C(D)$ and $u^0(f_*) \geq u^0(f)$ for all $f^\infty \in F_{-1}$, we have

$$\liminf_{\rho\downarrow 0} \{v^\rho(f_*^\infty) - v^\rho(f^\infty)\} \geq 0 \text{ for all } f^\infty \in C(D),$$

i.e. f_*^∞ is 0-discount optimal. The proof of the induction step follows by similar arguments and is left to the reader.

Suppose that f_*^∞ is Blackwell optimal. Then, by Lemma 7.2, f_*^∞ is n-discount optimal for $n = -1, 0, 1, \dots$. So, $f_*^\infty \in \cap_{n=-1}^\infty F_n$. Finally, let $f_*^\infty \in \cap_{n=-1}^\infty F_n$. Select an arbitrary $f^\infty \in C(D)$. If $f^\infty \in \cap_{n=-1}^\infty F_n$, then $u^n(f_*) = u^n(f)$ for $n = -1, 0, 1, \dots$, and consequently $v^\rho(f_*^\infty) = v^\rho(f^\infty)$ for all $\rho > 0$. In case $f^\infty \notin \cap_{n=-1}^\infty F_n$, then $f^\infty \notin F_n$ for some n and let n the minimal integer for which this holds. Then,

$v^\rho(f_*^\infty) - v^\rho(f^\infty) = (1+\rho)\rho^n \cdot \big\{[u^n(f_*) - u^n(f)] + \sum_{k=n+1}^\infty \rho^k \cdot [u^k(f_*) - u^k(f)]\big\}$ with $u^n(f_*) > u^n(f)$.

Hence, we can find a $\rho(f)$ such that $v^\rho(f_*^\infty) \geq v^\rho(f^\infty)$ for $0 < \rho \leq \rho(f)$. Since $C(D)$ is finite, there exists a ρ_* such that $v^\rho(f_*^\infty) \geq v^\rho(f^\infty)$ for $0 < \rho \leq \rho_*$ for all $f^\infty \in C(D)$, i.e. f_*^∞ is Blackwell optimal. □

Remarks

1. $f_*^\infty \in F_n$ if and only if $\big(u^{-1}(f), u^0(f), \dots, u^n(f)\big)$ is lexicographically the largest vector over the set vectors $\big(u^{-1}(g), u^0(g), \dots, u^n(g)\big)$ where $g^\infty \in C(D)$.
2. Suppose that, for some $n \geq -1$, F_n contains a single policy f_*^∞. Then, f_*^∞ is a Blackwell optimal policy.

Next, we will derive the optimality equations by using the optimality properties of a Blackwell optimal policy. Suppose that f_0^∞ is a Blackwell optimal policy. Then, by definition, f_0^∞ is discounted optimal for each $\rho \in (0, \rho_*)$ for some ρ_*. Hence, f_0^∞ satisfies

$$max_{f^\infty \in C(D)} \Big\{r(f) + \big\{\frac{1}{1+\rho} \cdot P(f) - I\big\} v^\alpha(f_0^\infty)\Big\} = 0.$$

Noting that $\frac{1}{1+\rho}\cdot P(f) - I = \frac{1}{1+\rho}\cdot\{[P(f)-I]-\rho\cdot I\}$ and that $v^\alpha(f_0^\infty)$ has the Laurent series expansion for ρ near to 0, we obtain the equation

$$max_{f^\infty\in C(D)}\Big\{r(f) + \big[P(f)-I-\rho\cdot I\big]\sum_{k=-1}^{\infty}\rho^k u^k(f_0)\Big\} = 0.$$

Rearranging terms yields

$$max_{f^\infty\in C(D)}\Big\{[P(f)-I]\tfrac{u^{-1}(f_0)}{\rho} + \{r(f)-u^{-1}(f_0)+[P(f)-I]u^0(f_0)\} + \\ \textstyle\sum_{k=1}^\infty \rho^k\{-u^{k-1}(f_0)+[P(f)-I]u^k(f_0)\}\Big\}=0.$$

For the above equality to hold for all ρ near 0 it requires that:

1. $[P(f)-I]u^{-1}(f_0)\le 0$ for all policies $f^\infty\in C(D)$.
2. For those f for which $\{[P(f)-I]u^{-1}(f_0)\}_i=0$ for some $i\in S$, we have the requirement $r_i(f)-u_i^{-1}(f_0)+\{P(f)u^0(f_0)\}_i-u_i^0(f_0)\le 0$.
3. For those f for which $\{[P(f)-I]u^{-1}(f_0)\}_i=0$ and $\{r(f)-u^{-1}(f_0)+[P(f)-I]u^0(f_0)\}_i=0$ for some $i\in S$, we have $-u_i^0(f_0)+[P(f)u^1(f_0)-I]_i\le 0$.

In this way one can formulate the requirements.

This observation shows that the following system of inductively defined equations characterizes the coefficients of the Laurent series expansion of a Blackwell optimal policy:

$$max_{a\in A(i)}\Big\{\sum_j p_{ij}(a)x_j^{-1}-x_i^{-1}\Big\}=0. \tag{7.7}$$

$$max_{a\in A^{(-1)}(i,x^{-1})}\Big\{r_i(a)+\sum_j p_{ij}(a)x_j^0-x_i^0-x_i^{-1}\Big\}=0 \tag{7.8}$$

$$max_{a\in A^{(k-1)}(i,x^{-1},x^0,\dots,x^{k-1})}\Big\{\sum_j p_{ij}(a)x_j^k-x_i^k-x_i^{k-1}\Big\}=0,\ k=1,2,\dots \tag{7.9}$$

where

$A^{(-1)}(i,x^{-1}) := argmax_{a\in A(i)}\big\{\sum_j p_{ij}(a)x_j^{-1}-x_i^{-1}\big\}$;

$A^{(0)}\big(i,x^{-1},x^0\big) := argmax_{a\in A^{(-1)}(i,x^{-1})}\big\{r_i(a)+\sum_j p_{ij}(a)x_j^0-x_i^0-x_i^{-1}\big\}$;

$A^{(k-1)}\big(i,x^{-1},x^0,\dots,x^{k-1}\big) := argmax_{a\in A^{(k-2)}(i,x^{-1},x^0,\dots,x^{k-2})}\big\{\sum_j p_{ij}(a)x_j^{k-1}-x_i^{k-1}-x_i^{k-2}\big\},\ k\ge 2.$

We refer to this system as the *sensitive discount optimality equations* and to the individual equations as the (-1)th equation, 0th equation, 1th equation, etc. The sets of maximizing decision rules depend on the sequence of the x^k and consequently the system of equations is highly non-linear. Observe that the (-1)th and the 0th equation are the multichain average reward optimality equations. From the results of Chaper 5 it follows that if x^{-1} and x^0 satisfy these two equations and $f(i)\in A^{(-1)}(i,x^{-1})$ for all $i\in S$, then $x^{-1}=\phi$ and f^∞ is an average optimal policy. We generalize this observation to n-discount optimality below.

Theorem 7.4

If the vector $(x^{-1},x^0,\dots,x^n)$ satisfies the following linear system

$$\left\{\begin{array}{lllll}\{I-P(f)\}x^{-1} & & & = & 0\\ x^{-1} & + & \{I-P(f)\}x^0 & = & r(f)\\ & & x^{k-1} + \{I-P(f)\}x^k & = & 0,\ 1\le k\le n\end{array}\right.$$

then $x^k=u^k(f)$ for $k=-1,0,1,\dots,n-1$ and if, in addition, either $-x^n+\{I-P(f)\}x^{n+1}=0$ or $P^*(f)x^n=0$, then $x^n=u^n(f)$.

Proof

Notice that $-x^n + \{I - P(f)\}x^{n+1} = 0$ implies $P^*(f)x^n = 0$. So, it is sufficient to consider the case with $P^*(f)x^n = 0$. It is straightforward to see that $\big(u^{-1}(f), u^0(f), \dots, u^n(f)\big)$ is a solution. Consider an arbitrary solution $\big(x^{-1}, x^0, \dots, x^n\big)$. Then, $\{I - P(f)\}x^{-1} = 0$ implies $x^{-1} = P^*(f)x^{-1}$, and consequently (from the second equation of the system), we get $x^{-1} = P^*(f)r(f) = u^{-1}(f)$. From the third equation, for $k = 1$, it follows that $P^*(f)x^0 = 0$, and from the second equation, $\{I - P(f) + P^*(f)\}x^0 = \{I - P^*(f)\}r(f)$. Therefore, we have

$$x^0 = \{I - P(f) + P^*(f)\}^{-1}\{I - P^*(f)\}r(f) = \{D(f) + P^*(f)\}\{I - P^*(f)\}r(f) = D(f)r(f) = u^0(f).$$

By induction on k, we will show that $x^k = u^k(f),\ k \geq 1$ (for $k \leq 0$ this is shown above). Assume that $x^{-1} = u^{-1}(f), x^0 = u^0(f), \dots, x^k = u^k(f)$. Since $x^k + \{I - P(f)\}x^{k+1} = 0$ and $P^*(f)x^{k+1} = 0$, we obtain

$$\begin{aligned} x^{k+1} &= -\{I - P(f) + P^*(f)\}^{-1}u^k(f) = -\{D(f) + P^*(f)\}(-1)^k\{D(f)\}^{k+1}r(f) \\ &= (-1)^{k+1}\{D(f)\}^{k+2}r(f) = u^{k+1}(f). \end{aligned}$$

□

7.4 Lexicographic ordering of Laurent series

Let, for $0 < \rho \leq \|D(f)\|^{-1}$, the matrix $H^\rho(f)$ be defined by

$$H^\rho(f) := (1 + \rho) \cdot \Big\{P^*(f) + \sum_{k=0}^{\infty} (-1)^k \rho^{k+1} D^{k+1}(f)\Big\}. \tag{7.10}$$

Theorem 7.5

(1) $H^\rho(f) = \rho \cdot \big\{I - \frac{1}{1+\rho}P(f)\big\}^{-1}$.

(2) $H^\rho(f)r(f) = \rho \cdot v^\rho(f^\infty)$.

(3) $\rho \cdot \{v^\rho(f^\infty) - x\} = H^\rho(f)\big\{r(f) + \frac{1}{1+\rho}P(f)x - x\big\}$ *for every* $x \in \mathbb{R}^N$.

Proof

(1)
$$\begin{aligned} \big\{I - \tfrac{1}{1+\rho}P(f)\big\}H^\rho(f) &= \big\{(1+\rho)I - P(f)\big\}\tfrac{1}{1+\rho}H^\rho(f) \\ &= \{(1+\rho)I - P(f)\}\Big\{P^*(f) + \textstyle\sum_{k=0}^\infty (-1)^k\rho^{k+1}D^{k+1}(f)\Big\} \\ &= \rho \cdot \big\{P^*(f) + \textstyle\sum_{k=0}^\infty (-1)^k\rho^{k+1}D^{k+1}(f)\big\} + \\ &\qquad \{I - P(f)\}\big\{P^*(f) + \textstyle\sum_{k=0}^\infty (-1)^k\rho^{k+1}D^{k+1}(f)\big\} \\ &= \rho \cdot \big\{P^*(f) + \textstyle\sum_{k=0}^\infty (-1)^k\rho^{k+1}D^{k+1}(f)\big\} + \\ &\qquad \{I - P(f)\}D(f)\textstyle\sum_{k=0}^\infty (-1)^k\rho^{k+1}D^k(f) \\ &= \rho \cdot \cdot P^*(f) + \textstyle\sum_{k=0}^\infty (-1)^k\rho^{k+2}D^{k+1}(f)\} + \\ &\qquad \{I - P^*(f)\}\textstyle\sum_{k=0}^\infty (-1)^k\rho^{k+1}D^k(f) \\ &= \rho \cdot P^*(f) + \textstyle\sum_{k=1}^\infty (-1)^{k-1}\rho^{k+1}D^k(f)\} + \\ &\qquad \rho \cdot \{I - P^*(f)\} + \textstyle\sum_{k=1}^\infty (-1)^k\rho^{k+1}D^k(f) \\ &= \rho \cdot I. \end{aligned}$$

(2) $\rho \cdot v^\rho(f^\infty) = \rho \cdot \{I - \frac{1}{1+\rho}P(f)\big\}^{-1}r(f) = H^\rho(f)r(f)$.

(3)
$$\begin{aligned} H^\rho(f)\big\{r(f) + \tfrac{1}{1+\rho}P(f)x - x\big\} &= \rho \cdot v^\rho(f^\infty) - H^\rho(f)\big\{I - \tfrac{1}{1+\rho}P(f)\big\}x \\ &= \rho \cdot v^\rho(f^\infty) - \rho \cdot x = \rho \cdot \{v^\rho(f^\infty) - x\}. \end{aligned}$$

□

Define the sets LS_1 and LS_2 of Laurent series by

$$LS_1 := \big\{u(\rho) \mid u(\rho) := \textstyle\sum_{k=-1}^\infty \rho^k u^k;\ u^k \in \mathbb{R}^N,\ k \geq -1;\ \limsup_{k\to\infty} \|u^k\|^{1/k} < \infty\big\}.$$

$$LS_2 := \big\{x(\rho) \mid x(\rho) := (1+\rho) \cdot \textstyle\sum_{k=-1}^\infty \rho^k x^k;\ x^k \in \mathbb{R}^N,\ k \geq -1;\ \limsup_{k\to\infty} \|x^k\|^{1/k} < \infty\big\}.$$

Lemma 7.5
$LS_1 = LS_2$.

Proof
Take any $u(\rho) \in LS_1$. Then, since $(1+\rho)^{-1} = 1 - \rho + \rho^2 - \rho^3 + \cdots$ (for $|\rho| < 1$), we have

$$u(\rho) = (1+\rho) \cdot \{\textstyle\sum_{j=0}^\infty (-\rho)^j\} \sum_{k=-1}^\infty \rho^k u^k = (1+\rho) \cdot \sum_{k=-1}^\infty \sum_{j=0}^\infty (-1)^j \rho^{k+j} u^k.$$

Let $i = k + j$, then $i = -1, 0, 1, \dots$ and $k \leq i$. Therefore, we may write

$$u(\rho) = (1+\rho) \cdot \textstyle\sum_{i=-1}^\infty \rho^i \{\sum_{k=-1}^i (-1)^{i-k} u^k\} = (1+\rho) \cdot \sum_{i=-1}^\infty \rho^i x^i,$$

where $x^i := \sum_{k=-1}^i (-1)^{i-k} u^k$. Because $\|x^i\| \leq \sum_{k=-1}^i \|u^k\| \leq (i+2) \cdot max_{-1 \leq k \leq i} \|u^k\|$, we obtain $\|x^i\|^{1/i} \leq (i+2)^{1/i} \cdot \big\{max_{-1\leq k\leq i} \|u^k\|\big\}^{1/i}$, and consequently

$$\limsup_{i\to\infty} \|x^i\|^{1/i} \leq \{\limsup_{i\to\infty} (i+2)^{1/i}\} \cdot \{\limsup_{i\to\infty} \|u^i\|^{1/i}\} = \limsup_{i\to\infty} \|u^i\|^{1/i} < \infty,$$

i.e. $u(\rho) \in LS_2$.
Conversely, let $x(\rho) \in LS_2$. Then,

$$x(\rho) = (1+\rho) \cdot \textstyle\sum_{k=-1}^\infty \rho^k x^k = \rho^{-1} x^{-1} + \sum_{k=0}^\infty \rho^k \cdot \{x^k + x^{k-1}\} = \sum_{k=-1}^\infty \rho^k u^k,$$

where $u^{-1} := x^{-1}$ and $u^k := x^k + x^{k-1},\ k \geq 0$. Since $\limsup_{k\to\infty} \|u^k\|^{1/k} \leq 2 \cdot \limsup_{k\to\infty} \|k^k\|^{1/k} < \infty$, we have $x(\rho) \in LS_1$. □

Because the sets LS_1 and LS_2 are identical, we denote this set as LS. Notice that LS is a linear vector space. We define a *lexicographic ordering* on LS: $u(\rho)$ is *nonnegative (nonpositive)* if the first nonzero vector of $(u^{-1}, u^0, u^1, \dots)$ is nonnegative (nonpositive), i.e.

$$\begin{cases} u(\rho) \geq_l 0 & \text{if } \liminf_{\rho\downarrow 0}\ \rho^{-k} u(\rho) \geq 0 \text{ for } k = -1, 0, 1, \dots. \\ u(\rho) >_l 0 & \text{if } u(\rho) \geq_l 0 \text{ and } u(\rho) \neq 0. \end{cases}$$

For $f^\infty \in C(D)$, let $L_f^\rho : LS \to LS$ be defined by $L_f^\rho\, x(\rho) := r(f) + (1+\rho)^{-1} \cdot P(f)x(\rho)$. The Laurent expansion of $L_f^\rho\, x(\rho) - x(\rho)$ becomes:

$$\begin{aligned} L_f^\rho\, x(\rho) - x(\rho) &= r(f) + (1+\rho)^{-1} \cdot P(f)\big\{(1+\rho) \textstyle\sum_{k=-1}^\infty \rho^k x^k\big\} - (1+\rho) \cdot \sum_{k=-1}^\infty \rho^k x^k \\ &= r(f) + \textstyle\sum_{k=-1}^\infty \rho^k P(f) x^k - \sum_{k=-1}^\infty \rho^k x^k - \sum_{k=-1}^\infty \rho^{k+1} x^k \\ &= r(f) + \textstyle\sum_{k=-1}^\infty \rho^k P(f) x^k - \sum_{k=-1}^\infty \rho^k x^k - \sum_{k=0}^\infty \rho^k x^{k-1} \\ &= \rho^{-1} \cdot \{P(f)x^{-1} - x^{-1}\} + \{r(f) + P(f)x^0 - x^0 - x^{-1}\} \\ &\qquad + \textstyle\sum_{k=1}^\infty \rho^k \cdot \{P(f)x^k - x^k - x^{k-1}\}. \end{aligned}$$

The equation above implies that $L_f^\rho\, x(\rho) \in LS$. The next theorem shows that $v^\rho(f^\infty)$ is a fixed-point of L_f^ρ.

Lemma 7.6
$L_f^\rho\, v^\rho(f^\infty) - v^\rho(f^\infty) = 0$.

Proof

For $x = v^\rho(f^\infty) = (1+\rho)\cdot\sum_{k=-1}^{\infty}\rho^k u^k(f)$, we have $x^k = u^k(f),\ k=-1,0,1,\dots$. Hence,

$$L_f^\rho\, x(\rho) - x(\rho) = \rho^{-1}\cdot\{P(f)u^{-1}(f) - u^{-1}(f)\} + \{r(f) + P(f)u^0(f) - u^0(f) - u^{-1}(f)\} \\ + \textstyle\sum_{k=1}^{\infty}\rho^k\cdot\{P(f)u^k(f) - u^k(f) - x^{k-1}(f)\}.$$

To establish the fixed-point, note that

$$P(f)u^{-1}(f) - u^{-1}(f) = \{P(f)P^*(f) - P^*(f)\}r(f) = 0.$$

$$r(f) + P(f)u^0(f) - u^0(f) - u^{-1}(f) = \{I + P(f)D(f) - D(f) - P^*(f)\}r(f) = 0.$$

$$P(f)u^k(f) - u^k(f) - u^{k-1}(f) = (-1)^{k-1}\{D(f)\}^k\{-P(f)D(f) + D(f) - I\}r(f) \\ = (-1)^{k-1}\{D(f)\}^k\{-P^*(f)\}r(f) = 0,\ k \geq 1.$$

□

Consider the mapping $B : LS \rightarrow LS$, where for $x(\rho) := (1+\rho)\cdot\sum_{k=-1}^{\infty}\rho^k x^k,\ Bx(\rho)$ is defined by

$$Bx(\rho) := \textstyle\sum_{k=-1}^{\infty}\rho^k B^{(k)}\big(x^{-1}, x^0, \dots, x^k\big)$$

with

$$\big\{B^{(-1)}\big(x^{-1}\big)\big\}_i := max_{a\in A(i)}\big\{\textstyle\sum_j p_{ij}(a)x_j^{-1} - x_i^{-1}\big\} \text{ and}$$

$$A^{(-1)}\big(i, x^{-1}\big) := argmax_{a\in A(i)}\big\{\textstyle\sum_j p_{ij}(a)x_j^{-1} - x_i^{-1}\big\},\ i \in S;$$

$$\big\{B^{(0)}\big(x^{-1}, x^0\big)\big\}_i := max_{a\in A^{(-1)}(i,x^{-1})}\big\{r_i(a) + \textstyle\sum_j p_{ij}(a)x_j^0 - x_i^0 - x_i^{-1}\big\} \text{ and}$$

$$A^{(0)}\big(i, x^{-1}, x^0\big) := argmax_{a\in A^{(-1)}(i,x^{-1})}\big\{r_i(a) + \textstyle\sum_j p_{ij}(a)x_j^0 - x_i^0 - x_i^{-1}\big\},\ i \in S;$$

and for $k \geq 1$

$$\big\{B^{(k)}\big(x^{-1}, x^0, \dots, x^k\big)\big\}_i := max_{a\in A^{(k-1)}(i,x^{-1},x^0,\dots,x^{k-1})}\big\{\textstyle\sum_j p_{ij}(a)x_j^k - x_i^k - x_i^{k-1}\big\} \text{ and}$$

$$A^{(k)}\big(i, x^{-1}, x^0, \dots, x^k\big) := argmax_{a\in A^{(k-1)}(i,x^{-1},x^0,\dots,x^{k-1})}\big\{\textstyle\sum_j p_{ij}(a)x_j^k - x_i^k - x_i^{k-1}\big\},\ i \in S.$$

Since we have derived that

$$\rho^{-1}\{P(f)x^{-1} - x^{-1}\} + \{r(f) + P(f)x^0 - x^0 - x^{-1}\} + \textstyle\sum_{k=1}^{\infty}\rho^k\{P(f)x^k - x^k - x^{k-1}\} =$$

$$L_f^\rho\, x(\rho) - x(\rho) = r(f) + (1+\rho)^{-1}P(f)x(\rho) - x(\rho) \in LS \text{ for all } f^\infty \in C(D),$$

$Bx(\rho)$ is an element of LS which is the result of lexicographically maximizing the elements $r(f) + (1+\rho)^{-1}P(f)x(\rho) - x(\rho)$ over the set $C(D)$, i.e.

$$Bx(\rho) = lexmax_{f^\infty\in C(D)}\{r(f) + (1+\rho)^{-1}P(f)x(\rho) - x(\rho)\} = lexmax_{f^\infty\in C(D)}\{L_f^\rho\, x(\rho) - x(\rho)\}. \quad (7.11)$$

Because, by Lemma 7.6, $L_g^\rho\, v^\rho(g^\infty) - v^\rho(g^\infty) = 0$ for all $g^\infty \in C(D)$, we obtain for all $g^\infty \in C(D)$,

$$Bv^\rho(g^\infty) = lexmax_{f^\infty\in C(D)}\{L_f^\rho\, v^\rho(g^\infty) - v^\rho(g^\infty)\} \geq_l L_g^\rho\, v^\rho(g^\infty) - v^\rho(g^\infty) = 0, \quad (7.12)$$

i.e. $Bv^\rho(g^\infty)$ is lexicographically nonnegative for all $g^\infty \in C(D)$.

Lemma 7.7

$H^\rho(f)\{r(f) + (1+\rho)^{-1}P(f)v^\rho(g^\infty) - v^\rho(g^\infty)\} = \rho\cdot\{v^\rho(f^\infty) - v^\rho(g^\infty)\}$.

Proof

By Theorem 7.5 part (3), we obtain $H^\rho(f)\{r(f)+(1+\rho)^{-1}P(f)v^\rho(g^\infty)-v^\rho(g^\infty)\} = \rho\cdot\{v^\rho(f^\infty)-v^\rho(g^\infty)\}$.

□

Next, we will show that $H^\rho(f)$ is a *positive operator* for every $f^\infty \in C(D)$, i.e.

$$H^\rho(f)\{u(\rho)\} \geq_l 0 \text{ if } u(\rho) \geq_l 0 \text{ and } H^\rho(f)\{u(\rho)\} >_l 0 \text{ if } u(\rho) >_l 0.$$

Theorem 7.6
$H^\rho(f)$ is a positive operator for all $f^\infty \in C(D)$.

Proof

$$\begin{aligned} H^\rho(f)u(\rho) &= \rho \cdot \left\{I - (1+\rho)^{-1}P(f)\right\}^{-1} u(\rho) = \rho \cdot \textstyle\sum_{t=0}^\infty \{(1+\rho)^{-1}P(f)\}^t u(\rho) \\ &= \rho \cdot u(\rho) + \rho \cdot \textstyle\sum_{t=1}^\infty \{(1+\rho)^{-1}P(f)\}^t u(\rho). \end{aligned}$$

Hence, it is sufficient to show that $P^t(f)u(\rho) \geq_l 0$ for all $t \geq 1$ and all $u(\rho) \geq_l 0$. Take any $t \geq 1$, any $i \in S$ and let $T(i) := \{j \in S \mid p^t_{ij}(f) > 0\}$. Suppose that k is such that $\{P^t(f)u^m\}_i = 0, \ -1 \leq m \leq k-1$, and $\{P^t(f)u^k\}_i \neq 0$. Because $0 = \{P^t(f)u^m\}_i = \sum_{j \in T(i)} p^t_{ij}(f)u^m_j$, we have $u^m_j = 0$ for all $j \in T(i)$ and all $m = -1, 0, \dots, k-1$. Since $u(\rho) \geq_l 0$, we have $u^k_j \geq 0$ for all $j \in T(i)$ and consequently, $\{P^t(f)u^k\}_i = \sum_{j \in T(i)} p^t_{ij}(f)u^k_j \geq 0$. Because $\{P^t(f)u^k\}_i \neq 0$, we get $\{P^t(f)u^k\}_i > 0$, i.e. $P^t(f)u(\rho) \geq_l 0$.

□

Theorem 7.7

(1) The equation $Bx = 0, \ x \in LS$, has a unique solution $x = v^\rho(f_0^\infty)$, where f_0^∞ is a Blackwell optimal policy.

(2) If $f^\infty \in C(D)$ satisfies $Bx = r(f) + (1+\rho)^{-1}P(f)x - x = 0$, then f^∞ is Blackwell optimal.

Proof

Let f_0^∞ be a Blackwell optimal policy, i.e. $v^\rho(f_0^\infty) \geq_l v^\rho(f^\infty)$ for all $f^\infty \in C(D)$. We first show that $Bv^\rho(f_0^\infty) = 0$ and $r(f) + (1+\rho)^{-1}P(f)v^\rho(f_0^\infty) - v^\rho(f_0^\infty) \leq_l 0, \ f^\infty \in C(D)$. In (7.12) it is shown that $Bv^\rho(f^\infty) \geq_l 0$ for all $f^\infty \in C(D)$. Hence, $Bv^\rho(f_0^\infty) \geq_l 0$. Suppose that $Bv^\rho(f_0^\infty) >_l 0$. Then, from (7.11), it follows that there is a policy f^∞ satisfying $r(f) + (1+\rho)^{-1}P(f)v^\rho(f_0^\infty) - v^\rho(f_0^\infty) >_l 0$. Then, by Theorem 7.6 and Lemma 7.7, we obtain $H^\rho(f)\{r(f) + (1+\rho)^{-1}P(f)v^\rho(f_0^\infty) - v^\rho(f_0^\infty)\} = \rho \cdot \{v^\rho(f^\infty) - v^\rho(f_0^\infty)\} >_l 0$, contradicting the Blackwell optimality of f_0^∞. Hence, we have shown that

$$Bv^\rho(f_0^\infty) = 0 \text{ and } r(f) + (1+\rho)^{-1}P(f)v^\rho(f_0^\infty) - v^\rho(f_0^\infty) \leq_l 0, \ f^\infty \in C(D). \tag{7.13}$$

Next, suppose that $Bx = 0$ for some $x \in LS$. Then, $r(f_0) + (1+\rho)^{-1}P(f_0)x - x \leq_l 0$. Since $H^\rho(f_0)$ is a positive operator, we obtain by Theorem 7.5 part (3),

$$0 \geq_l H^\rho(f_0)\{r(f_0) + (1+\rho)^{-1}P(f_0)x - x\} = \rho \cdot \{v^\rho(f_0^\infty) - x\}, \text{ i.e. } v^\rho(f_0^\infty) \leq_l x. \tag{7.14}$$

Therefore, $v^\rho(f_0^\infty)$ is the lexicographically smallest solution of the functional equation $Bx = 0$. Finally, suppose that $Bx = r(f) + (1+\rho)^{-1}P(f)x - x = 0$ for some $f^\infty \in C(D)$. Then, we obtain

$$0 = H^\rho(f)\{r(f) + (1+\rho)^{-1}P(f)x - x\} = \rho \cdot \{v^\rho(f^\infty) - x\}, \text{ i.e. } v^\rho(f^\infty) = x. \tag{7.15}$$

Combining (7.14) and (7.15) gives $v^\rho(f^\infty) \geq_l v^\rho(f_0^\infty)$. Since f_0^∞ is Blackwell optimal, we also have $v^\rho(f_0^\infty) \geq_l v^\rho(f^\infty)$, i.e. $v^\rho(f^\infty) = v^\rho(f_0^\infty)$, implying that f^∞ is also a Blackwell optimal policy and the functional equation $Bx = 0$ has a unique solution $x = v^\rho(f_0^\infty)$.

□

7.5 Policy iteration for n-discount optimality

In this section we derive for any $n \in \mathbb{N}$ a policy iteration algorithm which computes a policy that lexicographically maximes the vector $\big(u^{-1}(f), u^0(f), \dots, u^n(f)\big)$ over all $f^\infty \in C(D)$, i.e. an n-discount optimal policy. Furthermore, we will show that any n-discount optimal policy for $n \geq N-1$ is a Blackwell optimal policy.

Algorithm 7.1 *Determination of an n-discount optimal policy by policy iteration*
Input: Instance of an MDP and an integer $n \geq -1$.
Output: An n-discount optimal deterministic policy f^∞.

```
1. Select an arbitrary $f^\infty \in C(D)$.
2. Determine $\big(u^{-1}(f), u^0(f), \dots, u^{n+1}(f)\big)$, e.g. as unique solution of the system
   $\begin{cases} \{I - P(f)\}x^{-1} = 0 \\ x^{-1} + \{I - P(f)\}x^0 = r(f) \\ x^{k-1} + \{I - P(f)\}x^k = 0,\ 1 \leq k \leq n+1;\ P^*(f)x^{n+1} = 0 \end{cases}$
3. (a) for all $i \in S$ do
           begin $max(i) := max_{a \in A(i)} \{\sum_j p_{ij}(a)u_j^{-1}(f) - u_i^{-1}(f)\}$;
                 $A^{-1}(i) := argmax_{a \in A(i)} \{\sum_j p_{ij}(a)u_j^{-1}(f) - u_i^{-1}(f)\}$
           end
   (b) if $max(i) = 0$ for all $i \in S$ then go to step 3c
       else begin for all $i \in S$ do select $g(i) \in A^{-1}(i)$; go to step 5 end
   (c) for all $i \in S$ do
           begin $max(i) := max_{a \in A^{-1}(i)} \{r_i(a) + \sum_j p_{ij}(a)u_j^0(f) - u_i^0(f) - u_i^{-1}(f)\}$;
                 $A^0(i) := argmax_{a \in A^{-1}(i)} \{r_i(a) + \sum_j p_{ij}(a)u_j^0(f) - u_i^0(f) - u_i^{-1}(f)\}$
           end
   (d) if $max(i) = 0$ for all $i \in S$ then go to step 3e
       else begin for all $i \in S$ do select $g(i) \in A^0(i)$; go to step 5 end
   (e) for $k = 0$ until $n$ do
           begin
              for all $i \in S$ do
                  begin $max(i) := max_{a \in A^k(i)} \{\sum_j p_{ij}(a)u_j^{k+1}(f) - u_i^{k+1}(f) - u_i^k(f)\}$;
                        $A^{k+1}(i) = argmax_{a \in A^k(i)} \{\sum_j p_{ij}(a)u_j^{k+1}(f) - u_i^{k+1}(f) - u_i^k(f)\}$
                  end
              if $max(i) \neq 0$ for all $i \in S$ then
                  begin for all $i \in S$ do select $g(i) \in A^{k+1}(i)$; go to step 5 end
           end
4. $f^\infty$ is $n$-discount optimal (STOP).
5. for all $i \in S$ do $f(i) := g(i)$; return to step 2.
```

Remarks:

1. In step 2 of the algorithm, we may instead of the last requirement $P^*(f)x^{n+1} = 0$ also solve the additional equation $x^{n+1} + \{I - P(f)\}x^{n+2} = 0$, which implies $P^*(f)x^{n+1} = 0$.
2. If $|A^k(i)| = 1$ for one or more states, then for that states i step 3 of the algorithm can be skipped, because $A^{k+1}(i)$ consists of the same single action as $A^k(i)$.

Example 7.2 (continued)

We compute a 1-discount optimal policy, starting with the policy $f(1) = f(2) = f(3) = 1$.

Iteration 1:

Step 2:

We determine $\big(u^{-1}(f), u^0(f), u^1(f), u^2(f)\big)$ with the stationary and the deviation matrix:

$$P(f) = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \end{pmatrix} \quad \rightarrow \quad P^*(f) = \begin{pmatrix} 0 & 0 & 1 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \end{pmatrix} \quad \rightarrow \quad D(f) = \begin{pmatrix} 1 & 1 & -2 \\ 0 & 1 & -1 \\ 0 & 0 & 0 \end{pmatrix}.$$

$u^{-1}(f) = (0, 0, 0);\ u^0(f) = (2, 1, 0);\ u^1(f) = (-3, -1, 0);\ u^2(f) = (4, 1, 0).$

Step 3: (only for $i = 1$, because $|A(2)| = |A(3)| = 1$.)

a. $a = 1$: $\sum_j p_{ij}(a)u_j^{-1}(f) - u_1^{-1}(f) = 0$; $a = 2$: $\sum_j p_{ij}(a)u_j^{-1}(f) - u_1^{-1}(f) = 0$.

$max(1) = 0;\ A^{-1}(1) = \{1, 2\}$.

c. $a = 1$: $r_1(a) + \sum_j p_{ij}(a)u_j^0(f) - u_1^0(f) - u_1^{-1}(f) = 0$; $a = 2: r_1(a) + \sum_j p_{ij}(a)u_j^0(f) - u_1^0(f) - u_1^{-1}(f) = 0$.

$max(1) = 0;\ A^0(1) = \{1, 2\}$.

e. $k = 0$: $a = 1$: $\sum_j p_{ij}(a)u_j^1(f) - u_1^1(f) - u_1^0(f) = 0$; $a = 2$: $\sum_j p_{ij}(a)u_j^1(f) - u_1^1(f) - u_1^0(f) = 1$.

$max(1) = 1;\ A^{(1)}(1) = \{2\};\ g(1) = 2$.

Step 5:

$f(1) = 2,\ f(2) = 1,\ f(3) = 1$.

Iteration 2:

Step 2:

$$P(f) = \begin{pmatrix} 0 & 0 & 1 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \end{pmatrix} \quad \rightarrow \quad P^*(f) = \begin{pmatrix} 0 & 0 & 1 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \end{pmatrix} \quad \rightarrow \quad D(f) = \begin{pmatrix} 1 & 0 & -1 \\ 0 & 1 & -1 \\ 0 & 0 & 0 \end{pmatrix}.$$

$u^{-1}(f) = (0, 0, 0);\ u^0(f) = (2, 1, 0);\ u^1(f) = (-2, -1, 0);\ u^2(f) = (2, 1, 0).$

Step 3:

a. $a = 1$: $\sum_j p_{ij}(a)u_j^{-1}(f) - u_1^{-1}(f) = 0$; $a = 2$: $\sum_j p_{ij}(a)u_j^{-1}(f) - u_1^{-1}(f) = 0$.

$max(1) = 0;\ A^{-1}(1) = \{1, 2\}$.

c. $a = 1$: $r_1(a) + \sum_j p_{ij}(a)u_j^0(f) - u_1^0(f) - u_1^{-1}(f) = 0$; $a = 2: r_1(a) + \sum_j p_{ij}(a)u_j^0(f) - u_1^0(f) - u_1^{-1}(f) = 0$.

$max(1) = 0;\ A^0(1) = \{1, 2\}$.

e. $k = 0$: $a = 1$: $\sum_j p_{ij}(a)u_j^1(f) - u_1^1(f) - u_1^0(f) = -1$; $a = 2$: $\sum_j p_{ij}(a)u_j^1(f) - u_1^1(f) - u_1^0(f) = 0$.

$max(1) = 0;\ A^1(1) = \{2\}$.

Since $A^1(1)$ consists of one element the policy f^∞ with $f(1) = 2,\ f(2) = f(3) = 1$ is 1-optimal.

In order to show the correctness of Algorithm 7.1 we need some theorems which we present below. We introduce the following notation for two policies $f^\infty, g^\infty \in C(D)$:

$\psi^{-1}(f, g) = P(g)u^{-1}(f) - u^{-1}(f);\ \psi^0(f, g) = r(g) + P(g)u^0(f) - u^0(f) - u^{-1}(f);$

$\psi^k(f, g) = P(g)u^k(f) - u^k(f) - u^{k-1}(f)$ for $k = 1, 2, \dots$.

Theorem 7.8

For every $f^\infty, g^\infty \in C(D)$ and every $m \in \mathbb{N}$, we have

$\alpha v^\alpha(g^\infty) = \sum_{k=-1}^{m-1} \rho^k \big\{u^k(f) + \sum_{t=1}^\infty \alpha^t P^{t-1}(g)\psi^k(f, g)\big\} - \rho^m \sum_{t=1}^\infty \alpha^t P^{t-1}(g)u^{m-1}(f).$

Proof

$$\begin{array}{rcl}
\alpha v^\alpha(g^\infty) & = & \sum_{t=1}^\infty \alpha^t P^{t-1}(g) r(g) \\
& = & \sum_{t=1}^\infty \alpha^t P^{t-1}(g)\{r(g) + P(g)u^0(f) - u^0(f) - u^{-1}(f)\} + \\
& & \qquad \sum_{t=1}^\infty \alpha^t P^{t-1}(g)u^{-1}(f) - \sum_{t=1}^\infty \alpha^t P^{t-1}(g)\{P(g)u^0(f) - u^0(f)\} \\
& = & \sum_{t=1}^\infty \alpha^t P^{t-1}(g)\psi^0(f,g) + \sum_{t=1}^\infty \alpha^t P^{t-1}(g)u^{-1}(f) - \\
& & \qquad \sum_{t=1}^\infty \alpha^t P^{t-1}(g)\{P(g)u^0(f) - u^0(f)\}.
\end{array}$$

$$\begin{array}{rcl}
\sum_{t=1}^\infty \alpha^t P^{t-1}(g)u^{-1}(f) & = & \alpha u^{-1}(f) + \sum_{t=2}^\infty \alpha^t P^{t-1}(g)u^{-1}(f) \\
& = & \alpha u^{-1}(f) + \sum_{t=2}^\infty \alpha^t \big\{u^{-1}(f) + \sum_{s=1}^{t-1} \{P^s(g)u^{-1}(f) - P^{s-1}(g)u^{-1}(f)\}\big\} \\
& = & \alpha(1-\alpha)^{-1}u^{-1}(f) + \sum_{t=2}^\infty \alpha^t \sum_{s=1}^{t-1} P^{s-1}(g)\psi^{-1}(f,g) \\
& = & \alpha(1-\alpha)^{-1}u^{-1}(f) + \sum_{s=1}^\infty \big(\sum_{t=s+1}^\infty \alpha^t\big) P^{s-1}(g)\psi^{-1}(f,g) \\
& = & \rho^{-1}u^{-1}(f) + \sum_{s=1}^\infty \alpha^{s+1}(1-\alpha)^{-1} P^{s-1}(g)\psi^{-1}(f,g) \\
& = & \rho^{-1}\{u^{-1}(f) + \sum_{s=1}^\infty \alpha^s P^{s-1}(g)\psi^{-1}(f,g)\}.
\end{array}$$

For $k \geq 0$, we obtain

$$\begin{array}{rcl}
\rho \cdot \sum_{t=1}^\infty \alpha^t P^{t-1}(g)u^k(f) & = & \big(\frac{1}{\alpha} - 1\big) \sum_{t=1}^\infty \alpha^s P^{t-1}(g)u^k(f) \\
& = & \sum_{t=1}^\infty \alpha^{t-1} P^{t-1}(g)u^k(f) - \sum_{t=1}^\infty \alpha^t P^{t-1}(g)u^k(f) \\
& = & u^k(f) + \sum_{t=2}^\infty \alpha^{t-1} P^{t-1}(g)u^k(f) - \sum_{t=1}^\infty \alpha^t P^{t-1}(g)u^k(f) \\
& = & u^k(f) + \sum_{t=1}^\infty \alpha^t P^{t-1}(g)\{P(g)u^k(f) - u^k(f)\},
\end{array}$$

i.e. $\sum_{t=1}^\infty \alpha^t P^{t-1}(g)\{P(g)u^k(f) - u^k(f)\} = \rho \cdot \sum_{t=1}^\infty \alpha^t P^{t-1}(g)u^k(f) - u^k(f)$.

Since $u^k(f) = P(g)u^{k+1}(f) - u^{k+1}(f) - \psi^{k+1}(f,g)$, we can write

$$\begin{array}{l}
\sum_{t=1}^\infty \alpha^t P^{t-1}(g)\{P(g)u^k(f) - u^k(f)\} = \\
\rho \cdot \sum_{t=1}^\infty \alpha^t P^{t-1}(g)\{P(g)u^{k+1}(f) - u^{k+1}(f) - \psi^{k+1}(f,g)\} - u^k(f) = \\
\rho \cdot \sum_{t=1}^\infty \alpha^t P^{t-1}(g)\{P(g)u^{k+1}(f) - u^{k+1}(f)\} - \rho \cdot \sum_{t=1}^\infty \alpha^t P^{t-1}(g)\psi^{k+1}(f,g)\} - u^k(f).
\end{array}$$

Hence, using this formula for $k = 0$ and then for $k = 1$, we obtain

$$\begin{array}{l}
\sum_{t=1}^\infty \alpha^t P^{t-1}(g)\{P(g)u^0(f) - u^0(f)\} = \\
\rho \cdot \sum_{t=1}^\infty \alpha^t P^{t-1}(g)\{P(g)u^1(f) - u^1(f)\} - \rho \cdot \sum_{t=1}^\infty \alpha^t P^{t-1}(g)\psi^1(f,g) - u^0(f) = \\
\rho \cdot \Big\{\rho \cdot \sum_{t=1}^\infty \alpha^t P^{t-1}(g)\{P(g)u^2(f) - u^2(f)\} - \rho \cdot \sum_{t=1}^\infty \alpha^t P^{t-1}(g)\psi^2(f,g) - u^1(f)\Big\} \\
\qquad\qquad\qquad -\rho \cdot \sum_{t=1}^\infty \alpha^t P^{t-1}(g)\psi^1(f,g) - u^0(f) = \\
\rho^2 \cdot \sum_{t=1}^\infty \alpha^t P^{t-1}(g)\{P(g)u^2(f) - u^2(f) - \psi^2(f,g)\} - \rho \cdot u^1(f) - \rho \cdot \sum_{t=1}^\infty \alpha^t P^{t-1}(g)\psi^1(f,g) - u^0(f).
\end{array}$$

Similarly, by induction on m, it can be shown that

$$\begin{array}{l}
\sum_{t=1}^\infty \alpha^t P^{t-1}(g)\{P(g)u^0(f) - u^0(f)\} = \\
\rho^m \cdot \sum_{t=1}^\infty \alpha^t P^{t-1}(g)\{P(g)u^m(f) - u^m(f) - \psi^m(f,g)\} \\
\qquad\qquad\qquad - \sum_{k=1}^{m-1} \rho^k \big\{u^k(f) + \sum_{t=1}^\infty \alpha^t P^{t-1}(g)\psi^k(f,g)\big\} - u^0(f) = \\
\rho^m \cdot \sum_{t=1}^\infty \alpha^t P^{t-1}(g)u^{m-1}(f) - \sum_{k=1}^{m-1} \rho^k \big\{u^k(f) + \sum_{t=1}^\infty \alpha^t P^{t-1}(g)\psi^k(f,g)\big\} - u^0(f).
\end{array}$$

Finally, we obtain

$$\begin{array}{rl}
\alpha v^\alpha(g^\infty) & = \sum_{t=1}^\infty \alpha^t P^{t-1}(g)\psi^0(f,g) + \sum_{t=1}^\infty \alpha^t P^{t-1}(g)u^{-1}(f) - \sum_{t=1}^\infty \alpha^t P^{t-1}(g)\{P(g)u^0(f) - u^0(f)\} \\
& = \sum_{t=1}^\infty \alpha^t P^{t-1}(g)\psi^0(f,g) + \rho^{-1}\{u^{-1}(f) + \sum_{t=1}^\infty \alpha^t P^{t-1}(g)\psi^{-1}(f,g)\} \\
& \quad -\rho^m \cdot \sum_{t=1}^\infty \alpha^t P^{t-1}(g)u^{m-1}(f) + \sum_{k=1}^{m-1} \rho^k \big\{u^k(f) + \sum_{t=1}^\infty \alpha^t P^{t-1}(g)\psi^k(f,g)\big\} + u^0(f) \\
& = \sum_{k=-1}^{m-1} \rho^k \big\{u^k(f) + \sum_{t=1}^\infty \alpha^t P^{t-1}(g)\psi^k(f,g)\big\} - \rho^m \cdot \sum_{t=1}^\infty \alpha^t P^{t-1}(g)u^{m-1}(f).
\end{array}$$

□

Theorem 7.9

If f^∞ and g^∞ are subsequent policies in Algorithm 7.1, then $v^\rho(g^\infty) > v^\rho(f^\infty)$ for ρ sufficiently small.

Proof

Since $\psi^k(f,f) = 0$ for $k = -1, 0, 1 \ldots$, we obtain from Theorem 7.8 with $m = n+2$,

$v^\rho(g^\infty) - v^\rho(f^\infty) =$

$$(1+\rho)\cdot\big\{\textstyle\sum_{k=-1}^{n+1}\rho^k\cdot\sum_{t=1}^\infty \alpha^t P^{t-1}(g)\psi^k(f,g) + \rho^{n+2}\cdot\sum_{t=1}^\infty \alpha^t\{P^{t-1}(f)-P^{t-1}(g)\}u^{n+1}(f)\big\}.$$

Since

$$\begin{array}{rcl}\|\rho^{n+2}\cdot\sum_{t=1}^\infty \alpha^t\big\{P^{t-1}(f)-P^{t-1}(g)\big\}u^{n+1}(f)\| & \le & \rho^{n+2}(1-\alpha)^{-1}\|P^{t-1}(f)-P^{t-1}(g)\|\cdot\|u^{n+1}(f)\| \\ & \le & 2\rho^{n+2}(1+\rho)\cdot\|u^{n+1}(f)\|,\end{array}$$

$\rho^{n+2}\cdot\sum_{t=1}^\infty \alpha^t\big\{P^{t-1}(f)-P^{t-1}(g)\big\}u^{n+1}(f)$ is arbitrary close to 0 for ρ sufficiently small.

Since f^∞ and g^∞ are subsequent policies in Algorithm 7.1, we have $\psi^k(f,g) = 0$ for $k = -1, 0, \ldots, m-1$ and $\psi^m(f,g) > 0$, where $-1 \le m \le n+1$. If we define $\psi^k(f,g) := 0$ for $k \ge n+2$, then $\sum_{k=-1}^\infty \rho^k\psi^k(f,g) \in LS$ and $\sum_{k=-1}^\infty \rho^k\psi^k(f,g) >_l 0$. Since, by Theorem 7.5, part (1), $H^\rho(g) = \rho\cdot\big\{I - \frac{1}{1+\rho}P(g)\big\}^{-1}$ and, by Theorem 7.6, $H^\rho(g)$ is a positive operator, $\big\{I - \frac{1}{1+\rho}P(g)\big\}^{-1}$ is also a positive operator, implying that

$$\big\{I - \tfrac{1}{1+\rho}P(g)\big\}^{-1}\big\{\textstyle\sum_{k=-1}^\infty \rho^k\psi^k(f,g)\big\} >_l 0.$$

We can also write

$$\begin{array}{l}\big\{I - \frac{1}{1+\rho}P(g)\big\}^{-1}\sum_{k=-1}^\infty \rho^k\psi^k(f,g) = \sum_{t=1}^\infty \alpha^{t-1}P^{t-1}(g)\sum_{k=-1}^\infty \rho^k\psi^k(f,g) = \\ \sum_{t=1}^\infty \alpha^{t-1}P^{t-1}(g)\sum_{k=-1}^{n+1}\rho^k\psi^k(f,g) = \frac{1}{\alpha}\cdot\sum_{k=-1}^{n+1}\rho^k\sum_{t=1}^\infty \alpha^t P^{t-1}(g)\psi^k(f,g) = \\ (1+\rho)\cdot\big\{\sum_{k=-1}^{n+1}\rho^k\cdot\sum_{t=1}^\infty \alpha^t P^{t-1}(g)\psi^k(f,g)\big\}.\end{array}$$

Since we have shown that $v^\rho(g^\infty) - v^\rho(f^\infty)$ consists of two terms, where the first term is lexicographically positive and the second term is arbitrary close to 0 for ρ sufficiently small, we have $v^\rho(g^\infty) > v^\rho(f^\infty)$ for ρ sufficiently small. □

Theorem 7.10

Algorithm 7.1 is correct, i.e. it terminates with an n-discount optimal policy.

Proof

From Theorem 7.9 it follows that each subsequent policy $f^\infty \in C(D)$ is different from all previous. Since $C(D)$ is a finite set, the algorithm terminates, say with policy f^∞. In Theorem 7.9 is shown that for any policy g^∞,

$v^\rho(g^\infty) - v^\rho(f^\infty) = (1+\rho)\cdot\big\{\sum_{k=-1}^{n+1}\rho^k\cdot\sum_{t=1}^\infty \alpha^t P^{t-1}(g)\psi^k(f,g) + \rho^{n+2}\cdot\sum_{t=1}^\infty \alpha^t\{P^{t-1}(f) - P^{t-1}(g)u^{n+1}(f)\}\big\}$.

Since the algorithm terminates, we have $\sum_{k=-1}^{n+2}\rho^k\psi^k(f,g) \le_l 0$. Analogously as in the proof of Theorem 7.9 this implies that

$$(1+\rho)\cdot\big\{\textstyle\sum_{k=-1}^{n+1}\rho^k\cdot\sum_{t=1}^\infty \alpha^t P^{t-1}(g)\psi^k(f,g)\big\} \le_l 0.$$

Since the second term of the above expression for $v^\rho(g^\infty) - v^\rho(f^\infty)$ is again arbitrary close to 0 for ρ sufficiently small, we have shown that $lim_{\rho\downarrow 0}\,\rho^{-n}\{v^\rho(g^\infty) - v^\rho(f^\infty)\} \ge 0$ for every $g^\infty \in C(D)$, i.e. f^∞ is an n-discount optimal policy. □

We close this section with the proof that for $n \ge N-1$ an n-discount optimal policy is also Blackwell optimal. Therefore we need the following lemma.

Lemma 7.8
If $\psi^k(f,g) = 0$ for $k = 1, 2, \dots, N$, then $\psi^k(f,g) = 0$ for $k \geq N+1$.

Proof
Let $L := \{x \mid \{P(f) - P(g)\}x = 0\}$. For $k \geq 1$, we have

$$\begin{aligned} \psi^k(f,g) &= P(g)u^k(f) - u^k(f) - u^{k-1}(f) = P(g)u^k(f) - (-1)^k\{D(f) - I\}D^k(f)r(f) \\ &= P(g)u^k(f) - (-1)^k\{P(f)D(f) - P^*(f)\}D^k(f)r(f) \\ &= P(g)u^k(f) - (-1)^kP(f)D^{k+1}(f)r(f) = P(g)u^k(f) - P(f)u^k(f), \end{aligned}$$

i.e. $u^k(f) \in L$ for $k = 1, 2, \dots$.
Since L is a linear vector space in $\mathbb{R}^N$, the $N+1$ vectors $u^k(f),\ 1 \leq k \leq N+1$, are linear dependent. Because $u^k(f) = B^{k-1}x_0$ for $k \geq 1$, where $x_0 := u^1(f) = D(f)r(f)$ and $B := -D(f)$, the $N+1$ vectors $x_0, Bx_0, B^2x_0, \dots, B^Nx_0$ are linear dependent, i.e. for some $1 \leq k \leq N$, we have $B^kx_0 = \sum_{j=0}^{k-1} \lambda_jB^jx_0$ for some scalars $\lambda_0, \lambda_1, \dots, \lambda_{k-1}$. Hence, $B^kx_0 \in L$. Since $B^{k+1}x_0 = \sum_{j=0}^{k-1} \lambda_jB^{j+1}x_0$, the vector $B^{k+1}x_0$ is a linear combination of the elements $Bx_0, B^2x_0, \dots, B^kx_0$, which all belong to L, so $B^{k+1}x_0 \in L$. Similarly, by induction, it can be shown that $u^k(f) = B^{k-1}x_0 \in L,\ k \geq 1$, implying that $\psi^k(f,g) = 0$ for $k \geq 1$. □

Theorem 7.11
If Algorithm 7.1 is used to determine an $(N-1)$-discount optimal policy f^∞, then f^∞ is a Blackwell optimal policy.

Proof
If the algorithm terminates with policy f^∞, we have $\sum_{k=-1}^N \rho^k\psi^k(f,g) \leq_l 0$ for every policy g^∞, i.e. either $\sum_{k=-1}^N \rho^k\psi^k(f,g) <_l 0$ or $\sum_{k=-1}^N \rho^k\psi^k(f,g) = 0$. In the first case, we obtain analogously to Theorem 7.9 that $v^\rho(g^\infty) < v^\rho(f^\infty)$ for ρ sufficiently small. In the second case, we have $\psi^k(f,g) = 0,\ 1 \leq k \leq N$. From Lemma 7.8 it follows that $\psi^k(f,g) = 0$ also for $k \geq N+1$. Hence, $\psi^k(f,g) = 0$ for $k = 1, 2, \dots$. If we let $m \to \infty$ in Theorem 7.8, then we obtain

$$\alpha v^\alpha(g^\infty) = \sum_{k=-1}^\infty \rho^k\big\{u^k(f) + \sum_{t=1}^\infty \alpha^tP^{t-1}(g)\psi^k(f,g)\big\} = \sum_{k=-1}^\infty \rho^ku^k(f) = \alpha v^\alpha(f^\infty)$$

for every $\alpha \in [0,1)$. Hence, f^∞ is a Blackwell optimal policy. □

7.6 Linear programming and n-discount optimality (irreducible case)

Without any assumption about the chain structure, only for the criteria average and bias optimality there exist a satisfactory treatment by linear programming. In this section we show , under the assumption of irreducibility, a nice treatment for n-discount optimality, based on *nested linear programs.* Throughout this section we have the following assumption.

Assumption 7.1
For any policy $f^\infty \in C(D)$, the Markov chain $P(f)$ is irreducible.

7.6.1 Average optimality

The special linear programming approach for average rewards in the irreducible case was treated in section 6.1.3. There, we have shown that the value vector ϕ and an optimal policy can be found by the linear programs

$$min\Big\{v \;\Big|\; v+\sum_j\{\delta_{ij}-p_{ij}(a)\}u_j \geq r_i(a),\ (i,a)\in S\times A\Big\} \tag{7.16}$$

and

$$max\ \left\{\sum_{(i,a)} r_i(a)x_i(a) \;\middle|\; \begin{array}{rcl} \sum_{(i,a)}\{\delta_{ij}-p_{ij}(a)\}x_i(a) & = & 0,\ j\in S \\ \sum_{(i,a)} x_i(a) & = & 1 \\ x_i(a) & \geq & 0,\ (i,a)\in S\times A \end{array}\right\}, \tag{7.17}$$

respectively. Furthermore, we have shown (Theorem 6.5) that there is a bijection between the feasible solutions of the dual program (7.17) and the set $C(S)$ of stationary policies such that extreme solutions correspond to the set $C(D)$ of deterministic policies. This bijection is given by

$$x_i^\pi(a) := x_i(\pi)\cdot\pi_{ia},\ (i,a)\in S\times A \text{ and } \pi_{ia}^x := \tfrac{x_i(a)}{\sum_a x_i(a)},\ (i,a)\in S\times A,$$

where $x(\pi)$ is the stationary distribution of the transition matrix $P(\pi)$. The following result characterizes the set of all average optimal policies.

Theorem 7.12
Let (ϕ,u^*) be an optimal solution of program (7.16). Then, f^∞ is an average optimal policy if and only if $f^\infty\in A^{-1}$, where $A^{-1}:=\{f^\infty \mid \phi\cdot e+\{I-P(f)\}u^*=r(f)\}$.

Proof
If $f^\infty\in A^{-1}$, then $\phi\cdot e+\{I-P(f)\}u^*=r(f)$. Multiplying this equation with the stationary distribution $x(f)^T$ gives $\phi=x(f)^Tr(f)=\phi(f^\infty)$, i.e. f^∞ is an optimal policy.
Conversely, let f^∞ be an average optimal policy. Then, $\phi=\phi(f^\infty)=x(f)^Tr(f)=\sum_{(i,a)} r_i(a)x_i^f\big(f(i)\big)$, i.e. x^f is an optimal solution of the dual program (7.17). Since $x_i\big(f(i)\big)>0$ for all $i\in S$, we have by the complementary slackness property of linear programming $\phi+\sum_j\{\delta_{ij}-p_{ij}\big(f(i)\big)\}u_j^*=r_i\big(f(i)\big),\ i\in S$, i.e. $f^\infty\in A^{-1}$. □

7.6.2 Bias optimality

We first show that for any average optimal policy f^∞ the second term $u^0(f)$ of the Laurent expansion of $v^\alpha(f^\infty)$ can be obtained from the results of the previous section.

Theorem 7.13
Let f^∞ be an average optimal policy. Then, the bias term $u^0(f)=u^*-P^*(f)u^*$, where u^* is the u-part in an optimal solution of the linear program (7.16).

Proof
Let f^∞ be an average optimal policy. Then, by Theorem 7.12, $\phi\cdot e+\{I-P(f)\}u^*=r(f)$. Multiplying this equation with $D(f)$ gives $u^0(f)=D(f)r(f)=D(f)\{\phi\cdot e+[I-P(f)]u^*\}=\{I-P^*(f)\}u^*=u^*-P^*(f)u^*$. □

The policy f^∞ is bias optimal or 0-discount optimal if $u^0(f)=max\{u^0(g)\mid g^\infty\in A^{-1}\}$. Since $-P^*(g)u^*$ is the average reward of g^∞ with respect to immediate rewards $r_i^{(0)}(a):=-u_i^*$ for all $(i,a)\in S\times A^{-1}$, the maximization of $u^0(g) = u^* - P^*(g)u^*$ is equivalent to the maximization of the average reward corresponding to immediate rewards $r_i^{(0)}(a),\ (i,a) \in S \times A^{-1}$. Hence, for bias optimality, we can consider a new MDP model with truncated actions sets $A^{-1}(i) := \{a \in A(i) \mid \phi + \sum_j \{\delta_{ij} - p_{ij}(a)\}u_j^* = r_i(a)\},\ i \in S$, and with immediate rewards $r_i^{(0)}(a) := -u_i^*,\ (i,a) \in S \times A^{-1}$. We now present the primal and dual linear program for this truncated MDP:

$$min\Big\{v^{(0)} \;\Big|\; v^{(0)} + \sum_j \{\delta_{ij} - p_{ij}(a)\}u_j^{(0)} \geq -u_i^*,\ i \in S,\ a \in A^{-1}(i)\Big\} \tag{7.18}$$

and

$$max \left\{ -\sum_i u_i^* \sum_{a \in A^{-1}(i)} x_i^{(0)}(a) \;\middle|\; \begin{array}{rcl} \sum_i \sum_{a \in A^{-1}(i)} \{\delta_{ij} - p_{ij}(a)\}x_i^{(0)}(a) & = & 0,\ j \in S \\ \sum_i \sum_{a \in A^{-1}(i)} x_i^{(0)}(a) & = & 1 \\ x_i^{(0)}(a) & \geq & 0,\ i \in S,\ a \in A^{-1}(i) \end{array} \right\} \tag{7.19}$$

respectively. The next theorem is a consequence of above statements.

Theorem 7.14

Let $(v^{(0)}, u^{(0)})$ and $x^{(0)}$ be optimal solutions of the linear programs (7.18) and (7.19) respectively. Furthermore, let $A^0 := \{f^\infty \in A^{-1} \mid v^{(0)} + \sum_j \{\delta_{ij} - p_{ij}\big(f(i)\big)\}u^{(0)} = -u_i^*,\ i \in S\}$. Then, f^∞ is a bias optimal policy if and only if $f^\infty \in A^0$.

Example 7.3

$S = \{1,2,3\};\ A(1) = \{1,2\},\ A(2) = \{1\},\ A(3) = \{1\};\ r_1(1) = 3,\ r_1(2) = 4,\ r_2(1) = 2,\ r_3(1) = 1.$
$p_{11}(1) = 0,\ p_{12}(1) = 1,\ p_{13}(1) = 0;\ p_{11}(2) = p_{12}(2) = 0,\ p_{13}(2) = 1;\ p_{21}(1) = \frac{1}{2},\ p_{22}(1) = 0,\ p_{23}(1) = \frac{1}{2};$
$p_{31}(1) = 0,\ p_{32}(1) = 1,\ p_{33}(1) = 0.$
It is easy to verify that this is an irreducible MDP. The primal and dual linear programs for average optimality are:

$$min\Big\{v \;\Big|\; v + u_1 - u_2 \geq 3;\ v + u_1 - u_3 \geq 4;\ v + u_2 - \frac{1}{2}u_1 - \frac{1}{2}u_3 \geq 2;\ v + u_3 - u_2 \geq 1\Big\} \tag{7.20}$$

and

$$max \left\{ 3x_1(1) + 4x_1(2) + 2x_2(1) + x_3(1) \;\middle|\; \begin{array}{rrrrl} x_1(1) + & x_1(2) - & \frac{1}{2}x_2(1) & & = 0;\ x_1(1) \geq 0 \\ -x_1(1) + & + & x_2(1) & -\ x_3(1) & = 0;\ x_1(2) \geq 0 \\ - & x_1(2) - & \frac{1}{2}x_2(1) & +\ x_3(1) & = 0;\ x_2(1) \geq 0 \\ x_1(1) + & x_1(2) + & x_2(1) & +\ x_3(1) & = 1;\ x_3(1) \geq 0 \end{array} \right\}. \tag{7.21}$$

Optimal solutions of these programs are: for the primal $v^* = \phi = 2,\ u_1^* = 2,\ u_2^* = 1,\ u_3^* = 0$ and for the dual $x_1^*(1) = \frac{1}{4},\ x_1^*(2) = 0,\ x_2^*(1) = \frac{1}{2},\ x_3^*(1) = \frac{1}{4}$.
It is simple to verify that $A^{-1}(1) = A(1) = \{1,2\},\ A^{-1}(2) = A(2) = \{1\}$ and $A^{-1}(3) = A(3) = \{1\}$. Hence, the primal and dual programs for bias optimality are:

$$min \left\{ v^{(0)} \;\middle|\; \begin{array}{llllll} v^{(0)} + u_1^{(0)} - u_2^{(0)} & \geq -2; & v^{(0)} - \frac{1}{2}u_1^{(0)} + u_2^{(0)} - \frac{1}{2}u_3^{(0)} & \geq & -1 \\ v^{(0)} + u_1^{(0)} - u_3^{(0)} & \geq -2; & v^{(0)} - u_2^{(0)} + u_3^{(0)} & \geq & 0 \end{array} \right\} \tag{7.22}$$

and

$$max \left\{ -2x_1^{(0)}(1) - 2x_1^{(0)}(2) - x_2^{(0)}(1) \;\middle|\; \begin{array}{rrrrl} x_1^{(0)}(1) + & x_1^{(0)}(2) - & \frac{1}{2}x_2^{(0)}(1) & & = 0;\ x_1^{(0)}(1) \geq 0 \\ -x_1^{(0)}(1) + & + & x_2^{(0)}(1) & -\ x_3^{(0)}(1) & = 0;\ x_1^{(0)}(2) \geq 0 \\ - & x_1^{(0)}(2) - & \frac{1}{2}x_2^{(0)}(1) & +\ x_3^{(0)}(1) & = 0;\ x_2^{(0)}(1) \geq 0 \\ x_1^{(0)}(1) + & x_1^{(0)}(2) + & x_2^{(0)}(1) & +\ x_3^{(0)}(1) & = 1;\ x_3^{(0)}(1) \geq 0 \end{array} \right\}. \tag{7.23}$$

Optimal solutions of these programs are: for the primal $v^{(0)} = -\frac{4}{5},\ u_1^{(0)} = -\frac{6}{5},\ u_2^{(0)} = -\frac{4}{5},\ u_3^{(0)} = 0$ and for the dual $x_1^{(0)}(1) = 0,\ x_1^{(0)}(2) = \frac{1}{5},\ x_2^{(0)}(1) = \frac{2}{5},\ x_3^{(0)}(1) = \frac{2}{5}$.
Hence, $A^0(1) = \{2\},\ A^0(2) = \{1\}$ and $A^0(3) = \{1\}$, i.e. f^∞ with $f(1) = 2,\ f(2) = f(3) = 1$ is the only bias optimal policy.

7.6.3 n-discount optimality

In this subsection we propose an algorithm for the determination of an n-discount optimal policy based on a system of nested linear programs. This is a generalization of the approach discussed in the preceding subsection. Let us first introduce the pair of dual linear programs for the computation of an n-discount optimal policy. The primal program is

$$min\Big\{v^{(n)} \;\Big|\; v^{(n)} + \sum_j \{\delta_{ij} - p_{ij}(a)\}u_j^{(n)} \ge -u_i^{(n-1)},\ i \in S,\ a \in A^{n-1}(i)\Big\}, \tag{7.24}$$

where $(v^{(n-1)}, u^{(n-1)})$ is an optimal solution of the primal linear program for an $(n-1)$-discount optimal policy and $A^{n-1}(i) := \{a \in A^{n-2} \mid v^{(n-1)} + \sum_j \{\delta_{ij} - p_{ij}(a)\}u_j^{(n-1)} = -u^{(n-2)}\}$.
The dual linear program of (7.25) is

$$max\left\{-\sum_i u_i^{(n-1)} \sum_{a\in A^{n-1}(i)} x_i^{(n)}(a) \;\middle|\; \begin{array}{rcl} \sum_i \sum_{a\in A^{n-1}(i)} \{\delta_{ij} - p_{ij}(a)\}x_i^{(n)}(a) & = & 0,\ j \in S \\ \sum_i \sum_{a\in A^{n-1}(i)} x_i^{(n)}(a) & = & 1 \\ x_i^{(n)}(a) & \ge & 0,\ i \in S,\ a \in A^{n-1}(i) \end{array}\right\}. \tag{7.25}$$

By induction on n we will show the following theorem.

Theorem 7.15
Let $\big(v^{(n)}, u^{(n)}\big)$ and $x^{(n)}$ be optimal solutions of the linear programs (7.24) and (7.25) respectively. Furthermore, let $A^{(n)} := \big\{f^\infty \in A^{(n-1)} \mid v^{(n)} + \sum_j \{\delta_{ij} - p_{ij}\big(f(i)\big)\}u_j^{(n)} = -u_i^{(n-1)},\ i \in S\big\}$.
(1) If f^∞ be an $(n-1)$-discount optimal policy, then, $u^n(f) = u^{(n-1)} - P^*(f)u^{(n-1)}$.
(2) f^∞ is an n-discount optimal policy if and only if $f^\infty \in A^{(n)}$.

Proof
For $n = 0$ we refer to the Theorems 7.13 and 7.14. We proof the induction step for $n \ge 1$.
(1) Let f^∞ be an $(n-1)$-discount optimal policy.
$u^n(f) = -D(f)u^{n-1}(f) = -D(f)\{u^{(n-2)} - P^*(f)u^{(n-2)}\}$, the last equality by induction.
Since $D(f)P^*(f) = 0$, we obtain $u^n(f) = -D(f)u^{(n-2)}$. Because $f^\infty \in A^{(n-1)}$, we can write $u^n(f) = D(f)\big\{v^{(n-1)} \cdot e + \{I - P(f)\}u^{(n-1)}\big\}$. Since $D(f)e = 0$ and $D(f)\{I - P(f)\} = I - P^*(f)$, it follows that $u^n(f) = u^{(n-1)} - P^*(f)u^{(n-1)}$.
(2) Let f^∞ be an n-discount optimal policy. Then, according to the induction assumption, f^∞ is average optimal for the MDP model with truncated action sets $A^{(n-1)}(i),\ i \in S$ and rewards $-u_i^{(n-1)}$. We know that there exists a one-to-one correspondence between the deterministic optimal policies and the extreme solutions of the dual program (7.25). Hence, there exists an extreme optimal solution $x^{(n)}$ such that $x_i^{(n)}(f(i)) > 0$, for all $i \in S$. Then, from the complementary slackness property of linear programming we conclude that $v^{(n)} + \sum_j \{\delta_{ij} - p_{ij}\big(f(i)\big)\}u_j^{(n)} = -u_i^{(n-1)},\ i \in S$, i.e. $f^\infty \in A^{(n)}$.
Conversely, let $f^\infty \in A^{(n)}$. Then, $v^{(n)} \cdot e + \{I - P(f)\}u^{(n)} = -u^{(n-1)}$. By multiplication with $P^*(f)$, we obtain $v^{(n)} \cdot e = -P^*(f)u^{(n-1)}$. For any $f^\infty \in A^{(n-1)}$, we derive from the primal program (7.24) that $v^{(n)} \cdot e + \{I - P(g)\}u^{(n)} \ge -u^{(n-1)}$ for every $g^\infty \in A^{(n-1)}$. Consequently, by multiplication with

$P^*(g)$, we have $v^{(n)} \cdot e \geq -P^*(g)u^{(n-1)}$. Hence,
$u^n(f) = u^{(n-1)} - P^*(f)u^{(n-1)} = u^{(n-1)} + v^{(n)} \cdot e \geq u^{(n-1)} - P^*(g)u^{(n-1)} = u^n(g)$ for every $g^\infty \in A^{(n-1)}$, i.e. f^∞ is an n-discount optimal policy. □

7.7 Blackwell optimality and linear programming

In this section we will show how linear programming in the space of rational functions can be developed to compute optimal policies over the entire range of the discount factor. Furthermore, a procedure is presented for the computation of a Blackwell optimal policy.

Let $\mathbb{R}$ be the ordered field of the real numbers with the usual ordering. By $P(\mathbb{R})$ we denote the set of all polynomials in $x \in \mathbb{R}$ with real coefficients, i.e. the set of elements

$$p(x) = a_0 + a_1x + \cdots + a_nx^n \text{ where } a_i \in \mathbb{R},\ 1 \leq i \leq n \text{ for some positive integer } n, \tag{7.26}$$

where we assume that $a_n \neq 0$. The field $F(\mathbb{R})$ of rational functions with real coefficients consists of the elements

$$f(x) = \frac{p(x)}{q(x)}, \tag{7.27}$$

where p and q are elements of $P(\mathbb{R})$ having no common linear factors and q is not identically zero. So, each rational function is expressible in the form

$$f(x) = \frac{a_0 + a_1x + \cdots + a_nx^n}{b_0 + b_1x + \cdots + b_mx^m}. \tag{7.28}$$

The domain of a rational function consists of all but the finitely many real numbers where the denominator is 0. At these points, the numerator is nonzero, because there are no common linear factors. So when we compare two rational functions f and g, we can be sure that the common domain *dom* $f \cap$ *dom* g consists of all but finitely many real numbers.

To complete the description of the field, we need specity the addition $(+)$ and multiplication $(\cdot)$ operations on the set $F(\mathbb{R})$, and we need to single out two members 0 and 1 as the 0 and 1 elements. The latter is easy: the elements 0 and 1 are the constant functions having the values 0 and 1, respectively. The operations $+$ and $\cdot$ in $F(\mathbb{R})$ are defined in the usual way:

$$\Big(\frac{p}{q} + \frac{r}{s}\Big)(x) = \frac{p(x)s(x) + r(x)q(x)}{q(x)s(x)} \text{ and } \Big(\frac{p}{q} \cdot \frac{r}{s}\Big)(x) = \frac{p(x)r(x)}{q(x)s(x)}, \tag{7.29}$$

with an additional operation to cancelling any common linear factors in the numerator and denominator. As example, let $f(x) := \frac{1}{-1+x^2}$, $g(x) := \frac{x}{-1+x^2}$ and $h(x) := 1 + x$. Then,
$(f+g)(x) = \frac{1}{-1+x^2} + \frac{x}{-1+x^2} = \frac{1+x}{-1+x^2} = \frac{1}{-1+x}$ and $(f \cdot h)(x) = \frac{1}{-1+x^2} \cdot (1+x) = \frac{1+x}{-1+x^2} = \frac{1}{-1+x^2}$.

Next we need to augment this field with an ordering relation making it into an ordered field. We will denote the ordering relation by $>_l$. Since $f >_l g$ if and only if $f - g >_l 0$, it suffices to specify the set of positive rational functions. We have special interest in the value of these functions for x close to 0. Therefore, we define the *dominating coefficient* of a polynomial p given in formula (7.26) as the coefficient $a_k \neq 0$, where k is such that $a_i = 0,\ 0 \leq i \leq k-1$ (for the function 0, we define the dominating coefficient as 0). In this case we call k the *order* of p: *order*$(p) = k$. The dominating coefficient of polynomial p is denoted by $d(p)$. Notice that $d(p) = 0$ if and only if $p = 0$. Let P be the set of positive elements of $F(\mathbb{R})$, defined by

$$f(x) = \frac{p(x)}{q(x)} \in P \text{ if and only if } d(p)d(q) > 0. \tag{7.30}$$

We now define $f \geq_l g$ if either $f >_l g$ or $f = g$. With this definition the field $F(\mathbb{R})$ is a total ordered field (the proof is left to the reader as Exercise 7.10). Observe that the function $f(x) = \frac{1}{x}$ is 'infinity large' in the sense that $\frac{1}{x} >_l n$ for any $n \in \mathbb{N}$. Similarly, the reciprocal function $g(x) = x$ is 'infinity small' in the sense that $x <_l \frac{1}{n}$ for any $n \in \mathbb{N}$. Hence, the field is a non-Archimedian ordered field.[1]

The continuity of polynomials implies that the rational function $f = \frac{p}{q} \in P$ if and only if $\frac{p(x)}{q(x)} > 0$ for all x sufficiently near to 0. Hence, we obtain the following result.

Lemma 7.9
The rational function $f = \frac{p}{q} \in P$ if and only if there exists a positive real number x_0 such that $f(x) = \frac{p(x)}{q(x)} > 0$ for all $x \in (0, x_0)$.

We shall apply the above properties on discounted rewards as function of the discount factor α. As before, we will use the parameter $\rho = \frac{1-\alpha}{\alpha}$ instead of α. Note that $\alpha = \frac{1}{1+\rho}$ and that $\alpha \uparrow 1$ is equivalent to $\rho \downarrow 0$. The total expected discounted reward $v^\rho(f^\infty)$ for a policy $f^\infty \in C(D)$ is the unique solution of the linear system

$$\{(1+\rho)I - P(f)\}x = (1+\rho)r(f). \tag{7.31}$$

Solving (7.31) by Cramer's rule shows that for every $i \in S$, the function $v_i^\rho(f^\infty)$ is an element of $F(\mathbb{R})$, say $v_i^\rho(f^\infty) = \frac{p(\rho)}{q(\rho)}$. The degree of the polynomials p and q is at most N. By Blackwell's theorem, we know that the interval $[0,1)$ of the discount factor α can be broken into a finite number of intervals, say $[0 = \alpha_{s+1}, \alpha_s)$, $[\alpha_s, \alpha_{s-1})$, $\dots$, $[\alpha_0, \alpha_{-1} = 1)$, in such a way that there are policies f_k^∞, $k = 0, 1, \dots, s+1$, where f_k^∞ is α-discounted optimal for all $\alpha \in [\alpha_k, \alpha_{k-1})$. The policy f_0^∞ is a Blackwell optimal policy. Observe that in each interval the components v_i^ρ of the value vector v^ρ are elements of $F(\mathbb{R})$. So, for small ρ corresponding with the interval $[\alpha_0, \alpha_{-1} = 1)$, i.e. $0 < \rho < \frac{1-\alpha_0}{\alpha_0}$, v_i^ρ is an element of $F(\mathbb{R})$.
The optimality equation of discounted rewards implies

$$(1+\rho)v_i^\rho \geq (1+\rho)r_i(a) + \sum_j p_{ij}(a)v_j^\rho,\ (i,a) \in S \times A,\ \rho > 0. \tag{7.32}$$

Since v_i^ρ is an element of $F(\mathbb{R})$ for $\rho \in \big[0, \frac{1-\alpha_0}{\alpha_0}\big)$, we obtain from (7.32) the ordering relations

$$(1+\rho)v_i^\rho \geq_l (1+\rho)r_i(a) + \sum_j p_{ij}(a)v_j^\rho,\ (i,a) \in S \times A. \tag{7.33}$$

An N-vector $w(\rho)$ with elements in $F(\mathbb{R})$ is called *Blackwell-superharmonic* if

$$(1+\rho)w_i(\rho) \geq_l (1+\rho)r_i(a) + \sum_j p_{ij}(a)w_j(\rho),\ (i,a) \in S \times A. \tag{7.34}$$

Theorem 7.16
The discounted value vector v^ρ is the (componentwise) smallest Blackwell-superharmonic vector with components in $F(\mathbb{R})$, i.e. for any Blackwell-superharmonic vector $w(\rho)$, we have $w_i(\rho) \geq_l v_i^\rho$, $i \in S$.

Proof
From (7.33) it follows that the discounted value vector v^ρ is a Blackwell-superharmonic vector. Suppose that $w(\rho)$ is an arbitrary Blackwell-superharmonic vector. Since there are only a finite number of elements in $S \times A$ it follows from Lemma 7.9 that there exists a positive real number ρ_0 such that

[1] For more details about ordered fields (Archimedian and non-Archimedian) see: B.L. van der Waerden, *Algebra - Erster Teil*, Springer-Verlag (1966) 235–238.

$$(1+\rho)w_i(\rho) \geq (1+\rho)r_i(a) + \textstyle\sum_j p_{ij}(a)w_j(\rho),\ (i,a) \in S \times A,\ \rho \in [0,\rho_0).$$

Hence, for every $\alpha \in \big[\frac{1}{1+\rho_0}\big)$ the vector $w(\rho)$ is α-superharmonic in the sense of (3.30). Therefore, by the results of discounted rewards in Chapter 3, $w_i(\rho) \geq v_i^\rho$ for all $i \in S$ and all $\rho \in [0,\rho_0)$. Consequently, $w_i(\rho) \geq_l v_i^\rho,\ i \in S,\ \rho \in [0,\rho_0)$. □

Theorem 7.16 implies that the value vector v^ρ for the interval $[0,\rho_0)$ can be found as optimal solution of the following linear program in $F(\mathbb{R})$:

$$min\Big\{\sum_j w_j(\rho)\ \Big|\ (1+\rho)w_i(\rho) \geq_l (1+\rho)r_i(a) + \sum_j p_{ij}(a)w_j(\rho),\ (i,a) \in S \times A\Big\}. \tag{7.35}$$

Consider also the following linear program in $F(\mathbb{R})$, called the *dual program*:

$$max\left\{\sum_{(i,a)} (1+\rho)r_i(a)\cdot x_{ia}(\rho)\ \middle|\ \begin{array}{rcl} \sum_{(i,a)}\{(1+\rho)\delta_{ij} - p_{ij}(a)\}\cdot x_{ia}(\rho) & = & 1,\ j \in S \\ x_{ia}(\rho) & \geq_l & 0,\ (i,a) \in S \times A \end{array}\right\}. \tag{7.36}$$

For a fixed positive ρ, the linear programs (7.35) and (7.36) are equivalent to the linear programs of Chapter 3. Therefore, we also have for each fixed ρ a one-to-one correspondence between the basic feasible solutions and the policies of $C(D)$. For the present programs with elements from $F(\mathbb{R})$ we will, as in the simplex method with real numbers, rewrite the equalities $\sum_{(i,a)}\{(1+\rho)\delta_{ij} - p_{ij}(a)\}\cdot x_{ia}(\rho) = 1,\ j \in S$, such that at each iteration there is precisely one positive $x_{ia}(\rho)$ for each state i. Hence, the only difference with the usual simplex method with real numbers is that instead of real numbers the elements in the programs are rational functions.

At any iteration there is an extreme feasible solution $x(\rho)$ of (7.36), corresponding to a policy f^∞, and a reduced cost vector $w(\rho)$ such that the complementary slackness property is satisfied, i.e.

$$x_{ia}(\rho)\cdot\big\{\textstyle\sum_j\{(1+\rho)\delta_{ij} - p_{ij}(a)\}\cdot w_j(\rho) - (1+\rho)r_i(a)\big\} = 0,\ (i,a) \in S \times A.$$

Since $x_{jf(j)}(\rho) > 0,\ j \in S$, we have $\sum_j\{(1+\rho)\delta_{ij} - p_{ij}(f)\}\cdot w_j(\rho) = (1+\rho)r_i\big(f(i)\big),\ i \in S$. Hence, $w(\rho) = v^\rho(f^\infty)$. The validation of the above described approach and some additional properties follow from the following lemma.

Lemma 7.10

(1) *The elements in the simplex tableau can be written as rational functions with the same denominator, say $n(\rho)$, which is the product of the previous pivot elements.*

(2) *The numerators and common denominator are polynomials with degree at most N; the numerators of the reduced costs are polynomials with degree at most $N+1$.*

(3) *The pivot operations in the simplex tableau are as follows:*

a. *The new common denominator is the numerator of the current pivot element.*

b. *The new numerator of the pivot element is the current common denominator.*

c. *The new numerators of the other elements in the pivot row are unchanged.*

d. *The new numerators of the other elements in the pivot column are the old numerator multiplied by -1.*

e. *For the other elements, say an element with numerator $p(\rho)$, the new numerator becomes $\frac{p(\rho)t(\rho)-r(\rho)s(\rho)}{n(\rho)}$, which is a polynomial, where $t(\rho)$ is the numerator of the old pivot element, $r(\rho)$ is the numerator of the element in the pivot row and the same column as the element with numerator $p(\rho)$, and $s(\rho)$ is the numerator of the element in the pivot column and the same row as the element with numerator $p(\rho)$.*

Proof

(1) We can compute a simplex tableau corresponding to some policy f^∞ as follows. Let $z_j(\rho),\ j \in S$ be artificial variables, i.e. consider the system $\sum_{(i,a)} \{(1+\rho)\delta_{ij} - p_{ij}(a)\} \cdot x_{ia}(\rho) + z_j(\rho) = 1,\ j \in S$. Then exchange by the usual pivot operations $z_j(\rho)$ with $x_{jf(j)}(\rho)$ for $j = 1, 2, \dots, N$. The first basis matrix is the identity matrix I corresponding to the artificial variables. Hence, in the first simplex tableau the elements are polynomials (in fact linear functions) in ρ, which may be considered as rational functions with common denominator 1. It is well known from the theory of linear programming (see e.g. [341]) that the elements of a simplex tableau have a common denominator, namely the determinant of the basis matrix which is the product of all previous pivot elements when the first basis matrix is the identity matrix. This result, with a similar proof, is also valid of the elements are rational functions instead of real numbers.

(2) Any basis matrix is of the form $(1+\rho)I - P(f)$. So it has linear functions on the diagonal and constants on the off-diagonal elements. Hence, the determinant of the matrix is a polynomial with degree at most N. By Cramer's rule, the elements of the inverse have numerators which are polynomials with degree at most $N-1$. The elements in a column of the simplex tableau, except the reduced costs, are obtained by multiplication of the inverse of the basis matrix with the right hand side or a nonbasic column. Such columns are constants or linear functions. Hence, the polynomials of the numerators have degree at most N. Since the reduced costs are (rewritten) terms of the objective function $\sum_{(i,a)} (1+\rho) r_i(a) \cdot x_{ia}(\rho)$, they are obtained by multiplying the variables $x_{ia}(\rho)$ with a linear function, so these numerators have degree at most $N+1$.

(3) The transformation rules for the simplex method with rational functions are similar to these rules in case of real numbers. Let $\frac{t(\rho)}{n(\rho)}$ be the pivot element. Then $n(\rho)$ is the product of all previous pivots. The new product of the pivots is $n(\rho) \cdot \frac{t(\rho)}{n(\rho)} = t(\rho)$. The rules b, c and d are straightforward. Consider an element $\frac{p(\rho)}{n(\rho)}$ outside the pivot row or pivot column. Let $\frac{r(\rho)}{n(\rho)}$ be the element in the pivot row and the same column as the element $\frac{p(\rho)}{n(\rho)}$ and let $\frac{s(\rho)}{n(\rho)}$ be the element in the pivot column and the same row as the element $\frac{p(\rho)}{n(\rho)}$. Then the new element becomes: $\frac{p(\rho)}{n(\rho)} - \frac{r(\rho)}{n(\rho)} \cdot \frac{s(\rho)}{n(\rho)} \cdot \frac{n(\rho)}{t(\rho)} = \frac{1}{t(\rho)} \cdot \big\{ \frac{p(\rho)\cdot t(\rho) - r(\rho)\cdot s(\rho)}{n(\rho)} \big\}$. The the property that $t(\rho)$ is the common denominator of the new tableau implies that $\frac{p(\rho)\cdot t(\rho) - r(\rho)\cdot s(\rho)}{n(\rho)}$ is a polynomial. □

We shall solve the dual program (7.36) starting with ρ very large, or equivalently α very close to 0. For $\alpha = 0$ the policy f^∞ such that $r_i\big(f(i)\big) = max_a\, r_i(a),\ i \in S$ is an optimal policy. We start with the basic solution corresponding to this policy f^∞. We can compute the first feasible simplex tableau as follows. Let $z_j(\rho),\ j \in S$ be the artificial variables, i.e. we consider the system

$$\sum_{(i,a)} \{(1+\rho)\delta_{ij} - p_{ij}(a)\} \cdot x_{ia}(\rho) + z_j(\rho) = 1,\ j \in S.$$

Then we exchange by the usal pivot operations $z_j(\rho)$ with $x_{jf(j)}(\rho)$ for $j = 1, 2, \dots, N$. This tableau is optimal for all $\rho \geq \rho_*$, where ρ_* is the smallest ρ for which the reduced costs (the reduced costs are also elements of $F(\mathbb{R})$) are nonnegative. To compute ρ_* we have to compute the zeros of some polynomials. [2] The reduced cost that determines ρ_* determines the next pivot column. After a pivot operation we repeat this approach to obtain a next interval on which the new policy is optimal. In this way we continue until we have an interval that ends with $\rho_* = 0$. That final interval corresponds to a Blackwell optimal policy. This approach determines optimal policies over the entire range $[0, 1)$ of the discount factor α.

[2] The computation of real zeros of polynomials can be done by Maple. We refer also to the literature on numerical analysis and to [125] in which paper a method based on Sturm's Theorem is discussed.

Example 7.4

$S = \{1,2,3\}: \ A(1) = A(2) = A(3) = \{1,2\}$.
$r_1(1) = 8, \ r_1(1) = \frac{11}{4}; \ r_2(1) = 16, \ r_2(2) = 15; \ r_3(1) = 7, \ r_3(2) = 4..$
$p_{11}(1) = \frac{1}{2}, \ p_{12}(1) = \frac{1}{4}, \ p_{13}(1) = \frac{1}{4}; \ p_{11}(2) = \frac{1}{16}, \ p_{12}(2) = \frac{3}{4}, \ p_{13}(2) = \frac{3}{16};$
$p_{21}(1) = \frac{1}{2}, \ p_{22}(1) = 0, \ p_{23}(1) = \frac{1}{2}; \ p_{21}(2) = \frac{1}{16}, \ p_{22}(2) = \frac{7}{8}, \ p_{23}(2) = \frac{1}{16};$
$p_{31}(1) = \frac{1}{4}; \ p_{32}(1) = \frac{1}{4}; \ p_{33}(1) = \frac{1}{2}; \ p_{31}(2) = \frac{1}{8}; \ p_{32}(2) = \frac{3}{4}; \ p_{33}(2) = \frac{1}{8}$.
For this example the objective function becomes:
$(1+\rho) \cdot \{8x_{11}(\rho) + \frac{11}{4}x_{12}(\rho) + 16x_{21}(\rho) + 15x_{22}(\rho) + 7x_{31}(\rho) + 4x_{32}(\rho)\}$
The linear constraints are:

$$\begin{array}{rrrrrrrr}
 & (\frac{1}{2}+\rho)x_{11}(\rho) & + \ (\frac{15}{16}+\rho)x_{12}(\rho) & - \ \frac{1}{2}x_{21}(\rho) & - \ \frac{1}{16}x_{22}(\rho) & - \ \frac{1}{4}x_{31}(\rho) & - \ \frac{1}{8}x_{32}(\rho) & = 1 \\
- & \frac{1}{4}x_{11}(\rho) & - \ \frac{3}{4}x_{12}(\rho) & + \ (1+\rho)x_{21}(\rho) & + \ (\frac{1}{8}+\rho)x_{22}(\rho) & - \ \frac{1}{4}x_{31}(\rho) & - \ \frac{3}{4}x_{32}(\rho) & = 1 \\
- & \frac{1}{4}x_{11}(\rho) & - \ \frac{3}{16}x_{12}(\rho) & - \ \frac{1}{2}x_{21}(\rho) & - \ \frac{1}{16}x_{22}(\rho) & + \ (\frac{1}{2}+\rho)x_{31}(\rho) & + \ (\frac{7}{8}+\rho)x_{32}(\rho) & = 1
\end{array}$$

The first simplex tableau is (the common denominator is the top element in the second column of the simplex tableau; the first common denominator is 1).

	1	$x_{11}(\rho)$	$x_{12}(\rho)$	$x_{21}(\rho)$	$x_{22}(\rho)$	$x_{31}(\rho)$	$x_{32}(\rho)$
$z_1(\rho)$	1	$+\frac{1}{2}+\rho$	$+\frac{15}{16}+\rho$	$-\frac{1}{2}$	$-\frac{1}{16}$	$-\frac{1}{4}$	$-\frac{1}{8}$
$z_2(\rho)$	1	$-\frac{1}{4}$	$-\frac{3}{4}$	$+1+\rho$	$+\frac{1}{8}+\rho$	$-\frac{1}{4}$	$-\frac{3}{4}$
$z_3(\rho)$	1	$-\frac{1}{4}$	$-\frac{3}{16}$	$-\frac{1}{2}$	$+\frac{1}{16}$	$+\frac{1}{2}+\rho$	$+\frac{7}{8}+\rho$
	0	$-8-8\rho$	$-\frac{11}{4}-\frac{11}{4}\rho$	$-16-16\rho$	$-15-15\rho$	$-7-7\rho$	$-4-4\rho$

In the first iteration the pivot column is the column of the variable $x_{11}(\rho)$ and the pivot row is the row of $z_1(\rho)$. The next tableau becomes (with common denominator $\frac{1}{2}+\rho$).

	$\frac{1}{2}+\rho$	$z_1(\rho)$	$x_{12}(\rho)$	$x_{21}(\rho)$	$x_{22}(\rho)$	$x_{31}(\rho)$	$x_{32}(\rho)$
$x_{11}(\rho)$	1	+1	$+\frac{15}{16}+\rho$	$-\frac{1}{2}$	$-\frac{1}{16}$	$-\frac{1}{4}$	$-\frac{1}{8}$
$z_2(\rho)$	$\frac{3}{4}+\rho$	$+\frac{1}{4}$	$-\frac{9}{64}-\frac{1}{2}\rho$	$+\frac{3}{8}+\frac{3}{2}+\rho^2$	$+\frac{3}{16}+\frac{5}{8}\rho+\rho^2$	$-\frac{3}{16}-\frac{1}{4}\rho$	$-\frac{13}{16}-\frac{3}{4}\rho$
$z_3(\rho)$	$\frac{3}{4}+\rho$	$+\frac{1}{4}$	$+\frac{9}{64}+\frac{1}{16}\rho$	$-\frac{3}{8}-\frac{1}{2}\rho$	$-\frac{3}{16}-\frac{1}{16}\rho$	$+\frac{3}{16}+\rho+\rho^2$	$-\frac{3}{16}+\frac{11}{8}\rho+\rho^2$
	$8+8\rho$	$8+8\rho$	$+\frac{49}{8}+\frac{91}{8}\rho+\frac{21}{4}\rho^2$	$-12-28\rho-16\rho^2$	$-8-23\rho-15\rho^2$	$-\frac{11}{2}-\frac{25}{2}\rho-7\rho^2$	$-3-7\rho-4\rho^2$

After inserting $x_{21}(\rho)$ and $x_{31}(\rho)$ into the basis the next tableau is obtained, which is the first feasible tableau in which the common denominator is $\frac{15}{16}\rho + 2\rho^2 + \rho^3$.

	$\frac{15}{16}\rho+2\rho^2+\rho^3$	$z_1(\rho)$	$x_{12}(\rho)$	$z_2(\rho)$	$x_{22}(\rho)$	$z_3(\rho)$	$x_{32}(\rho)$
$x_{11}(\rho)$	$\frac{9}{8}+\frac{9}{4}\rho+\rho^2$	$+\frac{3}{8}+\frac{3}{2}\rho+\rho^2$	$+\frac{87}{67}\rho+\frac{39}{16}\rho^2+\rho^3$	$+\frac{3}{8}+\frac{1}{2}\rho$	$+\frac{21}{64}\rho+\frac{7}{16}\rho^2$	$+\frac{3}{8}+\frac{1}{4}\rho$	$+\frac{1}{32}\rho+\frac{1}{8}\rho^2$
$x_{21}(\rho)$	$\frac{9}{16}+\frac{3}{2}\rho+\rho^2$	$+\frac{3}{16}+\frac{1}{4}\rho$	$-\frac{3}{8}\rho-\frac{1}{2}\rho^2$	$+\frac{3}{16}+\rho+\rho^2$	$+\frac{9}{32}\rho+\frac{9}{8}\rho^2+\rho^3$	$+\frac{3}{16}+\frac{1}{4}\rho$	$-\frac{3}{8}\rho-\frac{1}{2}\rho^2$
$x_{31}(\rho)$	$\frac{9}{8}+\frac{9}{4}\rho+\rho^2$	$+\frac{3}{8}+\frac{1}{4}\rho$	$-\frac{3}{64}\rho+\frac{1}{16}\rho^2$	$+\frac{3}{8}+\frac{1}{2}\rho$	$+\frac{21}{64}\rho+\frac{7}{16}\rho^2$	$+\frac{3}{8}+\frac{3}{2}\rho+\rho^2$	$+\frac{41}{32}\rho+\frac{19}{8}\rho^2+\rho^3$
	$\frac{207}{8}+\frac{669}{8}\rho+\frac{355}{4}\rho^2+31\rho^3$	$+\frac{69}{8}+\frac{221}{8}\rho+\frac{103}{4}\rho^2+8\rho^3$	$+\frac{63}{32}\rho+\frac{269}{32}\rho^2+\frac{187}{16}\rho^3+\frac{21}{4}\rho^4$	$+\frac{69}{8}+\frac{257}{8}\rho+\frac{79}{2}\rho^2+16\rho^3$	$-\frac{297}{64}\rho-\frac{645}{64}\rho^2-\frac{71}{16}\rho^3+\rho^4$	$+\frac{69}{8}+\frac{201}{8}\rho+\frac{47}{2}\rho^2+7\rho^3$	$-\frac{17}{32}\rho+\frac{35}{32}\rho^2+\frac{37}{8}\rho^3+3\rho^4$

It can be shown that this tableau is optimal for $\rho \in [6.18, \infty)$, or equivalently, $\alpha \in [0, 0.14]$. For $\rho < 6.18$ the column of $x_{22}(\rho)$ becomes the next pivot column. The next tableau, after exchanging $x_{22}(\rho)$ with $x_{21}(\rho)$, is optimal for $\rho \in [0.91, 6.18]$, or equivalently, $\alpha \in [0.14, 0.52]$. For $\rho < 0.91$ the column of $x_{32}(\rho)$ becomes the next pivot column. The next tableau, after exchanging $x_{32}(\rho)$ with $x_{31}(\rho)$, is optimal for $\rho \in [0.27, 0.91]$, or equivalently, $\alpha \in [0.52, 0.79]$. Finally, for $\rho < 0.27$ the column of $x_{12}(\rho)$ becomes the pivot column. The final tableau, after exchanging $x_{12}(\rho)$ with $x_{11}(\rho)$, is optimal for $\rho \in (0, 0.27]$, or equivalently, $\alpha \in [0.79, 1)$. Hence, we have the following results.

$\alpha \in [0, 0.14]$: $f(1) = 1,\ f(2) = 1,\ f(3) = 1$ is the optimal policy.
$\alpha \in [0.14, 0.52]$: $f(1) = 1,\ f(2) = 2,\ f(3) = 1$ is the optimal policy.
$\alpha \in [0.52, 0.79]$: $f(1) = 1,\ f(2) = 2,\ f(3) = 2$ is the optimal policy.
$\alpha \in [0.79, 1)$: $f(1) = 2,\ f(2) = 2,\ f(3) = 2$ is the optimal policy.

The policy f^∞ with $f(1) = f(2) = 2$ is Blackwell optimal.
For the details of this example we refer to [125].

7.8 Bias optimality and policy iteration (unichain case)

We have seen in Example 6.7 that the policy iteration algorithm 6.4 will not find, in general, a bias optimal policy for unichain MDPs. We shall see that a simple adaptation of Algorithm 6.4 provides a correct algorithm.

Let (x, y) be a solution of the policy evaluation equation in the final iteration of Algorithm 6.4 corresponding to the average optimal policy f_*^∞, i.e. equation $x \cdot e + \{I - P(f_*)\}y = r(f_*)$, and let $t_i(a) := r_i(a) - \sum_j \{\delta_{ij} - p_{ij}(a)\}y_j - \phi$ for all $(i, a) \in S \times A$. Then, $x = \phi,\ t(f_*) = 0$ and $t_i(a) \leq 0$ for all $(i, a) \in S \times A$. Furthermore, define $A_2(i) := \{a \in A(i) \mid t_i(a) = 0\}$ for all $i \in S$, and $F^* := \{f^\infty \in C(D) \mid f(i) \in A_2(i)\}$. We know that y is unique up to a constant. So, although y is ambiguously defined, $A_2(i)$ is not ambiguous for all $i \in S$. Of course, f_*^∞ is contained in F^*, but F^* may contain additional policies when some state i has multiple actions a with $t_i(a) = 0$.

Note that F^* is the set of those policies f^∞ for which $\phi \cdot e + \{I - P(f)\}y = r(f)$. So, by (6.16), $u^0(f) = y - P^*(f)y$. Consequently, maximizing $u^0(f)$ over F^* is equivalent to maximizing $P^*(f)(-y)$ over F^*. The latter is the average reward in an altered MDP in which the action sets are $A_2(i),\ i \in S$, and the immediate reward in state i equals $-y$ for all $a \in A_2(i)$.

Theorem 7.17
If g^∞ is bias optimal, then $g^\infty \in F^*$.

Proof
Take any $f^\infty \notin F^*$. We shall show that f^∞ is not bias optimal. Let $(x = \phi, y)$ be a solution of the policy evaluation equation in the final iteration of Algorithm 6.4, i.e. $x \cdot e + \{I - P(f_*)\}y = r(f_*)$ for some policy f_*^∞ and $t_i(a) \leq 0$ for all $(i, a) \in S \times A$. Because $f^\infty \notin F^*$, we have $t(f) = r(f) - \{I - P(f)\}y - \phi \cdot e < 0$. Select policy g^∞ such that, for each $i \in S$, decision $g(i) \in A_2(i)$ and, in addition, $g(i) = f(i)$ whenever $t_i(f) = 0$. Hence, $g^\infty \in F^*$, so $t(g) = r(g) - \{I - P(g)\}y - \phi \cdot e = 0$. Similar to the proof of Theorem 6.10 we can show that either $\phi(f^\infty) < \phi(g^\infty)$ or $\phi(f^\infty) = \phi(g^\infty)$ and $u^0(f) < u^0(g)$. Hence, f^∞ is not a bias optimal policy. □

As a consequence of Theorem 7.17, optimizing $P^*(f)(-y)$ over F^* provides a bias optimal policy. Hence, the next algorithm determines a bias optimal policy and the bias value vector, where the *bias value vector* u^0 is defined by $u^0 := u^0(f)$ with f^∞ a bias optimal policy.

Algorithm 7.2 *Determination of a bias optimal policy by policy iteration (unichain case)*
Input: Instance of a unichain MDP.
Output: A bias optimal deterministic policy g^∞ and the bias value vector u^0.

```
1. Apply Algorithm 6.4, terminating with $(x = \phi, y)$.
2. for all $(i, a) \in S \times A$ do $t_i(a) := r_i(a) - \sum_j \{\delta_{ij} - p_{ij}(a\})y_j - \phi$
3. for all $i \in S$ do $A_2(i) := \{a \in A(i) \mid t_i(a) = 0\}$
4. for all $(i, a) \in S \times A_2$ do $r_i^2(a) := -y_i$
5. Apply Algorithm 6.4 to the altered MDP with action sets $A_2(i),\ i \in S$, and immediate rewards
   $r_i^2(a),\ (i, a) \in S \times A_2$ in order to find $g^\infty$ as an average optimal policy for the altered MDP with
   value vector $\phi^2$.
6. $g^\infty$ is a bias optimal policy for the original MDP with $u^0 := y + \phi^2$ as the bias value vector.
```

Example 6.7 (continued)
We will apply Algorithm 7.2 to Example 6.7.

1. If we apply Algorithm 6.4, starting with $f(1) = 2,\ f(2) = 1$, we terminate with $x = \phi = 4$ and $y = (0, 4)$.
2. $t_1(1) = t_1(2) = t_2(1) = 0$.
3. $A_2(1) = \{1, 2\},\ A_2(2) = \{1\}$.
4. $r_1^2(1) = 0,\ r_1^2(2) = 0,\ r_2^2(1) = -4$.
5. If we apply Algorithm 6.4, starting with $g(1) = 1,\ g(2) = 1$, we terminate with $\phi^2 = (0, 0)$ and with g^∞ as average optimal policy.
6. g^∞ a bias optimal policy and $u^0 = (0, 4) + (0, 0) = (0, 4)$ is the bias value vector.

7.9 Bias optimality and linear programming

7.9.1 The general case

In this section we present a three-step procedure that yields a bias optimal policy. In each step an MDP problem is solved. The first two of the three steps are to find an average optimal policy. In the first step the original MDP problem for average rewards is solved and the dual linear program provides an optimal solution $(x^* = \phi, y^*)$. In the second step an altered MDP is considered with value vector ϕ^2, similar as in Algorithm 7.2. We will show that the bias value vector u^0 satisfies $u_i^0 = y_i^* + \phi_i^2$ for all $i \in R$, where R is the set of the states that are recurrent under at least one bias optimal policy. Therefore, it remains to treat only the states that are transient under every bias optimal policy. Unfortunately, this set is unknown. However, we can solve the problem with a final MDP problem which is related to the total rewards of a certain, often simplified, MDP.

Program 1
We first solve the dual pair of linear programs for an average optimal policy. So, we compute optimal solutions $(v^* = \phi, u^*)$ and (x^*, y^*) of the pair of dual linear programs

$$min \left\{ \sum_j \beta_j v_j \;\middle|\; \begin{array}{rcll} \sum_j \{\delta_{ij} - p_{ij}(a)\} v_j & \geq & 0, & (i,a) \in S \times A \\ v_i + \sum_j \big(\delta_{ij} - p_{ij}(a)\big) u_j & \geq & r_i(a), & (i,a) \in S \times A \end{array} \right\} \tag{7.37}$$

and

$$max \left\{ \sum_{(i,a)} r_i(a) x_i(a) \;\middle|\; \begin{array}{llcl} \sum_{(i,a)} \{\delta_{ij} - p_{ij}(a)\} x_i(a) & & = & 0,\ j \in S \\ \sum_a x_j(a) & + \sum_{(i,a)} \{\delta_{ij} - p_{ij}(a)\} y_i(a) & = & \beta_j,\ j \in S \\ & x_i(a), y_i(a) & \geq & 0,\ (i,a) \in S \times A \end{array} \right\}, \tag{7.38}$$

where $\beta_j > 0,\ j \in S$, is arbitrarily chosen.

Program 2
Let

$$A_1(i) := \{a \in A(i) \mid \sum_j \{\delta_{ij} - p_{ij}(a)\} \phi_j = 0\},\ i \in S;$$
$$A_2(i) := \{a \in A_1(i) \mid \phi_i + \sum_j \big(\delta_{ij} - p_{ij}(a)\big) u^*_j = r_i(a)\},\ i \in S;$$
$$S_1 := \{i \in S \mid A_1(i) \neq \emptyset\};\ S_2 := \{i \in S \mid A_2(i) \neq \emptyset\}.$$

By equation (5.33), we have $S_1 = S$. Any policy $f^\infty \in C(D)$ induces a Markov chain $P(f)$. Let $R(f)$ and $T(f)$ be the sets of recurrent and transient states, respectively, in this Markov chain.

Lemma 7.11
Let $f^\infty \in C(D)$ be an average optimal policy. Then,
(1) $f(i) \in A_1(i),\ i \in S$.
(2) $f(i) \in A_2(i),\ i \in R(f)$.
(3) $u^0_i(f) = u^*_i - \{P^*(f)u^*\}_i,\ i \in R(f)$.
(4) $u^0_i(f) \leq u^*_i - \{P^*(f)u^*\}_i,\ i \in T(f)$.

Proof
(1) $P(f)\phi = P(f)P^*(f)r(f) = P^*(f)r(f) = \phi$. Consequently, $A_1(i) \neq \emptyset,\ i \in S$.
(2) From Theorem 5.20 it follows that (x^f, y^f), defined by (5.35) and (5.36), is an optimal solution of the dual program (5.29). In the proof of Theorem 5.20 is shown that $R(f) = S_x = \{i \mid x^f_i\big(f(i)\big) > 0\}$ and that $f(i) \in A_2(i),\ i \in S_x$ (see (5.37)).
(3) Since $d_{ij}(f) = 0$ for all $i \in R(f)$ and all $j \in T(f)$ (see section 5.3), it follows from part (2) that $u^0(f)_i = \{D(f)r(f)\}_i = \big\{D(f)\{\phi + [I - P(f)]u^*\}\big\}_i = \{[I - P^*(f)]u^*\}_i$ for all $i \in R(f)$.
(4) Since $d_{ij}(f) \geq 0$ for all $i, j \in T(f)$ (see section 5.3), we obtain

$$d_{ij}(f)\{\phi_j + \sum_k \big(\delta_{jk} - p_{jk}(f)\big)u^*_k\} \geq d_{ij}(f)r_j(f),\ i,j \in T(f).$$

Part (2) of this lemma implies

$$d_{ij}(f)\{\phi_j + \sum_k \big(\delta_{jk} - p_{jk}(f)\big)u^*_k\} = d_{ij}(f)r_j(f),\ i \in T(f),\ j \in R(f).$$

Hence, $u_i(f) = \{D(f)r(f)\}_i \leq \big\{D(f)\{\phi + [I - P(f)]u^*\}\big\}_i = \{[I - P^*(f)]u^*\}_i,\ i \in T(f)$. □

A bias optimal policy is an average optimal policy which in addition maximizes $u^0(f)$ over the set of average optimal policies. Lemma 7.11 shows that maximizing $u^0(f)$ over the average optimal policies is, for the states $i \in R(f) \subseteq S_2$, maximizing $-\{P^*(f)u^*\}_i$. Notice that $-P^*(f)u^*$ is the average reward for rewards $r^2_i(a) := -u^*_i$ for all (i,a). Lemma 7.11 also shows that $f(i) \in A_2(i),\ i \in R(f)$, for all average optimal policies. Since the states $S \backslash S_2$ are transient under all average optimal policies, we consider a modified MDP with state space S_2 and action sets $A_2(i),\ i \in S_2$. In order to have a correct MDP we have to remove in the states $i \in S_2$ the actions $a \in A_2(i)$ for which $p_{ij}(a) > 0$ for at least one state $j \in S \backslash S_2$. Since $R(f)$ is a closed set for all average optimal policies f^∞, actions corresponding to average optimal policies are not removed. Furthermore, all policies f^∞ in the modified MDP satisfy $\phi = P(f)\phi$ and $\phi + \{I - P(f)\}u^* = r(f)$. Hence,

$$\phi = P^*(f)r(f) = \phi(f^\infty) \text{ and } u^0(f) = D(f)r(f) = u^* - P^*(f)u^*. \tag{7.39}$$

In the next algorithm the states and actions of the modified MDP are constructed.

Algorithm 7.3 *Construction of the modified MDP*
Input: An MDP and the sets S_2 and $A_2(i),\ i \in S_2$, defined above Lemma 7.11.
Output: A correct MDP with state space S_2 and action sets $A_2(i),\ i \in S$.

```
1. if $p_{ij}(a) = 0$ for all $i \in S_2,\ a \in A_2(i),\ j \in S\backslash S_2$ then go to step 5.
2. select some $i \in S_2,\ a \in A_2(i)$ with $\sum_{j \in S\backslash S_2} p_{ij}(a) > 0$.
3. $A_2(i) := A_2(i)\backslash\{a\}$; if $A_2(i) = \emptyset$ then $S_2 := S_2\backslash\{i\}$.
4. return to step 1.
5. STOP.
```

The linear programs for an average optimal policy in the modified MDP are

$$\min\left\{\sum_j \beta_j w_j \;\middle|\; \begin{array}{rcll} \sum_j\{\delta_{ij} - p_{ij}(a)\}w_j & \geq & 0, & (i,a) \in S_2 \times A_2 \\ w_i + \sum_j\{\delta_{ij} - p_{ij}(a)\}z_j & \geq & -u^*_i, & (i,a) \in S_2 \times A_2 \end{array}\right\} \tag{7.40}$$

and

$$\max\left\{\sum_{(i,a)} (-u^*_i)t_i(a) \;\middle|\; \begin{array}{rcl} \sum_{(i,a)}\{\delta_{ij} - p_{ij}(a)\}t_i(a) & = & 0,\ j \in S_2 \\ \sum_a t_j(a) \;+\; \sum_{(i,a)}\{\delta_{ij} - p_{ij}(a)\}s_i(a) & = & \beta_j,\ j \in S_2 \\ t_i(a), s_i(a) & \geq & 0,\ (i,a) \in S_2 \times A_2 \end{array}\right\}. \tag{7.41}$$

where $\beta_j > 0,\ j \in S$, is arbitrarily chosen.

Program 3
Let (w^*, z^*) be an optimal solution of program (7.40) and let (t^*, s^*) be an extreme optimal solution of the program (7.41). From Chapter 5 we know that $w^* = \phi^2$, the value vector of the modified MDP, and that any f^∞_* satisfying $t^*_i\big(f_*(i)\big) > 0$ if $\sum_a t^*_i(a) > 0$ and $s^*_i\big(f_*(i)\big) > 0$ if $\sum_a t^*_i(a) = 0$ is an average optimal policy for the modified MDP.

In the sequel we will show (see Lemma 7.14) that f^∞_* is a bias optimal policy for every state i which is recurrent under at least one bias optimal policy. Notice this set of states is unknown. Furthermore, we have to determine a policy which is also bias optimal in the other states. Therefore, we use the following observation.

Let f^∞ be an average optimal policy for some MDP and suppose that the value vector of this MDP is the 0-vector. Then, the α-discounted reward vector $v^\alpha(f^\infty)$ satisfies by the Laurent series expansion $v^\alpha(f^\infty) = u^0(f) + \varepsilon(\alpha)$, where $lim_{\alpha\uparrow 1}\,\varepsilon(\alpha) = 0$. Hence, the bias term $u^0(f)$ may be considered as $lim_{\alpha\uparrow 1}\,v^\alpha(f^\infty)$, which is the total expected reward for policy f^∞ (assumed that the total expected reward for this policy is well-defined). Notice that the value vector of the average reward is the 0-vector if we use $r_i^3(a) := r_i(a) - \phi$ as immediate rewards for all (i,a).

We shall also show (see the proof of Lemma 7.15 below) that for a bias optimal policy f^∞ we have $u_i^0(f) \geq u_i^* + \phi_i^2$ for all $i \in S_2$. As we have seen in Chapter 4 we can use for total rewards the linear programming formulation for discounted rewards with $\alpha = 1$. We also include in the linear program the inequalities that the total reward is at least $u_i^* + \phi_i^2,\ i \in S_2$. Finally, from Lemma 7.11 part (1) it follows that only actions of $A_1(i),\ i \in S$, may be considered. By these observations the third set of linear programs are

$$min\left\{ \sum_j \beta_j g_j \;\middle|\; \begin{array}{rcll} \sum_j\{\delta_{ij} - p_{ij}(a)\}g_j & \geq & r_i(a) - \phi_i, & (i,a) \in S \times A_1 \\ g_i & \geq & u_i^* + \phi_i^2 & i \in S_2 \end{array} \right\} \tag{7.42}$$

and

$$max\left\{ \begin{array}{l} \sum_{(i,a)\in S\times A_1}(r_i(a) - \phi_i)q_i(a) \\ +\sum_{i\in S_2}(u_i^* + \phi_i^2)h_i \\ \end{array} \;\middle|\; \begin{array}{rl} \sum_{(i,a)\in S\times A_1}\{\delta_{ij} - p_{ij}(a)\}q_i(a) + \sum_{i\in S_2}\delta_{ij}h_i = & \beta_j,\ j \in S \\ q_i(a) \geq & 0,\ (i,a) \in S \times A_1 \\ h_i \geq & 0,\ i \in S_2 \end{array} \right\}. \tag{7.43}$$

Combining the above observation result in the following algorithm.

Algorithm 7.4 *Determination of a bias optimal policy by linear programming*
Input: Instance of an MDP.
Output: A bias optimal deterministic policy g^∞

```
1. Compute an optimal solution ($v^* = \phi, u^*$) of linear program (7.37).
2. for all $i \in S$ do
       begin $A_1(i) := \{a \in A(i) \mid \sum_j\{\delta_{ij} - p_{ij}(a)\}\phi_j = 0\}$;
             $A_2(i) := \{a \in A_1(i) \mid \phi_i + \sum_j\{\delta_{ij} - p_{ij}(a)\}u_j^* = r_i(a)\}$
       end
3. $S_2 := \{i \in S \mid A_2(i) \neq \emptyset\}$.
4. Determine the modified MDP with state space $S_2$ and action sets $A_2(i)$ by Algorithm 7.3.
5. Compute an optimal solution ($w^* = \phi^2, z^*$) of linear program (7.40) and an extreme optimal solution
   ($t^*, s^*$) of (7.41).
6. for all $i \in S_2$ do
       select $f_*(i)$ such that $t_i^*\big(f_*(i)\big) > 0$ if $\sum_a t_i^*(a) > 0$ and $s_i^*\big(f_*(i)\big) > 0$ if $\sum_a t_i^*(a) = 0$.
7. Compute an optimal solution $g^*$ of linear program (7.42) and an extreme optimal solution ($q^*, h^*$)
   of linear program (7.43).
8. $S_* := \{i \in S_2 \mid g_i^* = u_i^* + \phi_i^2\}$.
9. Select policy $g^\infty$ such that $g(i) = f_*(i)$ for $i \in S_*$ and $q_i^*\big(g(i)\big) > 0$ for $i \in S\backslash S_*$ (STOP).
```

Example 7.5

$S = \{1, 2, 3, 4\}$; $A(1) = A(2) = A(3) = A(4) = \{1, 2\}$. Let $\beta_1 = \beta_2 = \beta_3 = \beta_4 = \frac{1}{4}$.
$r_1(1) = 2,\ r_1(2) = 3;\ r_2(1) = 0,\ r_2(2) = 2;\ r_3(1) = 0,\ r_3(2) = -5;\ r_4(1) = 4,\ r_4(2) = 1.$
$p_{12}(1) = p_{13}(2) = p_{21}(1) = p_{23}(2) = p_{32}(1) = p_{34}(2) = p_{41}(1) = p_{43}(1) = 1$ (the other transition probabilities are 0).

1. The linear program (7.37) is:

$min\{\frac{1}{4}v_1 + \frac{1}{4}v_2 + \frac{1}{4}v_3 + \frac{1}{4}v_4\}$

subject to

$$\begin{array}{rcr}
v_1 - v_2 & \geq & 0 \\
v_1 - v_3 & \geq & 0 \\
-v_1 + v_2 & \geq & 0 \\
v_2 - v_3 & \geq & 0 \\
-v_2 + v_3 & \geq & 0 \\
v_3 - v_4 & \geq & 0 \\
-v_1 + v_4 & \geq & 0 \\
-v_3 + v_4 & \geq & 0 \\
v_1 + u_1 - u_2 & \geq & 2 \\
v_1 + u_1 - u_3 & \geq & 3 \\
v_2 - u_1 + u_2 & \geq & 0 \\
v_2 + u_2 - u_3 & \geq & 2 \\
v_3 - u_2 + u_3 & \geq & 0 \\
v_3 - u_2 + u_3 & \geq & -5 \\
v_4 - u_1 + u_4 & \geq & 4 \\
v_4 - u_3 + u_4 & \geq & 1
\end{array}$$

An optimal solution is: $v_1^* = \phi_1 = v_2^* = \phi_2 = v_3^* = \phi_3 = v_4^* = \phi_4 = 1$ (unique) and $u_1^* = 2,\ u_2^* = 1,\ u_3^* = 0,\ u_4^* = 6$ (not unique).

2. $A_1(1) = A_1(2) = A_1(3) = A_4(1) = \{1, 2\};\ A_2(1) = A_2(2) = A_2(3) = \{1, 2\},\ A_2(4) = \emptyset;$
3. $S_2 = \{1, 2, 3\}$.
4. Modified MDP: $S_2 = \{1, 2, 3\};\ A_2(1) = A_2(2) = \{1, 2\},\ A_2(3) = \{1\}$.
5. The primal program (7.40) is:

$min\{\frac{1}{4}w_1 + \frac{1}{4}w_2 + \frac{1}{4}w_3\}$

subject to

$$\begin{array}{rcr}
w_1 - w_2 & \geq & 0 \\
w_1 - w_3 & \geq & 0 \\
-w_1 + w_2 & \geq & 0 \\
w_2 - w_3 & \geq & 0 \\
-w_2 + w_3 & \geq & 0 \\
w_1 + z_1 - z_2 & \geq & -2 \\
w_1 + z_1 - z_3 & \geq & -2 \\
w_2 - z_1 + z_2 & \geq & -1 \\
w_2 + z_2 - z_3 & \geq & -1 \\
w_3 - z_2 + z_3 & \geq & 0
\end{array}$$

An optimal solution is: $w_1^* = \phi_1^2 = w_2^* = \phi_2^2 = w_3^* = \phi_3^2 = -\frac{1}{2}$ (unique) and $z_1^* = \frac{1}{2},\ z_2^* = 0,\ z_3^* = \frac{1}{2}$ (not unique). The dual program (7.41) becomes (without the nonnegativity constraints):

$max\{-2t_1(1)-2t_1(2)-t_2(1)-t_2(2)\}$

subject to

$$\begin{array}{rrrrrrrrrrl}
t_1(1) & +\ t_1(2) & -\ t_2(1) & & & & & & & & =\ 0 \\
-\ t_1(1) & & +\ t_2(1) & +\ t_2(2) & -\ t_3(1) & & & & & & =\ 0 \\
 & -\ t_1(2) & & -\ t_2(2) & +\ t_3(1) & & & & & & =\ 0 \\
t_1(1) & +\ t_1(2) & & & & +\ s_1(1) & +\ s_1(2) & -\ s_2(1) & & & =\ \frac{1}{4} \\
 & & t_2(1) & +\ t_2(2) & & -\ s_1(1) & & +\ s_2(1) & +\ s_2(2) & -\ s_3(1) & =\ \frac{1}{4} \\
 & & & & t_3(1) & & -\ s_1(2) & & s_2(2) & +\ s_3(1) & =\ \frac{1}{4}
\end{array}$$

An extreme optimal solution is: $t_1^*(1)=t_1^*(2)=t_2^*(1)=0,\ t_2^*(2)=t_3^*(1)=\frac{3}{8};\ s_1^*(1)=\frac{1}{4}$, $s_1^*(2)=0,\ s_2^*(1)=0,\ s_2^*(2)=\frac{1}{8},\ s_3^*(1)=0$.

6. $f_*(1)=1,\ f_*(2)=2,\ f_*(3)=1$.
7. The programs (7.42) and (7.43) are:

$min\{\frac{1}{4}g_1+\frac{1}{4}g_2+\frac{1}{4}g_3+\frac{1}{4}g_4\}$

subject to

$$\begin{array}{rrrrrl}
g_1 & -\ g_2 & & & \geq & 1; \\
g_1 & & -\ g_3 & & \geq & 2; \\
-\ g_1 & +\ g_2 & & & \geq & -1; \\
 & g_2 & -\ g_3 & & \geq & 1; \\
 & -\ g_2 & +\ g_3 & & \geq & -1; \\
 & & g_3 & -\ g_4 & \geq & -6; \\
-\ g_1 & & & +\ g_4 & \geq & 3; \\
 & & -\ g_3 & +\ g_4 & \geq & 0;
\end{array}
\qquad
\begin{array}{rrrrl}
g_1 & & & \geq & \frac{3}{2}; \\
 & g_2 & & \geq & \frac{1}{2}; \\
 & & g_3 & \geq & -\frac{1}{2};
\end{array}$$

and (without the nonnegativity constraints)

$max\{q_1(1)+2q_1(2)-q_2(1)+q_2(2)-g_3(1)-6q_3(2)+3q_4(1)+\frac{3}{2}h_1+\frac{1}{2}h_2-\frac{1}{2}h_3\}$

subject to

$$\begin{array}{rrrrrrrrrl}
q_1(1) & +\ q_1(2) & -\ q_2(1) & & & & -\ q_4(1) & & +\ h_1 & =\ \frac{1}{4} \\
-\ q_1(1) & & +\ q_2(1) & +\ q_2(2) & -\ q_3(1) & & & & +\ h_2 & =\ \frac{1}{4} \\
 & -\ q_1(2) & & -\ q_2(2) & +\ q_3(1) & +\ q_3(2) & & -\ q_4(2) & +\ h_3 & =\ \frac{1}{4} \\
 & & & & & -\ q_3(2) & +\ q_4(1) & +\ q_4(2) & & =\ \frac{1}{4}
\end{array}$$

respectively. An optimal solution of the primal program is $g_1^*=\frac{3}{2},\ g_2^*=\frac{1}{2},\ g_3^*=-\frac{1}{2},\ g_4^*=\frac{9}{2}$ and $q_1^*(1)=q_1^*(2)=q_2^*(1)=q_2^*(2)=q_3^*(1)=q_3^*(2)=0,\ q_4^*(1)=\frac{1}{4},\ q_4^*(2)=0,\ h_1^*=\frac{1}{2},\ h_2^*=\frac{1}{4}$, $h_3^*=\frac{1}{4}$ is an extreme optimal solution of the dual program.
8. $S_*=\{1,2,3\}$.
9. $g(1)=1,\ g(2)=2,\ g(3)=1,\ g(4)=1$ is a bias optimal policy.

In order to show that Algorithm 7.4 is correct, we need several lemmata.

Lemma 7.12

The policy g^∞, constructed in step 9 of Algorithm 7.4, is well-defined.

Proof

We have to show that $\sum_a q_j^*(a)>0$ for all $j\in S\backslash S_*$. Take any $j\in S\backslash S_*$, then either $j\in S\backslash S_2$ or $j\in S_2\backslash S_*$. In the last case we have $g_j^*>u_j^*+\phi_j^2$, implying, by the complementary slackness property of linear programming, that $h^*_j = 0$. So, if $j \in S \backslash S_*$, it follows from the constraints of program (7.43) that $\sum_a q^*_j(a) = \beta_j + \sum_{(i,a) \in S \times A_1} p_{ij}(a) q^*_i(a) \geq \beta_j > 0$. □

Lemma 7.13

Let f^∞ be an average optimal policy and let g be a feasible solution of program (7.42). Then, $\{[I - P(f)]g\}_i = r_i(f) - \phi_i$ for all $i \in R(f)$ and $g_i \geq u^0_i(f) + \{P^*(f)g\}_i$ for all $i \in T(f)$.

Proof

From Lemma 7.11 part (1) it follows that $f(i) \in A_1(i)$ for all $i \in S$. Hence, from the constraints of linear program (7.42) we obtain $[I - P(f)]g - r(f) + \phi \geq 0$. Since we have $p^*_{ii}(f) > 0,\ i \in R(f)$ and furthermore $P^*(f)\{[I - P(f)]g - r(f) + \phi\} = P^*(f)\{-r(f) + \phi\} = 0$, we obtain $\{[I - P(f)]g\}_i = r_i(f) - \phi_i,\ i \in R(f)$. Since $d_{ij}(f) \geq 0,\ i, j \in T(f)$ and $\{[I - P(f)]g\}_j = r_j(f) - \phi_j$ for all $j \in R(f)$, we can write

$$\begin{array}{rcl} 0 & \leq & \big\{D(f)\{[I - P(f)]g - r(f) + \phi\}\big\}_i \\ & = & \big\{D(f)\{[I - P(f)]g\}\big\}_i - \{D(f)r(f)\}_i + \{D(f)P^*(f)r(f)\}_i \\ & = & \{[I - P^*(f)]g\}_i - u^0_i(f),\ i \in T(f). \end{array}$$

□

Lemma 7.14

Let g^∞ be a bias optimal policy and let f^∞_* be an average optimal policy for the modified MDP as selected in step 6 of Algorithm 7.4. Then, $u^0_i = u^0_i(f_*) = u^*_i - \{P^*(f_*)u^*\}_i = u^*_i + \phi^2_i$ for all $i \in R(g)$.

Proof

By Lemma 7.11 part (2), $R(g) \subseteq S_2$. Take any $i \in R(g)$. Since g^∞ be a bias optimal policy, $u^0_i = u^0_i(g)$. Because f^∞_* is an optimal policy in the modified MDP, $\phi^2(f^\infty_*) = P^*(f)(-u^*) \geq P^*(g)(-u^*)$. Notice that, since the states $S \backslash S_2$ are transient under any average optimal policy, any policy in the modified MDP can be extended to an average optimal policy for the original MDP. Hence, by Lemma 7.11 part (3) and (4), we obtain $u^0_i = u^0_i(g) = u^*_i - \{P^*(g)u^*\}_i \leq u^*_i - \{P^*(f_*)u^*\}_i \leq u^0_i(f_*) \leq u^0_i$. Therefore, $u^0_i = u^0_i(f_*) = u^*_i - \{P^*(f_*)u^*\}_i = u^*_i + \phi^2_i$ for all $i \in R(g)$. □

Lemma 7.15

The bias value vector u^0 is the unique optimal solution of program (7.42).

Proof

By (7.39), we have $u^0_i \geq u^0_i(f_*) = u^*_i - \{P(f_*)u^*\}_i = u^*_i + \phi^2_i,\ i \in S_2$. We first show that u^0 is a feasible solution of program (7.42). Assume not, i.e. $\sum_j \{\delta_{ij} - p_{ij}(a)\} u^0_j < r_i(a) - \phi_i$ for some $(i, a) \in S \times A_1$.

Let g^∞ be a bias optimal policy and define the policy f^∞ by $f(j) := \begin{cases} g(j) & \text{if } j \neq i; \\ a & \text{if } j = i. \end{cases}$

Since $\{I - P(g)\}u^0 = \{I - P(g)\}u^0(g) = \{I - P(g)\}D(g)r(g) = \{I - P^*(g)\}r(g) = r(g) - \phi$, we have

$$\{r(f) + P(f)u^0 - u^0 - \phi\}_i > 0 \text{ and } \{r(f) + P(f)u^0 - u^0 - \phi\}_j = 0 \text{ for all } j \neq i. \tag{7.44}$$

$f(i) \in A_1(i),\ i \in S$, so $\phi = P^*(f)\phi$. Therefore, $0 \leq P^*(f)\{r(f) + P(f)u^0 - u^0 - \phi\} = \phi(f^\infty) - \phi \leq 0$, implying $P^*(f)\{r(f) + P(f)u^0 - u^0 - \phi\} = 0$. Hence, $p^*_{ii}(f) = 0$, i.e. $i \in T(f)$. Since $P(f)$ and $P(g)$ differ only in row i and $i \in T(f)$, we have $R(f) \subseteq R(g)$. For $j \in R(f)$ and $k \notin R(f)$, we have $d_{jk}(f) = d_{jk}(g) = 0$. Hence, $u^0_j(f) = \sum_{k \in R(f)} d_{jk}(f) r_k(f) = \sum_{k \in R(g)} d_{jk}(g) r_k(g) = u^0_j(g) = u^0_j,\ j \in R(f)$. Furthermore,

$$\{P^*(f)u^0\}_i = \sum_{j \in R(f)} p^*_{ij}(f) u^0_j = \sum_{j \in R(f)} p^*_{ij}(f) u^0_j(f) = \{P^*(f)D(f)r(f)\}_i = 0. \tag{7.45}$$

Since $d_{ii}(f) > 0$ and using (7.44) and (7.45), we can write

$$\begin{array}{rcl} u_i^0(f) & = & \{D(f)r(f)\}_i = \sum_j d_{ij}(f)r_j(f) \\ & > & \sum_j d_{ij}(f)\{[I-P(f)]u^0+\phi\}_j \;=\; \{D(f)[I-P(f)]u^0 + D(f)\phi\}_i \\ & = & \{[I-P^*(f)]u^0\}_i + \{D(f)P^*(f)\phi\}_i \;=\; u_i^0, \end{array}$$

implying a contradiction. So, we have shown that u^0 is a feasible solution of program (7.42).

Finally we show that u^0 is the, componentwise, smallest solution of (7.42).

Let w be an arbitrary feasible solution of (7.42).

If $j \in R(g)$, then Lemma 7.14 implies $u_j^0 = u_j^0(f_*) = u_j^* + \phi_j^2 \leq w_j$.

If $j \in T(g)$, then Lemma 7.13 implies $w_j \geq u_j^0(g) + \{P^*(g)w\}_j \geq u_j^0(g) = u_j^0$, the last inequality because $0 = P^*(g)D^*(g)r(g) = P^*(g)u^0(g) = P^*(g)u^0 \leq P^*(g)w$ (this inequality because $u_j^0 \leq w_j,\ j \in R(g)$).

Hence, we have shown that $u_j^0 \leq w_j$ for all $j \in S$. □

Lemma 7.16

$u_i^0 - \{P(f_*)u^0\}_i = r_i(f_*) - \phi_i,\ i \in S_*$.

Proof

We consider the modified MDP with state space S_2 and action sets $A(i),\ i \in S_2$. From the constraints of (7.42) and the feasibility of u^0 for (7.42) it follows that $u_i^0 - \{P(f_*)u^0\}_i \geq r_i(f_*) - \phi_i$ for all $i \in S_*$. Suppose that $u_i^0 - \{P(f_*)u^0\}_i > r_i(f_*) - \phi_i$ for some $i \in S_*$. Since $P^*(f_*)\{u^0 - P(f_*)u^0 - r(f_*) - \phi\} = 0$, state $i \notin R(f_*)$, i.e. $i \in T(f_*)$. Because $\{u^0 - P(f_*)u^0 - r(f_*) - \phi\}_j = 0$ and $d_{ij}(f_*) \geq 0$ for all $j \in T(f_*)$ and $d_{ii}(f_*) > 0$, we can write

$$\begin{array}{rcl} 0 & < & \big\{D(f_*)[u^0 - P(f_*)u^0 - r(f_*) - \phi]\big\}_i = \big\{D(f_*)[u^0 - P(f_*)u^0 - r(f_*) - P^*(f_*)r(f_*)]\big\}_i \\ & = & \{u^0 - P^*(f_*)u^0 - u^0(f_*)\}_i \leq \{u^0 - P^*(f_*)u^0(f_*) - u^0(f_*)\}_i = \{u^0 - u^0(f_*)\}_i. \end{array}$$

From Lemma 7.15 it follows that $S_* = \{j \in S_2 \mid u_j^0 = u_j^* + \phi_j^2\}$. Relation (7.39) and the optimality of f_*^∞ in the modified MDP imply that $u_j^0(f_*) = u_j^* - \{P^*(f_*)u^*\}_j = u_j^* + \phi_j^2,\ j \in S_2$. Hence,

$$u_j^0(f_*) = u_j^0,\ j \in S_*, \tag{7.46}$$

contradicting the previous statement $0 < \{u^0 - u^0(f_*)\}_i$. □

Lemma 7.17

S_* is a closed set in the Markov chain $P(g)$, where g^∞ is defined in step 9 of Algorithm 7.4.

Proof

Since $g(i) = f_*(i),\ i \in S_*$ and S_2 is closed in the Markov chain $P(f_*)$, we have to show that S_* is a closed set in in the Markov chain $P(f_*)$ for the modified MDP. From Lemma 7.16 and relation (7.46) we have for all $i \in S_*$

$$\begin{array}{rcl} 0 & = & \{u^0 - P(f_*)u^0 - r(f_*) - P^*(f_*)r(f_*)\}_i = \{u^0(f_*) - P(f_*)u^0 - r(f_*) - P^*(f_*)r(f_*)\}_i \\ 0 & = & \big\{u^0(f_*) + P(f_*)[u^0(f_*) - u^0] - P(f_*)u^0(f_*) - r(f_*) - P^*(f_*)r(f_*)\big\}_i \\ 0 & = & \{P(f_*)[u^0(f_*) - u^0]\}_i + \{[I - P(f_*)]D(f_*)r(f_*)\}_i - \{[I - P^*(f_*)]r(f_*)\}_i \\ 0 & = & \{P(f_*)[u^0(f_*) - u^0]\}_i = \sum_j p_{ij}(f_*)\{u^0(f_*) - u^0\}_j = \sum_{j \notin S_*} p_{ij}(f_*)\{u^0(f_*) - u^0\}_j. \end{array}$$

Since $u^0(f_*) = u^* - P^*(f_*)u^* \leq u^* + \phi^2$ on S_2, we have $u_j^0(f_*) - u_j^0 \leq u_j^* + \phi_j^* - u_j^0 < 0$ for all $j \notin S_*$. Hence, $p_{ij}(f_*) = 0,\ i \in S_*,\ j \notin S_*$, i.e. S_* is a closed set in the Markov chain $P(f_*)$. □

Lemma 7.18

The states of $S\backslash S_*$ are transient in the Markov chain $P(g)$, where g^∞ is defined in step 9 of Algorithm 7.4.

Proof

Suppose there is a state $j \in S\backslash S_*$ which is recurrent under $P(g)$. Since S_* is closed under $P(f_*)$ there exists a nonempty ergodic set $J \subseteq S\backslash S_*$. Let $J = \{j_1, j_2, \dots, j_m\}$. The constraints of program (7.43) imply

$$\begin{array}{lcl} \sum_{a \in A_1(j)} q_j^*(a) + h_j^* & = & \beta_j + \sum_{(i,a) \in S \times A_1} p_{ij}(a) q_i^*(a) \geq \beta_j > 0,\ j \in S_2 \\ \sum_{a \in A_1(j)} q_j^*(a) & = & \beta_j + \sum_{(i,a) \in S \times A_1} p_{ij}(a) q_i^*(a) \geq \beta_j > 0,\ j \in S\backslash S_2 \end{array}$$

Since (q^*, h^*) is an extreme optimal solution and since the linear program has N equality constraints, for each state j either $h_j^* > 0$ (and $q_j^*(a) = 0$ for all $a \in A_1(j)$) or $q_j^*(a) > 0$ for exactly one action, say action a_j (the other variables h_j^* and $q_j^*(a),\ a \neq a_j$, are zero). From the complementary slackness property of linear programming it follows that $h_j^* = 0$ for all $j \in S_2\backslash S_*$. Hence, in every state j_i of J we have exactly one positive variable, namely $q_{j_i a_{j_i}}^*,\ i = 1, 2, \dots, m$. The corresponding column vectors of the linear program with elements $\delta_{j_i k} - p_{j_i k}(a_{j_i}),\ k = 1, 2, \dots, N$, are linearly independent. Since J is closed, $\delta_{j_i k} - p_{j_i k}(a_{j_i}) = 0,\ k \notin J$. Therefore, we have

$$\sum_{k \in J} \{\delta_{j_i k} - p_{j_i k}(a_{j_i})\} = \sum_{k=1}^N \{\delta_{j_i k} - p_{j_i k}(a_{j_i})\} = 1 - 1 = 0,\ i = 1, 2, \dots, m,$$

which contradicts the linear independence of these vectors. □

Theorem 7.18

The policy g^∞, defined in step 9 of Algorithm 7.4, is bias optimal.

Proof

Since $q_i^*\big(g(i)\big) > 0$ for $i \in S\backslash S_*$, it follows from the complementary slackness property that

$$u_i^0 - \{P(g)u^0\}_i = r_i(g) - \phi_i,\ i \in S\backslash S_*. \tag{7.47}$$

$P(g)$ and $P(f_*)$ have the same rows on the closed set S_*. Then, by (7.47) and Lemma 7.16, we obtain $u^0 - P(g)u^0 = r(g) - \phi$. Since $g(i) \in A_1(i),\ i \in S$, we have $\phi = P(g)\phi = P^*(g)\phi$. Consequently, $0 = P^*(g)\{u^0 - P(g)u^0\} = P^*(g)\{r(g) - \phi\} = \phi(g^\infty) - \phi$, implying $D(g)\phi = D(g)P^*(g)r(g) = 0$ and $u^0(g) = D(g)r(g) = D(g)\{I - P(g)\}u^0 = u^0 - P^*(g)u^0$. Because $u_j^0(g) = u_j^0$ for all $j \in S_*$ (see (7.46)) and because the states of $S\backslash S_*$ are transient in the Markov chain $P(g)$ (see Lemma 7.18), we obtain $u^0(g) = u^0 - P^*(g)u^0 = u^0 - P^*(g)u^0(g) = u^0$. Hence, g^∞ is a bias optimal policy. □

7.9.2 The unichain case

Linear programming for bias optimality in the irreducible case was discussed in section 7.6.2. The present section deals with the unichain case. In the unichain case the value vector ϕ is constant and we consider ϕ as a scalar. Program (7.37) becomes

$$min\ \{v \mid v + \sum_j \big(\delta_{ij} - p_{ij}(a)\big)u_j \geq r_i(a),\ (i,a) \in S \times A\}. \tag{7.48}$$

with as dual program

$$max \left\{ \sum_{(i,a)} r_i(a)x_i(a) \left| \begin{array}{rcl} \sum_{(i,a)} \{\delta_{ij} - p_{ij}(a)\}x_i(a) & = & 0,\ j \in S \\ \sum_{(i,a)} x_i(a) & = & 1 \\ x_i(a) & \geq & 0,\ (i,a) \in S \times A \end{array} \right. \right\}. \tag{7.49}$$

Let the optimal solutions of (7.48) and (7.49) be $(v^* = \phi, u^*)$ and x^*, respectively. Define $A_2(i),\ i \in S$, and S_2 as in section 7.9.1, i.e. $A_2(i) := \{a \in A(i) \mid \phi + \sum_j \big(\delta_{ij} - p_{ij}(a)\big)u_j^* = r_i(a)\}$ for all $i \in S$ and $S_2 := \{i \in S \mid A_2(i) \neq \emptyset\}$. Also the programs for the modified MDP simplify and become

$$min\ \{w \mid w + \sum_j \{\delta_{ij} - p_{ij}(a)\}z_j \geq -u_i^*,\ (i,a) \in S_2 \times A_2\} \tag{7.50}$$

and

$$max\ \left\{ \sum_{(i,a)} (-u_i^*)t_i(a) \,\middle|\, \begin{array}{rcl} \sum_{(i,a)} \{\delta_{ij} - p_{ij}(a)\}t_i(a) & = & 0,\ j \in S_2 \\ \sum_{(i,a)} t_i(a) & = & 1 \\ t_i(a) & \geq & 0,\ (i,a) \in S_2 \times A_2 \end{array} \right\}. \tag{7.51}$$

In this unichain case the algorithm becomes as follows (the proof of correctness follows straightforward from the previous section).

Algorithm 7.5 *Determination of a bias optimal policy by linear programming (unichain case)*
Input: Instance of a unichain MDP.
Output: A bias optimal deterministic policy g^∞

```
1. Compute an optimal solution $(v^* = \phi, u^*)$ of linear program (7.48).
2. for all $i \in S$ do $A_2(i) := \{a \in A(i) \mid \phi + \sum_j\{\delta_{ij} - p_{ij}(a)\}u_j^* = r_i(a)\}$.
3. $S_2 := \{i \in S \mid A_2(i) \neq \emptyset\}$.
4. Determine the modified MDP with state space $S_2$ and action sets $A_2(i)$ by Algorithm 7.3.
5. Compute an optimal solution $(w^* = \phi^2, z^*)$ of linear program (7.50) and an extreme optimal solution
   $t^*$ of (7.51).
6. for all $i \in S_2$ do
       select $f_*(i)$ such that $t_i^*\big(f_*(i)\big) > 0$ if $\sum_a t_i^*(a) > 0$ and arbitrary from $A_2(i)$ if $\sum_a t_i^*(a) = 0$.
7. Compute an optimal solution $g^*$ of linear program (7.42) and an extreme optimal solution $(q^*, h^*)$
   of linear program (7.43), where $A_1(i) = A(i)$ for all $i \in S$.
8. $S_* := \{i \in S_2 \mid g_i^* = u_i^* + \phi_i^2\}$.
9. Select policy $g^\infty$ such that $g(i) = f_*(i)$ for $i \in S_*$ and $q_i^*\big(g(i)\big) > 0$ for $i \in S \backslash S_*$ (STOP).
```

7.10 Turnpike results and bias optimality (unichain case)

This section deals with the undiscounted MDP when the planning horizon is long but fixed and finite. Throughout this section we assume that each transition matrix $P(f)$ is an aperiodic Markov chain which has one ergodic set plus perhaps some transient states. Let v^n be the N-vector whose ith component is the maximum total expected reward when the initial state is i and when the planning horizon consists of n epochs. With $v^0 := 0$, the standard recursive relation of dynamic programming characterizes v^n by the equation system

$$v_i^n = max_a\,\{r_i(a) + p_{ij}(a)v_j^{n-1}\},\ i \in S,\ n = 1, 2, \dots. \tag{7.52}$$

Let f_n be an optimal decision rule in epoch n, i.e. $v^n = r(f_n) + P(f_n)v^{n-1}$. Then, policy R_n, defined by $R_n := (f_1, f_2, \dots, f_n)$, is an optimal policy for the total rewards over this finite horizon MDP, also called a *time-optimal policy.*

This section analyses the asymptotic behavior of v^n and f_n as n approaches infinity. Theorems describing such behavior are often called *turnpike theorems.* A policy $R_n = (f_1, f_2, \dots, f_n)$ is called *eventually stationary* if an integer m exists, with $1 \leq m < n$ such that $f_m = f_{m+1} = f_{m+2} = \cdots$. An interesting question is: is the time-optimal policy eventually stationary? One might hope so, since the effect of the end of the planning horizon on decision made at the beginning can be expected to vanish as the planning horizon approaches infinity. But this need not be the case, as illustrated by the next example.

Example 7.6

Consider the following MDP with transition probabilities and immediate rewards that are dependent on scalars p and q with $0 < p < q < 1$.

$S = \{1, 2\};\ A(1) = \{1, 2\},\ A(2) = \{1\};\ p_{11}(1) = p,\ p_{12}(1) = 1 - p;\ p_{11}(2) = q,\ p_{12}(2) = 1 - q;$
$p_{21}(1) = 1,\ p_{22}(1) = 0;\ r_1(1) = 2 - p,\ r_1(2) = 2 - q,\ r_2(1) = 0.$

The are two deterministic policies f_1^∞ and f_2^∞ with $f_1(1) = 1$ and $f_2(1) = 2$, respectively. The stationary distribution π^1 of the Markov chain $P(f_1)$ satisfies $\pi^1 = \big(\frac{1}{2-p}, \frac{1-p}{2-p}\big)$. Hence, $\phi(f_1^\infty) = (\pi^1)^T r(f_1) = 1$.

The solution of the policy evaluation equation (6.15), i.e. $x{\cdot}e + \{I - P(f)\}y = r(f)$, with y_1 is $x = 1,\ y_1 = 0$ and $y_2 = -1$. Since, by (6.16), $u^0(f) = y - P^*(f)y$, we obtain $u^0(f_1) = \big(\frac{1-p}{2-p}, \frac{-1}{2-p}\big)$.

Similarly, the computation for f_2^∞ yields $\pi^2 = \big(\frac{1}{2-q}, \frac{1-q}{2-q}\big)$, $\phi(f_2^\infty) = 1$ and $u^0(f_2) = \big(\frac{1-q}{2-q}, \frac{-1}{2-q}\big)$.

Since $0 < p < q < 1$, $u_1^0(f_1) > u_1^0(f_2)$ and $u_2^0(f_1) > u_2^0(f_2)$. So, f_1^∞ and f_2^∞ are both optimal policies, but f_1^∞ is bias optimal and f_2^∞ not.

We shall show that the time-optimal policy R_n oscillates between f_1^∞ and f_2^∞. The proof that this occurs can be given by observing that t^n oscillates around 0, where $t^n := v_1^n - v_2^n - 1$.

$v_1^n = max\{w^n(1), w^n(2)\}$, where $w^n(1) := 2{-}p{+}pv_1^{n-1}{+}(1{-}p)v_2^{n-1}$ and $w^n(2) := 2{-}q{+}qv_1^{n-1}{+}(1{-}q)v_2^{n-1}$.

Note that $w^n(1) - w^n(2) = -(p - q) + (p - q)v_1^{n-1} - (p - q)v_2^{n-1} = (p - q)t^{n-1}$ and $(p - q) < 0$.

Hence, $f_n(1) = 1$ if $t^{n-1} < 0$ and $f_n(1) = 2$ if $t^{n-1} > 0$.

If $t^{n-1} < 0$, then $t^n = v_1^n - v_2^n - 1 = \{2 - p + pv_1^{n-1} + (1 - p)v_2^{n-1}\} - v_1^{n-1} - 1 = (p - 1)t^{n-1} > 0$.

Similarly, if $t^{n-1} < 0$, then $t^n = (q - 1)t^{n-1} < 0$.

Since $t^0 = -1$, the sequence t^n oscillates, and $f_n(1) = 1$ when n is odd and $f_n(1) = 2$ when n is even.

In this example we will also show that $v^n \to n \cdot \phi + u^* + d \cdot e$, with $d > 0$, as $n \to \infty$. To compute the asymptotic form of v^n, one can exploit the fact that $f_n(1)$ oscillates. Working first with n even, we can write with $y^n := v^{2n},\ n = 1, 2, \dots$

$$\begin{aligned} y^n &= v^{2n} = r(f_2) + P(f_2)v^{2n-1} = r(f_2) + P(f_2)\{r(f_1) + P(f_1)v^{2n-2}\} \\ &= \{r(f_2) + P(f_2)r(f1)\} + P(f_2)P(f_1)y^{n-1}. \end{aligned}$$

Let $r := r(f_2) + P(f_2)r(f_1) = \binom{2+s}{2-p}$ and $P := P(f_2)P(f_1) = \begin{pmatrix} 1-s & s \\ p & 1-p \end{pmatrix}$, where $s := q(1 - p) \in [0, 1]$.

Then, $y^n = r + Py^{n-1}$ for $n = 1, 2, \dots$. Note that the Markov chain P is aperiodic and irreducible. Hence, $P^n \to P^*$ as $n \to \infty$. From Theorem 5.8, part (3), it follows that $y^n = (n\phi) \cdot e + u^0 + P^n u^0$ for all $n \in \mathbb{N}$.

Therefore, $y^n \to (ng) \cdot e + u^0$ as as $n \to \infty$, where $g = \pi^T r$ with π the stationary distribution of P.

A little algebra gives: $\pi_1 = \frac{p}{p+s},\ \pi_2 = \frac{s}{p+s};\ u_1^0 = \frac{s}{p+s},\ u_2^0 = -\frac{p}{p+s}$ and $g_1 = g_2 = 2$.

Since $v^{2n} = w^n$, we obtain $v^{2n} \to 2n \cdot \binom{1}{1} + \begin{pmatrix} 1 - \frac{p}{p+q-pq} \\ -\frac{p}{p+q-pq} \end{pmatrix}$.

Because in the original MDP the average value ϕ and the bias value vector u^* satisfy $\phi = 1$ and $u_1^* = \frac{1-p}{2-p},\ u_2^* = \frac{-1}{2-p}$, we have $v^{2n} \to (2n\phi) \cdot e + u^* + d \cdot e$ with $d := \frac{1}{2-p} - \frac{p}{p+q-pq}$ as $n \to \infty$.

It is easy to verify that $d > 0$.

Next, we consider n odd. We can write

$$v^{2n+1} = r(f_1) + P(f_1)v^{2n} \rightarrow r(f_1) + P(f_1)\{(2n\phi)\cdot e + u^* + d\cdot e\} \text{ as } n \rightarrow \infty.$$

Since $\phi \cdot e = P^*(f_1)r(f_1),\ u^* = D(f_1)r(f_1)$ and $I + P(f_1)D(f_1) = D(f_1) + P^*(f_1)$, we have

$$v^{2n+1} \rightarrow (2n+1)\phi\cdot e + u^* + d\cdot e \text{ as } n \rightarrow \infty.$$

Hence, for all n we have $v^n \rightarrow (n\phi)\cdot e + u^* + d\cdot e$, with $d > 0$, as $n \rightarrow \infty$.

In the sequel we will show that in any unichain MDP $v^n \rightarrow (n\phi)\cdot e + u^* + d\cdot e$, with $d > 0$, as $n \rightarrow \infty$. Let $e^n := v^n - (n\phi)\cdot e - u^*$ for $n = 0, 1, \dots$ (cf. (5.46)). Let f_*^∞ be a bias optimal policy and let $\pi(f_*)$ be the stationary distribution of $P(f_*)$. Then, $\pi(f_*)^T u^* = \pi(f_*)^T D(f_*)r(f_*) = 0$. Note that $e^0 = -u^*$, so $\pi(f_*)^T e^0 = 0$, implying that e^0 has normally both positive and negative elements.
First, we have

$$\begin{array}{rcl} e^n & = & max_a\,\{r_i(a) + \sum_j\, p_{ij}(a)v_j^{n-1} - n\cdot\phi - u_i^*\} \\ & = & max_a\,\{r_i(a) + \sum_j\, p_{ij}(a)[e_j^{n-1} + (n-1)\cdot\phi + u_j^*] - n\cdot\phi - u_i^*\} \\ & = & max_a\,\{[r_i(a) + \sum_j\, p_{ij}(a)u_j^* - \phi - u_i^*] + \sum_j\, p_{ij}(a)e_j^{n-1}\} \\ & = & max_a\,\{r_i^*(a) + \sum_j\, p_{ij}(a)e_j^{n-1}\}, \end{array}$$

where $r_i^*(a) := r_i(a) + \sum_j\, p_{ij}(a)u_j^* - \phi - u_i^*$ for all $(i,a) \in S \times A$.

The equation $e^n = max_a\,\{r_i^*(a) + \sum_j\, p_{ij}(a)e_j^{n-1}\},\ i \in S$, is the value iteration for an MDP with immediate rewards $r_i^*(a)$ and with terminal reward $e^0 := -u^*$. Consider Algorithm 6.4 with the bias optimal policy f_*^∞ as starting policy. Then, by Theorem 6.10, the algorithm terminates in the first iteration. Hence, we have $r_i(a) + \sum_j\, p_{ij}(a)y_j \le \phi(f_*^\infty) + y_i = \phi + y_i$ for all $(i,a) \in S \times A$. Since, by (6.16), $y = u^0(f_*) + c\cdot e$ for some scalar c, we obtain $r_i^*(a) \le 0$ for all $(i,a) \in S \times A$.

Furthermore, the maximizing actions for the value iteration scheme $\{v^n\}$ and $\{e^n\}$ are the same, namely:

$$\begin{array}{rcl} f_n(i) & = & argmax\{r_i(a) + \sum_j\, p_{ij}(a)v_j^{n-1}\} \ \Leftrightarrow \\ f_n(i) & = & argmax\{r_i(a) + \sum_j\, p_{ij}(a)[e_j^{n-1} + (n-1)\phi + u_j^*]\} \ \Leftrightarrow \\ f_n(i) & = & argmax\{r_i(a) + \sum_j\, p_{ij}(a)u_j^* + \sum_j\, p_{ij}(a)e_j^{n-1}\} \ \Leftrightarrow \\ f_n(i) & = & argmax\{r_i(a) + \sum_j\, p_{ij}(a)u_j^* - \phi - u_i^* + \sum_j\, p_{ij}(a)e_j^{n-1}\} \ \Leftrightarrow \\ f_n(i) & = & argmax\{r_i^*(a) + \sum_j\, p_{ij}(a)e_j^{n-1}\}. \end{array}$$

Let f_*^∞ be a bias optimal policy and let $R_n := (f_1, f_2, \dots, f_n),\ n \ge 1$, be a time optimal policy. Then, by Lemma 5.8, part (1),

$$P(f_*)e^n \le e^{n+1} \le P(f_{n+1})e^n \text{ for } n = 0, 1, \dots. \tag{7.53}$$

Then, it follows that $(min_j\, e_j^n)\cdot e = (min_j\, e_j^n)\cdot P(f_*)e \le P(f_*)e^n \le e^{n+1}$. Hence, $min_j\, e_j^n \le min_j\, e_j^{n+1}$, i.e. $min_j\, e_j^n$ is nondecreasing in n. Similarly, one can show that $max_j\, e_j^n$ is nonincreasing in n. In particular, since $e^0 = -u^*$, we have

$$min_j\,(-u_j^*) \le e_i^n \le max_j\,(-u_j^*) \text{ for all } i \in S \text{ and } n = 0, 1, \dots. \tag{7.54}$$

Define $m_i := \liminf_{n\rightarrow\infty} e^i,\ M_i := \limsup_{n\rightarrow\infty} e^i,\ i \in S$, and $\underline{m} := min_i\, m_i,\ \overline{M} := max_i\, M_i$.

Lemma 7.19
If $\pi_k(f_*) > 0$, then $\underline{m} = m_k = M_k$.

Proof

In the proof of Theorem 5.24 we have already shown that $m \geq P^*(f_*)M$. Therefore, it follows that $P^*(f_*)m \geq P^*(f_*)M \geq P^*(f_*)m$, i.e. $P^*(f_*)(M-m) = 0$. Hence, if $\pi_k(f_*) > 0$, then $M_k = m_k$. From $m \geq P^*(f_*)M$, we obtain $m \geq P^*(f_*)m$, i.e. $m_i \geq \sum_j \pi_j(f_*)m_j,\ i \in S$, implying $\underline{m} \geq \sum_j \pi_j(f_*)m_j \geq \underline{m}$. Hence, if $\pi_k(f_*) > 0$, then $m_k = \underline{m}$, which completes the proof that, if $\pi_k(f_*) > 0$, $\underline{m} = m_k = M_k$. □

Let $S_+ := \{k \mid \pi_k(f_*) > 0\}$. It is well known that S_+ is closed under $P(f_*)$. Furthermore, define S_M by $S_M := \{i \mid M_i = M\}$. The next lemma shows that S_M is closed under $P(f_M)$, where $f_M(i)$ for any $i \in S$ is defined by the following procedure:

Let $\{n_l\},\ l = 1, 2, \dots$ be a subsequence such that:

(1) $M_i = \lim_{l \to \infty} e^{n_l+1}$.

(2) $f_M(i) = f_{n_l+1}(i)$ for $l = 1, 2, \dots$ (since $A(i)$ is finite $f_M(i)$ is well-defined).

Lemma 7.20

S_M is closed under $P(f_M)$.

Proof

Since, by Lemma 5.8, part (1), $e^{n_l+1} \leq P(f_{n_l+1})e^{n_l}$, we can write for all $i \in S$,

$M_i = \lim_{l\to\infty} e^{n_l+1} \leq \limsup_{l\to\infty} \{\sum_j p_{ij}(f_M)e_j^{n_l}\} \leq \sum_j p_{ij}(f_M) \cdot \limsup_{l\to\infty} e_j^{n_l} \leq \sum_j p_{ij}(f_M)M_j$.

For $i \in S_M$, we have $M_i = \overline{M}$. Hence, $0 \geq \sum_j p_{ij}(f_M)(M_j - \overline{M}) = \sum_j p_{ij}(f_M)M_j - \overline{M} \geq M_i - \overline{M} = 0$. Since $p_{ij}(f_M)(M_j - \overline{M}) = 0$ for all $i \in S_M$ and all $j \in S$, we have $p_{ij}(f_M) = 0$ for $i \in S_M$ and $j \notin S_M$, i.e. S_M is closed under $P(f_M)$. □

Lemma 7.21

$\underline{m} = \overline{M}$ and $e^n \to d \cdot e$ for some $d \geq 0$.

Proof

Suppose $\underline{m} < \overline{M}$. Lemma 7.19 verifies $M_k = m_k = \underline{m}$, whenever $k \in S_+$. So, since $\underline{m} < \overline{M}$, the sets S_+ and S_M are disjoint. Define g^∞ such that $g(i) := f_*(i),\ i \in S_+$ and $g(i) := f_M(i),\ i \in S_M$. Then $P(g)$ has (at least) two ergodic subchains, which contradicts our hypothesis in this section. Therefore, $\underline{m} = \overline{M}$, i.e. $\lim_{n\to\infty} e^i$ exits for all $i \in S$ and is independent of i. Denote this limit by d; then, $e^n \to d \cdot e$ as $n \to \infty$. To see that $d \geq 0$, we use again Lemma 5.8, part (1), which provides,

$$e^n \geq P(f_*)e^{n-1} \geq \cdots \geq P^n(f_*)e^0 = -P^n(f_*)u^* = -P^n(f_*)D(f_*)r(f_*) \to -P^*(f_*)D(f_*)r(f_*) = 0.$$

Hence, $d \cdot e = \lim_{n\to\infty} e^n \geq 0$. □

Lemma 7.22

Let $A_2(i) := \{a \in A(i) \mid r_i^*(a) = 0\},\ i \in S$. Then, for all sufficiently large n, $f_n(i) \in A_2(i)$ for all $i \in S$.

Proof

We have already observed that $r_i^*(a) \leq 0$ for all $(i,a) \in S \times A$. Let $\varepsilon := -max_{i,a}\,\{r_i^*(a) \mid r_i^*(a) < 0\} > 0$. Take n_1 big enough such that $\|e^n - d \cdot e\| < \frac{1}{2}\varepsilon$ for all $n \geq n_1$. For $n > n_1$ and $a \notin A_2(i)$, we obtain $r_i^*(a) + \sum_j p_{ij}(a)e_j^{n-1} < -\varepsilon + \frac{1}{2}\varepsilon + d = d - \frac{1}{2}\varepsilon < e_i^n$. Hence, $a \neq f_n(i)$. □

A simple implication of Lemma 7.22 is that R_n is eventually stationary whenever $|A_2(i)| = 1$ for all $i \in S$, which is often the case. Define $v^{m,n}$ in terms of a planning horizon that is $m+n$ epochs long, where $v_i^{m,n}$

be the total expected reward for starting in state i, using a bias optimal policy f_*^∞ for the first m epochs and using the time-optimal policy R_n for the remaining n epochs. In the beginning of Section 6.2.2 we have derived that $v^m(f^\infty) = m \cdot \phi(f^\infty) + u^0(f) - P^m(f)u^(f)$ for every $f^\infty \in C(D)$. Hence, we have

$$\begin{aligned} v^{m,n} &= (m\phi) \cdot e + u^* - P^m(f_*)u^* + P^m(f_*)v^n \\ &= (m\phi) \cdot e + u^* - P^m(f_*)u^* + P^m(f_*)\{(n\phi) \cdot e + u^* + e^n\} \\ &= [(m+n)\phi] \cdot e + u^* + P^m(f_*)e^n. \end{aligned}$$

Lemma 7.23

For any $\varepsilon > 0$ there exists an integer n_1 such that for every $n \geq n_1$ and every m, $v^{m+n} - v^{m,n} \leq \varepsilon \cdot e$.

Proof

$$\begin{aligned} v^{m+n} - v^{m,n} &= \{[(m+n)\phi] \cdot e + u^* + e^{n+m}\} - \{[(m+n)\phi] \cdot e + u^* + P^m(f_*)e^n \\ &= e^{n+m} - P^m(f_*)e^n \geq 0, \end{aligned}$$

the last inequality by Lemma 5.8, part (1). As $n \to \infty$, we obtain $e^{m+n} - P^m(f_*)e^n \to d \cdot e - d \cdot e = 0$. So, for any fixed m and any n large enough, we have $v^{m+n} - v^{m,n} \leq \varepsilon \cdot e$. □

7.11 Overtaking, average overtaking and cumulative overtaking optimality

A policy R_* is *overtaking optimal* if $\liminf_{T\to\infty} \{v^T(R_*) - v^T(R)\} \geq 0$ for all policies R.

Example 7.7

$S = \{1,2,3\};\ A(1) = \{1\},\ A(2) = \{1,2\},\ A(3) = \{1\}$.
$r_1(1) = 0,\ r_2(1) = 1;\ r_2(2) = 0,\ r_3(1) = 1$.
$p_{11}(1) = 0,\ p_{12}(1) = 1,\ p_{13}(1) = 0;\ p_{21}(1) = 1,\ p_{22}(1) = 0,\ p_{23}(1) = 0;$
$p_{21}(2) = 0,\ p_{22}(2) = 0,\ p_{23}(2) = 1;\ p_{31}(1) = 0,\ p_{32}(1) = 1,\ p_{33}(1) = 0.$
There are two deterministic stationary policies f_1^∞ with $f_1(2) = 1$ and f_2^∞ with $f_2(2) = 2$.
Observe that:

$$v_1^T(f_1^\infty) = \begin{cases} \frac{1}{2}T - \frac{1}{2} & \text{if } T \text{ is odd} \\ \frac{1}{2}T & \text{if } T \text{ is even} \end{cases};\ v_2^T(f_1^\infty) = \begin{cases} \frac{1}{2}T + \frac{1}{2} & \text{if } T \text{ is odd} \\ \frac{1}{2}T & \text{if } T \text{ is even} \end{cases};\ v_3^T(f_1^\infty) = \begin{cases} \frac{1}{2}T + \frac{1}{2} & \text{if } T \text{ is odd} \\ \frac{1}{2}T + 1 & \text{if } T \text{ is even} \end{cases};$$

$$v_1^T(f_2^\infty) = \begin{cases} \frac{1}{2}T - \frac{1}{2} & \text{if } T \text{ is odd} \\ \frac{1}{2}T - 1 & \text{if } T \text{ is even} \end{cases};\ v_2^T(f_2^\infty) = \begin{cases} \frac{1}{2}T - \frac{1}{2} & \text{if } T \text{ is odd} \\ \frac{1}{2}T & \text{if } T \text{ is even} \end{cases};\ v_3^T(f_2^\infty) = \begin{cases} \frac{1}{2}T + \frac{1}{2} & \text{if } T \text{ is odd} \\ \frac{1}{2}T & \text{if } T \text{ is even} \end{cases}.$$

Hence,

$$v_1^T(f_1^\infty) - v_1^T(f_2^\infty) = \begin{cases} 0 & \text{if } T \text{ is odd} \\ 1 & \text{if } T \text{ is even} \end{cases};\ v_2^T(f_1^\infty) - v_2^T(f_2^\infty) = \begin{cases} 1 & \text{if } T \text{ is odd} \\ 0 & \text{if } T \text{ is even} \end{cases};$$

$$v_3^T(f_1^\infty) - v_3^T(f_2^\infty) = \begin{cases} 0 & \text{if } T \text{ is odd} \\ 1 & \text{if } T \text{ is even} \end{cases}.$$

Therefore, $\liminf_{T\to\infty} \{v_i^T(f_1^\infty) - v_i^T(f_2^\infty)\} = 0,\ i \in S$, and $\limsup_{T\to\infty} \{v_i^T(f_1^\infty) - v_i^T(f_2^\infty)\} = 1,\ i \in S$. So f_1^∞ dominates f_2^∞ in the overtaking optimal sense. in fact, one can show that f_1^∞ is overtaking optimal.

In contrast with other criteria, an overtaking optimal policy doesn't exist in general as the next example shows.

Example 7.8

$S = \{1,2,3\};\ A(1) = \{1,2\},\ A(2) = \{1\},\ A(3) = \{1\}.$
$r_1(1) = 1,\ r_1(2) = 0;\ r_2(1) = 0,\ r_3(1) = 2.$
$p_{11}(1) = 0,\ p_{12}(1) = 1,\ p_{13}(1) = 0;\ p_{11}(2) = 0,\ p_{12}(2) = 0,\ p_{13}(2) = 1;$
$p_{21}(1) = 0,\ p_{22}(1) = 0,\ p_{23}(1) = 1;\ p_{31}(1) = 0,\ p_{32}(1) = 1,\ p_{33}(1) = 0.$
There are two deterministic stationary policies f_1^∞ with $f_1(1) = 1$ and f_2^∞ with $f_2(1) = 2$. Observe that:

$$v_1^T(f_1^\infty) = \begin{cases} T & \text{if } T \text{ is odd} \\ T-1 & \text{if } T \text{ is even} \end{cases};\ v_2^T(f_1^\infty) = \begin{cases} T+1 & \text{if } T \text{ is odd} \\ T & \text{if } T \text{ is even} \end{cases};\ v_3^T(f_1^\infty) = \begin{cases} T+1 & \text{if } T \text{ is odd} \\ T & \text{if } T \text{ is even} \end{cases};$$

$$v_1^T(f_2^\infty) = \begin{cases} T-1 & \text{if } T \text{ is odd} \\ T & \text{if } T \text{ is even} \end{cases};\ v_2^T(f_2^\infty) = \begin{cases} T-1 & \text{if } T \text{ is odd} \\ T & \text{if } T \text{ is even} \end{cases};\ v_3^T(f_2^\infty) = \begin{cases} T+1 & \text{if } T \text{ is odd} \\ T & \text{if } T \text{ is even} \end{cases}.$$

Hence,

$$v_1^T(f_1^\infty) - v_1^T(f_2^\infty) = \begin{cases} 1 & \text{if } T \text{ is odd} \\ -1 & \text{if } T \text{ is even} \end{cases};\ v_2^T(f_1^\infty) - v_2^T(f_2^\infty) = v_3^T(f_1^\infty) - v_3^T(f_2^\infty) = 0 \text{ for all } T.$$

Therefore, $\liminf_{T\to\infty} \{v_1^T(f_1^\infty) - v_1^T(f_2^\infty)\} = -1$ and $\liminf_{T\to\infty} \{v_1^T(f_2^\infty) - v_1^T(f_1^\infty)\} = -1$.
Hence neither f_1^∞ dominates f_2^∞ nor f_2^∞ dominates f_1^∞ in the overtaking optimal sense. in fact, for this model there exists no overtaking optimal policy.

One might suspect that periodic chains are the sole cause of the nonexistence of an overtaking optimal policy. But Brown ([34]) provided an example that has no periodic chains, but nevertheless no overtaking optimal policy.

Example 7.8 shows that the criterion of overtaking optimality is overselective. Therefore, we consider the following less selective criterion. A policy R_* is called *average overtaking optimal* if

$$\liminf_{T\to\infty} \frac{1}{T} \sum_{t=1}^T \{v^t(R_*) - v^t(R)\} \geq 0 \text{ for all policies } R.$$

Notice that this criterion is the same as 0-average optimality which is also equivalent to bias optimality. It is easy to verify that in Example 7.8 both f_1^∞ and f_2^∞ are average overtaking optimal policies.

We also introduce another criterion which is less selective than overtaking optimality, but more selective than average overtaking optimality. A policy R_* is called *cumulative overtaking optimal* if

$$\liminf_{T\to\infty} \sum_{t=1}^T \{v^t(R_*) - v^t(R)\} \geq 0 \text{ for all policies } R.$$

Notice that if R_* is cumulative overtaking optimal it is also an average overtaking optimal policy, so that the criterion of cumulative overtaking optimality is more selective that the criterion of average overtaking optimality. It is easy to verify that in Example 7.8 f_1^∞ dominates f_2^∞ in the cumulative overtaking optimal sense, namely: $\sum_{t=1}^T \{v^t(f_1^\infty) - v^t(f_2^\infty)\} = \begin{cases} 1 & \text{if } T \text{ is odd;} \\ 0 & \text{if } T \text{ is even.} \end{cases}$

7.12 A weighted combination of discounted and average rewards

The two most commonly considered reward criteria for MPDs are the total discounted rewards and the long-run average rewards. The first tends to neglect the far future, while the second one tends to neglect the short-term rewards. In this section we consider a new optimality criterion consisting of a weighted combination of these two criteria. Example 7.9 (see below) motivates this choice. For the sensitive criteria, discussed so far in this chapter, priority is given to finding an average optimal policy, using additional criteria for further selection within the class of average optimal policies. An implication of the weighted combination criterion will be that an optimal policy might not exist, even when we allow nonstationary randomized policies. We present an iterative algorithm for computing an ε-optimal nonstationary policy with a simple structure.

Example 7.9

Suppose that the owner of a business is currently using a very dependable supplier of a rare material which is essential for his business. However, a new supplier proposes to meet his demand at a much lower price, but is less dependable. The owner of the business estimates that with probability 0.1 the new supplier may fail to deliver the raw material in any given year, and that such a failure would result in bankruptcy. Assuming that the annual profits with the old and new supplier will be 100.000 and 154.000 euro, respectively, that the discount factor $\alpha = 0.8$, and that each year a new contract has to be signed with the supplier.

Consider, for the decision problem whether or not to switch to the new supplier, the following MDP model. Introduce two states: state 1 for business operation and state 2 for bankruptcy. In state 1 there are two actions: $a = 1$ for switching to the new supplier and $a = 2$ for keeping the old supplier; state 2 is an absorbing state. Hence the transition probabilities are: $p_{11}(1) = 0.9, p_{12}(1) = 0.1$; $p_{11}(2) = 1$, $p_{12}(2) = 0$; $p_{21}(1) = 0$, $p_{22}(1) = 1$. The immediate rewards are: $r_1(1) = 154.000$; $r_1(2) = 100.000$; $r_2(1) = 0$.

There are two deterministic stationary policies: f_1^∞ and f_2^∞ corresponding to signing a contract with the new and old supplier, respectively. It is easy to see that the total expected discounted rewards are: $v^\alpha(f_1^\infty) = 550.000$ and $v^\alpha(f_2^\infty) = 500.000$. Hence, the policy f_1^∞ is α-discounted optimal. However, the use of this policy results finally in bankruptcy and the expected average reward $\phi(f_1^\infty) = 0$. Note that $\phi(f_2^\infty) = 100.000$.

Consider for some $k \geq 1$ a policy R_k of the form $R_k = (g_1, g_2, \ldots, g_k, h_1, h_2, \ldots)$, where $g_j := f_1$ for $j = 1, 2, \ldots, k$ and $h_j := f_2$ for $j = 1, 2, \ldots$. Then, it is easy to verify that $v^\alpha(R_k) = 550 - 0.72^k \cdot 50$, which is between 500.000 and 550.000. The probability of bankruptcy is $1 - (0.9)^k$. Hence, policies of this form will enable the owner to balance the increased profits of the new supplier against the risk of bankruptcy. Policies of the type R_k are more sensitive to the future then the discounted optimal policy f_1^∞.

For any policy R and initial state $i \in S$, the *weighted reward* $w_i[\lambda_1, \lambda_2](R)$ is defined by:

$$w_i[\lambda_1, \lambda_2](R) := \lambda_1 \cdot (1 - \alpha) v_i^\alpha(R) + \lambda_2 \cdot \phi_i(R), \tag{7.55}$$

where $\lambda_1, \lambda_2 \in [0, 1]$ are fixed parameters and $\alpha \in [0, 1)$ is the discount factor.

1. Note that for $\lambda_1 = 1$ and $\lambda_2 = 0$ this criterion is the total discounted reward; for $\lambda_1 = 0$ and $\lambda_2 = 1$ this criterion is the average reward.
2. A policy R^* that optimizes $w_i[\lambda_1, \lambda_2](R)$ is a *Pareto optimal policy* in a bi-objective optimization problem in which the decision maker wishes to 'simultaneously optimize' both the discounted and the average reward criterion.
3. The results of Derman and Strauch ([71]) imply that $\sup_C w_i[\lambda_1, \lambda_2](R) = \sup_{C(M)} w_i[\lambda_1, \lambda_2](R)$ for all $i \in S$, so we only need to consider Markov policies. It is obvious that the following inequalities hold:

$$\sup_{R \in C(D)} w_i[\lambda_1, \lambda_2](R) \leq \sup_{R \in C(S)} w_i[\lambda_1, \lambda_2](R) \leq \sup_{R \in C(M)} w_i[\lambda_1, \lambda_2](R) \leq \lambda_1 \cdot (1-\alpha) v_i^\alpha + \lambda_2 \cdot \phi_i, \ i \in S. \tag{7.56}$$

The upper bound on the right-hand side of (7.56) will be referred to as the *utopian bound*.

4. Note that in the discounted MDP there always exists an $\alpha_0 \in [0,1)$ and a deterministic policy f_0^∞ such that f_0^∞ is optimal for both the discounted and average reward criterion (such policy is also called a Blackwell optimal policy). Such a policy clearly attains the utopian bound and is therefore optimal for the weighted reward criterion if the discount factor $\alpha \geq \alpha_0$.

The next example will demonstrate that the first two inequalities in (7.56) can be strict. The example also shows that optimal policies do not generally exist for the weighted optimality criterion.

Example 7.10

Consider the MDP with $S = \{1,2\}$; $A(1) = \{1,2\}$, $A(2) = \{1\}$; $p_{11}(1) = 1, p_{12}(1) = 0$; $p_{11}(2) = 0$, $p_{12}(2) = 1$; $p_{21}(1) = 1, p_{22}(1) = 0$; $r_1(1) = 0$; $r_1(2) = -10$; $r2(1) = 12; \alpha = \frac{1}{2}$; $\lambda_1 = \frac{1}{4}$ and $\lambda_2 = \frac{3}{4}$.
We shall write $w(R)$ instead of $w[\lambda_1, \lambda_2](R)$.
There are two deterministic stationary policies: f_1^∞ and f_1^∞ corresponding to action 1 and 2 in state 1, respectively. It is easy to see that $v_1^\alpha(f_1^\infty) = 0$, $v_2^\alpha(f_1^\infty) = 12$, $\phi_1(f_1^\infty) = \phi_2(f_1^\infty) = 0$ and $v_1^\alpha(f_2^\infty) = -\frac{16}{3}$, $v_2^\alpha(f_2^\infty) = \frac{28}{3}$, $\phi_1(f_2^\infty) = \phi_2(f_2^\infty) = 1$. Hence, $w_1(f_1^\infty) = 0$, $w_2(f_1^\infty) = \frac{3}{2}$, $w_1(f_2^\infty) = \frac{1}{12}$ and $w_2(f_2^\infty) = \frac{23}{12}$.
Notice that the utopian bound in state 1 equals $\frac{3}{4}$ and in state 2 this bound is $\frac{9}{4}$.
Clearly, f_1^∞ is discounted optimal and f_2^∞ is average optimal. Furthermore, f_2^∞ is optimal for the weighted optimality criterion, and $sup_{C(D)}\, w_1(f^\infty) = \frac{1}{12}$ and $sup_{C(D)}\, w_2(f^\infty) = \frac{23}{12}$.
Let π^∞ be a randomized stationary policy which chooses action 1 in state 1 with probability $p \in [0,1]$. Then, it is straightforward to show that $v_1^\alpha(\pi^\infty) = -\frac{16(1-p)}{3-p}$, $v_2^\alpha(\pi^\infty) = \frac{4(7-p)}{3-p}$, $\phi_1(\pi^\infty) = \frac{2(1-p)}{2-p}$ and $\phi_2(\pi^\infty) = \frac{2(1-p)}{2-p}$. Therefore, $w_1(\pi^\infty) = \frac{1-p^2}{2(2-p)(3-p)}$ and $w_2(\pi^\infty) = \frac{23-21p+4p^2}{2(2-p)(3-p)}$.
Define the function $g(p)$ by $g(p) := \frac{1-p^2}{2(2-p)(3-p)}$. Then the best randomized stationary π^∞ in state 1 for the weighted combination of discounted and average reward is obtained by maximizing $g(p)$ over $p \in [0,1]$. Since $g'(p) = \frac{5-14p+5p^2}{2(2-p)(3-p)}$, we have $g'(p) = 0$ for $p = \frac{7-2\sqrt{6}}{5} \approx 0.42$. Hence, we obtain $sup_{C(S)}\, w_1(\pi^\infty) \approx g(0.42) \approx 0.10 > 1/12 = sup_{C(D)}\, w_1(f^\infty)$, we have shown that the first inequality of (7.56) is strict in this example.
Next, consider the Markov policy R_k, which uses f_1 for the first k stages, and then switches to f_2 permanently. Then, it is easy to see that $v_1^\alpha(R_k) = (\frac{1}{2})^k \cdot \frac{-16}{3}$ and $\phi(R_k) = 1$. Therefore, $w_1^\alpha(R_k) = (\frac{1}{2})^k \cdot \frac{-2}{3} + \frac{3}{4}$. Hence, R_k approaches the utopian bound $\frac{3}{4}$ arbitrarily close as $k \to \infty$, but never reaches it. This immediately implies that the second inequality of (7.56) is also strict.
We now show that an optimal policy does not exist. Suppose that $R^* = (\pi^1, \pi^2, \dots)$ is an optimal policy. Without loss of generality we may assume that R^* is a Markov policy. We must have $w_1(R^*) = \frac{3}{4}$, because R^* is as least so good as any policy R_k which uses f_1 for the first k stages, and then switches to f_2 permanently. Clearly $R^* \neq f_1^\infty$, because $w_1(f_1^\infty) = 0$. Therefore, there exists some k such that π^k is the first decision rule of R^* which chooses action 2 in state 1 with a positive probability, say with probability $p > 0$. The discounted rewards of R^* can be bounded as follows: the payoff for the first $k-1$ stages is 0, the payoff at stage k is $(\frac{1}{2})^{k-1} \cdot (-10p)$, the payoff at stage $k+1$ is at most $(\frac{1}{2})^k \cdot 12p$ and the payoff for all stages after $k+2$ is at most $(\frac{1}{2})^{k+1} \cdot v_1^\alpha = 0$. Therefore, $v_1^\alpha(R^*) \leq (\frac{1}{2})^{k-1} \cdot (-10p) + (\frac{1}{2})^k \cdot 12p = (\frac{1}{2})^k \cdot (-8p) < 0$. Because $\phi_1(R^*) \leq \phi_1 = 1$, we obtain $w_1(R^*) = \frac{1}{8} v_1^\alpha(R^*) + \frac{3}{4}\phi_1(R^*) < \frac{3}{4}$, which contradicts the optimality of R^*.

We shall say that a Markov policy $R = (\pi_1, \pi_2, \dots)$ is *ultimately deterministic* if there exists a positive integer t_0 and a policy $f^\infty \in C(D)$ such that $\pi^t = f$ for all $t \geq t_0$. Let $C(UD)$ denote the set of ultimately deterministic policies. In the next theorem we consider the case that the value vector ϕ of the average rewards is constant, i.e. independent of the starting state. This is for instance the case in unichain and communicating MDPs. The theorem shows that for MDPs with ϕ independent of the starting state the utopian bound is achieved and that ε-optimal ultimately deterministic policies with a simple structure can be constructed easily from two deterministic policies, namely from the discounted and the average optimal policy.

Theorem 7.19

Suppose that the value vector ϕ of the average reward is constant, say $\phi_i = c$ for all $i \in S$. Then,

(1) Let g^∞ be a discounted optimal policy and h^∞ be an average optimal policy.
Let $\varepsilon > 0$ be given and define $M := max_{(i,a)}\, r_i(a)$ and $m := min_{(i,a)}\, r_i(a)$.
Take $R_\varepsilon = (f_1, f_2, \dots)$ with $f_t := \begin{cases} g & \textit{for } t < t(\varepsilon) \\ h & \textit{for } t \geq t(\varepsilon) \end{cases}$ where $t(\varepsilon) := \lfloor \frac{\log\,[\varepsilon/(M-m)]}{\log\,\alpha} \rfloor + 1$.
Then, R_ε is a $(\lambda_1 \varepsilon)$-optimal policy.

(2) $sup_{R \in C(M)}\, w_i[\lambda_1, \lambda_2](R) = sup_{R \in C(UD)}\, w_i[\lambda_1, \lambda_2](R) = \lambda_1 \cdot (1-\alpha) v_i^\alpha + \lambda_2 \cdot \phi_i$.

Proof

Part 1:

Note that for any positive integer t_0, any policy R and any starting state i, we have

$\sum_{t=t_0+1}^\infty \alpha^{t-1}\, \mathbb{E}_{i,R}\, \{r_{X_t}, Y_t\} \leq \alpha^{t_0} \cdot \frac{M}{1-\alpha}$ and $\sum_{t=t_0+1}^\infty \alpha^{t-1}\, \mathbb{E}_{i,R}\, \{r_{X_t}, Y_t\} \geq \alpha^{t_0} \cdot \frac{m}{1-\alpha}$.

To assure that $\alpha^{t_0} \cdot \frac{M-m}{1-\alpha} < \frac{\varepsilon}{1-\alpha}$ it suffices to take $t_0 > \frac{\log\,[\varepsilon/(M-m)]}{\log\,\alpha}$. Thus, $t(\varepsilon)$ satisfies this inequality. Now, we can write for any $i \in S$,

$$\begin{array}{rcl} v_i^\alpha - v_i^\alpha(R_\varepsilon) & = & \sum_{t=1}^\infty \alpha^{t-1}\, \mathbb{E}_{i,g^\infty}\, \{r_{X_t}, Y_t\} - \sum_{t=1}^\infty \alpha^{t-1}\, \mathbb{E}_{i,R_\varepsilon}\, \{r_{X_t}, Y_t\} \\ & = & \sum_{t=t_0+1}^\infty \alpha^{t-1}\, \mathbb{E}_{i,g^\infty}\, \{r_{X_t}, Y_t\} - \sum_{t=t_0+1}^\infty \alpha^{t-1}\, \mathbb{E}_{i,R_\varepsilon}\, \{r_{X_t}, Y_t\} \\ & \leq & \alpha^{t_0} \cdot \frac{M-m}{1-\alpha} < \frac{\varepsilon}{1-\alpha}. \end{array}$$

Furthermore, we have $c = \phi_i = \phi_i(h^\infty) = \phi_i(R_\varepsilon)$, because rewards over the first t_0 stages do not influence the average reward over an infinite horizon and since the value vector in independent of the initial state i. Therefore, we obtain

$$w_i[\lambda_1, \lambda_2](R_\varepsilon) = \lambda_1 \cdot (1-\alpha) v_i^\alpha(R_\varepsilon) + \lambda_2 \cdot \phi_i(R_\varepsilon) > \lambda_1 \cdot (1-\alpha) v_i^\alpha + \lambda_2 \cdot \phi_i - (\lambda_1 \varepsilon) \text{ for all } i \in S,$$

showing that R_ε is a $(\lambda_1 \varepsilon)$-optimal policy.

Part 2:

By the existence of a $(\lambda_1 \varepsilon)$-optimal policy for any $\varepsilon > 0$, we can write for all $i \in S$

$$sup_{R \in C(M)}\, w_i[\lambda_1, \lambda_2](R) = sup_{R \in C(UD)}\, w_i[\lambda_1, \lambda_2](R) = \lambda_1 \cdot (1-\alpha) v_i^\alpha + \lambda_2 \cdot \phi_i. \qquad \square$$

We now turn our attention back to the general case, that is, the average reward is allowed to depend on the starting state. We first need the following lemma.

Lemma 7.24

Let $R = (\pi^1, \pi^2, \dots)$ be an arbitrary Markov policy and let $h^\infty \in C(D)$ be an average optimal policy. Take some $k \in \mathbb{N}$ and define the policy $R_k = (\rho_1, \rho_2, \dots)$ by $\rho^t := \begin{cases} \pi^t & \textit{if } t \leq k-1 \\ h & \textit{if } t \geq k \end{cases}$.
Then, $\phi_i(R) \leq \phi_i(R_k)$ for all $i \in S$.

Proof

Fix some positive integer k and some $i \in S$. Then, we can write

$$\phi_i(R) = \liminf_{T\to\infty} \tfrac{1}{T}\textstyle\sum_{t=1}^T \mathbb{E}_{i,R}\left\{r_{X_t}(Y_t)\right\} = \liminf_{T\to\infty} \tfrac{1}{T}\textstyle\sum_{t=k}^T \mathbb{E}_{i,R}\left\{r_{X_t}(Y_t)\right\},$$

the last equality because rewards over the first $k-1$ periods do not influence the average reward over an infinite horizon. Define $\hat{R} = (\hat{\pi}^1, \hat{\pi}^2, \dots)$ by $\hat{\pi}^1 := \pi^k,\ \hat{\pi}^2 := \pi^{k+1}, \dots$. Then, for any $t \geq k$, we have for all $(j,a) \in S \times A$,

$$\begin{array}{lcl} \mathbb{P}_{i,R}\{X_t = j, Y_t = a\} & = & \sum_l \mathbb{P}_{i,R}\{X_k = l\} \cdot \mathbb{P}_R\{X_t = j, Y_t = a \mid X_k = l\} \\ & = & \sum_l \mathbb{P}_{i,R_k}\{X_k = l\} \cdot \mathbb{P}_R\{X_t = j, Y_t = a \mid X_k = l\} \\ & = & \sum_l \mathbb{P}_{i,R_k}\{X_k = l\} \cdot \mathbb{P}_{\hat{R}}\{X_{t-k+1} = j, Y_{t-k+1} = a \mid X_1 = l\} \end{array},$$

the first equality by conditioning over the state at stage k, the second equality since R and R_k coincide up to time k and the last equality follows from the definition of $\hat{R}$. Hence,

$$\begin{array}{lcl} \mathbb{E}_{i,R}\{r_{X_t}(Y_t)\} & = & \sum_{(j,a)} \mathbb{P}_{i,R}\{X_t = j, Y_t = a\} \cdot r_j(a) \\ & = & \sum_l \mathbb{P}_{i,R_k}\{X_k = l\} \cdot \mathbb{E}_{l,\hat{R}}\{r_{X_{t-k+1}}(Y_{t-k+1})\}. \end{array},$$

Now, we obtain

$$\begin{array}{lcl} \phi_i(R) & = & \liminf_{T\to\infty} \frac{1}{T}\sum_{t=k}^T \mathbb{E}_{i,R}\left\{r_{X_t}(Y_t)\right\} \\ & = & \liminf_{T\to\infty} \frac{1}{T}\sum_{t=k}^T \sum_l \mathbb{P}_{i,R_k}\{X_k = l\} \cdot \mathbb{E}_{l,\hat{R}}\{r_{X_{t-k+1}}(Y_{t-k+1})\} \\ & \leq & \limsup_{T\to\infty} \frac{1}{T}\sum_{t=k}^T \sum_l \mathbb{P}_{i,R_k}\{X_k = l\} \cdot \mathbb{E}_{l,\hat{R}}\{r_{X_{t-k+1}}(Y_{t-k+1})\} \\ & \leq & \sum_l \mathbb{P}_{i,R_k}\{X_k = l\} \cdot \limsup_{T\to\infty} \frac{1}{T}\sum_{t=k}^T \cdot \mathbb{E}_{l,\hat{R}}\{r_{X_{t-k+1}}(Y_{t-k+1})\} \\ & = & \sum_l \mathbb{P}_{i,R_k}\{X_k = l\} \cdot \overline{\phi}_i(\hat{R}), \end{array}.$$

where $\overline{\phi}(\hat{R})$ is defined in Section 1.2.2. It can be shown (see p. 101 in [148]) that the average optimal stationary policy h^∞ is also optimal with respect to the criterion $sup_R\, \overline{\phi}(R)$. Therefore, we have

$$\begin{array}{lcl} \phi_i(R) & \leq & \sum_l \mathbb{P}_{i,R_k}\{X_k = l\} \cdot \overline{\phi}_l(h^\infty) \\ & = & \sum_l \mathbb{P}_{i,R_k}\{X_k = l\} \cdot \lim_{T\to\infty} \frac{1}{T}\sum_{t=1}^T \mathbb{E}_{l,h^\infty}\{r_{X_t}(Y_t)\} \\ & = & \sum_l \mathbb{P}_{i,R_k}\{X_k = l\} \cdot \lim_{T\to\infty} \frac{1}{T}\sum_{t=1}^T \mathbb{E}_{h^\infty}\{r_{X_{t+k-1}}(Y_{t+k-1}) \mid X_k = l\}. \end{array}.$$

Since the policies h^∞ and R_k have the same decision rules from stage k, we also have

$$\mathbb{E}_{h^\infty}\{r_{X_{t+k-1}}(Y_{t+k-1}) \mid X_k = l\} = \mathbb{E}_{R_k}\{r_{X_{t+k-1}}(Y_{t+k-1}) \mid X_k = l\} \text{ for all } t \geq 1.$$

Consequently, we can write

$$\begin{array}{lcl} \phi_i(R) & \leq & \sum_l \mathbb{P}_{i,R_k}\{X_k = l\} \cdot \lim_{T\to\infty} \frac{1}{T}\sum_{t=1}^T \mathbb{E}_{R_k}\{r_{X_{t+k-1}}(Y_{t+k-1}) \mid X_k = l\} \\ & = & \lim_{T\to\infty} \frac{1}{T}\sum_{t=1}^T \sum_l \mathbb{P}_{i,R_k}\{X_k = l\} \cdot \mathbb{E}_{R_k}\{r_{X_{t+k-1}}(Y_{t+k-1}) \mid X_k = l\} \\ & = & \phi_i(R_k), \end{array}.$$

the last inequality by conditioning over the state at stage k and the property that rewards during the first $k-1$ stages have no effect on the average reward over an infinite horizon. □

In the remaining part of this section we consider the weighted optimality criterion. Assume that are given: $i \in S$ as fixed initial state and ε as a tolerance. A policy R^* is called (i,ε)-optimal if $w_i[\lambda_1, \lambda_2](R^*) \geq sup_R\, w_i[\lambda_1, \lambda_2](R) - \varepsilon$.

Theorem 7.20

Let $h^\infty \in C(D)$ be an average optimal policy. Then, there exists a positive integer $t(\varepsilon)$ and a policy $R_\varepsilon = (\pi^1, \pi^2, \dots) \in C(UD)$ with $\pi^t = h$ for all $t \geq t(\varepsilon)$ such that R_ε is (i,ε)-optimal.

Proof

By the previous remark 3, there exists a Markov policy $\overline{R} = (\overline{\pi}^1, \overline{\pi}^2, \dots)$ which is $(i, \varepsilon/2)$-optimal. Define $R_\varepsilon = (\pi^1, \pi^2, \dots)$ by $\pi^t := \begin{cases} \overline{\pi}^t & \text{for } t \leq t(\varepsilon) - 1 \\ h & \text{for } t \geq t(\varepsilon) \end{cases}$ for some positive integer $t(\varepsilon)$.

From the first lines in the proof of Theorem 7.19 it follows that $t(\varepsilon)$ can be chosen such that

$$v_i^\alpha(R_\varepsilon) \geq v_i^\alpha - \tfrac{\varepsilon}{2\lambda_1(1-\alpha)} \geq v_i^\alpha(\overline{R}) - \tfrac{\varepsilon}{2\lambda_1(1-\alpha)}, \text{ i.e. } \lambda_1(1-\alpha)v_i^\alpha(R_\varepsilon) \geq \lambda_1(1-\alpha)v_i^\alpha(\overline{R}) - \varepsilon/2.$$

By the construction of policy R_ε and Lemma 7.24 it follows that $\phi_i(R_\varepsilon) \geq \phi_i(\overline{R})$. Hence,

$$\begin{aligned} w_i[\lambda_1, \lambda_2](R_\varepsilon) &= \lambda_1(1-\alpha)v_i^\alpha(R_\varepsilon) + \lambda_2\phi_i(R_\varepsilon) \\ &\geq \lambda_1(1-\alpha)v_i^\alpha(\overline{R}) - \varepsilon/2 + \lambda_2\phi_i(\overline{R}) \\ &= w_i[\lambda_1, \lambda_2](\overline{R}) - \varepsilon/2 \\ &\geq sup_R\, w_i[\lambda_1, \lambda_2](R) - \varepsilon. \end{aligned}$$

□

In the remainder of this section we shall construct an algorithm to compute an ε-optimal policy for the weighted combination of discounted and average rewards. To that end, we shall use Theorem 7.20 which guarantees the existence of (i, ε)-optimal ultimately deterministic policies whose 'tail' consists of an average optimal deterministic policy.

Lemma 7.25

Let $R_1 = (\pi^1, \pi^2, \pi^3, \dots)$ any ultimately deterministic policy and let $R_2 := (, \pi^2, \pi^3, \pi^4, \dots)$. Then, $w_i[\lambda_1, \lambda_2](R_1) = \lambda_1(1-\alpha)\sum_a r_i(a)\pi_{ia}^1 + \sum_a \sum_j p_{ij}(a)\pi_{ia}^1 w_j[\alpha\lambda_1, \lambda_2](R_2)$ for all $i \in S$.

Proof

By the definition of R_2 and the property that $R_1 \in C(UD)$, we obtain for any $i \in S$,

$$v_i^\alpha(R_1) = \textstyle\sum_a r_i(a)\pi_{ia}^1 + \alpha\sum_a \sum_j p_{ij}(a)\pi_{ia}^1 v_j^\alpha(R_2) \text{ and } \phi_i(R_1) = \sum_a p_{ij}(a)\pi_{ia}^1\phi_j(R_2).$$

Hence,

$$\begin{aligned} w_i[\lambda_1, \lambda_2](R_1 &= \lambda_1(1-\alpha)v_i^\alpha(R_1) + \lambda_2\phi_i(R_1) \\ &= \lambda_1(1-\alpha)\{\textstyle\sum_a r_i(a)\pi_{ia}^1 + \alpha\sum_a\sum_j p_{ij}(a)\pi_{ia}^1 v_j^\alpha(R_2)\} + \lambda_2\sum_a p_{ij}(a)\phi_j(R_2) \\ &= \lambda_1(1-\alpha)\textstyle\sum_a r_i(a)\pi_{ia}^1 + \sum_a\sum_j p_{ij}(a)\pi_{ia}^1\{\alpha\lambda_1(1-\alpha)v_j^\alpha(R_2) + \lambda_2\phi_j(R_2) \\ &= \lambda_1(1-\alpha)\textstyle\sum_a r_i(a)\pi_{ia}^1 + \sum_a\sum_j p_{ij}(a)\pi_{ia}^1 w_j[\alpha\lambda_1, \lambda_2](R_2). \end{aligned}$$

□

Lemma 7.26

Let $R_1 \in C(UD)$ be an ε-optimal policy with respect to $w[\alpha\lambda_1, \lambda_2](R)$. For each $i \in S$, let $a_i \in A(i)$ be the action that achieves $max_a\{\lambda_1(1-\alpha)r_i(a) + \sum_j p_{ij}(a)w_j[\alpha\lambda_1, \lambda_2](R_1)\}$. Let f_1 be the deterministic decision rule which uses action a_i in state i and let $R_2 := (f_1, R_1)$. Then, $R_2 \in C(UD)$ and is ε-optimal with respect to $w[\lambda_1, \lambda_2](R)$.

Proof

Take any $i \in S$ and some $\delta > 0$. Let $R_3 = (\pi_1, \pi_2, \dots) \in C(UD)$ be an (i, δ)-optimal policy with respect to $w[\lambda_1, \lambda_2](R)$ (such policy exists by Theorem 7.20). Define R_4 by $R_4 = (\pi_2, \pi^3, \dots)$. Then, we can write by Lemma 7.25, the definition of a_i and the ε-optimality of R_1 with respect to $w[\alpha\lambda_1, \lambda_2](R)$,

$$\begin{aligned} w_i[\lambda_1, \lambda_2](R_2) &= \lambda_1(1-\alpha)r_i(a_i) + \textstyle\sum_j p_{ij}(a_i)w_j[\alpha\lambda_1, \lambda_2](R_1) \\ &= max_a\{\lambda_1(1-\alpha)r_i(a) + \textstyle\sum_j p_{ij}(a)w_j[\alpha\lambda_1, \lambda_2](R_1)\} \\ &\geq max_a\big\{\lambda_1(1-\alpha)r_i(a) + \textstyle\sum_j p_{ij}(a)\{w_j[\alpha\lambda_1, \lambda_2](R_4) - \varepsilon\}\big\} \\ &= max_a\{\lambda_1(1-\alpha)r_i(a) + \textstyle\sum_j p_{ij}(a)w_j[\alpha\lambda_1, \lambda_2](R_4)\} - \varepsilon \\ &\geq \textstyle\sum_a \pi_{ia}^1\{\lambda_1(1-\alpha)r_i(a) + \sum_j p_{ij}(a)w_j[\alpha\lambda_1, \lambda_2](R_4)\} - \varepsilon \\ &= w_i[\lambda_1, \lambda_2](R_3) - \varepsilon. \end{aligned}$$

Now, let R be an arbitrary policy. Using the above inequality and the (i, δ)-optimality of policy R_3 with respect to $w_i[\lambda_1, \lambda_2](R)$, we get

$$w_i[\lambda_1, \lambda_2](R_2) \geq w_i[\lambda_1, \lambda_2](R_3) - \varepsilon \geq w_i[\lambda_1, \lambda_2](R)\delta - \varepsilon.$$

Since δ can be chosen arbitrarily small, and R and i are arbitrary chosen, we have shown that R_2 is ε-optimal with respect to $w[\lambda_1, \lambda_2](R)$. It is obvious that $R_2 \in C(UD)$.

Remarks

1. For any $t \geq 0$ and any policy R, $w_i[\alpha^t\lambda_1, \alpha_2](R) = \alpha^t\lambda_1(1-\alpha)v_i^\alpha(R) + \lambda_2\phi_i(R) \leq \alpha^t\lambda_1(1-\alpha)v_i^\alpha + \lambda_2\phi_i$ for all $i \in S$. Consequently, $sup_R\, w_i[\alpha^t\lambda_1, \alpha_2](R) \leq \alpha^t\lambda_1(1-\alpha)v_i^\alpha + \lambda_2\phi_i,\ i \in S$.
2. For any $\varepsilon > 0$ and any $f^\infty \in C(D)$, there exists a $T \in \mathbb{N}$ such that $\alpha^T\lambda_1(1-\alpha)\{v_i^\alpha - v_i^\alpha(f^\infty)\} \leq \varepsilon$ for all $i \in S$.
3. Let $f^\infty \in C(D)$ be an average optimal policy. Then, by the remarks 1 and 2, there exists a $T \in \mathbb{N}$ such that for all $i \in S$,
$$\begin{array}{rcl} sup_R\, w_i[\alpha^T\lambda_1, \lambda_2](R) & \leq & \alpha^T\lambda_1(1-\alpha)v_i^\alpha + \lambda_2\phi_i \\ & \leq & \alpha^T\lambda_1(1-\alpha)v_i^\alpha(f^\infty) + \varepsilon + \lambda_2\phi_i(f^\infty) \\ & = & w_i[\alpha^T\lambda_1, \lambda_2](f^\infty) + \varepsilon + \lambda_2\phi_i(f^\infty), \end{array}$$
i.e. f^∞ is an ε-optimal policy with respect to $w[\alpha^T\lambda_1, \lambda_2](R)$.
4. By remark 3, an ultimately stationary ε-optimal policy with respect to $w[\alpha^T\lambda_1, \lambda_2](R)$ exists. Then, repeated application of Lemma 7.26 assures the existence of an ε-optimal ultimately deterministic policy with respect to $w[\lambda_1, \lambda_2](R)$. This is executed in the next algorithm.

Algorithm 7.6 *Determination of an ε-optimal ultimately deterministic policy with respect to $w[\lambda_1, \lambda_2](R)$*
Input: Instance of an MDP, $\lambda_1 \in [0, 1],\ \lambda_2 \in [0, 1]$ and $\alpha \in [0, 1)$.
Output: An ε-optimal ultimately deterministic policy with respect to $w[\lambda_1, \lambda_2](R)$.

```
1. Compute the $\alpha$-discounted value vector $v^\alpha$, the average value vector $\phi$ and an average optimal policy
   $f^\infty \in C(D)$.

2. Determine the smallest positive integer $T$ such that $\alpha^T\lambda_1(1-\alpha)\{v_i^\alpha - v_i^\alpha(f^\infty)\} \leq \varepsilon$ for all $i \in S$.

3. $R := f^\infty$.

4. for $t = T$ step $-1$ until $1$ do
   begin
       for all $i \in S$ do $w_i[\alpha^t\lambda_1, \lambda_2](R) := \alpha^t\lambda_1(1-\alpha)v^\alpha(R) + \lambda_2\phi(R)$;
       for all $i \in S$ do
           select $a_i \in A(i)$ such that $a_i$ achieves $max_a\, \{\alpha^{t-1}\lambda_1(1-\alpha)r_i(a) + \sum_j p_{ij}(a)w_j[\alpha^t\lambda_1, \lambda_2](R)\}$;
       for all $i \in S$ do $f_t(i) := a_i$;
       $R := (f_t, R)$
   end

5. $R$ is an $\varepsilon$-optimal policy with respect to $w[\lambda_1, \lambda_2](R)$ (STOP).
```

Theorem 7.21
Algorithm 7.6 is correct, i.e. policy $R = (f_1, f_2, \dots, f_T, f, f, \dots)$ is an ε-optimal policy with respect to $w[\lambda_1, \lambda_2](R)$.

Proof
By remark 3, f^∞ is an ε-optimal policy with respect to $w[\alpha^T\lambda_1, \lambda_2](R)$. Then, by Lemma 7.26, the policy $(f_1, f, f, \dots)$ is a ε-optimal policy with respect to $w[\alpha^{T-1}\lambda_1, \lambda_2](R)$. By repeated application (T times) of this argument, we obtain that $R := (f_1, f_2, \dots, f_T, f, f, \dots)$ is an ε-optimal policy with respect to $w[\lambda_1, \lambda_2](R)$. □

7.13 A sum of discount factors

The use of the discounted reward criterion is consistent with the notion that what happens far in the future is unimportant. It arises through the notion that immediate rewards are better than delayed rewards. Discount factors depend on perceived investment opportunities. When there are several different investment opportunities then it is natural to consider the sum of several expected total discounted rewards with different discount factors. Such criteria arise in models of investment with different risk classes. Two cash flow streams with different risks would have different discount factors and the value of the portfolio is the sum of the discounted values of each cash flow in the portfolio.

As in the previous section, for this model there may not exist a stationary optimal or ε-optimal policy. However, we can prove the existence of an optimal ultimately deterministic policy. The algorithm is of the same level of complexity as the computation of optimal policies for the standard reward problem.

In this model we assume that there are several one-step rewards and discount factors, say rewards $r_i^k(a)$, $(i,a) \in S \times A$ and discount factors α_k for $k = 1, 2, \dots, K$. The total expected discounted reward for income stream k given initial state i and policy R is given by

$$v_i^k(R) := \sum_{t=1}^{\infty} (\alpha_k)^{t-1} \sum_{j,a} \mathbb{P}_{i,R}\{X_t = j, Y_t = a\} \cdot r_j^k(a). \tag{7.57}$$

The *sum of discounted rewards* when the initial state is i and policy R is used, is defined by

$$w_i(R) := \sum_{k=1}^{K} v_i^k(R). \tag{7.58}$$

The *value vector* w of this model is defined by $w_i := sup_R\, w_i(R),\ i \in S$. For $\varepsilon > 0$, a policy R_* is called ε-optimal if $w_i(R_*) \ge w_i - \varepsilon$ for all $i \in S$.

Example 7.11
In this example we show that for $\varepsilon > 0$ small enough, there exists no ε-optimal stationary policy. Moreover, we show that the best policy in the class $C(S)$ is not deterministic, but randomized.
Consider a usual MDP with $S = \{1,2\}; A(1) = \{1,2\}, A(2) = \{1\};\ p_{11}(1) = 1, p_{12}(1) = 0;\ p_{11}(2) = 0,$ $p_{12}(2) = 1;\ p_{21}(1) = 0, p_{22}(2) = 1;\ r_1(1) = 1, r_1(2) = 0;\ r_2(1) = 2$. Let $\alpha \in [0,1)$ be the discount factor.
There are two deterministic stationary policies: f_1^∞ and f_2^∞, corresponding to action 1 and 2 in, respectively. It is easy to see that $v_1(f_1^\infty) = \frac{1}{1-\alpha}$, $v_2(f_1^\infty) = \frac{2}{1-\alpha}$, $v_1(f_2^\infty) = \frac{2\alpha}{1-\alpha}$ and $v_2(f_2^\infty) = \frac{2}{1-\alpha}$. Hence, f_1^∞ is optimal for $\alpha \le \frac{1}{2}$ and f_2^∞ is optimal for $\alpha \ge \frac{1}{2}$.
Next, consider a stationary policy π^∞ with $\pi_{11} = \beta$ and $\pi_{12} = 1 - \beta$. Then, $P(\pi) = \left(\begin{smallmatrix}\beta & 1-\beta \\ 0 & 1\end{smallmatrix}\right)$ and $r(\pi) = \left(\begin{smallmatrix}\beta \\ 2\end{smallmatrix}\right)$. The vector $v^\alpha(\pi^\infty)$ is the unique solution of the system $\{I - \alpha P(\pi)\}x = r(\pi)$. After some algebra, we obtain $v_1^\alpha(\pi^\infty) = \frac{\beta+2\alpha-3\alpha\beta}{(1-\alpha)(1-\alpha\beta)}$ and $v_2^\alpha(\pi^\infty) = \frac{2\alpha}{1-\alpha}$.
Now, we shall compute the best policy in $C(S)$ for the sum criterion with $K = 2$. Let $r_i^k(a) = r_i(a)$ for

$k = 1, 2$ and for all $(i, a) \in S \times A$ and let $\alpha_1 = \frac{1}{5}$ and $\alpha_2 = \frac{3}{5}$. Then, we obtain
$v_1^{\alpha_1}(\pi^\infty) = \frac{5+5\beta}{10-2\beta)}$, $v_2^{\alpha_1}(\pi^\infty) = \frac{1}{2}$, $v_1^{\alpha_2}(\pi^\infty) = \frac{15-5\beta}{5-3\beta)}$ and $v_2^{\alpha_2}(\pi^\infty) = 3$. Hence, $w_1(\pi^\infty) = \frac{5\beta^2-120\beta+175}{6\beta^2-40\beta+50}$ and $w_2(\pi^\infty) = \frac{7}{2}$. Taking the derivative with respect to β and equating to zero, we see that $w_1(\pi^\infty)$ is optimal for $\beta = \frac{20-5\sqrt{5}}{13} \approx 0.8722$, with $v_1^{\alpha_1}(\pi^\infty) \approx 1.1339$, $v_1^{\alpha_2}(\pi^\infty) \approx 2.6341$ and $w_1(\pi^\infty) \approx 3.7680$.
Finally, define $R = (\pi_1, \pi_2, \dots) \in C(UD)$ with $\pi_1 := f_1$ and $\pi_2 := f_2$ for $t \geq 2$. Direct calculation yields $v_1^{\alpha_1}(R) = 1 + \frac{2\alpha_1^2}{1-\alpha_1} = 1.1$, $v_1^{\alpha_2}(R) = 1 + \frac{2\alpha_2^2}{1-\alpha_1} = 2.8$, so $w_1(R) = 3.9$.
We conclude that for $\varepsilon < 3.9 - 3.7680 = 0.1320$ there does not exists an ε-optimal policy in $C(S)$, and moreover that the best policy in C(S) is strictly better that the best policy in C(D).

Assume, without loss of generality, that the discount factors satisfy $\alpha_1 > \alpha_2 > \cdots > \alpha_K$. We establish the existence of optimal Markov policies for the sum criterion by embedding our model into an MDP model with an infinite state space $S \times \mathbb{N}$, where $\mathbb{N} := \{1, 2, \dots\}$. Denote the genetic state by (i, t) for $i \in S$ and $t \in \mathbb{N}$. The actions and immediate rewards stay unchanged, i.e. the action set in state (i, t) is $A(i)$ for all $i \in S$ and $t \in \mathbb{N}$, and $r^k_{(i,t)}(a) := r^k_i(a)$ for all $i \in S, t \in \mathbb{N}$ and $k = 1, 2, \dots, K$. The new transition probabilities are defined through

$$p_{(i,t)(j,s)}(a) := \begin{cases} p_{ij}(a) & \text{if } s = t+1 \\ 0 & \text{otherwise} \end{cases} \quad \text{for all } i, j \in S,\ t, s \in \mathbb{N} \text{ and } a \in A(i).$$

We can write the reward (7.57) by $w_i(R) = \mathbb{E}_{i,R}\left\{\sum_{t=1}^\infty (\alpha_1)^{t-1} \sum_{k=1}^K \left(\frac{\alpha_k}{\alpha_1}\right)^{t-1} r^k_{X_t}(Y_t)\right\}$.
Letting $r_{(i,t)}(a) := \sum_{k=1}^K \left(\frac{\alpha_k}{\alpha_1}\right)^{t-1} r^k_{(i,t)}(a) = \sum_{k=1}^K \left(\frac{\alpha_k}{\alpha_1}\right)^{t-1} r^k_i(a)$, we have for the sum of discounted rewards $\overline{w}(R)$ in the new MDP model

$$\overline{w}_{(i,1)}(R) = \mathbb{E}_{i,R}\left\{\sum_{t=1}^\infty (\alpha_1)^{t-1}\, r^k_{\overline{X}_t}(\overline{Y}_t)\right\}, \tag{7.59}$$

where $\overline{X}_t, \overline{Y}_t$ are the state and action, respectively, at stage t in the new MDP model.

It is well known (e.g. see [227], Theorem 6.2.10 on page 154) that an MDP with an infinite state space, finite action sets and bounded one-step rewards has an optimal policy in the class $C(D)$. Notice that there is a one-to-one correspondence between the set of stationary deterministic policies in the new MDP model and the set of deterministic Markov policies in the original MDP model. Namely, let $\overline{f}^\infty$ be a stationary deterministic policy in the new model with action $\overline{f}(i, t)$ in state (i, t); then, the corresponding deterministic Markov policy is policy $R = (f_1, f_2, \dots)$ with $f_t(i) := \overline{f}(i, t)$; furthermore, $\overline{w}_{(i,1)}(\overline{f}^\infty) = w_i(R)$ for all $i \in S$. Hence, let $\overline{f}^\infty_*$ be an optimal stationary deterministic policy in the new MDP model. Then, $R_* = (f_1^*, f_2^*, \dots)$ with $f_t^*(i) := \overline{f}_*(i, t)$ is an optimal deterministic Markov policy in the original model. Therefore, we have the following result.

Theorem 7.22
There exists an optimal deterministic Markov policy for any MDP with optimality criterion the sum of discount factors as defined in (7.58).

The next theorem shows the existence of an ε-optimal policy in the class $C(UD)$.

Theorem 7.23
For any $\varepsilon > 0$ there exists an ε-optimal ultimately deterministic policy.

Proof

Let $R_* = (g_1, g_2, \dots)$ be an optimal deterministic Markov policy for the sum of discounted rewards as optimality criterion. Choose $T \in \mathbb{N}$ such that $M \cdot \frac{(\alpha_1)^T}{1-\alpha_1} \leq \frac{\varepsilon}{2K}$, where $M := max_{1 \leq k \leq K}\, max_{(i,a)}\, |r_i^k(a)|$. Let $h^\infty \in C(D)$ be an optimal policy for the MDP with one-step-rewards $r^1(a)$ and discount factor α_1. Define the policy $R \in C(UD)$ by $R = (f_1, f_2, \dots)$ with $f_t := \begin{cases} g_t & \text{for } t \leq T; \\ h & \text{for } t > T. \end{cases}$ Then, we obtain

$$\begin{array}{rcl} w_i - w_i(R) & = & w_i(R_*) - w_i(R) \\ & = & \mathbb{E}_{i,R_*}\left\{\sum_{t=1}^\infty \sum_{k=1}^K (\alpha_k)^{t-1} r^k_{X_t}(Y_t)\right\} - \mathbb{E}_{i,R}\left\{\sum_{t=1}^\infty \sum_{k=1}^K (\alpha_k)^{t-1} r^k_{X_t}(Y_t)\right\} \\ & = & \mathbb{E}_{i,R_*}\left\{\sum_{t=T+1}^\infty \sum_{k=1}^K (\alpha_k)^{t-1} r^k_{X_t}(Y_t)\right\} - \mathbb{E}_{i,R}\left\{\sum_{t=T+1}^\infty \sum_{k=1}^K (\alpha_k)^{t-1} r^k_{X_t}(Y_t)\right\} \\ & \leq & \sum_{t=T+1}^\infty \sum_{k=1}^K (\alpha_k)^{t-1} M + \sum_{t=T+1}^\infty \sum_{k=1}^K (\alpha_k)^{t-1} M \\ & \leq & 2KM \sum_{t=T+1}^\infty (\alpha_1)^{t-1} = 2KM \cdot \frac{(\alpha_1)^T}{1-\alpha_1} \leq \varepsilon,\ i \in S. \end{array}$$

so the ultimately deterministic policy R is ε-optimal. □

Let v^k be the value vector of the discounted problem with discount factor α_k and one-step-rewards $r_i^k(a)$, $(i,a) \in S \times A$. Let the vector $\underline{v}^k$ be such that $v^k(R) \geq \underline{v}^k$ for all policies R, e.g. take for $\underline{v}^k$ the constant vector with elements $(1-\alpha_k)^{-1} \cdot min_{(i,a)}\, r_i^k(a)$. Denote by $A_1(i)$ the set of conserving actions in the problem with $k = 1$, i.e. $A_1(i) := \{a \in A(i) \mid v_i^1 = r_i^1(a) + \alpha_1 \sum_j p_{ij}(a) v_j^1\},\ i \in S$.

It is well known (see section 3.3) that f^∞ is an optimal policy for the problem with $k = 1$ if and only if $f(i) \in A_1(i)$ for all $i \in S$. Let $S_1 := \{i \in S \mid A_1(i) \neq A(i)\}$. Furthermore, let

$$\varepsilon_1 := \begin{cases} min_{i \in S_1}\, v_i^1 - max_{a \in A(i) \backslash A_1(i)}\, \{r_i^1(a) + \alpha_1 \sum_j p_{ij}(a) v_j^1\} & \text{if } S_1 \neq \emptyset \\ 0 & \text{otherwise} \end{cases} \quad \text{and let}$$

$$T_1 := \begin{cases} min\left\{t \geq 1 \mid \sum_{k=2}^K \left(\frac{\alpha_k}{\alpha_1}\right)^{t-1} \cdot max_i\, (v_i^k - \underline{v}_i^k) < \varepsilon_1\right\} & \text{if } \varepsilon_1 > 0; \\ 1 & \text{if } \varepsilon_1 = 0. \end{cases}$$

Lemma 7.27

Let $R = (f^1, f^2, \dots)$ be an optimal deterministic Markov policy for the sum of discounted rewards problem and let $t \geq T_1$. Then, $f^t(i) \in A_1(i)$ for all $i \in S$.

Proof

If $S_1 = \emptyset$, then $A_1(i) = A(i)$ for all $i \in S$, and the lemma is trivial. Therefore, we consider the case $S_1 \neq \emptyset$, in which case $\varepsilon_1 > 0$.

Take any $f \in C(D)$ and any $t \in N$. Let (i,l) be any pair of states satisfying $\mathbb{P}_{i,f^\infty}\{X_t = l\} > 0$. We will show that

$$\mathbb{E}_{i,R}\left\{\sum_{s=t}^\infty \sum_{k=1}^K (\alpha_k)^{s-1} r^k_{X_s}(Y_s) \;\Big|\; X_t = l\right\} \geq \mathbb{E}_{i,f^\infty}\left\{\sum_{s=t}^\infty \sum_{k=1}^K (\alpha_k)^{s-1} r^k_{X_s}(Y_s) \;\Big|\; X_t = l\right\}. \tag{7.60}$$

To prove (7.60) by contradiction, we define a policy $\overline{R} = (\overline{\pi}^1, \overline{\pi}^2, \dots)$ through

$$\overline{\pi}^s_{i_1 a_1 \cdots i_{s-1} a_{s-1} i_s} := \begin{cases} f(i_s) & \text{if } s \geq t \text{ and } i_t = l; \\ f^s(i_s) & \text{otherwise.} \end{cases}$$

Hence, $\overline{R}$ and R are identical unless $X_t = l$, which has a positive probability under policy f^∞, and in that case $\overline{R}$ follows f^∞ from stage t. Therefore, assuming that (7.60) does not hold, we can write

$$\begin{aligned}
w_i(R) &= \mathbb{E}_{i,R}\big\{\textstyle\sum_{s=1}^{\infty}\sum_{k=1}^{K}(\alpha_k)^{s-1}r^k_{X_s}(Y_s)\big\}\\
&= \mathbb{E}_{i,R}\big\{\textstyle\sum_{s=1}^{t-1}\sum_{k=1}^{K}(\alpha_k)^{s-1}r^k_{X_s}(Y_s)\big\} + \mathbb{E}_{i,R}\big\{\textstyle\sum_{s=t}^{\infty}\sum_{k=1}^{K}(\alpha_k)^{s-1}r^k_{X_s}(Y_s)\big\}\\
&= \mathbb{E}_{i,R}\big\{\textstyle\sum_{s=1}^{t-1}\sum_{k=1}^{K}(\alpha_k)^{s-1}r^k_{X_s}(Y_s)\big\} + \mathbb{E}_{i,R}\big\{\textstyle\sum_{s=t}^{\infty}\sum_{k=1}^{K}(\alpha_k)^{s-1}r^k_{X_s}(Y_s) \bigm| X_t = l\big\} +\\
&\qquad\qquad \mathbb{E}_{i,R}\big\{\textstyle\sum_{s=t}^{\infty}\sum_{k=1}^{K}(\alpha_k)^{s-1}r^k_{X_s}(Y_s) \bigm| X_t \neq l\big\}\\
&< \mathbb{E}_{i,R}\big\{\textstyle\sum_{s=1}^{t-1}\sum_{k=1}^{K}(\alpha_k)^{s-1}r^k_{X_s}(Y_s)\big\} + \mathbb{E}_{i,f^\infty}\big\{\textstyle\sum_{s=t}^{\infty}\sum_{k=1}^{K}(\alpha_k)^{s-1}r^k_{X_s}(Y_s) \bigm| X_t = l\big\} +\\
&\qquad\qquad \mathbb{E}_{i,R}\big\{\textstyle\sum_{s=t}^{\infty}\sum_{k=1}^{K}(\alpha_k)^{s-1}r^k_{X_s}(Y_s) \bigm| X_t \neq l\big\}\\
&= w_i(\overline{R}).
\end{aligned}$$

This contradicts the optimality of R, so (7.60) is proved. For the optimal deterministic Markov policy $R = (f^1, f^2, \dots)$ we consider the shifted policies $R^s := (f^s, f^{s+1}, \dots)$ for $s = 1, 2, \dots$. We can prove, analogously as we proved (7.60), that $\sum_{k=1}^K(\alpha_k)^{t-1}v^k_l(R^t) \geq \sum_{k=1}^K(\alpha_k)^{t-1}v^k_l(f^\infty)$, i.e.

$$\sum_{k=1}^K(\alpha_k)^{t-1}\{v^k_l(R^t) - v^k_l(f^\infty)\} \geq 0. \tag{7.61}$$

To prove Lemma 7.27 by contradiction, we assume that for some $t \geq T_1$ there exists a state $l \in S$ with $f^t(l) \notin A_1(l)$. Let $f^\infty \in C(D)$ be an optimal policy for the discounted problem with $k = 1$. Then, $f(i) \in A_1(i)$ for all $i \in S$. Then, we have by the definition of T_1,

$$\varepsilon_1 > \sum_{k=2}^K \big(\frac{\alpha_k}{\alpha_1}\big)^{t-1} \cdot (v^k_l - \underline{v}^k_l) \geq \sum_{k=2}^K \big(\frac{\alpha_k}{\alpha_1}\big)^{t-1} \cdot \{v^k_l(R^t) - v^k_l(f^\infty)\}. \tag{7.62}$$

The optimality of f^∞ for the discounted problem with $k = 1$ and inequality (7.61) imply $v^1_l - v_l(R^t) = v^1_l(f^\infty) - v_l(R^t)$ and $\alpha_1^{t-1}\{v_l(R^t) - v^1_l(f^\infty)\} + \sum_{k=2}^K \alpha_k^{t-1} \cdot \{v_l(R^t) - v^1_l(f^\infty)\} \geq 0$. Hence, $\sum_{k=2}^K \big(\frac{\alpha_k}{\alpha_1}\big)^{t-1} \cdot \{v_l(R^t) - v^1_l(f^\infty)\} \geq v^1_l(f^\infty) - v_l(R^t) = v^1_l - v_l(R^t)$. Consequently, by (7.62),

$$\varepsilon_1 > \sum_{k=2}^K \big(\frac{\alpha_k}{\alpha_1}\big)^{t-1} \cdot (v^k_l - \underline{v}^k_l) \geq \sum_{k=2}^K \big(\frac{\alpha_k}{\alpha_1}\big)^{t-1} \cdot \{v^k_l(R^t) - v^k_l(f^\infty)\} \geq v^1_l - v_l(R^t). \tag{7.63}$$

Because $f^t(l) \in A(i)\backslash A_1(l)$, we obtain from the definition of ε_1

$$\begin{aligned}
\varepsilon_1 &\leq v^1_l - \{r^1_l\big(f^t(l)\big) + \alpha_1 \textstyle\sum_j p_{lj}\big(f^t(l)\big)v^1_j\}\\
&\leq v^1_l - \{r^1_l\big(f^t(l)\big) + \alpha_1 \textstyle\sum_j p_{lj}\big(f^t(l)\big)v^1_j(R^{t+1})\}\\
&= v^1_l - v^1_l(R^t),
\end{aligned}$$

which contradicts (7.63). □

If $A_1(i)$ is a singleton for each $i \in S$, then Lemma 7.27 implies that $f^t(i)$ is unique for all $i \in S$ and for all $t \geq T_1$. Therefore, in that case there is an optimal ultimately deterministic policy. Optimal actions $f^t(i)$ for $t = 1, 2, \dots, T_1 - 1$ can be found as solution of a T_1-step dynamic programming model with the one-step rewards at stage t for $1 \leq t \leq T_1 - 1$ given by $\sum_{k=1}^K (\alpha_k)^{t-1}r^k_i(a),\ (i,a) \in S \times A$ and at stage T_1 there are terminal rewards $\sum_{k=1}^K (\alpha_k)^{T_1-1}v^k_i(f^\infty)$, where the policy f^∞ takes actions from the singletons $A_1(i),\ i \in S$.

If $A_1(l)$ is not a singleton for some $l \in S$, then for $t \geq T_1$ the action set in state i may be reduced to $A_1(i)$ for all $i \in S$. Moreover, for any deterministic Markov policy $R = (f^1, f^2, \dots)$ using actions from $A_1(i)$ for all $i \in S$ one has $v^1(R) = v^1$. Hence, the expected rewards for criterion v^1 from stage T_1 onward are the same for any deterministic Markov policy $R = (f^1, f^2, \dots)$ satisfying $f^t(i) \in A_1(i)$ for all $i \in S$ and for all $t \geq T_1$.

Thus, if our goal is to construct an optimal policy from state T_1 onward, then we have reduced the problem with K reward functions $r^1, r^2, \dots, r^K$ to the problem with $K-1$ reward functions $r^2, r^3, \dots, r^K$. Let $A_0(i) := A(i),\ i \in S$, and let $v^{k,k}$ be the value vector of the problem with rewards r^k and discount factor α_k. Furthermore, let $A_k(i),\ i \in S$, be the set of conserving actions for the problem with value vector $v^{k,k}$, i.e.

$A_k(i) := \{a \in A_{k-1} \mid v_i^{k,k} = r_i^k(a) + \alpha_k \sum_j p_{ij}(a) v_j^{k,k}\},\ i \in S.$

Define S_k, ε_k and T_k by

$$S_k := \{i \in S \mid A_k(i) \neq A_{k-1}(i)\}.$$

$$\varepsilon_k := \begin{cases} min_{i \in S_k} \{v_i^{k,k} - max_{a \in A_{k-1}(i) \backslash A_k(i)} \{r_i^k(a) + \alpha_k \sum_j p_{ij}(a) v_j^{k,k}\}\} & \text{if } S_k \neq \emptyset; \\ 0 & \text{otherwise.} \end{cases}$$

$$T_k := \begin{cases} min\{t \geq T_{k-1} \mid \sum_{l=k+1}^{K} \left(\frac{\alpha_l}{\alpha_k}\right)^{t-1} \cdot max_i\, (v_i^{k,l} - \underline{v}_i^{k,l}) < \varepsilon_k\} & \text{if } \varepsilon_k > 0; \\ T_{k-1} & \text{if } \varepsilon_k = 0, \end{cases}$$

where $v^{k,l}$ for $l = k+1, k+2, \dots, K$ is the value vector of the problem with rewards $r_i^l(a)$, discount factor α_l and action sets $A_{k-1}(i)$ and $\underline{v}^{k,l}$ is such that $v^{k,l}(R) \geq \underline{v}^{k,l}$ for all policies R, e.g. take $\underline{v}^{k,l}$ the constant vector with elements $(1-\alpha_l)^{-1} \cdot min_{(i,a) \in S \times A_{k-1}}\, r_i^l(a)$.

We are now ready to state an algorithm for the computation of an ultimately deterministic optimal policy. In this algorithm we use the notation $v^{1,l}$ and $\underline{v}^{1,l}$ for v^l and $\underline{v}^l$, respectively, for $l = 1, 2, \dots, K$.

Algorithm 7.7 *Determination of an ultimately deterministic optimal policy for an MDP with as optimality criterion the sum of discounted rewards.*

Input: Instance of an MDP, an integer $K \geq 2$ and discount factors $\alpha_1,\ \alpha_2,\ \alpha_K$ with $\alpha_1 > \alpha_2 > \cdots > \alpha_K$ and immediate rewards $r_i^k(a),\ (i,a) \in S \times A$.

Output: An ultimately deterministic optimal policy R.

```
1. $k := 1$; for all $i \in S$ do $A_0(i) := A(i)$; $T_0 := \infty$.

2. Compute for the MDP with discount factor $\alpha_k$ and immediate rewards $r_i^k(a)$:
     (a) the value vector $v^{k,k}$.
     (b) for all $i \in S$ do $A_k(i) := \{a \in A_{k-1} \mid v_i^{k,k} = r_i^k(a) + \alpha_k \sum_j p_{ij}(a) v_j^{k,k}\}$.
     (c) $S_k := \{i \in S \mid A_k(i) \neq A_{k-1}(i)\}$.
     (d) $\varepsilon_k := \begin{cases} min_{i \in S_k} \{v_i^{k,k} - max_{a \in A_{k-1}(i) \backslash A_k(i)} \{r_i^k(a) + \alpha_k \sum_j p_{ij}(a) v_j^{k,k}\}\} & \text{if } S_k \neq \emptyset; \\ 0 & \text{otherwise.} \end{cases}$

3. for $l = k+1, k+2, \dots, K$ do
        begin compute for the MDP with discount factor $\alpha_l$, immediate rewards $r_i^l(a)$ and action sets
              $A_{k-1}(i),\ i \in S$, the value vector $v^{k,l}$;
              $\underline{v}^{k,l} := (1-\alpha_l)^{-1} \cdot min_{(i,a) \in S \times A_{k-1}}\, r_i^l(a)$.
     $T_k := \begin{cases} min\{t \geq T_{k-1} \mid \sum_{l=k+1}^{K} \left(\frac{\alpha_l}{\alpha_k}\right)^{t-1} \cdot max_i\, (v_i^{k,l} - \underline{v}_i^{k,l}) < \varepsilon_k\} & \text{if } \varepsilon_k > 0; \\ T_{k-1} & \text{if } \varepsilon_k = 0, \end{cases}$
        end

4. if $A_k(i)$ is a singleton for all $i \in S$ or if k = K then
       begin $T = T_k$; for all $i \in S$ do $A_*(i) := A_k(i)$; go to step 5 end
   else
       begin $k := k+1$; go to step 2 end
```

```
5. select a policy $f^\infty \in C(D)$ such that $f(i) \in A_*(i)$ for all $i \in S$.
6. for $k = 1$ step $1$ until $K$ do
   compute for the MDP with discount factor $\alpha_k$ and immediate rewards $r_i^k(a)$ the values $v_i^k(f^\infty),\ i \in S$.
7. Consider the $T$-step dynamic programming model with state space $S$, action sets $A(i),\ i \in S$,
   transition probabilities $p_{ij}(a),\ (i,j,a) \in S \times S \times A$, one-step rewards at stage $t$ given by
   $\sum_{k=1}^K (\alpha_k)^{t-1} r_i^k(a),\ (i,a) \in S \times A$ and at stage $T$ there are terminal rewards $\sum_{k=1}^K (\alpha_k)^{T_1-1} v_i^k(f^\infty)$;
   compute optimal actions $f^t(i),\ t = 1,2,\dots,T-1$ of this $T$-step dynamic programming model.
8. Define the ultimately deterministic policy $R$ by $R := (f^1, f^2, \dots, f^{T-1}, f, f, \dots)$ (STOP).
```

Theorem 7.24
Algorithm 7.7 is correct, i.e. it constructs an ultimately deterministic optimal policy.

Proof
We apply Lemma 7.27 iteratively at most K times. After the kth iteration we replace the original model by the model that starts at time point T_k. This means that the initial rewards $r^1, r^2, \dots, r^K$ are replaced by $\alpha_1^{T_k-1}r^1, \alpha_2^{T_k-1}r^2, \dots, \alpha_1^{T_k-1}r^K$. Lemma 7.27 allows us to reduce the action sets to $A_k(i)$ after the kth iteration. After a finite number (at most K) of iterations we have from Lemma 7.27 that the policy $f \in C(D)$ defined in step 5 of the algorithm is an optimal policy from stage T onward. Furthermore, any solution of the finite stage dynamic programming problem described in step 7 of the algorithm provides an optimal policy for the stages $1, 2, \dots, T-1$. Hence, the algorithm is correct. □

Remark
In order to compute T_k we need to compute $v^{k,l}$ and $\underline{v}^{k,l}$ for $l = k+1, k+2, \dots, K$. We have already mentioned that to avoid the computation of the largest lower bound for $v^{k,l}$, i.e. $min_R\, v^{k,l}(R)$ one may use the trivial lower bound $(1-\alpha_l)^{-1} \cdot min_{(i,a) \in S \times A_{k-1}}\, r_i^l(a)$. Similarly, one may replace the smallest upper bound $v^{k,l} = max_R\, v^{k,l}(R)$ by the trivial upper bound $(1-\alpha_l)^{-1} \cdot max_{(i,a) \in S \times A_{k-1}}\, r_i^l(a)$. This results in a considerable reduction in the computation of T_k. However, this leads to a larger value of T_k and, consequently, to a larger value for T. Hence, it increases the complexity of the finite horizon problem in step 7 of the algorithm.

7.14 Bibliographic notes

The material presented in this chapter has its roots in Blackwell's seminal paper [29]. Among other results, Blackwell introduced the criteria we now refer to as 0-discount optimality and Blackwell optimality (he referred to policies which achieve these criteria as *nearly optimal* and *optimal*, respectively). He demonstrated the existence of a Blackwell optimal policy and convergence of the multichain average reward policy iteration method through use of a *partial* Laurent series expansion. That paper raised also the following challenging questions:

a. What is the relationship between 0-discount optimality and Blackwell optimality?
b. When are average optimal policies 0-discount optimal?
c. How does one compute 0-discount optimal and Blackwell optimal policies?

Veinott ([308], [311] and [312]) and Miller and Veinott ([199]) addressed these issues. In his 1966 paper [308], Veinott provided a policy iteration algorithm for finding a 0-discount optimal policy (he referred to such policy as 1-*optimal*). In his comprehensive 1969 paper [311], Veinott provides the link between 0-discount optimality and Blackwell optimality. Miller and Veinott developed the *complete* Laurent series expansion and related it to the lexicographic ordering (see section 7.4) and used this ordering to provide a finite policy iteration method for n-discount optimality for any n. For the presentation of the lexicographic ordering we refer also to Dekker ([53]) and Dekker and Hordijk ([54]). Miller and Veinott also showed that $N-1$-discount optimality is equivalent to Blackwell optimality (see section 7.5). In his 1974 paper ([312]) Veinott provides a more accessible overview of the above work and a simplified presentation of the main results in [311].

In [208] O'Sullivan presents another method for finding an n-discount optimal policy for a substochastic MDP by solving a sequence of $3n + 5$ simpler subproblems which can be solved using either policy improvement or linear programming. He also presents a new method for determining the coefficients of the Laurent expansion for a substochastic MDP. This method reduces the problem to that of finding the coefficients within a number of irreducible, substochastic systems.

The linear programming method for the computation of an n-discount optimal policy in the irreducible case (see section 7.6) is due to Avrachenkov and Altman ([7]).

The approach for the computation of a Blackwell optimal policy by linear programming, based on asymptotic linear programming (see section 7.7), was proposed by Hordijk, Dekker and Kallenberg ([125]). Unfortunately, this method cannot be used for the calculation of n-discount optimal policies that are not Blackwell optimal. Related papers are written by Smallwood ([274]) and Jeroslow ([141]).

The material of the sections 7.8 and 7.10 is taken from Denardos paper [61]. Denardo ([59]) has proposed the three-step procedure with linear programming to find a bias optimal policy. Kallenberg ([148]) has improved and streamlined this approach, which is presented in section 7.9.

Veinott ([308]) introduced in 1966 the criterion of average overtaking optimality. Already in 1965 Brown showed ([34]) that an overtaking optimal policy need not exists, in general. Example 7.8 is due to Denardo and Miller ([65]). Denardo and Rothblum ([66]) have presented an additional assumption under which overtaking optimal policies do exist. Lippman ([181]) has shown the equivalence between average overtaking optimality and 0-discount optimality. Sladky ([273]) generalized overtaking and average overtaking to n-average optimality. He showed the equivalence between n-average optimality and n-discount optimality. Rothblum and Veinott ([248]) and Rothblum ([246]) generalized and unified the overtaking optimality criteria through the introduction of a family of (n, k)-optimality criteria. Their concepts of $(0, 0)$, $(0, 1)$ and $(1, 0)$-optimality are equivalent to overtaking, average overtaking and cumulative overtaking optimality, respectively.

The section on a weighted combination of discounted and average rewards is due to Krass, Filar and Sinha ([172]). The last section, dealing with the sum of several expected total discounted rewards with different one-step rewards and discount factors, is based on a paper by Feinberg and Shwartz ([84]).

7.15 Exercises

Exercise 7.1

Give a direct proof Lemma 7.3, i.e. if a policy is n-average optimal, then it is m-average optimal for $m = -1, 0, \dots, n$.

Exercise 7.2
Sow that for all $n \geq 0,\ T \geq 2$ and $f^\infty \in C(D)$:

a. $v^{n,T}(f^\infty) = v^{n,T-1}(f^\infty) + v^{n-1,T}(f^\infty)$.

b. $v^{n,T}(f^\infty) = \binom{T+n}{n+1} r(f) + P(f)v^{n,T-1}(f^\infty)$.

Exercise 7.3
Consider the following model:
$S = \{1,2\};\ A(1) = \{1,2\},\ A(2) = \{1\};\ r_1(1) = 5,\ r_1(2) = 10,\ r_2(1) = -1.$
$p_{11}(1) = 0.5,\ p_{12}(1) = 0.5;\ p_{11}(2) = 0,\ p_{12}(1) = 1;\ p_{21}(1) = 0,\ p_{22}(1) = 1.$
Determine for both deterministic policies the α-discounted rewards and for which α the policy is optimal.

Exercise 7.4
Consider the following model:
$S = \{1,2\};\ A(1) = \{1,2\},\ A(2) = \{1\};\ r_1(1) = 1,\ r_1(2) = 2,\ r_2(1) = 0.$
$p_{11}(1) = 0.5,\ p_{12}(1) = 0.5;\ p_{11}(2) = 0,\ p_{12}(1) = 1;\ p_{21}(1) = 0,\ p_{22}(1) = 1.$
Determine for both deterministic policies the Laurent series expansion in $1-\alpha$ and derive from this expansion the vectors $u^k(f),\ k = -1, 0, 1, \dots$.

Exercise 7.5
Consider the following model:
$S = \{1,2,3\};\ A(1) = \{1,2\},\ A(2) = A(3) = \{1\};\ r_1(1) = a,\ r_1(2) = 1,\ r_2(1) = b,\ r_3(1) = 0.$
$p_{11}(1) = 0,\ p_{12}(1) = 1,\ p_{13}(1) = 0;\ p_{11}(2) = 0.5,\ p_{11}(2) = 0.5,\ p_{13}(2) = 0;$
$p_{21}(1) = 0,\ p_{22}(1) = 0,\ p_{23}(1) = 1;\ p_{31}(1) = 0,\ p_{32}(1) = 0,\ p_{33}(1) = 1.$
Determine a and b such that both deterministic policies are (-1)-discount, 0-discount and
1-discount optimal. Which policy is Blackwell optimal?

Exercise 7.6
Let for $f^\infty \in C(D)$ and $\rho > 0$ the *resolvent* $R^\rho(f)$ be defined by $R^\rho(f) = \{\rho I + (I - P(f)\}^{-1}$.
Show that

(1) $R^\rho(f) = \alpha\{I - \alpha P(f)\}^{-1}$, where $\alpha = \frac{1}{1+\rho}$.

(2) $\lim_{\rho \downarrow 0} \rho R^\rho(f) = P^*(f)$.

Exercise 7.7
Determine a 0-optimal policy with Algorithm 7.1 for the following model:
$S = \{1,2\};\ A(1) = \{1,2\},\ A(2) = \{1\};\ r_1(1) = 4,\ r_1(2) = 0,\ r_2(1) = 8.$
$p_{11}(1) = 1,\ p_{12}(1) = 0;\ p_{11}(2) = 0,\ p_{12}(2) = 1;\ p_{21}(1) = 1,\ p_{22}(1) = 0.$
Start with $f(1) = 2,\ f(2) = 1$.

Exercise 7.8
Show that $v^\alpha(R) - v^\alpha(f^\infty) = \sum_{k=-1}^\infty \rho^k \psi^k(f,g)$ for the nonstationary policy $R = (g, f, f, f, \dots)$.

Exercise 7.9

Consider the following model:
$S = \{1,2,3,4\};\ A(i) = \{1,2\},\ i = 1,2,3,4.$

$$\begin{array}{lllll}
p_{11}(1) = 0; & p_{12}(1) = 1; & p_{13}(1) = 0; & p_{14}(1) = 0; & r_1(1) = 1 \\
p_{11}(2) = 0; & p_{12}(2) = \frac{1}{2}; & p_{13}(2) = \frac{1}{2}; & p_{14}(2) = 0; & r_1(2) = \frac{1}{2} \\
p_{21}(1) = 0; & p_{22}(1) = 0; & p_{23}(1) = 1; & p_{24}(1) = 0; & r_2(1) = 0 \\
p_{21}(2) = 0; & p_{22}(2) = 0; & p_{23}(2) = \frac{1}{3}; & p_{24}(2) = \frac{2}{3}; & r_2(2) = -\frac{2}{3} \\
p_{31}(1) = 0; & p_{32}(1) = 0; & p_{33}(1) = 0 & p_{34}(1) = 1; & r_3(1) = 0. \\
p_{31}(2) = \frac{1}{4}; & p_{32}(2) = 0; & p_{33}(2) = 0; & p_{34}(2) = \frac{3}{4}; & r_3(2) = \frac{1}{2} \\
p_{41}(1) = 1; & p_{42}(1) = 0; & p_{43}(1) = 0; & p_{44}(1) = 0; & r_4(1) = 3 \\
p_{41}(2) = \frac{1}{4}; & p_{42}(2) = \frac{3}{4}; & p_{43}(2) = 0; & p_{44}(2) = 0; & r_4(2) = 3
\end{array}$$

With (i,j,k,l), where $i,j,k,l \in \{1,2\}$, we denote the 16 policies, so $(1,2,2,1)$ is the policy that takes action 1 in state 1 and 4, and action 2 in the states 2 and 3.

a. The model is irreducible: show that the policy $(1,2,2,1)$ is irreducible.

b. Formulate the primal and dual linear program for an (-1)-discount optimal policy.

c. Solve the linear programs (use any package that is availabe for you).

d. Determine, using the in c obtained solution, the set of (-1)-discount optimal policies.

e. Formulate the primal and dual linear program for an 0-discount optimal policy.

f. Solve the linear programs for an 0-discount optimal policy.

g. Determine, using the in f obtained solution, the set of 0-discount optimal policies.

h. Formulate the primal and dual linear program for an 1-discounted optimal policy.

i. Solve the linear programs for an 1-discounted optimal policy.

j. Determine, using the in i obtained solution, the set of 1-discounted optimal policies.

Exercise 7.10

Show that the ordering of $F(\mathbb{R})$ given by (7.30) is a correct total ordering.

Exercise 7.11

Consider the following MDP:

$S = \{1,2\};\ A(1) = \{1,2,3\},\ A(2) = \{1\};\ r_1(1) = 1,\ r_1(2) = \frac{3}{4},\ r_1(3) = \frac{1}{2};\ r_2(1) = 0.$

$p_{11}(1) = 0,\ p_{12}(1) = 1;\ p_{11}(2) = \frac{1}{2},\ p_{12}(2) = \frac{1}{2};\ p_{11}(3) = 1,\ p_{12}(3) = 0;,\ p_{21}(1) = 0,\ p_{22}(1) = 1.$

Determine by the linear programming method for rational functions optimal policies for all discount factors $\alpha \in (0.1)$.

Exercise 7.12

Consider the following MDP:
$S = \{1,2,3\};\ A(1) = \{1,2\},\ A(2) = \{1,2\},\ A(3) = \{1\}.$
$r_1(1) = 1,\ r_1(2) = 2;\ r_2(1) = 2,\ r_2(2) = 0;\ r_3(1) = 0.$
$p_{11}(1) = 1,\ p_{12}(1) = 0,\ p_{13}(1) = 0;\ p_{11}(2) = 0,\ p_{12}(2) = 0,\ p_{13}(1) = 1;$
$p_{21}(1) = 1,\ p_{22}(1) = 0,\ p_{23}(1) = 0;\ p_{21}(2) = 0,\ p_{22}(2) = 0,\ p_{23}(2) = 1;$
$p_{31}(1) = 1,\ p_{32}(1) = 0,\ p_{33}(1) = 0.$
Determine a bias optimal policy by the linear programming method for unichain MDPs.

Exercise 7.13

Consider the MDP of Example 7.8 with the two deterministic policies f_1^∞ and f_1^∞. Determine $\liminf_{T\to\infty} \frac{1}{T}\sum_{t=1}^T \{v_i^t(f_1^\infty) - v_i^t(f_2^\infty)\}$ and $\liminf_{T\to\infty} \frac{1}{T}\sum_{t=1}^T \{v_i^t(f_2^\infty) - v_i^t(f_1^\infty)\}$ for each $i \in S$. Is f_1^∞ an average overtaking policy? Is f_2^∞ an average overtaking policy?

Chapter 8

Special models

In this chapter we deal with the models which were introduced in section 1.3. The red-black gambling model was already discussed in section 4.12.1, the model ’How to serve in tennis’ was left as Exercise 1.5 to the reader and for the optimal stopping problem we refer to the sections 1.3.3 and 4.12.2. In the next sections we discuss the following models:

8.1 Replacement problems

In this section we discuss four variants of the replacement problem:
- a general replacement model;
- a replacement model with increasing deterioration;
- a skip to the right model with failure costs;
- a separable model.

8.1.1 A general replacement model

In this general replacement model we have state space $S = \{0, 1, \dots, N\}$, where state 0 corresponds to a new item, and action sets $A(0) = \{1\}$ and $A(i) = \{0, 1\},\ i \neq 0$, where action 0 means replacing the present item by a new item. We consider in this model costs instead of rewards. Let c be the cost of a new item. Furthermore, assume that an item of state i has trade-in-value s_i and maintenance costs c_i. If in state i action 0 is chosen, then $c_i(0) = c - s_i + c_0$ and $p_{ij}(0) = p_{0j},\ j \in S$; for action 1, we have $c_i(1) = c_i$ and $p_{ij}(1) = p_{ij},\ j \in S$. In contrast with other replacement models, where the state is determined by the age of the item, we allow that the state of the item may change to any other state. In this case the optimal replacement policy is in general no a control-limit rule. As optimality criterion we consider the expected total discounted reward. For this model the primal linear program, which yields the value vector v^α, is:

$$\min\left\{\sum_{j=0}^N \beta_j v_j \;\middle|\; \begin{array}{rcll} \sum_{j=0}^N (\delta_{ij} - \alpha p_{0j})v_j & \geq & -c + s_i - c_0, & 1 \leq i \leq N \\ \sum_{j=0}^N (\delta_{ij} - \alpha p_{ij})v_j & \geq & -c_i, & 0 \leq i \leq N \end{array}\right\}, \tag{8.1}$$

where $\beta_j > 0,\ j \in S$, are some given numbers. Because there is only one action in state 0, namely action 1, we have $v_0^\alpha = -c_0 + \alpha \sum_{j=0}^N p_{0j} v_j^\alpha$. Hence, instead of $\sum_{j=0}^N (\delta_{ij} - \alpha p_{0j})v_j \geq -c + s_i - c_0$, we can write $v_i - v_0 \geq -c + s_i$, obtaining the equivalent linear program

$$\min\left\{\sum_{j=0}^N \beta_j v_j \;\middle|\; \begin{array}{rll} v_i - v_0 & \geq r_i, & 1 \leq i \leq N \\ \sum_{j=0}^N (\delta_{ij} - \alpha p_{ij})v_j & \geq -c_i, & 0 \leq i \leq N \end{array}\right\}, \tag{8.2}$$

where $r_i := -c + s_i,\ i \in S$. The dual linear program of (8.2) is:

$$\max\left\{ \sum_{i=1}^N r_i x_i - \sum_{i=0}^N c_i y_i \;\middle|\; \begin{array}{rcll} -\sum_{i=1}^N x_i + \sum_{i=0}^N (\delta_{i0} - \alpha p_{i0}) y_i & = & \beta_0 & \\ x_j + \sum_{i=0}^N (\delta_{ij} - \alpha p_{ij}) y_i & = & \beta_j, & 1 \le j \le N \\ x_i & \ge & 0, & 1 \le i \le N \\ y_i & \ge & 0, & 0 \le i \le N \end{array} \right\}. \tag{8.3}$$

Theorem 8.1

There is a one-to-one correspondence between the extreme solutions of (8.3) and the set of deterministic policies.

Proof

Let (x, y) be an extreme solution of (8.3). Then, (x, y) has exactly $N + 1$ positive components. Since $y_0 = \beta_0 + \sum_{i=1}^N x_i + \alpha \sum_{i=0}^N p_{i0} y_i \ge \beta_0 > 0$ and $x_j + y_j = \beta_j + \alpha \sum_{i=0}^N p_{ij} y_i \ge \beta_j > 0,\ 1 \le j \le N$, in each state $j,\ 0 \le j \le N$, either x_j or y_j is strictly positive. Hence, (x, y) corresponds to the deterministic policy, defined by: if $x_j > 0$, then action 0 (replacement) is chosen and if $x_j = 0$, then $y_j > 0$ and action 1 (no replacement) is chosen.

Conversely, let f^∞ be a deterministic policy. Partition the states $\{1, 2, \dots, N\}$ in $S_0 \cup S_1$, where S_0 and S_1 correspond to the states in which action 0 and action 1, respectively, are chosen. Let $x_j := 0$ for all $j \in S_1$ and $y_j := 0$ for all $j \in S_0$. Then, the equations of (8.3) are equivalent to the following system of $N + 1$ equations with $N + 1$ variables:

$$\left\{ \begin{array}{ccccccl} -\sum_{j \in S_0} x_j & + & y_0 & - & \alpha \sum_{i \in S_1} p_{i0} y_i & = & \beta_0 \\ x_j & & & - & \alpha \sum_{i \in S_1} p_{ij} y_i & = & \beta_j,\ j \in S_0 \\ & & y_j & - & \alpha \sum_{i \in S_1} p_{ij} y_i & = & \beta_j, j \in S_1 \end{array} \right. \tag{8.4}$$

Consider a linear combination of the columns of (8.4) which yields the 0-vector:

$$\left\{ \begin{array}{ccccccl} -\sum_{j \in S_0} \mu_j & + & \mu_0 & - & \alpha \sum_{i \in S_1} p_{i0} \mu_i & = & 0 \\ \mu_j & & & - & \alpha \sum_{i \in S_1} p_{ij} \mu_i & = & 0,\ j \in S_0 \\ & & \mu_j & - & \alpha \sum_{i \in S_1} p_{ij} \mu_i & = & 0,\ j \in S_1 \end{array} \right. \tag{8.5}$$

Since $\mu_0 = \alpha \sum_{i \in S_1} p_{i0} \mu_i + \sum_{j \in S_0} \mu_j = \alpha \sum_{i \in S_1} p_{i0} \mu_i + \sum_{j \in S_0} \{\alpha \sum_{i \in S_1} p_{ij} \mu_i\} = \alpha \sum_{j \in S_1} \{p_{i0} + \sum_{j \in S_0} p_{ij}\} \mu_i$, we have

$$\mu_j = \alpha \sum_{i \in S_1} q_{ij} \mu_i,\ j \in \{0\} \cup S_1, \tag{8.6}$$

where $q_{ij} := \left\{ \begin{array}{cl} p_{i0} + \sum_{j \in S_0} p_{ij} & i \in S_1,\ j = 0, \\ p_{ij} & i \in S_1,\ j \in S_1. \end{array} \right.$

Remark that Q is a probability matrix on $\{0\} \cup S_1$, because $q_{ij} \ge 0$ for all i and j, and also we have $\sum_{j \in \{0\} \cup S_1} q_{ij} = p_{i0} + \sum_{j \in S_0} p_{ij} + \sum_{j \in S_1} p_{ij} = \sum_{j \in S} p_{ij} = 1$ for all i. Let $\nu_j := \mu_j,\ j \in \{0\} \cup S_1$, then (8.6) implies that $\nu^T(I - \alpha Q) = 0$. Since $I - \alpha Q$ is nonsingular, we have $\nu = 0$, i.e. $\mu_j = 0$ for all $j \in \{0\} \cup S_1$. Then, from (8.5) it follows that $\mu_j = 0,\ j \in S_0$, implying that $\mu_j = 0$ for all j. Hence, the columns of (8.4) are linear independent and, consequently, (x, y) is an extreme solution of (8.3) and the correspondence is one-to-one. □

Consider the simplex method to solve (8.3) and start with the basic solution that corresponds to the policy which chooses action 1 (no replacement) in all states. Hence, in the first simplex tableau $y_j,\ 0 \le j \le N$, are the basic variables and $x_i,\ 1 \le i \le N$, the nonbasic variables. Take the usual version of the simplex method in which the column with the most negative cost is chosen as pivot column. It turns out, as will be shown in Theorem 8.2, that this choice gives the optimal action for that state, i.e. in that state action 0, the replacement action, is optimal. Hence, after interchanging x_i and y_i, the column of y_i can be deleted. Consequently, we obtain the following *greedy simplex algorithm*.

Algorithm 8.1 *The greedy simplex algorithm*
Input: Instance of a general replacement problem.
Output: An optimal deterministic policy f^∞.

```
1. Start with the basic solution corresponding to the nonreplacing actions.
2. if the reduced costs are nonnegative then the corresponding policy is optimal (STOP).
   otherwise
      begin
         select the column with the most negative reduced cost as pivot column;
         execute the usual simplex transformation;
         delete the pivot column
      end
3. if all columns are removed then replacement in all states is the optimal policy (STOP).
   otherwise return to step 2.
```

Theorem 8.2 *The greedy simplex algorithm is correct and has complexity $\mathcal{O}(N^3)$.*

Proof
For the correctness of the algorithm it has to be shown that the deletion of the pivot column is allowed. This will be shown by induction on the number of iterations. The first simplex tableau is correct, because no column had been deleted. Suppose that previous iterations were correct and consider the present simplex tableau, corresponding to policy f^∞, with basic variables $y_0, y_i,\ i \in S_1$, and $x_i,\ i \in S_0$. The reduced cost corresponding to state i and action a is in general (see section 3.5):

$$\textstyle\sum_j \{\delta_{ij} - \alpha p_{ij}(a)\} v_j^\alpha(f^\infty) - r_i(a).$$

For action 0 (replacement) we denote the reduced cost in state i by $w_i(f)$ and this quantity becomes

$$w_i(f) = v_i^\alpha(f^\infty) - \alpha \textstyle\sum_j p_{0j} v_j^\alpha(f^\infty) + (c - s_i + c_0) = v_i^\alpha(f^\infty) - v_0^\alpha(f^\infty) - r_i,$$

the last equality because $v_0^\alpha(f^\infty) = -c_0 + \alpha \sum_j p_{0j} v_j^\alpha(f^\infty)$.
For action 1 (no replacement) we denote the reduced cost in state i by $z_i(f)$ and we obtain

$$z_i(f) = \textstyle\sum_j \{\delta_{ij} - \alpha p_{ij}\} v_j^\alpha(f^\infty) + c_i.$$

Since the reduced costs corresponding to basic variables are zero, we have

$$\left\{ \begin{array}{rcll} \sum_j \{\delta_{ij} - \alpha p_{ij}\} v_j^\alpha(f^\infty) & = & -c_i, & i \in S_1 \cup \{0\} \\ v_i^\alpha(f) - v_0^\alpha(f^\infty) & = & r_i, & i \in S_0 \end{array} \right.$$

Let $S_0^* := \{i \mid \text{action 0 is in state } i \text{ optimal}\},\ S_1^* := S \backslash S_0^*$ and let f_*^∞ be an optimal policy.

Then, state 0 is in $S_1^*,\ S_0 \subseteq S_0^*$ (by the induction hypothesis), and $\left\{ \begin{array}{rcll} \sum_j \{\delta_{ij} - \alpha p_{ij}\} v_j^\alpha(f_*^\infty) & = & -c_i, & i \in \\ v_i^\alpha(f_*^\infty) - v_0^\alpha(f_*^\infty) & = & r_i, & i \in \end{array} \right.$

Assume that the column of the nonbasic variable x_k is chosen as pivot column. Then, the reduced cost of column k is the most negative reduced cost: $w_k(f) < 0$ and $w_k(f) \leq w_i(f),\ 1 \leq i \leq N$. It is sufficient to show that in state k action 0 is optimal, i.e. $k \in S_0^*$. Let $d(f) := v^\alpha(f) - v^\alpha(f_*)$. Then, because f_*^∞ is optimal, $S_0 \subseteq S_0^*$ and $v_i^\alpha(f_*^\infty) - v_0^\alpha(f_*^\infty) = r_i,\ i \in S_0^*$, we obtain

$$\begin{cases} d_i(f) = \alpha \sum_j p_{ij} d_j(f), & i \in S_1^* \\ d_i(f) = d_0(f) + w_i(f) \geq d_0(f) + w_k(f), & i \in S_0^* \end{cases} \tag{8.7}$$

Let $m \in S_1^*$ be such that $d_m(f) = min_{i \in S_1^*}\ d_i(f)$ and suppose that $d_m(f) \leq d_0(f) + w_k(f)$. Then, $d_m(f) \leq d_i(f),\ 0 \leq i \leq N$, and (8.7) implies that $d_m(f) = \alpha \sum_j p_{mj} d_j(f) \geq \alpha d_m(f)$, i.e. $d_m(f) \geq 0$. Since $0 \leq d_m(f) \leq d_i(f),\ 0 \leq i \leq N$, and $d(f) \leq 0$, we have $d(f) = 0$. This means that $v^\alpha(f^\infty) = v^\alpha(f_*^\infty) = v^\alpha$. Hence, the present simplex tableau is optimal which contradicts $w_k(f) < 0$. Therefore, we have shown that

$$d_0(f) + w_k(f) < d_m(f) \leq d_i(f) \text{ for all } i \in S_1^*.$$

Suppose that $k \in S_1^*$. Then,

$$v_k^\alpha(f^\infty) - v_k^\alpha(f_*^\infty) = d_k(f) \geq d_m(f) > d_0(f) + w_k(f) = v_0^\alpha(f^\infty) - v_0^\alpha(f_*^\infty) + w_k(f).$$

Hence,

$$v_k^\alpha = v_k^\alpha(f_*^\infty) < v_k^\alpha(f^\infty) - v_0^\alpha(f^\infty) + v_0^\alpha(f_*^\infty) - v_k^\alpha(f^\infty) + v_0^\alpha(f^\infty) + r_k = v_0^\alpha + r_k,$$

i.e. v^α is infeasible for (8.2): contradiction. This completes the first part of the proof.
In the first step of the algorithm, the simplex tableau for a specific basic solution has to be determined. This is one matrix inversion and has complexity $\mathcal{O}(N^3)$. Since in each iteration one column is removed, there are at most N iterations. In each iteration a simplex tranformation is executed which takes $\mathcal{O}(N^2)$. Hence, the overall complexity of the algorithm is $\mathcal{O}(N^3)$. □

Remark:
An optimal stopping problem may be considered as a special case of a replacement problem with as optimality criterion the total expected reward, i.e. $\alpha = 1$. In an optimal stopping problem there are two actions in each state. The first action is the stopping action and the second action corresponds to continue. If the stopping action is chosen in state i, then a final reward s_i is earned and the process terminates. If the second action is chosen, then a reward r_i is received and the transition probability of being in state j at the next decision time point is $p_{ij},\ j \in S$. This optimal stopping problem can is a special case of the replacement problem with $p_{0j} = 0$ for all $j \in S,\ c_i(0) = -s_i$ and $c_i(1) = -r_i$ for all $i \in S$. Hence, also for the optimal stopping problem, the linear programming approach of this section can be used and the complexity is also $\mathcal{O}(N^3)$.

8.1.2 A replacement model with increasing deterioration

Consider a replacement model with state space $S = \{0, 1, \dots, N+1\}$. An item is in state 0 if and only if it is new; an item is in state $N+1$ if and only if it is inoperative. There are two actions: action 0 is to replace the item by a new one and action 1 is not to replace the item; in the states 0 and $N+1$ only one action is possible, action 1 and action 0, respectively. Action 0 gives an instantaneous transition to state 0. Hence, the transition probabilities are:

$$p_{ij}(0) = p_{0j},\ 1 \leq i \leq N+1,\ j \in S, \text{ and } p_{ij}(1) = p_{ij},\ 0 \leq i \leq N,\ j \in S.$$

We assume two types of cost, the cost $c_0 \geq 0$ to replace an operative item and the cost $c_0 + c_1$, where $c_1 \geq 0$, to replace an inoperative item. We state the following equivalent (see Lemma 8.1) assumptions. If state i is interpreted as the condition of an item, then Assumption 8.2 means increasing deterioration.

Assumption 8.1

The transition probabilities are such that for every nondecreasing function $x_j,\ j \in S$, the function $F(i) := \sum_{j=0}^{N+1} p_{ij}x_j$ is nondecreasing in i.

Assumption 8.2

The transition probabilities are such that for every $k \in S$, the function $G_k(i) = \sum_{j=k}^{N+1} p_{ij}$ is nondecreasing in i.

Lemma 8.1

The Assumptions 8.1 and 8.2 are equivalent.

Proof

Let Assumption 8.1 hold. Take any $k \in S$. Then, for the nondecreasing function $x_j := \begin{cases} 0 & j < k \\ 1 & j \geq k \end{cases}$ the function $F(i) := \sum_{j=0}^{N+1} p_{ij}x_j = \sum_{j=k}^{N+1} p_{ij} = G_k(i)$ is nondecreasing in i.
Conversely, let Assumption 8.2 hold. Take any nondecreasing function $x_j,\ j \in S$.
Then, with $c_k := x_k - x_{k-1} \geq 0,\ 1 \leq k \leq N+1$, we can write $x_j = \sum_{k=1}^{j} c_k + x_0,\ 1 \leq j \leq N+1$.
Therefore, we obtain

$$\begin{aligned} F(i) &= \sum_{j=0}^{N+1} p_{ij}x_j \ = \ p_{i0}x_0 + \sum_{j=1}^{N+1} p_{ij}\{\sum_{k=1}^{j} c_k + x_0\} \\ &= x_0 + \sum_{j=1}^{N+1}\sum_{k=1}^{j} c_k p_{ij} \ = \ x_0 + \sum_{k=1}^{N+1} c_k\,\{\sum_{j=k}^{N+1} p_{ij}\}. \end{aligned}$$

Since $\sum_{j=k}^{N+1} p_{ij} = G_k(i)$ is nondecreasing in i and $c_k \geq 0$ for $k = 1, 2, \dots, N+1$, the function $F(i)$ is also nondecreasing in i. □

We first consider the discounted rewards. The method of value iteration for this model becomes (with $v_i^0 := 0$ for all i) for $n = 0, 1, \dots$

$$v_i^{n+1} = \begin{cases} \alpha\sum_j p_{0j}v_j^n & ,\ i = 0 \\ min\{\alpha\sum_j p_{ij}v_j^n,\ c_0 + \alpha\sum_j p_{0j}v_j^n\} & ,\ 1 \leq i \leq N \\ c_0 + c_1 + \alpha\sum_j p_{0j}v_j^n & ,\ i = N+1. \end{cases} \tag{8.8}$$

We assume that Assumption 8.1 (or 8.2) holds. Clearly, v_i^0 is a nondecreasing function in i. Assume v_j^n is nondecreasing in j. Then, it follows from Assumption 8.1 and (8.8) that v_i^{n+1} is also nondecreasing in i. Hence, also $v_i^\alpha = \lim_{n\to\infty} v_i^n$ is nondecreasing in i.

Theorem 8.3

Let i_* be such that $i_* = max\{i \mid \alpha\sum_j p_{ij}v_j^\alpha \leq c_0 + \alpha\sum_j p_{0j}v_j^\alpha\}$. Then, the control-limit policy f_*^∞ which replaces in the states $i > i_*$ is a discounted optimal policy.

Proof

Since v_i^α is nondecreasing in i, by Assumption 8.1, also $\alpha\sum_j p_{ij}v_j^\alpha$ is nondecreasing in i. By the definition of i_* and because v^α is the unique solution of the optimality equation

$$v_i^\alpha = \begin{cases} \alpha\sum_j p_{0j}v_j^\alpha & ,\ i = 0 \\ min\{\alpha\sum_j p_{ij}v_j^\alpha,\ c_0 + \alpha\sum_j p_{0j}v_j^\alpha\} & ,\ 1 \leq i \leq N \\ c_0 + c_1 + \alpha\sum_j p_{0j}v_j^\alpha & ,\ i = N+1 \end{cases},$$

we obtain

$$v_i^\alpha = \begin{cases} \alpha\sum_j p_{ij}v_j^\alpha & ,\ 0 \leq i \leq i_* \\ c_0 + \alpha\sum_j p_{0j}v_j^\alpha & ,\ i_* < i \leq N \\ c_0 + c_1 + \alpha\sum_j p_{0j}v_j^\alpha & ,\ i = N+1 \end{cases}$$

implying that the control-limit policy f_*^∞ is optimal. □

Theorem 8.3 implies that the next algorithm computes an optimal control-limit policy for this model. The integer k in step 2a of the algorithm is the number of nonbasic variables corresponding to the replacing actions in the states $i = 1, 2, \dots, N$. Similar to Algorithm 8.1 it can be shown that the complexity of Algorithm 8.2 is $\mathcal{O}(N^3)$.

Algorithm 8.2 *Computation of an optimal control-limit policy.*
Input: Instance of a replacement problem with increasing deterioration.
Output: An optimal deterministic control-limit policy f^∞.

```
1.  (a) Start with the basic solution corresponding to the nonreplacing actions in the states
        $i = 1, 2, \dots, N$ and to the only action in the states 0 and $N+1$.
    (b) $k := N$.
2.  if the reduced costs are nonnegative then the corresponding policy is optimal (STOP).
    otherwise
      begin
        choose the column corresponding to state $k$ as pivot column;
        execute the usual simplex transformation;
        delete the pivot column;
        $k := k - 1$
      begin
3.  if $k = 0$ then replacement in all states $i \neq 0$ is the optimal policy (STOP)
    otherwise return to step 2
```

Remark
Next, we consider the average reward. By Theorem 8.3 for each $\alpha \in (0, 1)$ there exists a control-limit policy f_α^∞ that is α-discounted optimal. Let $\{\alpha_k,\ 1, 2, \dots\}$ be any sequence of discount factors such that $\lim_{k\to\infty} \alpha_k = 1$. Since there are only a finite number of different control-limit policies, there is a subsequence of $\{\alpha_k,\ 1, 2, \dots\}$ which has with one of these control-limit policies as optimal policy. Therefore, we may assume that $f_{\alpha_k}^\infty = f_0^\infty$ for all k. Letting $k \to \infty$, we obtain for every deterministic policy f^∞, $\phi(f^\infty) = \lim_{k\to\infty}(1-\alpha_k)v_k^\alpha(f^\infty) \leq \lim_{k\to\infty}(1-\alpha_k)v_k^\alpha(f_0^\infty) = \phi(f_0^\infty)$. Therefore, also for the average reward criterion there exists a control-limit optimal policy.

8.1.3 Skip to the right model with failure

This model is a slightly different from the previous one. Let the state space $S = \{0, 1, \dots, N+1\}$, where state 0 corresponds to a new item and state $N+1$ to failure. The states $i,\ 1 \leq i \leq N$, may be interpreted as the age of the item. The system has in state i a failure probability p_i during the next period. When failure occurs in state i, which is modeled as being transferred to state $N+1$, there is an additional cost f_i. In state $N+1$ the item has to be replaced by a new one. When there is no failure in state i, the next state is state $i+1$: the system skips to the right, i.e. the age of the item increases. As in the previous section action 0 and 1 correspond to no replacement and replacement, respectively; furthermore, action 0 gives an instantaneous transition to state 0. We assume two types of cost, the cost c to buy a new item and maintenance cost c_i for an item in state i. The action sets, the cost of a new item, the maintenance costs and the transition probabilities are as follows.

$S = \{0, 1, \ldots, N+1\};\ A(0) = \{1\};\ A(0) = \{0, 1\},\ 1 \leq i \leq N;\ A(N+1) = \{0\}.$

$$1 \leq i \leq N+1: \quad p_{ij}(0) = \begin{cases} 1 - p_0 & j = 1 \\ p_0 & j = N+1 \end{cases}; \quad c_i(0) = c + c_0 + p_0 f_0$$

$$0 \leq i \leq N: \quad p_{ij}(1) = \begin{cases} 1 - p_i & j = i+1 \\ p_i & j = N+1 \end{cases}; \quad c_i(1) = c_i + p_i f_i$$

We impose the following assumptions:

(A1) $c \geq 0;\ c_i \geq 0,\ f_i \geq 0,\ 0 \leq i \leq N.$

(A2) $p_0 \leq p_1 \leq \cdots \leq p_N$, i.e. older items have greater failure probability.

(A3) $c_0 + p_0 f_0 \leq c_1 + p_1 f_1 \leq \cdots \leq c_N + p_N f_N$, i.e. the sum of the maintenance and failure costs grow with the age of the item.

Take any $k \in S$. Since $\sum_{j=k}^{N+1} p_{ij}(1) = \begin{cases} p_i & i \leq k-2 \\ 1 & i \geq k-1 \end{cases}$, this summation is, by assumption A2, nondecreasing in i. Hence, Assumption 8.1, and consequently also Assumption 8.2, is satisfied. This enables us to treat this model in a similar way as the previous one.
The method of value iteration for this model becomes (with $v_i^0 = 0$ for all i) for $n = 0, 1, \ldots$

$$v_i^{n+1} = \begin{cases} c_0 + p_0 f_0 + \alpha \sum_j p_{0j}(1) v_j^n & ,\ i = 0 \\ min\{c + c_0 + p_0 f_0 + \alpha \sum_j p_{ij}(0) v_j^n,\ c_i + p_i f_i + \alpha \sum_j p_{ij}(1) v_j^n\} & ,\ 1 \leq i \leq N \\ c + c_0 + p_0 f_0 + \alpha \sum_j p_{N+1j}(0) v_j^n & ,\ i = N+1. \end{cases} \tag{8.9}$$

Since $p_{ij}(0) = p_{0j}(1)$ for all i and j, equation (8.9) is equivalent to

$$v_i^{n+1} = \begin{cases} c_0 + p_0 f_0 + \alpha \sum_j p_{0j}(1) v_j^n & ,\ i = 0 \\ min\{c + c_0 + p_0 f_0 + \alpha \sum_j p_{0j}(1) v_j^n,\ c_i + p_i f_i + \alpha \sum_j p_{ij}(1) v_j^n\} & ,\ 1 \leq i \leq N \\ c + c_0 + p_0 f_0 + \alpha \sum_j p_{0j}(1) v_j^n & ,\ i = N+1. \end{cases} \tag{8.10}$$

Similarly to the analysis in the previous section, we can derive the following results.

Theorem 8.4

(1) v_i^n is nondecreasing in i for every $n = 0, 1, \ldots$.

(2) Let i_* be such that $i_* = max\{i \mid c_i + p_i f_i + \alpha \sum_j p_{ij}(1) v_j^\alpha \leq c + c_0 + p_0 f_0 + \alpha \sum_j p_{0j}(1) v_j^\alpha\}$. Then, the control-limit policy f_*^∞ which replaces in the states $i > i_*$ is an optimal policy.

Remarks:

1. Algorithm 8.2 is also applicable to this model.
2. Similarly as in the previous section it can be shown that for the average reward criterion there exists also a control-limit optimal policy.

8.1.4 A separable replacement problem

Suppose that the MDP has the following structure:

$S = \{1, 2, \ldots, N\};\ A(i) = \{1, 2, \ldots, M\},\ i \in S.$
$p_{ij}(a) = p_j(a),\ i, j \in S,\ a \in A(i)$, i.e. the transitions are state independent.
$r_i(a) = s_i + t(a),\ i \in S,\ a \in A(i)$, i.e. the rewards are separable.

This model is a special case of a separable MDP (see also Section 8.7).

As example, consider the problem of periodically replacing a car. When a car is replaced, it can be replaced not only by a new one, but also by a car in an arbitrary state. Let s_i be the trade-in-value of a car of state i, $t(a)$ the costs of a car of state a. Then, $r_i(a) = s_i - t(a)$ and $p_{ij}(a) = p_j(a)$, where $p_j(a)$ is the probability that a car of state a is in state j at the next decision time point.

We fist consider as optimality criterion the discounted expected rewards. The next theorem shows that a one-step look-ahead policy is optimal.

Theorem 8.5
Let a_* be such that $-t(a_*) + \alpha \sum_j p_j(a_*)s_j = max_{1 \le a \le M} \{-t(a) + \alpha \sum_j p_j(a)s_j\}$. Then, the policy f_*^∞, defined by $f_*(i) := a_*$ for every $i \in S$, is an α-discounted optimal policy.

Proof
We first show that if an action, say a_1, is optimal in state 1 this action is also optimal in the other states. Let a_1 optimal in state 1, i.e.

$$\begin{array}{rcll}
r_1(a_1) + \alpha \sum_j p_{1j}(a_1)v_j^\alpha & \ge & r_1(a) + \alpha \sum_j p_{1j}(a)v_j^\alpha, \ a \in A(1) & \Leftrightarrow \\
s_1 - t(a_1) + \alpha \sum_j p_j(a_1)v_j^\alpha & \ge & s_1 - t(a) + \alpha \sum_j p_j(a)v_j^\alpha, \ 1 \le a \le M & \Leftrightarrow \\
s_i - t(a_1) + \alpha \sum_j p_j(a_1)v_j^\alpha & \ge & s_i - t(a) + \alpha \sum_j p_j(a)v_j^\alpha, \ i \in S, \ 1 \le a \le M & \Leftrightarrow \\
r_i(a_1) + \alpha \sum_j p_{ij}(a_1)v_j^\alpha & \ge & r_i(a) + \alpha \sum_j p_{ij}(a)v_j^\alpha \ i \in S, \ a \in A(i), &
\end{array}$$

i.e. action 1 is optimal in all states. Hence, we may restrict the policies to the set of actions $\{1, 2, \dots, M\}$ and we have to decide which $a \in \{1, 2, \dots, M\}$ is the optimal one. For any choice $f^\infty \in C(D)$ with $f(i) = a$ for all $i \in S$, we have $v_i^\alpha(f^\infty) = s_i - t(a) + \alpha \sum_j p_j(a)v_j^\alpha(f^\infty) = s_i + c(a), \ i \in S$, where $c(a) := -t(a) + \alpha \sum_j p_j(a)v_j^\alpha(f^\infty)$.
A policy $f_*^\infty \in C(D)$ with $f(i) = a_*$ is α-discounted optimal if and only if

$$\begin{array}{rcll}
s_i - t(a_*) + \alpha \sum_j p_j(a_*)v_j^\alpha(f_*^\infty) & \ge & s_i - t(a) + \alpha \sum_j p_j(a)v_j^\alpha(f_*^\infty) \ \forall \, (i, a) & \Leftrightarrow \\
s_i - t(a_*) + \alpha \sum_j p_j(a_*)\{s_j + c(a_*)\} & \ge & s_i - t(a) + \alpha \sum_j p_j(a)\{s_j + c(a_*)\} \ \forall \, (i, a) & \Leftrightarrow \\
-t(a_*) + \alpha \sum_j p_j(a_*)s_j & \ge & -t(a) + \alpha \sum_j p_j(a)s_j, \ 1 \le a \le M & \Leftrightarrow \\
-t(a_*) + \alpha \sum_j p_j(a_*)s_j & = & max_{1 \le a \le M}\{-t(a) + \alpha \sum_j p_j(a)s_j\}. & \square
\end{array}$$

Corollary 8.1
Let a_0 be such that $-t(a_0) + \sum_j p_j(a_0)s_j = max_{1 \le a \le M}\{-t(a) + \sum_j p_j(a)s_j\}$. Then, the policy f_0^∞, defined by $f_0(i) := a_0$ for every $i \in S$, is a Blackwell optimal and therefore also average optimal policy.

Proof
From Theorem 8.5 it follows that f_0^∞ is an α-discounted optimal policy for all $\alpha \in [\alpha_0, 1)$ for some $\alpha_0 \in [0, 1)$. Therefore, f_0^∞ is a Blackwell optimal policy. □

8.2 Maintenance and repair problems

8.2.1 A surveillance-maintenance-replacement model

Consider a system, in use or in storage, which is deteriorating. Suppose that the deteriorating occurs stochastically and that the conditioning of the system is known only if it is inspected, which is costly.

After inspection the manager of the system has two alternatives: (1) to replace the item by a new item; (2) to keep the item and do some repair on it. Under the second alternative he must decide the extend of repairs to be made and when to make the next inspection. If inspection is put too long the system may fail in the interim, the consequence of which is an incurred cost which is a function of how long the system has been inoperative. Assume that M denotes the upper bound on the number of periods that can elapse without an inspection.

Suppose that the uninspected system evolves according to a Markov chain with states $\{0, 1, \ldots, N+1\}$. The state 0 denotes a new system and the state $N+1$ an inoperative system. Let $P = (p_{ij})$ denote the transition matrix of this Markov chain with $p_{iN+1} > 0,\ 0 \le i \le N+1$, and $p_{N+1,N+1} = 1$. Assume that when a replacement is made an instantaneous transition to state 0 takes place; when a repair is made an instantaneous transition takes place to one of the states $1, 2, \ldots, N$, depending on the extend of repairs. Replacement or repairs are only made at the time of inspections.

Since there is a failure cost depending how long the system has been inoperative we use additional states $N+1(1), N+1(2), \ldots, N+1(M)$, where $N+1(m)$ denotes the fact that the system is observed to be in state $N+1$ and has been in state $N+1$ for m uninspected periods. Hence, the state space S will consist of the states $0, 1, \ldots, N+1, N+1(1), N+1(2), \ldots, N+1(M)$.

Let c_i denote the cost of inspection when the system is in state i. Let $r_{ij},\ 1 \le i \le N+1,\ 0 \le j \le N$ denote the cost to repair the system from state i to state j. In particular, r_{i0} is the cost to replace the item by a new one. In addition we let $r_{N+1(m)j},\ m = 1, 2, \ldots, M$, denote the cost to place the system in state j from state $N+1$ when prior to discovering the system in state $N+1$, the system has been in state $N+1$ for m uninspected periods. This cost represents, in addition to the repair or replacement costs, the cost associated with undetected failure.

At each state $i = 0, 1, \ldots, N$, an action a_{lm} consists in placing the system in state $l,\ 0 \le l \le N$, and deciding to skip m $(0 \le m \le M)$ time periods before observing the system again. If the system is observed in one of the states $N+1, N+1(1), \ldots, N+1(M)$, we assume that $a_{l0},\ 0 \le l \le N$ are the only possible actions. Hence, when the system is inspected in state i and action a_{lm} is chosen there are inspection and repair costs $c_i(a_{lm}) = c_i + r_{il}$ for each i, l and m and transition probabilities $p_{ij}(a_{lm}) = p_{lj}^{(m+1)}$ for each i, j, l and m.

As optimality criterion, we are interested in minimizing the expected average cost per unit time attributed to a surveillance-replacement-maintenance policy. Let $\{X_t, Y_t,\ t = 1, 2 \ldots\}$ be the observed states and actions, respectively, and let the quantities Z_{tijm} and $\overline{Z}_{Tijm}$ be defined by

$$Z_{tijm} = \begin{cases} 1 & \text{if } X_t = i,\ Y_t = a_{jm} \\ 0 & \text{otherwise} \end{cases} \quad t = 1, 2, \ldots; \ \overline{Z}_{Tijm} = \frac{1}{T}\sum_{t=1}^{T} Z_{tijm},\ T = 1, 2, \ldots.$$

If J_T is the average cost up to time T, then for each $i, j \in S$ we have

$$J_T = \frac{\sum_{t=1}^{t(T)} \sum_{i,j,m} (c_i + r_{ij}) Z_{tijm}}{\sum_{t=1}^{t(T)} \sum_{i,j,m} (m+1) Z_{tijm} + \theta},$$

where $t(T)$ is the last inspection time less than or equal to T and $\theta := T - t(T)$. We also have

$$J_T = \frac{\sum_{i,j,m} (c_i + r_{ij}) \overline{Z}_{t(T)ijm}}{\sum_{i,j,m} (m+1) \overline{Z}_{t(T)ijm} + \frac{\theta}{T}}.$$

Notice that $t(T) \to \infty$ when $T \to \infty$. It can be shown that

$$\lim_{T\to\infty} \overline{Z}_{t(T)ijm} = \lim_{T\to\infty} \overline{Z}_{Tijm} = \lim_{T\to\infty} \frac{1}{T} \sum_{t=1}^{T} \mathbb{P}\{X_t = i,\ Y_t = a_{jm}\}$$

exists for all i, j, m (cf. Chapter 7 in [69] or Section 4.7 in [148]). Denote $\lim_{T\to\infty} \overline{Z}_{Tijm}$ by $z_{i,j,m}$. Then, we obtain for the limiting average cost

$$\lim_{T\to\infty} J_T = \frac{\sum_{i,j,m} (c_i + r_{ij}) z_{ijm}}{\sum_{i,j,m} (m+1) z_{ijm}}.$$

Notice that the underlying Markov chain is a unichain Markov chain with state $N+1$ as the only absorbing state. It also can be shown that in this case (see again Chapter 7 in [69] or Section 4.7 in [148]) that there exists a stationary optimal policy π^∞ which can be derived from the following fractional linear program:

$$min \left\{ \frac{\sum_{i,j,m}(c_i + r_{ij})x_{ijm}}{\sum_{i,j,m}(m+1)x_{ijm}} \;\middle|\; \begin{array}{lcl} \sum_{i,l,m}\{\delta_{ij} - p_{ij}(a_{lm})\}x_i(a_{lm}) & = & 0 \text{ for all } j \\ \sum_{i,l,m} x_i(a_{lm}) & = & 1 \\ x_i(a_{jm}) \geq 0 \text{ for all } i,l,m & & \end{array} \right\}. \tag{8.11}$$

Let x be an optimal solution of program (8.11). Then, π^∞ with $\pi_{ia_{jm}} := \frac{x_{ia_{jm}}}{\sum_{j,m} x_{ia_{jm}}}$ for all i,j,m is an optimal stationary policy.

The above problem involves minimizing a ratio of linear functions subject to linear constraints where the lower linear form is always positive. Any problem of this form can always be transformed to a linear programming problem. Namely, suppose we have the fractional problem

$$min \left\{ \frac{\sum_{i=1}^n c_i x_i}{\sum_{i=1}^n d_i x_i} \;\middle|\; \begin{array}{l} \sum_{i=1}^n a_{ji}x_i = 0,\ j = 1,2,\ldots,m \\ \sum_{i=1}^n x_i \quad = 1 \\ x_i \geq 0,\ i = 1,2,\ldots,n \end{array} \right\}, \tag{8.12}$$

where $\sum_{i=1}^n d_i x_i > 0$ for any feasible x. Set $z_i := \frac{x_i}{\sum_{i=1}^n d_i x_i}$, $i = 1,2,\ldots,n$ and $z_{n+1} := \frac{1}{\sum_{i=1}^n d_i x_i}$. Then we can formulate the equivalent linear progream

$$min \left\{ \sum_{i=1}^n c_i x_i \;\middle|\; \begin{array}{ll} \sum_{i=1}^n a_{ji}z_i & = 0,\ j = 1,2,\ldots,m \\ \sum_{i=1}^n d_i z_i & = 1 \\ \sum_{i=1}^n z_i - z_{n+1} & = 0 \\ z_i \geq 0,\ i = 1,2,\ldots,n+1 & \end{array} \right\}. \tag{8.13}$$

From the one-to-one relation between the fractional and linear program (remark that the reverse mapping is $x_i := \frac{z_i}{z_{n+1}}$, $1 \leq i \leq n$) it follows that if z is an optimal solution of the linear program derived from the fractional program (8.11), then π^∞ with $\pi_{ia_{jm}} := \frac{z_{ia_{jm}}}{\sum_{j,m} z_{ia_{jm}}}$ for all i,j,m is an optimal stationary policy. It can also be shown that, for each state i, $z_{ia_{jm}}$ will be strictly positive for exactly one action a_{jm}. Thus, the optimal policy is stationary and deterministic.

8.2.2 Optimal repair allocation in a series system

Consider the maintenance and repair problem of Section 1.3.5. For this model it can be shown that the optimal policy is irrespective of the repair rates μ_i, $1 \leq i \leq n$, and is the policy that assigns the repairman to the failed component with the smallest failure rate λ_i, $1 \leq i \leq n$, (*SFR policy*), i.e. the longest expected lifetime.

When a policy $f^\infty \in C(D)$ is employed, the time evaluation of the state of the system can be described as a continuous, irreducible Markov chain. Furthermore, the average expected system operation time is equal to the probability of the system being in the functioning state $\underline{1} = (1,1,\ldots,1)$ (see [236]). Under a policy $f^\infty \in C(D)$ returns to state $\underline{1}$ generate a renewal process. Employing a renewal argument it can be shown (see [275]) that maximizing the probability of the system being in state $\underline{1}$ is equivalent to minimizing the expected first passage time to state $\underline{1}$ over all initial states.

Let $T_f(x)$ denote the expected first passage time from state x to state $\underline{1}$ when policy f^∞ is employed. The above remarks are formally stated in the following lemma.

Lemma 8.2
A policy $f_*^\infty \in C(D)$ is optimal with respect to the expected average system operation time if and only if $T_{f_*}(x) \le T_f(x)$ for all $x \neq \underline{1}$ and all policies $f^\infty \in C(D)$.

The action $f(x) = i$ means that the repairman is assigned to component i in state x. We also use the notations:

$$(1_k, x) = (x_1, x_2, \dots, x_{k-1}, 1, x_{k+1}, \dots, x_n);\ C_1(x) = \{i \mid x_i = 1\};\ \lambda_1(x) = \textstyle\sum_{i \in C_1(x)}\ \lambda_i;$$
$$(0_k, x) = (x_1, x_2, \dots, x_{k-1}, 0, x_{k+1}, \dots, x_n);\ C_0(x) = \{i \mid x_i = 0\}.$$

Given policy f^∞, the Markov chain remains in state x during an exponentially distributed time with rate $\lambda_1(x) + \mu_{f(x)}$. The transition probabilities of the Markov chain satisfy

$$p_{x,(1_{f(x)},x)}(f(x)) = \frac{\mu_{f(x)}}{\lambda_1(x) + \mu_{f(x)}};\ p_{x,(0_k,x)}(f(x)) = \frac{\lambda_k}{\lambda_1(x) + \mu_{f(x)}},\ k \in C_1(x).$$

Notice that by conditioning on the first transition out state x we see that $T_f(x)$ can be obtained as unique solution of the following system of linear equations

$$\left\{ \begin{array}{lcl} T_f(x) & = & \frac{1}{\lambda_1(x)+\mu_{f(x)}}\{1 + \mu_{f(x)} T_f(1_{f(x)}, x) + \sum_{i \in C_1(x)}\ \lambda_i T_f(0_i, x)\},\ x \neq \underline{1} \\ T_f(\underline{1}) & = & 0. \end{array} \right. \tag{8.14}$$

A standard result in MDP is that the policy $f_*^\infty \in C(D)$ is optimal if and only if the associated expectd first passage times $T_{f_*}(x)$ satisfy the following functional equation

$$T_{f_*}(x) = min_{j \in C_0(x)} \Big\{ \frac{1}{\lambda_1(x) + \mu_j} \{1 + \mu_j T_{f_*}(1_j, x) + \sum_{i \in C_1(x)} \lambda_i T_{f_*}(0_i, x)\} \Big\},\ x \in S. \tag{8.15}$$

Since

$$\begin{array}{rcl} \{\lambda_1(x) + \mu_j\} T_{f_*}(x) & \le (=) & 1 + \mu_j T_{f_*}(1_j, x) + \sum_{i \in C_1(x)}\ \lambda_i T_{f_*}(0_i, x)\ \Leftrightarrow \\ \{\sum_{i=1}^n (\lambda_i + \mu_i)\} T_{f_*}(x) & \le (=) & 1 + \mu_j T_{f_*}(1_j, x) + \sum_{k \neq j}\ \mu_k T_{f_*}(x)\ + \\ & & \sum_{i \in C_1(x)}\ \lambda_i T_{f_*}(0_i, x) + \sum_{i \notin C_1(x)}\ \lambda_i T_{f_*}(, x) \end{array}$$

the optimality equation (8.15) is equivalent to the optimality equation

$$T_{f_*}(x) = min_{j \in C_0(x)} \Big\{ \frac{1}{\sum_{i=1}^n (\lambda_i + \mu_i)} \{1 + \mu_j T_{f_*}(1_j, x) + \sum_{k \neq j} \mu_k T_{f_*}(x) + \sum_{i \in C_1(x)} \lambda_i T_{f_*}(0_i, x) + \sum_{i \notin C_1(x)} \lambda_i T_{f_*}(x)\} \Big\}. \tag{8.16}$$

Because

$$\mu_j T_{f_*}(1_j, x) + \textstyle\sum_{k \neq j} \mu_k T_{f_*}(x) + \sum_{i \in C_1(x)} \lambda_i T_{f_*}(0_i, x) + \sum_{i \notin C_1(x)} \lambda_i T_{f_*}(x) =$$
$$\textstyle\sum_{k=1}^n\ \mu_k T_{f_*}(x) + \mu_j \{T_{f_*}(1_j, x) - T_{f_*}(x)\} + \sum_{i=1}^n\ \lambda_i T_{f_*}(0_i, x) + \sum_{i \notin C_1(x)} \lambda_i \{T_{f_*}(x) - T_{f_*}(0_i, x)\}$$

and

$T_{f_*}(x) = T_{f_*}(1_j, x)$ for all $j \in C_1(x)$, and $T_{f_*}(x) = T_{f_*}(0_i, x)$ for all $i \notin C_1(x)$,

equation (8.16) is equivalent to

$$T_{f_*}(x) = \frac{1}{\theta} \Big\{ 1 + \sum_{k=1}^n \mu_k T_{f_*}(x) + \sum_{i=1}^n \lambda_i T_{f_*}(0_i, x) + min_{j \in C_0(x)} \{\mu_j \{T_{f_*}(1_j, x) - T_{f_*}(0_j, x)\}\} \Big\}, \tag{8.17}$$

where $\theta := \sum_{i=1}^n (\lambda_i + \mu_i)$.

Suppose $\lambda_1 < \lambda_2 < \cdots < \lambda_n$, and let $f_*^\infty \in C(D)$ be the policy which always puts repair on the component of smallest index, i.e. the policy we are trying to prove optimal. From (8.17) it is clear that f_*^∞ is optimal if for all $i < j$ with $i, j \in C_0(x)$ we have

$$\mu_i\{T_{f_*}(1_i, x) - T_{f_*}(0_i, x)\} \leq \mu_j\{T_{f_*}(1_j, x) - T_{f_*}(0_j, x)\}. \tag{8.18}$$

Imagine solving the problem using value iteration on the optimality equation written as (8.16), i.e.

$$T^{m+1}(x) = min_{j\in C_0(x)}\Big\{\frac{1}{\theta}\{1 + \mu_j T^m(1_j, x) + \sum_{k\neq j}\mu_k T^m(x) + \sum_{i\in C_1(x)}\lambda_i T^m(0_i, x) + \sum_{i\notin C_1(x)}\lambda_i T^m(x)\}\Big\}. \tag{8.19}$$

Consider the following inductive hypothesis $H(m)$ for $m = 0, 1, \ldots$:

$\mu_i\{T^m(1_i, x) - T^m(0_i, x)\} \leq \mu_j\{T^m(1_j, x) - T^m(0_j, x)\} \leq 0$ for all x and all $i < j$ with $i, j \in C_0(x)$.

We can take $T^0(x) = 0$, so $H(0)$ is true. Assuming $H(m)$ is true for all $m = 0, 1, \ldots$, then the minimizing value is in each iteration $f_*(x) = min_{i\in C_0(x)}\{i\}$. If we succeed in proving the inductive step we have shown the structure of the optimal policy as a SFR-policy. Assume that $H(m)$ is true. Then, for all states x we have

$$T^{m+1}(x) = \frac{1}{\theta}\{1 + \mu_{f_*(x)}T^m(1_{f_*(x)}, x) + \sum_{k\neq f_*(x)}\mu_k T^m(x) + \sum_{k\in C_1(x)}\lambda_k T^m(0_k, x) + \sum_{k\notin C_1(x)}\lambda_k T^m(x)\}, \tag{8.20}$$

where $f_*(x) := min_{i\in C_0(x)}\{i\}$. For the inductive step we have to show

$$\mu_i\{T^{m+1}(1_i, x) - T^{m+1}(0_i, x)\} \leq \mu_j\{T^{m+1}(1_j, x) - T^{m+1}(0_j, x)\} \leq 0 \text{ for all } x \text{ and all } i < j \text{ with } i, j \in C_0(x). \tag{8.21}$$

Take any state x and consider first the case in which i and j are the two smallest indices of $C_0(x)$. Notice that $(1_i, x) = (1_i, 0_j, x)$, $(0_i, x) = (0_i, 0_j, x)$, $f_*(1_i, x) = j$ and $f_*(0_i, x) = i$.
Then, we can write

$$\begin{aligned} T^{m+1}(1_i, x) &= \tfrac{1}{\theta}\{1 + \mu_j T^m(1_i, 1_j, x) + \mu_i T^m(1_i, 0_j, x) + \textstyle\sum_{k\neq i,j}\mu_k T^m(1_i, 0_j, x) + \\ &\qquad \textstyle\sum_{k\in C_1(1_i,x)}\lambda_k T^m(1_i, 0_k, x) + \sum_{k\in C_0(1_i,x)}\lambda_k T^m(1_i, x)\} \end{aligned}$$

Since

$$\textstyle\sum_{k\in C_1(1_i,x)}\lambda_k T^m(1_i, 0_k, x) = \lambda_i T^m(0_i, 0_j, x) + \sum_{k\in C_1(1_i,0_j,x), k\neq i}\lambda_k T^m(1_i, 0_j, 0_k, x)$$

and

$$\textstyle\sum_{k\in C_0(1_i,x)}\lambda_k T^m(1_i, x) = \lambda_j T^m(1_i, 0_j, x) + \sum_{k\in C_0(1_i,0_j,x), k\neq j}\lambda_k T^m(1_i, 0_j, 0_k, x),$$

we obtain

$$\begin{aligned} T^{m+1}(1_i, x) &= \tfrac{1}{\theta}\{1 + \mu_j T^m(1_i, 1_j, x) + \mu_i T^m(1_i, 0_j, x) + \textstyle\sum_{k\neq i,j}\mu_k T^m(1_i, 0_j, x) + \\ &\qquad \lambda_i T^m(0_i, 0_j, x) + \lambda_j T^m(1_i, 0_j, x) + \textstyle\sum_{k\neq i,j}\lambda_k T^m(1_i, 0_j, 0_k, x)\}. \end{aligned}$$

Similarly we obtain

$$\begin{aligned} T^{m+1}(0_i, x) &= \tfrac{1}{\theta}\{1 + \mu_i T^m(1_i, 0_j, x) + \mu_j T^m(0_i, 0_j, x) + \textstyle\sum_{k\neq i,j}\mu_k T^m(0_i, 0_j, x) + \\ &\qquad \lambda_i T^m(0_i, 0_j, x) + \lambda_j T^m(0_i, 0_j, x) + \textstyle\sum_{k\neq i,j}\lambda_k T^m(0_i, 0_j, 0_k, x)\}. \\ T^{m+1}(1_j, x) &= \tfrac{1}{\theta}\{1 + \mu_i T^m(1_i, 1_j, x) + \mu_j T^m(0_i, 1_j, x) + \textstyle\sum_{k\neq i,j}\mu_k T^m(0_i, 1_j, x) + \\ &\qquad \lambda_j T^m(0_i, 0_j, x) + \lambda_i T^m(0_i, 1_j, x) + \textstyle\sum_{k\neq i,j}\lambda_k T^m(0_i, 1_j, 0_k, x)\}. \\ T^{m+1}(0_j, x) &= \tfrac{1}{\theta}\{1 + \mu_i T^m(1_i, 0_j, x) + \mu_j T^m(0_i, 0_j, x) + \textstyle\sum_{k\neq i,j}\mu_k T^m(0_i, 0_j, x) + \\ &\qquad \lambda_j T^m(0_i, 0_j, x) + \lambda_i T^m(0_i, 0_j, x) + \textstyle\sum_{k\neq i,j}\lambda_k T^m(0_i, 0_j, 0_k, x)\}. \end{aligned}$$

Now we have

$$\mu_i\{T^{m+1}(1_i,x)-T^{m+1}(0_i,x)\}-\mu_j\{T^{m+1}(1_j,x)-T^{m+1}(0_j,x)\}=$$

$$\mu_i\{\mu_jT^m(1_i,1_j,x)-\mu_iT^m(1_i,0_j,x)\}+ \tag{1}$$

$$\mu_i\{\mu_iT^m(1_i,0_j,x)-\mu_jT^m(0_i,0_j,x)\}+ \tag{2}$$

$$\mu_i\{\textstyle\sum_{k\neq i,j}\mu_kT^m(1_i,0_j,x)-\sum_{k\neq i,j}\mu_kT^m(0_i,0_j,x)\}+ \tag{3}$$

$$\mu_i\{\lambda_iT^m(0_i,0_j,x)-\lambda_iT^m(0_i,0_j,x)\}+ \tag{4}$$

$$\mu_i\{\lambda_jT^m(1_i,0_j,x)-\lambda_jT^m(0_i,0_j,x)\}+ \tag{5}$$

$$\mu_i\{\textstyle\sum_{k\neq i,j}\lambda_kT^m(1_i,0_j,0_k,x)-\sum_{k\neq i,j}\lambda_kT^m(0_i,0_j,0_k,x)\}+ \tag{6}$$

$$\mu_j\{\mu_iT^m(1_i,0_j,x)-\mu_iT^m(1_i,1_j,x)\}+ \tag{7}$$

$$\mu_j\{\mu_jT^m(0_i,0_j,x)-\mu_jT^m(0_i,1_j,x)\}+ \tag{8}$$

$$\mu_j\{\textstyle\sum_{k\neq i,j}\mu_kT^m(0_i,0_j,x)-\sum_{k\neq i,j}\mu_kT^m(0_i,1_j,x)\}+ \tag{9}$$

$$\mu_j\{\lambda_jT^m(0_i,0_j,x)-\lambda_jT^m(0_i,0_j,x)\}+ \tag{10}$$

$$\mu_j\{\lambda_iT^m(0_i,0_j,x)-\lambda_iT^m(0_i,1_j,x)\}+ \tag{11}$$

$$\mu_j\{\textstyle\sum_{k\neq i,j}\lambda_kT^m(0_i,0_j,0_k,x)-\sum_{k\neq i,j}\lambda_kT^m(0_i,1_j,0_k,x)\}. \tag{12}$$

We show that (1) + (2) + $\cdots$ + (12) ≤ 0. Therefore, we first remark that, obviously, (4) = (10) = 0. Furthermore, we mention

$$\begin{aligned}
(6)+(12) &= \textstyle\sum_{k\neq i,j}\lambda_k\big\{\mu_i\{T^m(1_i,0_j,0_k,x)-T^m(0_i,0_j,0_k,x)\}-\\
&\qquad\qquad \mu_j\{T^m(0_i,1_j,0_k,x)-T^m(0_i,0_j,0_k,x)\}\big\}\\
&\leq 0 \text{ (by the inductive hypothesis } H(m)).\\
(5)+(10) &= (\lambda_j-\lambda_i)\big\{\mu_i\{T^m(1_i,0_j,x)-T^m(0_i,0_j,x)\}-\mu_j\{T^m(0_i,1_j,x)-T^m(0_i,0_j,x)\}\big\}\\
&\leq 0 \text{ (because } \lambda_j>\lambda_i \text{ and by the inductive hypothesis } H(m)).\\
(3)+(9) &= \textstyle\sum_{k\neq i,j}\mu_k\big\{\mu_i\{T^m(1_i,0_j,x)-T^m(0_i,0_j,x)\}-\mu_j\{T^m(0_i,1_j,x)-T^m(0_i,0_j,x)\}\big\}\\
&\leq 0 \text{ (by the inductive hypothesis } H(m)).\\
(1)+(2)+(7)+(8) &= \mu_i\mu_jT^m(1_i,1_j,x)-\mu_i^2T^m(1_i,0_j,x)+\mu_i^2T^m(1_i,0_j,x)-\\
&\quad\ \mu_i\mu_jT^m(0_i,0_j,x)+\mu_i\mu_jT^m(1_i,0_j,x)-\mu_i\mu_jT^m(1_i,1_j,x)+\\
&\quad\ \mu_j^2T^m(0_i,0_j,x)-\mu_j^2T^m(0_i,1_j,x)\\
&= \mu_j\big\{\mu_j\{T^m(1_i,0_j,x)-T^m(0_i,0_j,x)\}-\mu_j\{T^m(0_i,1_j,x)-T^m(0_i,0_j,x)\}\big\}\\
&\leq 0 \text{ (by the inductive hypothesis } H(m)).
\end{aligned}$$

This is the hardest case to check. The nonnegativity of $\mu_i\{T^{m+1}(1_i,x)-T^{m+1}(0_i,x)\}$ and the other cases are left to the reader (see also Exercise 8.1).

8.2.3 Maintenance of systems composed of highly reliable components

A situation that arises in the maintenance of systems which operate continuously and possess limited repair capacity can be modeled as follows. A system of known structure is composed of n components and it is maintained by $r < n$ repairmen. Each component and the system as a whole can be in only two states, functioning or failed.

The failure and repair times for the ith component are exponentially distributed with known parameters λ_i and μ_i, respectively. At most one repairman may be assigned to a failed component and it is possible to reassign a repairman from one failed component to another instantaneously. Failures may take place even while the system is not functioning.

Optimal policies can be obtained, in principle, using methods of Markov decision theory. However, the computational difficulties are prohibitive due to the large number of states. Therefore, explicit solutions and approximations are valuable. An explicit solution for a series system maintained by a single repairman was derived in the previous Section 8.2.2.

In practice many systems are composed of highly reliable components. Therefore, we assume the failure rate for the ith component is of the form $\rho\lambda_i,\ 1 \le i \le n$. Then, for small values of ρ all components are highly reliable. We shall derive formulas for the determination of policies that are optimal for small values of ρ. These policies are called *asymptotically optimal.* For a series system with $r \ge 2$ repairmen it turns out that asymptotically optimal policies assign repairmen to failed components i with the largest expected repair times μ_i. So, the result of Section 8.2.2 does not hold in the case of more than one repairman.

Let the state of the system be given by a vector $x = (x_1, x_2, \dots, x_n)$ with $x_i = 1$ or 0 if the ith component is functioning or failed. Thus $S = \{0,1\}^n$ is the set of all possible states. The relation between the status of the components and that of the system is given by a partition of S into two sets G and B of good and bad states, where if $x \in G$ the system is functioning and if $x \in B$ the system is failed. Alternatively, this relation can be specified by the structure function g, defined on S by

$$g(x) := \begin{cases} 1 & \text{if } x \in G; \\ 0 & \text{if } x \in B. \end{cases}$$

We assume that the system is coherent, i.e.,
(1) if $x \in G$ and $y \ge x$, then $y \in G$ and if $x \in B$ and $y \le x$, then $y \in B$.
(2) for any component i there exists a state $x \in G$ such that the state $(0_i, x) \in B$.
For $x \in S$ we define $C_0(x) := \{i \mid x_i = 0\}$ and $C_1(x) := \{i \mid x_i = 1\}$. A state $x \in B$ such that $y \in G$ for any $y \ge x,\ y \ne x$ is called a *cut*; it corresponds to a minimal set of components which by failing cause a system failure. The *size* of a cut x is the cardinality $|C_0(x)|$. Let $r(x) := min\{r, |C_0(x)|\}$, i.e., $r(x)$ denotes the maximum number of components that can be under repair when the system is in state x.

The above description leads to the following formulation of a continuous Markov decision problem: The state space $S = \{x = (x_1, x_2, \dots, x_n)\}$ with $x_i \in \{0,1\},\ i = 1, 2, \dots, n$. In state x the action set $A(x) = \{a \mid a \in C_0(x);\ |a| = r(x)\}$. We exclude actions that leave repairmen idle while there are failed components, since that policies that contain that actions can not be optimal.

When the system is in state x and action $a \in A(x)$ is chosen, we have the following transitions and reward:
(1) transition to state $(1_i, x)$ with rate μ_i;
(2) transition to state $(0_i, x)$ with rate $\rho\lambda_i$;
(3) reward rate $g(x)$.

Note that under any deterministic policy the status of all components can be described by a continuous time Markov chain $\{X(t) = (X_1(t), X_2(t), \dots, X_n(t)\}$, where $X_i(t) = x_i$ if the ith component is x_i at time t. It is known (see [236], p. 114) that optimal deterministic policies exist, both for the total discounted reward and the average reward criterion. Notice that the total discounted reward and the average reward are the expected total discounted time and the expected average time, respectively, that the system is in good states.

Let $b_f(x)$ be the expected total discounted time that the system is in bad states, when the initial state is x and policy f^∞ is employed. For notational simplicity we suppress the dependency of $b_f(x)$ on the discount rate $\beta \in (0, \infty)$ and the parameter ρ. Similarly, denote by $g_f(x)$ be the expected total discounted time that the system is in good states. Since $b_f(x) + g_f(x)$ is the expected total discounted time, we obtain $b_f(x) + g_f(x) = \int_0^\infty e^{-\beta t} dt = \frac{1}{\beta}$ for all $x \in S$ and $f^\infty \in C(D)$.

It is known (see [236] p. 120) that for any $f \in C(D)$ the corresponding values $b_f(x)$, $x \in S$, can be obtained as the unique solution of the following system of linear equations

$$b_f(x) = \frac{1}{\mu\big(f(x)\big) + \rho\lambda(x) + \beta} \cdot \Big\{\{1 - g(x)\} + \sum_{j \in f(x)} \mu_j b_f(1_j, x) + \rho \cdot \sum_{j \in C_1(x)} \lambda_j b_f(0_j, x)\Big\},\ x \in S, \quad (8.22)$$

where $\mu\big(f(x)\big) := \sum_{j \in f(x)} \mu_j$ and $\lambda(x) := \sum_{j \in C_1(x)} \lambda_j$.

Lemma 8.3

For any $x \in S$, $\beta \in (0, \infty)$ and $f^\infty \in C(D)$, there exists, for ρ sufficiently small, a power series expansion of $b_f(x)$ of the form $b_f(x) = \sum_{k=0}^\infty \rho^k b_f^{(k)}(x)$.

Proof

The system (8.22) can be written as

$$\{\mu\big(f(x)\big) + \beta\} \cdot b_f(x) = \{1 - g(x)\} + \textstyle\sum_{j \in f(x)} \mu_j b_f(1_j, x) + \rho \cdot \sum_{j \in C_1(x)} \lambda_j \{b_f(0_j, x) - b_f(x)\},\ x \in S.$$

which yields the system

$$b_f(x) = \frac{1}{\mu\big(f(x)\big) + \beta} \cdot \Big\{\{1 - g(x)\} + \sum_{j \in f(x)} \mu_j b_f(1_j, x) + \rho \cdot \sum_{j \in C_1(x)} \lambda_j \{b_f(0_j, x) - b_f(x)\}\Big\},\ x \in S. \quad (8.23)$$

In matrix form (8.23) can be written as

$$b_f = a(f) + C(f)b_f + \rho \cdot D(f)b_f. \quad (8.24)$$

Under an appropriate numbering of the states (e.g. if $|C_0(x)| > |C_0(y)|$ then y has a higher label than x), then the matrix $C(f)$ is upper triangular and all elements are less than 1. Hence, the inverse matrix $\{I - C(f)\}^{-1}$ exists. Thus, we have $b_f = \{I - C(f)\}^{-1} a(f) + \rho \cdot \{I - C(f)\}^{-1} D(f) b_f$, or in more compact form,

$$b_f = q(f) + \rho \cdot Q(f) b_f. \quad (8.25)$$

By iterating (8.25), we obtain for any $m \geq 0$,

$$b_f = \sum_{k=0}^m \rho^k \{Q(f)\}^k q_f + \rho^{m+1} \{Q(f)\}^{m+1} b_f. \quad (8.26)$$

For ρ sufficiently small, i.e. for $0 < \rho < \frac{1}{\|Q(f)\|}$, we have $b_f = \sum_{k=0}^\infty \rho^k \{Q(f)\}^k q_f$. Hence, for ρ sufficiently small, there exists a power series expansion of $b_f(x)$ of the form $b_f(x) = \sum_{k=0}^\infty \rho^k b_f^{(k)}(x)$. □

The next corollary provides a method for computing the coefficients $b_f^{(k)}(x)$ recursively for increasing $|C_0(x)|$.

Corollary 8.2

For any $x \in S$, $\beta \in (0, \infty)$ and $f^\infty \in C(D)$, the numbers $b_f^{(k)}(x)$ can be computed recursively for increasing $|C_0(x)|$ by the following equations:

$b_f^{(0)}(x) = 0$ *for* $x = e = (1, 1, \dots, 1)$;

$b_f^{(0)}(x) = \frac{1}{\mu(f(x)) + \beta} \cdot \Big\{\{1 - g(x)\} + \sum_{j \in f(x)} \mu_j b_f^{(0)}(1_j, x)\Big\}$ *for* $x \in S$ *with* $|C_0(x)| = 1, 2, \dots, n$;

$b_f^{(k+1)}(x) = \frac{1}{\mu(f(x)) + \beta} \cdot \Big\{\sum_{j \in f(x)} \mu_j b_f^{(k+1)}(1_j, x) + \sum_{j \in C_1(x)} \lambda_j \{b_f^{(k)}(0_j, x) - b_f^{(k)}(x)\}\Big\}$ *for* $x \in S$ *with* $|C_0(x)| = 0, 1, \dots, n$ *and for* $k = 0, 1, \dots$.

Proof

For notational simplicity we suppress the dependency on f. Since we have the relation $b^{(k)} = Q^k q$, we obtain $b^{(0)} = q = \{I - C\}^{-1}a$, i.e. $b^{(0)}$ is the unique solution of the linear system $(I-C)z = a$. Since $I-C$ is an upper triangular matrix with diagonal elements 1, we can compute $b^{(0)}(x)$ recursively for increasing $|C_0(x)|$. For $|C_0(x)| = 0$, we have $x = e$ and consequently,

$$b^{(0)}(x) = b^{(0)}(e) = a(e) = \frac{1}{\mu\big(f(e)\big)+\beta} \cdot \Big\{\{1-g(e)\}\Big\} = \frac{1}{\beta} \cdot 0 = 0.$$

Since $a(x) = \frac{1}{\mu\big(f(x)\big)+\beta} \cdot \{1-g(x)\}$ and because the matrix C has in the row of state x nonzero elements only for columns of states $y = (1_j, x)$ for every $j \in f(x)$ and these elements are equal to $\frac{1}{\mu\big(f(x)\big)+\beta} \cdot \mu_j$, we obtain

$$b^{(0)}(x) = \frac{1}{\mu\big(f(x)\big)+\beta} \cdot \Big\{\{1-g(x)\} + \textstyle\sum_{j\in f(x)} \mu_j b_f^{(0)}(1_j,x)\Big\} \text{ for } x \in S \text{ with } |C_0(x)| = 1, 2, \ldots, n.$$

Note that $b^{(k+1)}$ is the unique solution of the system $(I-C)z = Db^{(k)}$. Since the matrix D has in the row of state x nonzero elements only for columns of states $y = (0_j, x)$ and $y = x$ for every $j \in C_1(x)$, and these elements are equal to $\frac{1}{\mu\big(f(x)\big)+\beta} \cdot \lambda_j$ and $-\frac{1}{\mu\big(f(x)\big)+\beta} \cdot \lambda_j$, respectively, we obtain

$$b_f^{(k+1)}(x) = \frac{1}{\mu\big(f(x)\big)+\beta} \cdot \Big\{ \textstyle\sum_{j\in f(x)} \mu_j b_f^{(k+1)}(1_j,x) + \sum_{j\in C_1(x)} \lambda_j \{b_f^{(k)}(0_j,x) - b_f^{(k)}(x)\}\Big\} \text{ for } x \in S \text{ with}$$

$|C_0(x)| = 0, 1, \ldots, n.$ □

Example 8.1

Consider a system with $n = 3$ components and $r = 2$ repairmen. Let $\lambda_1 = 1,\ \lambda_2 = 2,\ \lambda_3 = 3,\ \mu_1 = 2,\ \mu_2 = 1,\ \mu_3 = 2$ and $\beta = 1$. Take $g(x) = 1$ if $x_1 = 1$ and $x_2 + x_3 \geq 1$. The numbering of the states, $g(x)$, and the actions $f(x)$, the values $\big(f(x)\big)$ and $\lambda(x)$ for every $x \in S$ are presented in the following table.

state	x	$g(x)$	$f(x)$	$\mu\big(f(x)\big)$	$\lambda(x)$
1	(0,0,0)	0	$\{1,2\}$	3	0
2	(0,0,1)	0	$\{1,2\}$	3	3
3	(0,1,0)	0	$\{1,3\}$	3	2
4	(1,0,0)	0	$\{2,3\}$	2	1
5	(0,1,1)	0	$\{1\}$	2	5
6	(1,0,1)	1	$\{2\}$	1	4
7	(1,1,0)	1	$\{3\}$	1	3
8	(1,1,1)	1	$\emptyset$	0	6

The system (8.23) becomes:

$$b_f(1) = \tfrac{1}{4} \cdot \{1 + 2b_f(4) + b_f(3) + \rho \cdot 0\}$$

$$b_f(2) = \tfrac{1}{4} \cdot \{1 + 2b_f(6) + b_f(5) + \rho \cdot \big\{3 \cdot \{b_f(1) - b_f(2)\}\big\}$$

$$b_f(3) = \tfrac{1}{4} \cdot \{1 + 2b_f(7) + b_f(5) + \rho \cdot \{2 \cdot \{b_f(1) - b_f(3)\}\}$$

$$b_f(4) = \tfrac{1}{3} \cdot \Big\{1 + b_f(7) + b_f(6) + \rho \cdot \{1 \cdot \{b_f(1) - b_f(4)\}\}\Big\}$$

$$b_f(5) = \tfrac{1}{3} \cdot \Big\{1 + 2b_f(8) + \rho \cdot \{2 \cdot \{b_f(2) - b_f(5)\} + 3 \cdot \{b_f(3) - b_f(5)\}\}\Big\}$$

$$b_f(6) = \tfrac{1}{2} \cdot \Big\{0 + b_f(8) + \rho\{1 \cdot \{b_f(2) - b_f(6)\} + 3 \cdot \{b_f(4) - b_f(6)\}\}\Big\}$$

$$b_f(7) = \tfrac{1}{2} \cdot \Big\{0 + b_f(8) + \rho \cdot \{1 \cdot \{b_f(3) - b_f(7)\} + 2 \cdot \{b_f(4) - b_f(7)\}\}\Big\}$$

$$b_f(8) = \tfrac{1}{1} \cdot \Big\{0 + \rho \cdot \{1 \cdot \{b_f(5) - b_f(8)\} + 2 \cdot \{b_f(6) - b_f(8)\} + 3 \cdot \{b_f(7) - b_f(8)\}\}\Big\}$$

Hence, $a(f) = (\frac{1}{4}, \frac{1}{4}, \frac{1}{4}, \frac{1}{3}, \frac{1}{3}, 0, 0, 0)^T$ and

$$C(f) = \begin{pmatrix} 0 & 0 & \frac{1}{4} & \frac{1}{2} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{4} & \frac{1}{2} & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{4} & 0 & \frac{1}{2} & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{1}{3} & \frac{1}{3} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{2}{3} \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{2} \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{2} \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}; \; D(f) = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \frac{3}{4} & -\frac{3}{4} & 0 & 0 & 0 & 0 & 0 & 0 \\ \frac{1}{2} & 0 & -\frac{1}{2} & 0 & 0 & 0 & 0 & 0 \\ \frac{1}{3} & 0 & 0 & -\frac{1}{3} & 0 & 0 & 0 & 0 \\ 0 & \frac{2}{3} & 1 & 0 & -\frac{5}{3} & 0 & 0 & 0 \\ 0 & \frac{1}{2} & 0 & \frac{3}{2} & 0 & -2 & 0 & 0 \\ 0 & 0 & \frac{1}{2} & 1 & 0 & 0 & -\frac{2}{3} & 0 \\ 0 & 0 & 0 & 0 & 1 & 2 & 3 & -6 \end{pmatrix}.$$

$$\text{The inverse } \{I - C(f)\}^{-1} = \begin{pmatrix} 1 & 0 & \frac{1}{4} & \frac{1}{2} & \frac{1}{16} & \frac{1}{6} & \frac{7}{24} & \frac{13}{48} \\ 0 & 1 & 0 & 0 & \frac{1}{4} & \frac{1}{2} & 0 & \frac{5}{12} \\ 0 & 0 & 1 & 0 & \frac{1}{4} & 0 & \frac{1}{2} & \frac{5}{12} \\ 0 & 0 & 0 & 1 & 0 & \frac{1}{3} & \frac{1}{3} & \frac{1}{3} \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & \frac{2}{3} \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & \frac{1}{2} \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & \frac{1}{2} \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}.$$

Now, we can compute $q(f) = \{I - C(f)\}^{-1}a(f) = (\frac{1}{2}, \frac{1}{3}, \frac{1}{3}, \frac{1}{3}, \frac{1}{3}, 0, 0, 0, 0)^T$ and

$$Q(f) = \{I - C(f)\}^{-1}D(f) = \begin{pmatrix} \frac{7}{24} & \frac{1}{8} & \frac{1}{12} & \frac{3}{8} & \frac{1}{6} & \frac{5}{24} & \frac{3}{8} & -\frac{13}{8} \\ \frac{3}{4} & -\frac{1}{3} & \frac{1}{4} & \frac{3}{4} & 0 & -\frac{1}{6} & \frac{5}{4} & -\frac{5}{2} \\ \frac{1}{2} & \frac{1}{6} & 0 & \frac{1}{2} & 0 & \frac{5}{6} & \frac{1}{2} & -\frac{5}{2} \\ \frac{1}{3} & \frac{1}{6} & \frac{1}{6} & \frac{1}{2} & \frac{1}{3} & 0 & \frac{1}{2} & -2 \\ 0 & \frac{2}{3} & 1 & 0 & -1 & \frac{4}{3} & 2 & 4 \\ 0 & \frac{1}{2} & 0 & \frac{3}{2} & \frac{1}{2} & -1 & \frac{3}{2} & -3 \\ 0 & 0 & \frac{1}{2} & 1 & \frac{1}{2} & 1 & 0 & -3 \\ 0 & 0 & 0 & 0 & 1 & 2 & 3 & -6 \end{pmatrix}.$$

For the computation of $b_f^{(0)}(x),\ x \in S$, we obtain:

$$\begin{array}{rcl} b_f^0(8) & = & 0. \\ b_f^0(7) & = & \frac{1}{2} \cdot \{0 + b_f^0(8)\} = 0. \\ b_f^0(6) & = & \frac{1}{2} \cdot \{0 + b_f^0(8)\} = 0. \\ b_f^0(5) & = & \frac{1}{3} \cdot \{1 + 2b_f^0(8)\} = \frac{1}{3}. \\ b_f^0(4) & = & \frac{1}{3} \cdot \{1 + b_f^0(7) + b_f^0(6)\} = \frac{1}{3}. \\ b_f^0(3) & = & \frac{1}{4} \cdot \{1 + 2b_f^0(7) + b_f^0(5)\} = \frac{1}{3}. \\ b_f^0(2) & = & \frac{1}{4} \cdot \{1 + 2b_f^0(6) + b_f^0(5)\} = \frac{1}{3}. \\ b_f^0(1) & = & \frac{1}{4} \cdot \{1 + 2b_f^0(4) + b_f^0(3)\} = \frac{1}{2}. \end{array}$$

The computation of $b_f^{(1)}(x),\ x \in S$, is as follows:

$$\begin{array}{rcl} b_f^1(8) & = & \frac{1}{1} \cdot \big\{1 \cdot \{b_f^0(5) - b_f^0(8)\} + 2 \cdot \{b_f^0(6) - b_f^0(8)\} + 3 \cdot \{b_f^0(7) - b_f^0(8)\}\big\} = \frac{1}{3}. \\ b_f^1(7) & = & \frac{1}{2} \cdot \big\{b_f^{(1)}(8) + 1 \cdot \{b_f^0(3) - b_f^0(7)\} + 2 \cdot \{b_f^0(4) - b_f^0(7)\}\big\} = \frac{2}{3}. \\ b_f^1(6) & = & \frac{1}{2} \cdot \big\{b_f^{(1)}(8) + 1 \cdot \{b_f^0(2) - b_f^0(6)\} + 3 \cdot \{b_f^0(4) - b_f^0(6)\}\big\} = \frac{5}{6}. \\ b_f^1(5) & = & \frac{1}{3} \cdot \big\{2b_f^{(1)}(8) + 2 \cdot \{b_f^0(2) - b_f^0(5)\} + 3 \cdot \{b_f^0(3) - b_f^0(5)\}\big\} = \frac{2}{9}. \end{array}$$

$$\begin{array}{lcl} b_f^1(4) & = & \frac{1}{3}\cdot\{b_f^{(1)}(7)+b_f^{(1)}(6)+1\cdot\{b_f^0(1)-b_f^0(4)\}\}=\frac{5}{9}. \\ b_f^1(3) & = & \frac{1}{4}\cdot\{2b_f^{(1)}(7)+b_f^{(1)}(5)+2\cdot\{b_f^0(1)-b_f^0(3)\}\}=\frac{17}{36}. \\ b_f^1(2) & = & \frac{1}{4}\cdot\{2b_f^{(1)}(6)+b_f^{(1)}(5)+3\cdot\{b_f^0(1)-b_f^0(2)\}\}=\frac{43}{72}. \\ b_f^1(1) & = & \frac{1}{4}\cdot\{2b_f^{(1)}(4)+b_f^{(1)}(3)\}=\frac{57}{144}. \end{array}$$

We next determine the leading coefficient $b_f^{(l)}(x)$ of the power series $b_f(x)=\sum_{k=0}^\infty \rho^k b_f^{(k)}(x)$, i.e. $b_f^{(l)}(x)\neq 0$ and $b_f^{(k)}(x)=0$ for $k=0,1,\dots,l-1$. It turns out that the order of the leading coefficients is determined by the structure of the system. We first need to define the following quantities:

$$\begin{array}{lcl} m(g) & := & min\big\{|C_0(x)| \bigm| x\in B\big\}; \\ B_{m(g)} & := & \big\{x\in B \bigm| |C_0(x)|=m(g)\big\}; \\ I(x) & := & min\big\{|C_0(y)|-|C_0(x)| \bigm| y\le x,\ y\in B\big\},\ x\in S. \end{array}$$

In this terminology of coherent structure, $m(g)$ is the size of a cut of minimum size, $B_{m(g)}$ is the set of all such states and $I(x)$ is the minimum number of components that must fail, when the system is in state x, in order to cause a system fail. The next lemma summarizes some properties of $I(x)$ that are easily verifiable from its definition and the fact that g is a coherent structure.

Lemma 8.4

For any state x the following properties hold:
(1) $I(e)=m(g)\ge I(x)$
(2) $I(0_i,x)\ge I(x)-1$ for every $i\in C_1(x)$.
(3) If $g(x)=1$, then $I(x)\ge 1$.
(4) $g(x)=0$ if and only if $I(x)=0$.
(5) If $y\in B_{m(g)}$ and $j_1,j_2,\dots,j_k\in C_0(y)$, then $I(x)=k$ for $x=(1_{j_1},1_{j_2},\dots,1_{j_k},y)$.

Lemma 8.5

For any state x and any policy $f^\infty\in C(D)$, we have $b_f^{(k)}(x)=0$ for $k=0,1,\dots,I(x)-1$.

Proof

Take any policy $f^\infty\in C(D)$. We will prove this lemma by induction on k.
For $k=0$, we must show that $b_f^{(0)}(x)=0$ for all x with $I(x)\ge 1$.
To prove this statement, we apply induction on $|C_0(x)|$.

If $|C_0(x)|=0$, then $x=e$, and by Corollary 8.2, $b_f^{(0)}(e)=0$.

Assume that $b_f^{(0)}(x)=0$ for all x with $I(x)\ge 1$ and with $|C_0(x)|=p\ge 1$, and consider any state y with $I(y)\ge 1$ and $|C_0(y)|=p+1$. Since $I(y)\ge 1$, we have $g(y)=1$. Take any $j\in f(y)$. Then, we obtain $|C_0(1_j,y)|=p$ and $I(1_j,y)\ge 1$. Hence, by the induction hypothesis, $b_f^{(0)}(1_j,y)=0$. Therefore, by Corollary 8.2, $b_f^{(0)}(x)=\frac{1}{\mu\left(f(y)\right)+\beta}\cdot\big\{\{1-g(y)\}+\sum_{j\in f(y)}\mu_j b_f^{(0)}(1_j,y)\big\}=0$.

Assume that the lemma is true for some $k\ge 0$, i.e. $b_f^{(k)}(x)=0$ for all x with $I(x)\ge k+1$. We have to show that $b_f^{(k+1)}(x)=0$ for all x with $I(x)\ge k+2$. Again, we apply induction on $|C_0(x)|$.

If $|C_0(x)|=0$, then $x=e$. By Corollary 8.2, $b_f^{(k+1)}(x)=\frac{1}{\beta}\cdot\Big\{\sum_{j=1}^n\lambda_j\{b_f^{(k)}(0_j,e)-b_f^{(k)}(e)\}\Big\}$.

Since we may assume that $I(e)\ge k+2$, the induction hypothesis provides $b_f^{(k)}(e)=0$. Because $I(0_j,e)\ge I(e)-1\ge k+1$, the induction hypothesis gives $b_f^{(k)}(0_j,e)=0$ for all j, implying $b_f^{(k+1)}(e)=0$.

Assume that $b_f^{(k+1)}(x)=0$ for all x with $I(x)\ge k+2$ and $|C_0(x)|=p\ge 1$, and consider any state y

with $I(y) \geq k+2$ and $|C_0(y)| = p+1$. Take any $j \in f(y)$. Then, we obtain $|C_0(1_j, y)| = p$ and $I(1_j, y) \geq I(y) \geq k+2$. Hence, by the induction hypothesis, $b_f^{(k+1)}(1_j, y) = 0$ and consequently, $\sum_{j \in f(y)} \mu_j b_f^{(k+1)}(1_j, y) = 0$. For $j \in C_1(y)$, we have $I(0_j, y) \geq I(y) - 1 \geq k+1$. Thus, the induction hypothesis implies $b_f^{(k)}(0_j, y) = 0$, and consequently, $\sum_{j \in C_1(y)} \lambda_j b_f^{(k)}(0_j, y) = 0$. Hence, by Corollary 8.2, $b_f^{(k+1)}(x) = \frac{1}{\mu\big(f(x)\big)+\beta} \cdot \Big\{ \sum_{j \in f(x)} \mu_j b_f^{(k+1)}(1_j, x) + \sum_{j \in C_1(x)} \lambda_j \{b_f^{(k)}(0_j, x) - b_f^{(k)}(x)\} \Big\} = 0$. □

From Corollary 8.2 and Lemma 8.5 it follows that the leading coefficient $b_f^{(I(x))}(x)$ can be computed recursively as follows:

(1) $b_f(0)(x) = 0$ for $x = e = (1, 1, \dots, 1)$.

(2) For all states x with $I(x) = 0$: $b_f^{(I(x))}(x) = \frac{1}{\mu\big(f(x)\big)+\beta} \cdot \{1 + \sum_{j \in f(x)} \mu_j b_f^{(0)}(1_j, x)\}$.

(3) For all states x with $I(x) \geq 0$:

$$b_f^{(I(x))}(x) = \frac{1}{\mu\big(f(x)\big)+\beta} \cdot \{\textstyle\sum_{j \in f(x)} \mu_j b_f^{(I(x))}(1_j, x) + \sum_{j \in C_1(x)} \lambda_j b_f^{(I(x)-1))}(0_j, x)\}.$$

Example 8.1 (continued)

$m(g) = 1;\ B_{m(g)} = \{(0,1,1)\};\ I(1) = I(2) = I(3) = I(4) = I(5) = 0,\ I(6) = I(7) = I(8) = 1.$
Since $I(6) = I(7) = I(8) = 1$, we have $b_f^{(0)}(6) = b_f^{(0)}(7) = b_f^{(0)}(8) = 0$.
The computation of the leading coefficients is as follows:
$b_f^{(I(5)}(5) = b_f^{(0)}(5) = \frac{1}{3} \cdot \{1 + b_f^{(0)}(8)\} = \frac{1}{3}$.
$b_f^{(I(4)}(4) = b_f^{(0)}(4) = \frac{1}{3} \cdot \{1 + b_f^{(0)}(7) + b_f^{(0)}(6)\} = \frac{1}{3}$.
$b_f^{(I(3)}(3) = b_f^{(0)}(3) = \frac{1}{4} \cdot \{1 + 2b_f^{(0)}(7) + b_f^{(0)}(5)\} = \frac{1}{3}$.
$b_f^{(I(2)}(2) = b_f^{(0)}(2) = \frac{1}{4} \cdot \{1 + 2b_f^{(0)}(6) + b_f^{(0)}(5)\} = \frac{1}{3}$.
$b_f^{(I(1)}(1) = b_f^{(0)}(1) = \frac{1}{4} \cdot \{1 + 2b_f^{(0)}(4) + b_f^{(0)}(3)\} = \frac{1}{2}$.
$b_f^{(I(8)}(8) = b_f^{(1)}(8) = \frac{1}{1} \cdot \{b_f^{(0)}(5) + 2b_f^{(0)}(6) + b_f^{(0)}(3)\} = \frac{1}{3}$.
$b_f^{(I(7)}(8) = b_f^{(1)}(7) = \frac{1}{2} \cdot \{b_f^{(1)}(8) + b_f^{(0)}(3) + 2b_f^{(0)}(4)\} = \frac{2}{3}$.
$b_f^{(I(6)}(6) = b_f^{(1)}(6) = \frac{1}{2} \cdot \{b_f^{(1)}(8) + b_f^{(0)}(2) + 3b_f^{(0)}(4)\} = \frac{5}{6}$.

Remarks

1. A policy $f_*^\infty \in C(D)$ minimizes the expected total discounted nonfunctioning time of the system for small values of ρ if and only if $b_{f_*}^{(I(x))}(x) = min\,\{b_f^{(I(x))}(x) \mid f^\infty \in C(D)\}$.
2. From the above formulas it follows by induction on $I(x)$ that $b_f^{(I(x))}(x) > 0$ for all x.
3. For any x with $I(x) = 0$ it follows from the formula $b_f^{(I(x))}(x) = \frac{1}{\mu\big(f(x)\big)+\beta} \cdot \{1 + \sum_{j \in f(x)} \mu_j b_f^{(0)}(1_j, x)\}$, that $b_f^{(I(x))}(x)$ is equal to the expected discounted first passage time from state x to state e in the absence of failures. We have the following partial characterization of an asymptotically optimal policy. If a policy is asymptotically optimal then it must assign repairmen to failed components in such a way that the expected discounted time that the system spends in failed states during the first passage time from any state $x \in B$ to state e is minimized.

We now turn to the problem of maximizing the average time the system is functioning. Let B_f denote the average time that the system spends in bad states, when policy $f^\infty \in C(D)$ is employed. Since for any policy f^∞ the continuous Markov chain $\{X(t) = \big(X_1(t), X_2(t), \dots, X_n(t)\big)\}$ is ergodic, B_f is independent of the initial state. Similarly, denote by G_f the average time that the system spends in good states, under policy f^∞. Obviously, $B_f + G_f = 1$ for all $f^\infty \in C(D)$. It is known (see [236] p. 126) that for any $f^\infty \in C(D)$ the value B_f, can be obtained as the unique solution of the following system of linear equations

$$\begin{aligned} H_f(x) &= \frac{1}{\mu\big(f(x)\big) + \rho\lambda(x)} \cdot \big\{\{1-g(x)\} + \sum_{j\in f(x)} \mu_j H_f(1_j, x) + \rho \cdot \sum_{j \in C_1(x)} \lambda_j H_f(0_j, x) - B_f\big\},\ x \in S. \\ H_f(e) &= 0. \end{aligned} \tag{8.27}$$

Lemma 8.6

For any $x \in S,\ x/not = e$ and any $f^\infty \in C(D)$, there exists, for ρ sufficiently small, power series expansions of B_f and $H_f(x)$ of the form $B_f = \sum_{k=1}^\infty \rho^k B_f^{(k)}$ and $H_f(x) = \sum_{k=0}^\infty \rho^k H_f^{(k)}(x)$.

Proof

The system (8.27) can be written as
$\mu\big(f(x)\big) \cdot H_f(x) = \{1-g(x)\} + \sum_{j\in f(x)} \mu_j H_f(1_j,x) + \rho \cdot \sum_{j\in C_1(x)} \lambda_j\{H_f(0_j,x) - H_f(x)\} - B_f$,
which provides for all $x \in S$ the equation
$H_f(x) = \frac{1}{\mu(f(x))} \cdot \big\{\{1-g(x)\} + \sum_{j\in f(x)} \mu_j H_f(1_j,x) + \rho \cdot \sum_{j\in C_1(x)} \lambda_j\{H_f(0_j,x) - H_f(x)\} - B_f\big\}$.
Similarly as in the proof of Lemma 8.3 we obtain $H_f(x) = \sum_{k=0}^\infty \rho^k H_f^{(k)}(x)$ for some $H_f^{(k)}(x),\ k = 0,1,\dots$. Take $x = e$. Then, because $H_f(e) = 0$ and $g(e) = 1$, we have $B_f = \rho \cdot \sum_{j=1}^n \lambda_j H_f(0_j, e)$. Since we have shown $H_f(0_j,e) = \sum_{k=0}^\infty \rho^k H_f^{(k)}(0_j,e)$, we obtain $B_f = \sum_{k=1}^\infty \rho^k B_f^{(k)}$ with $B_f^{(k)} = \sum_{j=1}^n \lambda_j H_f^{(k-1)}(0_j,e)$.

□

Corollary 8.3

For any $x \in S$ and any $f^\infty \in C(D)$, the numbers $H_f^{(k)}(x)$ can be computed recursively by increasing $|C_0(x)|$ by the following equations:
$H_f^{(0)}(e) = 0$;
$H_f^{(0)}(x) = \frac{1}{\mu(f(x))} \cdot \big\{\{1-g(x)\} + \sum_{j\in f(x)} \mu_j H^{(0)}(1_j,x)\big\}$ *for $x \in S,\ x \neq e$ for increasing $|C_0(x)|$;*
$B_f^{(k+1)} = \sum_{j=1}^n \lambda_j H_f^{(k)}(0_j,e)$;
$H_f^{(k+1)}(e) = 0$;
$H_f^{(k+1)}(x) = \frac{1}{\mu(f(x))} \cdot \big\{-B_f^{(k+1)} + \sum_{j\in f(x)} \mu_j H_f^{(k+1)}(1_j,x) + \sum_{j\in C_1(x)} \lambda_j\{H_f^{(k)}(0_j,x) - H_f^{(k)}(x)\}\big\}$ *for $x \in S,\ x \neq e$ for increasing $|C_0(x)|$.*

Proof

The formula for $B_f^{(k)}$ was derived in the proof of Lemma 8.6. The proof of the formula for $H_f^{(k)}(x)$ is similar to the proof of Corollary 8.2.

□

Example 8.1 (continued)

For the computation of $H_f^{(0)}(x),\ x \in S$, and $B_f^{(1)}$ we obtain:

$$\begin{aligned} H_f^{(0)}(8) &= 0; \\ H_f^{(0)}(7) &= \tfrac{1}{1} \cdot \{0 + H_f^{(0)}(8)\} = 0; \\ H_f^{(0)}(6) &= \tfrac{1}{1} \cdot \{0 + H_f^{(0)}(8)\} = 0; \\ H_f^{(0)}(5) &= \tfrac{1}{2} \cdot \{1 + 2H_f^{(0)}(8)\} = \tfrac{1}{2}; \\ H_f^{(0)}(4) &= \tfrac{1}{2} \cdot \{1 + H_f^{(0)}(7) + H_f^{(0)}(6)\} = \tfrac{1}{2}; \\ H_f^{(0)}(3) &= \tfrac{1}{3} \cdot \{1 + 2H_f^{(0)}(7) + H_f^{(0)}(5)\} = \tfrac{1}{2}; \\ H_f^{(0)}(2) &= \tfrac{1}{3} \cdot \{1 + 2H_f^{(0)}(6) + H_f^{(0)}(5)\} = \tfrac{1}{2}; \end{aligned}$$

$$\begin{array}{lcl}
H_f^{(0)}(1) & = & \frac{1}{3}\cdot\{1+2H_f^{(0)}(4)+H_f^{(0)}(3)\}=\frac{5}{6}; \\
B_f^{(1)} & = & \lambda_1 H_f^{(0)}(5)+\lambda_2 H_f^{(0)}(6)+\lambda_3 H_f^{(0)}(7)=\frac{1}{2}.
\end{array}$$

The computation of $H_f^{(1)}(x),\ x\in S$, and $B_f^{(2)}$ is as follows:

$$\begin{array}{lcl}
H_f^{(1)}(8) & = & 0; \\
H_f^{(1)}(7) & = & \frac{1}{1}\cdot\big\{-B_f^{(1)}+H_f^{(1)}(8)+1\cdot\{H_f^{(0)}(3)-H_f^{(0)}(7)\}+2\cdot\{H_f^{(0)}(4)-H_f^{(0)}(7)\}\big\}=1. \\
H_f^{(1)}(6) & = & \frac{1}{1}\cdot\big\{-B_f^{(1)}+H_f^{(1)}(8)+1\cdot\{H_f^{(0)}(2)-H_f^{(0)}(6)\}+3\cdot\{H_f^{(0)}(4)-H_f^{(0)}(6)\}\big\}=\frac{3}{2}. \\
H_f^{(1)}(5) & = & \frac{1}{2}\cdot\big\{-B_f^{(1)}+2H_f^{(1)}(8)+2\cdot\{H_f^{(0)}(2)-H_f^{(0)}(5)\}+3\cdot\{H_f^{(0)}(3)-H_f^{(0)}(5)\}\big\}=-\frac{1}{4}. \\
H_f^{(1)}(4) & = & \frac{1}{2}\cdot\big\{-B_f^{(1)}+H_f^{(1)}(7)+H_f^{(1)}(6)+1\cdot\{H_f^{(0)}(1)-H_f^{(0)}(4)\}\big\}=\frac{7}{6}. \\
H_f^{(1)}(3) & = & \frac{1}{3}\cdot\big\{-B_f^{(1)}+2H_f^{(1)}(7)+H_f^{(1)}(5)+2\cdot\{H_f^{(0)}(1)-H_f^{(0)}(3)\}\big\}=\frac{23}{36}. \\
H_f^{(1)}(2) & = & \frac{1}{3}\cdot\big\{-B_f^{(1)}+2H_f^{(1)}(6)+H_f^{(1)}(5)+3\cdot\{H_f^{(0)}(1)-H_f^{(0)}(2)\}\big\}=\frac{13}{12}. \\
H_f^{(1)}(1) & = & \frac{1}{3}\cdot\{-B_f^{(1)}+2H_f^{(1)}(4)+H_f^{(1)}(3)\}=\frac{89}{108}. \\
B_f^{(2)} & = & \lambda_1 H_f^{(1)}(5)+\lambda_2 H_f^{(1)}(6)+\lambda_3 H_f^{(1)}(7)=\frac{23}{4}.
\end{array}$$

Similarly as in the discounted case we can show the following results.

Lemma 8.7
For any $x\in S$ and any $f^\infty\in C(D)$, we have $H_f^{(k)}(x)=0$ for $k=0,1,\dots,I(x)-1$.

We can compute the leading coefficients $I(x)$ recursively as follows:

(1) $H_f^{(0)}(e)=0$.

(2) For all states x with $I(x)=0$: $H_f^{(I(x))}(x)=\frac{1}{\mu\left(f(x)\right)}\cdot\{1+\sum_{j\in f(x)}\mu_j H_f^{(0)}(1_j,x)\}$.

(3) For all states x with $I(x)\geq 1$:

$B_f^{(I(x))}=\sum_{j=1}^n\lambda_j H_f^{(I(x)-1)}(0_j,x)$; $H_f^{(I(e))}(e)=0$;

$H_f^{(I(x))}(x)=\frac{1}{\mu\left(f(x)\right)}\cdot\{-B_f^{(I(x))}+\sum_{j\in f(x)}\mu_j H_f^{(I(x))}(1_j,x)\}+\sum_{j\in C_1(x)}\lambda_j H_f^{(I(x)-1)}(0_j,x)\}$ for $x\neq e$.

Example 8.1 (continued)

The computation of the leading coefficients is as follows:

By Lemma 8.7: $H_f^{(0)}(6)=H_f^{(0)}(7)=H_f^{(0)}(8)=0$.

$$\begin{array}{lcl}
H_f^{(I(5))}(5) & = & H_f^{(0)}(5)=\frac{1}{2}\cdot\{1+2H_f^{(0)}(8)=\frac{1}{2}. \\
H_f^{(I(4))}(4) & = & H_f^{(0)}(4)=\frac{1}{2}\cdot\{1+H_f^{(0)}(7)+H_f^{(0)}(6)=\frac{1}{2}. \\
H_f^{(I(3))}(3) & = & H_f^{(0)}(3)=\frac{1}{3}\cdot\{1+2H_f^{(0)}(7)+H_f^{(0)}(5)=\frac{1}{2}. \\
H_f^{(I(2))}(2) & = & H_f^{(0)}(2)=\frac{1}{3}\cdot\{1+2H_f^{(0)}(6)+H_f^{(0)}(5)=\frac{1}{2}. \\
H_f^{(I(1))}(4) & = & H_f^{(0)}(1)=\frac{1}{3}\cdot\{1+2H_f^{(0)}(4)+H_f^{(0)}(3)=\frac{5}{6}. \\
B_f^{(I(8))} & = & B_f^{(1)}=\lambda_1 H_f^{(0)}(5)+\lambda_2 H_f^{(0)}(6)+\lambda_3 H_f^{(0)}(7)=\frac{1}{2}. \\
H_f^{(I(7))}(7) & = & H_f^{(1)}(7)=\frac{1}{1}\cdot\{-B_f^{(1)}+H_f^{(1)}(8)+H_f^{(0)}(3)+2H_f^{(0)}(4)\}=\frac{1}{2}. \\
H_f^{(I(6))}(6) & = & H_f^{(1)}(6)=\frac{1}{1}\cdot\{-B_f^{(1)}+H_f^{(1)}(8)+H_f^{(0)}(2)+3H_f^{(0)}(4)\}=\frac{3}{2}.
\end{array}$$

<u>Remark</u>

Notice that when the system is in a failed state x, i.e. $I(x)=0$, then $H_f^{(0)}(x)$ satisfies the equation $H_f^{(I(x))}(x)=\frac{1}{\mu\left(f(x)\right)}\cdot\{1+\sum_{j\in f(x)}\mu_j H_f^{(0)}(1_j,x)\}$, i.e. the expected time until the system is back in operation when the initial state is x, policy f^∞ is employed and there are no failures. Thus, we obtain the following, intuitively expected, partial characterization of policies that maximize the availability of the system for small values of ρ: when the system is failed such policies must assign repairmen to failed components in such a way that the expected time until the system is back in operation, in absence of failures, is minimized.

In the next theorem we show that in one of the two optimality criteria that have been considered (discounted operation time and average operation time) asymptotically optimal policies are optimal when all failure rates are sufficiently small.

Theorem 8.6
Let f_*^∞ be asymptotically optimal to one of the two optimality criteria that have been considered. Then, there exists a $\rho_* > 0$ such that f_*^∞ is optimal for all $\rho \in (0, \rho_*)$.

Proof
We prove the theorem for the problem of maximizing the expected total discounted time that the system is in good states. The proof for the criterion of maximizing the average time the system is functioning is similar.
Recall that for any $f^\infty \in C(D)$ and for any $\rho \in \big(0, \frac{1}{\|Q(f)\|}\big)$, the $b_f(x)$'s possess convergent power series of the form $b_f(x) = \sum_{k=0}^{\infty} \rho^k b_f^{(k)}(x)$. Since there are finite many policies in $C(D)$, it follows that the power series representation of all $b_f(x)$'s are convergent for all $f^\infty \in C(D)$ and for any $\rho \in (0, \rho_1)$, where $\rho_1 := min_{f^\infty \in C(D)} \frac{1}{\|Q(f)\|}$.
Now, for any $x \in S$ and any pair $f_1^\infty, f_2^\infty \in C(D)$, it follows (see [249], p. 177) that $b_{f_1}(x) - b_{f_2}(x)$ may change sign a finite number of times. Since $b_{f_*}^{(I(x))}(x) = min_{f^\infty \in C(D)}\, b_f^{(I(x))}(x)$ for all $x \in S$, and there are finite many policies in $C(D)$ and states in S, the theorem follows. □

In the following application we restrict our attention to determining policies which are asymptotically optimal with respect to the availability criterion.

Application 8.1 *Series and parallel systems*
Consider first the n component series system maintained by r repairmen. The only functioning state is state $e = (1, 1, \ldots, 1)$. In Section 8.2.2 it was established that when $r = 1$ the optimal policy always assigns the repairman to the failed component with the smallest failure rate (SFR policy).

From the above remark we know that an asymptotically optimal policy f_*^∞ minimizes the expected time to state e from any initial state x in the absence of failures. Thus, in the terminology of stochastic scheduling an asymptotic optimal policy minimizes the expected makespan for allocating $|C_0(x)|$ tasks (repairs) on r identical processors (repairmen), for any state x.

For $r = 2$, it has been shown (see [35]) that an optimal policy assigns repairmen to failed components in $|C_0(x)|$ according to the $LEPT$ (Longest Expected Processing Time) rule. In the context of the series system an asymptotic optimal policy assigns repairmen to the failed components with the longest expected repair times. Notice that this $LEPT$ policy is optimal for sufficiently small failure rates (by Theorem 8.6).

For the parallel system the only failed state is state $(0, 0, \ldots, 0)$. Hence, $I(x) = |C_1(x)|$, and consequently $H_f^{(k)}(x) = 0$ for $k = 0, 1, \ldots, |C_1(x)| - 1$. It is easy to show by induction on $|C_0(x)|$ that the policy which always assigns repairmen to failed components with the smallest repair rates is asymptotically optimal.

8.3 Production and inventory control

8.3.1 No backlogging

The basic form of the production control model with no backlogging is as follows. Demand of a single product occurs during each of T consecutive time periods. The demand that occurs during a given period can be satisfied by production during that period or during any earlier period, as inventory is carried forward. This prescribes the case of *no backlogging.* Inventory at epoch 1 is zero, and inventory at the end of period T is also required to be zero. The model includes production costs and inventory costs. The objective is to schedule the production so as to satisfy demand at minimum total cost.

For the data and variables in the periods $t = 1, 2, \dots, T$ we use the following notation.

D_t = the demand in period t

$c_t(a)$ = the cost of production a units in period t

$h_t(i)$ = inventory cost when the inventory is i at the end of period t

a_t = decision variable for the production in period t

I_t = the inventory on hand at the beginning of period t

When the production and demand occur in integer quantities, the problem of meeting demand at minimal total cost can be formulated as the following integer programming problem.

$$min \left\{ \sum_{t=1}^{T} \{c_t(a_t) + h_t(I_t)\} \;\middle|\; \begin{array}{ll} I_1 = I_{T+1} = 0 & \\ I_t + a_t = D_t + I_{t+1}, & t = 1, 2, \dots, T \\ a_t \geq 0 \text{ and integer}, & t = 1, 2, \dots, T \\ I_t \geq 0 \text{ and integer}, & t = 2, 3, \dots, T \end{array} \right\}. \tag{8.28}$$

To find an optimal production plan by dynamic programming, note that the cost of operating the system during periods t through T depends on the inventory I_t, but not on prior inventories and not on prior production levels. Therefore, we constitute the states as the pairs (i, t), where i denote the inventory and t the period. Let

$f(i, t)$:= the minimum cost of satisfying demand during periods t through T if the inventory at epoch t is i.

The cost of an optimal production plan is $f(0, 1)$. The inventory $I_{T+1} = 0$, which constrains inventory and production during period T. We obtain $f(i, T) = h_T(i) + c_T(D_T - i)$ for $i = 0, 1, \dots, D_T$.

Consider a period $t < T$, and let a_t denote the quantity produced during period t. The production a_t is feasible if $I_t + a_t$ is at least as large as the demand D_t and if $I_t + a_t$ is no larger that the total demand during all remaining periods. Therefore, a_t is a *feasible* production if $a_t \in A(i, t)$, where $A(i, t)$ is defined by $A(i, t) := \{a \in \mathbb{N}_0 \mid D_t - i \leq a \leq \sum_{s=t}^{T} D_s - i\}$.

The above observations give rise to the functional equation

$$\begin{cases} f(i, T) &= h_T(i) + c_T(D_T - i),\ i = 0, 1, \dots, D_T. \\ f(i, t) &= min_{a \in A(i,t)}\{h_t(i) + c_t(a) + f(i + a - D_t, t + 1)\},\ t = T-1, T-2, \dots, 1,\ 0 \leq i \leq \sum_{s=t}^{T} D_s \end{cases} \tag{8.29}$$

The preceding functional equation can be solved by backward induction.

In many economic situations the cost functions are concave, reflecting decreasing marginal costs. A function g with domain in $\mathbb{Z}$ is called *concave* if

$$g(n + 2) - g(n + 1) \leq g(n + 1) - g(n) \text{ for all } n. \tag{8.30}$$

Let $\Delta g(n) := g(n+1) - g(n)$ and $\Delta^2 g(n) := \Delta g(n+1) - \Delta g(n) = g(n+2) - 2g(n+1) + g(n),\ n \in \mathbb{Z}$. Then, concavity is equivalent to $\Delta^2 g(n) \leq 0$ for all n. The next theorem shows that for concave production and inventory functions the optimal production plan has a special structure.

Theorem 8.7
If the production and inventory functions, c_t and h_t respectively, are concave for all periods t, then there exists an optimal production plan $(a_1, a_2, \ldots, a_T)$ for which $I_t \cdot a_t = 0$ for $t = 1, 2, \ldots, T$.

Proof
Consider an optimal production plan $(a_1, a_2, \ldots, a_{T+1})$. If there are more optimal plans, take the plan for which $\sum_{t=1}^{T} (a_t + I_t)$ is minimal. Aiming for a contradiction, we assume that $I_t > 0$ and $a_t > 0$ for some $2 \leq t \leq T$. Since, $I_t > 0$ there has been production in at least one of the previous periods and let s the last period prior than t in which $a_s > 0$. Then, $a_{s+1} = a_{s+2} = \cdots = a_{t-1} = 0$.
Consider two other production plans, which are - of course - not cheaper than the optimal production plan:

$$a_j^{'} := \begin{cases} a_s + 1 & j = s \\ a_t - 1 & j = t \\ a_j & j \neq s, t \end{cases} \quad \text{and } a_j^{''} := \begin{cases} a_s - 1 & j = s \\ a_t + 1 & j = t \\ a_j & j \neq s, t \end{cases}$$

Note that $I_k^{'} = I_k + 1,\ s < k \leq t$ and $I_k^{''} = I_k - 1,\ s < k \leq t$. Therefore,

$$c_s(a_s) + c_t(a_t) + \textstyle\sum_{k=s+1}^{t} h_k(I_k) \leq c_s(a_s + 1) + c_t(a_t - 1) + \sum_{k=s+1}^{t} h_k(I_k + 1)$$

and

$$c_s(a_s) + c_t(a_t) + \textstyle\sum_{k=s+1}^{t} h_k(I_k) \leq c_s(a_s - 1) + c_t(a_t + 1) + \sum_{k=s+1}^{t} h_k(I_k - 1).$$

Add these two inequalities and rearrange the sum as

$$0 \leq \Delta^2 c_s(a_s - 1) + \Delta^2 c_t(a_t - 1) + \textstyle\sum_{k=s+1}^{t} \Delta^2 h_k(I_k - 1) \leq 0,$$

the last inequality by the concaveness of the production and inventory functions. Hence, the three plans are all optimal. However, $\sum_{t=1}^{T}(a_t^{''} + I_t^{''}) = \sum_{t=1}^{T}(a_t + I_t) - (t - s) < \sum_{t=1}^{T}(a_t + I_t)$. This contradicts the supposed minimality of that sum, which completes the proof. □

Theorem 8.7 demonstrates that production need only occur in period t if the inventory at the start of that period is zero. Consequently the quantity produced during period t must equal the total demand of the periods $t, t+1, \ldots, u-1$ for some $t+1 \leq u \leq T+1$.
Let $d_{tu} := \sum_{k=t}^{u-1} D_k,\ t+1 \leq u \leq T+1$, the demand of the periods $t, t+1, \ldots, u-1$.
This gives rise to a dynamic programming formulation in which state t represents the situation of having no inventory on hand at the start of period t. Let

$f(t)$:= the minimum cost of satisfying demands during periods t through T if the inventory at epoch t is 0.

Let c_{tu} denote the total of costs incurred during periods $t, t+1, \ldots, u-1$ if transition occurs from state t to state u, i.e. the first production after period t is in period u. Inventory at epoch t equals 0, so c_{tu} includes the holding cost $h_t(0)$. Exactly d_{tu} units are produced in period t, so c_{tu} includes the production cost $c_t(d_{tu})$. Exactly d_{ku} units of inventory remain at any epoch k between $t+1$ and u, so c_{tu} includes the holding cost $h_k(d_{ku})$. This leads to the relation $c_{tu} = h_t(0) + c_t(d_{tu}) + \sum_{k=t+1}^{u-1} h_k(d_{ku})$ and the functional equation

$$\begin{cases} f(T+1) & = & 0 \\ f(t) & = & min_{\{u \mid t+1 \leq u \leq T+1\}} \{c_{tu} + f(u)\},\ t = T, T-1, \ldots, 1. \end{cases} \tag{8.31}$$

The tables of $\{d_{tu}\}$ and $\{c_{tu}\}$ can be built with work proportional to T^2. The solution of (8.31) with backward induction is also proportional to T^2. Hence, the overall complexity of this approach is of order $\mathcal{O}(T^2)$.

8.3.2 Backlogging

When backlogging is allowed, demand may accumulate and be satisfied by productions during subsequent periods. The only effect to program (8.28) is to delete the requirement that the variables I_2 through I_T be nonnegative. When I_t is nonnegative, it still represents an inventory of I_t units; when I_t is negative, it now represents a *shortage* of $-I_t$ units of unfilled (backlogged) demand that must be satisfied by production during periods t through T. Hence, the set of states (i,t) and the set $A(i,t)$ of feasible productions are:

$$\{(i,t) \mid 1 \leq t \leq T;\ -\textstyle\sum_{s=1}^{t-1} D_s \leq i \leq \sum_{s=t}^{T} D_s\};\ A(i,t) = \{a \in \mathbb{N}_0 \mid 0 \leq a \leq \sum_{s=t}^{T} D_s - i\}.$$

In the case of backlogging h_t is called the *holding/shortage* cost function for period t. When I_t is nonnegative, $h_t(I_t)$ remains equal to the cost of having I_t units of inventory on hand at the start of period t; when I_t is negative, $h_t(I_t)$ becomes the cost of having a shortage of $-I_t$ units of unfilled demand on hand at the start of period t. The functional equation of the production model in which backlogging is allowed becomes:

$$\begin{cases} f(i,T) &= h_T(i) + c_T(D_T - i),\ i = -\sum_{s=1}^{T-1} D_s, \dots, D_T. \\ f(i,t) &= min_{a \in A(i,t)}\{h_t(i) + c_t(a) + f(i+a-D_t, t+1)\}, \\ & \qquad\qquad t = T-1, T-2, \dots, 1;\ -\sum_{s=1}^{t-1} D_s \leq i \leq \sum_{s=t}^{T} D_s \end{cases} \tag{8.32}$$

Also in this case we consider concave cost functions. The model has *concave shortage cost* if, for any t, the function h_t is convave on $\{0, -1, -2, \dots\}$, i.e. $h_t(n+2) - h_t(n+1) \leq h_t(n+1) - h_t(n)$ for $n = -2, -3, \dots$. In other words, the model has concave holding and shortage costs if, for each t, the function h_t satisfies

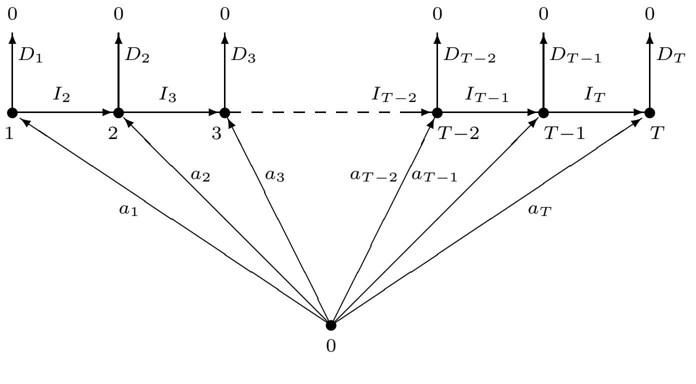
$$h_t(n+2) - h_t(n+1) \leq h_t(n+1) - h_t(n),\ n \in \mathbb{Z}\backslash\{-1\}. \tag{8.33}$$

When the production and holding/shortage costs are concave, the solution to the production planning problem has the form given in the following theorem.

Theorem 8.8
If the production and holding/shortage functions, c_t and h_t respectively, are concave for all periods t, then there exists an optimal production plan $(a_1, a_2, \dots, a_T)$ having this property: if $a_m > 0$ and $a_n > 0$ with $m < n$, then $I_t = 0$ for at least one $t \in \{m+1, m+2, \dots, n\}$.

Before we prove Theorem 8.8 we model the problem as a network flow problem.

For $t = 1, 2, \dots, T$ a node arises in the network; furthermore we add a node 0. There are arcs $(t, t+1)$, $t = 1, 2, \dots, T-1$; $(0,t)$ and $(t,0)$ for $t = 1, 2, \dots, T$. From node 0, we send a variable flow a_t into node t, $t = 1, 2, \dots, T$; from node t we send D_t to node 0 $(t = 1, 2, \dots, T)$ and I_{t+1} to node $t+1$ $(t = 1, 2, \dots, T-1)$. Flow into node t equals $I_t + a_t$, where $I_1 = 0$, and flow out of node t equals $D_t + I_{t+1}$, where $I_{T+1} = 0$.



Hence, flow conservation in the nodes t corresponds to the constraints of program (8.28) and flow conservation in node 0 corresponds to a total production of $a_1 + a_2 + \cdots + a_T = D_1 + D_2 + \cdots + D_T$.

We shall call a flow $(a_1, a_2, \ldots, a_T)$ *feasible* if $a_t \geq 0$ and integer for $t = 1, 2, \ldots, T$. Consider a feasible flow. An arc (i, j) is *active* if flow along is not zero. A *loop* is said to exist if one can start at a node and return to it by traversing a sequence of distinct active arcs, not necessarily in the direction of the arcs. Suppose, for instance, that a_2, I_3 and a_3 are all positive. Then, the node sequence $(0, 2, 3, 0)$ prescribes a loop; also the node sequence $(0, 3, 2, 0)$ prescribes a loop.

Proof of Theorem 8.8

Consider an optimal production plan $(a_1, a_2, \ldots, a_{T+1})$. If there are more optimal plans, take the plan for which $\sum_{t=1}^{T} (a_t + I_t)$ is minimal. Aiming for a contradiction, we assume that $a_m > 0$ and $a_n > 0$ with $m < n$, and $I_t \neq 0$ for $t = m+1, m+2, \ldots, n$. Hence, $(0, m, m+1, \ldots, n, 0)$ is a loop. A feasible flow results if we increase a_m by 1, decrease a_n by 1, and consequently, increase I_k by 1 for $m + 1 \leq k \leq n$. Similarly, we may decrease a_m by 1, increase a_n by 1, and decrease I_k by 1 for $m + 1 \leq k \leq n$. These perturbed flows cannot decrease cost below the costs of the optimal plan. So,

$$c_m(a_m) + c_n(a_n) + \textstyle\sum_{k=m+1}^{n} h_k(I_k) \leq c_m(a_m + 1) + c_n(a_n - 1) + \sum_{k=m+1}^{n} h_k(I_k + 1)$$

and

$$c_m(a_m) + c_n(a_n) + \textstyle\sum_{k=m+1}^{n} h_k(I_k) \leq c_m(a_m - 1) + c_n(a_n + 1) + \sum_{k=m+1}^{m} h_k(I_k - 1).$$

Add these two inequalities and rearrange the sum as

$$0 \leq \Delta^2 c_m(a_m - 1) + \Delta^2 c_n(a_n - 1) + \textstyle\sum_{k=m+1}^{n} \Delta^2 h_k(I_k - 1) \leq 0,$$

the last inequality by the concaveness of the production and holding/shortage functions ($\Delta^2 h_k(I_k - 1)$ is not necessarily nonnegative if $I_k = 0$, but notice that $I_k \neq 0$ for $k = m+1, m+2, \ldots, n$). Hence, the three plans are all optimal. However,

$$\textstyle\sum_{t=1}^{T} (a_t'' + I_t'') = \sum_{t=1}^{T} (a_t + I_t) - (t - s) < \sum_{t=1}^{T} (a_t + I_t).$$

This contradicts the supposed minimality of that sum, which completes the proof. □

Let $a^* = (a_1, a_2, \ldots, a_T)$ be an optimal production plan of the type described in Theorem 8.8. Consider any period $t \leq T$ such that $I_t = 0$ (since $I_1 = 0$ such a t exists). Consider also the lowest-numbered $u > t$ such that $I_u = 0$ (since $I_{T+1} = 0$ such a u exists). Exactly $d_{tu} = \sum_{k=t}^{u-1} D_k$ units must be produced during periods t through $u - 1$. We argue by contradiction that production of these d_{tu} units is concentrated in one period k, where $t \leq k \leq u - 1$. Suppose not: that is, that a^* splits this production between periods k and $l > k$. Since $a_k > 0$ and $a_l > 0$ with $k < l$, it follows from Theorem 8.8 that $I_p = 0$ for some p with $k + 1 \leq p \leq l \leq u - 1$. The minimality of u precludes this. Hence, the production of these d_{tu} units is not split.

For $t \leq k \leq u - 1$ let $c_{tu}(k)$ denote the total cost incurred during periods t through $u - 1$ if the total demand d_{tu} occurring during these periods is satisfied by production in period k, i.e.

$$c_{tu}(k) = h_t(0) + c_k(d_{tu}) + \sum_{s=t+1}^{k} h_s(-d_{ts}) + \sum_{s=k+1}^{u-1} h_s(d_{su}). \tag{8.34}$$

In a dynamic programming formulation state t again denotes the situation of having no inventory on hand at the start of period t. Transition occurs from state t to state u if it is decided to produce d_{tu} units during some intermediate period k. The cheapest transition from state t tot state u costs c^*_{tu}, where

$$c^*_{tu} := min_{\{k \mid t \leq k \leq u-1\}} \, c_{tu}(k). \tag{8.35}$$

As usual, $f(t)$ is defined, for $t = 1, 2, \ldots, T$, by

$f(t) :=$ the minimum cost of satisfying demands during periods t through T if $I_t = 0$.

One gets the functional equation

$$\begin{cases} f(T+1) & = & 0 \\ f(t) & = & min_{\{u \mid t+1 \leq u \leq T+1\}} \{c^*_{tu} + f(u)\}, \ t = T, T-1, \ldots, 1. \end{cases} \tag{8.36}$$

This functional equation is similar to the functional equation for the concave-cost case without backlogging. The difference is that c^*_{tu} replaces c_{tu}. Once a table is built, (8.36) can be solved with work proportional to T^2, just in the case of backlogging. However, the work needed to build a table of c^*_{tu} from (8.35) is proportional to T^3, not to T^2.

8.3.3 Inventory control and single-critical-number policies

This section concerns an inventory model over a finite horizon of T periods with uncertain demand and without backlogging. The symbols i and a are used consequently throughout this section, where i denotes the inventory on hand at the start of a period, just prior to deciding whether to place an order, and a denotes the inventory on hand at the start of a period, just after deciding whether to place an order (ordering is instantaneous). So $a \geq i$ and the number of units ordered is $a - i$. We assume that stock is indivisible, so a and i are integers.

Let the states be (i, t), depicting the situation of having i units of inventory on hand at the start of period t, just before deciding whether and how much to order. Let $f(i, t)$ be the minimum discounted cost over the remaining periods, given state (i, t). We are interested in $f(0, 1)$ and state $(i, T + 1)$ represents the situation with inventory of i units at the end of period T. Hence, $f(i, T + 1) = -si, \ i \geq 0$.

The data of the model are as follows:

D_t = the (uncertain) demand during period t.
$p_t(j)$ = the probability that the demand in period t is $j, \ j = 0, 1, \ldots$.
R = the (retail) unit sales price (independent of the period).
k = the unit ordering cost (independent of the period).
s = the unit salvage value at the end of period T.
r = the interest rate per period.
α = the discount factor, which equals $\frac{1}{1+r}$.

The net cost incurred during period t equals:
- ordering cost: $(a - i)k$;
- interest charge on inventory: $(1 - \alpha)ak$;
- expected sales in period t: $\alpha R \, \mathbb{E} \cdot \big\{min\{D_t, a\}\big\}$, where $\mathbb{E}\,\big\{min\{D_t, a\}\big\} = \sum_{j=0}^{a-1} jp_t(j) + a \sum_{j \,\geq\, a} p_t(j)$.
Therefore, we obtain the following optimality equation

$$\begin{cases} f(i, T+1) & = & -si, \ i \geq 0 \\ f(i, t) & = & \inf_{\{a \mid a \geq i\}} \big\{(a-i)k + (1-\alpha)ak - \alpha R \cdot \{\sum_{j=0}^{a-1} jp_t(j) + a \sum_{j \,\geq\, a} p_t(j)\} + \\ & & \qquad \alpha \cdot \mathbb{E}\{f((a - D_t)^+, t+1)\}\big\} \\ & = & inf_{\{a \mid a \geq i\}} \big\{(a-i)k + (1-\alpha)ak - \alpha R\{\sum_{j=0}^{a-1} jp_t(j) + a \sum_{j \,\geq\, a} p_t(j)\} + \\ & & \alpha \cdot \{\sum_{j=0}^{a-1} p_t(j) f(a-j, t+1) + a \sum_{j \,\geq\, a} p_t(j) f(0, t+1)\}\big\}, \ i \geq 0, \ t = T, T-1, \ldots, 1. \end{cases} \tag{8.37}$$

Define $F(i,t) := f(i,t) + ik$ for all i and t. Then, $f(a-j,t+1) = F(a-j,t+1) - (a-j)k$ for all $j < a$, and $f(0,t+1) = F(0,t+1)$. Therefore, the optimality equation (8.37) can be written as

$$\begin{cases} F(i,T+1) & = \quad (k-s)i, \ i \ge 0 \\ F(i,t) & = \quad \inf_{\{a \mid a \ge i\}} \big\{(2-\alpha)ak - \alpha R \cdot \{\sum_{j=0}^{a-1} jp_t(j) + a\sum_{j \ge a} p_t(j)\} + \\ & \qquad \alpha \cdot \{\sum_{j=0}^{a-1} p_t(j)F(a-j,t+1) + \sum_{j \ge a} p_t(j)F(0,t+1)\} - \alpha \sum_{j=0}^{a-1} p_t(j)(a-j)k\big\}, \\ & \qquad\qquad i \ge 0, \ t = T, T-1, \dots, 1. \end{cases} \tag{8.38}$$

Since

$$\begin{aligned} \alpha \sum_{j=0}^{a-1} p_t(j)(a-j)k &= \alpha ka \sum_{j=0}^{a-1} p_t(j) - \alpha k \sum_{j=0}^{a-1} jp_t(j) = \alpha ka\{1 - \sum_{j \ge a} p_t(j)\} - \alpha k \sum_{j=0}^{a-1} jp_t(j) \\ &= \alpha ka - \alpha ka \sum_{j \ge a} p_t(j)\} - \alpha k \sum_{j=0}^{a-1} jp_t(j), \end{aligned}$$

we have the optimality equation

$$\begin{cases} F(i,T+1) & = \quad (k-s)i, \ i \ge 0 \\ F(i,t) & = \quad \inf_{\{a \mid a \ge i\}} \big\{2(1-\alpha)ak - \alpha(R-k) \cdot \{\sum_{j=0}^{a-1} jp_t(j) + a\sum_{j \ge a} p_t(j)\} + \\ & \qquad \alpha \cdot \{\sum_{j=0}^{a-1} p_t(j)F(a-j,t+1) + \sum_{j \ge a} p_t(j)F(0,t+1)\}\big\}, \ i \ge 0, \ t = T, T-1, \dots, 1. \end{cases} \tag{8.39}$$

Notice that the expression in (8.39) for which the infimum is taken over all $a \ge i$ is independent of i. Let

$$\begin{cases} L_t(a) := 2(1-\alpha)ak - \alpha(R-k) \cdot \big\{ \sum_{j=0}^{a-1} jp_t(j) + a\sum_{j \ge a} p_t(j)\big\} + \\ \qquad\qquad \alpha \cdot \big\{ \sum_{j=0}^{a-1} p_t(j)F(a-j,t+1) + \sum_{j \ge a} p_t(j)F(0,t+1)\big\}. \end{cases} \tag{8.40}$$

Theorem 8.9

Let $s < k < R, \ k > 0$ and $\mathbb{E}\{D_t\} < \infty$ for all t.

(1) The function $L_t(a)$ is convex for all t.

(2) For all t, there exists a nonnegative integer S_t such that

$$L_t(S_t) = \min_{\{a \ge 0\}} L_t(a) \text{ and } F(i,t) = \begin{cases} L_t(S_t) & \text{for } i \le S_t; \\ L_t(i) & \text{for } i > S_t. \end{cases}$$

Proof

Part (1)

We first prove that $2(1-\alpha)ak - \alpha(R-k)\{\sum_{j=0}^{a-1} jp_t(j) + a\sum_{j \ge a} p_t(j)\}$ is convex. Therefore, it is sufficient to show that $G_t(a) = \sum_{j=0}^{a-1} jp_t(j) + a\sum_{j \ge a} p_t(j)$ is concave.

$$\Delta G_t(a) = \sum_{j=0}^{a} jp_t(j) + (a+1)\sum_{j \ge a+1} p_t(j) - \sum_{j=0}^{a-1} jp_t(j) + a\sum_{j \ge a} p_t(j) = \sum_{j \ge a+1} p_t(j).$$

Hence, $\Delta^2 G_t(a) = \Delta G_t(a+1) - \Delta G_t(a) = p_t(a+1) \ge 0$: G_t is concave for all t.

Next, we show that L_T is convex. Since $L_T(a) = G_T(a) + \alpha(k-s)\sum_{j=0}^{a-1}(a-j)p_T(j)$, it is sufficient to show that $g_T(a) := \sum_{j=0}^{a-1}(a-j)p_T(j)$ is convex.

$$\Delta g_T(a) \quad = \quad \sum_{j=0}^{a}(a+1-j)p_T(j) - \sum_{j=0}^{a-1}(a-j)p_T(j) = \sum_{j=0}^{a} p_T(j),$$

and consequently, $\Delta^2 g_t(a) = \Delta g_t(a+1) - \Delta g_t(a) = p_T(a+1) \ge 0$: g_T is concave.

We now show that $L_t(a) \to \infty$ if $a \to \infty$. We have

$$L_T(a) \quad = \quad 2(1-\alpha)ak - \alpha(R-k)\big\{ \sum_{j=0}^{a-1} jp_T(j) + a\sum_{j \ge a} p_T(j)\big\} + \alpha(k-s) \ \sum_{j=0}^{a-1} p_t(j)(a-j).$$

Since $\sum_{j=0}^{a-1} jp_T(j) + a\sum_{j \ge a} p_T(j) \le \mathbb{E}\{D_T\}$ and $\sum_{j=0}^{a-1} p_t(j)(a-j) \ge \sum_{j \ge 0} p_t(j)(a-j) = a - \mathbb{E}\{D_T\}$, we obtain $L_T(a) \ge a\{2(1-\alpha)k + \alpha(k-s)\} - \alpha(R-s)\,\mathbb{E}\{D_T\}$, implying that $L_T(a) \to \infty$ if $a \to \infty$.

Finally, we show inductively for $t = T, T-1, \dots, 1$ that L_t is convex and that $L_t(a) \to \infty$ if $a \to \infty$. Assume that L_{t+1} is convex and that $L_{t+1}(a) \to \infty$ if $a \to \infty$. Then, L_{t+1} attains its minimum at some integer S_{t+1} that satisfies $L_{t+1}(S_{t+1}) = \min_{a \geq 0} L_{t+1}(a)$ and $F(i, t+1) = \begin{cases} L_{t+1}(S_{t+1}) & \text{for } i \leq S_{t+1}; \\ L_{t+1}(i) & \text{for } i > S_{t+1}. \end{cases}$

For the convexity of $L_t(a)$ it is sufficient to show the convexity of

$$H_t(a) := \textstyle\sum_{j=0}^{a-1} p_t(j) F(a-j, t+1) + \sum_{j\,\geq\, a} p_t(j) F(0, t+1).$$

We can write

$$\begin{aligned} \Delta H_t(a) &= \textstyle\sum_{j=0}^{a} p_t(j) F(a+1-j, t+1) + \sum_{j\,\geq\, a+1} p_t(j) F(0, t+1) \\ &\qquad - \textstyle\sum_{j=0}^{a-1} p_t(j) F(a-j, t+1) - \sum_{j\,\geq\, a} p_t(j) F(0, t+1) \\ &= \textstyle\sum_{j=0}^{a} p_t(j) \{F(a+1-j, t+1) - F(a-j, t+1)\}, \end{aligned}$$

implying

$$\begin{aligned} \Delta^2 H_t(a) &= \textstyle\sum_{j=0}^{a+1} p_t(j) \{F(a+2-j, t+1) - F(a+1-j, t+1)\} - \\ &\qquad \textstyle\sum_{j=0}^{a} p_t(j) \{F(a+1-j, t+1) - F(a-j, t+1)\} \\ &= p_t(a+1)\{F(1, t+1) - F(0, t+1)\} + \textstyle\sum_{j=0}^{a} p_t(j) \big\{\{F(a+2-j, t+1) - \\ &\qquad F(a+1-j, t+1)\} - \{F(a+1-j, t+1) - F(a-j, t+1)\}\big\} \\ &= p_t(a+1)\{F(1, t+1) - F(0, t+1)\} + \textstyle\sum_{j=0}^{a} p_t(j) \Delta^2 F(a-j, t+1). \end{aligned}$$

Notice that

$$\Delta^2 F(a-j, t+1) = \begin{cases} \Delta^2 L_{t+1}(a-j) \geq 0 & a-j \geq S_{t+1} \\ \Delta L_{t+1}(a+1-j) \geq 0 & a+1-j = S_{t+1} \\ 0 & a+2-j \leq S_{t+1} \end{cases}$$

Since $F(1, t+1) = inf_{\{a \geq 1\}} L_{t+1}(a) \geq inf_{\{a \geq 0\}} L_{t+1}(a) = F(0, t+1)$, we obtain $\Delta^2 H_t(a) \geq 0$ for all $a \geq 0$ and consequently L_t is a convex function.

To complete the proof, we must show that $L_t(a) \to \infty$ as $a \to \infty$.

Pick M large enough that $\mathbb{P}\{D_t \leq M\} \geq 0.5$. Then consider $a \geq M + S_{t+1}$.

In the case that $D_t \leq M$, we have $(a - D_t)^+ = a - D_t \geq a - M \geq S_{t+1}$, and the fact that L_{t+1} is nondecreasing for $i \geq S_{t+1}$ assures that $F((a-D_t)^+, t+1) \geq L_{t+1}(a - M)$.

In the case $D_t > M$, the fact that $F(i, t+1)$ is nondecreasing assures that $F((a-D_t)^+, t+1) \geq F(0, t+1)$. Hence,

$$\mathbb{E}\{F((a-D_t)^+, t+1)\} \geq 0.5 L_{t+1}(a-M) + 0.5 F(0, t+1) \text{ for } a \geq M + S_{t+1}.$$

Therefore, we obtain

$$L_t(a) = 2(1-\alpha)ak - \alpha(R-k)\,\mathbb{E}\{D_t\} + 0.5 L_{t+1}(a-M) + 0.5 F(0, t+1),$$

implying that $L_t(a) \to \infty$ as $a \to \infty$ for all $\alpha \in [0, 1]$, even when α equals 0 or 1. □

An optimal policy has the following structure: if the inventory at the start of period t is at least S_t, no order is placed; if the inventory is $i < S_t$ then exactly $S_t - i$ units are ordered. This one-parameter ordering rule is called a *single-critical-number policy.*

8.3.4 Inventory control and (s, S)-policies

This section concerns a model of inventory control with also uncertain demand, but with a fixed charge for placing an order. Ordering is instantaneous and has to be paid at delivery. This model covers both backlogging and no backlogging. The ordering cost includes a cost per unit ordered and a fixed charge of *setup cost* for placing any order. Let $K_t \geq 0$ be the setup cost and h_t the unit ordering cost in period t. The cost-minimizing ordering rule for this type of model is often characterized by two numbers per period, as follows. Each period has an *order-up-to-quantity* S_t and a *reorder point* $s_t \le S_t$. If the inventory I_t at the start of period t is at least s_t, no order is placed; if $I_t < i_t$ then exactly $S_t - I_t$ units are ordered. This two-parameter ordering rule is called an (s, S)-policy. An (s, S)-policy has the propery that any order in period t must be for more than $S_t - s_t$ units, large enough that the benefit of the added inventory offsets the setup cost. In addition to the notation mentioned previously, we use the following notation:

R_t = the (retail) unit sales price during period t (customers pay at the end of the period in which they place their orders, even if their orders are backlogged).

$e(i^+)$ = the salvage value of having $i^+ = max\{i, 0\}$ units of inventory at the end of period T.

$h_t(a)$ = the expectation of the inventory cost during period t, given that the inventory is a at the start of period t, just after deciding whether to place an order.

$I_t(a, D_t)$ = the inventory on hand at the beginning of period $t+1$, given as a function of y and D_t.

To describe the ordering cost we employ the function $H(z) := \begin{cases} 0 & \text{for } z \le 0; \\ 1 & \text{for } z > 0. \end{cases}$.

Then, the cost of ordering z units at the start of period t equals $K_t H(z) + z \cdot k_t$. Let $c_t(i, a)$ denote the present value at the start of period t of the expected cost incurred during this period, given in terms of period's before-ordering inventory i and after-ordering inventory a:

$$c_t(i, a) = K_t H(a - i) + (a - i)k_t + h_t(a) - \alpha\big\{a - \mathbb{E}\{I_t(a, D_t)\}\big\}R_t. \tag{8.41}$$

We have $\mathbb{E}\{I_t(a, D_t)\} = \begin{cases} a - \sum_{j=0}^{\infty} p_t(j)j & \text{if backlogging is allowed;} \\ a - \sum_{j=0}^{a} p_t(j)j - a\sum_{j=a+1}^{\infty} p_t(j) & \text{if backlogging is not allowed.} \end{cases}$.

The states are (i, t) denoting the situation of having i units of inventory on hand at the start of period t, just before deciding whether and how much to order. Interpret $f(i, t)$ as the present value at the start of period t of the cost incurred from then to the end of the planning horizon if state (i, t) is observed and if an optimal policy is followed. This leads to the following optimality equation

$$\begin{cases} f(i, T+1) &= K_{T+1}H(-i) + k_{T+1}(-i)^+ - e(i^+) \\ f(i, t) &= \inf_{\{a \,|\, a \ge i\}} \big\{c_t(i, a) + \alpha\,\mathbb{E}\{f(I_t(a, D_t), t+1)\}\big\}, \; t = T, T-1, \dots, 1. \end{cases} \tag{8.42}$$

where the first equation accounts for the cost of disposing of excess demand $(-i)^+$ by a special end-of-planning-horizon order and for the salvage value $e(i^+)$, converted to a cost. The term $-ik_t$ in (8.41) is independent of the decision a and can be factored out of (8.42). This motivates a change of variables. Let $F(i, t) := f(i, t) + ik_t$. Then, with

$$G_t(a) := h_t(a) + (k_t - \alpha R_t)a + \alpha(R_t - k_{t+1})\,\mathbb{E}\{I_t(a, D_t)\}, \tag{8.43}$$

we obtain (the verification is left to the reader)

$$\begin{cases} F(i, T+1) &= K_{T+1}H(-i) + k_{T+1}i^+ - e(i^+) \\ F(i, t) &= \inf_{\{a \,|\, a \ge i\}} \big\{K_t H(a - i) + G_t(a) + \alpha\,\mathbb{E}\{F(I_t(a, D_t), t+1)\}\big\}, \; t = T, T-1, \dots, 1. \end{cases} \tag{8.44}$$

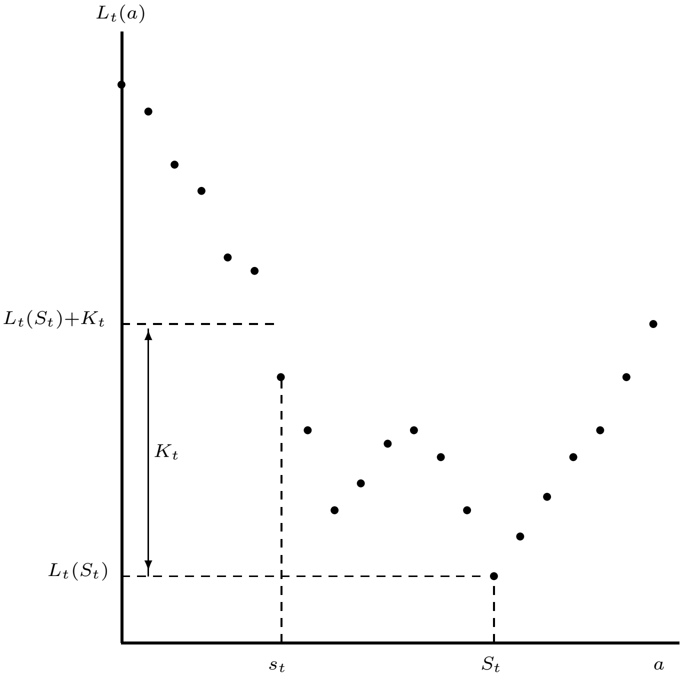
The quantity $G_t(a)$ is called the *operating cost*; it accounts for all units of cash flow during perod t except the setup cost. The first term in $G_t(a)$ is the inventory carrying cost. To interpret the remaining terms, we imagine that the starting inventory y is purchased from the supplier at unit cost k_t, that ending inventory $I_t(a, D_t)$ is returned to the supplier at unit price k_{t+1}, and that sales of $a - I_t(a, D_t)$ units occur at unit price R_t.

Next, we write the optimality equation as

$$F(i,t) = \inf_{\{a\,|\,a\geq i\}} \big\{K_t H(a-i) + L_t(a)\big\}, \text{ where } L_t(a) = G_t(a) + \alpha\,\mathbb{E}\{F(I_t(a,D_t),t+1)\}. \tag{8.45}$$

Up to this point, the development has followed the same pattern as in the previous section, where we showed the convexity of L_t. In the case of fixed setup cost the function L_t is not convec, in general. It may have the structure shown in the picture, which has two local minima. The point identified in the figure as S_t is the point where the function L_t attains its global minimum. The point identified as s_t is the smallest value of a for which $L_t(a) \leq L_t(S_t) + K_t$. Since K_t is nonnegative, one has $s_t \leq S_t$. So S_t and s_t satisfy

$$\begin{aligned} L_t(S_t) &= \inf_a \{L_t(a)\} \\ s_t &= \inf\{a \mid L_T(a) \leq L_t(S_t) + K_t\} \end{aligned} \tag{8.46}$$



It is now argued that an (s,S)-policy is optimal for the strange shaped function of the figure. Consider state (i,t) with $i \geq s_t$. Note that it is optimal to set $a = i$, because in that case $K_t + L_t(a) \geq L_t(i)$ for all $a \geq i$, i.e. the setup cost cannot be recouped by an $a \geq i$. Now consider a state (i,t) with $i < s_t$. Then, $K_t + L_t(S_t) \leq L_t(i)$ and $K_t + L_t(S_t) \leq K_t + L_t(a)$ for any a. The next theorem gives conditions under which an (s,S)-policy is optimal.

Theorem 8.10

Suppose that the following five conditions hold.

(1) For $t = 1,2,\ldots,T$, the function $G_t(\cdot)$, defined in (8.43), is convex.

(2) For $t = 1,2,\ldots,T$, the setup costs K_t satisfies $K_t \geq \alpha K_{t+1}$.

(3) For $t = 1,2,\ldots,T$ and for each value of D_t, the function $I_t(\cdot,D_t)$ is convex and nondecreasing

(4) The function $k_{T+1}i^+ - e(i^+)$ is convex and nondecreasing in i.

(5) All expectations are finite and $inf_a\, L_t(a)$ is attained for all $t = 1,2,\ldots,T$.

Then, for $t = 1,2,\ldots,T$, there exists S_t and s_t that satisfy $L_t(S_t) = \inf_a \{L_t(a)\}$ and

$s_t = \inf\{a \mid L_t(a) \leq L_t(S_t) + K_t\}$. Moreover, $F(i,t) = \begin{cases} L_t(S_t) + K_t & \textit{if } i < s_t; \\ L_t(i) & \textit{if } i \geq s_t. \end{cases}$

Before we prove the theorem we make some preparations.

With $K \geq 0$ a function $f : \mathbb{Z} \to \mathbb{R}$ is called *K-convex* if any triple $a < b < c$ satisfies

$$f(c) + K \geq f(b) + (c-b)\Big(\frac{f(b)-f(a)}{b-a}\Big).$$

Hence, the straight line passing through the points $(a,f(a))$ and $(b,f(b))$ has in c a value of at most $f(c) + K$. Since b is between a and c we can write $b = \alpha a + (1-\alpha)c$ for some $\alpha \in (0,1)$, and getting the following equivalent form of K-convexity.

$$\begin{aligned} f(c)+K &\geq f\{\alpha a+(1-\alpha)c\}+\alpha(c-a)\cdot\tfrac{f\{\alpha a+(1-\alpha)c\}-f(a)}{(1-\alpha)(c-a)} \;\Leftrightarrow \\ (1-\alpha)\{f(c)+K\} &\geq (1-\alpha)f\{\alpha a+(1-\alpha)c\}+\alpha\cdot\big\{f\{\alpha a+(1-\alpha)c\}-f(a)\big\} \;\Leftrightarrow \\ (1-\alpha)\{f(c)+K\} &\geq f\{\alpha a+(1-\alpha)c\}-\alpha f(a). \end{aligned}$$

Notice that this inequality is also valid if $a = c$ and/or $\alpha \in \{0,1\}$. Hence, K-convexity is equivalent to $\alpha f(a)+(1-\alpha)\{f(c)+K\} \geq f\{\alpha a+(1-\alpha)c\}$ for all $a \leq c$ and all $\alpha \in [0,1]$.

A function $f : \mathbb{Z} \to \mathbb{R}$ is K-*quasi-convex* if any triple $a < b < c$ with $f(a) < f(b)$ satisfies $f(c)+K \geq f(b)$. Notice that an increase in a K-quasi-convex function cannot be followed by a decrease that exceeds K.

Lemma 8.8

A function $f : \mathbb{Z} \to \mathbb{R}$ that is K-convex is also K-quasi-convex.

Proof

Suppose that $a < b < c$ and $f(a) < f(b)$. Since f is K-convex, we obtain

$$f(c)+K \geq f(b)+(c-b)\Big(\tfrac{f(b)-f(a)}{b-a}\Big) > f(b).$$

□

Lemma 8.9

Let $f : \mathbb{Z} \to \mathbb{Z}$ be convex and nondecreasing and let $g : \mathbb{Z} \to \mathbb{R}$ be K-convex. Furthermore, we have $g(a) \leq g(c)+K$ for all $a < c$. Then, $g\{f(x)\}$ is K-convex.

Proof

Take any $a < c$ and $\alpha \in (0,1)$, and let $b = \alpha a+(1-\alpha)c$. Then, we have to show that

$$\alpha g\{f(a)\}+(1-\alpha)\big\{g\{f(c)\}+K\big\} \geq g\{f(b)\}.$$

Since f is convex and nondecreasing, we have $f(a) \leq f(b) \leq \alpha f(a)+(1-\alpha)f(c) \leq f(c)$. Hence, there exists a number $\beta \in [0,1]$ such that $f(b) = \beta f(a)+(1-\beta)f(c)$, implying

$$\beta f(a)+(1-\beta)f(c) = f(b) \leq \alpha f(a)+(1-\alpha)f(c), \text{ or equivalently, } (\beta-\alpha)\{f(a)-f(b)\} \leq 0.$$

Since f is nondecreasing, we obtain $\beta \geq \alpha$. Because g is K-convex on Y, $f(a) \leq f(b) \leq f(c)$ yields $\beta g\{f(a)\}+(1-\beta)\big\{g\{f(c)\}+K\big\} \geq g\{f(b)\}$. Therefore,

$$\begin{aligned} &\alpha g\{f(a)\}+(1-\alpha)\big\{g\{f(c)\}+K\big\}-g\{f(b)\} \geq \\ &\alpha g\{f(a)\}+(1-\alpha)\big\{g\{f(c)\}+K\big\}-\beta g\{f(a)\}-(1-\beta)\big\{g\{f(c)\}+K\big\} = \\ &(\alpha-\beta)\Big\{g\{f(a)\}-g\{f(c)\}-K\Big\} \geq 0, \end{aligned}$$

because $\beta \geq \alpha$ and $g\{f(a)\} \leq g\{f(c)\}+K$, the last inequality since the property of g given in the formulation of the lemma. □

Lemma 8.10

Let the function $L : \mathbb{Z} \to \mathbb{R}$ be K-quasi-convex. Suppose that S such that $L(S) = inf_a\,\{L(a)\}$ exists. Furthermore, let $F(i) = \inf_{\{a \geq i\}}\,\{KH(a-i)+L(a)\}$.

Then, $F(i) = \begin{cases} L(S)+K & \textit{if } i < s \\ L(i) & \textit{if } i \geq s \end{cases}$, *where* $s; = \inf\{a \mid L(a) \leq L(S)+K\}$.

Proof

Since $L(S) \leq L(S)+K$, the definition of s assures that $s \leq S$, possibly $s = -\infty$. We will show the result in 4 cases, depending on the position of i with respect to s and S.

Case 1: $i < s$.

The definition of s assures that $L(i) > L(S)+K$. Since $i < S$, we have $F(i) = L(S)+K$.

Case 2: $i = s > -\infty$.
By the definition of s we have $L(i) \le L(S) + K$, and consequently, $F(i) = L(i)$ in this case.
Case 3: $s < i \le S$.
Suppose that $L(i) > L(S) + K$. Then, $i \ne s$. So, $s < i < S$ and $L(s) \le L(S) + K < L(i)$. Hence, by the K-quasi-convexity of L, $L(S) + K \ge L(i)$, implying a contradiction. Hence, $L(i) \le L(S) + K$, and we obtain $F(i) = L(i)$ in this case.
Case 4: $i > S$.
Suppose that $F(i) < L(i)$. Then, the definition of F assures that there exists a c that satisfies $c > i$ with $L(c) + K < L(i)$. Then, $S < i < c$ and $L(S) < L(i)$. Hence, by the K-quasi-convexity of L, $L(c) + K \ge L(i)$, implying a contradiction. So, $F(i) = L(i)$ in this case. □

The next lemma provides conditions under which $F : \mathbb{Z} \to \mathbb{R}$ is a K-convex function that satiesfies
$F(b) \le F(c) + K$ if $b < c$.

Lemma 8.11

Let $L : \mathbb{Z} \to \mathbb{R}$ be a K-convex function. Then, F is K-convex, and for all elements $b < c$ we have $F(b) \le F(c) + K$.

Proof

Take any $b < c$. Since $F(i) = inf_{\{a\,|\,a \ge i\}}\,\{KH(a-i) + L(a)\}$, we obtain $F(b) \le K + L(c)$. Lemma 8.8 shows that L is K-quasi-convex. Hence, Lemma 8.10 applies, and F satisfies $F(i) = \begin{cases} L(S) + K & \text{if } i < s \\ L(i) & \text{if } i \ge s \end{cases}$, where $s := \inf\{a \mid L(a) \le L(S) + K\}$.
Consider elements $a < b < c$. We have to show $F(b) \le F(c) + K + \frac{c-b}{b-a} \cdot \{F(a) - F(b)\}$.
If $F(a) \ge F(b)$ this follows immediately from $F(b) \le F(c) + K$.
If $a \le s$, Lemma 8.10 assures that $F(a) = L(a)$, $F(b) = L(b)$ and $F(c) = L(c)$, and the result is immediate from the K-convexity of L. In the remaining case, $a < s$ and $F(a) < F(b)$, Lemma 8.10 shows $F(a) = L(S) + K \ge F(s)$. Hence, $F(b) = L(b)$ and $s < b$, implying

$$\tfrac{c-b}{b-a} \cdot \{F(a) - F(b)\} > \tfrac{c-b}{b-s} \cdot \{F(a) - F(b)\} \ge \tfrac{c-b}{b-s} \cdot \{F(s) - F(b)\} \ge F(b) - F(c) - K,$$

the last inequality follows from the case $a = s$. □

The preceding Lemmas are now molded into a proof of Theorem 8.10.

Proof of Theorem 8.10

Condition (4) shows that $F(\cdot, T+1)$ is a K_{T+1}-convex function that satisfies, for all elements $b < c$, $F(b, T+1) \le F(c, T+1) + K_{T+1}$. This initializes the following inductive hypothesis:

> $F(\cdot, t+1)$ is a K_{t+1}-convex function that satisfies $F(b, t+1) \le F(c, t+1) + K_{t+1}$ for all elements $b < c$.

This hypothesis and condition (3) let us supply Lemma 8.9 with $g(\cdot) = F(\cdot, t+1)$ and with $f(\cdot) = I_t(\cdot, D_t)$. Lemma 8.9 shows that $F\{I_t(\cdot, D_t), t+1\}$ is K_{t+1}-convex. Since condition (2) gives $K_t \ge \alpha K_{t+1}$, this shows that $\alpha F\{I_t(\cdot, D_t), t+1\}$ is K_t-convex. Since K-convexity is preserved under convex combinations, (5) suffices for the K_t-convexity of $\alpha\,\mathbb{E}\left\{F\{I_t(\cdot, D_t), t+1\}\right\}$. So, condition (1) shows that $L_t(\cdot)$ is K_t-convex. Lemma 8.8 shows that $L_t(\cdot)$ is K_t-quasi convex, and condition (5) implies that $L_t(S_t) = \inf_a \{L_t(a)\}$. Hence, Lemma 8.10 shows that $F(\cdot, t)$ satisfies $F(i, t) = \begin{cases} L_t(S_t) + K_t & \text{if } i < s_t; \\ L_t(i) & \text{if } i \ge s_t. \end{cases}$.
Lemma 8.11 shows that $F(\cdot, t)$ is a K_t-convex function that satisfies $F(b, t) \le F(c, t) + K_t$ for all elements $b < c$. This completes the proof. □

8.4 Optimal control of queues

A queueing system includes *servers, customers* and *queues* for the customers awaiting service. The queues are also called *buffers.* We will discuss several types of queueing models.

8.4.1 The single-server queue

Customers enter the queue, wait their turn, are served by the single server, and depart the system. We might place a *controller* at the entrance to the queue to decide which customers to admit to the queues (*admission control*). Or we could impose a control on the server that could adjust the rate at which customers are served (*service rate control*). Both methods of control can be imposed simultaneously.

1. Admission control for batch arrivals

The state of the system is the number of customers in the buffer at the beginning of a time slot, and thus $S = \{0, 1, \dots\}$. At the beginning of each slot a batch of customers arrives and p_j is the probability that j customers arrive ($j = 0, 1, \dots$). In every state there are two actions available:
1 = accept the incoming batch or 0 = reject the incoming batch.

Case a: The action must be chosen before the size of the batch is observed.

There is a nonnegative holding cost $h(i)$ incurred when there are i customers in the buffer (assume $h(0) = 0$). There is a positive rejection cost r incurred whenever a batch is rejected. Hence, the immediate cost
$c(i,a) = \begin{cases} h(i) + r & \text{if } a = 0,\ i \in S; \\ h(i) & \text{if } a = 1,\ i \in S. \end{cases}$

Service occurs according to a geometric distribution with fixed rate μ, where $0 < \mu < 1$. This means that the probability of a successful service in any slot is μ. If the service is unsuccessful, then another try is made with the same probability of success, and this continues until the customer has been successful served. If a batch arrives to an empty buffer and is accepted, then its customers are available for service at the beginning of the following slot.

Hence, the transition probabilities are:

$$\begin{array}{llllll} i = 0: & p_{00}(0) = 1; & i \geq 1: & p_{i,i-1}(0) = \mu; & p_{i,i-1}(1) = \mu p_0; \\ & p_{0j}(1) = p_j,\ j \geq 0; & & p_{i,i}(0) \quad = 1 - \mu; & p_{i,i+j}(1) = \mu p_{j+1} + (1-\mu)p_j,\ j \geq 0. \end{array}$$

Consider value iteration with $\alpha = 1$, i.e.

$$\begin{array}{lll} v_0^n & = & min\{r + \sum_j p_{0j}(0)v_j^{n-1},\ \sum_j p_{0j}(1)v_j^{n-1}\} = min\{r + v_0^{n-1},\ \sum_j p_j v_j^{n-1}\}. \\ v_i^n & = & min\{h(i) + r + \sum_j p_{ij}(0)v_j^{n-1},\ h(i) + \sum_j p_{ij}(1)v_j^{n-1}\} \\ & = & h(i) + min\{r + \mu v_{i-1}^{n-1} + (1-\mu)v_i^{n-1},\ \mu \sum_j p_j v_{i-1+j}^{n-1} + (1-\mu)\sum_j p_j v_{i+j}^{n-1}\},\ i \geq 1. \end{array}$$

Lemma 8.12

Assume that $h(i)$ is nondecreasing in i and consider value iteration with $v_i^0 = 0,\ i \in S$. Then, v_i^n is nondecreasing in i for all $n \geq 0$.

Proof

The lemma is shown by induction on n. We first show that v_i^n is finite for all n and i by showing that $v_i^n \leq nr + (n+1)h(i),\ i \in S$. For $n = 0$ we have $0 = v_i^0 \leq 0 \cdot r + 1 \cdot (h(i) = h(i),\ i \in S$. Assume that $v_i^n \leq nr + (n+1)h(i),\ i \in S$. Then,

$$\begin{array}{rcl} v_i^{n+1} & \leq & h(i) + r + \mu v_{i-1}^n + (1-\mu)v_i^n \\ & \leq & h(i) + r + \mu\{nr + (n+1)h(i-1)\} + (1-\mu)\{nr + (n+1)h(i)\} \\ & \leq & h(i) + r + \mu\{nr + (n+1)h(i)\} + (1-\mu)\{nr + (n+1)h(i)\} \\ & = & h(i) + r + nr + (n+1)h(i) = (n+1)r + (n+2)h(i),\ i \in S. \end{array}$$

$v^0 \equiv 0$ is nondecreasing in i. Assume that v_i^n is nondecreasing in i. Since

$$h(1) + r + \mu v_0^n + (1-\mu)v_1^n \geq r + \mu v_0^n + (1-\mu)v_0^n = r + v_0^n$$

and

$$h(1) + \mu \textstyle\sum_j p_j v_j^n + (1-\mu)\sum_j p_j v_{j+1}^n \geq \mu \sum_j p_j v_j^n + (1-\mu)\sum_j p_j v_j^n = \sum_j p_j v_j^n,$$

each term in the minimum of v_0^{n+1} is bounded above by the corresponding term in v_1^{n+1}, and hence $v_0^{n+1} \leq v_1^{n+1}$. Suppose that the minimum in v_i^{n+1} $(i \geq 1)$ is obtained by the rejection action. By the induction hypothesis v_{i-1}^n and v_i^n are nondecreasing in i and consequently is $h(i) + r + \mu v_{i-1}^n + (1-\mu)v_i^n$ nondecreasing in i. Now suppose that the minimum in v_i^{n+1} $(i \geq 1)$ is obtained by the accepting action. For each fixed j, by the induction hypothesis, v_{i-1+j}^n and v_{i+j}^n are nondecreasing in i. Since $\sum_j p_j v_{i-1+j}^n$ and $\sum_j p_j v_{i+j}^n$ are convex combinations of v_{i-1+j}^n and v_{i+j}^n, $\sum_j p_j v_{i-1+j}^n$ and $\sum_j p_j v_{i+j}^n$ are nondecreasing in i. Hence, $h(i) + \mu \sum_j p_j v_{i-1+j}^n + (1-\mu)\sum_j p_j v_{i+j}^n$ is nondecreasing in i. Thus both terms in the minimum are nondecreasing in i, and consequently v_i^{n+1} is nondecreasing in i. □

Case b: The size of the incoming batch may be observed before the action is chosen.
In this case we take as states the pairs (i,k), where i denotes the number of customers in the buffer and k the size of the incoming batch: $S = \{(i,k) \mid i = 0,1,\dots;\ k = 0,1,\dots\}$. The holding cost is as in Case a and there is a positive rejection cost $r(k)$ incurred whenever a batch of size k is rejected $(r(0) = 0)$. Hence, the cost structure is as follows: $c\{(i,k),a) = \begin{cases} h(i) & ,\ i \in S,\ k = 0; \\ h(i) + r & ,\ i \in S,\ k \geq 1,\ a = 0; \\ h(i) & ,\ i \in S,\ k \geq 1,\ a = 1. \end{cases}$

For the transition probabilities we obtain for all $k \geq 0$ and $j \geq 0$:

$$\begin{array}{llll} i = 0: & p_{(0,k)(0,j)}(0) = p_j; \quad i \geq 1: & p_{(i,k)(i-1,j)}(0) = \mu p_j; & p_{(i,k)(i+k-1,j)}(1) = \mu p_j; \\ & p_{(0,k)(k,j)}(1) = p_j; & p_{(i,k)(i,j)}\ \ (0) = (1-\mu)p_j; & p_{(i,k)(i+k,j)}\ \ (1) = (1-\mu)p_j. \end{array}$$

2. Admission control for an $M/M/1$ queue
Assume that customers arrive according to a Poisson process with parameter λ, and assume that the service time is exponentially distributed with parameter μ. We observe the system at each arrival and departure (*semi-Markov model*). As state space we use $S = \{0,1,2,\dots\} \times \{0,1\}$. The system is in state $(i,0)$ if there are i customers in the system and there is a departure; then, the only action $a = 0$ is to continue. The state $(i,1)$ occurs when there are i jobs in the system and a new customer arrives; in state $(i,1)$ the controller may admit $(a = 1)$ or refuse $(a = 0)$ service to the arrival.

In state $(0,0)$ the only action is to continue: with probability 1 the next state is state $(0,1)$ and the time until the next transition is exponentially distributed with rate λ. In state $(0,1)$ there are two actions: if $a = 0$ (refuse) the next state is with probability 1 again state $(0,1)$ and the time until the next transition is exponentially distributed with rate λ; if $a = 1$ (admission) the time until the next transition is exponentially distributed with rate $\lambda + \mu$, and the next state is with probability $\frac{\lambda}{\lambda+\mu}$ state $(1,1)$ and with probability $\frac{\mu}{\lambda+\mu}$ state $(0,0)$.

In the states $(i,0)$, with $i \geq 1$, the only action $a = 0$ is to continue. Then, the next state is with probability $\frac{\lambda}{\lambda+\mu}$ state $(i,1)$ and with probability $\frac{\mu}{\lambda+\mu}$ state $(i-1,0)$; the time until the next transition is exponentially distributed with rate $\lambda + \mu$.

In the states $(i, 1)$, with $i \geq 1$, there are two actions. If $a = 0$ (refuse) the next state is with probability $\frac{\lambda}{\lambda+\mu}$ again state $(i, 1)$ and with probability $\frac{\mu}{\lambda+\mu}$ state $(i - 1, 0)$. If $a = 1$ (admission) the next state is with probability $\frac{\lambda}{\lambda+\mu}$ again state $(i + 1, 1)$ and with probability $\frac{\mu}{\lambda+\mu}$ state $(i, 0)$. The time until the next transition is exponentially distributed with rate $\lambda + \mu$.

Let $\nu_{(i,b)}(a)$ be the parameter of the exponential distribution of the time until the next observation and let $p_{(i,b)(j,c)}(a)$ be the probability that the next state is state (j, c), given the current state (i, b) and the action a. Then, we have

$$\nu_{(i,b)}(a) = \begin{cases} \lambda & \text{if } i = 0,\ b = 0 \text{ or } 1,\ a = 0; \\ \lambda + \mu & \text{if } i = 0,\ b = 1,\ a = 1 \text{ or } i \geq 1. \end{cases}$$

$$p_{(i,b)(j,c)}(a) \begin{cases} 1 & \text{if } (i, b) = (0, 0),\ a = 0,\ (j, c) = (0, 1) \text{ or } (s, b) = (0, 1),\ a = 0,\ (j, c) = (0, 1); \\ \frac{\lambda}{\lambda+\mu} & \text{if } i \geq 1,\ b = 1,\ a = 0,\ (j, b) = (i, 1) \text{ or } i \geq 0,\ b = 1,\ a = 1,\ (j, b) = (i + 1, 1); \\ \frac{\mu}{\lambda+\mu} & \text{if } i \geq 0,\ b = 1,\ a = 1,\ (j, b) = (i, 0) \text{ or } i \geq 1,\ b = 1,\ a = 0,\ (j, b) = (i - 1, 0) \\ & \text{or } i \geq 1,\ b = 0,\ a = 0,\ (j, b) = (i - 1, 0); \\ 0 & \text{otherwise} \end{cases}$$

With the exception of the states $(0, 0)$ and $(0, 1)$ all transitions occur at rate $\lambda + \mu$. To *uniformize* the system we alter the transition structure in only these states:

$p'_{(0,0)(0,0)}(0) = \frac{\mu}{\lambda+\mu}$; $p'_{(0,0)(0,1)}(0) = \frac{\lambda}{\lambda+\mu}$;
$p'_{(0,1)(0,0)}(0) = \frac{\mu}{\lambda+\mu}$; $p'_{(0,1)(0,1)}(0) = \frac{\lambda}{\lambda+\mu}$;
$p'_{(i,b)(j,c)}(a) = p_{(i,b)(j,c)}(a)$ if $i = 0,\ b = 1,\ a = 1$ or $i \geq 1$.

In the uniformized system, we observe the system more often when it is empty than in the untransformed system, so that this transformation increases the probability that it occupies $(0, 0)$ and $(0, 1)$ for $a = 0$. We may also interpret this transformation as adding "fictitious" service completions at these states.

Furthermore, we assume that each arriving customer contributes r units of revenue and the system incurs a holding cost at rate $h(i)$ per unit time whenever there are i jobs in the system, where $h(0) = 0$.

As utility function we consider the *discounted model*, in which we assume continuous-time discounting at rate $\alpha > 0$. This means that the present value of one unit received t units in the future equals $e^{-\alpha t}$. For $(i, b) \in S$ and $a = 0$ or 1, let $r'_{(i,b)}(a)$ denote the expected total discounted reward between two decision epochs in the uniformized system, given that the system occupies state (i, b) and the decision maker chooses action a. The expected discounted holding cost during one epoch, given that the system occupies state (i, b) and the decision maker chooses action a, is per unit:

$$\mathbb{E}^a_{(i,b)} \{\int_0^\tau e^{-\alpha t} dt\} = \frac{1}{\alpha}\, \mathbb{E}^a_{(i,b)} \{1 - e^{-\alpha\tau}\} = \frac{1}{\alpha} \int_0^\infty \{1 - e^{-\alpha t}\}\, t\, f(t)\, dt,$$

where $f(t)$ is the density of the exponential distribution with parameter $\lambda + \mu$, i.e. $f(t) = (\lambda + \mu)e^{-(\lambda+\mu)t}$ for all $t \geq 0$. Hence,

$$\begin{aligned} \mathbb{E}^a_{(i,b)} \{\int_0^\tau e^{-\alpha t} dt\} &= \frac{\lambda+\mu}{\alpha} \int_0^\infty \{1 - e^{-\alpha t}\} e^{-(\lambda+\mu)t}\, t\, dt \\ &= \frac{\lambda+\mu}{\alpha} \Big\{ \int_0^\infty e^{-(\lambda+\mu)t}\, t\, dt - \int_0^\infty e^{-(\alpha+\lambda+\mu)t}\, t\, dt \Big\} \\ &= \frac{\lambda+\mu}{\alpha} \Big\{ \frac{1}{\lambda+\mu} - \frac{1}{\alpha+\lambda+\mu} \Big\} = \frac{\lambda+\mu}{\alpha} \Big\{ \frac{\alpha}{(\lambda+\mu)(\alpha+\lambda+\mu)} \Big\} = \frac{1}{\alpha+\lambda+\mu}. \end{aligned}$$

Now it follows that the rewards in the uniformized system satisfy

$r'_{(0,0)}(0) = 0$; $r'_{(i,0)}(0) = \frac{-h(i)}{\alpha+\lambda+\mu}$, $i \geq 1$; $r'_{(i,1)}(1) = r + \frac{-h(i)}{\alpha+\lambda+\mu}$, $i \geq 0$.
$r'_{(0,1)}(0) = 0$; $r'_{(i,1)}(0) = \frac{-h(i)}{\alpha+\lambda+\mu}$, $i \geq 1$;

The optimality equation for this model becomes (cf. Kallenberg [148], Chapter 7):

$$\begin{aligned} v_{(i,b)} &= max_a\{r'_{(i,b)}(a) + \{\textstyle\int_0^\infty e^{-\alpha t} f(t)\,dt\} \sum_{(j,c)} p'_{(i,b)(j,c)}(a) v_{(j,c)}\} \\ &= max_a\{r'_{(i,b)}(a) + \tfrac{\lambda+\mu}{\alpha+\lambda+\mu} \sum_{(j,c)} p'_{(i,b)(j,c)}(a) v_{(j,c)}\},\ (i,b) \in S. \end{aligned}$$

Hence, more explicitly,

$$\begin{aligned} v_{(0,0)} &= \tfrac{\lambda+\mu}{\alpha+\lambda+\mu}\{\tfrac{\mu}{\lambda+\mu} v_{(0,0)} + \tfrac{\lambda}{\lambda+\mu} v_{(0,1)}\}. \\ v_{(0,1)} &= max\Big\{r - \tfrac{h(1)}{\alpha+\lambda+\mu} + \tfrac{\lambda+\mu}{\alpha+\lambda+\mu}\{\tfrac{\mu}{\lambda+\mu} v_{(0,0)} + \tfrac{\lambda}{\lambda+\mu} v_{(1,1)}\},\ \tfrac{\lambda+\mu}{\alpha+\lambda+\mu}\{\tfrac{\mu}{\lambda+\mu} v_{(0,0)} + \tfrac{\lambda}{\lambda+\mu} v_{(0,1)}\}\Big\}. \\ v_{(i,0)} &= -\tfrac{h(1)}{\alpha+\lambda+\mu} + \tfrac{\lambda+\mu}{\alpha+\lambda+\mu}\{\tfrac{\mu}{\lambda+\mu} v_{(i-1,0)} + \tfrac{\lambda}{\lambda+\mu} v_{(i,1)}\},\ i \ge 0. \\ v_{(i,1)} &= max\Big\{r - \tfrac{h(i+1)}{\alpha+\lambda+\mu} + \tfrac{\lambda+\mu}{\alpha+\lambda+\mu}\{\tfrac{\mu}{\lambda+\mu} v_{(i,0)} + \tfrac{\lambda}{\lambda+\mu} v_{(i+1,1)}\}, \\ &\qquad -\tfrac{h(i)}{\alpha+\lambda+\mu} + \tfrac{\lambda+\mu}{\alpha+\lambda+\mu}\{\tfrac{\mu}{\lambda+\mu} v_{(i-1,0)} + \tfrac{\lambda}{\lambda+\mu} v_{(i,1)}\}\Big\} \\ &= max\{r + v_{(i+1,0)}, v_{(i,1)}\},\ i \ge 1. \end{aligned}$$

It can be shown that, if $h(i)$ is nondecreasing and convex, there exists an optimal *control limit* policy.

3. Service rate control

As state space we have again $S = \{0, 1, \dots\}$. In state 0 there is no control action available since there are no customers to serve. We may think of the action 0 = take no service action. In state $i \ge 1$ actions consist of the allowable service rate $0 < a_1 < a_2 < \cdots < a_m < 1$. This means that the server must serve if the buffer is nonempty ($a_1 > 0$) and that perfect service is unavailable ($a_m < 1$). The holding cost is the same as in arrival control. There is a nonnegative cost $c(k)$ of choosing to serve at rate a_k during a particular slot (the cost in state 0 is 0). Hence, the immediate cost $c(i,k) = \begin{cases} 0 & \text{if } i = 0; \\ h(i) + c(k) & \text{if } i \ge 1,\ 1 \le k \le m. \end{cases}$

The transition probabilities are:

$$\begin{aligned} i = 0:\ \ p_{0j}(0) = p_j,\ j \ge 0;\quad i \ge 1:\ \ & p_{i,i-1}(k) = a_k p_0 && 1 \le k \le m; \\ & p_{i,i+j}(k) = a_k p_{j+1} + (1 - a_k) p_j && 1 \le k \le m,\ j \ge 0. \end{aligned}$$

8.4.2 Parallel queues

In parallel queues are a number of K servers with individual queues. Customers arrive at the router and are send to one of these servers. It is assumed that once the routing has taken place, the customer cannot switch from one queue to another. We assume that the service rates of the servers are constant. The control mechanism is involved through the routing decision for an arriving customer. An appropriate state description is the vector $i = (i_1, i_2, \dots, i_K)$, where i_k is the number of customers in queue k ($k = 1, 2, \dots, K$). The cost is then a function of the pair (i, k), where k is the action chosen, i.e. the server to which the customer is routed. This cost consists of a holding cost reflecting the number of customers in each queue and a cost of routing to queue k.

Suppose that the customers that arrived in slot t were routed to queue k but that at the beginning of slot $t + 1$ the controller wishes to route the newly arriving customers to queue $l \ne k$. We allow that this switch causes a *switching cost*. To handle this situation, we would enlarge the state description to be (i, k), where the current buffer content vector i is augmented with the previous routing decision. The cost is then a function of the state-action pair $\{(i, k), l\}$.

Let us assume that we have batch arrivals. The problem concerns the routing of an incoming batch to one of the K parallel servers. Each server maintains its own queue, and server k serves its customers at geometric rate μ_k, where $0 < \mu_k < 1,\ k = 1, 2, \dots, K$. We also assume that the routing decision is made before the size of the incoming batch is observed.

There is a nonnegative holding cost $h_k(i_k)$ associated with the content of queue k. The total holding cost is $h(i) = \sum_{k=1}^K h_k(i_k)$. In addition there is a nonnegative cost $c(k,l)$ for changing the routing from server k to server l, where $c(k,k) = 0$ for each k. The cost structure is: $c\{(i,k),l)\} = h(i) + c(k,l)$.

Some thoughtful notation can facilitate the writing of the transition probabilities. Let $j(l)$ be the K-dimensional vector with j in the l-th place and 0 elsewhere. Then,

$$p_{(0,k)(j(l),l)}(l) = p_j,\ 1 \le k \le K,\ 1 \le l \le K,\ j \ge 0.$$

Now let $i \neq 0$ be a state vector and let $F(i) = \{j \mid i_j > 0\}$. Let $E(i) \subseteq F(i)$ be the subset of $F(i)$ (possibly empty) representing those servers who complete service during the current slot. The probability of this event is

$$\mathbb{P}\{E(i)\} = \prod_{k \in E(i)} \mu_k \prod_{k \in F(i) \setminus E(i)} (1 - \mu_k).$$

Finally let $e(E(i))$ be a vector with 1 in every coordinate of $E(i)$ and 0 elsewhere. If the system is in state (i,k) there is a probability p_j that the next batch contains j customers and there is a probability $\mathbb{P}\{E(i)\}$ that the servers of $E(i)$ complete their services. Hence, we have the following transition probabilities in case the router assigns the next batch to server l:

$$p_{(i,k)(i+j(l)-e(E(i)),l)}(l) = p_j\,\mathbb{P}\{E(i)\},\ i \neq 0,\ E(i) \subseteq F(i),\ 1 \le k \le K,\ 1 \le l \le K,\ j \ge 0.$$

8.5 Stochastic scheduling

In a scheduling problem, jobs have to be processed on a number of machines. Each machine can only process one job at a time. Each job i has a given processing time T_{ij} on machine j. In stochastic scheduling, these processing times are random variables. At certain time points decisions have to be made, e.g. which job is assigned to which machine. There is a utility function by which different policies can be measured, and we want to find a policy that optimizes the utility function. We will illustrate this in a number of examples.

8.5.1 Maximizing finite-time returns on a single processor

Suppose there are n jobs to be performed sequentially within a fixed time T. The ith job takes an exponentially amount of time with rate μ_i and, if completed within time T, earns the decision maker an amount r_i. At the start and whenever a job is completed the decision maker must decide which of the remaining jobs to process, with his objective being to maximize the total expected earnings.

It follows from the lack-of-memory property of the exponential distribution that, if job i is attempted for a time dt, then it will be completed with probability $\mu_i dt + o(dt)$, thus the expected gain will be $\mu_i r_i dt + o(dt)$. Hence, it seems as if the expected return is the same as if we earned $\mu_i r_i$ per unit time that job i is being performed. To show this formally, suppose that t units of time remain when job i is initiated. If X_i is the time needed to perform this job, then the expected return from job i is

$$\mathbb{E}\{\text{return from job } i\} = r_i \cdot \mathbb{P}\{X_i < t\} = r_i(1 - e^{-\mu_i t}) = \mu_i r_i \cdot \tfrac{1-e^{-\mu_i t}}{\mu_i}.$$

Since for any nonnegative stochastic variable Y with density function $f(y)$ we have

$$\mathbb{E}\{Y\} = \int_0^\infty y f(y) dy = \int_0^\infty \{\int_0^y dx\} f(y) dy = \int_0^\infty \{\int_x^\infty f(y) dy\} dx = \int_0^\infty \mathbb{P}\{Y > x\} dx,$$

and hence,

$$\mathbb{E}\{\min(X_i,t)\} = \int_0^\infty \mathbb{P}\{\min(X_i,t) > x\} dx = \int_0^t e^{-\mu_i x} dx = \tfrac{1-e^{-\mu_i t}}{\mu_i}.$$

Therefore, we obtain

$$\mathbb{E}\{Y\} = \mu_i r_i\,\mathbb{E}\{\min(X_i,t)\} = \mu_i r_i\,\mathbb{E}\{\text{length of time job } i \text{ is worked on}\}.$$

Hence, it follows that, for any policy R,

$$\mathbb{E}_R\{\text{total return}\} = \sum_{i=1}^{n} \mu_i r_i \, \mathbb{E}_R\{\text{length of time job } i \text{ is worked on}\}. \tag{8.47}$$

That is, the total expected return is the same as it would be if we earned money at a rate $\mu_i r_i$ whenever job i is worked on. From this we see that the expected amount earned by time T is maximized by working on jobs in decreasing order of $\mu_i r_i$. So at any decision time point the decision maker chooses job k where $\mu_k r_k = \max_i \{\mu_i r_i \mid \text{job } i \text{ is not completed}\}$.

8.5.2 Optimality of the μc-rule

1. One server allocation to parallel queues with preemption

Customers arrive at a system of m parallel queues and one server. The system operates at discrete time points, i.e. arrival times and service times take values in the set $\{1, 2, \dots\}$. Furthermore, the arrival times are arbitrary and the service time T_i, for a customer in queue i, is geometrically distributed with rate μ_i,

$$\mathbb{P}\{T_i = n\} = (1-\mu_i)^{n-1} \cdot \mu_i, \; n \in \mathbb{N}, \text{ with } \mu_i \in (0,1), \; 1 \le i \le m, \text{ and } \mathbb{E}\{T_i\} = \mu_i^{-1}.$$

At any time point $t = 1, 2, \dots$ the server chooses a customer from one of the queues; this is an example of a server assignment model. Services may be interrupted and resumed later on (*preemption*). For each customer in queue i, a cost c_i is charged per unit of time that this customer is in the system. A policy is a rule to assign each server to one of the queues. Which policy minimizes the total cost in T periods? This model is more interesting than the nonpreemptive model, which is a rather trivial example (cf. Exercise 1.8).

Let $N_i^t(R)$ be the number of customers in period t in queue i, if policy R is used. Then, the performance measure is $\min_R \; \mathbb{E}\big\{\sum_{t=1}^{T} \sum_{i=1}^{m} c_i \cdot N_i^t(R)\big\}$. The next theorem indicates that the so-called μc-rule is an optimal policy. This rule assigns the server to queue k, where k is a nonempty queue satisfying $\mu_k c_k = \max_i\{\mu_i c_i \mid \text{queue } i \text{ is nonempty}\}$. Note that $\mu_i c_i$ is the expected cost per unit of service for a customer in queue i, and by using the μc-rule, the largest reduction of the expected cost in the next period is obtained.

Theorem 8.11

The μc-rule is optimal for the preemptive allocation of a single server to parallel queues.

Proof

Assume that the μc-rule is optimal after some time $t \le T$ (any rule is optimal after time T, because we consider a finite horizon of T periods). It will be shown that this rule is also optimal at time t. Then, by backward induction, it is clear that the μc-rule is optimal over the whole horizon.

For any sample path of the states and actions of the stochastic process we make the following observation. Consider a policy that serves a customer in queue j at time t while there is a customer in queue i at time t, where i and j are such that $c_i\mu_i > c_j\mu_j$. Denote by τ the first time after time t that this policy services a customer in queue i (let $\tau = T+1$ if the policy does not serve a customer in queue i during the times $t+1, t+2, \dots, T$).

Modify the policy by serving a customer in queue i at time t and a customer in queue j at time τ, i.e. interchange the actions at times t and τ. The effect of this modification can be calculated as follows. With probability μ_i the service of the customer in queue i will be completed in epoch t in the modified policy. Thus with probability μ_i the cost of the customer in queue i is reduced by $\sum_{s=t+1}^{\tau} c_i$. Similarly, with probability μ_j the cost of the customer in queue j is increased by $\sum_{s=t+1}^{\tau} c_j$. Thus, the the expected reduction in cost is $(c_i\mu_i - c_j\mu_j)(\tau - t) > 0$. This shows that the μc-rule is an optimal policy. □

2. Serving Poisson arrivals nonpreemptively with a single server

Jobs of different classes arrive as independent Poisson arrivals. The jobs of class i go to queue i for $i = 1, 2, \dots, m$. A job in queue i has a mean service time equal to $\frac{1}{\mu_i}$ and a waiting cost of c_i per unit of time. All the service times are independent. The problem is to find a nonpreemptive server allocation policy that minimizes the long-term *average waiting cost* per unit of time, i.e. $\min_R \ \mathbb{E}\left\{\sum_{i=1}^m c_i \cdot N_i(R)\right\}$, where $N_i(R)$ denotes the long-term average number of customers in queue i in the system, given policy R, i.e. $N_i(R) = \lim_{T\to\infty} \frac{1}{T}\sum_{t=1}^T N_i^t(R)$ with $N_i^t(R)$ be the number of customers in period t in queue i, if policy R is used.

Theorem 8.12

The μc-rule is optimal for serving Poisson arrivals nonpreemptively with a single server.

Proof

The proof is based on a *working-conserving property.*

First one observes that it suffices to consider nonidling policies, i.e. policies under which the server is never idle when there is a customer to serve. Indeed, one can always consider that an idling policy is in fact serving a class $m+1$ of customers with $c_{m+1} = 0$. If the result is true for nonidling policies, the fact that $\mu_{m+1}c_{m+1} = 0$ implies that the class $m+1$ shouls be served last, i.e. that an optimal policy will be nonidling.

Second, consider $\sum_{i=1}^m \frac{1}{\mu_i} \cdot N_i(R)$. The term $\frac{1}{\mu_i} \cdot N_i(R)$ is the expected time the server has to work in queue i in the steady state situation, given policy R. So, $\sum_{i=1}^m \frac{1}{\mu_i} \cdot N_i(R)$ is the expected service time for the whole system in the steady state situation, i.e. the average workload of the system. Using an argument as in Little's formula, this workload is independent of the policy. So, we write $W = \sum_{i=1}^m \frac{1}{\mu_i} \cdot N_i(R)$, from which we obtain $N_1(R) = \mu_1\{W - \sum_{i=2}^m \frac{1}{\mu_i} \cdot N_i(R)\}$.

Third, assume that $c_1\mu_1 \geq c_2\mu_2 \geq \cdots \geq c_m\mu_m$. Then, we have

$$\begin{aligned}
\textstyle\sum_{i=1}^m c_i \cdot N_i(R) &= c_1 \cdot N_1(R) + \textstyle\sum_{i=2}^m c_i \cdot N_i(R) \\
&= \mu_1 c_1 W + \textstyle\sum_{i=2}^m \{c_i - \frac{\mu_1 c_1}{\mu_i}\} \cdot N_i(R) \\
&= \mu_1 c_1 W + \textstyle\sum_{i=2}^m \frac{1}{\mu_i}(\mu_i c_i - \mu_1 c_1) \cdot N_i(R).
\end{aligned}$$

The coefficients of $N_i(R),\ i = 2, 3, \dots, m$ in the above expression are nonnegative. Hence, $\sum_{i=1}^m c_i \cdot N_i(R)$ is minimized by the policy that makes $N_i(R)$ as large as possible ($i = 2, 3, \dots, m$). Such a policy must necessarily serve a customer of queue 1 whenever it can.

Fourth, consider all the nonidling policies that serve queue 1 whenever it can. The set of times available for those policies to serve the other queues is the same for all these policies. One can check that all these policies have the same value of $\sum_{i=2}^m \frac{1}{\mu_i} \cdot N_i(R)$ (by Little's formula again). Repeating the above argument shows that among these policies, the ones that minimizes $\sum_{i=1}^m c_i \cdot N_i(R)$ must serve queue 2 whenever that can. Continuing in this way concludes the proof. □

8.5.3 Optimality of threshold policies

Waiting for a fast server or using a slow one

Customers arrive at a service facility that has two servers. The arrival times form a Poisson process with rate λ. The service times are assumed to be exponentially distributed with the respective rates μ_1 (for server 1) and μ_2 (for server 2), where $\mu_1 \geq \mu_2$. Service is nonpreemptive. When one of the servers becomes available, the decision has to be taken whether or not to send a customer to this server.

This is a customer assignment model. The model is not discrete, but continuous in time. Let $N^t(R)$ be the number of customers in the system at time t. As performance measure the total discounted costs are used, i.e. $\min_R \ \mathbb{E}\left\{\int_0^\infty \ e^{-\alpha t}N^t(R)dt\right\}$, where $\alpha > 0$, which is the continuous analogy of the total discounted costs in the discrete case.

The trade-off is between waiting for the fast server to become available and committing a customer to the slow queue. The next theorem shows that for this model an optimal *threshold policy* exists, namely server 1 will always be used when it becomes available, and the slower server, server 2, is only used when the total number of customers in the queue exceeds some threshold number N.

Theorem 8.13
There is some number N such that the optimal policy is to use the fast server all time and to send a customer to the slow server (when this slow server is available) if and only if the number of customers in the system at that time is at least N.

Proof
We give an outline of the proof. Decision times are services completion times and arrival times when at least one server is idle. We will rely on the fact that there is a stationary deterministic optimal policy. Consider a stationary deterministic policy. The policy cannot be optimal unless it uses the fast server whenever possible. If the policy uses the fast server whenever possible, then it is specified by a subset A of $\{1, 2, 3, \dots\}$ with the interpretation that a customer is sent to the slower server at decision times when that server is idle and when the queue length belongs to A. It then remains to show that the set A must be of the form $A = \{N, N + 1, N + 2, \dots\}$. This is done by contradiction.
Assume that the set A contains a "gap". That is, assume that $A = \{\dots, M, N, \dots, \dots\}$ with $N \geq M + 2$. Say that at $t = 0$ the fast server is busy, the slow server is idle and there are $M + 1$ customers waiting to be allocated to a server. The policy will then wait until the queue length reaches either M or N before sending a customer to the slow server. It is sufficient to show that the policy can be improved by sending a customer at time $t = 0$ to the slow server.
To see this, denote by σ the service time of that customer sent at time 0 to the slower server. This service time is known at time σ. Pretend that the policy corresponding to A was in fact used, by doing as if the customer had not been sent at time 0 but had been sent only when the queue length hits either M or N, at time τ, say, and by pretending that the slow server is busy during $[\tau, \tau + \sigma]$. This shows that the modified policy behaves as the one corresponding A, except that one customer leaves the queue at time σ instead of time $\tau + \sigma$.
It remains only to show that for all $M \geq 1$ there is some $N > M$ such that $N \in A$. Again, this can be shown by contradiction (the intuition is that if the que length is very large, than it is very likely that a customer at the end of the queue would be served by the slow sever before the fast server could become available). □

8.5.4 Optimality of join-the-shortest-queue policies

Customer allocation to parallel queues
Customers arrive at arbitrary known times at a system consisting of m identical $M/M/1$ queues in parallel. That is, the service times in all queues are independent and exponentially distributed with the same rate

μ. The problem is to choose, at each arrival time, which queue the arriving customer should join so as to minimize

$$min_R \ \mathbb{E}\Big\{\int_0^\infty \ e^{-\alpha t}\sum_{i=1}^m N_i^t(R)\,dt\Big\}, \tag{8.48}$$

where $\alpha > 0$ is the discount rate and $N_i^t(R)$ is the number of customers at time t in queue i, given policy R. The information available when the decision is made is the evaluation of the vector of queue length up to that time and the set of arrival times. It is assumed that the arrival times are such that $\mathbb{E}\{\int_0^\infty \ e^{-\alpha t}\sum_{i=1}^m \ N_i^t(R)\,dt\}$ is finite for at least one policy R.

An SQP (*shortest queue policy*) is a policy that sends each arriving customer to the shortest queue. In Theorem 8.14 it will be shown that an optimal SQP exists. It should be noted that an SQP is clearly *individually* optimal for each customer for arbitrary decisions of the customers who arrived before him. However, this does not imply that the policy is optimal *socially*, i.e. in the sense of minimizing $\mathbb{E}\big\{\int_0^\infty \ e^{-\alpha t}\sum_{i=1}^m \ N_i^t(R)\,dt\big\}$. Indeed, it is often the case that individuals have to accept sacrifices for the benefit of society at large. Mathematically, each customer should take into account not only the personnel cost of that customer (here, the discounted waiting time), but also the impact of the decision on the other customers (here, on those who will arrive behind him).

For the proof of this result we use the *forward induction method*. This method can be described as follows. Denote by X_t the state process corresponding to a policy and by Y_t the process corresponding to another policy. Suppose that there exists a partial ordering $\mathbf{B}$ on the set of possible states with the following two properties:

(1) it should be such that it is possible to prove that $X_t\,\mathbf{B}\,Y_t$ implies that $X_s\,\mathbf{B}\,Y_s$ for all $s \geq t$.

(2) the ordering should imply that the cost corresponding to X_t is not larger than the cost corresponding to Y_t.

Then it follows that the policy corresponding to X_t is an optimal policy.

Theorem 8.14

The customer allocation to parallel queues model has an optimal "join-the-shortest-queue" policy.

Proof

For two random variables V and W taking values in $\{0,1,2,\dots\}^m$ we write $V\,\mathbf{B}\,W$ if there exists two random variables V^* and W^* such that:

(a) V^* has the same distribution as V.

(b) W^* has the same distribution as W.

(c) $\mathbb{P}\{S_i(W^*) \geq S_i(V^*),\ 1 \leq i \leq m\} = 1$

where $S_i(V^*)$ denotes the sum of the i largest components of V^* and similarly for $S_i(W^*)$.

Denote by X_t the vector of queue lengths at time t corresponding to the SQP, and by Y_t the vector of queue lengths at time t corresponding to an arbitrary policy R. Assume that we have shown $X_t\,\mathbf{B}\,Y_t,\ t \geq 0$. Using the well known and easily verified fact that any $\{0,1,2,\dots\}$-valued random variable X satisfies $\mathbb{E}\{X\} = \sum_{k=0}^\infty \mathbb{P}\{X \geq k\}$, we obtain

$$\begin{aligned}
\mathbb{E}\big\{\textstyle\sum_{i=1}^m N_i^t(R)\big\} &= \mathbb{E}\big\{\textstyle\sum_{i=1}^m \{Y_t\}_i\big\} = \textstyle\sum_{k=0}^\infty \mathbb{P}\{\textstyle\sum_{i=1}^m \{Y_t\}_i \geq k\} \\
&= \textstyle\sum_{k=0}^\infty \mathbb{P}\{\textstyle\sum_{i=1}^m \{Y_t^*\}_i \geq k\} = \textstyle\sum_{k=0}^\infty \mathbb{P}\{S_m(Y_t^*) \geq k\} \\
&\geq \textstyle\sum_{k=0}^\infty \mathbb{P}\{S_m(X_t^*) \geq k\} = \textstyle\sum_{k=0}^\infty \mathbb{P}\{\textstyle\sum_{i=1}^m \{X_t^*\}_i \geq k\} \\
&= \textstyle\sum_{k=0}^\infty \mathbb{P}\{\textstyle\sum_{i=1}^m \{X_t\}_i \geq k\} = \mathbb{E}\big\{\textstyle\sum_{i=1}^m \{X_t\}_i\big\} \\
&= \mathbb{E}\big\{\textstyle\sum_{i=1}^m N_i^t(SQP)\big\}.
\end{aligned}$$

Hence, the cost (at any time t) in (8.48) corresponding to the SQP is not larger than the cost corresponding to policy R. Thus the partial order $\mathbf{B}$ has the second desired property mentioned in the description of the forward induction method.

To prove that $X_t \, \mathbf{B} \, Y_t \ t \geq 0$, let $0 = t_0 \leq t_1 < t_2 < t_3 < \cdots$ be the values of the arrival and potential service completion times. Assume that $X_t \, \mathbf{B} \, Y_t$ for some $t \geq 0$ (for $t = 0 \ X_t \, \mathbf{B} \, Y_t$ holds since $X_0 = Y_0$), where $t_{n-1} \leq t < t_n$ for some $n \geq 1$. It then suffices to show that $X_{t_n} \, \mathbf{B} \, Y_{t_n}$.

First consider the case when t_n is an arrival time. Let X_t^* and Y_t^* be such that the properties (a), (b) and (c), mentioned in the begin of the proof, hold. Notice that $S_m(X_{t_n}^*) = S_m(X_t)+1$, while $S_i(Y_{t_n}^*) = S_i(Y_t)+1$ for all $i \geq k$ if policy R sends the arriving customer to the kth largest queue. This shows $X_{t_n} \, \mathbf{B} \, Y_{t_n}$ for this case.

Next consider the case when the event t_n is a potential completion time. Define $X_{t_n}^*$ and $Y_{t_n}^*$ by deciding that if the potential service completion time occurs in the kth longest queue of X_t^*, then the same is true for Y_t^*. This modifies the joint distribution but not the marginals (here one uses the memoryless property of the exponential distribution, implying that the probability that a completion occurs in the kth longest queue is independent of k and the same is true for Y_t^*), so that (a) and (b) will hold for $V^* = X_{t_n}^*$ and $W^* = Y_{t_n}^*$.

To verify (c) one uses the fact that if the potential service completion occurs in the kth longest queue of X_t^*, then $S_i(X_{t_n}^*) = \begin{cases} S_i(X_t^*) & \text{if } i < k \\ S_i(X_t^*) - 1 & \text{if } i \geq k \end{cases}$ and $S_i(Y_{t_n}^*) = \begin{cases} S_i(Y_t^*) & \text{if } i < k \\ S_i(Y_t^*) - 1 & \text{if } i \geq k \end{cases}$

Hence, we conclude that $S_i(Y_{t_n}^*) \geq S_i(X_{t_n}^*)$ for $i = 1, 2, \ldots, m$. □

8.5.5 Optimality of LEPT and SEPT policies

Many results can be shown by the principle of dynamic programming. In this section we present several examples using the optimality equation of dynamic programming.

1. Guessing a diamond

A deck of 52 cards is to be turned over one at a time. Before each card is turned we are given the opportunity to say whether or not it will be a diamond. We are allowed to say that a card is a diamond only once. The objective is to maximize the probability of being correct.

Theorem 8.15

All the decisions rules that select at least one card before all the diamonds are turned over are optimal.

Proof

Denote by $v_n(m)$ the maximum probability when there are n cards left to be turned and when m cards of those n cards are diamonds. Obviously, $v_m(m) = 1, \ 1 \leq m \leq 13$ and $v_n(0) = 0, \ n \geq 1$. The first claim is that

$$v_n(m) = max\Big\{\frac{m}{n}, \frac{m}{n} v_{n-1}(m-1) + \frac{n-m}{n} v_{n-1}(m)\Big\}, \ n \geq 2, \ 1 \leq m \leq min\{n, 13\}. \qquad (8.49)$$

To prove this, notice that the first term in the maximization is the probability of being correct if the decision is to declare that the next card is a diamond. We will show that the second term gives the maximum probability of being correct if the decision is not to declare that the next card is a diamond. Indeed, in the latter case there are two possibilities. With probability $\frac{m}{n}$, the next card is a diamond, so there are $n-1$ cards left with $m-1$ diamonds, with a maximum probability of being correct equal to

$v_{n-1}(m-1)$. With probability $\frac{n-m}{n}$, the next card is not a diamond, and there $n-1$ cards left with m diamonds, with a maximum probability of being correct equal to $v_{n-1}(m)$.
The second claim is all the decisions rules that select at least one card before all the diamonds are turned over are optimal. It is sufficient to show that $v_n(m) = \frac{m}{n},\ n \geq 2,\ 1 \leq m \leq min\{n, 13\}$.
We apply induction on n ($n = 2$ is trivial). Assume that $v_{n-1}(m) = \frac{m}{n-1}$ for all $1 \leq m \leq min\{n-1, 13\}$. Then, $\frac{m}{n}v_{n-1}(m-1) + \frac{n-m}{n}v_{n-1}(m) = \frac{m(m-1)}{n(n-1)} + \frac{(n-m)m}{n(n-1)} = \frac{m}{n}$. □

2. Processing a set exponential jobs on parallel machines

A set of n jobs has to be processed, each by one of m identical processors. The jobs have independent and exponentially distributed service times with rates $\mu_1 \leq \mu_2 \leq \cdots \leq \mu_n$. The n jobs are ready to be processed at tome 0, thus there are no arrivals. Two objectives will be considered: the *expected makespan* $MS := \mathbb{E}\big\{\max\{T_1, T_2, \ldots, T_n\}\big\}$ and the *expected flowtime* $FT := \mathbb{E}\big\{\sum_{j=1}^{n} T_j\big\}$ where T_j is the completion time of job $j,\ j = 1, 2, \ldots, n$.

A $LEPT$ policy is a policy that, at time 0 and at each service completion allocates the jobs to available servers in the order $1, 2, \ldots, n$, i.e. largest expected processing times first ($LEPT$). A $SEPT$ policy is a policy that, at time 0 and at each service completion allocates the jobs to available servers in the order $n, n-1, \ldots, 1$, i.e. shortest expected processing times first ($SEPT$).
It can we shown that a $LEPT$ policy is optimal for MS, the expected makespan, and that a $SEPT$ policy is optimal for FT, the expected flowtime. We will sketch these results, using the optimality equation of dynamic programming, for the case of two processors ($m = 2$). Furthermore, we present an alternative proof for the optimality of the $LEFT$ policy.

Theorem 8.16

Consider a stochastic scheduling problem in which n jobs with exponential processing times are scheduled on two identical machines. Then, the expected makespan is minimized by the LEPT policy.

Proof (outline)

Assume that the $LEPT$ policy is optimal when there are at most $n-1$ jobs to process (this assumption is verified for $n = 2$). It will be shown to be optimal for n jobs. Let $MS(i)$ be the minimum makespan for the jobs $\{1, 2, \ldots, n\}\backslash\{i\},\ 1 \leq i \leq n$. By the hypothesis, this makespan $MS(i)$ is achieved by the $LEPT$ policy. We will conditioning on the first of the two jobs initially processed, say the jobs i and j. Notice that the minimum of the exponential distribution for the jobs i and j is also an exponential distribution with parameter $\mu_i + \mu_j$, and that the fractions $\frac{\mu_i}{\mu_i+\mu_j}$ and $\frac{\mu_j}{\mu_i+\mu_j}$ are the probabilities that job i and job j, respectively, is first completed job. At that completion time, the remaining service time for the other job is also an exponential distribution with the same parameter. Hence, we obtain

$$MS = min_{i<j}\Big\{\frac{1}{\mu_i+\mu_j} + \frac{\mu_i}{\mu_i+\mu_j}MS(i) + \frac{\mu_j}{\mu_i+\mu_j}MS(j)\Big\}, \tag{8.50}$$

or equivalently,

$$0 = min_{i<j}\big\{1 + \mu_i\{MS(i) - MS\} + \mu_j\{MS(j) - MS\}\big\}, \tag{8.51}$$

and the minimum in (8.51) is achieved by the same pair (i, j) as in (8.50). To show that $LEPT$ is also optimal when there are n jobs to process, one has to show that the minimum in (8.51) is achieved by $(i, j) = (1, 2)$. Let $D_{ij} = \mu_i\{MS(i) - MS\} - \mu_j\{MS(j) - MS\},\ i < j$. Then, it can be shown by induction on n that $D_{ij} \leq 0$ if $i < j$, implying the result. □

Theorem 8.17

Consider a stochastic scheduling problem in which n jobs with exponential processing times are scheduled on two identical machines. Then, the expected flowtime is minimized by the SEPT policy.

Proof (outline)

The proof has the same structure as the proof of Theorem 8.16. Assume that the *SEPT* policy is optimal when there are at most $n-1$ jobs to process (this assumption is verified for $n = 2$). It will be shown to be optimal for n jobs. Let $FT(i)$ be the minimum flowtime for the jobs $\{1, 2, \ldots, n\}\backslash\{i\},\ 1 \leq i \leq n$. By the hypothesis, this flowtime $FT(i)$ is achieved by the *SEPT* policy. We will conditioning on the first of the two jobs initially processed, say the jobs i and j. The completion time of the first completed job has expectation $\frac{1}{\mu_i+\mu_j}$, which will be part of each completion time $T_j,\ j = 1, 2, \ldots, n$. After that time the remaining $n-1$ jobs have exponential distributions with the original rates. Hence, we obtain

$$FT = min_{i<j}\Big\{\frac{n}{\mu_i+\mu_j} + \frac{\mu_i}{\mu_i+\mu_j}FT(i) + \frac{\mu_j}{\mu_i+\mu_j}FT(j)\Big\}, \tag{8.52}$$

or equivalently,

$$0 = min_{i<j}\big\{n + \mu_i\{FT(i) - FT\} + \mu_j\{FT(j) - FT\}\big\}, \tag{8.53}$$

and the minimum in (8.53) is achieved by the same pair (i, j) as in (8.52). To show that *SEPT* is also optimal when there are n jobs to process, one has to show that the minimum in (8.51) is achieved by $(i, j) = (n, n-1)$. This can be done in a similar way as in Theorem 8.16. □

Alternative proof for the optimality of the *LEFT* policy

It will help our analysis to assume that at time 0 one of the two processors is occupied on a job 0 and will remain accupied for a time X_0, where X_0 is assumed to have an arbitrary distribution and is independent of the other jobs. For any permutation $i_1, i_2, \ldots, i_n$ of $1, 2, \ldots, n$, putting the jobs on the processors in that order defines a schedule. Hence, is policy is a schedule $(0, i_1, i_2, \ldots, i_n)$. Let X_j be the stochastic duration of job $j,\ j = 0, 1, 2, \ldots, n$ and let D be the amount of time that only one of the processors is busy. That is, at time $MS - D$ one of the processors completes work on a job and finds no other jobs available. As the total amount of work processed is $M + (M - D) = \sum_{j=0}^{n} X_j$, Hence, minimizing the expected difference of the times at which the procesosrs become idle also leads to minimizing the expected makespan. The following lemma will be used to show that the *LEPT* policy is optimal.

Lemma 8.13

Consider the policies $R = (0, 2, 1, 3, 4, \ldots, n)$ *and* $R_* = (0, 1, 2, \ldots, n)$. *Then,* $\mathbb{E}_{R_*}\{D\} \leq \mathbb{E}_R\{D\}$.

Proof

Let $p(j)$ and $p_*(j),\ j = 0, 1, \ldots, n$ be the probabilities that the last job to be completed is job j, under policies R and R_*, respectively. Clearly, $p(0) = p_*(0) = \mathbb{P}\{X_0 > \sum_{j=1}^{n} X_j\}$. We shall prove by induction on n that

$$p_*(1) \leq p(1) \text{ and } p_*(j) \geq p(j),\ j = 2, 3, \ldots, n. \tag{8.54}$$

This is obvious if $n = 1$ ($p_*(1) = p(1) = \mathbb{P}\{X_0 \leq X_1\}$). Assume (8.54) is true whenever there are only $n-1$ jobs (in addition to job 0) to be scheduled, and let $q_*(j)$ and $q(j)$ be the probabilities that job j is the last of jobs $0, 1, 2, \ldots, n-1$ under policies R^* and R, respectively. Then, by the induction hypothesis

$$q_*(1) \leq q(1) \text{ and } q_*(j) \geq q(j),\ j = 2, 3, \ldots, n-1. \tag{8.55}$$

Now consider the n-job case. However, using the lack of memory of the exponential distribution and the fact that job n is the last to begin processing under both policies, we have

$$p(j) = q(j) \cdot \tfrac{\mu_n}{\mu_n+\mu_j},\ p_*(j) = q_*(j) \cdot \tfrac{\mu_n}{\mu_n+\mu_j},\ j = 1, 2, \ldots, n-1.$$

Hence, from (8.55), we obtain $p_*(1) \leq p(1)$ and $p_*(j) \geq p(j),\ j = 2, 3, \ldots, n-1$. Finally, using

$$p(n) = 1 - \sum_{j=0}^{n-1} p(j) = 1 - \sum_{j=0}^{n-1} q(j) \cdot \{1 - \tfrac{\mu_j}{\mu_n+\mu_j}\} = \sum_{j=0}^{n-1} q(j) \cdot \tfrac{\mu_j}{\mu_n+\mu_j}$$

and similarly $p_*(n) = \sum_{j=0}^{n-1} q_*(j) \cdot \frac{\mu_j}{\mu_n+\mu_j}$, one can write

$$\begin{aligned} p_*(n) - p(n) &= \sum_{j=0}^{n-1} \{p_*(j) - p(j)\} = \sum_{j=0}^{n-1} \{q_*(j) - q(j)\} \cdot \tfrac{\mu_j}{\mu_n+\mu_j} \\ &= \{q_*(1) - q(1)\} \cdot \tfrac{\mu_1}{\mu_n+\mu_1} + \sum_{j=2}^{n-1} \{q_*(j) - q(j)\} \cdot \tfrac{\mu_j}{\mu_n+\mu_j} \\ &\geq \tfrac{\mu_1}{\mu_n+\mu_1} \sum_{j=1}^{n-1} \{q_*(j) - q(j)\} = 0, \end{aligned}$$

where the inequality follows because $\mu_j \geq \mu_1$ implies that $\frac{\mu_j}{\mu_n+\mu_j} \geq \frac{\mu_1}{\mu_n+\mu_1}$.

Consider any policy $\pi = (0, i_1, i_2, \ldots, i_n)$ and assume that job $j,\ j \geq 1$, is the last job to be completed. Since at time $M - D$ job j is the last job to be completed, the remaining processing time is exponential distributed with rate μ_j, so we have $\mathbb{E}_\pi\{D \mid \text{job } j \text{ is the last job to be completed}\} = \frac{1}{\mu_j},\ j = 1, 2, \ldots, n$. Furthermore, $\mathbb{E}_\pi\{D \mid \text{job 0 is the last job to be completed}\} = \mathbb{E}_\pi\{X_0 - \sum_{j=1}^{n} X_j \mid X_0 > \sum_{j=1}^{n} X_j\}$.

Therefore, one can write

$$\begin{aligned} \mathbb{E}_\pi\{D\} &= \sum_{j=0}^{n} p(j) \cdot \mathbb{E}_\pi\{D \mid \text{job } j \text{ is the last job to be completed}\} \\ &= \sum_{j=1}^{n} \tfrac{p(j)}{\mu_j} + p(0) \cdot \mathbb{E}_\pi\{X_0 - \sum_{j=1}^{n} X_j \mid X_0 > \sum_{j=1}^{n} X_j\} \end{aligned}$$

Hence, we have

$$\begin{aligned} \mathbb{E}_{R_*}\{D\} - \mathbb{E}_R\{D\} &= \sum_{j=1}^{n} \tfrac{1}{\mu_j}\{p_*(j) - p(j)\} = \tfrac{1}{\mu_1}\{p_*(1) - p(1)\} + \sum_{j=2}^{n} \tfrac{1}{\mu_j}\{p_*(j) - p(j)\} \\ &\leq \tfrac{1}{\mu_1}\{p_*(1) - p(1)\} + \sum_{j=2}^{n} \tfrac{1}{\mu_1}\{p_*(j) - p(j)\} = \tfrac{1}{\mu_1} \sum_{j=1}^{n} \{p_*(j) - p(j)\} \\ &= \tfrac{1}{\mu_1}\{\big(1 - p_*(0)\big) - \big(1 - p(0)\big)\} = 0. \end{aligned}$$

□

Note

From the proof of Lemma 8.13 it follows that the lemma is true for any order of the jobs in which $\mu_1 = min_{1 \leq j \leq n}\, \mu_j$.

Theorem 8.18

The LEFT policy is optimal.

Proof

Consider an arbitrary policy that does not initially process 1, say policy $(0, i_1, i_2, \ldots, i_k, i_{k+1}, 1, \ldots)$. By considering this at the time at which only one of the jobs $0, i_1, i_2, \ldots, i_k$ have not yet finished its processing, we see, using Lemma 8.13, that the schedule $(0, i_1, i_2, \ldots, i_k, 1, i_{k+1}, \ldots)$ has a smaller expected makespan. Continuing in this way we see that $(0, 1, i_1, i_2, \ldots, i_k, i_{k+1}, \ldots)$ is better. If $i_2 \neq 2$ then, repeating this argument, we show that $(0, 1, 2, i_1, i_2, \ldots, i_k, i_{k+1}, \ldots)$ is better. Continuing in this matter shows that the policy $(0, 1, 2, \ldots, n)$ is optimal. Since we may use for job 0 a job with processing time $X_0 = 0$, we have shown that the *LEPT* policy is optimal.

□

Remark

Whereas Theorem 8.18 only proved that scheduling tasks in increasing order of their exponential service rates is optimal among the $n!$ policies that determine their ordering in advance, it is also optimal among all policies. That is, it remains optimal even when the choice of tasks to begin processing is allowed to depend on what has occurred up to that time. This is shown by induction as follows. It is immediate when $n = 1$, so assume it to be true whenever there are $n - 1$ tasks to be processed. Now, whichever of the n tasks is initially processed (alongside task 0), at the moment one of the two processors becomes free, it follows by the induction hypothesis that the remaining tasks should be scheduled in increasing order of their rates. Hence, the only policies we need consider are those n policies for which task i ($i = 1, 2, \dots, n$) is scheduled first, and the remaining tasks are scheduled in increasing order of their rates. But Theorem 8.18 shows that the optimal policy of this type is the one that schedules the n tasks in increasing order of their rates. This completes the induction.

Stochastic ordering

We say that the random variable $X \geq_{st} Y$ if $\mathbb{P}\{X > a\} \geq \mathbb{P}\{Y > a\}$ for all a.

Lemma 8.14

If $X \geq_{st} Y$, then $\mathbb{E}\{X\} \geq \mathbb{E}\{Y\}$.

Proof

Assume first that X and Y are nonnegative random variables. Then,

$$\mathbb{E}\{X\} = \int_0^\infty \mathbb{P}\{X > x\}dx \geq \int_0^\infty \mathbb{P}\{Y > x\}dx = \mathbb{E}\{X\}.$$

In general, we can write any random variable Z as the difference of two nonnegative random variables as $Z = Z^+ - Z^-$, where $Z^+ := \begin{cases} Z & \text{if } Z \geq 0 \\ 0 & \text{if } Z < 0 \end{cases}$ and $Z^- := \begin{cases} 0 & \text{if } Z \geq 0; \\ -Z & \text{if } Z < 0. \end{cases}$

We leave it as an exercise (see Exercise 8.6) to show that $X \geq_{st} Y$ implies $X^+ \geq_{st} Y^+$ and $X^- \leq_{st} Y^-$. Hence, $\mathbb{E}\{X\} = \mathbb{E}\{X^+\} - \mathbb{E}\{X^-\} \geq \mathbb{E}\{Y^+\} - \mathbb{E}\{Y^-\} = \mathbb{E}\{Y\}$. □

Lemma 8.15

$X \geq_{st} Y \;\Leftrightarrow\; \mathbb{E}\{f(X)\} \geq \mathbb{E}\{f(Y)\}$ *for all nondecreasing functions f.*

Proof

Suppose first that $X \geq_{st} Y$ and let f be an nondecreasing function. Then it is, by Lemma 8.14, sufficient to show that $f(X) \geq_{st} f(Y)$. Letting $f^{-1}(a) = inf\{x \mid f(x) > a\}$, we have

$$\mathbb{P}\{f(X) > a\} = \mathbb{P}\{X > f^{-1}(a)\} \geq \mathbb{P}\{Y > f^{-1}(a)\} = \mathbb{P}\{f(Y) > a\}.$$

Now suppose that $\mathbb{E}\{f(X)\} \geq \mathbb{E}\{f(X)\}$ for all nondecreasing functions f. For any a, let f_a be the nondecreasing function $f_a(x) := \begin{cases} 1 & \text{if } x > a; \\ 0 & \text{if } x \leq a. \end{cases}$

Then, because $\mathbb{E}\{f_a(X)\} = \mathbb{P}\{X > a\}$ and $\mathbb{E}\{f_a(Y)\} = \mathbb{P}\{Y > a\}$, we see from $\mathbb{E}\{f_a(X)\} \geq \mathbb{E}\{f_a(Y)\}$ that $\mathbb{P}\{X > a\} \geq \mathbb{P}\{Y > a\}$, i.e. $X \geq_{st} Y$. □

Remark

It can be shown that the policy given in Theorem 8.18 has the property that it stochastically minimizes the makespan. That is, that for any a, the probability that the makespan exceeds a is minimized by this policy. This is a stronger result than that in Theorem 8.18, which states only that the policy minimizes the expected makespan. In addition, it can also be shown that the stated policy stochastically maximizes the time until one of the processors becomes idle. That is, in the notation of this section, it maximizes the probability that $M - D$ exceeds a for each a.

8.5.6 Maximizing finite-time returns on two processors

Consider the same model as in section 8.5.1, but now there are two servers. It follows as in (8.47) that the total expected return under any policy R can be expressed as

$$\mathbb{E}_R\{\text{total return}\} = \sum_{i=1}^{n} \mu_i r_i \, \mathbb{E}_R\{\text{length of time job } i \text{ is worked on}\}. \tag{8.56}$$

Thus, at first glance, it might seem that an optimal policy would be to sequence the tasks in decreasing order of $\mu_i r_i$, as in the case when there is only a single server. To see that this need not be the case, suppose that $\mu_i r_i = 1,\ i = 1, 2, \ldots, n$. Then, the conjecture would imply that all orderings are optimal. Further, assume that $\mu_1 < \mu_2 < \cdots < \mu_n$. The expected return by time T for any policy is equal to the expected total processing time on all tasks by T. Because the $LEPT$ policy uniquely stochastically maximizes the time until one of the processors becomes idle (uniquely because the rate are strictly increasing, see the proof of Theorem 8.13), it also uniquely stochastically maximizes the total processing time by T and is thus uniquely optimal under our new objective function. However, this contradicts the conjecture that it is optimal to process tasks in decreasing order of $\mu_i r_i$, for, in the case $\mu_i r_i = 1$ for all i, this conjecture implies that all orderings are optimal. We can, however, prove that the policy that works on the jobs in decreasing order of $\mu_i r_i$ is optimal in a special case. In order to prove this special case we need the following lemma.

Lemma 8.16
Let $T_1, T_2, \ldots, T_n,\ S_1, S_2, \ldots, S_n$ and $c_1, c_2, \ldots, c_n$ be nonnegative numbers such that $\sum_{i=1}^{j} T_i \geq \sum_{i=1}^{j} S_i$, for $j = 1, 2, \ldots, n$ and $c_1 \geq c_2 \cdots \geq c_n \geq 0$. Then, $\sum_{i=1}^{n} c_i T_i \geq \sum_{i=1}^{n} c_i S_i$.

Proof
Let $T_{n+1} = 0,\ T = \sum_{i=1}^{n+1} T_i,\ S_{n+1} = T - \sum_{i=1}^{n} S_i \geq 0$. Also, let X and Y be random variables such that $\mathbb{P}\{X = i\} = \frac{T_i}{T},\ \mathbb{P}\{Y = i\} = \frac{S_i}{T},\ i = 1, 2, \ldots, n+1$. Now the hypothesis of the lemma states that

$$\mathbb{P}\{X \leq j\} = \tfrac{1}{T}\textstyle\sum_{i=1}^{j} T_i \geq \tfrac{1}{T}\sum_{i=1}^{j} S_i = \mathbb{P}\{X \leq j\} \text{ for } j = 1, 2, \ldots, n+1, \text{ i.e. } X \leq_{st} Y.$$

Let $c_{n+1} = 0$, then c is a nonincreasing function. Hence, it follows from Lemma 8.15 that $\mathbb{E}\{c_X\} \geq \mathbb{E}\{c_Y\}$, implying $\sum_{i=1}^{n} c_i T_i \geq \sum_{i=1}^{n} c_i S_i$. □

Theorem 8.19
If $\mu_1 \leq \mu_2 \leq \cdots \leq \mu_n$ and $\mu_1 r_1 \geq \mu_2 r_2 \geq \cdots \geq \mu_n r_n \geq 0$, then sequencing the tasks in the order $1, 2, \ldots, n$ maximizes the expected return by T for each $T > 0$.

Proof
Fix T and let T_j denote the expected total processing time of task j by time T. Now, because the policy that sequences according to $1, 2, \ldots, n$ stochastically maximizes the time until one of the processors becomes idle, it follows that it also stochastically maximizes the total processing time by T. Because this remains true even when the set of tasks is $1, 2, \ldots, j$, it follows that $\sum_{i=1}^{j} T_i$ is, for each j maximized by the plicy under consoderation. The result follows now from Lemma 8.16 with $c_i = \mu_i r_i$ for all i. □

8.5.7 Tandem queues

Each of n jobs needs to be processed on two machines, say A and B. After receiving service on machine A, a job moves to machine B, and upon completion time of service at B it leaves the system. Let A_j and

B_j be the service time of job j on machine A and B respectively. The objective is to determine the order in which to process jobs at machine A to minimize the expected time until all jobs have been processed on both machines. For the deterministic case, Johnson ([144]) shows that the makespan is minimized if jobs are arranged in the following transitive order on both machines:

$$\text{job } i \text{ precedes job } j \;\Leftrightarrow\; min\{A_i, B_j\} \le min\{A_j, B_i\}.$$

Here we assume that A_j and B_j are exponentially distributed with rates λ_j and μ_j respectively. Then,

$$\mathbb{E}\left\{min\{A_i, B_j\}\right\} = \tfrac{1}{\lambda_i+\mu_j} \text{ and } \mathbb{E}\left\{min\{A_j, B_i\}\right\} = \tfrac{1}{\lambda_j+\mu_i}.$$

Taking expectations on both sides of Johnson's rule one obtains the rule

$$\text{job } i \text{ precedes job } j \;\Leftrightarrow\; \lambda_i - \mu_i \ge \lambda_j - \mu_j.$$

We will show that Johnson's rule is also optimal for exponential processing times.

First, to gain some insight, let us consider the case in $n = 2$. If job 1 is processed first on machine A, then the expected completion time, denoted by $\mathbb{E}\{C_{1,2}\}$ is given by

$$\mathbb{E}\{C_{1,2}\} = \frac{1}{\lambda_1} + \frac{1}{\mu_1+\lambda_2} + \frac{\mu_1}{\mu_1+\lambda_2}\cdot\left\{\frac{1}{\lambda_2}+\frac{1}{\mu_1}\right\} + \frac{\lambda_2}{\mu_1+\lambda_2}\cdot\left\{\frac{1}{\mu_1}+\frac{1}{\mu_2}\right\}.$$

This follows because $\frac{1}{\lambda_1}$ is the expected time until job 1 is completed on machine A, at which time job 1 goes to machine B and job 2 goes to A. Then $\frac{1}{\mu_1+\lambda_2}$ is the expected time either job 2 is completed at A (with probability $\frac{\lambda_2}{\mu_1+\lambda_2}$) either job 1 is completed at B (with probability $\frac{\mu_1}{\mu_1+\lambda_2}$). The other two terms are then obtained by conditioning on whichever occurs first.

Similarly, by reversing the order we have that $\mathbb{E}\{C_{2,1}\}$ is given by

$$\mathbb{E}\{C_{2,1}\} = \frac{1}{\lambda_2} + \frac{1}{\mu_2+\lambda_1} + \frac{\mu_2}{\mu_2+\lambda_1}\cdot\left\{\frac{1}{\lambda_1}+\frac{1}{\mu_2}\right\} + \frac{\lambda_1}{\mu_2+\lambda_1}\cdot\left\{\frac{1}{\mu_2}+\frac{1}{\mu_1}\right\}.$$

With some algebra, left to the reader (see Exercise 8.7) one can show that

$$\mathbb{E}\{C_{1,2}\} \le \mathbb{E}\{C_{2,1}\} \;\Leftrightarrow\; \lambda_1 - \mu_1 \ge \lambda_2 - \mu_2,$$

i.e. Johnson's rule is true. We now show that this remains true when there are more than two jobs. With A_j the processing time for job j on machine A and C the time until all jobs have been processed on both machines, then $R := C - \sum_{j=1}^{n} A_j$ is the *remainder time*, that is, it represents the amount of work that remains at machine B when machine A has completed its processing.

Hence, $\mathbb{E}\{R\} = \mathbb{E}\{C\} - \sum_{j=1}^{n} \frac{1}{\lambda_j}$, so minimizing $\mathbb{E}\{C\}$ is equivalent to minimizing $\mathbb{E}\{R\}$. We shall prove that the policy that schedules jobs at machine A in decreasing order of $\lambda_j - \mu_j$ minimizes $\mathbb{E}\{R\}$. In fact, we shall use an interchange argument to show that this ordering stochastically minimizes R, and thus minimizes $\mathbb{E}\{R\}$.

Consider first the case $n = 2$, and suppose that, initially at time $t = 0$, machine B is occupied with the amount work w. That is, B must spend w units working on prior work before it can start processing either job 1 or job 2. Let $R_{1,2}(w)$ the remainder, i.e. $R_{1,2}(w) = C - A_1 - B_1$, when job 1 is scheduled first, and similarly for $R_{2,1}(w)$. The following lemma shows that the suggested ordering stochastically minimizes $R(w)$ for any w.

Lemma 8.17

If $\lambda_1 - \mu_1 \ge \lambda_2 - \mu_2$, then for any w, $R_{1,2}(w) \le_{st} R_{2,1}(w)$.

Proof

We have to compare $\mathbb{P}\{R_{1,2}(w) > a\}$ with $\mathbb{P}\{R_{2,1}(w) > a\}$. When $w \ge A_1 + A_2$, then there probabilities are equal, because in either cases $R = w + B_1 + B_2 - A_1 - A_2$, with B_j the processing time for job j

on machine B, $j = 1, 2$. Hence, we need only look at $\mathbb{P}\{R_{1,2}(w) > a \mid A_1 + A_2 > w\}$. Now, statingthat $A_1 + A_2 > w$ is equivalent to stating that at some time job 1 will be in machine B and job 2 in machine A. Hence, using the lack of memory of the exponential distribution, and conditioning on which machine finishes first, we see that

$$\begin{aligned}
\mathbb{P}\{R_{1,2}(w) > a \mid A_1 + A_2 > w\} &= \tfrac{\mu_1}{\mu_1+\lambda_2} e^{-\mu_2 a} + \tfrac{\lambda_2}{\mu_1+\lambda_2}\, \mathbb{P}\{e^{\mu_1} + e^{\mu_2} > a\} \\
&= \tfrac{\mu_1}{\mu_1+\lambda_2} e^{-\mu_2 a} + \tfrac{\lambda_2}{\mu_1+\lambda_2} \Big\{ e^{-\mu_1 a} + \textstyle\int_0^a \mu_1 e^{-\mu_1 x} e^{-\mu_2 (a-x} \, dx \Big\} \\
&= \tfrac{\mu_1}{\mu_1+\lambda_2} e^{-\mu_2 a} + \tfrac{\lambda_2}{\mu_1+\lambda_2} \Big\{ e^{-\mu_1 a} + \mu_1 e^{-\mu_2 a} \textstyle\int_0^a e^{-(\mu_1-\mu_2)x} \, dx \Big\} \\
&= \tfrac{\mu_1}{\mu_1+\lambda_2} e^{-\mu_2 a} + \tfrac{\lambda_2}{\mu_1+\lambda_2} \Big\{ e^{-\mu_1 a} + \tfrac{\mu_1}{\mu_1-\mu_2} e^{-\mu_2 a} \cdot \{1 - e^{-(\mu_1-\mu_2)a}\} \Big\} \\
&= \tfrac{\mu_1}{\mu_1+\lambda_2} e^{-\mu_2 a} + \tfrac{\lambda_2}{\mu_1+\lambda_2} \cdot \tfrac{1}{\mu_1-\mu_2} \Big\{ \mu_1 e^{-\mu_2 a} + \mu_2 e^{-\mu_1 a} \Big\} \\
&= \tfrac{\mu_1(\mu_1-\mu_2+\lambda_2)e^{-\mu_2 a} - \mu_2\lambda_2 e^{-\mu_1 a}}{(\mu_1+\lambda_2)((\mu_1-\mu_2)}.
\end{aligned}$$

Because the expression $\mathbb{P}\{R_{2,1}(w) > a \mid A_1 + A_2 > w\}$ is similar, we have

$$\mathbb{P}\{R_{2,1}(w) > a \mid A_1 + A_2 > w\} \quad = \quad \tfrac{\mu_2(\mu_2-\mu_1+\lambda_1)e^{-\mu_1 a} - \mu_1\lambda_1 e^{-\mu_2 a}}{(\mu_2+\lambda_1)((\mu_2-\mu_1)}.$$

Hence, we see that

$$\begin{aligned}
&\mathbb{P}\{R_{2,1}(w) > a \mid A_1 + A_2 > w\} - \mathbb{P}\{R_{1,2}(w) > a \mid A_1 + A_2 > w\} \\
&\qquad = \tfrac{\mu_1\mu_2}{(\mu_1+\lambda_2)(\mu_2+\lambda_1)} \cdot \tfrac{e^{-\mu_1 a} - e^{-\mu_2 a}}{\mu_2-\mu_2} \cdot \{(\lambda_1 - \mu_1) - (\lambda_2 - \mu_2)\} \;\geq\; 0,
\end{aligned}$$

which completes the proof of this lemma. □

Theorem 8.20

For any initial workload of machine B, R is stochastically minimized, and thus $\mathbb{E}\{C\}$ is minizized, by scheduling jobs to be processed on A in decreasing order of $\lambda_j - \mu_j$.

Proof

Consider first any of the $n!$ policies in which the ordering is fixed at time 0. Furthermore, suppose that $\lambda_1 - \mu_1 = max_j\,\{\lambda_j - \mu_j\}$ and that the ordering calls for job j on A immediately before job 1. Then at the moment at which machine A is to begin on job j, no matter what the remaining work is at machine B at that moment, it follows from Lemma 8.17 that, if we interchange the jobs 1 and j, then the remaining work at machine B when both 1 and j have been processed at A will be stochastically reduced. But it is obvious that, for a given set of jobs to be processed in both machines, the remainder time is a stochastically increasing function of the initial workload of machine B.

Hence, the remainder time is stochastically reduced by the interchange. Repeated use of this interchange argument shows that the suggested policy stochastically minimizes the remainder time among all the $n!$ policies whose ordering is fixed at time 0. Hence, it minimizes the expexted completion time among all such policies.

To show that it is optimal amomg all policies follows by induction (it is immediate for $n = 1$). Assume it whenever there are $n - 1$ jobs to be processed on the two machines no matter the initial workload of machine B. Now no matter which job is initially processed at machine A, at the moment its processing at A is completed, it follows by the induction hypothesis that the remaining jobs are processed in decreasing order of the difference of their rates at machines A and B. Hence, we need only consider fixed-order policies, and thus this policy is optimal. □

8.6 Multi-armed bandit problems

8.6.1 Introduction

The multi-armed bandit problem was introduced in Section 1.3.9. The state space S is the Cartesian product $S = S_1 \times S_2 \times \cdots \times S_n$. Each state $i = (i_1, i_2, \dots, i_n)$ has the same action set $A = \{1, 2, \dots, n\}$, where action k means that project k is chosen, $k = 1, 2, \dots, n$. So, at each stage one can be working on exactly one of the projects. When project k is chosen in state i - the chosen project is called the *active project* - the immediate reward and the transition probabilities only depend on the active project, whereas the states of the remaining projects are frozen. Let r_{i_k} and $p_{i_k j},\ j \in S_k$, denote these quantities when action k is chosen. As a utility function the total discounted reward is chosen.

Example 8.2
Consider three sequences of nonnegative numbers, denoted by $\{x_n^1,\ n = 1, 2, 3, \dots\}$, $\{x_n^2,\ n = 1, 2, 3, \dots\}$ and $\{x_n^3,\ n = 1, 2, 3, \dots\}$, respectively. At each time one selects one of the sequences and x_n^k is the reward obtained the n-th time that sequence k is chosen. Denote by R_t the reward at time t. The problem is to find the *optimal order* in which the sequences are chosen so as to maximize $R = \sum_{t=1}^{\infty} \alpha^{t-1} R_t$, where $\alpha \in (0, 1)$ is a discount factor such that $\sum_{n=1}^{\infty} \alpha^{t-1} x_n^k < \infty$ for $k = 1, 2, 3$.
This is a deterministic version of the multi-armed bandit problem with state space $S = S_1 \times S_2 \times S_3$, where $S_i := \{0, 1, 2, \dots\}$. The state (i_1, i_2, i_3) means that sequence k was chosen i_k times, $k = 1, 2, 3$; $r_i(k) := x_{i+1}^k$ and $p_{ij}(k) := 1$ for $j = i + 1$ (the other transition probabilities are 0).
Consider sequence k and assume that it has been selected $n_k - 1$ times, so that the next reward from this sequence is $x_{n_k}^k$. Define $G_k(n_k)$ by

$$G_k(n_k) := \sup_{\tau \geq n_k} \frac{\sum_{t=n_k}^{\tau} \alpha^{t-1} x_t^k}{\sum_{t=n_k}^{\tau} \alpha^{t-1}},\ k = 1, 2, 3. \tag{8.57}$$

The interpretation is that $G_k(n_k)$ is the *maximum discounted reward per unit of discounted time* that can be obtained from the remainder of sequence k. The numbers $G_k(n_k)$ are called the *Gittins indices.* We shall show that the policy that selects in state $(i_1 = n_1 - 1, i_2 = n_2 - 1, i_3 = n_3 - 1)$ the sequence with the largest of the indices $G_1(n_1),\ G_2(n_2),\ G_3(n_3)$ is optimal. Such policy is called an *index policy.* Notice that the calculation of the indices is done sequence by sequence. This result is a *decomposition* of the original problem.

8.6.2 A single project with a terminal reward

Consider the one-armed bandit problem with stopping option, i.e. in each state there are two options: action 1 is the stopping option and then one earns a terminal reward M and by action 2 the process continue with in state i an immediate reward r_i and transition probabilities p_{ij}. Let $v^\alpha(M)$ be the value vector of this optimal stopping problem. Then, $v^\alpha(M)$ is the unique solution of the optimality equation (cf. section 4.12.2)

$$v_i^\alpha(M) = max\{M,\, r_i + \alpha \sum_j p_{ij} v_j^\alpha(M)\},\ i \in S \tag{8.58}$$

and of the linear program

$$min \left\{ \sum_j v_j \;\middle|\; \begin{array}{rcll} \sum_j \{\delta_{ij} - \alpha p_{ij}\} v_j & \geq & r_i, & i \in S \\ v_i & \geq & M, & i \in S \end{array} \right\}. \tag{8.59}$$

Furthermore, we have shown in section 4.12.2 the following result.

Theorem 8.21
Let (x, y) be an extreme optimal solution of the dual program of (8.59), i.e.

$$max \left\{ \sum_j r_i x_i + M \cdot \sum_j y_i \;\middle|\; \begin{array}{rcll} \sum_i \{\delta_{ij} - \alpha p_{ij}\} x_i + y_j & = & 1, & i \in S \\ x_i, y_i & \geq & 0, & i \in S \end{array} \right\}. \tag{8.60}$$

Then, the policy f^∞ such that $f(i) := \begin{cases} 2 & \text{if } x_i > 0 \\ 1 & \text{if } x_i = 0 \end{cases}$ is an optimal policy.

Lemma 8.18
$v_i^\alpha(M) - M$ is a nonnegative continuous nonincreasing function in M, for all $i \in S$.

Proof
The nonnegativity of $v_i^\alpha(M) - M$ is directly from (8.58). By the method of value iteration $v^\alpha(M)$ can be approximated, arbitrary close, by

$$v_i^1(M) := M,\ i \in S;\ v_i^{n+1}(M) := \max\{M,\, r_i + \alpha \textstyle\sum_j p_{ij} v_j^n(M)\},\ i \in S,\ n = 1, 2, \dots.$$

Hence, $v^\alpha(M) - M \cdot e$ can be approximated, arbitrary close, by

$$w_i^1(M) := 0,\ i \in S;\ w_i^{n+1}(M) := \max\{0,\, r_i + \alpha \textstyle\sum_j p_{ij} w_j^n(M) - (1 - \alpha)M\},\ i \in S,\ n = 1, 2, \dots.$$

By induction on n it is easy to see that $w_i^n(M)$ is continuous and nonincreasing in M. Hence, for all $i \in S$, the limit $v_i^\alpha(M) - M$ is also a continuous and nonincreasing function in M. □

Define the Gittins indices $G_i,\ i \in S$, by

$$G_i := \min\{M \mid v_i^\alpha(M) = M\}. \tag{8.61}$$

Hence, $v_i^\alpha(G_i) = G_i$ and, by Lemma 8.18, $v_i^\alpha(M) = M$ for all $M \geq G_i$.

Theorem 8.22
For any M, the policy $f^\infty \in C(D)$ which chooses the stopping action in state i if and only if $M \geq G_i$ is optimal.

Proof
Take any M and let (x, y) be an extreme optimal solution of the dual linear program (8.60). From Theorem 8.21 we see that if $y_i = 0$ (and consequently $x_i > 0$), then the action 'continue' is optimal; when $y_i > 0$ (and consequently $x_i = 0$), then it is optimal to stop in state i.
First, let $M < G_i$, i.e. $v_i^\alpha(M) > M$. Then, by the complementary slackness property of linear programming $y_i = 0$ and it is optimal to continue in state i.
Next, let $M \geq G_i$, implying that $v_i^\alpha(M) = M$. Suppose that the stopping action is not optimal. Then, $v_i^\alpha(M) = r_i + \alpha \sum_j p_{ij} v_j^\alpha(M) > M$, which yields a contradiction. Therefore, it is optimal to stop in state i. □

For $M = G_i$ both actions (stop or continue) are optimal. Hence, an interpretion of the Gittins index G_i is that it is the terminal reward under which in state i both actions are optimal. Therefore, this number is also called the *indifference value.*

8.6.3 Multi-armed bandits

Consider the multi-armed bandit model with an additional option (action 0) in each state. Action 0 is a stopping option and then one earns a terminal reward M. For each state $i = (i_1, i_2, \dots, i_n)$ we denote i-th component of the value vector by $v_i^\alpha(M)$.

Lemma 8.19
$v_i^\alpha(M)$ is a nondecreasing, convex function in M, for all $i \in S$.

Proof
Choose a fixed state i. It is obvious that $v_i^\alpha(M)$ is a nondecreasing function in M. Consider an arbitrary policy $f^\infty \in C(D)$, and let $\tau(f)$ be the corresponding stopping time, i.e. the stochastic number of steps before stopping. Then, one can write

$$\begin{array}{rcl} v_i^\alpha(f^\infty, M) & = & \mathbb{E}_{i,f}\left\{\text{discounted reward until time } \tau(f) + M \cdot \alpha^{\tau(f)}\right\} \\ & = & \mathbb{E}_{i,f}\left\{\text{discounted reward until time } \tau(f)\right\} + M \cdot \mathbb{E}_{i,f}\left\{\alpha^{\tau(f)}\right\}. \end{array}$$

Hence,

$$v_i^\alpha(M) = max_{f^\infty \in C(D)} \Big\{ \mathbb{E}_{i,f}\left\{\text{discounted reward until time } \tau(f)\right\} + M \cdot \mathbb{E}_{i,f}\left\{\alpha^{\tau(f)}\right\}\Big\} \tag{8.62}$$

Since $v_i^\alpha(M)$ is the maximum of a finite number of terms, each of which is linear in M, $v_i^\alpha(M)$ is a convex function in M. □

Technical remark:
Since $v_i^\alpha(M)$ is the maximum of a finite number of linear functions, $\frac{\partial}{\partial M} v_i^\alpha(M)$ exists at almost all values of M.

Lemma 8.20
Let i be a fixed initial state and let $\tau(M)$ be the stopping time under the optimal policy $f^\infty(M)$, where M is the terminal reward. Then $\frac{\partial}{\partial M} v_i^\alpha(M) = \mathbb{E}_{i,f(M)}\{alpha^{\tau(M)}\}$.

Proof
Choose any $\varepsilon > 0$. If we employ $f^\infty(M)$ for a problem having terminal reward $M + \varepsilon$, we receive

$$\mathbb{E}_{i,f(M)}\left\{\text{discounted reward until time } \tau(M)\right\} + (M+\varepsilon) \cdot \mathbb{E}_{i,f(M)}\left\{\alpha^{\tau(M)}\right\}.$$

From (8.62) it follows that

$$\begin{array}{rcl} v_i^\alpha(M+\varepsilon) & \geq & \mathbb{E}_{i,f(M)}\left\{\text{discounted reward until time } \tau(M)\right\} + (M+\varepsilon) \cdot \mathbb{E}_{i,f(M)}\left\{\alpha^{\tau(M)}\right\} \\ & = & v_i^\alpha(M) + \varepsilon \cdot \mathbb{E}_{i,f(M)}\left\{\alpha^{\tau(M)}\right\}. \end{array}$$

Similarly, we can derive that $v_i^\alpha(M-\varepsilon) \geq v_i^\alpha(M) - \varepsilon \cdot \mathbb{E}_{i,f(M)}\left\{\alpha^{\tau(M)}\right\}$.
Hence, $\frac{v_i^\alpha(M+\varepsilon) - v_i^\alpha(M)}{\varepsilon} \geq \mathbb{E}_{i,f(M)}\left\{\alpha^{\tau(M)}\right\} \geq \frac{v_i^\alpha(M) - v_i^\alpha(M-\varepsilon)}{\varepsilon}$, implying $\frac{\partial}{\partial M} v_i^\alpha(M) = \mathbb{E}_{i,f(M)}\{a^{\tau(M)}\}$. □

Let $v_i^\alpha(M)$ denote the optimal value and let G_i be the indifference value when only a single project is available and its state is i. Now, consider the multiproject case and suppose the state is $i = (i_1, i_2, \dots, i_j, \dots, i_n)$ and let us speculate about whether or not we would ever again operate project j.

If $G_{i_j} > M$ then, because it would not be optimal to stop even if project j were the only project available, it is clear that we would never stop before operating project j. On the other hand, what if $G_{i_j} \leq M$? Would we ever want to operate project j under this circumstance? Whereas it is not obvious that we would never operate project j when $G_{i_j} \leq M$, it does seem somewhat intuitive, so let us accept this as a working hypothesis and see where it leads. That is, let us suppose that once a project reaches a state under which it would be optimal to stop if it were the sole project available, then the optimal policy never again operates that project. From this it follows that the optimal policy will stop in state i when $G_{i_j} \leq M$ for all $j = 1, 2, \dots, n$.

Our speculations lead to the hypothesis that:
(1) project j would never be operated if state i is such that $G_{i_j} \leq M$;
(2) stop should occur if and only if $G_{i_j} \leq M$ for all $j = 1, 2, \dots, n$.

For a given initial state $i = (i_1, i_2, \dots, i_j, \dots, i_n)$, let $\tau_j(M)$ denote the optimal time before we stop when only project j is available, $j = 1, 2, \dots, n$. That is, $\tau_j(M)$ is the time project j has to be operated upon, when its initial state is i_j, until it reaches a state for which M is at least the indifference value. Also, let $\tau(M)$ denote the optimal stopping time for the multiproject case. Because the changes of state of individual projects are in no way affected by what occurs in other projects, it follows that, under our working hypothesis, $\tau(M) = \sum_{j=1}^n \tau_j(M)$. In addition, because $\tau_j(M),\ j = 1, 2, \dots, n$ are independent random variables, we have $\mathbb{E}\{\alpha^{\tau(M)}\} = \mathbb{E}\{\alpha^{\sum_{j=1}^n \tau_j(M)}\} = \prod_{j=1}^n \mathbb{E}\{\alpha^{\tau_j(M)}\}$.

Hence, we obtain by Lemma 8.20

$$\frac{\partial}{\partial M} v_i^\alpha(M) = \mathbb{E}_{i,f(M)}\{\alpha^{\tau(M)}\} = \prod_{j=1}^n \mathbb{E}_{i,f(M)}\{\alpha^{\tau_j(M)}\} = \prod_{j=1}^n \frac{\partial}{\partial M} v_{i_j}^\alpha(M) \tag{8.63}$$

Let $(1-\alpha)C$ be an upper bound of all one-period rewards. Then, C is an upper bound of the total discounted reward (without the terminal reward). Hence, if $M \geq C$, then the stopping action is optimal in all states, i.e. $v_i^\alpha(M) = M$ for $M \geq C$. Integrating (8.63) yields $\int_M^C \frac{\partial}{\partial m} v_i^\alpha(m)dm = \int_M^C \prod_{j=1}^n \frac{\partial}{\partial m} v_{i_j}^\alpha(m)dm$, implying

$$v_i^\alpha(M) = C - \int_M^C \prod_{j=1}^n \frac{\partial}{\partial m} v_{i_j}^\alpha(m)dm. \tag{8.64}$$

We now prove that (8.64) is indeed valid by showing that $C - \int_M^C \prod_{j=1}^n \frac{\partial}{\partial m} v_{i_j}^\alpha(m)dm$ satisfies the optimality equation. Furthermore, the proof gives also the structure of the optimal policy.

Theorem 8.23

For any state $i = (1_1, i_2, \dots, i_n)$ and any terminal reward M, we have

(1) $v_i^\alpha(M) = C - \int_M^C \prod_{j=1}^n \frac{\partial}{\partial m} v_{i_j}^\alpha(m)dm,\ M \leq C.$

(2) The optimal policy takes the stopping action if $M \geq M_{i_j}^\alpha$ for all $j = 1, 2, \dots, n$ and continues with project k if $M_{i_k}^\alpha = max_j\, M_{i_j}^\alpha > M$.

Proof

Let $x_i(M) = C - \int_M^C \prod_{j=1}^n \frac{\partial}{\partial m} v_{i_j}^\alpha(m)dm$. We shall show that $x_i(M)$ satisfies the optimality equation. Let $y_i(k, M) = \prod_{j \neq k} \frac{\partial}{\partial M} v_{i_j}^\alpha(M)$. Because, from Lemma 8.19, $v_{i_j}^\alpha(m)$ is a nondecreasing and convex function of m, it follows that $\frac{\partial}{\partial m} v_{i_j}^\alpha(m)$ is a nonnegative (from nondecreasing) and nondecreasing (from convexity) function of m. Hence, $y_i(k, M)$ is also a nonnegative and nondecreasing function of M. Since $x_i(M)$ can be written as $x_i(M) = C - \int_M^C y_i(k, m) \frac{\partial}{\partial m} v_{i_k}^\alpha(m)dm$, we see, by integration by parts, that

$$x_i(M) = C - y_i(k, m) v_{i_k}^\alpha(m) \,\Big|_{m=M}^{m=C} + \int_M^C v_{i_k}^\alpha(m) dy_i(k, m).$$

Since $v_{i_j}^\alpha(M) = M$ for $M \geq C$, we have $\frac{\partial}{\partial M} v_{i_j}^\alpha(M) = 1$ for $M \geq C$. Therefore, $y_i(k, M) = 1$ for $M \geq C$. Hence, using that $y_i(k, C) = 1$ and $v_{i_k}^\alpha(C) = C$,

$$x_i(M) = y_i(k, M) v_{i_k}^\alpha(M) + \int_M^C v_{i_k}^\alpha(m) dy_i(k, m). \tag{8.65}$$

Similarly, we have

$$r_{i_k}+\alpha\textstyle\sum_{j\in S_k} p_{i_kj}x_{(i_1,i_2,\dots,i_{s-1},j,i_{s+1},\dots,i_n)}(M)=r_{i_k}+\alpha\sum_{j\in S_k} p_{i_kj}\big\{y_i(k,M)v_j^\alpha(M)+\int_M^C v_j^\alpha(m)dy_i(k,m)\big\}.$$

Hence,

$$x_i(M)-\big\{r_{i_k}+\alpha\textstyle\sum_{j\in S_k} p_{i_kj}x_{(i_1,i_2,\dots,i_{s-1},j,i_{s+1},\dots,i_n)}(M)\big\}\;=$$
$$y_i(k,M)v_{i_k}^\alpha(M)+\textstyle\int_M^C v_{i_k}^\alpha(m)dy_i(k,m)-\big\{r_{i_k}+\alpha\sum_{j\in S_k} p_{i_kj}\{y_i(k,M)v_j^\alpha(M)+\int_M^C v_j^\alpha(m)dy_i(k,m)\}\big\}.$$

We can also write $r_{i_k}=r_{i_k}\{y_i(k,M)+y_i(k,C)-y_i(k,M)\}=r_{i_k}y_i(k,M)+r_{i_k}\int_M^C dy_i(k,m)$.

Therefore,

$$x_i(M)-\big\{r_{i_k}+\alpha\textstyle\sum_{j\in S_k} p_{i_kj}x_{(i_1,i_2,\dots,i_{s-1},j,i_{s+1},\dots,i_n)}(M)\big\}\;=$$
$$y_i(k,M)\big\{v_{i_k}^\alpha(M)-r_{i_k}-\alpha\sum_{j\in S_k} p_{i_kj}v_j^\alpha(M)\big\}+\int_M^C\big\{v_{i_k}^\alpha(m)-r_{i_k}-\alpha\sum_{j\in S_k} p_{i_kj}v_j^\alpha(m)\big\}dy_i(k,m).\quad(8.66)$$

Since

$$v_{i_k}^\alpha(M)\ge r_{i_k}+\alpha\sum_{j\in S_k} p_{i_kj}v_j^\alpha(M)\text{ for all actions }k,\quad(8.67)$$

we obtain

$$x_i(M)\ge r_{i_k}+\alpha\sum_{j\in S_k} p_{i_kj}x_{(i_1,i_2,\dots,i_{s-1},j,i_{s+1},\dots,i_n)}(M)\text{ for all actions }k=1,2,\dots,n.\quad(8.68)$$

Let us see under what conditions equality occurs in (8.68). First, note that (8.67) holds with equality if continuation is optimal when only project k is available, i.e. when $M\le G_{i_k}$. In that case we have from (8.66)

$$x_i(M)-\big\{r_{i_k}+\alpha\textstyle\sum_{j\in S_k} p_{i_kj}x_{(i_1,i_2,\dots,i_{s-1},j,i_{s+1},\dots,i_n)}(M)\big\}\;=$$
$$\int_{G_{i_k}}^C\big\{v_{i_k}^\alpha(m)-r_{i_k}-\alpha\sum_{j\in S_k} p_{i_kj}v_j^\alpha(m)\big\}dy_i(k,m)\text{ if }M\le G_{i_k}.\quad(8.69)$$

Since $v_{i_j}^\alpha(m)=m$ for $m\ge G_{i_j}$, we have $y_i(k,m)=\prod_{j\neq k}\frac{\partial}{\partial m}v_{i_j}^\alpha(m)=1$ for $m\ge max_{j\neq k}G_{i_j}$. So, $dy_i(k,m)=0$ for $m\ge max_{j\neq k}G_{i_j}$. Hence, using this we see from (8.69) that for $M\le G_{i_k}=max_j\,G_{i_j}$, we obtain

$$x_i(M)=r_{i_k}+\alpha\sum_{j\in S_k} p_{i_kj}x_{(i_1,i_2,\dots,i_{s-1},j,i_{s+1},\dots,i_n)}(M).\quad(8.70)$$

For $M\ge max_j\,G_{i_j}$, we have $v_{i_j}^\alpha(m)=m$ for $j=1,2,\dots,n$, and thus $y_i(k,m)=1$, implying that $dy_i(k,m)=0$ for $m\ge M$. Hence, from (8.65) we see that

$$x_i(M)=v_{i_k}^\alpha(M)=M\text{ if }M\ge max_j\,G_{i_j}.\quad(8.71)$$

In addition, also using (8.65) and the monotonicity of $v_k(m)$ in m, we have for all M

$$x_i(M)\ge y_i(k,M)v_{i_k}^\alpha(M)+v_{i_k}^\alpha(M)\{y_i(k,C)-y_i(k,M)\}=v_{i_k}^\alpha(M)\ge M.\quad(8.72)$$

Hence, we have from (8.68) and (8.72)

$$\begin{cases} x_i(M) & \ge & r_{i_k}+\alpha\sum_{j\in S_k} p_{i_kj}x_{(i_1,i_2,\dots,i_{s-1},j,i_{s+1},\dots,i_n)}(M),\ i\in S,\ k=1,2,\dots,n\\ x_i(M) & \ge & M,\ i\in S\end{cases}$$

Furthermore, we have from (8.70) and (8.71)

$$\begin{cases} x_i(M) & = & r_{i_k}+\alpha\sum_{j\in S_k} p_{i_kj}x_{(i_1,i_2,\dots,i_{s-1},j,i_{s+1},\dots,i_n)}(M),\ i\in S & \text{if }M\le max_j\,G_{i_j}=G_{i_k}\\ x_i(M) & = & M,\ i\in S & \text{if }M\ge max_j\,G_{i_j}=G_{i_k}\end{cases}$$

Hence, $x(M)$ satisfies the optimality equation

$$x_i(M) = max\{M, max_{1\leq k\leq n}\{r_{i_k} + \alpha \sum_{j\in S_k} p_{i_k j} x_{(i_1,i_2,\dots,i_{s-1},j,i_{s+1},\dots,i_n)}(M)\}\},\ i \in S, \tag{8.73}$$

and the optimal policy is as stated. □

Remark

The preceding theorem shows that the optimal policy in the multi-project case can be determined by an analysis of the n single-project problems, with the optimal decision in state $i = (i_1, i_2, \dots, i_n)$ being to operate on that project k having the largest indifference value G_{i_k} if this value is greater than M and to stop otherwise.

Alternative interpretation of the Gittins index

Consider the one-armed bandit problem having initial state i. Because when $M = G_i$ the optimal policy is indifferent between stopping and continuing, so that for any stopping random stopping time τ, $G_i \geq \mathbb{E}\{\text{discounted reward before } \tau\} + G_i \cdot \mathbb{E}\{\alpha^\tau\}$, with equality for the optimal policy. Hence,

$$\begin{aligned}(1-\alpha)G_i &= max_{\tau\geq 1} \frac{\mathbb{E}\{\text{discounted reward before } \tau\}}{\{1-\mathbb{E}\{\alpha^\tau\}\}/(1-\alpha)} \\ &= max_{\tau\geq 1} \frac{\mathbb{E}\{\text{discounted reward before } \tau\}}{\mathbb{E}\{1+\alpha+\cdots+\alpha^{\tau-1}\}} \\ &= max_{\tau\geq 1} \frac{\mathbb{E}\{\text{discounted reward before } \tau\}}{\mathbb{E}\{\text{discounted time before } \tau\}},\end{aligned}$$

where the expectations are conditional on the initial state i. Hence, another way of describing the optimal policy in the multi-armed bandit problem is as follows. For each individual project look for the stopping time τ whose ratio of expected discounted reward and expected discounted time prior to τ is maximal. Then work on the project with the largest ratio. In the case there also is the additional option of stopping, one should stop if all ratios are smaller than $(1-\alpha)M$.

8.6.4 Methods for the computation of the Gittins indices

1. The parametric linear programming method

We have already seen that for a single project with terminal reward M the solution can be obtained from a linear programming problem, namely program (8.60). For M big enough, e.g. for $M \geq C$, where $C := (1-\alpha)^{-1} \cdot max_i\, r_i$, we know that $v_i^\alpha(M) = M$ for all states i. Furthermore, we have seen that the Gittins index $G_i = min\{M|\ v_i^\alpha(M) = M\}$ (cf. (8.61)).

One can solve program (8.60) as a parametric linear programming problem with parameter M. Starting with $M = C$ one can decrease M and find for each state i the largest M for which it is optimal to keep working on the project, which is in fact $min\{M|\ v_i^\alpha(M) = M\} = G_i$, in the order of decreasing M-values. One can start with the simplex tableau in which all y-variables are in the basis and in which the x-variables are the nonbasic variables. This tableau is optimal for $M \geq C$. Decrease M until we meet a basis change, say the basic variable y_i will be exchanged with the nonbasic variable x_i. Then, we know the M-value which is equal to G_i. In this way we continue and repeat the procedure N times, where N is the number of states in the current project. The used pivoting row and column do not influence any further pivoting step, so we can delete these row and column from the simplex tableau.

Example 8.3

Consider a project with the following data.
$S = \{1, 2, 3\}$; $\alpha = \frac{1}{2}$; $r_1 = 8,\ r_2 = 6,\ r_3 = 4$.
$p_{11} = 0,\ p_{12} = 1,\ p_{13} = 0;\ p_{21} = 0,\ p_{22} = 0,\ p_{23} = 1;\ p_{31} = 1,\ p_{32} = 0,\ p_{33} = 0.$

The linear program becomes (the objective function is splitted up into two rows, one for the x-part and one for parametric y-part; the y-variables have to be expressed in the nonbasic x-variables, i.e. we obtain for the last row $y_1 + y_2 + y_3 = 3 - \frac{1}{2}x_1 - \frac{1}{2}x_2 - \frac{1}{2}x_3$).

$$max\{8x_1 + 6x_2 + 4x_3 + My_1 + My_2 + My_3\}$$

subject to

$$\begin{array}{rcrcrcrcrcrcl} x_1 & & & - & \frac{1}{2}x_3 & + & y_1 & & & & & = & 1;\ x_1,\, y_1 \geq 0; \\ -\frac{1}{2}x_1 & + & x_2 & & & & & + & y_2 & & & = & 1;\ x_2,\, y_2 \geq 0; \\ & - & \frac{1}{2}x_2 & + & x_3 & & & & & + & y_3 & = & 1;\ x_3,\, y_3 \geq 0. \end{array}$$

The first tableau becomes:

		x_1	x_2	x_3
y_1	1	^{*1}	0	$-\frac{1}{2}$
y_2	1	$-\frac{1}{2}$	1	0
y_3	1	0	$-\frac{1}{2}$	1
x_0	0	-8	-6	-4
M	3	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$

The objective function is $3M + (8 - \frac{1}{2}M)x_1 + (6 - \frac{1}{2}M)x_2 + (4 - \frac{1}{2}M)x_3$.
Hence, this tableau is optimal for $x_1 = x_2 = x_3 = 0$ if $8 - \frac{1}{2}M \leq 0$, $6 - \frac{1}{2}M \leq 0$ and $4 - \frac{1}{2}M \leq 0$, i.e. if $M \geq 16$.
For $M = 16$ there is indifference in state 1: $G_1 = 16$.
Then, we exchange x_1 and y_1 and obtain a new simplex tableau in which the row of y_1 and the column of x_1 can be deleted.
The second tableau is:

		x_2	x_3
y_2	$\frac{3}{2}$	^{*1}	$-\frac{1}{4}$
y_3	1	$-\frac{1}{2}$	1
x_0	8	-6	-8
M	$\frac{5}{2}$	$\frac{1}{2}$	$\frac{3}{4}$

This tableau is optimal for $x_2 = x_3 = 0$ if $6 - \frac{1}{2}M \leq 0$ and $8 - \frac{3}{4}M \leq 0$, i.e. if $M \geq 12$.
For $M = 12$ there is indifference in state 2: $G_2 = 12$.
Then, we exchange x_2 and y_2 and obtain a new simplex tableau in which the row of y_2 and the column of x_2 can be deleted.
The final tableau is:

		x_3
y_3	$\frac{7}{4}$	$\frac{7}{8}$
x_0	17	$-\frac{19}{2}$
M	$\frac{7}{4}$	$\frac{7}{8}$

This tableau is optimal for $x_3 = 0$ if $\frac{19}{2} - \frac{7}{8}M \leq 0$, i.e. if $M \geq \frac{76}{7}$. Hence, $G_3 = \frac{76}{7}$.

Computational complexity

We can easily determine the computational complexity. Each update of an element in a simplex tableau needs at most two arithmetic operations (multiplication and divisions as well as additions and subtractions): for instance, the value 17 in the last tableau of Example 8.3 is computed by $8 - (-6) \cdot \frac{3}{2} = 17$. Hence, the total number of arithmetic operations in this method is at most $2 \cdot \sum_{k=1}^{N} k^2 = \frac{1}{3}N(N+1)(2N+1) = \frac{2}{3}N^3 + \mathcal{O}(N^2) = \mathcal{O}(N^3)$.

2. The restart-in-k method

We will derive another interpretation for the Gittins index G_k in a fixed state k. The optimality equation for a single project with terminal reward M is, cf. (8.58),

$$v_i^\alpha(M) = max\{M,\, r_i + \alpha \sum_j p_{ij} v_j^\alpha(M)\},\ i \in S. \tag{8.74}$$

We have seen that G_k is the indifference value, i.e. for $M = G_k$ we have

$$v_k^\alpha(G_k) = G_k = r_k + \alpha \sum_j p_{kj} v_j^\alpha(G_k). \tag{8.75}$$

Using (8.74) and (8.75) yields

$$v_i^\alpha(G_k) = max\big\{r_k + \alpha \sum_j p_{kj} v_j^\alpha(G_k) = G_k,\, r_i + \alpha \sum_j p_{ij} v_j^\alpha(M_k)\big\},\ i \in S. \tag{8.76}$$

With the abbreviation $v_i^k := v_i^\alpha(G_k),\ i \in S$, we get the following expression

$$v_i^k = max\big\{r_i + \alpha \sum_j p_{ij} v_j^k,\, r_k + \alpha \sum_j p_{kj} v_j^k\big\},\ i \in S. \tag{8.77}$$

Hence, G_k is the k-th component of the value vector of the MDP where there are in each state two actions. This problem can be interpreted as follows: in each state there are two options, either to continue working on the project in the given state i, or to restart working in state k, where the total expected discounted reward must be maximized. This gives the problem the name *restart-in-k* problem. By solving this MDP we find $G_k = v_k^k$. Notice that we now have a characterization of the Gittins index without using a terminal reward.

We define C_k for the restart-in-k problem as the set of states i for which it is optimal to continue in that state. If the MDP is solved we find G_k and $v_i^\alpha(G_k),\ i \in S$. The next theorem shows that C_k contains exactly those states j for which $G_j \geq G_k$. When we are in state $(i_1, i_2, \dots, i_n)$ and decide to work on project k because $G_{i_k} \geq G_{i_l}$ for all $1 \leq l \leq n$, and when we also move in project k from state i_k to a state $j \in C_k$, then we know that the largest Gittins index is still the index of the state j of project k, without knowing the value of G_j. So, the theorem tell us that we only have to calculate a new index when we enter a state which is not in C_k.

Theorem 8.24
$C_k = \{j \mid G_j \geq G_k\}$ with $C_k := \{j \mid$ for the restart-in-k problem it is optimal to continue in state $j\}$.

Proof
$j \notin C_k$ if and only if it is not optimal to continue in state j for the restart-in k problem or, equivalently, for the optimal stopping problem with terminal reward $M = G_k$. Since G_j is the indifference value in state j, it is not optimal to continue in state j if and only if $M > G_j$. Therefore, $G_k > G_j$. So, $j \notin C_k \Leftrightarrow G_k > G_j$, i.e. $C_k = \{j \mid G_j \geq G_k\}$. □

We can solve the restart-in-k problem by any method for discounted MDPs. If we use the linear programming method the program becomes

$$min \left\{ \sum_j v_j \;\middle|\; \begin{array}{l} \sum_j \{\delta_{ij} - \alpha p_{ij}\} v_j \geq r_i,\ i \in S \\ \sum_j \{\delta_{ij} - \alpha p_{kj}\} v_j \geq r_k,\ i \in S,\ i \neq k \end{array} \right\}. \tag{8.78}$$

Example 8.3 (continued)
The linear program for G_1 is:

$$min \left\{ v_1 + v_2 + v_3 \;\middle|\; \begin{array}{lll} v_1 \geq 8 + \frac{1}{2}v_2; & v_2 \geq 6 + \frac{1}{2}v_3; & v_3 \geq 4 + \frac{1}{2}v_1 \\ v_2 \geq 8 + \frac{1}{2}v_2; & v_3 \geq 8 + \frac{1}{2}v_2 & \end{array} \right\}.$$

The optimal solution is: $v_1 = v_2 = v_3 = 16 \rightarrow G_1 = v_1 = 16$ and $C_1 = \{1\}$.

The linear program for G_2 is:

$$min \left\{ v_1 + v_2 + v_3 \;\middle|\; \begin{array}{l} v_1 \geq 8 + \frac{1}{2}v_2; \quad v_2 \geq 6 + \frac{1}{2}v_3; \quad v_3 \geq 4 + \frac{1}{2}v_1 \\ v_1 \geq 6 + \frac{1}{2}v_3; \quad v_3 \geq 6 + \frac{1}{2}v_3 \end{array} \right\}.$$

The optimal solution is: $v_1 = 14;\ v_2 = v_3 = 12 \ \rightarrow\ G_2 = v_2 = 12$ and $C_2 = \{1, 2\}$.
The linear program for G_3 is:

$$min \left\{ v_1 + v_2 + v_3 \;\middle|\; \begin{array}{l} v_1 \geq 8 + \frac{1}{2}v_2; \quad v_2 \geq 6 + \frac{1}{2}v_3; \quad v_3 \geq 4 + \frac{1}{2}v_1 \\ v_1 \geq 4 + \frac{1}{2}v_1; \quad v_2 \geq 4 + \frac{1}{2}v_1 \end{array} \right\}.$$

The optimal solution is: $v_1 = \frac{96}{7};\ v_2 = \frac{80}{7};\ v_3 = \frac{76}{7};\ \rightarrow\ G_3 = v_3 = \frac{76}{7}$ and $C_3 = \{1, 2, 3\}$.

Computation on-line
It is interesting to ask what indices must be computed and when this must be done. In the first period, it is necessary to compute the n initial indices, one for each project. Subsequently, it suffices to compute at most *one* index in each period. In particular, one computes the index of a project k in a period only when its state i_k leaves the optimal continuation set C_{i_k}. Thus, by Theorem 8.24, if the indices are computed on-line only as needed, the indices computed for each project will decrease strictly over time.

An alternative linear program
Since the v-variables are unrestricted in sign, one may substitute v_j by $y_j + z$, where z is unrestricted and $y_j \geq 0$ for all j. Then, program (8.24) can be written as

$$min \left\{ \sum_j y_j + N \cdot z \;\middle|\; \begin{array}{l} (1-\alpha)z + \sum_j \{\delta_{ij} - \alpha p_{ij}\}y_j \geq r_i,\ i \in S,\ i \neq k \\ (1-\alpha)z + \sum_j \{\delta_{ij} - \alpha p_{kj}\}y_j \geq r_k,\ i \in S \\ z \text{ unrestricted},\ y_j \geq 0, j \in S \end{array} \right\}. \tag{8.79}$$

Consider the second part of the constraints: $(1-\alpha)z + y_i \geq r_k + \alpha \sum_j p_{kj}v_j \geq r_k,\ i \in S$, which is equivalent with $(1-\alpha)z + min_i\, y_i \geq r_k + \alpha \sum_j p_{kj}v_j \geq r_k$. If the y_j becomes ε smaller for each j and z becomes ε bigger, then the objective function keeps its value and the constraints remain satisfied. So we can take $min_i\, y_i = 0$ and the linear program becomes

$$min \left\{ \sum_j y_j + N \cdot z \;\middle|\; \begin{array}{lcl} (1-\alpha)z + \sum_j \{\delta_{ij} - \alpha p_{ij}\}y_j & \geq & r_i,\ i \in S,\ i \neq k \\ (1-\alpha)z - \alpha \sum_j p_{kj}y_j & \geq & r_k,\ i \in S \\ z \text{ unrestricted},\ y_j \geq 0, j \in S & & \end{array} \right\}. \tag{8.80}$$

For the optimal solution v^* of (8.24) we have $v_i^* \geq r_k + \alpha \sum_j p_{kj}v_j^* = v_k^*,\ i \in S$. Hence, $y_k^* = min_i\, y_i^* = 0$, and consequently, $G_k = v_k^* = min_i\, y_i^* + z^* = z^*$, where (y^*, z^*) is the optimal solution of program (8.79).

Example 8.3 (continued)
The linear program for G_1 is:

$$min\{y_1 + y_2 + y_3 + 3z \mid \tfrac{1}{2}z + y_2 \geq 6 + \tfrac{1}{2}y_3;\ \tfrac{1}{2}z + y_3 \geq 4 + \tfrac{1}{2}y_1;\ \tfrac{1}{2}z \geq 8 + \tfrac{1}{2}y_2;\ y_1, y_2, y_3 \geq 0\}.$$

The optimal solution is: $y_1 = y_2 = y_3 = 0;\ z = 16 \ \rightarrow\ G_1 = z = 16$.
The linear program for G_2 is:

$$min\{y_1 + y_2 + y_3 + 3z \mid \tfrac{1}{2}z + y_1 \geq 8 + \tfrac{1}{2}y_2;\ \tfrac{1}{2}z + y_3 \geq 4 + \tfrac{1}{2}y_1;\ \tfrac{1}{2}z \geq 6 + \tfrac{1}{2}y_3;\ y_1, y_2, y_3 \geq 0\}.$$

The optimal solution is: $y_1 = 2,\ y_2 = y_3 = 0;\ z = 12 \ \rightarrow\ G_2 = z = 12$.
The linear program for G_3 is:

$$min\{y_1 + y_2 + y_3 + 3z \mid \tfrac{1}{2}z + y_1 \geq 8 + \tfrac{1}{2}y_2;\ \tfrac{1}{2}z + y_2 \geq 6 + \tfrac{1}{2}y_3;\ \tfrac{1}{2}z \geq 4 + \tfrac{1}{2}y_1;\ y_1, y_2, y_3 \geq 0\}.$$

The optimal solution is: $y_1 = \frac{20}{7},\ y_2 = \frac{4}{7},\ y_3 = 0;\ z = \frac{76}{7} \ \rightarrow\ G_3 = z = \frac{76}{7}$.

3. The largest-remaining-index method

Theorem 8.25

Suppose that $G_1 \geq G_2 \geq \cdots \geq G_k$ for some k, and $G_k \geq G_i$ for $i = k+1, k+2, \dots, n$.
Let l_k be such that M_{l_k} be such that $G_{l_k} = max_{i>k} G_i$. Then, we have

$$(1-\alpha)G_{l_k} = max_{i>k} \frac{\{\big(I-\alpha P(k)\big)^{-1}r\}_i}{\{\big(I-\alpha P(k)\big)^{-1}e\}_i}, \ \textit{where } \{P(k)\}_{ij} := \begin{cases} p_{ij} & , \ j \leq k; \\ 0 & , \ j > k. \end{cases}$$

Proof

Since $v_i^\alpha(G_{l_k}) \geq r_i + \alpha \sum_j p_{ij} v_j^\alpha(G_{l_k})$ and $v_i^\alpha(M) = M$ for $M \geq G_i, \ i \in S$, we can write

$$\begin{array}{rcl} v_i^\alpha(G_{l_k}) & \geq & r_i + \alpha \sum_{j \leq k} p_{ij} v_j^\alpha(G_{l_k}) + \alpha \sum_{j>k} p_{ij} v_j^\alpha(G_{l_k}) \\ & = & r_i + \alpha \sum_{j \leq k} p_{ij} v_j^\alpha(G_{l_k}) + \alpha G_{l_k} \{1 - \sum_{j \leq k} p_{ij}\}. \end{array}$$

In vector notation, with $v = v^\alpha(G_{l_k})$, this becomes

$$v \geq r + \alpha P(k)v + \alpha G_{l_k} e - \alpha G_{l_k} P(k)e = r + \alpha P(k)v - (1-\alpha)G_{l_k} e + G_{l_k}\{I - \alpha P(k)\}e.$$

So, $\{I-\alpha P(k)\}v \geq r - (1-\alpha)G_{l_k} e + G_{l_k}\{I-\alpha P(k)\}e$. Since $\{I-\alpha P(k)\}$ is nonsingular with nonnegative inverse, we can write

$$v \geq \{I - \alpha P(k)\}^{-1} r - (1-\alpha)G_{l_k}\{I - \alpha P(k)\}^{-1}e + \ G_{l_k} e.$$

Componentwise, for all $i \geq k$,

$$G_{l_k} = v_i^\alpha(G_{l_k}) \geq \{\big(I-\alpha P(k)\big)^{-1}r\}_i - (1-\alpha)G_{l_k}\{\big(I-\alpha P(k)\big)^{-1}e\}_i + G_{l_k},$$

with equality for $i = k$. From this it follows that $(1-\alpha)G_{l_k} \geq \frac{\{(I-\alpha P(k))^{-1}r\}_i}{\{(I-\alpha P(k))^{-1}e\}_i}$ for all $i \geq k$ with equality for $i = k$. Therefore,

$$(1-\alpha)G_{l_k} = max_{i>k} \frac{\{(I-\alpha P(k))^{-1}r\}_i}{\{(I-\alpha P(k))^{-1}e\}_i}. \qquad \square$$

To compute G_{l_k}, we have to invert the matrix $\{I - \alpha P(k)\}$, which can be written as $\begin{pmatrix} A_k & 0 \\ B_k & I \end{pmatrix}$. It can now easily be checked that $\{I-\alpha P(k)\}^{-1} = \begin{pmatrix} A_k^{-1} & 0 \\ -B_k A_k^{-1} & I \end{pmatrix}$. The inversion of a matrix of order k can be done in $\mathcal{O}(k^3)$ steps. Hence the computation of the Gittins indices of a project with N states has complexity $\mathcal{O}(N^4)$. Fortunately, since subsequent matrices $P(k)$ are similar, this can be done efficiently in a recursive way. In this way time can be saved and the computation can be done in $\mathcal{O}(N^3)$ steps, as we will see.

Write $A_{k+1} = \begin{pmatrix} A_k & p \\ q & x \end{pmatrix}$, so the inverse $A_{k+1}^{-1} = \begin{pmatrix} N & t \\ s & y \end{pmatrix}$, where $p := (p_{1,k+1}, p_{2,k+1}, \dots, p_{k,k+1})^T$, $q := (p_{k+1,1}, p_{k+1}, 2, \dots, p_{k+1,k})^T$ and $x := p_{k+1,k+1}$. Since $A_{k+1}A_{k+1}^{-1} = I$, we get

$$A_k N + ps = I; \tag{8.81}$$
$$qN + xs = 0; \tag{8.82}$$
$$A_k t + py = 0; \tag{8.83}$$
$$qt + xy = 1. \tag{8.84}$$

From (8.83) and (8.84), we obtain

$$t = -yA_k^{-1}p, \ -yqA_k^{-1}p + xy = 1 \ \rightarrow \ y = \frac{1}{x - qA_k^{-1}p}, \tag{8.85}$$

and from (8.81)

$$N = A_k^{-1}(I - ps) = A_k^{-1} - A_k^{-1}ps. \tag{8.86}$$

Insertion into (8.82) gives $0 = qA_k^{-1} - \{qA_k^{-1}p + x\}s = qA_k^{-1} + \frac{1}{y}s \ \rightarrow \ s = -yqA_k^{-1}$. With (8.85) and (8.86), we obtain $N = A_k^{-1} + \frac{1}{y}ts$. Therefore, we have shown that

$$A_{k+1}^{-1} = \begin{pmatrix} A_k^{-1} + \frac{1}{y}ts & t \\ s & y \end{pmatrix}, \text{ where } y := \frac{1}{x - qA_k^{-1}p},\ \ t := -yA_k^{-1}p \text{ and } s := -yqA_k^{-1}.$$

All these calculations can be done in $\mathcal{O}(k^2)$ steps, because at most a vector of k components and a $(k \times k)$-matrix have to be multiplied. The calculation of the matrix $B_{k+1}A_{k+1}^{-1}$ costs using the standard method $\mathcal{O}(k^3)$ steps, but on this number can also be saved if $B_kA_k^{-1}$ is known.

Write $B_k = \begin{pmatrix} f^T \\ F_k \end{pmatrix}$ and $B_{k+1} = \begin{pmatrix} F_k & g \end{pmatrix}$, where f^T is the top row of B_k. Then, we obtain

$$B_kA_k^{-1} = \begin{pmatrix} f^TA_k^{-1} \\ F_kA_k^{-1} \end{pmatrix} \text{ and } B_{k+1}A_{k+1}^{-1} = \begin{pmatrix} F_k & g \end{pmatrix} \begin{pmatrix} A_k^{-1} + \frac{1}{y}ts & t \\ s & y \end{pmatrix} = \begin{pmatrix} F_kA_k^{-1} + \frac{1}{y}F_kts + gs & F_kt + gy \end{pmatrix}.$$

Because $B_kA_k^{-1}$ is known, the matrix $B_{k+1}A_{k+1}^{-1}$ can also be calculated in $\mathcal{O}(k^2)$ steps, so the complexity of this method for the computation of the Gittins indices of one project with N states is $\sum_{k=1}^N \mathcal{O}(k^2)$ which is $\mathcal{O}(N^3)$.

Example 8.3 (continued)

For the largest Gittins index we have: $(1-\alpha)G_{l_0} = max_i\ r_i = 8$ for $i = 1 \ \rightarrow \ G_{l_0} = G_1 = 16$.

Since the transition matrix $P = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{pmatrix}$ and the first index is G_1, we have for $P(1)$ the matrix $\begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \end{pmatrix}$. Hence, $I - \alpha P(1) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ \frac{1}{2} & 0 & 1 \end{pmatrix}$ and $\{I - \alpha P(1)\}^{-1} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ \frac{1}{2} & 0 & 1 \end{pmatrix}$.

Therefore, $\{I - \alpha P(1)\}^{-1}r = (8, 6, 8)$ and $\{I - \alpha P(1)\}^{-1}e = (1, 1, \frac{3}{2})$.

Hence, $(1-\alpha)G_{l_1} = max\left\{\frac{6}{1}, \frac{8}{3/2}\right\} = 6$ for $i = 2 \ \rightarrow \ G_{l_1} = G_2 = 12$.

Since $G_1 \geq G_2$ are the two largest Gittins indices, we have $P(2) = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \end{pmatrix}$.

Hence, $I - \alpha P(2) = \begin{pmatrix} 1 & -\frac{1}{2} & 0 \\ 0 & 1 & 0 \\ -\frac{1}{2} & 0 & 1 \end{pmatrix}$ and $\{I - \alpha P(2)\}^{-1} = \begin{pmatrix} 1 & \frac{1}{2} & 0 \\ 0 & 1 & 0 \\ \frac{1}{2} & \frac{1}{4} & 1 \end{pmatrix}$.

Therefore, $\{I - \alpha P(2)\}^{-1}r = (11, 6, \frac{19}{2})$ and $\{I - \alpha P(2)\}^{-1}e = (\frac{3}{2}, 1, \frac{7}{4})$.

Hence, $(1-\alpha)G_{l_2} = \frac{19/2}{7/4} = \frac{38}{7} \ \rightarrow \ G_{l_2} = G_3 = \frac{76}{7}$.

4. The bisection/successive approximation method

In this section an iterative method, combining bisection and successive approximations, is proposed for computing intervals containing the Gittins indices. The final intervals could be of a specific maximum length, or merely disjoint. In the first case we have approximations of the Gittins indices; in the second case we have a ranking of the Gittins indices, which in many applications is sufficient. The initial intervals are $[L_i, U_i],\ i \in S$, satisfying $L_i \leq G_i \leq U_i,\ i \in S$. We start with $M \in [L_k, U_k]$ for some state k. We will show that it is possible to obtain in a bounded number of iterations a smaller interval, also containing G_k, with M as one of its end points, i.e. the next interval is either $[M, U_k]$ or $[L_k, M]$. This new interval is again denoted as $[L_k, U_k]$ and the next M is computed by bisection: $M := \frac{1}{2}[L_k, U_k]$.

Since the policy that never takes the stopping action has at most $(1-\alpha)^{-1} \cdot max_j\, r_j$ as expected discounted return, for any $M \geq (1-\alpha)^{-1} \cdot max_j\, r_j$ it is in any state optimal to stop with terminal reward M, i.e. $v_i^\alpha(M) = M,\ i \in S$, for any $M \geq (1-\alpha)^{-1} \cdot max_j\, r_j$. Hence, $(1-\alpha)^{-1} \cdot max_j\, r_j$ is an upper bound for all Gittins indices $G_i,\ i \in S$. The next lemma provides stronger initial bounds of the Gittins indices; these bounds also depend on the state.

Lemma 8.21
$r_i + \frac{\alpha}{1-\alpha} \cdot min_k\, r_k \leq G_i \leq r_i + \frac{\alpha}{1-\alpha} \cdot max_k\, r_k,\ i \in S.$

Proof
Because for any M, $v_j^\alpha(M)$ is the value of an optimal stopping problem, for any $j \in S$, $v_j^\alpha(M)$ is at least the value of the policy that continues always, which in turn is at least $(1-\alpha)^{-1} \cdot min_k\, r_k$. Since G_i is the indifference value of the optimal stopping problem with terminal reward $M = G_i$, we also have $G_i = r_i + \alpha \sum_j p_{ij} v_j^\alpha(G_i)$. Therefore,

$$G_i = r_i + \alpha \textstyle\sum_j p_{ij} v_j^\alpha(G_i) \geq r_i + \alpha \sum_j p_{ij}\{(1-\alpha)^{-1} \cdot min_k\, r_k\} = r_i + \frac{\alpha}{1-\alpha} \cdot min_k\, r_k.$$

Because $v_j^\alpha(M)$ is nondecreasing in M and $v_j^\alpha(R) = R$, where $R := (1-\alpha)^{-1} \cdot max_k\, r_k$, and also $R \geq G_i$, we obtain

$$G_i = r_i + \alpha \textstyle\sum_j p_{ij} v_j^\alpha(G_i) \leq r_i + \alpha \sum_j p_{ij} v_j^\alpha(R) = r_i + \frac{\alpha}{1-\alpha} \cdot max_k\, r_k. \qquad \square$$

Since $v_j^\alpha(M) \geq M,\ j \in S$, we also have $v_i^\alpha(G_i) = max\{G_i, r_i + \alpha \sum_j p_{ij} v_j^\alpha(G_i)\} \geq max\{G_i, r_i + \alpha G_i\}$ for all $i \in S$. Thus, $G_i = v_i^\alpha(G_i) \geq r_i + \alpha G_i$, i.e. $G_i \geq (1-\alpha)^{-1} \cdot r_i$. Hence, we have the following initial lower and upper bound of G_i, denoted by L_i and U_i, respectively:

$$L_i := max\{r_i + \frac{\alpha}{1-\alpha} \cdot min_k\, r_k, (1-\alpha)^{-1} \cdot r_i\};\ U_i := r_i + \frac{\alpha}{1-\alpha} \cdot max_k\, r_k. \tag{8.87}$$

If these bounds are equal, obviously $G_i = L_i = U_i$. Notice that this is that case for i_*, where i_* is such that $r_{i_*} = max_k\, r_k$; namely, in that case we have $U_{i_*} = r_{i_*} + \frac{\alpha}{(1-\alpha)} r_{i_*} = (1-\alpha)^{-1} r_{i_*} \leq L_{i_*}$. Hence

$$L_{i_*} = G_{i_*} = U_{i_*} = (1-\alpha)^{-1} \cdot max_k\, r_k, \text{ where } i_* \text{ satisfies } r_{i_*} = max_k\, r_k. \tag{8.88}$$

Given a fixed $M \in [L_k, U_k]$, we first approximate $v^\alpha(M)$ by successive approximations. The maximal expected discounted return $v^\alpha(M)$ satisfies the optimality equation

$$v_i^\alpha(M) = max\{M, r_i + \alpha \sum_j p_{ij} v_j^\alpha(M)\},\ i \in S \tag{8.89}$$

and can be computed by the following value iteration scheme

$$\begin{cases} v_i^0(M) & := & M,\ i \in S \\ v_i^n(M) & := & max\{M, r_i + \alpha \sum_j p_{ij} v_j^{n-1}(M)\},\ i \in S,\ n = 1, 2, \dots \end{cases} \tag{8.90}$$

We know that $v_i^\alpha(M) = \lim_{n\to\infty} v_i^n(M),\ i \in S$. By induction on n it is easy to verify that $v_i^n(M)$ is nondecreasing in n for all $i \in S$. Since $G_k = min\{M \mid M = v_k^\alpha(M)\}$ and by the property that $v_k^\alpha(M) - M$ is a nonincreasing function of M (see Lemma 8.18) it follows that $M < G_k$ is equivalent to $v_k^\alpha(M) > M$, which in turn is equivalent to $v_k^n(M) > M$ for some n, i.e. $r_k + \alpha \sum_j p_{kj} v_j^{n-1}(M) > M$ for some n. The next lemma gives a lower bound of such n.

Lemma 8.22

Given $k \in S$ and M satisfying $G_k > M$, then $v_k^n(M) > M$ for all $n > n_1$, where the number n_1 is defined by $n_1 := \frac{log\left\{\frac{v_k^\alpha(M)-M}{(1-\alpha)^{-1}\cdot max_j\, r_j - M}\right\}}{log\,\alpha}$.

Proof

From the theory of contracting mappings it follows that

$$\|v^\alpha(M) - v^n(M)\|_\infty \le \alpha^n \cdot \|v^\alpha(M) - v^0(M)\|_\infty = \alpha^n \cdot max_j\, \{v_j^\alpha(M) - M\}.$$

Note that $n > n_1$ is equivalent to $\alpha^n \cdot \{(1-\alpha)^{-1} \cdot max_j\, r_j - M\} < v_k^\alpha(M) - M$. Since $M < G_k$, we have $v_k^\alpha(M) - M > 0$. Furthermore, with $R := (1-\alpha)^{-1} \cdot max_j\, r_j$, we also have $v_j^n(M) < R,\ j \in S$ for all $n \in \mathbb{N}$ (by induction on n this follows directly from the value iteration scheme). Hence, we obtain $v_j^\alpha(M) = \lim_{n\to\infty} v_j^n(M) \le R,\ j \in S$. Now, we can write for $n > n_1$,

$$\begin{aligned} v_k^\alpha(M) - v_k^n(M) &\le \|v^\alpha(M) - v^n(M)\|_\infty \le \alpha^n \cdot max_j\, \{v_j^\alpha(M) - M\} \\ &\le \alpha^n \cdot (R - M) < v_k^\alpha(M) - M, \end{aligned}$$

i.e. $v_k^n(M) > M$. □

The operational meaning of Lemma 8.22 is that for any $k \in S$ and any $M \in [L_k, U_k]$ at most n_1 iterations are needed to decide whether $G_k \le M$:

$$\text{if } v_k^n(M) \le M \text{ for some } n > n_1, \text{ then } G_k \le M. \tag{8.91}$$

Unfortunately, since we do not know the value $v_k^\alpha(M)$, we do not know the number n_1. However, we can use another known number n_2 instead of n_2, provided that $n_2 \ge n_1$. Such a number is for instance $n_2 := \frac{log\left\{\frac{\varepsilon}{(1-\alpha)^{-1}\cdot max_j\, r_j - M}\right\}}{log\,\alpha}$, where $\varepsilon > 0$ is a lower bound of the quantity $v_k^\alpha(M) - M$.

In actual computations n_2 turns out to be rather small. For example, if we denote $A := \log\{(1-\alpha)^{-1} \cdot max_j\, r_j - M$, and if the tolerance $\varepsilon = 10^{-2}$, then

$$n_2 = \begin{cases} 6.6 + 3.3A & \text{if } \alpha = 0.5 \\ 12.9 + 6.4A & \text{if } \alpha = 0.7 \\ 43.7 + 21.8A & \text{if } \alpha = 0.9 \end{cases}$$

We show in the next lemma that the conclusion $G_k \le M$ may be made also if for any n the value $r_k + \alpha \sum_j p_{kj} v_j^{n-1}(M)$ is sufficiently below M. The result of this lemma may be viewed as a suboptimality test.

Lemma 8.23

If for some n, $r_k + \alpha \sum_j p_{kj} v_j^{n-1}(M) \le M - \alpha^n \cdot \{(1-\alpha)^{-1} \cdot max_j\, r_j - M\}$, then $G_k \le M$.

Proof

Suppose that $G_k > M$. Then, we have $v_k^\alpha(M) = r_k + \alpha \sum_j p_{kj} v_j^\alpha(M) > M$.
Assuming $r_k + \alpha \sum_j p_{kj} v_j^{n-1}(M) \le M - \alpha^n \cdot \{(1-\alpha)^{-1} \cdot max_j\, r_j - M\}$, we obtain

$$\begin{aligned} \alpha \textstyle\sum_j p_{kj}\{v_j^\alpha(M) - v_j^{n-1}(M)\} &> (M - r_k) - (M - r_k - \alpha^n \cdot \{(1-\alpha)^{-1} \cdot max_j\, r_j - M\}) \\ &= \alpha^n \cdot \{(1-\alpha)^{-1} \cdot max_j\, r_j - M\}. \end{aligned}$$

Using the inequality $v_j^\alpha(M) \le (1-\alpha)^{-1} \cdot max_k\, r_k,\ j \in S$ (see the proof of Lemma 8.22), we have on the other side,

$$\begin{aligned} \alpha \textstyle\sum_j p_{kj}\{v_j^\alpha(M) - v_j^{n-1}(M)\} &\le \alpha \cdot \|v^\alpha(M) - v^{n-1}(M)\|_\infty \\ &\le \alpha^n \cdot \|v^\alpha(M) - v^0(M)\|_\infty \le \alpha^n \cdot \{(1-\alpha)^{-1} \cdot max_j\, r_j - M\}, \end{aligned}$$

which provides a contradiction. □

The bisection/successive approximation method contains a number of bisections until a *stopping criterion* is satisfied. A bisection is executed after a number of successive approximations, a number which is determined by an *approximation criterion.*
Possible stopping criteria are:

(1) the intervals $[L_i, U_i],\ i \in S$, are disjoint; this criterion is appropriate for ranking the Gittins indices.

(2) $U_i - L_i \leq \delta$ for all $i \in S$, where δ is a specified tolerance; this criterion is appropriate for approximating all Gittins indices.

(3) $U_i - L_i \leq \delta$ for some $i \in S$, where δ is a specified tolerance; this criterion is appropriate for approximating some Gittins index.

Possible approximation criterion are:

(1) When $n \geq n_2$, where $n_2 := \frac{log\left\{\frac{\varepsilon}{(1-\alpha)^{-1} \cdot max_j\, r_j - M}\right\}}{log\, \alpha}$, with $\varepsilon > 0$ is a specified lower bound of the quantity $v_i^\alpha(M) - M$.

(2) If $r_k + \frac{\alpha}{1-\alpha} \cdot min_j\, r_j \leq G_k \leq r_k + \frac{\alpha}{1-\alpha} \cdot max_j\, r_j$.

Below we present the bisection/successive approximation algorithm.

Algorithm 8.3 *The bisection/successive approximation method.*
Input: A multi-armed bandit problem, an approximation criterion and a stopping criterion.
Output: Ranking or approximation of the Gittins indices (depending on the chosen stopping criterion).

```
1. for all $i \in S$ do
       begin $L_i := max\{r_i + \frac{\alpha}{1-\alpha} \cdot min_k\, r_k, (1-\alpha)^{-1} \cdot r_i\}$; $U_i := r_i + \frac{\alpha}{1-\alpha} \cdot max_k\, r_k$;
           if $L_i = U_i$ then $G_i := L_i$ (the Gittins index in this state is computed).
       end
2. Select a state $k$ for which the Gittins index has to be approximated; $M := \frac{1}{2}(L_k + U_k)$.
3. $n := 0$; for all $j \in S$ do $v_j := M$;
4. for all $i \in S$ do
     (a) $s_i := r_i + \alpha \sum_j p_{ij} v_j$.
     (b) if $s_i > M$ then $L_i := \max\{L_i, M\}$.
     (c) if $s_i \leq M - \alpha^n \cdot \{(1-\alpha)^{-1} \cdot max_j\, r_j - M\}$ then $U_i := \min\{U_i, M\}$.
5. if the approximation criterion is not satisfied then
       begin $n := n + 1$; for all $i \in S$ do $v_i := \max\{M, s_i\}$; go to step 4 end
   else go to step 6.
6. if the stopping criterion is not satisfied then go to step 2.
   else STOP.
```

The selection of state k in step 2 is indicated by the stopping criterion. For example, under criterion (2), select k such that $U_k - L_k = \max_i (U_i - L_i)$.

One successive approximation requires $\mathcal{O}(|S^2|)$ operations. Therefore, the total computational effort is about $|S^2|$ times the number of successive approximations. To approximate with tolerance δ one Gittins index G_k, at most $log_2\{\frac{U_k - L_k}{\delta}\}$ midpoints M need to be considered, where $[L_k, U_k]$ is the initial interval.

Example 8.3 (continued)
As selection of state k in step 2, we select k such that $U_k - L_k = \max_i (U_i - L_i)$.
We will execute 3 approximations of Algorithm 8.3.
For the initial bounds $[L_i, U_i]$, we obtain $L_1 = 16,\ U_1 = 16;\ L_2 = 12,\ U_2 = 14;\ L_3 = 8,\ U_3 = 12$. Notice that the intervals $[L_i, U_i]$ are already disjoint and $G_1 = 16$. We select $k = 3$ and for the first M we take the midpoint of the largest interval $[L_3, U_3]$, i.e. $M = \frac{1}{2}(8 + 12) = 10$.
Approximation 1:
$n = 0:\ v_1 = v_2 = v_3 = 10;\ M - \alpha^n \cdot \{(1-\alpha)^{-1} \cdot max_j\, r_j - M = 4$.
$i = 1:\ s_1 = 8 + \frac{1}{2} \cdot 10 = 13;\ L_1 = \max(16, 10) = 16$.
$i = 2:\ s_2 = 6 + \frac{1}{2} \cdot 10 = 11;\ L_2 = \max(12, 10) = 12$.
$i = 3:\ s_3 = 4 + \frac{1}{2} \cdot 10 = 9$. *Approximation 2:*
$n = 1:\ v_1 = 13,\ v_2 = 11,\ v_3 = 10;\ M - \alpha^n \cdot \{(1-\alpha)^{-1} \cdot max_j\, r_j - M = 7$.
$i = 1:\ s_1 = 8 + \frac{1}{2} \cdot 11 = \frac{27}{2};\ L_1 = \max(16, 10) = 16$.
$i = 2:\ s_2 = 6 + \frac{1}{2} \cdot 10 = 11;\ L_2 = \max(12, 10) = 12$.
$i = 3:\ s_3 = 4 + \frac{1}{2} \cdot 13 = \frac{21}{2}$. *Approximation 3:*
$n = 2:\ v_1 = \frac{27}{2},\ v_2 = 11,\ v_3 = \frac{21}{2};\ M - \alpha^n \cdot \{(1-\alpha)^{-1} \cdot max_j\, r_j - M = \frac{17}{2}$.
$i = 1:\ s_1 = 8 + \frac{1}{2} \cdot 11 = \frac{27}{2};\ L_1 = \max(16, 10) = 16$.
$i = 2:\ s_2 = 6 + \frac{1}{2} \cdot \frac{21}{2} = \frac{45}{4};\ L_2 = \max(12, 10) = 12$.
$i = 3:\ s_3 = 4 + \frac{1}{2} \cdot \frac{27}{2} = \frac{43}{4}$.
After these three approximations we have the intervals $[L_1, U_1] = [16, 16],\ [L_2, U_2] = [12, 14],\ [L_3, U_3] = [\frac{43}{4}, 12]$.

8.7 Separable problems

8.7.1 Introduction

Separable MDPs have the property that for certain pairs $(i, a) \in S \times A$:

(1) the immediate reward is the sum of tow terms, one depends only on the current state and the the other depends only on the chosen action: $r_i(a) = s_i + t_a$.

(2) the transition probabilities depend only on the action and not on the state from which the transition occurs: $p_{ij}(a) = p_j(a),\ j \in S$.

Let $S_1 \times A_1$ be the subset of $S \times A$ for which the pairs (i, a) satisfy (1) and (2). We also assume that the action sets of A_1 are *nested*: let $S_1 = \{1, 2, \dots, m\}$, then $A_1(1) \supseteq A_1(2) \supseteq \cdots \supseteq A_1(m) \neq \emptyset$.
Let $S_2 := S \backslash S_1,\ A_2(i) := A(i) \backslash A_1(i),\ 1 \leq i \leq m$ and $A_2(i) := A(i),\ m + 1 \leq i \leq N$. We also introduce the notations $B(i) := A_1(i) - A_{i+1}(i),\ 1 \leq i \leq m - 1$ and $B(m) := A_1(m)$. Then, $A_1(i) = \bigcup_{j=i}^{m} B(j)$ and the sets $B(j)$ are disjunct. We allow that S_2, A_2 or $B(i)$ is an empty set.

If the system is observed in state $i \in S_1$ and the decision maker will choose an action from $A_1(i)$, then, the decision process can be considered as follows. First, a reward s_i is earned and the system makes a zero-time transition to an additional state $N + i$. In this additional state there are two options: either to take an action $a \in B(i)$ or to take an action $a \in A_1(i) \backslash B(i) = A_1(i + 1)$. In the first case the reward t_a is earned and the process moves to state j with probability $p_j(a),\ j \in S$; in the second case we are in the same situation as in state $N + i + 1$, i.e. a zero-time transition is made from state $N + i$ to state $N + i + 1$.

8.7.2 Examples (part 1)

1. Automobile replacement problem

We own a car of a certain age. Our decision problem is to keep it or sell it and, if we sell, what age car to replace it with. Let us agree to review the number of states down, we assume that every car breaks down irreparably as soon as it becomes 10 years old. Number the states from 0 to 40. For $i \leq 39$, state i refers to a car that is $3i$ months old: we say this car is "of age i". State 40 indicates a car that has just become 10 years and, therefore, has just broken down irreparably. At state $i \leq 39$ we can make decision r (for retain) to keep our current car for at least one more time period, or we can trade it in on a car of age k with $0 \leq k \leq 39$. Since there are 41 possible decisions for states 0 through 39, namely $A(i) = \{r, 0, 1, \dots, 39\}$, and 40 possible decisions for state 40, namely $A(40) = \{0, 1, \dots, 39\}$, there are nearly 41^{41} different policies.

Consider the following relevant data:

c_j = the cost of buying a car of age j, $0 \leq j \leq 39$;
t_i = the trade-in value of a car of age i, $0 \leq i \leq 40$;
e_i = the expected cost of operating a car of age i for one time period, $0 \leq i \leq 39$;
p_i = the probability that a car of age i will last at least one more time period, $0 \leq i \leq 39$;

To simplify things, assume that if a car fails to survive a time period, it ages at the end of that period to 10 years, causing a transfer to state 40. Trade-ins take place at the beginning of time periods. If the retain decision r is selected in state i, the rewards and transition probabilities are:

$$r_i(r) = -e_i;\; p_{ij}(r) = \begin{cases} p_i & \text{if } j = i+1; \\ 1-p_i & \text{if } j = 40; \\ 0 & \text{if } j \neq i+1,\, 40. \end{cases} \qquad \text{with } p_{39,40}(r) = 1.$$

Similarly, if a trade-in decision a is selected in state i, the replacement car of age a must be kept for at least one time period and the rewards and transition probabilities are:

$$r_i(a) = t_i - c_a - e_a;\; p_{ij}(a) = \begin{cases} p_a & \text{if } j = a+1; \\ 1-p_a & \text{if } j = 40; \\ 0 & \text{if } j \neq i+1,\, 40. \end{cases} \qquad \text{with } p_{i,40}(39) = 1.$$

For a replacement decision $a \neq r$ in state $0 \leq i \leq 39$, the transition probabilities depend only on a and not on i and the expected reward $r_i(a) = s_i + t_a$ with $s_i = t_i$ and $t_a = -(c_a + e_a)$. Since we also have $A_1(i) = \{0, 1, \dots, 40\}$ for $i = 0, 1, \dots, 40$, the problem is separable with $S_1 = S = \{0, 1, \dots, 40\}$ and $A_1(i) = \{0, 1, \dots, 39\}$, $0 \leq i \leq 40$; $A_2(i) = \{r\}$, $0 \leq i \leq 39$, $A_2(40) = \emptyset$; $B(i) = \emptyset$, $0, 1, \dots, 39$ and $B(40) = \{0, 1, \dots, 39\}$. Ignoring those i's for which $B(i) = \emptyset$, we require a single extra state. Call it state 41. In economic terms, state 41 corresponds to having no car and needing to buy one immediately. Instantaneous transition to state 41 is available from states 0 through 40 by trading in one's current car and this transition is mandatory from state 40.

Hence, we may consider the model as an MDP with state space $S^* = \{0, 1, \dots, 41\}$, action sets

$$A^*(i) = \begin{cases} \{r, t\} & i = 0, 1, \dots, 39 \\ \{t\} & i = 40 \\ \{0, 1, \dots, 39\} & i = 41 \end{cases} \qquad \text{with decision } t \text{ can be interpreted as the trade-in with}$$

in state i reward t_i and an immediate transition to the no-car state 41. Purchase decision a from state 41 has expected reward $-(c_a + e_a)$ and transition probability p_a to state $a+1$ and $1 - p_a$ to state 40. The transformed problem affords a reduction in the number of policies from 41^{41} to $40 \cdot 2^{40}$ and a reduction in the total number of decisions from 1680 $(40 \cdot 41 + 40)$ to 121 $(40 \cdot 2 + 1 + 40)$, while increasing the number of states from 41 to 42.

The reduction in problem size is so drastic that we can be assured of substantial computational savings in the policy iteration, linear programming and value iteration methods for both discounted and averaging versions of the probem. Note finally that every sequence of two transitions gives rise to at least one change in epoch, a fact that we shall find useful in the analysis.

2. Inventory problem

For a prototype of a variety of inventory models that are separable, consider the model with integer on-hand quantities, instantaneous replenishment, linear ordering cost, a set-up charge if any units are ordered, excess sales lost and a storage capacity of N items. State i denotes i units on hand at the beginning of the period, and a set $A(i)$ of available decisions from state i is given by $A(i) = \{i, i+1, \dots, N\}$ with $a \in A(i)$ denoting either an order of $a - i$ items if $a > i$ or no order if $a = i$. The ordering cost is 0 if $a = i$ and $K + k(a - i)$ if $a > i$, where K and k are, respectively, the set-up and per item ordering cost. Let p_j be the probability that j sales opportunities appear during the time period, and let h_a be the expected one-period holding and sales cost given a items on hand at the beginning of the period, immediately after delivery of the order.

If no order is placed, i.e. $a = i$, the expexted one-period reward $r_i(i) = -h_i$ and the transition probabilities are given by $p_{ij}(i) = \begin{cases} p_{i-j} & \text{if } 1 \le j \le i; \\ \sum_{k \ge i} p_k & \text{if } j = 0; \\ 0 & \text{if } j > i. \end{cases}$

Similarly, if $a > i$, the expected one-period reward and the transition probabilities are, respectively, given by $r_i(a) = -K - k(a - i) - h_a$ and $p_{ij}(a) = \begin{cases} p_{a-j} & \text{if } 1 \le j \le a; \\ \sum_{k \ge a} p_k & \text{if } j = 0; \\ 0 & \text{if } j > a. \end{cases}$

In the case that $a > i$, the transition probabilities $p_{ij}(a),\ j \in S$ are independent of i and the expected reward $r_i(a) = s_i + t_a$ with $s_i = -K + k \cdot i$ and $t_a = -k \cdot a - h_a$ Then, with $A_1(i) = \{i+1, i+2, \dots, N\}$ for $0 \le i \le N - 1$ and $A_2(i) = \{i\}$ for $0 \le i \le N - 1$, the problem satisfies the conditions of separability with $B(i) = \{i + 1\}$ for $i = 0, 1, \dots, N - 1$. The reduction in this problem is less dramatic as in the automobile replacement problem: the state space increases from $N + 1$ to $2N + 1$ and the total number of decisions is reduced from $\sum_{i=0}^{N} (N - i + 1) = \frac{1}{2}(N + 1)(N + 2)$ to $4N + 1$ (two options in the states $0, 1, \dots, N - 1$, two options in the N additional states and one option in state N).

8.7.3 Discounted rewards

The description in section 8.7.1 as a problem with zero-time and one-time transitions gives rise to consider the transformed model with $N + m$ states and to the following linear program for the computation of the value vector v^α.

$$min \left\{ \sum_{i=1}^{N} v_i + \sum_{i=1}^{m} y_i \;\middle|\; \begin{array}{lcll} v_i & \ge & r_i(a) + \alpha \sum_{j=1}^{N} p_{ij}(a) v_j & 1 \le i \le N,\ a \in A_2(i) \\ v_i & \ge & s_i + y_i & 1 \le i \le m \\ y_i & \ge & t_a + \alpha \sum_{j=1}^{N} p_j(a) v_j & 1 \le i \le m,\ a \in B(i) \\ y_i & \ge & y_{i+1} & 1 \le i \le m - 1 \end{array} \right\}. \tag{8.92}$$

The first set of inequalities corresponds to the non-separable set $S \times A_2$ with one-time transitions; the second set inequalities to the zero-time transitions from the state i to $N + i,\ 1 \le i \le m$; the third set of inequalities to the set $S_1 \times B$ with one-time transitions and the last set inequalities corresponds to the zero-time transitions from the state $N + i$ to $N + i + 1,\ 1 \le i \le m - 1$.

The dual of program (8.92), where the dual variables $x_i(a),\ \lambda_i,\ w_i(a),\ \rho_i$ correspond to the four sets of constraints in (8.92), is:

$$max \sum_{i=1}^{N} \sum_{a \in A_2(i)} r_i(a)x_i(a) + \sum_{i=1}^{m} s_i\lambda_i + \sum_{i=1}^{m} \sum_{a \in B(i)} w_i(a) \tag{8.93}$$

subject to the constraints

$$\begin{array}{rcrcrcl} \sum_{i=1}^N \sum_{a \in A_2(i)} \{\delta_{ij} - \alpha p_{ij}(a)\}x_i(a) & + & \sum_{i=1}^m \delta_{ij}\lambda_i & - & \sum_{i=1}^m \sum_{a \in B(i)} p_j(a)w_i(a) & = & 1,\ 1 \le j \le N \\ \rho_j - \rho_{j-1} & - & \lambda_j & + & \sum_{a \in B(j)} w_j(a) & = & 1,\ 1 \le j \le m-1 \\ -\rho_{m-1} & - & \lambda_m & + & \sum_{a \in B(m)} w_m(a) & = & 1 \end{array}$$

$x_i(a) \ge 0,\ 1 \le i \le N,\ a \in A_2(i);\ \lambda_i \ge 0,\ 1 \le i \le m;\ w_i(a) \ge 0,\ 1 \le i \le m,\ a \in B(i);$
$\rho_i \ge 0,\ 1 \le i \le m-1.$

Without using the transformed problem, the linear program to compute the value vector v^α is:

$$min\Big\{ \sum_{i=1}^{N} v_i \ \Big|\ v_i \ge r_i(a) + \alpha \sum_{j=1}^{N} p_{ij}(a)v_j,\ 1 \le i \le N,\ a \in A(i)\Big\}. \tag{8.94}$$

Lemma 8.24
Let v feasible for (8.94) and define y by $y_i = max_{a \in A_1(i)} \{t_a + \alpha \sum_{j=1}^N p_j(a)v_j\},\ 1 \le i \le m$. Then,
(1) (v, y) is a feasible solution of (8.92).
(2) $\sum_{i=1}^N v_i + \sum_{i=1}^m y_i \ge \sum_{i=1}^N v_i^\alpha + \sum_{i=1}^m max_{a \in A_1(i)} \{t_a + \alpha \sum_{j=1}^N p_j(a)v_j^\alpha\}$.

Proof
First we have to show that (v, y) satisfies the four parts of the constraints of (8.92). The first and third part are obviously satisfied. For the second part, notice that for all $1 \le i \le m$ and $a \in A_1(i)$ we have

$$v_i \ge s_i + t_a + \alpha \sum_{j=1}^N p_j(a)v_j \ \rightarrow\ v_i \ge s_i + max_{a \in A_1(i)}\{t_a + \alpha \sum_{j=1}^N p_j(a)v_j\} = s_i + y_i.$$

For the fourth part, we write for $i = 1, 2, \dots, m-1$

$$y_i - y_{i+1} = max_{a \in A_1(i)} \{t_a + \alpha \sum_{j=1}^N p_j(a)v_j\} - max_{a \in A_1(i+1)} \{t_a + \alpha \sum_{j=1}^N p_j(a)v_j\} \ge 0,$$

the last inequality because $A_1(i+1) \subseteq A_1(i)$.

Finally, because v^α is the componentwise smallest solution of (8.94), cf. Theorem 3.16, we have $v_i \ge v_i^\alpha,\ 1 \le i \le N$, and consequently,

$$\begin{array}{rcl} \sum_{i=1}^N v_i + \sum_{i=1}^m y_i & = & \sum_{i=1}^N v_i + \sum_{i=1}^m max_{a \in A_1(i)} \{t_a + \alpha \sum_{j=1}^N p_j(a)v_j\} \\ & \ge & \sum_{i=1}^N v_i^\alpha + \sum_{i=1}^m max_{a \in A_1(i)} \{t_a + \alpha \sum_{j=1}^N p_j(a)v_j^\alpha\}. \end{array}$$

□

Corollary 8.4
Since v^α is the unique optimal solution of (8.94), we have shown that (v^α, y^α) is the unique optimal solution of (8.92), where $y_i^\alpha = max_{a \in A_1(i)} \{t_a + \alpha \sum_{j=1}^N p_j(a)v_j^\alpha\},\ 1 \le i \le m$.

Theorem 8.26
Let (x, λ, w, ρ) be an optimal solution of linear program (8.93). Define $S_x := \{j \mid \sum_{a \in A_2(j)} x_j(a) > 0\}$ and $k_j := min\{k \ge j \mid \sum_{a \in B(k)} w_k(a) > 0\},\ j \in S \backslash S_x$. Take any policy $f^\infty \in C(D)$ such that $x_j\big(f(j)\big) > 0$ if $j \in S_x$ and $w_{k_j}\big(f(j)\big) > 0$ if $j \in S \backslash S_x$. Then, f^∞ is well-defined and an α-discounted optimal policy.

Proof

From the definition of S_x it follows that $f(j)$ is well-defined if $j \in S_x$. From the first set of the constraints of (8.93) it follows that for $j = 1, 2, \dots, N$, we have

$$\textstyle\sum_{a \in A_2(j)} x_j(a) + \sum_{i=1}^m \delta_{ij}\lambda_i \geq 1 + \alpha \sum_{i=1}^N \sum_{a \in A_2(i)} p_{ij}(a)x_i(a) + \sum_{i=1}^m \sum_{a \in B(i)} p_j(a)w_i(a) \geq 1.$$

Therefore, if $j \notin S_*$, then $1 \leq j \leq m$ and $\lambda_j > 0$. Then, by adding the corresponding last constraints of (8.93), we obtain

$$\begin{array}{rcl} \sum_{k=j}^m \sum_{a \in B(k)} w_k(a) & = & \sum_{k=j}^{m-1} \{1 + \lambda_k + (\rho_{k-1} - \rho_k)\} + \{1 + \lambda_m + \rho_{m-1}\} \\ & = & \sum_{k=j}^m \{1 + \lambda_k\} + \rho_{j-1} > 0. \end{array}$$

Hence, k_j is well-defined, and therefore the policy f^∞ is well-defined.

For the proof of the optimality of f^∞ we first consider a state $i \in S_x = \{j \mid \sum_{a \in A_2(j)} x_j(a) > 0\}$. In such state i, we have $x_i\big(f(i)\big) > 0$ and, by the complementary slackness property of linear programming, $v_i^\alpha = r_i(f) + \alpha \sum_{j=1}^N p_{ij}(f)v_j^\alpha$.

Then, we show that in a state $i \in S \backslash S_x$ we also have $v_i^\alpha = r_i(f) + \alpha \sum_{j=1}^N p_{ij}(f)v_j^\alpha$. Consider a state $i \in S \backslash S_x$, i.e. $1 \leq i \leq m$, $\sum_{a \in A_2(j)} x_j(a) = 0$ and $\lambda_i > 0$. Let $w_k\big(f(i)\big) > 0$, i.e. $\sum_{a \in B(j)} w_j(a) = 0$ for $j = i, i+1, \dots, k-1$ and $\sum_{a \in B(k} w_k(a) > 0$. By the complementary slackness property of linear programming, we have

$$\begin{array}{lcl} \lambda_i > 0 & \rightarrow & v_i^\alpha = s_i + y_i^\alpha \\ \sum_{a \in B(j)} w_j(a) = 0,\ j = i, i+1, \dots, k-1 & \rightarrow & y_i^\alpha = y_{i+1}^\alpha = \cdots = y_k^\alpha \\ w_k\big(f(i)\big) > 0 & \rightarrow & y_k^\alpha = t_{f(i)} + \alpha \sum_{j=1}^N p_j(f)v_j^\alpha \end{array}$$

Hence,

$$v_i^\alpha = s_i + y_i^\alpha = v_i^\alpha = s_i + y_k^\alpha = s_i + t_{f(i)} + \alpha \textstyle\sum_{j=1}^N p_j(f)v_j^\alpha = r_i\big(f(i)\big) + \alpha \sum_{j=1}^N p_{ij}(f)v_j^\alpha.$$

Therefore, we have shown that $v_i^\alpha = r_i(f) + \alpha \sum_{j=1}^N p_{ij}(f)v_j^\alpha,\ i \in S$, in vector notation

$$v^\alpha = r(f) + \alpha P(f)v^\alpha \ \rightarrow\ \{I - P(f)\}v^\alpha = r(f) \ \rightarrow\ v^\alpha = \{I - P(f)\}^{-1}r(f) = v^\alpha(f^\infty),$$

i.e. f^∞ is an α-discounted optimal policy. □

8.7.4 Average rewards - unichain case

Consider the problem again in the transformed model with $N + m$ states and with zero-time and one-time transitions. This interpretation gives rise to the following linear program for the computation of the value vector ϕ.

$$min\left\{ x \;\middle|\; \begin{array}{cccll} x + & y_i & \geq & r_i(a) + \sum_{j=1}^N p_{ij}(a)y_j & 1 \leq i \leq N,\ a \in A_2(i) \\ & y_i & \geq & s_i + z_i & 1 \leq i \leq m \\ x + & z_i & \geq & t_a + \sum_{j=1}^N p_j(a)y_j & 1 \leq i \leq m,\ a \in B(i) \\ & z_i & \geq & z_{i+1} & 1 \leq i \leq m-1 \end{array} \right\}. \tag{8.95}$$

The dual of program (8.95), where the dual variables $x_i(a),\ \lambda_i,\ w_i(a),\ \rho_i$ correspond to the four sets of constraints in (8.95), is:

$$max \sum_{i=1}^N \sum_{a \in A_2(i)} r_i(a)x_i(a) + \sum_{i=1}^m s_i\lambda_i + \sum_{i=1}^m \sum_{a \in B(i)} w_i(a) \tag{8.96}$$

subject to the constraints

$$\begin{array}{rcrcrcl}
\sum_{i=1}^N \sum_{a\in A_2(i)}\{\delta_{ij} - p_{ij}(a)\}x_i(a) & + & \sum_{i=1}^m \delta_{ij}\lambda_i & - & \sum_{i=1}^m \sum_{a\in B(i)} p_j(a)w_i(a) & = & 0,\ 1\le j\le N \\
\rho_j - \rho_{j-1} & - & \lambda_j & + & \sum_{a\in B(j)} w_j(a) & = & 0,\ 1\le j\le m-1 \\
-\rho_{m-1} & - & \lambda_m & + & \sum_{a\in B(m)} w_m(a) & = & 0 \\
\sum_{i=1}^N \sum_{a\in A_2(i)} x_i(a) & & & + & \sum_{i=1}^m \sum_{a\in B(i)} w_i(a) & = & 1
\end{array}$$

$x_i(a) \ge 0,\ 1\le i\le N,\ a\in A_2(i);\ \lambda_i \ge 0,\ 1\le i\le m;\ w_i(a)\ge 0,\ 1\le i\le m,\ a\in B(i);$

$\rho_0 = 0;\ \rho_i \ge 0,\ 1\le i\le m-1.$

Without using the transformed problem, the linear program to compute the value ϕ is:

$$min\Big\{x \ \Big|\ x + y_i \ge r_i(a) + \sum_{j=1}^N p_{ij}(a)y_j,\ 1\le i\le N,\ a\in A(i)\Big\}. \tag{8.97}$$

Lemma 8.25

Let (x,y) feasible for (8.97) and define z by $z_i = max_{a\in A_1(i)}\{t_a + \sum_{j=1}^N p_j(a)y_j\} - x,\ 1\le i\le m$. Then, (x,y,z) is a feasible solution of (8.95) and $x\ge\phi$.

Proof

First we have to show that (x,y,z) satisfies the four parts of the constraints of (8.95). The first and third part are obviously satisfied. For the second part, notice that for all $i=1,2,\dots,m$ and $a\in A_1(i)$ we have

$$x + y_i \ge s_i + t_a + \textstyle\sum_{j=1}^N p_j(a)y_j \ \to\ y_i \ge s_i + max_{a\in A_1(i)}\{t_a + \sum_{j=1}^N p_j(a)y_j\} - x = s_i + z_i.$$

For the fourth part, we write for $i=1,2,\dots,m-1$

$$z_i - z_{i+1} = max_{a\in A_1(i)}\{t_a + \textstyle\sum_{j=1}^N p_j(a)y_j\} - max_{a\in A_1(i+1)}\{t_a + \sum_{j=1}^N p_j(a)y_j\} \ge 0,$$

the last inequality because $A_1(i+1)\subseteq A_1(i)$. Finally, because ϕ is the optimal solution of (8.97), we have $x\ge\phi$. □

Corollary 8.5

Since any optimal solution (x^*,y^*) of linear program (8.97) satisfies $x^*=\phi$, the optimum value of (8.95) is also ϕ. Furthermore, $(x^*=\phi,y^*,z^*)$ is an optimal solution of program (8.95), where z^* is defined by $z_i^* := max_{a\in A_1(i)}\{t_a + \sum_{j=1}^N p_j(a)y_j^*\} - \phi,\ 1\le i\le m$.

Theorem 8.27

Let (x,λ,w,ρ) be an optimal solution of linear program (8.96). Define $S_x := \{j \mid \sum_{a\in A_2(j)} x_j(a) > 0\}$ and $k_j := min\{k\ge j \mid \sum_{a\in B(k)} w_k(a) > 0\},\ j\in S_w$, where $S_w := \{j\in S\backslash S_x \mid \sum_{a\in A_1(j)} w_j(a) > 0\}$. Take any policy $f^\infty \in C(D)$ such that $x_j\big(f(j)\big) > 0$ if $j\in S_x$, $w_{k_j}\big(f(j)\big) > 0$ if $j\in S_w$ and $f(j)$ arbitrarily chosen if $j\notin S_x\cup S_w$. Then, f^∞ is an average optimal policy.

Proof

Let (ϕ,y,z) be an optimal solution of (8.96). Then, by the complementary slackness property of linear programming, we have

$$x_i(a)\cdot\{\phi + y_i - r_i(a) - \sum_{j=1}^N p_{ij}(a)y_j\} = 0,\ 1\le i\le N;\ a\in A_2(i) \tag{8.98}$$

$$\lambda_i\cdot\{y_i - s_i - z_i\} = 0,\ 1\le i\le m \tag{8.99}$$

$$w_i(a)\cdot\{\phi + z_i - t_a - \sum_{j=1}^N p_j(a)y_j\} = 0,\ 1\le i\le m;\ a\in B(i) \tag{8.100}$$

$$\rho_i\cdot\{z_i - z_{i+1}\} = 0,\ 1\le i\le m-1 \tag{8.101}$$

Let $S_+ := \{j \in S \mid \sum_{a\in A_2(j)} x_j(a) + \lambda_j > 0\}$ and take any $i \in S_+$.
If $\sum_{a\in A_2(i)} x_i(a) > 0$, then from equation (8.98), we obtain

$$\phi = r_i(f) - y_i + \sum_{j=1}^N p_{ij}(f)y_j\,,\ i \in S_x. \tag{8.102}$$

If $\sum_{a\in A_2(i)} x_i(a) = 0$, then $1 \leq i \leq m,\ \lambda_i > 0$, and equation (8.99) gives $y_i = s_i + z_i$.
Furthermore, we have $\sum_{a\in A_1(i)} w_i(a) > 0$, namely: adding the corresponding constraints of (8.96) yields $\sum_{a\in A_1(i)} w_i(a) = \sum_{j=i}^m \sum_{a\in B(j)} w_j(a) = \sum_{j=i}^m \lambda_j + \rho_{i-1} \geq \lambda_i > 0$. The definition of k_i implies, denoting k_i by k, that $\sum_{a\in B(j)} w_j(a) = 0$ for $j = i, i+1, \dots, k-1$.
Hence, by the constraints of program (8.96), we obtain $\rho_j = \lambda_j + \rho_{j-1}$ for $j = i, i+1, \dots, k-1$, implying $\rho_i = \lambda_i + \rho_{i-1} \geq \lambda_i > 0,\ \rho_{i+1} = \lambda_{i+1} + \rho_i \geq \rho_i > 0,\ \dots,\ \rho_{k-1} = \lambda_{k-1} + \rho_{k-2} \geq \rho_{k-2} > 0$. Then, it follows from (8.101) that $z_i = z_{i+1} = \cdots = z_k = 0$. Since $w_k\big(f(i)\big) > 0$, by (8.100), we can write

$$\phi = t_{f(i)} + \textstyle\sum_{j=1}^N p_j(f(i))y_j - z_k = t_{f(i)} + \sum_{j=1}^N p_j(f(i))y_j - z_i = s_i + t_{f(i)} + \sum_{j=1}^N p_j(f(i))y_j - y_i,$$

implying

$$\phi = r_i(f) - y_i + \sum_{j=1}^N p_{ij}(f)y_j\,,\ i \in S_+\backslash S_x. \tag{8.103}$$

Combining (8.102) and (8.103) yields

$$\phi = r_i(f) - y_i + \sum_{j=1}^N p_{ij}(f)y_j\,,\ i \in S_+. \tag{8.104}$$

Next, we show that S_+ is closed under $P(f)$, i.e. $p_{ij}(f) = 0,\ i \in S_+,\ j \notin S_+$. Suppose that $p_{ij}(f) > 0$ for some $i \in S_+$ and $j \notin S_+$. Since $j \notin S_+,\ \sum_{a\in A_2(j)} x_j(a) + \sum_{i=1}^m \delta_{ij}\lambda_i = 0$.
If $i \in S_x$, then, from the constraints of program (8.96) it follows that,

$$\begin{aligned} 0 &= \textstyle\sum_{a\in A_2(j)} x_j(a) + \sum_{i=1}^m \delta_{ij}\lambda_i \\ &= \textstyle\sum_{i=1}^N \sum_{a\in A_2(i)} p_{ij}(a)\}x_i(a) + \sum_{i=1}^m \sum_{a\in B(i)} p_j(a)w_i(a) \geq p_{ij}(f)x_i(f(i)) > 0, \end{aligned}$$

implying a contradiction.
If $i \in S_+\backslash S_x$, then, from the constraints of program (8.96) it follows that,

$$\begin{aligned} 0 &= \textstyle\sum_{a\in A_2(j)} x_j(a) + \sum_{i=1}^m \delta_{ij}\lambda_i \\ &= \textstyle\sum_{i=1}^N \sum_{a\in A_2(i)} p_{ij}(a)\}x_i(a) + \sum_{i=1}^m \sum_{a\in B(i)} p_j(a)w_i(a) \geq p_j(f(i))w_{k_i}(f(i)) > 0, \end{aligned}$$

which also implies a contradiction.
Since $P(f)$ is a unichain Markov chain and S_+ is closed, the states of $S\backslash S_+$ are transient. Let $\pi(f)$ be the stationary distribution of the unichain Markov chain $P(f)$, then it follows from (8.104) that

$$\phi \cdot e = \phi \cdot \{\pi(f)^T e\} \cdot e = \pi(f)^T\{r(f) - y + P(f)y\} \cdot e = \phi(f^\infty) \cdot e,$$

i.e. f^∞ is an average optimal policy. □

8.7.5 Average rewards - general case

Again, the interpretation of the transformed model gives rise to consider the following linear program in order to compute the value vector ϕ.

$$min\left\{\sum_{j=1}^N x_j+\sum_{j=1}^m w_j \;\middle|\; \begin{array}{lllllll} x_i & & & & \geq & \sum_{j=1}^N p_{ij}(a)x_j & 1\leq i\leq N,\ a\in A_2(i)\\ x_i & & & & \geq & w_i & 1\leq i\leq m\\ & w_i & & & \geq & \sum_{j=1}^N p_j(a)x_j & 1\leq i\leq m,\ a\in B(i)\\ & w_i & & & \geq & w_{i+1} & 1\leq i\leq m-1\\ x_i & & +\ y_i & & \geq & r_i(a)+\sum_{j=1}^N p_{ij}(a)y_j & 1\leq i\leq N,\ a\in A_2(i)\\ & & y_i & & \geq & s_i+z_i & 1\leq i\leq m\\ & w_i & & +\ z_i & \geq & t_a+\sum_{j=1}^N p_j(a)y_j & 1\leq i\leq m,\ a\in B(i)\\ & & & z_i & \geq & z_{i+1} & 1\leq i\leq m-1 \end{array}\right\}. \tag{8.105}$$

The dual of program (8.105), where the dual variables $y_i(a),\ \mu_i,\ z_i(a),\ \sigma_i,\ x_i(a),\ \lambda_i,\ w_i(a),\ \rho_i$ correspond to the eight sets of constraints in (8.105), is:

$$max\sum_{i=1}^N\sum_{a\in A_2(i)} r_i(a)x_i(a)+\sum_{i=1}^m s_i\lambda_i+\sum_{i=1}^m\sum_{a\in B(i)} t_aw_i(a) \tag{8.106}$$

subject to the constraints

$$\begin{array}{rcrcrcrcl} \sum_{i=1}^N\sum_{a\in A_2(i)}\{\delta_{ij}-p_{ij}(a)\}y_i(a) & + & \sum_{i=1}^m\delta_{ij}\mu_i & - & \sum_{i=1}^m\sum_{a\in B(i)}p_j(a)z_i(a) & + & \sum_{a\in A_2(i)}x_j(a) & = & 1,\ 1\leq j\leq N\\ \sigma_j-\sigma_{j-1} & - & \mu_j & + & \sum_{a\in B(j)}w_j(a) & + & \sum_{a\in B(j)}z_j(a) & = & 1,\ 1\leq j\leq m\\ \sum_{i=1}^N\sum_{a\in A_2(i)}\{\delta_{ij}-p_{ij}(a)\}x_i(a) & + & \sum_{i=1}^m\delta_{ij}\lambda_i & - & \sum_{i=1}^m\sum_{a\in B(i)}p_j(a)w_i(a) & & & = & 0,\ 1\leq j\leq N\\ \rho_j-\rho_{j-1} & - & \lambda_j & + & \sum_{a\in B(j)}w_j(a) & & & = & 0,\ 1\leq j\leq m \end{array}$$

$\rho_0=\rho_m=\sigma_0=\sigma_m=0;\ x_i(a),\ y_i(a),\ z_i(a),\ w_i(a),\ \lambda_i,\ \mu_i,\ \rho_i,\ \sigma_i\geq 0$ for all i and a.

Without using the transformed problem, the linear program to compute the value ϕ is:

$$min\left\{\sum_{j=1}^N x_j \;\middle|\; \begin{array}{lcrcl} \sum_{j=1}^N\{\delta_{ij}-p_{ij}(a)\}x_j & & \geq & 0 & 1\leq i\leq N,\ a\in A(i)\\ x_i & + & \sum_{j=1}^N\{\delta_{ij}-p_{ij}(a)\}u_j \ \geq & r_i(a) & 1\leq i\leq N,\ a\in A(i) \end{array}\right\}. \tag{8.107}$$

Lemma 8.26

Let (x,u) be a feasible solution of program (8.107) and define w,y,z by $y_i:=x_i+u_i,\ 1\leq i\leq N$, $w_i:=max_{a\in A_1(i)}\sum_{j=1}^N p_j(a)x_j,\ 1\leq i\leq m$ and $z_i:=max_{a\in A_1(i)}\{t_a+\sum_{j=1}^N p_j(a)u_j\},\ 1\leq i\leq m$. Then,

(1) (x,w,y,z) is a feasible solution of linear program (8.105).

(2) $\sum_{j=1}^N x_j+\sum_{j=1}^m w_j\geq\sum_{j=1}^N\phi_j+\sum_{j=1}^m max_{a\in A_1(j)}\sum_{k=1}^N p_k(a)\phi_k$.

Proof

The proof of part (1) consists of the verification of the eight sets of the constraints of (8.105).

a. $x_i\geq\sum_{j=1}^N p_{ij}(a)x_j,\ 1\leq i\leq N,\ a\in A_2(i)$ (since (x,u) is feasible of (8.107)).

b.
$$\begin{aligned} x_i-w_i &= x_i-max_{a\in A_1(i)}\sum_{j=1}^N p_j(a)x_j\\ &\geq max_{a\in A(i)}\sum_{j=1}^N p_j(a)\,x_j-max_{a\in A_1(i)}\sum_{j=1}^N p_j(a)\,x_j\geq 0,\ 1\leq i\leq m. \end{aligned}$$

(the first inequality because (x,u) is feasible of (8.107)).

c. $w_i - \sum_{j=1}^N p_j(a)x_j = max_{a \in A_1(i)} \sum_{j=1}^N p_j(a)x_j - \sum_{j=1}^N p_j(a)x_j \geq 0,\ 1 \leq i \leq m,\ a \in B(i).$

d. $w_i - w_{i+1} = max_{a \in A_1(i)} \sum_{j=1}^N p_j(a)x_j - max_{a \in A_1(i+1)} \sum_{j=1}^N p_j(a)x_j \geq 0,\ 1 \leq i \leq m-1$ (because $A_1(i+1) \subseteq A_1(i)$).

e.
$$\begin{aligned} x_i + \textstyle\sum_{j=1}^N \{\delta_{ij} - p_{ij}(a)\}y_j &= x_i + \textstyle\sum_{j=1}^N \{\delta_{ij} - p_{ij}(a)\}(x_j + u_j) \\ &\geq x_i + \textstyle\sum_{j=1}^N \{\delta_{ij} - p_{ij}(a)\}u_j \geq r_i(a),\ 1 \leq i \leq N,\ a \in A_2(i). \end{aligned}$$

f.
$$\begin{aligned} y_i - z_i &= x_i + u_i - max_{a \in A_1(i)} \{t_a + \textstyle\sum_{j=1}^N p_j(a)u_j\} \\ &= min_{a \in A_1(i)} \{x_i + u_i - t_a - \textstyle\sum_{j=1}^N p_j(a)u_j\} \\ &\geq min_{a \in A_1(i)}\{r_i(a) - t_a\} = min_{a \in A_1(i)}\, s_i = s_i,\ 1 \leq i \leq m. \end{aligned}$$

g.
$$\begin{aligned} w_i + z_i - \textstyle\sum_{j=1}^N p_j(a)y_j &= max_{a \in A_1(i)} \textstyle\sum_{j=1}^N p_j(a)x_j + \\ &\qquad max_{a \in A_1(i)} \{t_a + \textstyle\sum_{j=1}^N p_j(a)u_j\} - \textstyle\sum_{j=1}^N p_j(a)y_j \\ &\geq \textstyle\sum_{j=1}^N p_j(a)x_j + t_a + \textstyle\sum_{j=1}^N p_j(a)(u_j - y_j) = t_a,\ 1 \leq i \leq m,\ a \in B(i). \end{aligned}$$

h.
$$\begin{aligned} z_i - z_{i+1} &= max_{a \in A_1(i)} \{t_a + \textstyle\sum_{j=1}^N p_j(a)u_j\} - max_{a \in A_1(i+1)} \{t_a + \textstyle\sum_{j=1}^N p_j(a)u_j\} \\ &\geq 0,\ (\text{because } A_1(i+1) \subseteq A_1(i)). \end{aligned}$$

For the proof of part (2), we can use $x \geq \phi$ and write
$$\begin{aligned} \textstyle\sum_{j=1}^N x_j + \sum_{j=1}^m w_j &= \textstyle\sum_{j=1}^N x_j + \sum_{j=1}^m max_{a \in A_1(j)} \sum_{k=1}^N p_k(a)x_k \\ &\geq \textstyle\sum_{j=1}^N \phi_j + \sum_{j=1}^m max_{a \in A_1(j)} \sum_{k=1}^N p_k(a)\phi_k. \end{aligned}$$ □

Lemma 8.27

Let (x, w, y, z) is a feasible solution of (8.105). Then,
(1) $w_i \geq \sum_{j=1}^N p_j(a)x_j$ for all $1 \leq i \leq m,\ a \in A_1(i)$.
(2) (x, y) is a feasible solution of (8.107) and $x \geq \phi$.

Proof

For the proof of part (1) take any $1 \leq i \leq m$ and $a \in A_1(i)$, say $a \in B(k)$ for some $i \leq k \leq m$. Then, we have $w_i \geq w_k \geq \sum_{j=1}^N p_j(a)x_j$. For the proof of part (2) we have to verify the constraints of (8.107). It is obvious that $\sum_{j=1}^N \{\delta_{ij} - p_{ij}(a)\}x_j \geq 0,\ 1 \leq i \leq N,\ a \in A_2(i)$ and $x_i + \sum_{j=1}^N \{\delta_{ij} - p_{ij}(a)\}y_j \geq r_i(a),\ 1 \leq i \leq N,\ a \in A_2(i)$.

Take any $1 \leq i \leq m$ and $a \in A_1(i)$, say $a \in B(k)$ for some $i \leq k \leq m$. Then,
$$\textstyle\sum_{j=1}^N \{\delta_{ij} - p_{ij}(a)\}x_j = x_i - \sum_{j=1}^N p_j(a)x_j \geq w_i - w_k \geq 0$$
and
$$\begin{aligned} x_i + \textstyle\sum_{j=1}^N \{\delta_{ij} - p_{ij}(a)\}y_j &= x_i + y_i - \textstyle\sum_{j=1}^N p_j(a)y_j \\ &\geq w_i + s_i + z_i - \textstyle\sum_{j=1}^N p_j(a)y_j \\ &\geq s_i + t_a + \textstyle\sum_{j=1}^N p_j(a)y_j - \sum_{j=1}^N p_j(a)y_j = r_i(a). \end{aligned}$$
Hence, (x, y) is a feasible solution of (8.107). Furthermore, $x \geq \phi$, because ϕ is the componentwise smallest superharmonic vector. □

Theorem 8.28

(1) The linear programs (8.105) and (8.106) have finite optimal solutions.
(2) If (x, w, y, z) is an optimal solution of (8.105), then $x = \phi$ and $w_j = max_{a \in A_1(j)} \sum_{k=1}^N p_k(a)\phi_k$.

Proof

We know that (8.107) has a finite optimal solution (x^*, u^*) with $x^* = \phi$. Hence, program (8.105) is feasible and bounded, implying that (8.105) and its dual (8.106) have finite optimal solutions. Consider an optimal solution $(x = \phi, u)$ of (8.107). Then, it follows from Lemma 8.26 part (2) that the corresponding solution $(x = \phi, w, y, z)$ is an optimal solution of (8.105), because

$$\textstyle\sum_{j=1}^N x_j + \sum_{j=1}^m w_j = \sum_{j=1}^N \phi_j + \sum_{j=1}^m max_{a \in A_1(j)} \sum_{k=1}^N p_k(a)\phi_k.$$

Let (x, w, y, z) is an optimal solution of (8.105). Then,

$$\textstyle\sum_{j=1}^N x_j + \sum_{j=1}^m w_j = \sum_{j=1}^N \phi_j + \sum_{j=1}^m max_{a \in A_1(j)} \sum_{k=1}^N p_k(a)\phi_k.$$

By Lemma 8.27, $x \geq \phi$ and $w_j \geq max_{a \in A_1(j)} \sum_{k=1}^N p_k(a)\phi_k,\ 1 \leq j \leq m$. Hence, $x = \phi$ and $w_j = max_{a \in A_1(j)} \sum_{k=1}^N p_k(a)\phi_k$. □

Lemma 8.28

For any pair of feasible solutions (x, w, y, z) and $(y, \mu, z, \sigma, x, \lambda, w, \rho)$ of (8.105) and (8.106), respectively, the following equalities hold:

$$\begin{aligned}
\Big\{ \sum_{j=1}^N \{\delta_{ij} - p_{ij}(a)\} x_j \Big\} \cdot x_i(a) &= 0,\ 1 \leq i \leq N,\ a \in A_2(i) && (8.108)\\
\{x_i - w_i\} \cdot \lambda_i &= 0,\ 1 \leq i \leq m && (8.109)\\
\Big\{ w_i - \sum_{j=1}^N p_j(a) x_j \Big\} \cdot w_i(a) &= 0,\ 1 \leq i \leq m,\ a \in B(i) && (8.110)\\
\{w_i - w_{i+1}\} \cdot \rho_i &= 0,\ 1 \leq i \leq m-1 && (8.111)
\end{aligned}$$

Proof

From the constraints of (8.105) and the nonnegativity of the variables of (8.106) it follows that the right hand sides of (8.108), (8.109), (8.110) and (8.111) are at least 0. If we add all left hand sides, we obtain

$$\begin{aligned}
&\textstyle\sum_{i=1}^N \sum_{a \in A_2(i)} \sum_{j=1}^N \{\delta_{ij} - p_{ij}(a)\} x_j \cdot x_i(a) + \sum_{i=1}^m \{x_i - w_i\} \cdot \lambda_i \ + \\
&\qquad\textstyle\sum_{i=1}^m \sum_{a \in B(i)} \{w_i - \sum_{j=1}^N p_j(a) x_j \cdot w_i(a) + \sum_{i=1}^{m-1} \{w_i - w_{i+1}\} \cdot \rho_i \ = \\
&\textstyle\sum_{j=1}^N x_j \cdot \Big\{ \sum_{i=1}^N \sum_{a \in A_2(i)} \{\delta_{ij} - p_{ij}(a)\} x_i(a) + \sum_{i=1}^m \delta_{ij} \lambda_i - \sum_{i=1}^m \sum_{a \in B(i)} p_j(a) w_i(a) \Big\} \ + \\
&\qquad\textstyle\sum_{i=1}^m w_i \Big\{ -\lambda_i + \sum_{a \in B(i)} w_i(a) + \rho_i - \rho_{i-1} \Big\} = 0,
\end{aligned}$$

since the terms between brackets are zero by the constraints of (8.106). Hence, the relations (8.108), (8.109), (8.110) and (8.111) are proven. □

From the complementary slackness property of linear programming is follows that optimal solutions (x, w, y, z) and $(y, \mu, z, \sigma, x, \lambda, w, \rho)$ of (8.105) and (8.106), respectively, satisfy

$$\begin{aligned}
\Big\{ \sum_{j=1}^N \{\delta_{ij} - p_{ij}(a)\} x_j \Big\} \cdot y_i(a) &= 0,\ 1 \leq i \leq N,\ a \in A_2(i) && (8.112)\\
\{x_i - w_i\} \cdot \mu_i &= 0,\ 1 \leq i \leq m && (8.113)
\end{aligned}$$

$$\begin{aligned}
\Big\{w_i - \sum_{j=1}^{N} p_j(a)x_j\Big\} \cdot z_i(a) &= 0,\ 1 \le i \le m,\ a \in B(i) && (8.114)\\
\{w_i - w_{i+1}\} \cdot \sigma_i &= 0,\ 1 \le i \le m-1 && (8.115)\\
\Big\{x_i + \sum_{j=1}^{N} \{\delta_{ij} - p_{ij}(a)\}y_j - r_i(a)\Big\} \cdot x_i(a) &= 0,\ 1 \le i \le N,\ a \in A_2(i) && (8.116)\\
\{y_i - z_i - s_i\} \cdot \lambda_i &= 0,\ 1 \le i \le m && (8.117)\\
\Big\{w_i + z_i - \sum_{j=1}^{N} p_j(a)y_j - t_a\Big\} \cdot w_i(a) &= 0,\ 1 \le i \le m,\ a \in B(i) && (8.118)\\
\{z_i - z_{i+1}\} \cdot \rho_i &= 0,\ 1 \le i \le m-1 && (8.119)
\end{aligned}$$

Lemma 8.29

Let (x,w,y,z) and $(y,\mu,z,\sigma,x,\lambda,w,\rho)$ be optimal solutions of (8.105) and (8.106), respectively. Let $m_i = min\{j \ge i \mid \sum_{a\in B(j)} w_j(a) > 0\}$ and $n_i = min\{j \ge i \mid \sum_{a\in B(j)}\{w_j(a) + z_j(a)\} > 0\}$. Define a policy $f^\infty \in C(D)$ such that

$$\begin{aligned}
x_i\big(f(i)\big) > 0 \quad &\mathit{if} \quad \sum_{a\in A_2(i)} x_i(a) > 0 && (8.120)\\
w_{m_i}\big(f(i)\big) > 0 \quad &\mathit{if} \quad \sum_{a\in A_2(i)} x_i(a) = 0 \ \mathit{and}\ \lambda_i > 0 && (8.121)\\
y_i\big(f(i)\big) > 0 \quad &\mathit{if} \quad \sum_{a\in A_2(i)} x_i(a) = \lambda_i = 0 \ \mathit{and}\ y_i\big(f(i)\big) > 0 && (8.122)\\
w_{n_i}\big(f(i)\big) > 0 \quad &\mathit{if} \quad \sum_{a\in A_2(i)} x_i(a) = \lambda_i = \sum_{a\in A_2(i)} y_i(a) = 0 \ \mathit{and} \sum_{a\in A_1(i)} w_{n_i}(a) > 0 && (8.123)\\
z_{n_i}\big(f(i)\big) > 0 \quad &\mathit{if} \quad \sum_{a\in A_2(i)} x_i(a) = \lambda_i = \sum_{a\in A_2(i)} y_i(a) = \sum_{a\in A_1(i)} w_{n_i}(a) = 0 && (8.124)
\end{aligned}$$

Then,
(1) f^∞ is well-defined.
(2) $x_i + \sum_{j=1}^N \{\delta_{ij} - p_{ij}(f)\}y_j = r_i(f),\ i \in S_+ = \{j \in S \mid \sum_{a\in A_2(i)} x_j(a) + \lambda_j > 0\}$.
(3) $\sum_{j=1}^N \{\delta_{ij} - p_{ij}(f)\}x_j = 0,\ i \in S$.

Proof

(1) Suppose that $\sum_{a\in A_2(i)} x_i(a) = 0,\ \lambda_i > 0$ and $\sum_{a\in B(j)} w_j(a) = 0$ for all $j \ge i$. Then, by the constraints of (8.106), we obtain $0 = \sum_{j=i}^m \{\rho_j - \rho_{j-1} - \lambda_j\} = -\rho_{i-1} - \sum_{j=i}^m \lambda_j \le \lambda_i < 0$, implying a contradiction. Hence, f^∞ is well-defined if $\sum_{a\in A_2(i)} x_i(a) = 0$ and $\lambda_i > 0$. Because $\sum_{a\in B(m)}\{w_m(a) + z_m(a)\} = 1 + \mu_m + \sigma_{m-1} > 0$, n_i is well-defined for all i. Therefore, the policy f^∞ is well-defined.

(2) Take any $i \in S_+$.
If $\sum_{a\in A_2(i)} x_i(a) > 0$, then by (8.116), $x_i + \sum_{j=1}^N \{\delta_{ij} - p_{ij}(f)\}y_j = r_i(f)$.
If $\sum_{a\in A_2(i)} x_i(a) = 0$, then $\lambda_i > 0$, and by (8.109) and (8.117), $x_i = w_i$ and $y_i = s_i + z_i$.
The definition of m_i implies that $\sum_{a\in B(j)} w_j(a) = 0,\ j = i, i{+}1, \dots, m_i{-}1$ and $w_{m_i}\big(f(i)\big) > 0$.
Hence, by the constraints of (8.106), we obtain $\rho_j = \rho_{j-1} + \lambda_j,\ j = i, i+1, \dots, m_i - 1$, i.e.

$\rho_j = \rho_{i-1} + \sum_{k=i}^{j} \geq \lambda_i > 0,\ j = i, i+1, \dots, m_i - 1$. Then, by (8.111) and (8.119), $w_j = w_{j+1},\ z_j = z_{j+1}$ for $j = i, i+1, \dots, m_i - 1$, implying $w_i = w_{m_i},\ z_i = z_{m_i}$. Since $w_{m_i}\big(f(i)\big) > 0$, by (8.118), we have $w_{m_i} + z_{m_i} - \sum_{j=1}^N p_j(f)y_j - t_{f(i)} = 0$. Hence,

$$\begin{array}{rcl} r_i(f) & = & s_i + t_{f(i)} = y_i - z_i + w_{m_i} + z_{m_i} - \sum_{j=1}^N p_j(f)y_j \\ & = & y_i - z_i + w_i + z_i - \sum_{j=1}^N p_j(f)y_j = w_i + \sum_{j=1}^N \{\delta_{ij} - p_{ij}(f)\}y_j \\ & = & x_i + \sum_{j=1}^N \{\delta_{ij} - p_{ij}(f)\}y_j. \end{array}$$

(3) If $f(i)$ is determined by (8.120) or (8.122), the result follows from (8.108) and (8.112), respectively. Suppose that $f(i)$ is determined by (8.121), then $\lambda_i > 0$ and $w_{m_i}\big(f(i)\big) > 0$. By (8.110), $w_{m_i} = \sum_{j=1}^N p_j(f)x_j$. In the proof of part (2) is shown that $\lambda_i > 0$ implies $w_{m_i} = w_i = x_i$. Therefore, $\sum_{j=1}^N \{\delta_{ij} - p_{ij}(f)\}x_j = w_{m_i} - \sum_{j=1}^N p_j(f)x_j = 0$.
Finally, suppose that $f(i)$ is determined by (8.123) and (8.124). Then, by (8.110) or (8.114), $w_{n_i} = \sum_{j=1}^N p_j(f)x_j$. Since $\sum_{a \in A_2(i)} x_i(a) = \sum_{a \in A_2(i)} y_i(a) = 0$, it follows from the constraints of (8.106) that $\mu_i > 0$, implying by (8.113), that $w_i = x_i$. Furthermore, from the definition of n_i, it follows that $\sum_{a \in B(j)}\{w_j(a) + z_j(a)\} = 0,\ j = i, i+1, \dots, n_i - 1$. Then, by the constraints of (8.106) it follows that $\sigma_j > 0,\ j = i, i+1, \dots, n_i - 1$, and consequently, by (8.115), $w_j = w_{j+1},\ j = i, i+1, \dots, n_i - 1$. Combining these results yields

$$\sum_{j=1}^N \{\delta_{ij} - p_{ij}(f)\}x_j = w_i - \sum_{j=1}^N p_j(f)x_j = w_{n_i} - \sum_{j=1}^N p_j(f)x_j = 0. \qquad \square$$

Lemma 8.30

Let $(y, \mu, z, \sigma, x, \lambda, w, \rho)$ be an optimal solution of (8.106), and let S_+ and policy f^∞ be defined as in Lemma 8.29. Then, S_+ is closed under $P(f)$, i.e. $p_{ij}(f) = 0$ for every $i \in S_+$ and $j \notin S_+$.

Proof

Suppose that $p_{ij}(f) > 0$ for some $i \in S_+$ and $j \notin S_+$. Since $i \in S_+$, the action $f(i)$ is defined either by (8.120) or by (8.121). Furthermore, since $j \notin S_+$, $\sum_{a \in A_2(j)} x_j(a) + \sum_{i=1}^m \delta_{ij}\lambda_i = 0$.
If $f(i)$ is defined by (8.120), then by the constraints of (8.106), we can write

$$\begin{array}{rcl} 0 & = & \sum_{a \in A_2(j)} x_j(a) + \sum_{i=1}^m \delta_{ij}\lambda_i = \sum_{i=1}^N \sum_{a \in A_2(i)} p_{ij}(a)x_i(a) + \sum_{i=1}^m \sum_{a \in B(i)} p_j(a)w_i(a) \\ & \geq & p_{ij}(f)x_i\big(f(i)\big) > 0, \end{array}$$

implying a contradiction.
If $f(i)$ is defined by (8.121), then we obtain

$$\begin{array}{rcl} 0 & = & \sum_{a \in A_2(j)} x_j(a) + \sum_{i=1}^m \delta_{ij}\lambda_i = \sum_{i=1}^N \sum_{a \in A_2(i)} p_{ij}(a)x_i(a) + \sum_{i=1}^m \sum_{a \in B(i)} p_j(a)w_i(a) \\ & \geq & p_j(f)w_{m_i}\big(f(i)\big) > 0, \end{array}$$

which also gives a contradiction. $\square$

Lemma 8.31

Let $(y, \mu, z, \sigma, x, \lambda, w, \rho)$ be an extreme optimal solution of (8.106), and let S_+ and policy f^∞ be defined as in Lemma 8.29. Then, the states of $S \backslash S_+$ are transient in the Markov chain with transition matrix $P(f)$.

Proof

Suppose that there exists a state $j \in S\backslash S_+$, which is recurrent under $P(f)$. Since S_+ is closed, there is an ergodic class $J \subseteq S\backslash S_+$. Let $J = J_1 \cup J_2 \cup J_3$, where J_1, J_2 and J_3 are the states of J in which $f(i)$ is determined by (8.122), (8.123) and (8.124), respectively.

We first show that $J_2 = \emptyset$. From the constraints of (8.106) it follows that for any $j \in S\backslash S_+$, we have $\sum_{i=1}^N \sum_{a\in A_2(i)} p_{ij}(a)x_i(a) + \sum_{i=1}^m \sum_{a\in B(i)} p_j(a)w_i(a) = 0$, i.e. every term in this equation is 0. Suppose that $i \in J_2$. Then, $w_{n_i}\big(f(i)\big) > 0$, and consequently $p_{ij}(f) = 0$ for every $j \in S\backslash S_+$. This contradicts that $S\backslash S_+$ contains an ergodic class J. For $i \in J_1$, we have $y_i\big(f(i)\big) > 0$. For $i \in J_3$, we have $\sum_{a\in A_2(i)}\{x_i(a) + y_i(a)\} = 0,\ \sum_{a\in B(j)}\{w_j(a) + z_j(a)\} = 0,\ i \le j \le n_i - 1$, and $z_{n_i}\big(f(i)\big) > 0$. From the constraints of (8.106) it follows that $\mu_i > 0$ and $\sigma_j > 0,\ i \le j \le n_i - 1$.

Next, consider the columns, denoted by $a^{(i)}, b^{(i)}, c^{(i)}$ and $d^{(j)}$, of the matrix of the constraints of (8.106) corresponding to the positive variables: $a^{(i)}$ corresponds to $y_i\big(f(i)\big),\ i \in J_1$, $b^{(i)}$ to $z_{n_i}\big(f(i)\big),\ i \in J_3$, $c^{(i)}$ to $\mu_i,\ i \in J_3$ and $d^{(j)}$ to $\sigma_j,\ j = i, i+1, \dots, n_i - 1$ for $i \in J_3$.

These columns have the following $2N + 2m$ components:

$$a_k^{(i)} = \begin{cases} \delta_{ik} - p_{ik}(f) & k = 1, 2, \dots, N \\ 0 & k = N+1, N+2, \dots, 2N+2m \end{cases} \tag{8.125}$$

$$b_k^{(i)} = \begin{cases} -p_k(f) & k = 1, 2, \dots, N \\ \delta_{n_i, k-N} & k = N+1, N+2, \dots, N+m \\ 0 & k = N+m+1, N+m+2, \dots, 2N+2m \end{cases} \tag{8.126}$$

$$c_k^{(i)} = \begin{cases} \delta_{ik} & k = 1, 2, \dots, N \\ -\delta_{i, k-N} & k = N+1, N+2, \dots, N+m \\ 0 & k = N+m+1, N+m+2, \dots, 2N+2m \end{cases} \tag{8.127}$$

$$d_k^{(j)} = \begin{cases} 0 & k = 1, 2, \dots, N \\ \delta_{j,k-N} - \delta_{j,k-N-1} & k = N+1, N+2, \dots, N+m \qquad j = i, i+1, \dots, n_i - 1 \\ 0 & k = N+m+1, N+m+2, \dots, 2N+2m \end{cases} \tag{8.128}$$

Since $(y, \mu, z, \sigma, x, \lambda, w, \rho)$ is an extreme optimal solution of (8.106) and the above columns correspond to strictly positive variables, these columns are linearly independent. Let p be the number of elements in $\cup_{i\in J_3}\{i, i+1, \dots, n_i - 1\}$. Then, there are $q = |J_1| + 2 \cdot |J_3| + p$ independent vectors. Notice that all columns have zeros in the last $N+m$ components. Since an ergodic class is closed, the components $k \notin J,\ 1 \le k \le N$ of the vectors are zero, because for $i \in J$, $\delta_{ik} = p_{ik}(f) = 0$. Furthermore, we observe that the components $N+k,\ 1 \le k \le m$ are zero, except $\{n_i \mid i \in J_3\}$ (in $b^{(i)}$), $\{i \mid i \in J_3\}$ (in $c^{(i)}$) and $\cup_{i\in J_3}\{i, i+1, \dots, n_i - 1\}$ (in $d^{(j)}$).

Hence, there are at most $|J| + |J_3| + p = |J_1| + 2 \cdot |J_3| + p = q$ components (of the $2N = 2m$ components) which can be positive.

Consider the contracted vectors, obtained from $a^{(i)}, b^{(i)}, c^{(i)}$ and $d^{(j)}$, by deleting the components that are 0 in all vectors. Then, the q contracted vectors have at most q components and are still independent, i.e. they have exactly q components and the corresponding matrix is nonsingular. On the other hand, the components of each vector add up to 0, which contradicts the nonsingularity. This completes the proof of this lemma. □

Theorem 8.29

The policy f^∞, defined in Lemma 8.29, is an average optimal policy.

Proof

Let (x, w, y, z) be optimal solution of (8.105). Then, by Theorem 8.28, $x = \phi$, and, by Lemma 8.29 part (3), $\phi = P(f)\phi$. Consequently, $\phi = P^*(f)\phi$, where $P^*(f)$ is the stationary matrix of $P(f)$. Since the states of $S\backslash S_+$ are transient in the Markov chain with transition matrix $P(f)$, see Lemma 8.31, we have $p^*_{ik}(f) = 0$ for every $i \in S,\ k \notin S_+$. Hence, we can write by Lemma 8.29 part (2),

$$\begin{aligned}\phi_i(f^\infty) &= \{P^*(f)r(f)\}_i = \textstyle\sum_{k\in S} p^*_{ik}(f)r_k(f) = \sum_{k\in S_+} p^*_{ik}(f)r_k(f)\\ &= \textstyle\sum_{k\in S_+} p^*_{ik}(f)\{\phi_k + \sum_{j=1}^N [\delta_{kj} - p_{kj}(f)]y_j\}\\ &= \{P^*(f)\phi\}_i + \{P^*(f)[I-P(f)]y\}_i = \{P^*(f)\phi\}_i = \phi_i,\ i \in S,\end{aligned}$$

i.e. f^∞ is an average optimal policy. □

8.7.6 Examples (part 2)

1. Replacement problem

Consider the following replacement problem:

State space $S = \{0, 1, \ldots, N\}$; action sets $A(i) = \{r, 1, \ldots, N\}$, where r is the action 'retain' (keep the item for at least one more time period) and action $1 \le a \le N$ means that we replace the item for another item of state a. The automobile replacement problem of Section 8.7.2 is an example of a replacement problem.

Consider the following relevant data:

s_a = the cost of buying an item of state a, $1 \le a \le N$;
u_i = the trade-in value of an item of state i, $1 \le i \le N$;
t_a = the expected cost of operating an item of state a for one time period, $1 \le a \le N$;
p_{ij} = the probability that an item of state i is transfered to state j in one time period, $1 \le i, j \le N$.

The standard MDP for this model has the rewards and transition probabilities:

$$r_i(a) = \begin{cases} -t_i & i = 1, 2, \ldots, N;\ a = r \\ u_i - (s_a + t_a) & i = 1, 2, \ldots, N;\ a = 1, 2, \ldots, N \end{cases}$$

$$p_{ij}(a) = \begin{cases} p_{ij} & i, j = 1, 2, \ldots, N;\ a = r \\ p_{aj} & i, j = 1, 2, \ldots, N;\ a = 1, 2, \ldots, N \end{cases}$$

The standard linear program (8.107) has $2 \cdot \sum_{i\in S}\sum_{a\in A(i)} = 2N(N+1)$ constraints and $2N$ variables. For the reduced formulation as separable problem, we obtain:

$$S_1 = S;\ S_2 = \emptyset;\ A_1(i) = \{1, 2, \ldots, N\},\ 1 \le i \le N;\ A_2(i) = \{r\},\ 1 \le i \le N.$$

Notice that $m = N,\ B(i) = \emptyset,\ 1 \le i \le N-1$ and $B(N) = \{1, 2, \dots, N\}$. From the constraints of (8.106) it follows that $\rho_j = \sum_{k=1}^{j} \lambda_k$ and $\sigma_j = \sum_{k=1}^{j} \mu_k + j$ for $j = 1, 2, \dots, N-1$. Hence, the variables ρ_j and σ_j can be deleted from (8.106) for all j. Then, program (8.106) can be formulated as

$$max \sum_{i=1}^{N} -t_i x_i + \sum_{i=1}^{N} u_i \lambda_i - \sum_{a=1}^{N} (s_a - t_a) w_a \tag{8.129}$$

subject to the constraints

$$\begin{array}{rcrcrcrcl}
\sum_{i=1}^{N}\{\delta_{ij} - p_{ij}\} y_i & + & \mu_j & - & \sum_{a=1}^{N} p_j(a) z_a & + & x_j & = & 1,\ 1 \le j \le N \\
 & - & \sum_{j=1}^{N} \mu_j & + & \sum_{a=1}^{N} w_a & + & \sum_{a=1}^{N} z_a & = & N \\
\sum_{i=1}^{N}\{\delta_{ij} - p_{ij}\} x_i & + & \lambda_j & - & \sum_{a=1}^{N} p_j(a) w_a & & & = & 0,\ 1 \le j \le N \\
 & - & \sum_{j=1}^{N} \lambda_j & + & \sum_{a=1}^{N} w_a & & & = & 0
\end{array}$$

$x_i,\ y_i,\ z_a,\ w_a,\ \lambda_i,\ \mu_i \ge 0$ for all i and a.

The relation $-\sum_{j=1}^{N} \lambda_j + \sum_{a=1}^{N} w_a = 0$ can be deleted, because this is implied by the previous set of equalities, namely: $\sum_{j=1}^{N} \big\{ \sum_{i=1}^{N}\{\delta_{ij} - p_{ij}\} x_i + \lambda_j - \sum_{a=1}^{N} p_j(a) w_a \big\} = \sum_{j=1}^{N} \lambda_j - \sum_{a=1}^{N} w_a$. Hence, the linear program becomes

$$max \sum_{i=1}^{N} -t_i x_i + \sum_{i=1}^{N} u_i \lambda_i - \sum_{a=1}^{N} (s_a - t_a) w_a \tag{8.130}$$

subject to the constraints

$$\begin{array}{rcrcrcrcl}
\sum_{i=1}^{N}\{\delta_{ij} - p_{ij}\} y_i & + & \mu_j & - & \sum_{a=1}^{N} p_j(a) z_a & + & x_j & = & 1,\ 1 \le j \le N \\
 & - & \sum_{j=1}^{N} \mu_j & + & \sum_{a=1}^{N} w_a & + & \sum_{a=1}^{N} z_a & = & N \\
\sum_{i=1}^{N}\{\delta_{ij} - p_{ij}\} x_i & + & \lambda_j & - & \sum_{a=1}^{N} p_j(a) w_a & & & = & 0,\ 1 \le j \le N
\end{array}$$

$x_i,\ y_i,\ z_a,\ w_a,\ \lambda_i,\ \mu_i \ge 0$ for all i and a.

This linear program has $6N$ variables and $2N+1$ constraints. Let $(y, \mu, z, x, \lambda, w)$ be an extreme optimal solution of program 8.130. An optimal action in state $i \in S$, as defined in Lemma 8.29, becomes for this replacement problem:

If $x_i > 0$ or $x_i = \lambda_i = 0$ and $y_i > 0$: take the action r (retain).

If $x_i = 0$ and $\lambda_i > 0$ or $x_i = \lambda_i = y_i = 0$ and $\sum_{a=1}^{N} w_a > 0$: take an action a with $w_a > 0$.

If $x_i = \lambda_i = y_i = \sum_{a=1}^{N} w_a = 0$: take an action a with $z_a > 0$.

2. Inventory problem

Consider the inventory problem of Section 8.7.2. We have seen that there are $N+1$ states and that the total number of decisions is equal to $\frac{1}{2}(N+1)(N+2)$. Therefore, the standard LP formulation (8.107) has $(N+1)(N+2)$ constraints and $2(N+1)$ variables. In the reduced formulation of this separable problem, we have

$S_1 = \{0, 1, \dots, N-1\};\ S_2 = \{N\}$.

$A_1(i) = \{i+1, i+2, \dots, N\}\ \rightarrow\ B(i) = \{i+1\},\ 0 \le i \le N-1;\ A_2(i) = \{i\},\ 0 \le i \le N.$

The dual linear program (8.106) for this inventory problem becomes

$$max \sum_{i=0}^{N} -h_i x_i + \sum_{i=0}^{N-1} (-K+ci)\lambda_i + \sum_{i=0}^{N-1} \{-c(i+1) - h_{i+1}\} w_{i+1} \tag{8.131}$$

subject to the constraints

$$\begin{array}{rcrcrcrcl}
y_0 - \sum_{i=0}^N \big\{\sum_{k\geq i} p_k\big\} y_i & + & \mu_0 & - & \sum_{i=0}^{N-1} \big\{\sum_{k\geq i+1} p_k\big\} z_{i+1} & + & x_0 & = & 1 \\
y_j - \sum_{i=j}^N p_{i-j} y_i & + & \mu_j & - & \sum_{i=j}^N p_{i-j} z_i & + & x_j & = & 1,\ 1 \leq j \leq N-1 \\
(1-p_0) y_N & & & - & p_0 z_N & + & x_N & = & 1 \\
\sigma_j - \sigma_{j-1} & - & \mu_j & + & w_{j+1} & + & z_{j+1} & = & 1,\ 0 \leq j \leq N-1 \\
x_0 - \sum_{i=0}^N \big\{\sum_{k\geq i} p_k\big\} x_i & + & \lambda_0 & - & \sum_{i=0}^{N-1} \big\{\sum_{k\geq i+1} p_k\big\} w_{i+1} & & & = & 0 \\
x_j - \sum_{i=j}^N p_{i-j} x_i & + & \lambda_j & - & \sum_{i=j}^N p_{i-j} w_i & & & = & 0,\ 1 \leq j \leq N-1 \\
(1-p_0) x_N & & & - & p_0 w_N & & & = & 0 \\
\rho_j - \rho_{j-1} & - & \lambda_j & + & w_{j+1} & & & = & 0,\ 0 \leq j \leq N-1
\end{array}$$

$\rho_{-1} = \rho_{N-1} = \sigma_{-1} = \sigma_{N-1} = 0;\ x_i,\ y_i,\ z_i,\ w_i,\ \lambda_i,\ \mu_i,\ \rho_i,\ \sigma_i \geq 0$ for all i.

This linear program has $8N$ variables and $2(2N+1)$ constraints. Let $(y,\mu,z,\sigma,x,\lambda,w,\rho)$ be an extreme optimal solution of program (8.131). An optimal action in state $i \in S$, as defined in Lemma 8.29, where $m_i = min\{j \geq i+1 \mid w_j > 0\}$ and $n_i = min\{j \geq i+1 \mid w_j + z_j > 0\}$, becomes for this inventory problem:

If $x_i > 0$: no order.
If $x_i = 0$ and $\lambda_i > 0$: order $m_i - 1$ items.
If $x_i = \lambda_i = 0$ and $y_i > 0$: no order.
If $x_i = \lambda_i = y_i = 0$: order $n_i - 1$ items.

Remark
In the case that the optimal policy is an (s,S)-policy, the underlying Markov chain is unichained. Then, a linear program with $4N$ variables and $2N+2$ constraints suffices (see Exercise 8.11).

8.8 Bibliographic notes

The general replacement model of Section 8.1.1 is strongly related to a paper by Gal ([100]), in which paper the method of policy iteration was considered. With the same approach the average reward case for an irreducible MDP can be treated. The replacement model with increasing deterioration, that has a control-limit optimal policy, appears in Derman ([68]). The skip to the right model with failure is due to Kallenberg ([152]). The separable replacement model was discussed in Sobel ([277]). It may also be viewed as a special case of the SER-SIT game (see ([214]).

The surveillance-maintenance-replacement problem is taken from Derman ([69]). The problem of optimal repair allocation in a series system appears in [159]. We follow a proof contributed by

Weber (personal communication). Section 8.2.3 is taken from a paper by Katehakis and Derman (see [160]).

The section production and inventory control is taken from Denardo ([63]): Chapter 5 (for our sections 8.3.1 and 8.3.2), Chapter 6 (for our section 8.3.3) and Chapter 7 (for our section 8.3.4). The production control problem has received considerable attention in the literature. Dynamic programming formulations for the concave-cost case were due initially to Wagner and Whithin ([317]), and, independently, to Manne ([192]). Extensions to backlogging are due to Zangwill ([339], [340]) and Manne and Veinott ([194]). Work on single-critical-number policies include Bellman, Glicksberg and Gross ([18]), Karlin ([156]) and Veinott ([307], [309]). The notion that the ordering cost can be absorbed into the holding cost may be implicit in Beckmann ([16]). Scarf ([251]) analyzed an inventory model with set-up costs, backlogging and convex operating costs. He introduced K-convexity and used it to show that an (s, S) policy is optimal. Veinott ([309]) analysed this model with quasi-convex costs. Porteus [221]) introduced K-quasi-convexity.

The queueing control models are taken from Sennott ([262]) with the exception of the admission control of an $M/M/1$ queue model which can be found in Puterman ([227]). Lippman ([182]) applies uniformization to characterize optimal policies in several exponential queueing control systems. Serfozo ([264]) formalizes this approach in the context of countable-state continuous-time models. Bertsekas ([22]) and Walrand ([318]) contain many interesting applications of the use of uniformization in queueing control models.

The material of section 8.5 is taken from Ross [239] (sections 8.5.1, 8.5.5, 8.5.6 and 8.5.7) and Walrand [318] (sections 8.5.2, 8.5.3 and 8.5.4). The work of section 8.5.1 appeared in [72]. The classical μc-rule is given by Cox and Smith ([47]). The proof of Theorem 8.11 for the optimality of the μc-rule is due to Buyukkoc, Varaiya and Walrand ([36]). The optimality of the threshold policy in section 8.5.3 was shown by Lin and Kumar ([180]). The first to prove the optimality of the SQP was Winston ([335]), in 1977. The proof of Theorem 8.14 is a variant of a proof given by Ephremides, Varaiya and Walrand ([78]). Theorem 8.15 is from Ross ([239]). Theorem 8.16 is due to Bruno, Downey and Frederickson [35]) and Theorem 8.17 to Glazebrook ([106]). The alternative proof of the optimality of the $LEFT$ policy is from Pinedo and Weiss ([216]). For the work on stochastic minimizing the makespan or the time until one of the processors is idle we refer to Weber ([319]). The tandem queue model can be found in Weiss [321]), which presents a nice survey of multiserver scheduling models. Another survey of such models is given by Pinedo and Schrage ([215]). For dynamic programming and stochastic scheduling we refer also to Koole ([171]). In [131] Hordijk and Koole have introduced a new type of arrival processes, called a Markov decision arrival process. This arrival process can be controlled and allows for an indirect dependence on the number of customers in the queues. As a special case, they showed the optimality of $LEPT$ and the μc-rule in the last node of a controlled tandem network for various cost structures.

The fundamental contribution on the multi-armed bandit problem, the optimality of the index policy, is due to Gittins ([104] and [105]). The presentation of this result as formulated in the proofs of Lemma 8.20 and Theorem 8.23 is taken from Ross ([239]). Other proofs of this theorem are given by Whittle ([332] and [333]) who introduced the term Gittins index in honor to Gittins, Katehakis and Veinott ([162]), Weber ([320]), Tsitsiklis ([291] and [292]) and Weiss ([322]), who in fact established an index theorem for the more general branching bandits model. Bertsimas and Nino-Mora ([23]) provided a proof for many other classes of multi-armed bandit problems. The parametric linear programming method with complexity $\mathcal{O}(N^3)$ was proposed by Kallenberg ([149]). He improved an order $\mathcal{O}(N^4)$ method of Chen and Katehakis ([38]), who have introduced the linear program (8.80). In [205] Nino-Mora presents a fast-pivoting algorithm that computes the N Gittins indices in the discounted and undiscounted case by performing $\frac{2}{3}N^3 + \mathcal{O}(N^2)$ arithmetic operations. The interpretation as restart-in-k problem is given by Katehakis and Veinott ([162]) and the method of the largest-remaining-index rule is due to Varaiya, Walrand and Buyukkoc ([306]). The bisection/successive approximation method was proposed by Ben-Israel and Fläm ([20]). Other contributions on this subject were made by Katehakis and Rothblum ([161], who considered the problem under alternative optimality criteria, namely sensitive discount optimality, average reward optimality and average overtaking optimality, and Glazebrook and Owen ([107]).

De Ghellinck and Eppen ([52]) examined separable MDPs with the discounted rewards as optimality criterion. They streamlined the linear program of D'Epenoux ([67]). Denardo introduced in [57] the notion of zero-time transitions. Discounted and averaging versions (for the unichain case) are then shown to yield policy iteration and linear programming formulations. In the discounted case, the linear program is identical to that of De Ghellinck and Eppen. Kallenberg ([150]) has shown that for the average reward criterion also in the multichain case a simpler linear program can be used to solve the original problem. The automobile replacement problem was first considered by Howard ([134]). The totally separable problem of Exercise 8.12 is a special case of a stochastic game studied in [214] and [277].

8.9 Exercises

Exercise 8.1

a. Show for the case in which i and j are the two smallest indices of $C_0(x)$ the nonpositivity of the inductive hypothesis $H(m+1)$ (see (8.21)), i.e. assuming $H(m)$ show that $\mu_j\{T^{m+1}(1_j,x) - T^{m+1}(0_j,x)\} \leq 0$.

b. Show for the case $f_*(x) < i < j$ of the inductive hypothesis $H(m+1)$.

Exercise 8.2

Consider the following production and inventory control model without backlogging:

$T = 5;\ D_1 = 1,\ D_2 = 4,\ D_3 = 5,\ D_4 = 3,\ D_5 = 1.$

$$c_t(a) = \begin{cases} 0 & \text{if } a = 0 \\ 7 & \text{if } a \geq 1 \end{cases},\ 1 \leq t \leq T;\ h_t(i) = i,\ i \geq 0,\ 1 \leq t \leq T.$$

Compute an optimal production plan.

Exercise 8.3

Consider the production and inventory control model of Exercise 8.2 with backlogging.

Let the shortage cost functions be $h_t(i) = -2i,\ i \leq 0,\ 1 \leq t \leq T$.

Compute an optimal production plan.

Exercise 8.4

Consider the following inventory control model with a single-critical-number optimal policy.

Let $s = 2;\ k = 3;\ R = 5;\ \alpha = 0.9;\ T = 4$.

The demand is as follows:

t	$p_t(0)$	$p_t(1)$	$p_t(2)$	$p_t(3)$	$p_t(4)$	$p_t(5)$
1	0.2	0.3	0.3	0.1	0.1	0.0
2	0.0	0.1	0.2	0.3	0.3	0.1
3	0.0	0.1	0.2	0.3	0.3	0.1
4	0.2	0.3	0.3	0.1	0.1	0.0

Compute an optimal single-critical-number policy.

Exercise 8.5

Consider the following inventory control model with fixed ordering cost, in which backlogging is not allowed.

$T = 4;\ \alpha = 0.9;\ k_t = 3,\ 1 \leq t \leq 4;\ K_t = 1,\ 1 \leq t \leq 5;\ R_t = 5,\ 1 \leq t \leq 4;\ e(i) = 2i,\ i \geq 0.$

$h_t(a) = (1 - \alpha)k_t a,\ 1 \leq t \leq 4.$

The demand is as follows:

t	$p_t(0)$	$p_t(1)$	$p_t(2)$	$p_t(3)$	$p_t(4)$	$p_t(5)$
1	0.2	0.3	0.3	0.1	0.1	0.0
2	0.0	0.1	0.2	0.3	0.3	0.1
3	0.0	0.1	0.2	0.3	0.3	0.1
4	0.2	0.3	0.3	0.1	0.1	0.0

Compute an optimal policy.

Exercise 8.6

Show that $X \geq_{st} Y$ implies $X^+ \geq_{st} Y^+$ and $X^- \leq_{st} Y^-$.

Exercise 8.7

Show that $\mathbb{E}\{C_{1,2}\} \leq \mathbb{E}\{C_{2,1}\} \ \Leftrightarrow\ \lambda_1 - \mu_1 \geq \lambda_2 - \mu_2$, where $\mathbb{E}\{C_{1,2}\}$ is defined in section 8.5.7.

Exercise 8.8

Consider the model of Example 8.2 with N sequences of nonnegative numbers. Let for any k the sequence $\{x_n^k \mid n = 1, 2, \dots\}$ be nonincreasing in n. Show that the policy that chooses the sequence with the largest next reward (such policy is called a *myopic policy*) is optimal.

Exercise 8.9

Consider the model of Example 8.2 with $\alpha = 0.5$ and with as the three sequences:
$x^1 = \{3, 2, 4, 0, 0, \dots\},\ x^2 = \{2, 3, 2, 0, 0, \dots\}$ and $x^3 = \{2, 1, 4, 0, 0, \dots\}$.
What is the optimal order of the sequences to maximize $\sum_{t=1}^{\infty} \alpha^{t-1} R_t$.

Exercise 8.10

Consider a multi-armed bandit problem with three projects and with discount factor $\alpha = \frac{1}{2}$.
The data of the projects are:

Project 1: $S_1 = \{1, 2, 3, 4\};\ r^1_1 = 4,\ r^1_2 = 2,\ r^1_3 = 4,\ r^1_4 = 0.$
$p^1_{12} = p^1_{23} = p^1_{34} = p^1_{44} = 1$ (the other transition probabilities are 0).

Project 2: $S_2 = \{1, 2, 3, 4\};\ r^2_1 = 2,\ r^2_2 = 6,\ r^2_3 = 2,\ r^2_4 = 0.$
$p^2_{12} = p^2_{23} = p^2_{34} = p^2_{44} = 1$ (the other transition probabilities are 0).

Project 3: $S_3 = \{1, 2, 3, 4\};\ r^3_1 = 3,\ r^3_2 = 3,\ r^3_3 = 4,\ r^4_4 = 0.$
$p^3_{12} = p^3_{23} = p^3_{34} = p^3_{44} = 1$ (the other transition probabilities are 0).

a. Determine the 12 Gittins indices by the interpretation with stopping times.

b. Determine the 12 Gittins indices by the parametric linear programming method.

c. Determine the 12 Gittins indices by the restart-in-k method.

d. Determine the 12 Gittins indices by the largest-remaining-index method.

e. If the starting state is $(1, 1, 1)$, i.e. in each project we start in state 1, what will be the sequence of the projects in an optimal policy?

Exercise 8.11

Consider the inventory model as described in Section 8.7.6. Show that in the unichain case a linear program with $4N$ variables and $2N + 2$ constraints suffices.

Exercise 8.12

Consider the *totally separable problem*, i.e. an MDP with $S = \{1, 2, \dots, N\};\ A(i) = \{1, 2, \dots, M\}$, $i \in S:\ r_i(a) = s_i + t_a,\ (i, a) \in S \times A$ and $p_{ij}(a) = p_j(a),\ (i, a) \in S \times A,\ j \in S$.
Let the action a_* be defined by $t_{a_*} + \sum_{j=1}^{N} p_{a_* j} s_j = max_{1 \le a \le M}\{t_a + \sum_{j=1}^{N} p_{aj} s_j\}$.
Show that the policy f_*^∞ with $f(i) = a_*,\ i \in S$, is an average optimal policy for this totally separable problem.

Chapter 9

Other topics

9.1 Complexity results

9.1.1 Complexity theory

We present a summary of *computational complexity*. Most of the readers will have some intuitive idea about what is meant by the running time of an algorithm. Although this intuition will be sufficient to understand the substance of the matter, in some cases it is important to formalize this intuition. This is particular the case when we deal with concepts like $\mathcal{P}$, $\mathcal{NP}$, $\mathcal{NC}$, $\mathcal{NP}$-complete and $\mathcal{P}$-complete. Most complexity results are framed in terms of decision problems, i.e. problems that require a yes/no response for each input.

Below we make some of the notions more precise. We shall not elaborate all technical details. For a more background information we refer to the books of Aho, Hopcroft and Ullman ([1]), Garey and Johnson ([102]), Papadimitriou ([210]) and Schrijver ([253]).

An *algorithm* is a finite list of instructions to solve any instance of some problem for which the algorithm is developed. The classical mathematical formalization of an algorithm is the *Turing machine*. However, for our goal an informal description will be sufficient. We measure the *running time* of an algorithm as the total number of *elementary steps* to solve the problem. Examples of elementary steps are: variable assignments, instructions such as **for**, **repeat**, **while**, **if**, **then**, **else**, **go to**, and simple arithmetic operations like addition, subtraction, multiplication and division.

We are interested in a good upper bound of the number of elementary steps as a function of the *input size*. The input to an algorithm usually consists of a list of numbers. If these numbers are integers, we can encode them in binary representation with $\lfloor log_2 n \rfloor + 1$ bits for storing an integer n. The input size of an instance is the total number of bits needed for the binary representation.

We are special interested in the rates of growth asymptotically, i.e. when the input size tends to infinity. We now will describe the symbols $\mathcal{O}$, Ω and Θ, which are used in the context of comparing the rates of growth.

Let $f(n)$ and $g(n)$ be two nonnegative functions of n. We say that $f(n) = \mathcal{O}\big((g(n)\big)$ if there exist c and n_0, such that $f(n) \leq c \cdot g(n)$ for all $n \geq n_0$. This means, informally, that f does not grow at a faster rate than g. For example, $5n^3 + 3n = \mathcal{O}(n^3)$. We say that $f(n) = \Omega\big((g(n)\big)$ if there exist c and n_0, such that $f(n) \geq c \cdot g(n)$ for all $n \geq n_0$. This means, informally, that f does not grow at a slower rate than g. We say that $f(n) = \Theta\big((g(n)\big)$ if there exist c_1, c_2 and n_0, such that $c_1 \cdot g(n) \leq f(n) \leq c_2 \cdot g(n)$ for all $n \geq n_0$. This means, informally, that f and g have the same rate of growth. A function f has a *polynomial growth* if $f(n) = \mathcal{O}(n^p)$ for some $p \in \mathbb{N}$; f has an *exponential growth* if $f(n) = \Omega(c^n)$ for some $c > 1$.

The most commonly used measure of time complexity, the *worst-case time complexity*, of an algorithm A is the maximum amount of time taken on any input of size n; we denote this quantity as $t_A(n)$. Time complexities are classified by the nature of the function $t_A(n)$. For instance, an algorithm with $t_A(n) = \mathcal{O}(n)$ is called a *linear time* algorithm, and an algorithm with $t_A(n) = \Omega(2^n)$ is said to be an *exponential time* algorithm. An algorithm A is said to be of *polynomial time* if its running time is upper bounded by a polynomial expression in the size of the input for the algorithm, i.e. $t_A(n) = \mathcal{O}(n^k)$ for some constant k. An algorithm is said to run in *polylogarithmic time* if $t_A(n) = \mathcal{O}\big((log\, n)^k\big)$, for some constant k.

In some contexts, especially in optimization, one differentiates between strongly polynomial and (weakly) polynomial algorithms. For these concepts not only the running time is of importance, but also the space used by the algorithm.

Consider a problem with an input of size n, given by n bits. A polynomial time algorithm is allowed to perform elementary steps with these bits where the total number of steps is bounded by a polynomial in n. What often is neglected in the analysis of an algorithm is the potentially growing size of numbers. Note that the well-known algorithm to determine the decimal digits of 2^k - by using as input size of k the $n = log_2\, k$ bits (roughly speaking) and iteratively n times squaring 2 - needs a linear number of elementary steps, namely n multiplications. However, the space to represent the output 2^k is proportional to $log_2\, 2^k = k = 2^n$, which is exponential in the input size n.

An algorithm is said to be of *strongly polynomial* if:

(1) the number of elementary steps in the arithmetic model of computation is bounded by a polynomial in the size of the input;

(2) the space used by the algorithm is bounded by a polynomial in the size of the input.

Any algorithm with these two properties can be converted to a polynomial time algorithm by replacing the arithmetic operations by suitable algorithms for performing the arithmetic operations on a Turing machine. If the second requirement above is omitted, then this is not true anymore. An algorithm which runs in polynomial time but which is not strongly polynomial is said to run in (weakly) polynomial time. The linear programming problem has a polynomial time algorithm (see [157]), but a strongly polynomial algorithm for linear programming is not known.

We now present the formal descriptions of the complexity classes $\mathcal{P}$, $\mathcal{NP}$, $\mathcal{NC}$ and the concepts of reducibility, $\mathcal{NP}$-complete and $\mathcal{P}$-complete problems.

The class $\mathcal{P}$

We say that a decision problem belongs to the class $\mathcal{P}$ if there is an algorithm A and a number k such that for each instance I of the problem the algorithm A will produce a solution in time $\mathcal{O}(n^k)$, where n is the input size, i.e. the number of bits in the input string that represents I. The set $\mathcal{P}$ is also called the set of *easy* decision problems; other synonyms are *tractable, efficient* or *fast*.

The class $\mathcal{NP}$

A decision problem belongs to the class $\mathcal{NP}$ if there is an algorithm A that has the following property: associated with each instance I for which the answers is *yes*, there exists a *certificate* $C(I)$ such that the algorithm A recognizes in polynomial time that I is a yes-problem. Hence, $\mathcal{NP}$ is the class of decision problems for which it is easy to check the correctness of a problem with a yes answer with the aid of special information, the certificate. We are not asking to find a solution, but only to verify that an alleged yes-solution really is correct.

The class $\mathcal{NP}$ includes many combinatorial optimization problems. As an example, consider the *graph-coloring* problem: given a graph and a positive integer k, is it possible to color the vertices with k colors such that neighbors have different colors. This problem is not known to be in $\mathcal{P}$. It is however in $\mathcal{NP}$, and here is the algorithm and the certificate. Let G be a graph and k a positive integer such that G is k-colorable. The certificate of G is a list of the colors that get assigned to each vertex of G in some proper k-coloring of the vertices of G. We have to check that for each edge of G the two endpoints have a different color. This can be done in $\mathcal{O}(m)$ time, where m is the number of edges.

The class $\mathcal{NC}$

A decision problem belongs to the class $\mathcal{NC}$ if there is an algorithm A for a parallel computer with p processors and numbers k and l such that p is polynomial in the input size n, i.e. $p = \mathcal{O}(n^k)$, and for each instance I of the problem the algorithm A will produce a solution in polylogaritmic time, i.e. in $\mathcal{O}\big((log\, n)^l\big)$.

Obviously, by multiplying the polylogarithmic time and the polynomial number of processors, all problems in $\mathcal{NC}$ are in $\mathcal{P}$. The great enigma for parallel computation, analogous to the $\mathcal{P} = \mathcal{NP}$ question for sequential computation, is whether $\mathcal{P} = \mathcal{NC}$. That is, while $\mathcal{P} = \mathcal{NP}$ asks whether there are problems in $\mathcal{NP}$ that are inherently nonpolynomial, $\mathcal{P} = \mathcal{NC}$ asks whether there are problems in $\mathcal{P}$ that are inherently sequential.

Polynomial reducibility

Let P and Q be two decision problems. We say that P can be polynomial reduced to Q if every instance I of P in polynomial time can be converted to an instance J of Q in such a way that I and J have the same answer ('yes' or 'no'). Hence, P is not harder to solve than Q, i.e. if there exists a polynomial algorithm for Q, there also exists a polynomial algorithm for P.

$\mathcal{NP}$-complete
A problem Q is $\mathcal{NP}$-complete if: (1) $Q \in \mathcal{NP}$; (2) Any $P \in \mathcal{NP}$ is polynomial reducible to Q. So, $\mathcal{NP}$-complete problems are the hardest problems in $\mathcal{NP}$. For many decision problems it is shown that they are $\mathcal{NP}$-complete. No polynomial algorithm is known for solving an $\mathcal{NP}$-complete problem. The open question whether $\mathcal{P} = \mathcal{NP}$ basically boils down to whether any $\mathcal{NP}$-complete problem can be solved in polynomial time.

Obviously, $\mathcal{P} \subseteq \mathcal{NP}$. $\mathcal{P}$ consists of the problems for which it is 'easy' to find a solution and $\mathcal{NP}$ consists of the problems for which it is 'easy' to check a solution of a yes-problem. One generally sets the problems in $\mathcal{P}$ against the $\mathcal{NP}$-complete problems, although there is still no proof that these two concepts really are distinct.[1] For almost every combinatorial optimization problem one has been able either to prove that it is solvable in polynomial time, or that it is $\mathcal{NP}$-complete. But theoretically it is still a possibility that these two concepts are just the same.

Example 9.1 *The satisfiability problem (SAT)*
A *Boolean variable* x is a variable that can assume only the values *true* and *false*. Boolean variables x and y can be combined by the logical connectives *and* (denoted by $x \wedge y$) and *or* (denoted by $x \vee y$); furthermore, for each Boolean variable x we have the *negation* (denoted by $\overline{x}$). One can form Boolean formulas in much the same way that real variables can be combined by arithmetic operations to form algebraic expressions. For example $\overline{x_3} \wedge x_2 \wedge (x_1 \vee \overline{x_2} \vee x_3)$ is a Boolean formula.

Given a value $t(x)$ for each variable x, we can evaluate a Boolean formula, just as we would an algebraic expression. For example, the Boolean formula $\overline{x_3} \wedge x_2 \wedge (x_1 \vee \overline{x_2} \vee x_3)$, evaluated at the set of values $t(x_1) = true, t(x_2) = true$ and $t(x_3) = false$, gives the value *true*.

So, the formula above can be made true by some appropriate assignment: such Boolean formula is called *satisfiable*. Not all Boolean formulas are satisfiable; there are some that cannot be made true by any assignment, essentially because they are encodings of a contradiction.

For example, consider $(x_1 \vee x_2 \vee x_3) \wedge (x_1 \vee \overline{x_2}) \wedge (x_2 \vee \overline{x_3}) \wedge (x_3 \vee \overline{x_1}) \wedge (\overline{x_1} \vee \overline{x_2} \vee \overline{x_3})$. For satisfiability, all subformulas within parentheses (called *clauses*) that contain *literals* (that is, variables or negations) must be *true*. The first clause says that - in order to have satisfiability - at least one of the variables must be *true*. The next three clauses force all variables to be the same, so all variables must be *true*; but then, the last clause is *false*. Hence, the formula is unsatisfiable. The satisfiability problem (SAT) is as follows: *Given m clauses $C_1, C_2, \ldots, C_m$ involving the variables $x_1, x_2, \ldots, x_n$, is the formula $C_1 \wedge C_2 \wedge \cdots \wedge C_m$ satisfiable?*
Of course, SAT can be solved by trying all possible assignments to see if one satisfies the formula. This is not an efficient algorithm, however, since there are 2^n assignments, so the algorithm has an exponential running time.

The theory of $\mathcal{NP}$-completeness started with Cooks paper [46], which contains the proof that SAT is $\mathcal{NP}$-complete. The wealth of the consequences of Cooks work and its close relationship to combinatorial optimization were made clear by Karp (see [158]).

[1] The $\mathcal{P} = \mathcal{NP}$ question is one of the seven Millennium Problems and solving it brings one million dollar from the Clay Mathematics Institute. See also www.claymath.org/millennium/P_vs_NP/

$\mathcal{P}$-complete

The notion of $\mathcal{P}$-complete decision problems is useful in the analysis of both: (1) which problems are difficult to parallelize effectively; (2) which problems are difficult to solve in polynomial space. Formally, a decision problem Q is $\mathcal{P}$-complete if: (1) $Q \in P$; (2) any $P \in \mathcal{P}$ is reducible to Q by using an *appropriate reduction.*

The specific type of reduction used varies and may affect the exact set of $\mathcal{P}$-complete problems. If we use $\mathcal{NC}$-reductions, that is, reductions which can operate in polylogarithmic time on a parallel computer with a polynomial number of processors, then - under the unproven assumption that $\mathcal{NC} \neq \mathcal{P}$ - all $\mathcal{P}$-complete problems lie outside $\mathcal{NC}$.

So, $\mathcal{P}$-complete problems cannot be effectively parallelized, because otherwise all problems in $\mathcal{P}$ can be solved in polylogarithmic time on a parallel computer with a polynomial number of processors, which contradicts $\mathcal{NC} \neq \mathcal{P}$. $\mathcal{P}$-complete problems are the hardest problems in $\mathcal{P}$. No polylogarithmic algorithm on a parallel computer with a polynomial number of processors is known for solving a $\mathcal{P}$-complete problem. The open question whether $\mathcal{P} = \mathcal{NC}$ basically boils down to whether any $\mathcal{P}$-complete problem can be solved in polylogarithmic time on a parallel computer with a polynomial number of processors.

The basic $\mathcal{P}$-complete problem under $\mathcal{NC}$-reductions is the *circuit value problem* (*CVP*). It plays the same role in parallel complexity as the *SAT* problem in $\mathcal{NP}$-complete problems.

A *circuit* C is a finite sequence of triples, i.e. $C = \{(a_i, b_i, c_i), 1 \leq i \leq N\}$. For each $1 \leq i \leq N$, a_i is one of the Boolean operations **false**, **true**, **and** and **or**; b_i, c_i are nonnegative integers smaller than i. If a_i is either **false** or **true**, then the triple is called an *input* and $b_i = c_i = 0$. If a_i is either **and** or **or**, then the triple is called a *gate* and $b_i, c_i \geq 1$. The value of a triple is defined recursively as follows. First, the value of the input $(\mathbf{true}, 0, 0)$ is **true**, and the value of the input $(\mathbf{false}, 0, 0)$ is false. The value of a gate (a_i, b_i, c_i) is the Boolean operation a_i applied to the value of the b_i-th and c_i-th triples. The value of the circuit C is the value of the last gate. Finally, the CVP is the following problem: given a circuit C, is its value **true**? Ladner has shown (see [175]) that the CVP problem is $\mathcal{P}$-complete under $\mathcal{NC}$-reductions.

Example 9.2

Let $C = \{(\mathbf{false}, 0, 0), (\mathbf{true}, 0, 0), (\mathbf{and}, 1, 2), (\mathbf{or}, 3, 1)\}$. The values of the triples 1 and 2, the inputs, are **false** and **true**, respectively. Triple 3, an **and**-gate, has value $\mathbf{false} \wedge \mathbf{true} = \mathbf{false}$; triple 4, an **or**-gate, has value $\mathbf{false} \vee \mathbf{false} = \mathbf{false}$. Hence, the value of the circuit is **false**.

$\mathcal{P}$-hard and $\mathcal{NP}$-hard problems

A problem P is $\mathcal{P}$-hard if all problem in $\mathcal{P}$ can be reduced to P with respect to an appropriate reduction. A $\mathcal{P}$-hard problem P is as hard as any problem in $\mathcal{P}$. To show that P is $\mathcal{P}$-hard it is sufficient to show that some $\mathcal{P}$-complete problem can be reduced to P. For a $\mathcal{P}$-hard problem it is not necessary to belong to $\mathcal{P}$.

A problem P is $\mathcal{NP}$-hard if all problem in $\mathcal{NP}$ can be polynomially reduced to P. An $\mathcal{NP}$-hard problem P is as hard as any problem in $\mathcal{NP}$. To show that P is $\mathcal{NP}$-hard it is sufficient to show that some $\mathcal{NP}$-complete problem can be polynomially reduced to P. For an $\mathcal{NP}$-hard problem it is not necessary to belong to $\mathcal{NP}$.

9.1.2 MDPs are $\mathcal{P}$-complete

Consider an MDP with one of the following optimality criteria:
(1) total expected reward over a finite horizon;
(2) total expected discounted reward over an infinite horizon;
(3) average expected reward over an infinite horizon.
We shall show that in all three cases the MDP is $\mathcal{P}$-complete.

(1) *Total expected reward over a finite horizon with T stages*
We first show that this MDP belongs to the class $\mathcal{P}$. Let $M := \sum_{i=1}^{N} |A(i)|$. As input size of this problem we take $n := max(M, T)$. Then, for each $1 \leq t \leq T$, step 2 of Algorithm 2.1 has complexity $\mathcal{O}(M \cdot N)$, and the overall complexity of Algorithm 2.1 is $\mathcal{O}(T \cdot M \cdot N) = \mathcal{O}(n^3)$, which is polynomial in the size n of the problem.

Next, we show that any CVP can be reduced to an MDP in polynomial time. This MDP will have total reward 0 or negative for any given starting state, so is may be considered as a decision problem, and the value of the CVP is **true** if and only if the optimal total reward is 0 for starting state N, the size of the circuit.

Let $C = \{(a_i, b_i, c_i),\ 1 \leq i \leq N\}$ be a circuit. We construct the following MDP. The state space $S := \{0, 1, \dots, N\}$, where state i corresponds to the triple (a_i, b_i, c_i) for $i = 1, 2, \dots, N$. State 0 is an absorbing state without rewards, i.e. there is only one action which has probability 1 to stay in state 0 and the reward is 0. If state i corresponds to an input $(a_i, 0, 0)$, then there is also only one action in this state which has probability 1 to transit to state 0 and the reward is 0 if $a_i =$ **true** and -1 if $a_i =$ **false**. If a_i is an **or** gate, then there are two actions in state i, each with reward 0 and deterministic transitions; the first action has a transition to state b_i and the second action has a transition to state c_i. So, in such state one can decide whether the next state is b_i or c_i. If a_i is an **and** gate, there is only one action in state i, which has reward 0 and transitions to the states b_i and c_i, each with probability 1. As initial state we take the last state N and the time horizon is also N, the size of the circuit.

The following observations are obvious: (1) this construction is a polynomial in time complexity; (2) the total reward is either 0 or negative; (3) at each stage with probability 1 the system moves to a smaller state, so we end after at most $T = N$ steps in the absorbing state 0.

We claim that the optimal expected total reward from starting state N is 0 if and only if the value of C is **true**. Suppose that the optimal expected reward, starting in state N, is 0. Then, it follows that there are decisions so that the state with negative rewards, i.e. inputs $(\textbf{false}, 0, 0)$ are not reached. Thus these decisions are choices of a **true** gate among b_i, c_i for each **or** gate of the circuit, so that its overall value is **true**. Conversely, if the value is **true**, there must be a way to choose an input gate for each **or** gate so that the **false** inputs are not reached, or, equivalently, the states with negative rewards are not visited. Hence, the optimal expected total reward from starting state N is 0.

Notes

1. The above proof shows that even the stationary finite horizon problem is $\mathcal{P}$-hard.
2. As input size n we have chosen $n = max(M, T)$. If we don't allow that the input size is dependent on the horizon T, it is not known whether the stationary finite horizon problem is in $\mathcal{P}$, because of the following difficulty. We could be given a stationary process with horizon $T = 2^N$ and the input size could be of size $\mathcal{O}(N)$. Still, the dynamic programming algorithm for this problem would take time proportional to $N \cdot T$, and thus exponential in the input size.
3. In the nonstationary case, the input must specify the transition probabilities and rewards for each $1 \le t \le T$, and so the input size is at least T.

(2) *Total expected discounted reward over an infinite horizon*

Since an MDP with optimality criterion the total expected discounted reward over an infinite horizon can be formulated as a linear programming problem, and since the linear program problem belongs to $\mathcal{P}$, MDP also belongs to $\mathcal{P}$. We can also show that CVP can be reduced to an MDP in polynomial time. It is easy to verify that essentially the same construction as for the finite horizon works also for an MDP with discounted rewards.

(3) *Average expected reward over an infinite horizon*

An MDP with optimality criterion the average expected reward over an infinite horizon can be formulated as a linear programming problem, so this MDP also belongs to $\mathcal{P}$. We can also show that CVP can be reduced to an MDP in polynomial time, but we need a modification of our construction: we don't need state 0, but we need transitions from the states corresponding to the inputs back to the initial state.

Below we present two examples with each of these three criteria: in the first example the value of the circuit is **false** and in the second example the value of the circuit is **true**.

Example 9.3

Let $C = \{(\textbf{false}, 0, 0), (\textbf{true}, 0, 0), (\textbf{and}, 1, 2), (\textbf{or}, 3, 1)\}$. This is the same circuit as in Example 9.2 and the value of the circuit is **false**.

The data of the MDP for the criterion of total reward over a finite horizon are:

$T = 4;\ S = \{0, 1, 2, 3, 4\};\ A\{0\} = A\{1\} = A\{2\} = A\{3\} = \{1\},\ A\{4\} = \{1, 2\};$

$p_{00}(1) = 1,\ p_{10}(1) = 1,\ p_{20}(1) = 1,\ p_{31}(1) = p_{32}(1) = \frac{1}{2},\ p_{43}(1) = 1,\ p_{41}(2) = 1;$

$r_0(1) = 0,\ r_1(1) = -1,\ r_2(1) = 0,\ r_3(1) = 0,\ r_4(1) = 0,\ r_4(2) = 0.$

The value of this MDP, starting in state 4, is $-\frac{1}{2} \neq 0$.

The data of the MDP for the criterion of total discounted reward over an infinite horizon are the same (without T). Let $\alpha = \frac{1}{2}$. The value of this MDP, starting in state 4, is $-\frac{1}{8} \neq 0$.

The data of the MDP for the criterion of average reward over an infinite horizon are:.

$S = \{1, 2, 3, 4\}$; $A\{1\} = A\{2\} = A\{3\} = \{1\}$, $A\{4\} = \{1, 2\}$;
$p_{14}(1) = 1$, $p_{24}(1) = 1$, $p_{31}(1) = p_{32}(1) = \frac{1}{2}$, $p_{43}(1) = 1$, $p_{41}(2) = 1$;
$r_1(1) = -1$, $r_2(1) = 0$, $r_3(1) = 0$, $r_4(1) = 0$, $r_4(2) = 0$.
The value of this MDP, starting in state 4, is $-\frac{1}{6} \neq 0$.

Example 9.4
Let $C = \{(\textbf{false}, 0, 0), (\textbf{true}, 0, 0), (\textbf{and}, 1, 2), (\textbf{or}, 3, 2)\}$. The values of the triple 1 and 2, the inputs, are **false** and **true**, respectively. Triple 3, an **and**-gate, has value **false** $\wedge$ **true** $=$ **false**; triple 4, an **or**-gate, has value **false** $\vee$ **true** $=$ **true**. Hence, the value of the circuit is **true**.
The data of the MDP for the criterion of total reward over a finite horizon are:
$T = 4:$ $S = \{0, 1, 2, 3, 4\}$; $A\{0\} = A\{1\} = A\{2\} = A\{3\} = \{1\}$, $A\{4\} = \{1, 2\}$;
$p_{00}(1) = 1$, $p_{10}(1) = 1$, $p_{20}(1) = 1$, $p_{31}(1) = p_{32}(1) = \frac{1}{2}$, $p_{43}(1) = 1$, $p_{42}(2) = 1$;
$r_0(1) = 0$, $r_1(1) = -1$, $r_2(1) = 0$, $r_3(1) = 0$, $r_4(1) = 0$, $r_4(2) = 0$.
The value of this MDP, starting in state 4, is 0.
The data of the MDP for the criterion of total discounted reward over an infinite horizon are the same (without T). Let $\alpha = \frac{1}{2}$. The value of this MDP, starting in state 4, is 0.
The data of the MDP for the criterion of average reward over an infinite horizon are:.
$S = \{1, 2, 3, 4\}$; $A\{1\} = A\{2\} = A\{3\} = \{1\}$, $A\{4\} = \{1, 2\}$;
$p_{14}(1) = 1$, $p_{24}(1) = 1$, $p_{31}(1) = p_{32}(1) = \frac{1}{2}$, $p_{43}(1) = 1$, $p_{42}(2) = 1$;
$r_1(1) = -1$, $r_2(1) = 0$, $r_3(1) = 0$, $r_4(1) = 0$, $r_4(2) = 0$.
The value of this MDP, starting in state 4, is 0.

9.1.3 DMDPs are in $\mathcal{NC}$

An MDP is said to be *deterministic* if each action uniquely determines the next state of the process. In other words, the probability distribution associated with each action assigns probability 1 to one of the states. Deterministic Markov decision problems are denoted as DMDPs. A DMDP can be conveniently represented as a network, i.e. a directed graph with weights on the arcs. The vertices of the graph correspond to the states of the DMDP and the arcs correspond to the actions. If in state i action $a \in A(i)$ is chosen which has a transition with probability 1 to state j, then the graph has an arc from state i to state j with as weight the cost $c_i(a)$ (we assume in this section that we have costs instead of rewards, which can be assumed without loss of generality by taking $c_i(a) := -r_i(a)$ for all $(i, a) \in S \times A$). For DMDPs, we may assume that in every state i there is at most one transition to any state j (if there are more, we always will take the transition with the lowest cost). Hence, $|A(i)| \leq N$ for all states i.

We shall show below that the deterministic cases of the finite horizon (stationary and nonstationary), discounted, and average reward MDPs are in $\mathcal{NC}$. Our approach is to look at these problems as variants of the graph-theoretic shortest path problem. We consider the decision problem with a fixed starting state i_1. The particular variants of the problem are then equivalent to certain variants of the shortest path problem. If a DMDP with a given starting state belongs to $\mathcal{NC}$, then also the decision problem for all possible starting states belongs to $\mathcal{NC}$, because in stead of p processors we use Np processors (p processors for each stating state), which is also polynomial in the input size if p is polynomial in the input size .

The nonstationary finite horizon problem

The parallel algorithms that we describe employ a technique to yield fast parallel (or space efficient) algorithms known as *path doubling* (see [250] and [271]). The idea is, once we have computed all optimal paths between any two states, where each path starts at time t_1 and ends at time t_2, and similarly between t_2 and t_3, to compute in one step all optimal paths between t_1 and t_3. We shall see below that we can think of this as 'multiplying' two $N \times N$ matrices $A(t_1, t_2)$ and $A(t_2, t_3)$, where the (i, j)th entry of $A(t_1, t_2)$ is the cost of the optimal path from state i to state j between the times t_1 and t_2.

Note that $\{A(t_1, t_3)\}_{ij} = min_k \{A(t_1, t_2\}_{ik} + \{A(t_2, t_3)\}_{kj}$. For the usual matrix multiplication $A \cdot B$, where A and B have elements a_{ij} and b_{ij}, respectively, we have $\{A \cdot B\}_{ij} = \sum_k a_{ik} \cdot b_{kj}$. Therefore, for $t_1 < t_2 < t_3$, the matrix $A(t_1, t_3)$ can be obtained by 'multiplication' of the matrices $A(t_1, t_2)$ and $A(t_2, t_3)$, in which the multiplication of $a_{ik} \cdot b_{kj}$ is replaced by addition $\{A(t_1, t_2\}_{ik} + \{A(t_2, t_3)\}_{kj}$, and the addition $(\sum_k)$ is replaced by the operation of taking the minimum (min_k).

For each (i, j) we compute $min_k \{A(t_1, t_2\}_{ik} + \{A(t_2, t_3)\}_{kj}$ in the following way, using N parallel processors:

(1) Compute independently on N parallel processors the elements $\{A(t_1, t_2\}_{ik} + \{A(t_2, t_3)\}_{kj}$ for $k = 1, 2, \ldots, N$. For each processor this computation needs $\mathcal{O}(1)$ time complexity.

(2) Pair up the N elements and compute the pairwise minima, each minimum of two elements can be computed in $\mathcal{O}(1)$ time, to reduce the size of the array by a half and repeat it $log_2 N$ times to find the minimum of the entire array.

So, the computation of each $\{A(t_1, t_3)\}_{ij}$ can be done with N parallel processors in $log_2 N$ parallel steps of $\mathcal{O}(1)$ time and consequently the computation of the whole matrix $\{A(t_1, t_3)\}$ can be done with N^3 parallel processors in $log_2 N$ time complexity.

This approach immediately suggests an $\mathcal{NC}$ parallel algorithm for the finite horizon nonstationary problem. We start with the $T - 1$ matrices $A(t, t + 1)$ for $t = 1, 2, \ldots, T - 1$, where $\{A(t, t + 1)\}_{ij}$ equals the cost of the decision leading at time t from state i to state j, if such decision exists at time t, and equal to ∞ otherwise. To solve the decision problem, we must compute $A(1, T)$ and this can be done by 'multiplying' these matrices in $\lceil log_2 T \rceil$ stages.

For example, for $T = 9$, in stage 1 we start with $A(1, 2)$, $A(2, 3)$, $A(3, 4)$, $A(4, 5)$, $A(5, 6)$, $A(6, 7)$, $A(7, 8)$ and $A(8, 9)$. In stage 2, we compute $A(1, 3) := A(1, 2) \cdot A(2, 3)$, $A(3, 5) := A(3, 4) \cdot A(4, 5)$, $A(5, 7) := A(5, 6) \cdot A(6, 7)$ and $A(7, 9) := A(7, 8) \cdot A(8, 9)$. Then, in stage 3, we obtain the matrices $A(1, 5) := A(1, 3) \cdot A(3, 5)$ and $A(5, 9) := A(5, 7) \cdot A(7, 9)$. Finally, in stage 4, we end with the matrix $A(1, 9) := A(1, 5) \cdot A(5, 9)$.

Each multiplication needs $log_2 N$ parallel steps when N^3 processors are used. Hence, when $T \cdot N^3$ processors are used, each stage of the $log_2 T$ stages can be computed in $log_2 N$ parallel steps.

Therefore, the parallel time of the computation of $A(1,T)$ is $(log_2 T) \cdot (log_2 N)$. Since the size of the input $n = T \cdot N^2$, we have that p, the number of processors, and T^*, the parallel time, satisfy $p = \mathcal{O}(n^2)$ and $T^* = \mathcal{O}\big((log_2 n)^2\big)$, respectively. Hence, this is an $\mathcal{NC}$ parallel algorithm and the following theorem holds.

Theorem 9.1
The nonstationary finite horizon deterministic Markov decision problem is in $\mathcal{NC}$.

Note
Notice that this technique does not solve the nonstationary finite horizon problem, whose input is of size $n = max(N^2, T)$. We have to attack this problem by a more sophisticated technique, which is explained later.

The infinite horizon undiscounted DMDP
As we shall show in section 9.5.2, the infinite horizon average cost DMDP is equivalent to finding the cycle in the corresponding network that is reachable from the starting state i_1, and has the minimum mean-weight cycle. To make sure that we do not consider cycles that are not reachable from the starting state i_1, we first determine the nodes that are reachable from i_1. This can be done in $log_2 N$ parallel time (see [271]). The cycle with the minimum mean-weight can be found by computing, in parallel, for each $k = 1, 2, \ldots, N$, the shortest cycle of length k, and accompanying the results, each divided by k. To compute the shortest cycle of length k, we essentially have to compute the kth power of matrix A, whose (i,j)th entry is equal to the cost of the decision leading from state i to state j, if such decision exists, and equal to ∞ otherwise. This can be done with $k \cdot N^2$ processors in $(log_2 k) \cdot (log_2 N)$ parallel steps. If we use N^4 processors, the total time of this approach is of order $(log_2 N)^2$. Hence, this is an $\mathcal{NC}$ algorithm and the next theorem holds.

Theorem 9.2
The infinite horizon undiscounted deterministic Markov decision problem is in $\mathcal{NC}$.

The infinite horizon discounted DMDP
Define a *sigma in* i_1 in a directed graph as a path P from i_1 until the first repetition of a node. In other words, a sigma in i_1 is a path P of the form $P = \{i_1, i_2, \ldots, i_k, j_1, j_2, \ldots, j_l, j_1\}$, where all nodes $i_1, i_2, \ldots, i_k, j_1, j_2, \ldots, j_l$ are distinct. The discounted cost $c(P)$ of the sigma $P = \{i_1, i_2, \ldots, i_k, j_1, j_2, \ldots, j_l, j_1\}$ satisfies

$$c(P) = \sum_{t=1}^{k} \alpha^{t-1} c(i_t, i_{t+1}) + \frac{\alpha^k}{1-\alpha^l} \cdot \sum_{t=1}^{l} \alpha^{t-1} c(j_t, j_{t+1(modl)}).$$

The discounted cost of a sigma coincides with the discounted cost of an infinite path that follows the sigma and repeats the cycle forever. It follows from the fact that the discounted DMDP has a stationary optimal policy, that the cost of an optimal policy is the optimal discounted cost of a sigma in the corresponding directed graph.

We can compute the optimal sigma as follows. First, we compute the shortest discounted cost of a path of length k for $k = 1, 2, \ldots, N$ among any pair of nodes by multiplying the matrices $A, \alpha A, \alpha^2 A, \alpha^{k-1} A$, where A is the matrix whose (i, j)th entry is equal to the cost of the decision leading from state i to state j, if such decision exists, and equal to ∞ otherwise. This can be done in at most $(log_2 N)^2$ steps by using N^4 processors.

Let $B_1, B_2, \ldots, B_N$ be the resulting products. Notice that the (i, j)th entry of B_k is the discounted length of the shortest path from i to j with exactly k arcs. Once this is done, we compute, for any starting node i_1 and each node j, for each $k, l = 1, 2, \ldots, N$ the value $\lambda + \frac{\alpha^k}{1-\alpha^l} \cdot \mu$, where λ is the (i_1, j)th entry of B_k and μ is the (j, j)th entry of B_l. Of all these values, we pick the minimum. If we use N^4 processors, the total time needs order $(log_2 N)^2$ parallel steps. Therefore, there is an $\mathcal{NC}$ algorithm for the infinite horizon discounted DMDP, which result is stated in the following theorem.

Theorem 9.3
The infinite horizon discounted deterministic Markov decision problem is in $\mathcal{NC}$.

The stationary finite horizon problem
The nonstationary finite horizon problem has input size $n = T \cdot N^2$, but the input size of the stationary finite horizon problem is $max\{N^2, T\}$, so the statement that the stationary finite horizon problem is in $\mathcal{NC}$ because it is a special case of the nonstationary finite horizon problem, which is in $\mathcal{NC}$ by Theorem 9.1, is not correct.

The stationary DMDP over a finite horizon with horizon T is equivalent to finding the shortest path with T arcs in the corresponding graph. If $T \leq N^2$, the previous technique for the nonstationary finite horizon problem has parallel time $(log_2 T) \cdot (log_2 N) = \mathcal{O}\big((log_2 N)^2\big)$ when we use $p = \mathcal{O}(N^4)$ processors. Hence, the problem belongs to $\mathcal{NC}$. Therefore, we now assume that $N^2 < T$, implying that the input size is determined by T, which can be encoded with $log_2 T$ bits; so, we have to find an algorithm that should run in a number of parallel steps that is polynomial in $log\,(log_2 T)$.

Without loss of generality, assume that the arc lengths are such that no ties in the length of paths are possible (this can be achieved by perturbing the lengths). Consider the shortest path, starting in i_1, with T arcs. Since $T > N^2$ there are many repetitions of nodes on this path. Consider the first such repetition, that is, the first time the path forms a sigma, and remove the cycle from the path. Then, consider the first repetition in the resulting sequence. Continuing in this way, we can decompose the path into a simple path plus several simple cycles.

We first need to show that we can assume that only one simple cycle is repeated at least N times, namely the one that has the shortest average length of arcs. In proof, consider two simple cycles with m_1 and m_2 arcs, repeated n_1 and n_2 times, respectively, with $n_1, n_2 \geq N$. Assume the cycle with m_1 arcs has the smallest average of arc length. Since $n_1, n_2 \geq N \geq m_1, m_2$, we can repeat the first cycle m_2 times more, and repeat the second cycle m_1 times less to obtain another path with T arcs of smaller length. Thus, only one cycle is repeated at least N times. Furthermore, since we have no ties, for each $1 \leq k \leq N$ at most one cycle with k arcs is repeated.

Therefore, the shortest path of T arcs has the following structure: it consists of a path with $l < N^3$ arcs (this is the simple path plus at most N repetitions of one cycle of k arcs, for each $1 \leq k \leq N$) plus a simple cycle repeated many times to fill the required number of arcs. Therefore, for each value of $l < N^3$, each node j and each $1 \leq k \leq N$ that divides $T - l$ we do, in parallel, the following: we compute the shortest path P of length l from i_1 through j, the shortest cycle C of length k through j, and the total cost of this path, i.e. $c(P) + \frac{T-l}{k} \cdot C(C)$. Of the resulting constructions, we pick the cheapest. By this approach, we obtain the following result.

Theorem 9.4
The stationary finite horizon deterministic Markov decision problem is in $\mathcal{NC}$.

9.1.4 For discounted MDPs, the policy iteration and linear programming method are strongly polynomial

In this section we show that, for discounted MDPs, the policy iteration and the linear programming algorithm both are strongly polynomial algorithms. We have already mentioned (see the remark at the end of section 3.4) that one iteration of policy iteration is strongly polynomial. It is also easy to verify that one iteration of the the linear programming algorithm is also strongly polynomial, with $\mathcal{O}\big(N \cdot (M-N)\big)$ arithmetic operations. Hence, we have to show that the number of iterations in policy iteration and linear programming has strongly polynomial complexity.

Let k_i be the number of actions available in state i, i.e. $|A(i)| = k_i$ for $i = 1, 2, \dots, N$. For notation convenience, let $A(1) = \{1, 2, \dots, k_1\},\ A(2) = \{k_1 + 1, k_1 + 2, \dots, k_1 + k_2\}, \dots,$ $A(N) = \{\sum_{i=1}^{N-1} k_i + 1, \sum_{i=1}^{N-1} k_i + 2, \dots, \sum_{i=1}^{N} k_i = M\}$.

The linear program (in fact this is the dual program (3.32) with $\beta_j = 1$ for all j) can be written as

$$max\,\{r^Tx \mid Ax = e;\ x \geq 0\}, \tag{9.1}$$

where A is an $N \times M$-matrix, r, x are M-vectors and e is a vector of all ones. The matrix A has the special form $A = E - \alpha P$, with $E_{ij} := \begin{cases} 1 & \text{if } j \in A(i) \\ 0 & \text{otherwise} \end{cases}$ and $P_{ij} := \begin{cases} p_{li}(j) & \text{if } j \in A(l) \\ 0 & \text{otherwise} \end{cases}$.
Since each action j belongs to exactly one $A(i)$, we have $e^TE = e^T$ (e is the vector of all ones, where its dimension depends on the context; the first e has dimension N and the e in the right-hand side has dimension M), and because $\sum_i p_{li}(j) = 1$ for every l and j, we also have $e^TP = e^T$, and consequently, E^T and P^T are stochastic matrices, and $e^TA = (1 - \alpha)e^T$.

The dual program of (9.1) is

$$min\,\{e^Tv \mid A^Tv \geq r\},\ \text{ or equivalently, }\ min\,\{e^Tv \mid A^Tv - s = r;\ s \geq 0\}. \tag{9.2}$$

We have seen that the dual program (9.2) has the unique solution v^α, the value vector, and consequently $s^* := A^Tv^\alpha - r$ is the unique optimal value for s.

A deterministic policy f^∞ chooses exactly one action $f(i) \in A(i)$ for each state i. Obviously, we have a total of $\prod_{i=1}^N k_i$ different policies. For any deterministic policy f^∞, let A_f, E_f and P_f be the $N \times N$-submatrix of A, E and P, respectively, consisting of the columns corresponding to $f(i),\ i \in S$. Then, $E_f = I$ (the identity matrix) and $A_f = I - \alpha P_f$.

It is well known that A_f is nonsingular, has a nonnegative inverse and is a feasible basis for the linear program (9.1). Let x^f be the basic feasible solution for policy f^∞, i.e. x^f is the unique solution of the linear system $A_f x = e$. The corresponding basic solution v^f of the dual program (9.2) is the unique solution of the linear system $A_f^T v = r^f$. The basic pair of solutions (x^f, v^f) are optimal solutions of (9.1) and (9.2) if and only if $A^T v^f \geq r$.

Example 3.1 (continued)

For this example, we have $N = 3;\ M = 9;\ A(1) = \{1, 2, 3\},\ A(2) = \{4, 5, 6\},\ A(3) = \{7, 8, 9\}$.

$$E = \begin{pmatrix} 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \end{pmatrix} \text{ and } P = \begin{pmatrix} 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 \end{pmatrix}.$$

$r = (1, 2, 3, 6, 4, 5, 8, 9, 7)$.

Hence, the dual pair of linear programs (9.1) and (9.2) are in this example:

$max\,\{x_1 + 2x_2 + 3x_3 + 6x_4 + 4x_5 + 5x_6 + 8x_7 + 9x_8 + 7x_9\}$

subject to

$$\begin{array}{rrrrrrrrrl} \frac{1}{2}x_1 & + \; x_2 & + \; x_3 & - \frac{1}{2}x_4 & & & - \frac{1}{2}x_7 & & & = 1 \\ & - \frac{1}{2}x_2 & & + \; x_4 & + \frac{1}{2}x_5 & + \; x_6 & & - \frac{1}{2}x_8 & & = 1 \\ & & - \frac{1}{2}x_3 & & & - \frac{1}{2}x_6 & + \; x_7 & + \; x_8 & + \frac{1}{2}x_9 & = 1 \end{array}$$

$x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8, x_9 \geq 0$

and

$min\,\{v_1 + v_2 + v_3\}$

subject to

$$\begin{array}{rrrl rrrl rrrl} \frac{1}{2}v_1 & & & \geq 1; & -\frac{1}{2}v_1 & + \; v_2 & & \geq 6; & -\frac{1}{2}v_1 & & + \; v_3 & \geq 8 \\ v_1 & - \frac{1}{2}v_2 & & \geq 2; & & \frac{1}{2}v_2 & & \geq 4; & & -\frac{1}{2}v_2 & + \; v_3 & \geq 9 \\ v_1 & & - \frac{1}{2}v_3 & \geq 3; & & v_2 & - \frac{1}{2}v_3 & \geq 5; & & & \frac{1}{2}v_3 & \geq 7 \end{array}$$

Take f^∞ with $f(1) = 2,\ f(2) = 4,\ f(3) = 9$. The system $A_f x = e$ is: $\left\{ \begin{array}{rrrl} x_2 & - \frac{1}{2}x_4 & & = 1 \\ -\frac{1}{2}x_2 & + \; x_4 & & = 1 \\ & & \frac{1}{2}x_9 & = 1 \end{array} \right.$

with solution $x_2 = 2,\ x_4 = 2,\ x_9 = 2$. The dual system $A^T v = r_f$ is: $\left\{ \begin{array}{rrrl} v_1 & - \frac{1}{2}v_2 & & = 2 \\ -\frac{1}{2}v_1 & + \; v_2 & & = 6 \\ & & \frac{1}{2}v_3 & = 7 \end{array} \right.$

with solution $v_1 = \frac{20}{3},\ v_2 = \frac{28}{3},\ v_3 = 14$. The pair (x^f, v^f) is not optimal, because v^f is not feasible for the dual program.

The optimality conditions for the dual pair of linear programs (9.1) and (9.2) are:

$$\begin{aligned} Ax &= e \\ A^Tv - s &= r \\ x_js_j &= 0,\ j = 1, 2, \dots, M \\ x, s &\geq 0 \end{aligned}$$

Lemma 9.1

(1) $e^Tx = \frac{N}{1-\alpha}$ for every feasible solution x of (9.1).

(2) $1 \leq x^f_j \leq \frac{N}{1-\alpha}$ for every basic variable x^f_j, $1 \leq j \leq N$, for every basic solution x^f of (9.1).

Proof

(1) Let x be a feasible solution x of (9.1). Then, we have $N = e^Te = e^TAx = (1-\alpha)e^Tx$, implying $e^Tx = \frac{N}{1-\alpha}$.

(2) Let x^f be a basic solution of (9.1). Then, by part (1), $x^f_j \leq \frac{N}{1-\alpha}$, $1 \leq j \leq N$. Furthermore, $e = A_fx^f = (I - \alpha P_f)x^f$, implying $x^f = (I - \alpha P_f)^{-1}e = \sum_{t=1}^{\infty} \alpha^{t-1}P_f^{t-1}e \geq e$, i.e. $x^f_j \geq 1$ for $j = 1, 2, \dots, N$. □

Let f^∞ be any deterministic policy and let $A = (A_f, A_\nu)$ and $r = \binom{r^f}{r^\nu}$ (the ν-variables are the nonbasic variables). Then, for the basic pair of dual solutions $x = \binom{x^f}{x^\nu}$ and $(v^f, s) = \big(v^f, \binom{s^f}{s^\nu}\big)$, we have $x^\nu = s^f = 0$, and consequently,

$$\begin{aligned} A_fx^f + A_\nu x^\nu &= e &\leftrightarrow\quad x^f &= A_f^{-1}e;\ x^\nu = 0 \\ A_f^Tv^f - s^f &= r^f &\leftrightarrow\quad v^f &= (A_f^T)^{-1}r^f \\ A_\nu^Tv^f - s^\nu &= r^\nu &\leftrightarrow\quad s^\nu &= -r^\nu + A_\nu^T(A_f^T)^{-1}r^f;\ s^f = 0 \end{aligned}$$

If $s_\nu \geq 0$, then $x = \binom{x^f}{x^\nu} = \binom{A_f^{-1}e}{0}$ and $(v, s) = \big(v^f, \binom{s^f}{s^\nu}\big) = \Big((A_f^T)^{-1}r^f, \binom{0}{-r^\nu + A_\nu^T(A_f^T)^{-1}r^f}\Big)$ are a pair of dual optimal solutions of (9.1) and (9.2), respectively.

Lemma 9.2

There is a unique partition $B \subseteq \{1, 2, \dots, M\}$ and $C \subseteq \{1, 2, \dots, M\}$, i.e. $B \cap C = \emptyset$ and $B \cup C = \{1, 2, \dots, M\}$, with $|B| \geq N$ and $|C| \leq M - N$ such that:

(1) *There is at least one optimal solution pair $\big(x^*, \binom{v^*}{s^*}\big)$ that is strictly complementary, i.e. $x^*_j > 0$ for all $j \in B$ and $s^*_j > 0$ for all $j \in C$.*

(2) *For all optimal pairs $\big(x^*, \binom{v^*}{s^*}\big)$, we have $x^*_j = 0$ for all $j \in C$ and $s^*_j = 0$ for all $j \in B$.*

Proof

The strict complementary result is well known for general linear programming (see [108]) with $B \subseteq \{1, 2, \dots, M\}$ the set of variables that are positive for at least one optimal solution and C the set of variables that are zero in all optimal solutions. It is obvious that $|B| \geq N$, and therefore $|C| \leq M - N$, and that $x^*_j = 0$ for all $j \in C$ when x^* is optimal for (9.1).

Notice that the dual program (9.2) has a unique optimal solution $(v^*, s^*) = (v^\alpha, -r + A^Tv^\alpha)$. Take any $j \in B$. Then, there exists an optimal solution x of (9.1) with $x_j > 0$. From the complementary slackness property it follows that $s^*_j = 0$. □

The interpretation of Lemma 9.2 is as follows: since there may exist multiple optimal policies for an MDP, B contains those state-actions each of which appears in at least one optimal policy, and C contains the rest state-actions neither of which appears in an optimal policy. Lets call each action in C a non-optimal action. Then, by the uniqueness of the optimal s^*, we have $s^*_j > 0$ if and only if $j \in C$.

Let f^∞ be a deterministic policy with corresponding basic solution $x = \binom{x^f}{x^\nu}$ and basic dual solution $(v, s) = \big(v^f, \binom{s^f}{s^\nu}\big)$. By the complementary slackness of a dual pair of solutions, we have $s^f = 0$, implying $v^f = (A_f^T)^{-1} r_f$.

Then, $r^T x = (r^f)^T x^f + (r^\nu)^T x^\nu$ and $Ax = A_f x^f + A_\nu x^\nu = e$, implying $x^f = (A_f)^{-1}(e - A_\nu x^\nu)$. Hence,

$$\begin{aligned} r^T x &= (r^f)^T \{A_f^{-1}(e - A_\nu x^\nu)\} + (r^\nu)^T x^\nu \\ &= (r^f)^T A_f^{-1} e - \{(r^f)^T A_f^{-1} A_\nu - (r^\nu)^T\} x^\nu \\ &= (r^f)^T A_f^{-1} e - \{A_\nu^T (A_f^T)^{-1} r^f - r^\nu\}^T x^\nu \\ &= (r^f)^T A_f^{-1} e - (\overline{r}^\nu)^T x^\nu, \end{aligned}$$

where $\overline{r}^\nu := A_\nu^T (A_f^T)^{-1} r^f - r^\nu$. Let $\overline{r}^f := A_f^T v^f - r^f$. Then, $\overline{r}^f = A_f^T (A_f^T)^{-1} r^f - r^f = 0$. The vector $\overline{r} = \binom{\overline{r}^f}{\overline{r}^\nu}$ is called the *reduced cost vector*.

The simplex method

If $\overline{r}^\nu \geq 0$, then is the current policy f^∞ is optimal. Otherwise, let $k := argmin_j\, \overline{r}^\nu_j < 0$ and suppose that $k \in A(i)$. The simplex method will break a tie arbitrarily, and it updates (i.e. changes the current policy f^∞) in exactly one state-action, action k, that is, it updates only the state with the most negative reduced cost.

The policy iteration method

The classic policy iteration method is to update every state that has a negative reduced cost. For each state i, let $k_i := argmin_{j\, A(i)} \overline{r}^\nu_j$. Then, for every state i with $\overline{r}^\nu_j < 0$, k_i will replace the current action $f(i)$).

We first prove the strongly polynomial result for the simplex method. For the improvement of the new policy g^∞ over the current policy f^∞, we have the following result.

Lemma 9.3

For the optimal objective value $z^* = e^T v^\alpha$ and for two consecutive policies f^∞ and g^∞ with Δ the most negative reduced cost, we have the following bounds:

(1) $z^* \leq (r^f)^T x^f - \frac{N}{1-\alpha} \cdot \Delta$.

(2) $z^* - (r^g)^T x^g \leq \{1 - \frac{1-\alpha}{N}\} \cdot \{z^* - (r^f)^T x^f\}$.

Proof

(1) For every feasible solution x of (9.1), we can write $r^T x = (r^f)^T x^f - (\overline{r}^\nu)^T x^\nu$. Since $r^f = 0$, $\overline{r} \geq \Delta \cdot e$ and, by Lemma 9.1, $e^T x = \frac{N}{1-\alpha}$, we also have $(\overline{r}^\nu)^T x^\nu = \overline{r}^T x \geq \Delta \cdot e^T x = \Delta \cdot \frac{N}{1-\alpha}$.

Hence, for all feasible solution x of (9.1), $r^Tx \leq (r^f)^Tx^f - \Delta \cdot \frac{N}{1-\alpha}$. In particular, for an optimal solution x^* of (9.1), we obtain $z^* = r^Tx^* \leq (r^f)^Tx^f - \frac{N}{1-\alpha} \cdot \Delta$.

(2) Since the new basic solution x^g has components with values at least 1 (see Lemma 9.1), the objective value is increased by at least $-\Delta$ and, by part (1) of the present lemma, $-\Delta \geq \frac{1-\alpha}{N} \cdot \{z^* - (r^f)^Tx^f\}$. Hence, $(r^g)^Tx^g - (r^f)^Tx^f \geq \frac{1-\alpha}{N} \cdot \{z^* - (r^f)^Tx^f\}$, implying $z^* - (r^g)^Tx^g \leq \{1 - \frac{1-\alpha}{N}\} \cdot \{z^* - (r^f)^Tx^f\}$. □

Corollary 9.1

If the simplex methods generates the sequence of deterministic policies $f_0^\infty, f_1^\infty, \dots, f_t^\infty, \dots$, then $z^* - (r^{f_t})^Tx^{f_t} \leq \{1 - \frac{1-\alpha}{N}\}^t \cdot \{z^* - (r^{f_0})^Tx^{f_0}\}$ for $t = 0, 1, \dots$.

Lemma 9.4

(1) If policy f^∞ is non-optimal, then there is a state-action $j \in C$ which is basic for f^∞ and satisfies $s_j^* \geq \frac{1-\alpha}{N} \cdot \{z^* - (r^f)^Tx^f\}$, where s^* is the optimal dual slack vector of (9.2).

(2) Let the simplex methods generate the sequence of policies $f_0^\infty, f_1^\infty, \dots, f_t^\infty, \dots$, where f_0^∞ is a non-optimal policy with non-optimal state-action $j \in C$ identified in part (1). Then, if x_j is a basic variable in policy f_t^∞, we have $x_j^{f_t} \leq \frac{N^2}{1-\alpha} \cdot \frac{z^*-(r^{f_t})^Tx^{f_t}}{z^*-(r^{f_0})^Tx^{f_0}}$.

Proof

(1) Since $A^Tv^\alpha - s^* = r$, we have for every feasible solution x of (9.2) $(A^Tv^\alpha - s^*)^Tx = r^Tx \;\leftrightarrow\; (v^\alpha)^TAx - (s^*)^Tx = r^Tx \;\leftrightarrow\; (v^\alpha)^Te - (s^*)^Tx = r^Tx$. Therefore, $z^* - (r^f)^Tx^f = (v^\alpha)^Te - (r^f)^Tx^f = (s^*)^Tx^f = \sum_i s_i^*x_i^f$. Since this summation has N terms, there must be a component j of x^f which satisfies $s_j^*x_j^f \geq \frac{1}{N} \cdot \{z^* - (r^f)^Tx^f\} > 0$ (the last inequality because policy f^∞ is non-optimal). Since $x_j^f \leq \frac{N}{1-\alpha}$ (see Lemma 9.1), we obtain $s_j^* \geq \frac{1-\alpha}{N^2} \cdot \{z^* - (r^f)^Tx^f\} > 0$. Because $s_j^* > 0$, we also have $j \in C$.

(2) Suppose that f_0^∞ is a non-optimal policy with non-optimal state-action $j \in C$ identified in part (1). Take any $t \in \mathbb{N}_0$. Then, we can write $z^* - (r^{f_t})^Tx^{f_t} = (s^*)^Tx^{f_t} \geq s_j^*x_j^{f_t}$. Since $j \in C$, we have $s_j^* > 0$ and we can write $x_j^{f_t} \leq \frac{z^*-(r^{f_t})^Tx^{f_t}}{s_j^*} \leq \frac{N^2}{1-\alpha} \cdot \frac{z^*-(r^{f_t})^Tx^{f_t}}{z^*-(r^{f_0})^Tx^{f_0}}$. □

These lemmas lead to the following key result.

Theorem 9.5

Let f_0^∞ be any non-optimal policy. Then, there is a non-optimal state-action $j \in C$, which is basic for f_0^∞, but which would never appear in any of the policies generated by the simplex method after $t > \frac{N}{1-\alpha} \cdot log\, \frac{N^2}{1-\alpha}$ iterations, starting from f_0^∞.

Proof

From Corollary 9.1, after t iterations of the simplex method, we have $\frac{z^*-(r^{f_t})^Tx^{f_t}}{z^*-(r^{f_0})^Tx^{f_0}} \leq \{1 - \frac{1-\alpha}{N}\}^t$. Therefore, after $t > \frac{N}{1-\alpha} \cdot log\, \frac{N^2}{1-\alpha}$ iterations, we have by Lemma 9.4,

$$x_j^{f_t} \leq \frac{N^2}{1-\alpha} \cdot \frac{z^* - (r^{f_t})^Tx^{f_t}}{z^* - (r^{f_0})^Tx^{f_0}} \leq \frac{N^2}{1-\alpha} \cdot \{1 - \frac{1-\alpha}{N}\}^t.$$

Since $log\,(1-x) \leq -x$ for all $x<1$, we can show that $\frac{N^2}{1-\alpha} \cdot \{1-\frac{1-\alpha}{N}\}^t < 1$ for $t > \frac{N}{1-\alpha} \cdot log\,\frac{N^2}{1-\alpha}$, namely: $\log\,\frac{N^2}{1-\alpha} + t \cdot log\,\{1-\frac{1-\alpha}{N}\} \leq \log\,\frac{N^2}{1-\alpha} - t \cdot \frac{1-\alpha}{N} < 0$ for $t > \frac{N}{1-\alpha} \cdot log\,\frac{N^2}{1-\alpha}$, implying $\frac{N^2}{1-\alpha} \cdot \{1-\frac{1-\alpha}{N}\}^t < 1$ for $t > \frac{N}{1-\alpha} \cdot log\,\frac{N^2}{1-\alpha}$.
Suppose that state-action j appears in any of the policies generated by the simplex method after $t > \frac{N}{1-\alpha} \cdot log\,\frac{N^2}{1-\alpha}$ iterations. Then, $x_j^{f_t} < 1$, which contradicts Lemma 9.1. □

Let $T := \left\lfloor \frac{N}{1-\alpha} \cdot log\,\frac{N^2}{1-\alpha} \right\rfloor + 1$. Then, a non-optimal action of the starting policy f_0^∞ will never be an action of policy f_t^∞ for all $t \geq T$. If policy f_T^∞ is not optimal, there must be a non-optimal action of this policy that would never be an action of a policy f_t^∞ for all $t \geq 2T$. We can repeat this argument and in each of these cycles of T simplex iterations at least one new non-optimal action is eliminated from appearance in any of the future policy cycles, generated by the simplex method. However, we have at most $|C| \leq M-N$ (see Lemma 9.2) many such non-optimal actions to eliminate. Hence, after at most $T \cdot (M-N)$ iterations the simplex method terminates with an optimal policy. This result is summarized in the following theorem.

Theorem 9.6
The simplex method with the most-negative reduced cost pivoting rule is a strongly polynomial algorithm. Starting from any policy, the method terminates in at most $T \cdot (M-N)$ iterations, where $T := \left\lfloor \frac{N}{1-\alpha} \cdot log\,\frac{N^2}{1-\alpha} \right\rfloor + 1$. Furthermore, each iteration uses $\mathcal{O}\big(N \cdot (M-N)\big)$ arithmetic operations.

We now turn our attention to the classic policy iteration method, in which in each iteration every state that has a negative reduced cost is updated and the current action is replaced by the action with the most-negative reduced cost for this state. We have already seen that the policy iteration method may be viewed as a block-pivoting simplex algorithm. In this block-pivoting simplex algorithm for MDPs in one iteration at most $1 \leq k \leq N$ usual pivoting iterations are executed. Hence, one iteration of the classic policy iteration method uses $\mathcal{O}\big(N^2 \cdot (M-N)\big)$ arithmetic operations.

For the block-pivoting simplex algorithm, we have the following facts:
(1) Lemma 9.1 and Lemma 9.2 hold, since they are independent of which of the two methods (block-pivoting or usual pivoting) is used.
(2) Lemma 9.3 still holds for the block-pivoting simplex method, because the the action corresponding to the most-negative reduced cost Δ is always one of the incoming basic variables for the block-pivoting simplex method. Consequently, Corollary 9.1 also holds.
(3) The properties established by Lemma 9.1 are also independent of how the policy sequence is generated as long as the state-action with the most-negative reduced cost is included in the next policy, so that they hold for the block-pivoting simplex method as well.
(4) Theorem 9.5 also holds, since the proof is based on Corollary 9.1 and Lemma 9.1.

Based on the above mentioned facts, we have the following result for the classic policy iteration method.

Theorem 9.7

The classic policy iteration method is a strongly polynomial algorithm. Starting from any policy, the method terminates in at most $T \cdot (M - N)$ iterations, where $T := \left\lfloor \frac{N}{1-\alpha} \cdot log\, \frac{N^2}{1-\alpha} \right\rfloor + 1$. Furthermore, each iteration uses $\mathcal{O}\big(N^2 \cdot (M-N)\big)$ arithmetic operations.

<u>Remark 1</u>

The pivoting rule makes the difference. Melekopoglou and Condon ([197]) showed that a special policy iteration algorithm, where only the action for the state with the smallest index is updated, needs an exponential number of iterations. Notice that this smallest-index rule is a popular rule in the simplex method for general LP problems, because it avoids cycling in the presence of degeneracy. On the other hand, the most-negative reduced cost rule is exponential for solving some other LP problems. Thus, searching for suitable pivoting rules for solving different LP problems is essential, and one cannot rule out the simplex method simply because the behavior of one pivoting rule on one problem is shown to be exponential. The question: *Is there any strongly polynomial algorithm for solving the MDP regardless the discount factor* α is still an open problem.

<u>Remark 2</u>

Mansour and Singh ([195]) have derived an upper bound on the number of iterations for the policy iteration method that does not depend on the discount factor α. This bound is $\frac{1}{N} \cdot 2^N$ for an MDP that has in each state 2 actions. They showed that the *greedy* policy iteration method, which greedily accepts all single-state action changes that are improvements, will take at most $\mathcal{O}\big(\frac{1}{N} \cdot 2^N\big)$ iterations. Below we present their results in detail.

We introduce a partial ordering between the deterministic policies as follows. For two deterministic policies f^∞ and g^∞, we define $g^\infty \succ f^\infty$ if $v_i^\alpha(g^\infty) \geq v_i^\alpha(f^\infty)$ for each state i, and for at least one state j, $v_j^\alpha(g^\infty) > v_j^\alpha(f^\infty)$. If for every state i we have $v_i^\alpha(g^\infty) \geq v_i^\alpha(f^\infty)$, then $g^\infty \equiv f^\infty$, i.e. g^∞ and f^∞ are equivalent. The partial ordering tells us when a policy is better than another and when they are incomparable. Clearly any optimal policy is better then all suboptimal policies and equivalent to all other optimal policies. This partial order is central to our analysis.

Assume that we have an MDP with two actions in each state. We know from the policy iteration method that in every iteration the current policy, say f^∞, is replaced by the next policy, say f^∞, where $g^\infty \succ f^\infty$. Therefore, policies h^∞ such that $f^\infty \preceq h^\infty \prec g^\infty$ are skipped. How many such policies h^∞ are here in each iteration? There is at least one such policy: the current policy f^∞. This, of course, implies a trivial upper bound of 2^N iterations. For special choices of the next policy g^∞ we shall perform a more careful analysis of the number of skipped policies. The more policies we can skip at each iteration, the better, i.e. lower, the upper bound will be.

Lemma 9.5

Let f^∞ and g^∞ be two policies whose actions differ in only one state. Then, f^∞ and g^∞ are comparable, i.e. either $g^\infty \succ f^\infty$, $g^\infty \prec f^\infty$ or $g^\infty \equiv f^\infty$.

Proof

Suppose that f^∞ and g^∞ differ only in state i, i.e. $g(k) = f(k)$ for all $k \neq i$ and $g(i) \neq f(i)$.

Then, $s_{kg(k)} := r_k(g) + \alpha \sum_j p_{kj}(g) v_j^\alpha(f^\infty) - v_k^\alpha(f^\infty) = 0$ for all $k \neq i$.

For $s_{ig(i)} := r_i(g) + \alpha \sum_j p_{ij}(g) v_j^\alpha(f^\infty) - v_i^\alpha(f^\infty)$ there are three possibilities:

(1) $s_{ig(i)} > 0$. Then, $L_g v^\alpha(f^\infty > v^\alpha(f^\infty)$, implying $v^\alpha(g^\infty) > v^\alpha(f^\infty)$, i.e. $g^\infty \succ f^\infty$.

(2) $s_{ig(i)} < 0$. Then, $L_g v^\alpha(f^\infty < v^\alpha(f^\infty)$, implying $v^\alpha(g^\infty) < v^\alpha(f^\infty)$, i.e. $g^\infty \prec f^\infty$.

(3) $s_{ig(i)} = 0$. Then, $L_g v^\alpha(f^\infty = v^\alpha(f^\infty)$, implying $v^\alpha(g^\infty) = v^\alpha(f^\infty)$, i.e. $g^\infty \equiv f^\infty$. □

Given a policy f^∞, let $SA(f) := \{(i,a) \mid s_{ia}(f) > 0\}$ be the set of improving state-actions and $S(f) := \{i \mid (i,a) \in SA(f)$ for at least one action $a \in A(i)\}$ be the set of states that have at least one improving action. Let $f_1^\infty, f_2^\infty, \dots$ be a sequence of policies generated by a run of the policy iteration method.

Lemma 9.6

There are no indices k and l with $k < l$ such that $S(f_k) \subseteq S(f_l)$.

Proof

For any pair of policies f^∞ and g^∞ such $f(i) = g(i)$ for all $i \in S(f)$, we have $f^\infty \succ g^\infty$, namely:

> Consider an MDP M^* such that in the states of $S(f)$ the action $f(i)$ is the only action. Clearly, both $f^\infty)$ and g^∞ are valid policies for M^*. On the other hand, in M^* there are no improving actions. Hence, f^∞ is an optimal policy in M^*. Therefore, $f^\infty \succ g^\infty$.

Now, we prove the lemma by contradiction. Assume that $k < l$ and $S(f_k) \subseteq S(f_l)$.

Let $SA_{kl} := \{(i,a) \in SA(f_k) \mid a \neq f_l(i)\}$. If $(i,a) \in SA_{kl}$, then $(i,a) \in SA(f_l)$, namely:

> If $(i,a) \in SA_{kl}$, then $i \in S(f_k) \subseteq S(f_l)$. Because $i \in S(f_l)$ and $s_{if_l(i)}(f_l) = 0$, we have $s_{ia}(f_l) > 0$, since there are only two actions. Therefore, $(i,a) \in SA(f_l)$.

Note that $SA_{kl} \neq \emptyset$, because otherwise $f_k(i) = f_l(i)$ for all $i \in S(f_k)$, implying $f_k^\infty \succ f_l^\infty$ which contradicts that $k < l$ in the sequence of policies generated by a run of the policy iteration method. Then, let g^∞ be a policy obtained from f_l^∞ in the policy iteration method by taking improving actions on SA_{kl}, i.e. taking the actions of $f_k \in SA_{kl}$ (because there are only two actions in each state). Hence, $g^\infty \succ f_l^\infty \succ f_k^\infty$. On the other hand, since $f_k(i) = g(i)$ for all $i \in S(f_k)$, we have $f_k^\infty \succeq g^\infty$, which gives a contradiction. □

So far we have showed that a nonempty subset of states can appear at most once in general policy iteration. This still leaves open the possibility that all subsets appear in the run of the algorithm, and thus we observe all 2^N policies. The next step is to show that each time we replace a current policy f^∞ by the next policy g^∞ by taking the other action in all states of $S(f)$, then we rule out more policies. This is done in the *greedy policy iteration method.*

Greedy policy iteration

In the greedy policy iteration method, the next policy g^∞ is obtained from the current policy f^∞ by taking in all states of $S(f)$ the improving action.

Lemma 9.7

Let g^∞ be the policy obtained from f^∞ by in the greedy policy iteration method, and let $k := |S(f)|$. Then, there exist k different policies $h_1^\infty, h_2^\infty, \dots, h_k^\infty$ such that $g^\infty \succeq h_i^\infty \succ f^\infty$ for $i = 1, 2, \dots, k$.

Proof

The proof is by induction on k. If $k = 1$, then $h_1 := g$ satisfies $g^\infty \succeq h_1^\infty \succ f^\infty$. For the rest of the proof assume that $k \geq 2$. Consider all the single state modifications of f^∞ using the alternative action in exactly one of the states of $S(f)$ and let $g_1^\infty, g_2^\infty, \dots, g_k^\infty$ be the corresponding policies. Since the policies are partial ordered at least one policy, say g_1^∞, has the property that for every $2 \leq i \leq k$ either $g_1^\infty \preceq g_i^\infty$ or g_1^∞ and g_i^∞ are incomparable.

Without loss of generality we may assume that $SA(f) = \{(s_1, a_1), (s_2, a_2), \dots, (s_k, a_k)\}$ and that $g_1(s_1) = a_1$. For the pairs (s_i, a_i), $2 \leq i \leq k$, we shall show later that $(s_i, a_i) \in SA(g_1^\infty)$.

First, we consider all the single state modifications of g_1^∞ by taking in state s_i the action a_i for $i = 2, 3, \dots, k$. Let $g_{1,2}^\infty, g_{1,3}^\infty, \dots, g_{1,k}^\infty$ be the corresponding policies. By Lemma 9.5 either $g_1^\infty \succ g_{1,i}^\infty$ or $g_1^\infty \preceq g_{1,i}^\infty$ for $i = 2, 3, \dots, k$. We shall show that $g_1^\infty \succ g_{1,i}^\infty$ is not possible.

> For a proof, assume that $g_1^\infty \succ g_{1,i}^\infty$. Note that g_i^∞ and $g_{1,i}^\infty$ differ only in state s_1. Hence, by Lemma 9.5, either $g_i^\infty \succ g_{1,i}^\infty$ or $g_i^\infty \preceq g_{1,i}^\infty$. If $g_{1,i}^\infty \succeq g_i^\infty$, then $g_1^\infty \succ g_{1,i}^\infty \succeq g_i^\infty$, contradicting the property of policy g_1^∞. So, $g_i^\infty \succ g_{1,i}^\infty$, implying $\big(s_1, f(s_1)\big) \in SA(g_{1,i})$. Since $g_1^\infty \succ g_{1,i}^\infty$, we also have $\big(s_i, f(s_i)\big) \in SA(g_{1,i})$. Because f is obtained from $g_{1,i}$ by replacing the actions a_1 and a_i by $f(s_1)$ and $f(s_i)$, respectively, we obtain $f^\infty \succ g_{1,i}^\infty$, contradicting the fact that $g_{1,i}^\infty \succ f^\infty$. So we have shown that $g_{1,i}^\infty \succ g_1^\infty$.

Since $g_{1,i}^\infty \succ g_1^\infty$ for $i = 2, 3, \dots, k$, we have $(s_i, a_i) \in SA(g_1)$ for $i = 2, 3, \dots, k$. Therefore, $|S(g_1)| = |S(f)| - 1 = k - 1$. Let $h_1 := g_1$. Then, the lemma follows using also the induction hypothesis on $g_1^\infty = h_1^\infty$.

Theorem 9.8

The greedy policy iteration method considers at most $\mathcal{O}\big(\frac{1}{N} \cdot 2^N\big)$ different policies.

Proof

Let f^∞ be a policy that occurs in some iteration of the greedy policy iteration method. We distinguish the following two cases: (1) $|S(f)| > \frac{1}{3}N$; (2) $|S(f)| \leq \frac{1}{3}N$.

Case (1): $|S(f)| > \frac{1}{3}N$

By Lemma 9.7, we have at least $\frac{1}{3}N$ policies better than the current policy f^∞ that are ruled out after this iteration. Since the MDP has 2^N policies, there are at most $\frac{2^N}{N/3} = 3 \cdot \frac{2^N}{N}$ iterations of this type.

Case (2): $|S(f)| \leq \frac{1}{3}N$

By Lemma 9.6, we do not consider the same set of improving actions twice. Hence, the total number of iterations of this type is at most $\sum_{k=0}^{N/3} \binom{N}{k}$.

Assuming that $\sum_{k=0}^{N/3} \binom{N}{k} \leq 2 \cdot \binom{N}{N/3} \leq 3 \cdot \frac{2^N}{N}$, which we shall prove later, we have shown that the total number of iterations is at most $3 \cdot \frac{2^N}{N} + 3 \cdot \frac{2^N}{N} = 6 \cdot \frac{2^N}{N} = \mathcal{O}\big(\frac{1}{N} \cdot 2^N\big)$.

Proof that $\sum_{k=0}^{m} \binom{N}{k} \leq 2 \cdot \binom{N}{m}$ for $m \leq \frac{1}{3}N$

We apply induction on m. For $m = 0$, the inequality is obvious. Assume that $\sum_{k=0}^{m} \binom{N}{k} \leq 2 \cdot \binom{N}{m}$ for some m and consider $\sum_{k=0}^{m+1} \binom{N}{k} = \sum_{k=0}^{m} \binom{N}{k} + \binom{N}{m+1}$. Then, by the induction hypothesis, $\sum_{k=0}^{m+1} \binom{N}{k} \leq 2 \cdot \binom{N}{m} + \binom{N}{m+1}$. Hence, we have to show that $2 \cdot \binom{N}{m} \leq \binom{N}{m+1}$ for $m + 1 \leq \frac{1}{3}N$.

We have

$$2 \cdot \binom{N}{m} = 2 \cdot \frac{N!}{(N-m)!m!} \leq \binom{N}{m+1} = \frac{N!}{(N-m-1)!(m+1)!} \;\Leftrightarrow\; \frac{2}{N-m} \leq \frac{1}{m+1} \; Lleftrightarrow \; m+1 \leq \frac{1}{3}(N+1).$$

If $m \leq \frac{1}{3}N$, then certainly $m + 1 = \frac{1}{3}(N + 1)$.

Proof that $2 \cdot \binom{N}{N/3} \leq 3 \cdot \frac{2^N}{N}$

We apply induction on N. For convenience, we take $N = 3, 6, 9, \ldots$.

If $N = 3$, we obtain $2 \cdot \binom{3}{1} = 6 \leq 3 \cdot \frac{2^3}{3} = 8$. Assume that $2 \cdot \binom{N}{N/3} \leq 3 \cdot \frac{2^N}{N}$ for some N and consider $N + 3$ instead of N. Then, we can write

$$\begin{aligned} 2 \cdot \binom{N+3}{N/3+1} &= 2 \cdot \binom{N}{N/3} \cdot \frac{(N+1)(N+2)(N+3)}{(\frac{2}{3}N+1)(\frac{2}{3}N+2)(\frac{1}{3}N+1)} \\ &\leq 3 \cdot \frac{2^N}{N} \cdot \frac{(N+1)(N+2)(N+3)}{(2N+3)(2N+6)(N+3)} \cdot 27 = 81 \cdot \frac{2^{N-1}}{N} \cdot \frac{(N+1)(N+2)}{(2N+3)(N+3)}. \end{aligned}$$

To show that $2 \cdot \binom{N+3}{N/3+1} \leq 3 \cdot \frac{2^{N+3}}{N+3}$, we have to show that $\frac{(N+1)(N+2)}{N(2N+3)} \leq \frac{16}{27}$, which is equivalent to $5N^2 - 6N - 54 \geq 0$, i.e. $N \geq 4$. Since we have already shown the inequality for $N = 3$, the inequality is verified for all N. □

Exponential version of policy iteration

We shall present an MDP and a version of the policy iteration method for which the number of iterations is exponential in the number of states.

In this MDP, the state space $S = S_1 \cup S_2 \cup S_3$, where:

- S_1 has N *decision states*, labeled $1, 2, \ldots, N$;
- S_2 has $N + 1$ *chance states*, labeled $0, N + 1, N + 2, \ldots, 2N$;
- S_3 has 2 *sinks*, labeled A and B.

The states of S_1 have two actions, labeled 0 and 1. If in state i action 0 is chosen, then there is a deterministic transition to state $i - 1$ $(1 \leq i \leq N)$; if in state i action 1 is chosen, there is a deterministic transition to state $N + i$ $(1 \leq i \leq N)$.

In the states of S_2 there is only one action, so there is no choice. In the states $N + i$ $(2 \leq i \leq N)$ there are transitions to the states $N + i - 1$ and $i - 2$, each with probability $\frac{1}{2}$; in state $N + 1$ there are transitions to the sinks A and B, each with probability $\frac{1}{2}$; in state 0 there are transitions to the sink A and the state N, each also with probability $\frac{1}{2}$. The sinks A and B are absorbing states in which the process terminates with a final cost: 1 in sink A and 0 in sink B. All other costs are 0.

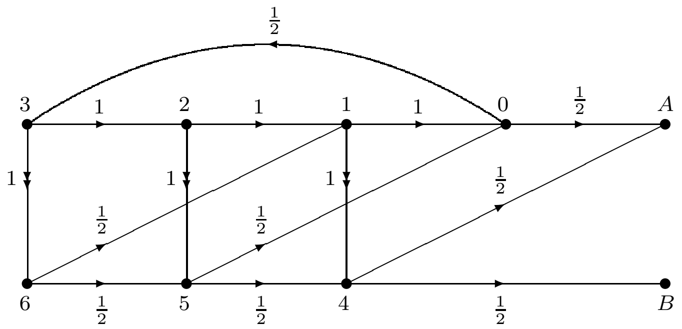
As utility function the total cost is considered. This can be interpreted as the probability to reach sink A. Minimizing the total cost is equivalent to minimizing the probability to terminate in sink A. Since for every policy the process terminates in A or B, this is equivalent to maximizing the probability to terminate in sink B. This model satisfies Assumption 1.2. Furthermore, the model is transient, because under every policy and starting state the process terminates with probability 1 in one of the sinks. As we have shown in Chapter 4, the properties of the discounted model with $\alpha = 1$, i.e. the utility function is the total cost, are also valid for the transient model.

We terminate in a sink from either state 0 or state $N+1$. In both cases there is a probability of $\frac{1}{2}$ to end in sink A. Hence, the total cost for any starting state from the decision states S_1 is at least 1. Consider the policy that takes action 0 in the states $N, N-1, \dots, 2$ and action 1 in state 1. For this policy the total cost for any starting state from S_1 is equal to $\frac{1}{2}$. So, this is the optimal policy.

Since there are only two actions in the states of S_1, a policy can be represented as an N-bits vector $f = f_N f_{N-1} \cdots f_1$, where f_i is the label of the action taken in state i $(1 \leq i \leq N)$. So, the optimal policy is $00\dots01$ with total cost $\frac{1}{2}$ for the states of S_1. The total cost, given starting state i and policy f, is denoted by $v_i(f)$. Note that $v_{N+1}(f) = \frac{1}{2}$ for every policy f. By induction on N, it is easy to verify that for the optimal policy f_* we have $v_{N+i}(f_*) = \frac{2^i+1}{2^{i+1}}$ for $i = 2, 3, \dots, N$.

Example 9.5

Take $N = 3$. The MDP model can be represented by the following directed graph.



For the policy $f = 011$, the total cost for the possible starting states can be computed as unique solution of the following system of linear equations:

$$\begin{array}{rclrclrcl}
v_0(f) & = & \frac{1}{2}v_A(f) + \frac{1}{2}v_3(f); & v_3(f) & = & v_2(f); & v_6(f) & = & \frac{1}{2}v_1(f) + \frac{1}{2}v_5(f); \\
v_1(f) & = & v_4(f); & v_4(f) & = & \frac{1}{2}v_A(f) + \frac{1}{2}v_B(f); & v_A(f) & = & 1; \\
v_2(f) & = & v_5(f); & v_5(f) & = & \frac{1}{2}v_0(f) + \frac{1}{2}v_4(f); & v_B(f) & = & 0.
\end{array}$$

Note that the equations for the states in $S_2 \cup S_3$, in this case the states $0, 4, 5, 6, A, B$, are independent of the policy.

The unique solution of this system is: $v_0(f) = \frac{5}{6}$; $v_1(f) = \frac{1}{2}$; $v_2(f) = \frac{2}{3}$; $v_3(f) = \frac{2}{3}$; $v_4(f) = \frac{1}{2}$; $v_5(f) = \frac{2}{3}$; $v_6(f) = \frac{7}{12}$; $v_A(f) = 1$; $v_B(f) = 0$.

Similarly, we can compute the total cost for every policy and every starting state. This is summarized in the following tabular.

f	$v_0(f)$	$v_1(f)$	$v_2(f)$	$v_3(f)$	$v_4(f)$	$v_5(f)$	$v_6(f)$	$v_A(f)$	$v_B(f)$
000	1	1	1	1	$\frac{1}{2}$	$\frac{3}{4}$	$\frac{7}{8}$	1	0
100	$\frac{9}{10}$	$\frac{9}{10}$	$\frac{9}{10}$	$\frac{4}{5}$	$\frac{1}{2}$	$\frac{7}{10}$	$\frac{4}{5}$	1	0
110	$\frac{9}{10}$	$\frac{9}{10}$	$\frac{7}{10}$	$\frac{4}{5}$	$\frac{1}{2}$	$\frac{7}{10}$	$\frac{4}{5}$	1	0
010	$\frac{5}{6}$	$\frac{5}{6}$	$\frac{2}{3}$	$\frac{2}{3}$	$\frac{1}{2}$	$\frac{2}{3}$	$\frac{3}{4}$	1	0
011	$\frac{5}{6}$	$\frac{1}{2}$	$\frac{2}{3}$	$\frac{2}{3}$	$\frac{1}{2}$	$\frac{2}{3}$	$\frac{7}{12}$	1	0
111	$\frac{11}{14}$	$\frac{1}{2}$	$\frac{9}{14}$	$\frac{4}{7}$	$\frac{1}{2}$	$\frac{9}{14}$	$\frac{4}{7}$	1	0
101	$\frac{11}{14}$	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{4}{7}$	$\frac{1}{2}$	$\frac{9}{14}$	$\frac{4}{7}$	1	0
001	$\frac{3}{4}$	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{5}{8}$	$\frac{9}{16}$	1	0

We see in this tabular that each new policy is better than the previous one. Hence, there is a sequence of 8 improving policies. In this example there are $2^N = 8$ policies. A new policy is obtained from the previous one by taking the first improving action when the decision states are ordered as in the policy, i.e. the sequence $N, N-1, \dots, 1$. This example, for a general N, gives the desired exponential number of iterations. Furthermore, the sequence of policies for a model with one more decision state can be obtained from the previous one by first adding a 0 after the previous sequence, so for $N = 4$ we obtain 0000, 1000, 1100, 0100, 0110, 1110, 1010, 0010. Then, add a 1 after the previous sequence and pass through this sequence in the reverse order, so for $N = 4$: 0011, 1011, 1111, 0111, 0101, 1101, 1001, 0001. Hence, the whole sequence for $N = 4$ we have the complete sequence 0000, 1000, 1100, 0100, 0110, 1110, 1010, 0010, 0011, 1011, 1111, 0111, 0101, 1101, 1001, 0001. Notice that this is a sequence of binary numbers in which a subsequent number differs from the previous one in only one bit. Such a sequence is called a *Gray code*.

We shall consider the *simple policy iteration algorithm*. In this algorithm we take in each iteration only one improving action and this new action is chosen in the state with the highest index. For the MDP model of this the algorithm is as follows.

Algorithm 9.1 *Simple policy iteration algorithm*
Input: Instance of an MDP as described above.
Output: An optimal deterministic policy f^∞, represented by $f = f_N f_{N-1} \dots f_2 f_1$.

```
1. Choose as initial policy $f = f_N f_{N-1} \dots f_2 f_1$ with $f_k = 0$ for $k = N, N-1, \dots, 1$.

2. for $k = N, N-1, \dots, 1$ do
       if if state $k$ has an improving action then
           begin $f_k := 1 - f_k$; go to step 3 end

3. if in step 2 no improving action is found then $f = f_N f_{N-1} \dots f_2 f_1$ is an optimal policy
   (STOP)

   else go to step 2.
```

Define for policy $f = f_N f_{N-1} \dots f_2 f_1$ the numbers $a_1, a_2, \dots, a_N$ by $\begin{cases} a_1 & = & -\frac{1}{2}; \\ a_{k+1} & = & (\frac{1}{2} - f_k)a_k, \ 1 \le k \le N-1. \end{cases}$

Lemma 9.8

$a_{k+1} = \frac{1}{2}(-1)^{f_k} a_k$ *for* $k = 1, 2, \dots, N-1$.

Proof

If $f_k = 1$, then $\frac{1}{2}(-1)^{f_k} a_k = -\frac{1}{2}a_k = (\frac{1}{2} - f_k)a_k = a_{k+1}$.
If $f_k = 0$, then $\frac{1}{2}(-1)^{f_k} a_k = \frac{1}{2}a_k = (\frac{1}{2} - f_k)a_k = a_{k+1}$. □

Lemma 9.9

$v_{N+k}(f) - v_{k-1}(f) = \frac{a_k}{a_{k-1}}\{v_{N+k-1}(f) - v_{k-2}(f)\}$ *for* $k = 2, 3, \dots, N$.

Proof

For any $k = 2, 3, \dots, N$, we have

$$\begin{array}{rcl} v_{N+k}(f) - v_{k-1}(f) & = & \{\frac{1}{2}v_{N+k-1}(f) + \frac{1}{2}v_{k-2}(f)\} - \{f_{k-1}v_{N+k-2}(f) + (1 - f_{k-1})v_{k-2}(f)\} \\ & = & (\frac{1}{2} - f_{k-1})v_{N+k-1}(f) - (\frac{1}{2} - f_{k-1})v_{k-2}(f) \\ & = & (\frac{1}{2} - f_{k-1})\{v_{N+k-1}(f) - v_{k-2}(f)\} \\ & = & \frac{a_k}{a_{k-1}}\{v_{N+k-1}(f) - v_{k-2}(f)\} \end{array}$$ □

The transitions in state 0 imply $v_0(f) = \frac{1}{2}v_A(f) + \frac{1}{2}v_N(f) = \frac{1}{2} + \frac{1}{2}v_N(f)$. The next lemma gives a recurrence relation for the total cost in the other states.

Lemma 9.10

For every $1 \le k \le N$, *we have* $v_{N+k}(f) = v_{k-1}(f) + a_k v_N(f)$ *and* $v_k(f) = v_{k-1}(f) + f_k a_k v_N(f)$.

Proof

We apply induction on k.

For $k = 1$, we have

$v_{N+1}(f) = \frac{1}{2}v_A(f) + \frac{1}{2}v_B(f) = \frac{1}{2}$ and $v_0(f) + a_1 v_N(f) = \frac{1}{2} + \frac{1}{2}v_N(f) - v_N(f) = \frac{1}{2}$.
If $f_1 = 0$, then $v_0(f) + f_1 a_1 v_N(f) = v_0(f)$ and $v_1(f) = v_0(f)$.
If $f_1 = 1$, then $v_0(f) + f_1 a_1 v_N(f) = v_0(f) - \frac{1}{2}v_N(f) = \frac{1}{2}$ and $v_1(f) = v_{N+1}(f) = \frac{1}{2}$.
Therefore, we have shown the lemma for $k = 1$.

The proof of the induction step is as follows.

$$\begin{array}{rcl} v_{N+k}(f) - v_{k-1}(f) & = & \frac{a_k}{a_{k-1}}\{v_{N+k-1}(f) - v_{k-2}(f)\} \\ & = & \frac{a_k}{a_{k-1}} \cdot a_{k-1}v_N(f) = a_k v_N(f), \end{array}$$

the the first equality by Lemma 9.9 and the second equality by the induction hypothesis.
If $f_k = 0$, then $v_k(f) = v_{k-1}(f)$ and $v_{k-1}(f) + f_k a_k v_N(f) = v_k - 1(f)$.
If $f_k = 1$, then $v_k(f) = v_{N+k}(f)$ and $v_{k-1}(f) + f_k a_k v_N(f) = v_{k-1}(f) + a_k v_N(f) = v_{N+k}(f)$,
the last equality is shown some lines above. This completes the proof of this lemma. □

Lemma 9.11

State k has an improving action if and only if either $f_k = 0$ and $a_k < 0$ or $f_k = 1$ and $a_k > 0$.

Proof

For $k = 1$, we have an improving action if and only if $f_1 = 0$ (because then $v_1(f) = \frac{1}{2}+\frac{1}{2}v_N(f) = \frac{3}{4}$ and for $f_1 = 1$, we have $v_1(f) = v_{N+1}(f) = \frac{1}{2}$); we also have $a_1 < 0$, so the lemma holds.
For $k \geq 2$, we distinguish between the following four cases:
(1) State k has an improving action and $f_k = 0$:
By Lemma 9.10, $v_k(f) = v_{k-1}(f) > v_{N+k}(f) = vk - 1(f) + a_k v_N(f) \ \rightarrow \ a_k < 0$.
(2) State k has an improving action and $f_k = 1$:
By Lemma 9.10, $v_k(f) = v_{N+k}(f) = vk - 1(f) + a_k v_N(f) > v_{k-1}(f) \ \rightarrow \ a_k > 0$.
(3) State k has no improving action and $f_k = 0$:
By Lemma 9.10, $v_k(f) = v_{k-1}(f) \leq v_{N+k}(f) = vk - 1(f) + a_k v_N(f) \ \rightarrow \ a_k \geq 0$.
(4) State k has no improving action and $f_k = 1$:
By Lemma 9.10, $v_k(f) = v_{N+k}(f) = vk - 1(f) + a_k v_N(f) \leq v_{k-1}(f) \ \rightarrow \ a_k \leq 0$. □

Corollary 9.2

If state k has an improving action and in some state $l > k$ the alternative action is chosen, then state k has still an improving action.

Proof

Since a_k is not influenced by the action in larger states (see Lemma 9.8), the property of Lemma 9.11still holds. Therefore, then state k has still an improving action. □

Lemma 9.12

If in the current iteration f_k is changed and $f_N f_{N-1} \cdots f_{k+2} f_{k+1} = 00\ldots01$, then all states $N, N-1, \ldots, k+2, k+1$ have improving actions.

Proof

First, suppose that f_k is switched from 0 to 1, i.e. for $f_k = 0$ state k has an improving action. From Lemma 9.11 it follows that $a_k < 0$. By Lemma 9.8, $a_{k+1} = \frac{1}{2}(-1)^{f_k} a_k = -\frac{1}{2}a_k > 0$. Because $f_{k+1} = 1$, state $k+1$ has an improving action. Since $a_{k+2} = \frac{1}{2}(-1)^{f_{k+1}} a_{k+1} = -\frac{1}{2}a_{k+1} < 0$ and $f_{k+2} = 0$, state $k+2$ has also an improving action. It can similarly be proved by induction that the other larger numbered state have improving actions, too.
Next, suppose that f_k is switched from 1 to 0. From Lemma 9.11 it follows that $a_k > 0$. By Lemma 9.8, $a_{k+1} = \frac{1}{2}(-1)^{f_k} a_k = \frac{1}{2}a_k > 0$, so by Lemma 9.11, state $k+1$ has an improving action. The remaining part of the proof is the same as before. □

Lemma 9.13

The following two statements hold for every $1 \leq k \leq N$:

(1) If $f_N f_{N-1} \cdots f_{k+1} f_k = 00\ldots01$ and the states $N, N-1, \ldots, k+1, k$ have improving actions, then during the next $2^{N-k+1} - 1$ iterations of the simple policy iteration algorithm switches are made in the states $N, N-1, \ldots, k+1, k$ to reach the policy $f_N f_{N-1} \cdots f_{k+1} f_k = 00\ldots00$.

(2) If $f_N f_{N-1} \cdots f_{k+1} f_k = 00\ldots00$ and the states $N, N-1, \ldots, k+1, k$ have improving actions, then during the next $2^{N-k+1} - 1$ iterations of the simple policy iteration algorithm switches are made in the states $N, N-1, \ldots, k+1, k$ to reach the policy $f_N f_{N-1} \cdots f_{k+1} f_k = 00\ldots01$.

Proof

The lemma will be proved by induction on k, where $k = N, N-1, \ldots, 2, 1$. It is obvious that the statements hold for $k = N$. Assume as induction hypothesis that the statements hold for $k = m$. We shall prove that the statements hold for $k = m - 1$. We shall prove statement (1); statement (2) can be proved following the same reasoning.

Consider the policy $f_N f_{N-1} \cdots f_m f_{m-1} = 00\ldots01$ and suppose that the states $N, N-1, \ldots, m, m-1$ have improving actions. The simple policy iteration algorithm examines the states in the order $N, N-1, \ldots$. From the second statement of the induction hypothesis, we deduce that it performs $2^{N-m+1} - 1$ iterations to reach the policy for which $f_N f_{N-1} \cdots f_{m+1} f_m = 00\ldots01$.

The state $m - 1$ has still an improving action (see Corollary 9.2), so f_{m-1} is switched in the next iteration from 1 to 0. By Lemma 9.12, all states $N, N-1, \ldots, m$ have improving actions. From the first statement of the induction hypothesis, we deduce that the algorithm performs $2^{N-m+1} - 1$ iterations to reach the policy for which $f_N f_{N-1} \cdots f_{m+1} f_m = 00\ldots00$.

Hence, after $(2^{N-m+1} - 1) + 1 + (2^{N-m+1} - 1) = 2^{N-(m-1)+1} - 1$ iterations of the simple policy iteration algorithm the policy $f_N f_{N-1} \cdots f_m f_{m-1} = 00\ldots00$ is reached. □

Theorem 9.9

The simple policy iteration algorithm requires an exponential number of iterations in the worse case.

Proof

Since we start with the policy $f = 00\ldots00$, we have $a_1 = -\frac{1}{2}$ and $a_{k+1} = \frac{1}{2} a_k$ for $k = 1, 2, \ldots, N-1$, implying $a_k = -(\frac{1}{2})^k < 0$. Then, by Lemma 9.11, all states have improving actions. By Lemma 9.13, part (2) with $k = 1$, the simple policy iteration algorithm requires $2^N - 1$ iterations to reach the optimal policy $f = 00\ldots01$. □

9.1.5 Value iteration for discounted MDPs

In the value iteration method, given an initial vector v^1, a sequence $v^2, v^3, \ldots$ of vectors and a sequence $f_1^\infty, f_2^\infty, \ldots$ of policies is computed, using the formula $v^{n+1} = Uv^n = L_{f_n} v^n$ The vector v^n and the policy f_n^∞ are used as approximations for the value vector v^α and for an optimal policy f_*^∞. From Lemma 3.9, we have the bound $\|v^\alpha(f_n^\infty) - v^\alpha\|_\infty \leq 2\alpha^n (1-\alpha)^{-1} \cdot \|Uv^1 - v^1\|_\infty,\ n \in \mathbb{N}$.

In order to obtain an ε-optimal policy, we set $2\alpha^n(1-\alpha)^{-1} \cdot \|Uv^1 - v^1\|_\infty \le \varepsilon$, which yields

$$n \ge n_1 := \frac{1}{|\log \alpha|} \cdot \log \left\{ \frac{2 \cdot \|Uv^1 - v^1\|_\infty}{\varepsilon \cdot (1-\alpha)} \right\}. \tag{9.3}$$

Similarly, we can derive another bound based on the *span*. From Theorem 3.8 part (2), we obtain $\|v^\alpha(f_n^\infty) - v^\alpha\|_\infty \le \alpha(1-\alpha)^{-1} \cdot \mathit{span}\,(Uv^n - v^n)$. Then, using the result of Exercise 3.9, we have $\|v^\alpha(f_n^\infty) - v^\alpha\|_\infty \le \alpha^n(1-\alpha)^{-1} \cdot \mathit{span}\,(Uv^1 - v^1),\ n \in \mathbb{N}$. In order to obtain an ε-optimal policy, we set $\alpha^n(1-\alpha)^{-1} \cdot \mathit{span}\,(Uv^1 - v^1) \le \varepsilon$, which yields

$$n \ge n_2 := \frac{1}{|\log \alpha|} \cdot \log \left\{ \frac{\mathit{span}\,(Uv^1 - v^1)}{\varepsilon \cdot (1-\alpha)} \right\}. \tag{9.4}$$

Each iteration of the value iteration method needs at most $N \cdot M$ arithmetic operations, where $M := \sum_{i=1}^N A(i)|$. Hence, we have the following result, which shows that an ε-optimal policy can be computed in polynomial time.

Theorem 9.10

(1) *The value iteration method Algorithm 3.4 with stopping criterion* $\|y - x\|_\infty \le \frac{1}{2} \cdot \frac{1-\alpha}{\alpha} \cdot \varepsilon$ *computes an* ε*-optimal policy and a* $\frac{1}{2}\varepsilon$*-approximation of the value vector* v^α *and has complexity* $\mathcal{O}(N \cdot M \cdot n_1)$*, where* $n_1 := \frac{1}{|\log \alpha|} \cdot \log \left\{ \frac{2 \cdot \|Uv^1 - v^1\|_\infty}{\varepsilon \cdot (1-\alpha)} \right\}$.

(2) *The value iteration method Algorithm 3.4 with stopping criterion* $\mathit{span}\,(y - x) \le \frac{1-\alpha}{\alpha} \cdot \varepsilon$ *computes an* ε*-optimal policy and a* $\frac{1}{2}\varepsilon$*-approximation of the value vector* v^α *and has complexity* $\mathcal{O}(N \cdot M \cdot n_2)$*, where* $n_2 := \frac{1}{|\log \alpha|} \cdot \log \left\{ \frac{\mathit{span}\,(Uv^1 - v^1)}{\varepsilon \cdot (1-\alpha)} \right\}$.

Next, we shall show that, under a reasonable assumption, also an optimal policy can be computed in polynomial time. Therefore, we need the following assumption.

Assumption 9.1

The rewards $r_i(a),\ (i,a) \in S \times A$ and the components $v_i^1,\ i \in S$, are integers; the discount factor α and the transition probabilities $p_{ij}(a),\ (i,a) \in S \times A,\ j \in S$, are rational numbers.

Let δ be the smallest integer such that: (1) $\delta \cdot \alpha$ is integer; (2) $\delta \cdot p_{ij}(a)$ is integer for all $(i,a,j) \in S \times A \times S$; (3) $|r_i(a)| \le \delta$ for all $(i,a) \in S \times A$; (4) $|v_i^1| \le \delta$ for all $i \in S$.

The quantity δ represents the accuracy in the problem data. Let p be the number of nonzero transition probabilities $p_{ij}(a),\ (i,a,j) \in S \times A \times S$. Then, the *input size* of the problem, i.e. the number of binary bits needed to write down the discount factor α, the number of states N, the rewards $r_i(a),\ (i,a) \in S \times A$, the transition probabilities $p_{ij}(a),\ (i,a,j) \in S \times A \times S$ and the initial values $v_i^1,\ i \in S$, is at most some constant times L, where $L := p \cdot \log \delta$.

Lemma 9.14

If $\|v^n - v^\alpha\|_\infty \le \frac{1}{2\,\delta^{2N+2} \cdot N^N}$*, then the corresponding policy* f_n^∞ *is optimal.*

Proof

v^α is the unique solution of the linear system $\{\delta^2 I - (\delta\alpha)(\delta P(f_*)\}x = \delta^2 r(f_*)$, where f_*^∞ is an optimal policy. Note that all entries in this system are integer. Solving this system by Cramers rule gives $v_i^\alpha = \frac{w_i}{m}$, $i \in S$, with w_i, $i \in S$, and m integer and $m = det\,(Q)$, where $Q = \delta^2\{I - \alpha P(f_*)\}$. Notice that the entries of Q satisfy $|q_{ij}| \leq \delta^2$ for all i, j. Hence, by the definition of the determinant, $det(Q) = q_{11}q'_{11} + q_{12}q'_{12} + \cdots + q_{1N}q'_{1N}$, where $q'_{ij} = (-1)^{i+j} \cdot det(Q_{ij})$ with Q_{ij} the (i,j)-minor of Q, i.e. the submatrix of Q obtained by crossing out the ith row and j column. Therefore, we obtain the inequality $m \leq \delta^{2N} N^N$.

Consider a nonoptimal action a in state i, i.e. $r_i(a) + \alpha \sum_j p_{ij}(a) v_j^\alpha \neq v_i^\alpha$. Then, we can write

$$r_i(a) + \alpha \textstyle\sum_j p_{ij}(a) v_j^\alpha = \frac{\delta^2 m r_i(a) + \delta^2 \alpha \sum_j p_{ij}(a) w_j}{\delta^2 m} \neq v_i^\alpha = \frac{\delta^2 w_i}{\delta^2 m}.$$

Since the numerators $\delta^2 m r_i(a) + \delta^2 \alpha \sum_j p_{ij}(a) w_j$ and $\delta^2 w_i$ both are integers, it must be that $r_i(a) + \alpha \sum_j p_{ij}(a) v_j^\alpha$ and v_i^α differ by at least $\frac{1}{\delta^2 m} \geq \frac{1}{\delta^{2N+2} N^N}$.

Therefore, we have

$$\begin{array}{rcl} |r_i(a) + \alpha \sum_j p_{ij}(a) v_j^n - v_i^\alpha| & = & |\{r_i(a) + \alpha \sum_j p_{ij}(a) v_j^\alpha - v_i^\alpha\} - \{\alpha \sum_j p_{ij}(a)(v_j^\alpha - v_j^n)\}| \\ & \geq & |r_i(a) + \alpha \sum_j p_{ij}(a) v_j^\alpha - v_i^\alpha| - |\alpha \sum_j p_{ij}(a)(v_j^\alpha - v_j^n)|\} \\ & \geq & \frac{1}{\delta^{2N+2} N^N} - \alpha \|v^\alpha - v^n\|_\infty > \frac{1}{\delta^{2N+2} N^N} - \frac{1}{2\delta^{2N+2} N^N} \\ & = & \frac{1}{2\delta^{2N+2} N^N} \geq \|v^\alpha - v^n\|_\infty. \end{array}$$

Since $\|v^{n+1} - v^\alpha\|_\infty \leq \|v^n - v^\alpha\|_\infty$, action a is no action in f_n^∞. Hence, f_n^∞ contains only optimal actions, i.e. f_n^∞ is an optimal policy. □

Theorem 9.11

The value iteration method Algorithm 3.4 with stopping criterion $\|y - x\|_\infty \leq \frac{1}{2} \cdot \frac{1-\alpha}{\alpha} \cdot \varepsilon$ *computes an optimal policy in at most* n_3 *iterations, where* $n_3 := \frac{1}{|\log \alpha|} \cdot \log\left\{2\delta^{2N+2} N^N \cdot \{\|v^1\|_\infty + \frac{R}{1-\alpha}\}\right\}$ *with* $R := max_{(i,a)}\,|r_i(a)\|$.

Proof

Since $\|v^{n+1} - v^\alpha\|_\infty \leq \alpha^n \cdot \|v^1 - v^\alpha\|_\infty$, we have $\|v^{n+1} - v^\alpha\|_\infty \leq \varepsilon$ if $n \geq \frac{1}{|\log \alpha|} \cdot \log\left\{\frac{\|v^1 - v^\alpha\|_\infty}{\varepsilon}\right\}$.

If $\varepsilon \leq \frac{1}{2\delta^{2N+2} N^N}$, then $\|v^{n+1} - v^\alpha\|_\infty \leq \frac{1}{2\delta^{2N+2} N^N}$ if $n \geq \frac{1}{|\log \alpha|} \cdot \log\left\{2\delta^{2N+2} N^N \cdot \|v^1 - v^\alpha\|_\infty\right\}$.

Because $\|v^1 - v^\alpha\|_\infty \leq \|v^1\|_\infty + \|v^\alpha\|_\infty \leq \|v^1\|_\infty + \frac{R}{1-\alpha}$, the inequality $n \geq n_3$ implies

$n \geq \frac{1}{|\log \alpha|} \cdot \log\left\{2\delta^{2N+2} N^N \cdot \|v^1 - v^\alpha\|_\infty\right\}$ and consequently $\|v^{n+1} - v^\alpha\|_\infty \leq \frac{1}{2\delta^{2N+2} N^N}$.

Then, by Lemma 9.14, f_{n+1}^∞ is optimal, i.e. after at most n_3 iterations the algorithm terminates with an optimal policy. □

Because the discount factor α is considered as a constant and both $\|v^1\|_\infty \leq \delta$ and $R \leq \delta$, the number of iterations n_3 satisfies $n_3 = \mathcal{O}\big(N \cdot \log(\delta N)\big)$, which is polynomial in L. Furthermore, each iteration has at most $N \times M$ arithmetic operations. Therefore, under Assumption 9.1, the value iteration method is a polynomial-time algorithm for the computation of an optimal policy.

9.2 Additional constraints

9.2.1 Introduction

Formulating MDPs only in terms of the standard utility functions can be quite insufficient. Instead of introducing a single utility that has to be maximized (minimized) we often consider a situation where one type of profit (costs) has to be maximized (minimized) while keeping other types of rewards (costs) above (below) some given bounds. The first reference in this area is a paper of Derman and Klein ([70]). They consider an inventory problem for which the total costs are minimized under the constraint that the shortage is bounded by a given number. We will first present two examples of constrained problems from telecommunication.

Telecommunication networks are designed to enable simultaneous transmission of heterogeneous types of information. At the access to the network, or at the nodes within the network itself, the different types of traffic typically compete for a shared resource. Typical performance measures are the transmission delay, the throughput, probabilities of losses of packets, etc. Then, several constrained MDP problems can be considered, e.g.

(1) The maximization of the throughput, subject to constraints on its delay.

A tradeoff exists between achieving high throughput, on the one hand, and low expected delays on the other.

(2) Dynamic control of access of different traffic types.

In this model the problem is considered where several different traffic types compete for some resource; some weighted sum of average delays of some traffic is to be minimized, whereas for some other traffic types, a weighted sum of average delays should be bounded by some given limit.

For constrained Markov decision problems, for short CMDP, the nice property for the standard utility functions that there exists a deterministic optimal policy doesn't hold, in general. Even optimality simultaneously for all starting states is no longer valid. Therefore, we will optimize with respect to a given initial distribution β, i.e. β_j is the probability that state j is the starting state, $j \in S$. A special case is $\beta_j = \delta_{ij}$, i.e. that state i is the (fixed) starting state.

In many cases the reward or costs functions are specified in terms of expectations of some functions of the *state-action probabilities* $x_{ia}^R(t)$, defined for any policy R by

$$x_{ia}^{\beta,R}(t) := \sum_{j \in S} \beta_j \cdot \mathbb{P}_R\{X_t = i,\ Y_t = a \mid X_1 = j\},\ t = 1, 2, \dots . \tag{9.5}$$

9.2.2 Infinite horizon and discounted rewards

For the additional constraints we assume that, besides the immediate costs $r_i(a)$, there are for $k = 1, 2, \dots, m$ also certain immediate costs $c_i^k(a),\ (i, a) \in S \times A$. A policy R is called a *feasible* policy for a CMDP if the total expected discounted costs over the infinite horizon, denoted for the k-th cost function as $c_k^\alpha(\beta, R)$ and defined by $c_k^\alpha(\beta, R) := \sum_{t=1}^\infty \alpha^{t-1} \sum_{i,a} x_{ia}^{\beta,R}(t) c_i^k(a)$, is at most

$b_k,\ k = 1, 2, \dots, m$. An *optimal policy* R^* is a feasible policy that maximizes $v^\alpha(\beta, R)$, defined by $v^\alpha(\beta, R) := \sum_j \beta_j \cdot v^\alpha_j(R) = \sum_{t=1}^\infty \alpha^{t-1} \sum_{i,a} x^{\beta,R}_{ia}(t) r_i(a)$, over all feasible policies R, i.e.

$$v^\alpha(\beta, R^*) = sup_R \{v^\alpha(\beta, R) \mid c^\alpha_k(\beta, R) \leq b_k,\ k = 1, 2, \dots, m\}. \tag{9.6}$$

Define $x^\alpha_{ia}(\beta, R) := \sum_{t=1}^\infty \alpha^{t-1} x^{\beta,R}_{ia}(t),\ (i, a) \in S \times A$, as the *total discounted state-action frequencies*. Then, $v^\alpha(\beta, R) = \sum_{i,a} x^\alpha_{ia}(\beta, R) r_i(a)$ and $c^\alpha_k(\beta, R) = \sum_{i,a} x^\alpha_{ia}(\beta, R) c^k_i(a),\ k = 1, 2, \dots, m$.

Define the vector sets $K,\ K(M),\ K(S),\ K(D)$ and P, with components $(i, a) \in S \times A$ by

$$\begin{array}{rcl}
K & := & \{x^\alpha(\beta, R) \mid R \text{ is an arbitrary policy}\}; \\
K(M) & := & \{x^\alpha(\beta, R) \mid R \text{ is a Markov policy}\}; \\
K(S) & := & \{x^\alpha(\beta, R) \mid R \text{ is a stationary policy}\}; \\
K(D) & := & \{x^\alpha(\beta, R) \mid R \text{ is a deterministic policy}\}; \\
P & := & \left\{ x \;\middle|\; \begin{array}{rcl} \sum_{(i,a)} \{\delta_{ij} - \alpha p_{ij}(a)\} x_{ia} & = & \beta_j,\ j \in S \\ x_{ia} & \geq & 0,\ (i, a) \in S \times A \end{array} \right\}.
\end{array}$$

For any $|S \times A|$-vector $x \in P$, we define an $|S|$-vector, also denoted by x, by $x_i := \sum_a x_{ia},\ i \in S$. From the context it will be clear whether an $|S \times A|$-vector x or an $|S|$-vector x is meant.

Theorem 9.12
$K = K(M) = K(S) = \overline{K(D)} = P$, where $\overline{K(D)}$ is the closed convex hull of the finite set of vectors $K(D)$.

Proof
The equality $K = K(M)$ follows directly from Theorem 1.1. Furthermore, it is obvious that $K(D) \subseteq K(S) \subseteq K(M) \subseteq K$. We first show that $K \subseteq \overline{K(D)}$, then $K = K(M) = \overline{K(S)} = \overline{K(D)}$, where $\overline{K(S)}$ is the closed convex hull of the infinite set of vectors $K(S)$, and finally we show that $K(S) = P$, which implies - because P is a closed convex set - that $\overline{K(S)} = K(S)$.
For the proof of $K \subseteq \overline{K(D)}$, suppose the contrary. Then, there exists a policy R such that $x^\alpha(\beta, R) \in K$ and $x^\alpha(\beta, R) \notin \overline{K(D)}$. Since $\overline{K(D)}$ is a closed convex set, it follows from the Separating Hyperplane Theorem (see e.g. [155] pp.397–398) that there are coefficients $r_i(a),\ (i, a) \in S \times A$, such that

$$\sum_{i,a} x^\alpha_{ia}(\beta, R) r_i(a) > \sum_{i,a} x_{ia} r_i(a) \text{ for all } x \in \overline{K(D)}. \tag{9.7}$$

Consider the discounted MDP with immediate rewards $r_i(a),\ (i, a) \in S \times A$. We have seen in Chapter 3 that there exists an optimal policy $f^\infty \in C(D)$. Because $x^\alpha(\beta, R) \in K$, we can write

$$\textstyle\sum_{i,a} x^\alpha_{ia}(\beta, R) r_i(a) = v^\alpha(\beta, R) \leq v^\alpha(\beta, f^\infty) = \sum_{i,a} x^\alpha_{ia}(\beta, f^\infty) r_i(a),$$

which contradicts (9.7). Hence, we have shown that $K \subseteq \overline{K(D)}$, and consequently

$$K(D) \subseteq K(S) \subseteq K(M) = K \subseteq \overline{K(D)} \text{ and } \overline{K(S)} = \overline{K(D)}.$$

Next, we will show that $\overline{K(D)} \subseteq K(M)$, implying $K = K(M) = \overline{K(S)} = \overline{K(D)}$. Take any $x \in \overline{K(D)}$. Let $C(D) = \{f_1^\infty, f_2^\infty, \dots, f_n^\infty\}$. Then, $x_{ia} = \sum_{k=1}^n p_k\, x_{ia}^\alpha(\beta, f_k^\infty),\ (i,a) \in S \times A$ for certain $p_k \geq 0$ with $\sum_{k=1}^n p_k = 1$. By Theorem 1.1, there exists a policy $R \in C(M)$ satisfying

$$\textstyle\sum_{j\in S} \beta_j \cdot \mathbb{P}_R\{X_t = i,\ Y_t = a \mid X_1 = j\} = \sum_{j\in S} \beta_j \cdot \sum_{k=1}^n p_k\, \mathbb{P}_{f_k^\infty}\{X_t = i,\ Y_t = a \mid X_1 = j\},$$

for all $(i,a) \in S \times A$ and $t = 1, 2, \dots$. Hence,

$$\begin{array}{rcl} x_{ia} & = & \sum_{k=1}^n p_k\, x_{ia}^\alpha(\beta, f_k^\infty) \ = \ \sum_{k=1}^n p_k \sum_{t=1}^\infty \alpha^{t-1} x_{ia}^{\beta, f_k^\infty}(t) \ = \ \sum_{t=1}^\infty \alpha^{t-1} \sum_{k=1}^n p_k\, x_{ia}^{\beta, f_k^\infty}(t) \\ & = & \sum_{t=1}^\infty \alpha^{t-1} \sum_{k=1}^n p_k \sum_{j\in S} \beta_j \cdot \mathbb{P}_{f_k^\infty}\{X_t = i,\ Y_t = a \mid X_1 = j\} \\ & = & \sum_{t=1}^\infty \alpha^{t-1} \sum_{j\in S} \beta_j \cdot \mathbb{P}_R\{X_t = i,\ Y_t = a \mid X_1 = j\} \ = \ x_{ia}^\alpha(\beta, R),\ (i,a) \in S \times A. \end{array}$$

Therefore, $x = x^\alpha(\beta, R) \in K(M)$.

Finally, we we show that $K(S) = P$. For each $x \in P$, let $\pi^\infty \in C(S)$ be defined by

$$\pi_{ia} := \frac{x_{ia}}{x_i} \text{ if } x_i = \textstyle\sum_a x_{ia} > 0 \text{ and arbitrary if } x_i = 0. \tag{9.8}$$

Then, $\pi_{ia} x_i = x_{ia}$ for all $(i,a) \in S \times A$. Since $x \in P$, we can write

$$\begin{array}{rcl} \beta_j & = & \sum_a x_{ja} - \alpha \sum_{(i,a)} p_{ij}(a) x_{ia} \ = \ x_j - \alpha \sum_{(i,a)} p_{ij}(a) \pi_{ia} x_i \\ & = & x_j - \alpha \sum_i p_{ij}(\pi) x_i,\ j \in S, \end{array}$$

or, in vector notation, $\beta^T = x^T\{I - \alpha P(\pi)\}$, implying $x^T = \beta^T\{I - \alpha P(\pi)\}^{-1}$, i.e.

$$x_i = \textstyle\sum_{t=1}^\infty \alpha^{t-1} \sum_{j\in S} \beta_j \cdot \mathbb{P}_{\pi^\infty}\{X_t = i \mid X_1 = j\},\ i \in S.$$

Hence,

$$\begin{array}{rcl} x_{ia} & = & x_i \pi_{ia} \ = \ \sum_{t=1}^\infty \alpha^{t-1} \sum_{j\in S} \beta_j \cdot \mathbb{P}_{\pi^\infty}\{X_t = i,\ Y_t = a \mid X_1 = j\} \\ & = & x_{ia}^\alpha(\beta, \pi^\infty),\ (i,a) \in S \times A. \end{array}$$

showing $P \subseteq K(S)$. Conversely, take any $x^\alpha(\beta, \pi^\infty) \in K(S)$. Then,

$x_{ia}^\alpha(\beta, \pi^\infty) = \sum_{t=1}^\infty \alpha^{t-1} \sum_{j\in S} \beta_j \cdot \{P(\pi)^{t-1}\}_i \cdot \pi_{ia} = \big\{\beta^T \cdot \{\sum_{t=1}^\infty \{\alpha P(\pi)\}^{t-1}\big\}_i \cdot \pi_{ia} \geq 0$, for all $(i,a) \in S \times A$, which can be written as

$$x_{ia}^\alpha(\beta, \pi^\infty) = \big\{\beta^T \cdot \{I - \alpha P(\pi)\}^{-1}\big\}_i \cdot \pi_{ia} \text{ or } x_{ia}^\alpha(\beta, \pi^\infty) = x_i^\alpha(\beta, \pi^\infty) \cdot \pi_{ia},$$

where $x_i^\alpha(\beta, \pi^\infty) := \big\{\beta^T \cdot \{I - \alpha P(\pi)\}^{-1}\big\}_i$. From this expression it follows that

$$\begin{array}{rcl} \sum_{(i,a)} \{\delta_{ij} - \alpha p_{ij}(a)\} x_{ia}^\alpha(\beta, \pi^\infty) & = & x_j^\alpha(\beta, \pi^\infty) - \alpha \sum_i \{\sum_a p_{ij}(a) \pi_{ia}\} x_i^\alpha(\beta, \pi^\infty) \\ & = & x_j^\alpha(\beta, \pi^\infty) - \alpha \sum_i p_{ij}(\pi) x_i^\alpha(\beta, \pi^\infty) \ = \ \big\{\big(x^\alpha(\beta, \pi)\big)^T \{I - \alpha P(\pi)\}\big\}_j \\ & = & \big\{\beta^T \{I - \alpha P(\pi)\}^{-1} \{I - \alpha P(\pi)\}\big\}_j = \ \beta_j,\ j \in S. \end{array}$$

Hence, $x^\alpha(\beta, \pi^\infty) \in P$, completing the proof that $P = K(S)$. □

In order to solve the CMDP (9.6), we consider the following linear program

$$max \left\{ \sum_{(i,a)} r_i(a) x_i(a) \;\middle|\; \begin{array}{rcl} \sum_{(i,a)} \{\delta_{ij} - \alpha p_{ij}(a)\} x_i(a) & = & \beta_j,\ j \in S \\ \sum_{(i,a)} c_i^k(a) x_i(a) & \leq & b_k,\ k = 1, 2, \dots, m \\ x_i(a) & \geq & 0,\ (i,a) \in S \times A \end{array} \right\} \tag{9.9}$$

Theorem 9.13

(1) *The linear program (9.9) is infeasible if and only if the CMDP (9.6) is infeasible.*

(2) *If x is an optimal solution of program (9.9), then π^∞, defined by (9.8), is a stationary optimal policy for the CMDP (9.6).*

Proof

(1) Assume that the linear program (9.9) is infeasible and that the CMDP (9.6) is feasible, i.e. there exists a policy R satisfying $c_k^\alpha(\beta, R) = \sum_{i,a} x_{ia}^\alpha(\beta, R) c_i^k(a) \leq b_k,\ k = 1, 2, \ldots, m$. Since $K = P$, there exists an $x \in P$ with $x = x(\beta, R)$. Hence x is a feasible solution of the linear program (9.9), which yields a contradiction. The reverse statement can be shown in a similar way.

(2) Let x be an optimal solution of program (9.9) and let π^∞ be defined by (9.8). Then, π^∞ is a feasible solution of the CMDP (9.6) with $v^\alpha(\beta, \pi^\infty) = \sum_{i,a} x_{ia}^\alpha(\beta, \pi^\infty) r_i(a) = \sum_{i,a} x_{ia} r_i(a)$ as value of the objective function. Let R_* be an arbitrary feasible solution of (9.6). Then, $x^\alpha(\beta, R_*)$ is a feasible solution of (9.9) with

$$v^\alpha(\beta, R_*) = \textstyle\sum_{i,a} x_{ia}^\alpha(\beta, R_*) r_i(a) \leq \sum_{i,a} x_{ia} r_i(a) = v^\alpha(\beta, \pi^\infty), \text{ i.e.}$$

π^∞ is an optimal policy of the CMDP (9.6). □

Remarks

1. If $\beta_j > 0,\ j \in S$, then it is shown in Theorem 3.18 that the mapping $\pi_{ia}^x := \frac{x_{ia}}{\sum_a x_{ia}}$, $(i, a) \in S \times A$, is a bijection between P and $K(S)$ with as inverse mapping x^π, defined by $x_i^\pi(a)(\pi) := \left\{\beta^T \cdot \{I - \alpha P(\pi)\}^{-1}\right\}_i \cdot \pi_{ia},\ (i, a) \in S \times A$. Furthermore, the extreme points of P correspond to the deterministic policies of $C(D)$.
2. If the linear program (9.9) is feasible, an extreme optimal solution has at most $N + m$, i.e. the number of the constraints in (9.9)), strictly positive variables. Hence, there exists an optimal stationary policy with in at most m states a randomization.

Algorithm 9.2 *Construction of an optimal stationary policy for the CMDP (9.6)*

Input: Instance of an MDP, an initial distribution β, immediate costs $c_i^k(a),\ (i, a) \in S \times A$ and bounds b_k for $k = 1, 2, \ldots, m$.

Output: Either the statement that problem (9.6) is infeasible or an optimal stationary policy π^∞ of problem (9.6).

1. Determine an optimal policy x for linear program (9.9).
2. **if** program (9.9) is infeasible **then** problem (9.6) is infeasible

   else the stationary policy π^∞, defined by $\pi_{ia} := \frac{x_i(a)}{\sum_a x_i(a)}$ if $\sum_a x_i(a) > 0$ and arbitrarily if $\sum_a x_i(a) = 0$ is an optimal stationary policy for problem (9.6).

Monotone optimal policies

Consider the constrained MDP problem (9.6) with $S = \{1, 2, \dots, N\},\ A(i) = \{1, 2, \dots, M\}, i \in S$, where S and A are ordered in the usual way, and under the conditions $B1, B2, B3$ and $B4$ of Assumption 3.2 from section 3.9, i.e.

(B1) $r_i(a)$ is nonincreasing in i for all a;

(B2) $\sum_{j=k}^N p_{ij}(a)$ is nondecreasing in i for all k and a.

(B3) $r_i(a)$ is supermodular on $S \times A$;

(B4) $\sum_{j=k}^N p_{ij}(a)$ is submodular on $S \times A$ for all k.

Furthermore, we impose the following additional assumptions:

(B5) $c_i^k(a)$ is nonincreasing in i for all a and all k;

(B6) $c_i^k(a)$ is submodular on $S \times A$ for all k.

We have already seen that problem (9.6) has an optimal stationary policy with in at most m states a randomization and that such optimal policy can be obtained from linear program (9.9). The dual program of (9.9) is:

$$min \left\{ \sum_{j=1}^N \beta_j v_j + \sum_{k=1}^m \lambda_k b_k \;\middle|\; \begin{array}{rcll} \sum_{j=1}^N \{\delta_{ij} - \alpha p_{ij}(a)\} v_j + \sum_{k=1}^m c_i^k(a)\lambda_k & \geq & r_i(a), & (i,a) \in S \times A \\ \lambda_k & \geq & 0, & 1 \leq k \leq m \end{array} \right\} \tag{9.10}$$

From the theory of linear programming we know that a necessary and sufficient condition for optimality of a feasible solution x^* of (9.9) is the existence of (v^*, λ^*) such that:

(1) (v^*, λ^*) is feasible for (9.9);

(2) $x_i^*(a) \cdot \{r_i^*(a) - \sum_{j=1}^N \{\delta_{ij} - \alpha p_{ij}(a)\} v_j^*\} = 0$ for all $(i, a) \in S \times A$, where
$r_i^*(a) := r_i(a) - \sum_{k=1}^m c_i^k(a)\lambda_k^*$;

(3) $\lambda_k^* \cdot \{\sum_{(i,a)} c_i^k(a) - b_k\} = 0$ for $k = 1, 2, \dots, m$.

Since $\lambda_k^* \geq 0$ for $1 \leq k \leq m$, it follows from $B1, B5, B3$ and $B6$ that $r_i^*(a)$ is nonincreasing in i for all a and is supermodular on $S \times A$.

A stationary policy π^∞ is called a *randomized nondecreasing policy* if for every $1 \leq i \leq N-1$, the following property holds: if $\pi_{ia} = 0$ for all $1 \leq a \leq b$, then also $\pi_{i+1,a} = 0$ for all $1 \leq a \leq b$. Hence, the sets of actions which are used for π are nonincreasing in the state. The next theorem shows that, under the assumptions $B1$ until $B6$, the constrained MDP problem (9.6) has an optimal randomized nondecreasing stationary policy.

Theorem 9.14

Consider a feasible constrained MDP problem (9.6) for which the assumptions $B1$ until $B6$ hold. Then, this problem has a randomized nondecreasing stationary optimal policy.

Proof

Let x^* be an optimal solution of (9.9) with (v^*, λ^*) as corresponding optimal solution of (9.10). From the optimality properties (1) and (2) it follows that x^* and v^* are optimal solutions of the unconstrained problem with rewards $r^*_i(a)$ instead of $r_i(a)$, $(i, a) \in S \times A$. Note that this unconstrained problem satisfies $B1$ until $B4$. From the proof of Theorem 3.36, we know that $s^*_i(a) := r^*_i(a) + \alpha \cdot \sum_{j=1}^N p_{ij}(a)v^*_j$ is supermodular on $S \times A$. Hence, for all $1 \leq i \leq N - 1$ and all $1 \leq a \leq b \leq M$, we have $s^*_i(a) + s^*_{i+1}(b) \geq s^*_i(b) + s^*_{i+1}(a)$, i.e.

$$s^*_i(b) - s^*_i(a) \geq s^*_{i+1}(b) - s^*_{i+1}(a) \text{ for all } 1 \leq i \leq N - 1 \text{ and all } 1 \leq a \leq b \leq M. \tag{9.11}$$

Let b be the smallest optimal action in state i for the unconstrained problem, i.e. we have $v^*_i = s^*_i(b) > s^*_i(a)$ for all $1 \leq a < b$. Then, it follows from (9.11) that $s^*_{i+1}(b) > s^*_{i+1}(a)$, implying $0 \geq s^*_{i+1}(b) - v^*_{i+1} > s^*_{i+1}(a) - s^*_{i+1}$. From the orthogonality property (2) it follows that $x^*_{i+1}(a) = 0$ for all $1 \leq a < b$. This completes the proof of the existence of a randomized nondecreasing stationary policy. □

Example 9.6

Consider the model of Example 3.7 for which we add the constraint $x_6(1) + x_7(1) + x_8(1) \leq 0.4$. Notice that this constraint satisfies $B5$ and $B6$. We assume that we start with a new item, i.e., in state 1 and let $\beta_1 = 1$, $\beta_i = 0$ for $i = 2, 3, \dots, 8$. The corresponding linear program is:

$max\{-x_7(1) - 5x_8(1) - 2x_1(2) - 2x_2(2) - 2x_3(2) - 2x_4(2) - 2x_5(2) - 2x_6(2) - 2x_7(2) - 2x_8(2)\}$

subject to the constraints

$x_1(1) + x_1(2) = 1 + 0.9 \cdot \{0.03x_1(1) + x_1(2) + x_2(2) + x_3(2) + x_4(2) + x_5(2) + x_6(2) + x_7(2) + x_8(2)\}$;

$x_2(1) + x_2(2) = 0 + 0.9 \cdot \{0.07x_1(1) + 0.02x_2(1)\}$;

$x_3(1) + x_3(2) = 0 + 0.9 \cdot \{0.05x_1(1) + 0.03x_2(1) + 0.05x_3(1)\}$;

$x_4(1) + x_4(2) = 0 + 0.9 \cdot \{0.1x_1(1) + 0.1x_2(1) + 0.05x_3(1) + 0.05x_4(1)\}$;

$x_5(1) + x_5(2) = 0 + 0.9 \cdot \{0.1x_1(1) + 0.1x_2(1) + 0.1x_3(1) + 0.05x_4(1) + 0.02x_5(1)\}$;

$x_6(1) + x_6(2) = 0 + 0.9 \cdot \{0.2x_1(1) + 0.2x_2(1) + 0.1x_3(1) + 0.1x_4(1) + 0.08x_5(1) + 0.05x_6(1)\}$;

$x_7(1) + x_7(2) = 0 + 0.9 \cdot \{0.2x_1(1) + 0.2x_2(1) + 0.2x_3(1) + 0.2x_4(1) + 0.1x_5(1) + 0.1x_6(1) + 0.1x_7(1)\}$;

$x_8(1) + x_8(2) = 0 + 0.9 \cdot \{0.25x_1(1) + 0.35x_2(1) + 0.5x_3(1) + 0.6x_4(1) + 0.8x_5(1) + 0.85x_6(1) + 0.9x_7(1) + x_8(1)\}$;

$x_6(1) + x_7(1) + x_8(1) \leq 0.4$;

$x_1(1), x_2(1), x_3(1), x_4(1), x_5(1), x_6(1), x_7(1), x_8(1), x_1(2), x_2(2), x_3(2), x_4(2), x_5(2), x_6(2), x_7(2), x_8(2) \geq 0.$

An optimal solution of this program and the corresponding policy is presented in the next table:

i	$x_i(1)$	$x_i(2)$	π_{i1}	π_{i1}	i	$x_i(1)$	$x_i(2)$	π_{i1}	π_{i1}
$i = 1$	4.4564	0	1	0	$i = 5$	0.4756	0	1	0
$i = 2$	0.2859	0	1	0	$i = 6$	0.4000	0.5666	0.4138	0.5862
$i = 3$	0.2181	0	1	0	$i = 7$	0	1.0540	0	1
$i = 4$	0.4572	0	1	0	$i = 8$	0	2.0862	0	1

Notice that the only difference with the optimal policy for the unconstrained model is that in state 6 there is randomization. Next, we add a second constraint $x_5(1)+x_6(1)+2x_7(1)+4x_8(1) \leq 0.6$, which also satisfies $B5$ and $B6$. An optimal solution of this program and the corresponding policy is presented in the following table:

i	$x_i(1)$	$x_i(2)$	π_{i1}	π_{i1}	i	$x_i(1)$	$x_i(2)$	π_{i1}	π_{i1}
$i=1$	4.5764	0	1	0	$i=5$	0.2000	0.2832	0.4139	0,5861
$i=2$	0.2936	0	1	0	$i=6$	0.4000	0.5714	0.4118	0.5882
$i=3$	0.2239	0	1	0	$i=7$	0	1.0554	0	1
$i=4$	0.4695	0	1	0	$i=8$	0	1.9265	0	1

We see that in this case in two states (state 5 and again state 6) there is randomization.

The structure of the value function

We have seen in Corollary 5.2 that in the unconstrained case the value vector v^α is a continuous, piecewise rational function in α with no singular point in the interval $[0,1)$. We will show the same property for constrained discounted MDPs. If the discount factor is not fixed but varies over the whole interval $[0,1)$, the formulation

$$sup_R \, \{v^\alpha(\beta,R) \mid c_k^\alpha(\beta,R) \leq b_k, \; k=1,2,\dots,m\}. \tag{9.12}$$

is less appropriate, because in general the functions $v^\alpha(\beta,R)$ and $c_k^\alpha(\beta,R), \; 1 \leq k \leq m$ tend to infinity if α tends to 1. Therefore, we consider the scaled version

$$sup_R \, \{(1-\alpha)v^\alpha(\beta,R) \mid (1-\alpha)c_k^\alpha(\beta,R) \leq d_k, \; k=1,2,\dots,m\}, \tag{9.13}$$

which is an equivalent problem for any fixed α. Note that this transformation is invariant for the property of continuous, piecewise rational function in a with no singular point in the interval $[0,1)$. The optimum value of (9.12) is called the *value* $v^\alpha(\beta)$ of the discounted constrained Markov decision problem, and the optimum value of (9.13)is *scaled value* $w^\alpha(\beta)$.

<u>Remark</u>

Since $(1-\alpha)v^\alpha(\beta,R) = \lim_{T\rightarrow\infty} \frac{\mathbb{E}_{\beta,R}\{\sum_{t=1}^T \alpha^{t-1} r_{X_t}(Y_t)\}}{\mathbb{E}_{\beta,R}\{\sum_{t=1}^T \alpha^{t-1}\}}$, and a similar expression holds for the cost functions $(1-\alpha)c_k^\alpha(\beta,R) \leq b_k, \; k=1,2,\dots,m$, the scaled versions of $v^\alpha(\beta,R)$ and $c_k^\alpha(\beta,R)$ may be considered as the total discounted rewards (or costs) per total discounted time, so as a discounted time average. We have already seen that we may restrict the policy space of a discounted constrained Markov decision problem to the set of stationary policies. For any stationary policy π^∞, we have $\lim_{\alpha\uparrow 1}(1-\alpha)v^\alpha(\beta,\pi^\infty) = \phi(\beta,\pi^\infty)$ (the proof is similar to the proof of Theorem 5.8, part (2)). So as the discount factor tends to 1, a scaled discounted constrained Markov decision problem converges to an undiscounted constrained Markov decision problem.

From the analysis in the first part of this section it follows that problem (9.13) can be solved by the following linear program:

$$max \left\{ \sum_{(i,a)} (1-\alpha)r_i(a)x_i(a) \;\middle|\; \begin{array}{rcl} \sum_{(i,a)} \{\delta_{ij} - \alpha p_{ij}(a)\}x_i(a) & = & \beta_j, \; j \in S \\ \sum_{(i,a)} (1-\alpha)c_i^k(a)x_i(a) & \leq & d_k, \; k=1,2,\dots,m \\ x_i(a) & \geq & 0, \; (i,a) \in S \times A \end{array} \right\} \tag{9.14}$$

Using the interest rate $\rho := \frac{1-\alpha}{\alpha}$, i.e., $(1-\alpha) = \alpha$, and the variables $y_i(a) = \alpha y_i(a)$ for all $(i,a) \in S \times A$, we obtain the following equivalent formulation:

$$max \left\{ \sum_{(i,a)} \{\rho r_i(a)\} y_i(a) \;\middle|\; \begin{array}{rcl} \sum_{(i,a)} \{(1+\rho)\delta_{ij} - p_{ij}(a)\} y_i(a) & = & \beta_j, \ j \in S \\ \sum_{(i,a)} \{\rho c_i^k(a)\} y_i(a) & \leq & d_k, \ k = 1, 2, \dots, m \\ y_i(a) & \geq & 0, \ (i,a) \in S \times A \end{array} \right\} \quad (9.15)$$

If this problem is solved with the phase I - phase II technique (see e.g. [341]), then in phase I the following linear program with artificial variables $z_j,\ j \in S$, and slack variables $s_k,\ 1 \leq k \leq m$, is considered:

$$max \left\{ z_0 \;\middle|\; \begin{array}{rcrcl} \sum_{(i,a)} \{(1+\rho)\delta_{ij} - p_{ij}(a)\} y_i(a) & + & z_j & = & \beta_j, \ j \in S \\ \sum_{(i,a)} \{\rho c_i^k(a)\} y_i(a) & + & s_k & = & d_k, \ k = 1, 2, \dots, m \\ \sum_j z_j & + & z_0 & = & 0 \\ \multicolumn{5}{l}{y_i(a) \geq 0, \ (i,a) \in S \times A; \ z_j \geq 0, \ j \in S; \ s_k \geq 0, \ 1 \leq k \leq m} \end{array} \right\} \quad (9.16)$$

For program (9.16) an initial feasible basis is available, namely the basic variables $z_j,\ j \in S$, $s_k,\ 1 \leq k \leq m$, and z_0. Then, for the optimal solution (y^*, z^*, s^*) of program (9.16) there are two possibilities:

a. $z^* < 0$: in this case program (9.15) has no feasible solution;

b. $z^* = 0$: in this case y^* is a feasible basis solution of (9.15) and phase II can be started, in which the original objective function will be optimized.

We will analyze problem (9.15) in the same way as we analyzed in Section 7.7 linear program (7.36). We make the following observations about the programs (9.15) and (9.16):

1. There are only a finite number of different bases for the linear program 9.15.
2. For each of these bases the corresponding simplex tableau has the following properties:
   a. The elements are rational functions in ρ.
   b. The tableau is optimal if and only if both the basic variables and the dual variables are nonnegative. Each variable is rational function in ρ; so, it is nonnegative on a finite number of closed intervals. Hence, each basis can be optimal on only a finite number of intervals, which are closed intervals of the interest rate.
   c. Infeasibility of problem (9.15) corresponds to an optimal tableau in phase I with $z^* < 0$, which corresponds to open intervals.
   d. Feasibility and optimality of problem (9.15) corresponds to an optimal tableau in phase II, which corresponds to closed intervals.

Combining the above observations and using the property that any rational function in ρ is also a rational function in α, we obtain the following result.

Theorem 9.15

There exist numbers $0 = \alpha_0 < \alpha_1 < \cdots < \alpha_{p-1} < \alpha_p = 1$ such that:

(1) For every $j = 1, 2, \dots, p$ either problem (9.13) is infeasible for all $\alpha \in [\alpha_{j-1}, \alpha_j)$ or there is a stationary policy $\pi^\infty(j)$ that is optimal for all $\alpha \in [\alpha_{j-1}, \alpha_j)$.

(2) The policy $\pi^\infty(j)$ is a rational function in α and corresponds on the interval $[\alpha_{j-1}, \alpha_j)$ to a fixed set of basic variables.

(3) When problem (9.13) is feasible on $[\alpha_{j-1}, \alpha_j)$, then the value is also a rational function in α on that interval.

Example 5.4 (continued)

Take $\beta_1 = \beta_2 = 0.5$ and add one constraint: $(1-\alpha)c_1^\alpha(\beta, R) \le 1$, where $c_1^\alpha(\beta, R)$ is the total discounted cost for an MDP with one-step costs $c_1(1) = c_1(2) = c_1(3) = 4,\ c_2(1) = 0$.

Simple calculations give for the three deterministic policies: $c_1^\alpha(\beta, f_1^\infty) = 2$, $c_1^\alpha(\beta, f_2^\infty) = \frac{2}{1-\alpha}$ and $c_1^\alpha(\beta, f_3^\infty) = \frac{2}{1-0.5\alpha}$. Hence, f_1^∞ is feasible for $\frac{1}{2} \le \alpha < 1$, f_2^∞ is feasible for all $\alpha \in [0, 1)$ and f_3^∞ is feasible for $\frac{2}{3} \le \alpha < 1$. The linear program (9.15) becomes:

$$max \left\{ \begin{array}{l|rcrcrcrcl} \rho y_1(1)+ & (1+\rho y_1(1) & + & \rho y_1(2) & + & (\frac{1}{2}+\rho y_1(3) & & & = & \frac{1}{2} \\ \frac{1}{2}\rho y_1(2)+ & -y_1(1) & & & - & \frac{1}{2}y_1(3) & + & \rho y_2(1) & = & \frac{1}{2} \\ \frac{3}{4}\rho y_1(3)+ & 4\rho y_1(1) & + & 4\rho y_1(2) & + & 4\rho y_1(3) & & & \le & 1 \\ & & & & & y_1(1),\ y_1(2),\ y_1(3), & & y_2(1) & \ge & 0 \end{array} \right\}.$$

The linear program (9.16) for phase I of the simplex method is:

$$max \left\{ z_0 \left| \begin{array}{rcrcrcrcrcl} (1+\rho y_1(1) & + & \rho y_1(2) & + & (\frac{1}{2}+\rho y_1(3) & & & + & z_1 & = & \frac{1}{2} \\ -y_1(1) & & & - & \frac{1}{2}y_1(3) & + & \rho y_2(1) & + & z_2 & = & \frac{1}{2} \\ 4\rho y_1(1) & + & 4\rho y_1(2) & + & 4\rho y_1(3) & & & + & s_1 & = & 1 \\ & & & & z_0 & + & z_1 & + & z_2 & = & 0 \\ & & y_1(1),\ y_1(2), & & y_1(3),\ y_2(1), z_1, & & z_2, & & s_1 & \ge & 0 \end{array} \right. \right\}.$$

The corresponding first simplex tableau is (this tableau is similar to the simplex tableau in Example 7.4; we also add the original objective function as last equation in the tableaus):

	1	$y_1(1)$	$y_1(2)$	$y_1(3)$	$y_2(1)$
z_1	$\frac{1}{2}$	$1+\rho$	ρ	$\frac{1}{2}+\rho$	0
z_2	$\frac{1}{2}$	-1	0	$-\frac{1}{2}$	ρ
s_1	1	4ρ	4ρ	4ρ	0
z_0	-1	$-\rho$	$-\rho$	$-\rho$	$-\rho$
	0	$-\rho$	$-\frac{1}{2}\rho$	$-\frac{3}{4}\rho$	0

In the first iteration the pivot column is the column of the variable $y_1(1)$ and the pivot row is the row of z_1. The next tableau becomes (with common denominator $1+\rho$). In this way we obtain the next tableau.

	$1+\rho$	z_1	$y_1(2)$	$y_1(3)$	$y_2(1)$
$y_1(1)$	$\frac{1}{2}$	1	ρ	$\frac{1}{2}+\rho$	0
z_2	$1+\frac{1}{2}\rho$	1	ρ	$\frac{1}{2}\rho$	$\rho+\rho^2$
s_1	$1-\rho$	-4ρ	4ρ	2ρ	0
z_0	$-1-\frac{1}{2}\rho$	ρ	$-\rho$	$-\frac{1}{2}\rho$	$-\rho-\rho^2$
	$\frac{1}{2}\rho$	ρ	$-\frac{1}{2}\rho+\frac{1}{2}\rho^2$	$-\frac{1}{4}\rho+\frac{1}{4}\rho^2$	0

In the second iteration we exchange the variables $y_2(1)$ and z_2. Then the common denominator is $\rho+\rho^2$). In this way we obtain the following tableau.

	$\rho+\rho^2$	z_1	$y_1(2)$	$y_1(3)$	z_2
$y_1(1)$	$\frac{1}{2}\rho$	ρ	ρ^2	$\frac{1}{2}\rho+\rho^2$	0
$y_2(1)$	$1+\frac{1}{2}\rho$	1	ρ	$\frac{1}{2}\rho$	$1+\rho$
s_1	$\rho-\rho^2$	$-4\rho^2$	$4\rho^2$	$2\rho^2$	0
z_0	0	$\rho+\rho^2$	0	0	$\rho+\rho^2$
	$\frac{1}{2}\rho^2$	ρ^2	$-\frac{1}{2}\rho^2+\frac{1}{2}\rho^3$	$-\frac{1}{4}\rho^2+\frac{1}{4}\rho^3$	0

This tableau corresponds with the deterministic policy f_1^∞ and is feasible if $\rho-\rho^2 \geq 0$, i.e. $\rho \in (0,1]$ or $\alpha \in [\frac{1}{2},1)$. Policy f_1^∞ is optimal if also $-\frac{1}{2}\rho^2+\frac{1}{2}\rho^3 \geq 0$ and $-\frac{1}{4}\rho^2+\frac{1}{4}\rho^3 \geq 0$, i.e. if $\rho \geq 1$. Hence only for $\rho = 1$, i.e. $\alpha = \frac{1}{2}$ policy f f_1^∞ is feasible and optimal.

For $\rho < 1$, we have to execute a pivot operation. Since phase I of the simplex method is finished, the z-variables can be removed from the tableau and we continue with phase II. Take the column of $y_1(2)$ as pivot column and the row of s_1 as pivot row. This yields the following tableau.

	$4\rho^2$	s_1	$y_1(3)$
$y_1(1)$	ρ^2	$-\rho^2$	$2\rho^2$
$y_2(1)$	3ρ	ρ	0
$y_1(2)$	$\rho-\rho^2$	$-4\rho^2$	$2\rho^2$
	$\frac{1}{2}\rho^2+\frac{1}{2}\rho^3$	$\frac{1}{2}\rho^2-\frac{1}{2}\rho^3$	0

This tableau is feasible and optimal if $\rho-\rho^2 \geq 0$ and $\frac{1}{2}\rho^2-\frac{1}{2}\rho^3 \geq 0$, i.e. if $\rho \in (0,1]$, which equivalent to $\alpha \in (\frac{1}{2},1]$. Hence, the constrained problem is feasible and has an optimal stationary policy π^∞ if $\rho \in (0,1]$. The optimal stationary policy π^∞, as function of ρ, is

$$\pi_{11} = \frac{y_1(1)}{y_1(1)+y_1(2)+y_1(3)} = \rho;\ \pi_{12} = \frac{y_1(2)}{y_1(1)+y_1(2)+y_1(3)} = 1-\rho;\ \pi_{13} = \frac{y_1(3)}{y_1(1)+y_1(2)+y_1(3)} = 0.$$

Furthermore, we obtain from this final simplex tableau, because s_1 is a nonbasic variable, that $(1-\alpha)c_1(\beta,\pi^\infty) = 1$ for all $\alpha \in (\frac{1}{2},1]$.

Remarks

1. If problem (9.13) is feasible in the neighborhood of $\alpha = 1$, i.e. there exists a stationary policy $\pi^\infty(p)$ such that $(1-\alpha)c_1(\beta,\pi^\infty) \leq d_k$ for $k = 1,2,\dots,m$, then the value $w^\alpha(\beta)$ satisfies $w^\alpha(\beta) = (1-\alpha)v^\alpha\bigl(\beta,\pi^\infty(p)\bigr)$ for $\alpha_{p-1} \leq \alpha < \alpha_p = 1$. Note that $(1-\alpha)v^\alpha\bigl(\beta,\pi^\infty(p)\bigr) \leq M$, where $M := max_{(i,a)}\, r_i(a)$ and furthermore $(1-\alpha)v^\alpha\bigl(\beta,\pi^\infty(p)\bigr) = \sum_{(i,a)} \{\rho r_i(a)\} y_i^*(a)$, where

y^* is the optimal solution of (9.15) on the interval $[\alpha_{p-1}, 1)$ with components $y_i^*(a)$ which are rational functions in ρ with no singular points for $\rho \in (0, \infty)$. Hence, $w^\alpha(\beta)$ can be expressed as a power series in $(1-\alpha)$, i.e., $w^\alpha(\beta) = \sum_{k=0}^\infty a_k(1-\alpha)^k$ for some real numbers $a_0, a_1, \dots$. Therefore, $v^\alpha(\beta) = (1-\alpha)^{-1}w^\alpha(\beta)$ has a Laurent series expansion in the neighborhood of $\alpha = 1$.

2. Theorem 5.10 shows that $v^\rho(\pi^\infty) = (1+\rho)\{\rho^{-1}P^*(\pi)r(\pi) + \sum_{k=0}^\infty (-\rho)^k)\{D(pi)\}^{k+1}r(\pi)\}$ for $0 < \rho < \|D(\pi)\|^{-1}$ and for any stationary policy π^∞. This expression is called the Laurent expansion about the origin $\rho = 0$ (actually, in Theorem 5.10 this result is shown for deterministic policies, but the proof is similar for stationary policies). Since the randomizations may depend on ρ, it is not obvious that $v^\rho(\beta, \pi^\infty(p)$ has a similar Laurent expansion, although it has a Laurent expansion of another form. The next example shows this phenomenon.

Example 5.4 (continued)
We have $w^\rho(\beta) = \frac{\frac{1}{2}\rho^2+\frac{1}{2}\rho^3}{4\rho^2} = \frac{1}{8}(1+\rho)$ for all $\rho \in (0, 1]$. As function of the discount factor α, we have $w^\alpha(\beta) = \frac{1}{8} \cdot \frac{1}{\alpha} = \frac{1}{8}\sum_{k=0}^\infty (1-\alpha)^k$ for all $\alpha \in [\frac{1}{2}, 1)$. As Laurent expansion for the value function $v^\alpha(\beta)$ we obtain $v^\alpha(\beta) = (1-\alpha)^{-1}w^\alpha(\beta) = \frac{1}{8}\sum_{k=-1}^\infty (1-\alpha)^k$ for all $\alpha \in [\frac{1}{2}, 1)$. For the optimal stationary policy π^∞, we can write

$$P(\pi) = \left(\begin{smallmatrix} 1-\rho & \rho \\ 0 & 0 \end{smallmatrix}\right);\ P^*(\pi) = \left(\begin{smallmatrix} 0 & 1 \\ 0 & 1 \end{smallmatrix}\right);\ D(\pi) = \rho^{-1}\left(\begin{smallmatrix} 1 & -1 \\ 0 & 0 \end{smallmatrix}\right);\ D^k(\pi) = \rho^{-1}\left(\begin{smallmatrix} 1 & -1 \\ 0 & 0 \end{smallmatrix}\right).$$

For the Laurent expansion of $v^\rho(\pi^\infty)$, as given in the above Remark 2, we need $0 < \rho < \|D(\pi)\|^{-1}$. Since $\|D(\pi)\| = \rho^{-1}$, this expansion is not valid.

9.2.3 Infinite horizon and total rewards

For this section we have the assumption that the model is substochastic, i.e. $\sum_j p_{ij}(a) \le 1$ for all $(i, a) \in S \times A$. Given an initial distribution β, let the total expected reward and the total expected costs for a transient policy R be denoted by $v(\beta, R)$ and $c^k(\beta, R)$, i.e.

$$v(\beta, R) := \sum_{t=1}^\infty \sum_i \beta_i \cdot \sum_{(j,a)} \mathbb{P}_R\{X_t = j,\ Y_t = a \mid X_1 = i\} \cdot r_j(a);$$

$$c_k(\beta, R) := \sum_{t=1}^\infty \sum_i \beta_i \cdot \sum_{(j,a)} \mathbb{P}_R\{X_t = j,\ Y_t = a \mid X_1 = i\} \cdot c_j^k(a).$$

Notice that $v(\beta, R)$ and $c^k(\beta, R)$ are well defined and finite for any transient policy R. The constrained problem for transient policies, given some real numbers $b_k,\ 1 \le k \le m$, is defined as:

$$sup_{R \text{ transient}} \{v(\beta, R) \mid c_k(\beta, R) \le b_k,\ k = 1, 2, \dots, m\}. \tag{9.17}$$

Given an initial distribution β and a transient policy R, we define $x_{ia}(\beta, R)$, the total expected state-action frequencies of $(i, a) \in S \times A$ by $x_{ia}(\beta, R) := \sum_{t=1}^\infty x_{ia}^{\beta,R}(t)$, where $x_{ia}^{\beta,R}(t)$ is defined in (9.5). Then, for any transient policy R,

$$v(\beta, R) = \sum_{i,a} x_{ia}(\beta, R)r_i(a) \text{ and } c_k(\beta, R) = \sum_{i,a} x_{ia}(\beta, R)c_i^k(a) \text{ for } k = 1, 2, \dots, m.$$

Let the vector sets $K, K(M), K(S), K(D)$ and P, with components $(i,a) \in S \times A$, be defined by:

$$\begin{array}{rcl} K & := & \{x(\beta,R) \mid R \text{ is an arbitrary transient policy}\}; \\ K(M) & := & \{x(\beta,R) \mid R \text{ is a transient Markov policy}\}; \\ K(S) & := & \{x(\beta,R) \mid R \text{ is a transient stationary policy}\}; \\ K(D) & := & \{x(\beta,R) \mid R \text{ is a transient deterministic policy}\}; \\ P & := & \left\{ x \;\middle|\; \begin{array}{rcl} \sum_{(i,a)}\{\delta_{ij} - p_{ij}(a)\}x_{ia} & = & \beta_j,\ j \in S \\ x_{ia} & \geq & 0,\ (i,a) \in S \times A \end{array} \right\}. \end{array}$$

Theorem 9.16
$\overline{K(D)} \subseteq K(S) = K(M) = K = P$, where $\overline{K(D)}$ is the closed convex hull of the finite set of vectors $K(D)$.

Proof
The equality $K = K(M)$ follows directly from Theorem 1.1. Furthermore, it is obvious that $K(D) \subseteq K(S) \subseteq K(M) \subseteq K$. Since P is a polyhedron, Theorem 4.7 implies $\overline{K(D)} \subseteq P = K(S)$ and consequently, $\overline{K(D)} \subseteq P = K(S) \subseteq K(M) = K$. Therefore, it is sufficient to show that $K(M) \subseteq P$. Take any transient Markov policy $R = (\pi^1, \pi^2, \dots)$. We have seen in the proof of Theorem 4.14 that $x(R)$ is a feasible solution of (4.15), i.e. $x(R) \in P$. □

The next example shows that $K(D) \neq P$ is possible and that anomalies may occur when we allow general initial distributions.

Example 9.7
$S = \{1,2\};\ A(1) = \{1,2\}, A(2) = \{2\};\ p_{11}(1) = 1, p_{12}(1) = 0;\ p_{11}(2) = 0, p_{12}(2) = 1;\ p_{21}(1) = 0,$ $p_{22}(1) = \frac{1}{2}$. First, take $\beta_1 = \beta_2 = 1$. There is only one transient deterministic policy f^∞ and this policy has $f(1) = 2,\ f(2) = 1$. For this policy, we have $x_{11}(f) = 0,\ x_{12}(f) = \frac{1}{2},\ x_{21}(f) = 2$. The set P is given by:

$$P = \left\{ x \;\middle|\; \begin{array}{ccccc} & x_{12} & & = & \frac{1}{2} \\ - & x_{12} & + \ \frac{1}{2}x_{21} & = & \frac{1}{2} \\ \multicolumn{3}{c}{x_{11},\ x_{12},\ x_{21} \geq 0} & & \end{array} \right\} = \{x \mid x_{11} \geq 0,\ x_{12} = \tfrac{1}{2},\ x_{21} = 2\}.$$

$\overline{K(D)} = \{x \mid x_{11} =,\ x_{12} = \frac{1}{2},\ x_{21} = 2\}$.

Next, take $\beta_1 = 0,\ \beta_2 = 1$. Then, the set P becomes:

$$P = \left\{ x \;\middle|\; \begin{array}{ccccc} & x_{12} & & = & 0 \\ - & x_{12} & + \ \frac{1}{2}x_{21} & = & 1 \\ \multicolumn{3}{c}{x_{11},\ x_{12},\ x_{21} \geq 0} & & \end{array} \right\} = \{x \mid x_{11} \geq 0,\ x_{12} = 0,\ x_{21} = 2\}.$$

Since $x_{12} = 0$ for all feasible solutions, it is natural to choose in state 1 action 1 with probability 1. However, this policy is not transient.

Because the above difficulties, we make the following assumption for this section.

Assumption 9.2
The initial distribution β has strictly positive components, i.e., $\beta_j > 0,\ j \in S$ and $\sum_j \beta_j = 1$.

In order to solve problem (9.17) we consider the following linear program:

$$max \left\{ \sum_{(i,a)} r_i(a)x_i(a) \;\middle|\; \begin{array}{rcl} \sum_{(i,a)} \{\delta_{ij} - p_{ij}(a)\}x_i(a) & = & \beta_j,\ j \in S \\ \sum_{(i,a)} c_i^k(a)x_i(a) & \leq & b_k,\ k = 1, 2, \ldots, m \\ x_i(a) & \geq & 0,\ (i,a) \in S \times A \end{array} \right\} \tag{9.18}$$

Theorem 9.17

(1) The linear program (9.18) is infeasible if and only if the CMDP (9.17) is infeasible.

(2) If the linear program (9.18) has an infinite optimal solution, then there exists no optimal transient policy for the CMDP (9.18).

(3) If x is a finite optimal solution of program (9.18), then $\pi^\infty(x)$, defined by (4.16), is a stationary transient optimal policy for the CMDP (9.17).

(4) If R_* is a transient optimal policy for the CMDP (9.17), then $x(R_*)$ is a finite optimal solution of program (9.18)

Proof

1. Because $K = P$ (see Theorem 9.16), problem (9.17) is feasible if and only if problem (9.18) is feasible.
2. If problem (9.18) has an infinite optimal solution, then also the unconstrained problem has an infinite optimal solution. In Section 4.6 we have seen that in that case there does not exist an optimal transient policy for problem (9.17).
3. Let x be a finite optimal solution of problem (9.18) and let $\pi^\infty(x)$ be defined by (4.16). Since, by Theorem 4.7, there is a bijection between the set of transient stationary policies and the feasible solutions of (4.15), $\pi^\infty(x)$ is a feasible policy for (9.17). Let R be an arbitrary feasible policy for problem (9.17). Then, $v(\beta, R) = \sum_{(j,a)} r_j(a)x_{ja}(\beta, R) \leq \sum_{(j,a)} r_j(a)x_{ja}$ and $c_k(\beta, R) = \sum_{(j,a)} c_j^k(a)x_{ja}(\beta, R) \leq b_k,\ 1 \leq k \leq m$. Hence, $\pi^\infty(x)$ is an optimal policy for (9.17).
4. This property follows directly from $c_k(\beta, R_*) = \sum_{(j,a)} c_j^k(a)x_{ja}(\beta, R_*) \leq b_k$ for $1 \leq k \leq m$ and $v(\beta, R_*) = \sum_{(j,a)} r_j(a)x_{ja}(\beta, R_*) \geq v(\beta, R) = \sum_{(j,a)} r_j(a)x_{ja}(\beta, R)$ for all transient policies R.

□

Algorithm 9.3 *Construction of an optimal stationary policy for the CMDP (9.17)*

Input: Instance of an MDP, an initial distribution β with $\beta_j > 0,\ j \in S$ and $\sum_j \beta_j = 1$, immediate costs $c_i^k(a),\ (i,a) \in S \times A$ and bounds b_k for $k = 1, 2, \ldots, m$.

Output: Either the statement that problem (9.17) is infeasible or an optimal stationary policy π^∞ of problem (9.17).

```
1. Solve the linear program (9.18).
2. **if** program (9.18) is infeasible **then begin** problem (9.17) is infeasible; **go to** step 3 **end**
   **else begin if** program (9.18) has an infinite solution **then**
                  **begin** problem (9.17) is infeasible; **go to** step 3 **end**
         **else if** program (9.18) has optimal policy $x$ **then**
                  **begin** the stationary policy $\pi^\infty$, defined by $\pi_{ia} := \frac{x_{ia}}{\sum_a x_{ia}}$ for all
                     $(i,a) \in S \times A$, is an optimal stationary policy for problem (9.17)
                  **end**
      **end**
3. STOP
```

9.2.4 Infinite horizon and total rewards for transient MDPs

In this section we discuss *transient* MDP, i.e. $\sum_{t=1}^\infty \mathbb{P}_R\{X_t = j,\ Y_t = a \mid X_1 = i\} < \infty$ for all $i \in S,\ (j,a) \in S \times A$ and all policies R. With this assumption there are no problems as exhibit in Example 9.7. Therefore, we may allow any initial distribution β. The constrained problem is in this case

$$sup_R\,\{v(\beta,R) \mid c_k(\beta,R) \le b_k,\ k = 1,2,\dots,m\}. \tag{9.19}$$

Quite similar to the proofs of Theorem 9.12 and Theorem 9.13, the following results can be shown, where $K, K(M), K(S)$ and P are similar defined as in the previous section 9.2.3.

Theorem 9.18
$K = K(M) = K(S) = \overline{K(D)} = P$, where $\overline{K(D)}$ is the closed convex hull of the finite set of vectors $K(D)$.

Theorem 9.19

(1) *The linear program (9.18) is infeasible if and only if the CMDP (9.19) is infeasible.*

(2) *If x is an optimal solution of program (9.18), then π^∞, defined by $\pi_{ia} := \frac{x_i(a)}{\sum_a x_i(a)}$ if $\sum_a x_i(a) > 0$ and arbitrarily if $\sum_a x_i(a) = 0$, is a stationary optimal policy for (9.19).*

The following algorithm solves the CMDP (9.19).

Algorithm 9.4 *Construction of an optimal stationary policy for the CMDP (9.19)*
Input: Instance of an MDP, an initial distribution β, immediate costs $c_i^k(a),\ (i,a) \in S \times A$ and bounds b_k for $k = 1,2,\dots,m$.
Output: Either the statement that problem (9.19) is infeasible or an optimal stationary policy π^∞ of problem (9.19).

```
1. Solve the linear program (9.18).
2. **if** program (9.18) is infeasible **then begin** problem (9.19) is infeasible; **go to** step 3 **end**
   **else begin if** program (9.18) has optimal solution $x$ **then**
                 **begin** the stationary policy $\pi^\infty$, defined by $\pi_{ia} := \frac{x_i(a)}{\sum_a x_i(a)}$ if $\sum_a x_i(a) > 0$
                       and arbitrarily if $\sum_a x_i(a) = 0$, is a stationary optimal policy for
                       problem (9.19);
                       **go to** step 3
                 **end**
       **end**
3. STOP
```

9.2.5 Finite horizon

In this section, we consider a nonstationary MDP, where besides the immediate rewards $r_i^t(a)$ there are also certain costs $c_i^{k,t}(a)$ for $(i,a) \in S \times A$ $1 \leq t \leq T$ and for $k = 1, 2, \dots, m$. Given initial distribution β, let the total expected reward and the total expected costs for policy R be denoted by $v^T(\beta, R)$ and $c^T(\beta, R)$, i.e.

$$v^T(\beta, R) := \textstyle\sum_{t=1}^T \sum_i \beta_i \cdot \sum_{(j,a)} \mathbb{P}_R\{X_t = j,\ Y_t = a \mid X_1 = i\} \cdot r_j^t(a);$$

$$c_k^T(\beta, R) := \textstyle\sum_{t=1}^T \sum_i \beta_i \cdot \sum_{(j,a)} \mathbb{P}_R\{X_t = j,\ Y_t = a \mid X_1 = i\} \cdot c_j^{k,t}(a),\ 1 \leq k \leq m.$$

The constrained problem is, given some real numbers $b_k,\ k = 1, 2, \dots, m$

$$sup_R\, \{v^T(\beta, R) \mid c_k^T(\beta, R) \leq b_k,\ k = 1, 2, \dots, m\}. \tag{9.20}$$

We shall use the transformation of Section 4.8. In this way, the finite horizon nonstationary MDP is considered as a transient stationary MDP. Therefore, we can use the results of the previous section 9.2.4. We introduce variables $x_{i,t}(a)$ with as interpretation the total expected frequencies for which $(X_t, Y_t) = (i, a)$, given a policy R and some initial distribution β. Then,

$$v^T(\beta, R) = \textstyle\sum_{t=1}^T \sum_{i,a} r_i^t(a) x_{i,t}(a);\ c_k^T(\beta, R) = \sum_{t=1}^T \sum_{i,a} c_i^{k,t}(a) x_{i,t}(a),\ 1 \leq k \leq m.$$

Hence, in order to solve (9.20) the linear programming problem, derived from (9.18), becomes:
$max\ \sum_{t=1}^T \sum_{(i,a)} r_i^t(a) x_{i,t}(a)$
subject to

$$\begin{array}{rcll}
\sum_a x_{j,1}(a) & & & = & \beta_j,\ j \in S \\
\sum_a x_{j,t}(a) & - & \sum_{(i,a)} p_{ij}^{t-1}(a) x_{i,t-1}(a) & = & 0,\ j \in S,\ 2 \leq t \leq T \\
x_{T+1} & - & \sum_{(i,a)} x_{i,T}(a) & = & 0 \\
\multicolumn{3}{l}{\sum_{t=1}^T \sum_{i,a} c^{k,t} x_{i,t} x_{i,t}(a)} & \leq & b_k,\ k = 1, 2, \dots, m \\
 & & x_{i,t}(a) & \geq & 0,\ (i,a) \in S \times A,\ 1 \leq t \leq T
\end{array}$$

By applying Theorem 9.19, we obtain the following result.

Theorem 9.20

(1) The above linear program is infeasible if and only if the CMDP (9.20) is infeasible.

(2) If x is an optimal solution of the above linear program, then $R^* = (\pi^1, \pi^2, \dots, \pi^T)$ is an optimal policy, where $\pi^t_{ia} := \frac{x_{i,t}(a)}{\sum_a x_{i,t}(a)}$ if $\sum_a x_{i,t}(a) > 0$ and arbitrarily if $\sum_a x_{i,t}(a) = 0$.

Remark

From the computational point of view we propose the following approach:

1. Use the special simplex algorithm of Section 4.8 (Algorithm 4.7) to compute an unconstrained optimal solution and a dual feasible solution.
2. Use the dual simplex method to compute an optimal solution for the constrained problem.

9.2.6 Infinite horizon and average rewards

Consider a similar problem as in the discounted case, but with average rewards, with respect to immediate rewards $r_i(a)$, and costs, with respect to immediate costs $c^k_i(a),\ (i,a) \in S \times A$, for $k = 1, 2, \dots, m$. Let β be an arbitrary initial distribution. For any policy R, let the average reward and the average k-th cost function with respect to the initial distribution β be defined by

$$\phi(\beta, R) := \liminf_{T\to\infty} \tfrac{1}{T} \textstyle\sum_{t=1}^T \sum_{j\in S} \beta_j \cdot \sum_{(i,a)} \mathbb{P}_R\{X_t = i,\ Y_t = a \mid X_1 = j\} \cdot r_i(a)$$

and

$$c^k(\beta, R) := \liminf_{T\to\infty} \tfrac{1}{T} \textstyle\sum_{t=1}^T \sum_{j\in S} \beta_j \cdot \sum_{(i,a)} \mathbb{P}_R\{X_t = i,\ Y_t = a \mid X_1 = j\} \cdot c^k_i(a),$$

respectively. A policy R is a feasible policy for a CMDP with average rewards and costs if $c^k(R) \le b_k,\ k = 1, 2, \dots, m$. An *optimal policy* R^* for this criterion is a feasible policy that maximizes $\phi(\beta, R)$, i.e.

$$\phi(\beta, R^*) = sup_R\,\{\phi(\beta, R) \mid c^k(\beta, R) \le b_k,\ k = 1, 2, \dots, m\}. \tag{9.21}$$

For any policy R, any initial distribution β and any $T \in \mathbb{N}$, we denote the *expected state-action frequencies* in the first T periods by

$$x^{\beta,T}_{ia}(R) := \frac{1}{T} \sum_{t=1}^T \sum_{j\in S} \beta_j \cdot \mathbb{P}_R\{X_t = i,\ Y_t = a \mid X_1 = j\},\ (i,a) \in S \times A. \tag{9.22}$$

By $X(\beta, R)$ we denote the set of all limit points of the vectors $\{x^{\beta,T}(R),\ T = 1, 2, \dots\}$. These limit points $x(\beta, R)$ are limit points in the $S \times A$-dimensional vector space of vectors $x^{\beta,T}(R)$ with components $x^{\beta,T}_{ia}(R),\ (i,a) \in S \times A$. Any $x^{\beta,T}(R)$ satisfies $\sum_{(i,a)} x^{\beta,T}_{ia}(R) = 1$ and therefore also $\sum_{(i,a)} x_{ia}(\beta, R) = 1$ for all $x(\beta, R) \in X(\beta, R)$.

For $\pi^\infty \in C(S)$ we have $\mathbb{P}_{\pi^\infty}\{X_t = i,\ Y_t = a \mid X_1 = j\} = \big\{P^{t-1}(\pi)\big\}_{ji} \cdot \pi_{ia}$ for all (i,a) and consequently, $\lim_{T\to\infty} x^{\beta,T}_{ia}(\pi^\infty) = \sum_{j\in S} \beta_j \cdot \big\{P^*(\pi)\big\}_{ji} \cdot \pi_{ia}$, i.e. $X(\beta,\pi^\infty)$ consists of one element, namely $x(\beta,\pi)$, where $x_{ia}(\beta,\pi) := \big\{\beta^T P^*(\pi)\big\}_i \cdot \pi_{ia},\ (i,a) \in S \times A$. Let the policy set C_1 be the set of *convergent policies*, defined by $C_1 := \{R \mid X(\beta,R)$ consists of one element$\}$. Hence, $C(S) \subseteq C_1$. Furthermore, define the vector sets $L,\ L(M),\ \ L(C),\ L(S)$ and $L(D)$ by

$$\begin{aligned}
L \quad &:= \quad \{x(\beta,R) \in X(\beta,R) \mid R \text{ is an arbitrary policy}\};\\
L(M) \quad &:= \quad \{x(\beta,R) \in X(\beta,R) \mid R \text{ is a Markov policy}\};\\
L(C) \quad &:= \quad \{x(\beta,R) \in X(\beta,R) \mid R \text{ is a convergent policy}\};\\
L(S) \quad &:= \quad \{x(\beta,R) \in X(\beta,R) \mid R \text{ is a stationary policy}\};\\
L(D) \quad &:= \quad \{x(\beta,R) \in X(\beta,R) \mid R \text{ is a deterministic policy}\}.
\end{aligned}$$

General case

In the general case the Markov chain $P(f)$ for any $f^\infty \in C(D)$ may be irreducible, unichain or multichain. We will show that $L = L(M) = L(C) = \overline{L(S)} = \overline{L(D)}$. Therefore, we require that there exists a deterministic optimal policy with respect to the average rewards $\overline{\phi}(R)$, defined by

$$\overline{\phi}_j(R) := \limsup_{T\to\infty} \frac{1}{T} \sum_{t=1}^{T} \sum_{(i,a)} \mathbb{P}_R\{X_t = i,\ Y_t = a \mid X_1 = j\} \cdot r_i(a),\ j \in S. \tag{9.23}$$

Lemma 9.15

Let $f^\infty \in C(D)$ be an optimal policy with respect to the average rewards $\phi(R)$. Then, f^∞ is also an optimal policy with respect to the average rewards $\overline{\phi}(R)$.

Proof

From Theorem 1.1 it follows that it is sufficient to prove that $\overline{\phi}(f^\infty) \geq \overline{\phi}(R)$ for all Markov policies R. Let $R = (\pi^1, \pi^2, \dots)$ be an arbitrary Markov policy. Since the value vector ϕ is superharmonic (cf. Theorem 5.17), there exists a vector $u \in \mathbb{R}^N$ such that $\phi_i \geq \sum_j p_{ij}(a)\phi_j$ and $\phi_i + u_i \geq r_i(a) + \sum_j p_{ij}(a)u_j$ for all $(i,a) \in S \times A$. Hence, $\phi \geq P(\pi^t)\phi$ and $\phi + u - P(\pi^t)\phi \geq r(\pi^t)$ for $t = 1, 2, \dots$. Consequently,

$$\begin{aligned}
\textstyle\sum_{t=1}^{T} P(\pi^1)P(\pi^2)\cdots P(\pi^{t-1})r(\pi^t) \quad &\leq \quad \textstyle\sum_{t=1}^{T} P(\pi^1)P(\pi^2)\cdots P(\pi^{t-1}) \cdot \{\phi + u - P(\pi^t)u\}\\
&\leq \quad T \cdot \phi + u - P(\pi^1)P(\pi^2)\cdots P(\pi^T)u,\ T \in \mathbb{N}.
\end{aligned}$$

Since $\frac{1}{T}\{u - P(\pi^1)P(\pi^2)\cdots P(\pi^T)u\} \ \to\ 0$ for $T \to \infty$, we can write

$$\begin{aligned}
\overline{\phi}_j(R) \quad &= \quad \textstyle\limsup_{T\to\infty} \frac{1}{T} \sum_{t=1}^{T} \{P(\pi^1)P(\pi^2)\cdots P(\pi^{t-1})r(\pi^t)\}_j\\
&\leq \quad \textstyle\limsup_{T\to\infty} \frac{1}{T} \{T \cdot \phi + u - P(\pi^1)P(\pi^2)\cdots P(\pi^T)u\}_j \ = \ \phi_j = \phi_j(f^\infty),\ j \in S. \qquad \square
\end{aligned}$$

Theorem 9.21
$L = L(M) = L(C) = \overline{LS)} = \overline{L(D)}.$

Proof
The proof has the same structure as the proof as Theorem 9.12. The equality $L = L(M)$ follows directly from Theorem 1.1. Furthermore, it is obvious that $L(D) \subseteq L(S) \subseteq L(C) \subseteq L$. We first show that $L \subseteq \overline{L(D)}$. Suppose the contrary. Then, there exists a policy R such that $x(\beta, R) \in L$ and $x(\beta, R) \notin \overline{L(D)}$. Since $\overline{L(D)}$ is a closed convex set, it follows from the Separating Hyperplane Theorem, that there are coefficients $r_i(a),\ (i,a) \in S \times A$, such that

$$\sum_{i,a} x_{ia}(\beta, R) r_i(a) > \sum_{i,a} x_{ia} r_i(a) \text{ for all } x \in \overline{L(D)}. \tag{9.24}$$

Consider the MDP with immediate rewards $r_i(a),\ (i,a) \in S \times A$. We have seen in Chapter 5 that there exists an average optimal policy $f^\infty \in C(D)$ with respect to $\phi(R)$. By Lemma 9.15, f^∞ is also average optimal with respect to $\overline{\phi}(R)$. Because $x(\beta, R) \in L$, there exists a sequence $\{T_k,\ k = 1, 2, \dots\}$ such that $x_{ia}(\beta, R) = \lim_{k\to\infty} x_{ia}^{\beta,T_k}(R),\ (i,a) \in S \times A$. Hence,

$$\begin{array}{rcl}
\sum_{(i,a)} r_i(a) x_{ia}(\beta, R) & = & \sum_{(i,a)} r_i(a) \cdot \lim_{k\to\infty} x_{ia}^{\beta,T_k}(R) \\
& = & \lim_{k\to\infty} \frac{1}{T_k} \sum_{t=1}^{T_k} \sum_{j\in S} \beta_j \cdot \sum_{(i,a)} \mathbb{P}_R\{X_t = i,\ Y_t = a \mid X_1 = j\} \cdot r_i(a) \\
& \le & \sum_{j\in S} \beta_j \cdot \limsup_{k\to\infty} \frac{1}{T_k} \sum_{t=1}^{T_k} \sum_{(i,a)} \mathbb{P}_R\{X_t = i,\ Y_t = a \mid X_1 = j\} \cdot r_i(a) \\
& = & \sum_{j\in S} \beta_j \cdot \overline{\phi}(R) \le \sum_{j\in S} \beta_j \cdot \overline{\phi}(f^\infty) = \sum_{(i,a)} r_i(a) x_{ia}(\beta, f^\infty),
\end{array}$$

which contradicts (9.24), completing the proof that $L \subseteq \overline{L(D)}$. Since $L(D) \subseteq L(S) \subseteq L(C) \subseteq L$, we obtain $\overline{L(S)} = \overline{L(D)}$.
From $L(C) \subseteq L = L(M) \subseteq \overline{L(S)} = \overline{L(D)}$, it remains to show that $\overline{L(D)} \subseteq L(M) \cap L(C)$. Take any $x \in \overline{L(D)}$. Let $C(D) = \{f_1^\infty, f_2^\infty, \dots, f_n^\infty\}$. Then, $x_{ia} = \sum_{k=1}^n p_k\, x_{ia}(\beta, f_k^\infty),\ (i,a) \in S \times A$ for certain $p_k \ge 0$ with $\sum_{k=1}^n p_k = 1$. By Theorem 1.1, there exists a policy $R \in C(M)$ satisfying

$$\sum_{j\in S} \beta_j \cdot \mathbb{P}_R\{X_t = i,\ Y_t = a \mid X_1 = j\} = \sum_{j\in S} \beta_j \cdot \sum_{k=1}^n p_k\, \mathbb{P}_{f_k^\infty}\{X_t = i,\ Y_t = a \mid X_1 = j\},$$

for all $(i,a) \in S \times A$ and $t = 1, 2, \dots$. Hence,

$$\begin{array}{rcl}
x_{ia} & = & \sum_{k=1}^n p_k\, x_{ia}(\beta, f_k^\infty) \\
& = & \sum_{k=1}^n p_k \lim_{T\to\infty} \frac{1}{T} \sum_{t=1}^T \sum_{j\in S} \beta_j \cdot \mathbb{P}_{f_k^\infty}\{X_t = i,\ Y_t = a \mid X_1 = j\} \\
& = & \lim_{T\to\infty} \frac{1}{T} \sum_{t=1}^T \sum_{j\in S} \beta_j \cdot \mathbb{P}_R\{X_t = i,\ Y_t = a \mid X_1 = j\} \\
& = & x_{ia}(\beta, R),\ (i,a) \in S \times A.
\end{array}$$

Therefore, $x = x(\beta, R) \in L(M)$, and $x = \lim_{T\to\infty} x^{\beta,T}(R) \in L(C)$, which completes the proof of the theorem. □

Analogously to the discounted case we introduce a polyhedron, namely

$$Q := \left\{ x \;\middle|\; \begin{array}{rcrcl}
\sum_{(i,a)}\{\delta_{ij} - p_{ij}(a)\} x_{ia} & & & = & 0,\ j \in S \\
\sum_a x_{ja} & + & \sum_{(i,a)}\{\delta_{ij} - p_{ij}(a)\} y_{ia} & = & \beta_j,\ j \in S \\
 & & x_{ia},\ y_{ia} & \ge & 0,\ (i,a) \in S \times A
\end{array} \right\}.$$

Hence, Q is the projection (on the x-space) of the feasible solutions (x, y) of the dual linear program (5.29) for the computation of an average optimal policy.

Theorem 9.22
$L = Q$.

Proof
Theorem 9.21 implies that it is sufficient to show that $\overline{L(D)} = Q$. For $\pi^\infty \in C(S)$, we have $x_{ia}(\beta, \pi) = \{\beta^T P^*(\pi)\}_i \cdot \pi_{ia},\ (i, a) \in S \times A$. Then, with y^π defined by (5.36), we have shown in Theorem 5.19 that (x^π, y^π) is a feasible solution of dual linear program (5.29) (it can easily be checked that the proof of Theorem 5.19 is also valid when $\beta_j = 0$ for some $j \in S$).
Hence, $L(D) \subseteq L(S) \subseteq Q$. Since Q is the projection of a polyhedron, Q is also a polyhedron and consequently $\overline{L(D)} \subseteq Q$. If $x \in Q$, then it follows from the definition of Q that $x_{ia} \geq 0$ for all $(i, a) \in S \times A$ and $\sum_{(j,a)} x_{ja} = \sum_j \beta_j = 1$. Therefore, Q is a polytope, i.e. a bounded polyhedron. Hence, Q is the closed convex hull of a finite number of extreme points, and consequently it is sufficient to show that any extreme point of Q belongs to $L(D)$.
Let x^* be an arbitrary extreme point of Q and let Q^* be the closed convex hull of the extreme points of Q that are different from x^*. Then, $x^* \notin Q^*$ and, by the Separating Hyperplane Theorem, there are coefficients $r_i(a),\ S \times A$, such that

$$\sum_{i,a} r_i(a)x^*_{ia} > \sum_{i,a} r_i(a)x_{ia} \text{ for all } x \in Q^*. \tag{9.25}$$

From (9.25) it follows that any optimal solution $(\overline{x}, \overline{y})$ of

$$max \left\{ \sum_{(i,a)} r_i(a)x_i(a) \;\middle|\; \begin{array}{rrcl} \sum_{(i,a)} \{\delta_{ij} - p_{ij}(a)\}x_i(a) & & = & 0,\ j \in S \\ \sum_a x_j(a) & + \sum_{(i,a)} \{\delta_{ij} - p_{ij}(a)\}y_i(a) & = & \beta_j,\ j \in S \\ & x_i(a), y_i(a) & \geq & 0,\ (i, a) \in S \times A \end{array} \right\} \tag{9.26}$$

satisfies $\overline{x} = x^*$. Let $f_*^\infty \in C(D)$ be an average optimal policy for the MDP with immediate rewards $r_i(a),\ S \times A$. Then, by Theorem 5.20, (x^f, y^f) - defined by (5.35) and (5.36) - is an optimal solution of (9.26). Hence, $x^* = x^f \in C(D)$, which completes the proof. □

Example 9.8
From the Theorems 9.21 and 9.22 it follows that any extreme piint of Q is an element of $L(D)$. This example will show that the converse statement is not true, in general. Furthermore, this example shows that $L(S) \neq Q$ is possible and that Q can be a real subset of

$$Q_0 := \left\{ x \;\middle|\; \begin{array}{lcl} \sum_{(i,a)}\{\delta_{ij} - p_{ij}(a)\}x_{ia} & = & 0,\ j \in S \\ \sum_{(i,a)} x_{ia} & = & 1 \\ x_{ia} \geq 0,\ (i, a) \in S \times A & & \end{array} \right\}.$$

Consider the following MDP: $S = \{1,2,3\};\ A(1) = \{1,2\},\ A(2) = \{1,2\},\ A(3) = \{1\};$
$p_{11}(1) = 0,\ p_{12}(1) = 1,\ p_{13}(1) = 0;\ p_{11}(2) = 0,\ p_{12}(2) = 0,\ p_{13}(2) = 1;$
$p_{21}(1) = 0,\ p_{22}(1) = 1;\ p_{23}(1) = 0;\ p_{21}(2) = 1,\ p_{22}(2) = 0;\ p_{23}(2) = 0;$
$p_{31}(1) = 0,\ p_{32}(1) = 0,\ p_{33}(1) = 1$. Take $\beta_1 = \beta_2 = \beta_3 = \frac{1}{3}$.

Any stationary policy π^∞ induces a Markov chain with $P(\pi) = \begin{pmatrix} 0 & \pi_1 & 1-\pi_1 \\ \pi_2 & 1-\pi_2 & 0 \\ 0 & 0 & 1 \end{pmatrix}$.

For the computation of $P^*(\pi)$ and x^π we distinguish between the following three cases.

Case 1: $\pi_1 = 1$:

$$P^*(\pi) = \begin{pmatrix} \frac{\pi_2}{1+\pi_2} & \frac{1}{1+\pi_2} & 0 \\ \frac{\pi_2}{1+\pi_2} & \frac{1}{1+\pi_2} & 0 \\ 0 & 0 & 1 \end{pmatrix} \text{ and } \begin{array}{l} x_1^\pi(1) = \frac{2}{3} \cdot \frac{\pi_2}{1+\pi_2};\ x_1^\pi(2) = 0; \\ x_2^\pi(1)) = \frac{2}{3} \cdot \frac{1-\pi_2}{1+\pi_2};\ x_2^\pi(2) = \frac{2}{3} \cdot \frac{\pi_2}{1+\pi_2}; \\ x_3^\pi(1) = \frac{1}{3}. \end{array}$$

Case 2: $\pi_1 \neq 1$ and $\pi_2 = 0$:

$$P^*(\pi) = \begin{pmatrix} 0 & \pi_1 & 1-\pi_1 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \text{ and } \begin{array}{l} x_1^\pi(1) = 0;\ x_1^\pi(2) = 0; \\ x_2^\pi(1) = \frac{1}{3} \cdot (1+\pi_1);\ x_(^\pi 2) = 0; \\ x_3^\pi(1) = \frac{1}{3} \cdot (2-\pi_1). \end{array}$$

Case 3: $\pi_1 \neq 1$ and $\pi_2 \neq 0$:

$$P^*(\pi) = \begin{pmatrix} 0 & 0 & 1 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \end{pmatrix} \text{ and } \begin{array}{l} x_1^\pi(1) = 0;\ x_1^\pi(2) = 0; \\ x_2^\pi(1) = 0;\ x_2^\pi(2) = 0; \\ x_3^\pi(1) = 1. \end{array}$$

Since in each case $x_1^\pi(1) = x_2^\pi(2),\ x_1^\pi(2) = 0,\ x_1^\pi(1) + x_1^\pi(2) + x_2^\pi(1) + x_2^\pi(2) + x_3^\pi(1) = 1$, we can describe $L(S)$ in the following space with the nonnegative variables $x_{11},\ x_{21}$ and x_{31}:
$L(S) = \{2x_{11}{+}x_{21} = \frac{2}{3}; x_{31} = \frac{1}{3}\} \cup \{x_{11} = 0; x_{21}{+}x_{31} = 1, \frac{1}{3} \leq x_{31} \leq \frac{2}{3}\} \cup \{x_{11} = x_{21} = 0; x_{31} = 1\}$.
The four deterministic policies correspond to $\pi_1 = 1,\ \pi_2 = 1;\ \pi_1 = 1,\ \pi_2 = 0;\ \pi_1 = 0,\ \pi_2 = 1$ and $\pi_1 = 0,\ \pi_2 = 0$, respectively. The corresponding elements of $L(S)$ are:
$x_1^{f_1}(1) = \frac{1}{3},\ x_2^{f_1}(1) = 0,\ x_3^{f_1}(1) = \frac{1}{3};\ x_1^{f_2}(1) = 0,\ x_2^{f_2}(1) = \frac{2}{3},\ x_3^{f_2}(1) = \frac{1}{3},$
$x_1^{f_3}(1) = 0,\ x_2^{f_3}(1) = 0,\ x_3^{f_3}(1) = 1;\ x_1^{f_4}(1) = 0,\ x_2^{f_4}(1) = \frac{1}{3},\ x_3^{f_4}(1) = \frac{2}{3}.$

Q is the closed convex hull of $x^{f_1},\ x^{f_2},\ x^{f_3}$ and x^{f_4}. Hence $x := \frac{1}{4}\{x^{f_1} + x^{f_2} + x^{f_3} + x^{f_4}\} \in Q$ and $x_{11} = \frac{1}{12},\ x_{21} = \frac{1}{4},\ x_{31} = \frac{7}{12}$ and it can easily verified that x is not an element of $L(S)$.
Since Q is the closed convex hull of $x^{f_1},\ x^{f_2},\ x^{f_3}$ and x^{f_4}, we have

$$Q = \left\{ \begin{array}{rcl} x_{11} + x_{12} + x_{21} + x_{22} + x_{31} & = & 1 \\ x_{12} = 0;\ x_{11} = x_{22};\ x_{31} & \geq & \frac{1}{3} \\ x_{11}, x_{12}, x_{21}, x_{22}, x_{31} & \geq & 0 \end{array} \right\} \text{ and } Q_0 = \left\{ \begin{array}{rcl} x_{11} + x_{12} - x_{22} & = & 0 \\ -\ x_{11} \qquad\quad + x_{22} & = & 0 \\ -\ x_{12} \qquad\quad & = & 0 \\ x_{11} + x_{12} + x_{21} + x_{22} + x_{31} & = & 1 \end{array} \right\}.$$

Since x^*, defined by $x^*_{11} := \frac{1}{2},\ x^*_{12} := 0,\ x^*_{21} := 0,\ x^*_{22} := \frac{1}{2},\ x^*_{31} := 0$ belongs to Q_0 and not to Q, i.e. Q is a real subset of Q_0.

In order to solve the CMDP (9.21) we consider the linear program

$$max \left\{ \sum_{(i,a)} r_i(a)x_i(a) \;\middle|\; \begin{array}{lcl} \sum_{(i,a)} \{\delta_{ij} - p_{ij}(a)\}x_i(a) & = & 0,\ j \in S \\ \sum_a x_j(a) \ + \ \sum_{(i,a)} \{\delta_{ij} - p_{ij}(a)\}y_i(a) & = & \beta_j,\ j \in S \\ \sum_{(i,a)} c_i^k(a)x_i(a) & \leq & b_k,\ k = 1, 2, \dots, m \\ x_i(a), y_i(a) \geq 0,\ (i,a) \in S \times A & & \end{array} \right\}. \tag{9.27}$$

Theorem 9.23

(1) *Problem (9.21) is feasible if and only if problem (9.27) is feasible.*

(2) *The optima of (9.21) and (9.27) are equal.*

(3) *If R is optimal for problem (9.21), then $x(\beta, R)$ is optimal for (9.27).*

(4) *Let (x, y) be an optimal solution of problem (9.27) and let $x = \sum_{k=1}^n p_k x(\beta, f_k)$, where $p_k \geq 0$ and $\sum_{k=1}^n p_k = 1$ and $f_1^\infty, f_2^\infty, \dots, f_n^\infty$ are the stationary policies of $C(D)$. Let $R \in C(M)$ be the policy of Theorem 1.1 that satisfies*
$\sum_j \beta_j \cdot \mathbb{P}_R\{X_t = i, Y_t = a \mid X_1\} = \sum_j \beta_j \cdot \sum_k p_k \cdot \mathbb{P}_{f_k^\infty}\{X_t = i, Y_t = a \mid X_1\} = \beta_j\}$
for all $(i, a) \in S \times A$ and all $t \in \mathbb{N}$. Then, R is an optimal solution of problem (9.21).

Proof

The theorems 9.21 and 9.22 imply that $Q = L(C)$. Moreover, $\phi(\beta, R) = \sum_{i,a} x_{ia}(\beta, R)r_i(a)$ for any $R \in C_1$. By these observations, the parts (1), (2) and (3) are straightforward. For the proof of part (4) we can similarly as in the proof of Theorem 9.21 show that $x = x(\beta, R)$ and $R \in C_1$. Therefore, $\phi(\beta, R) = \sum_{i,a} x_{ia}(\beta, R)r_i(a) = \sum_{i,a} x_{ia}r_i(a) =$ optimum of problem (9.27). Hence, R is an optimal policy for problem (9.21). □

To compute an optimal policy of problem (9.21) from an optimal solution (x, y) of the linear program (9.27), we first have to express x as $x = \sum_{k=1}^n p_k x(\beta, f_k)$, where $p_k \geq 0$ and $\sum_{k=1}^n p_k = 1$. Next, we have to determine the policy $R = (\pi_1, \pi^2, \dots) \in C(M)$ such that this policy satisfies
$\sum_j \beta_j \cdot \mathbb{P}_R\{X_t = i, Y_t = a \mid X_1\} = \sum_j \beta_j \cdot \sum_k p_k \cdot \mathbb{P}_{f_k^\infty}\{X_t = i, Y_t = a \mid X_1\} = \beta_j\}$ for all $(i, a) \in S \times A$ and all $t \in \mathbb{N}$. The decision rules $\pi^t,\ t \in \mathbb{N}$, can be determined by formula (1.8) in Theorem 1.1.

Algorithm 9.5 *Construction of an optimal policy $R \in L(M) \cap L(C)$ for CMDP problem (9.21)*

Input: Instance of an MDP, an initial distribution β, immediate costs $c_i^k(a),\ (i, a) \in S \times A$ and bounds b_k for $k = 1, 2, \dots, m$.

Output: Either the statement that (9.21) is infeasible or an optimal policy $R \in L(M) \cap L(C)$ of problem (9.21).

```
1. Solve the linear program (9.27).

2. **if** program (9.27) is infeasible **then begin** problem (9.21) is infeasible; **go to** step 7 **end**
   **else begin if** program (9.28) has optimal solution $(x, y)$ **then go to** step 3 **end**
```

3. **for all** $f^\infty \in C(D)$ **do** compute $P^*(f)$ (assume $C(D) = \{f_1^\infty, f_2^\infty, \cdots, f_n^\infty\}$)

4. $x_{ia}^k := \begin{cases} \sum_j \beta_j \{P^*(f_k)\}_{ji} & \text{if } a = f_k(i) \\ 0 & \text{if } a \neq f_k(i) \end{cases}, \quad i \in S,\; k = 1, 2, \dots, n.$

5. Determine $p_k,\; k = 1, 2, \dots, n$ as feasible solution of the following linear system (this computation can be performed by the Phase I technique of the simplex method)
$$\begin{cases} \sum_{k=1}^n p_k x_{ia}^k & = & x_{ia} & a \in A(i),\; i \in S \\ \sum_{k=1}^n p_k & = & 1 & \\ p_k & \geq & 0 & k = 1, 2, \dots, n \end{cases}$$

6. $R := (\pi^1, \pi^2, \dots)$, where π^t is defined by
$$\pi_{ia}^t := \begin{cases} \frac{\sum_j \beta_j \sum_k p_k \{P^{t-1}(f_k)\}_{ji} \cdot \delta_{af_k(i)}}{\sum_j \beta_j \sum_k p_k \{P^{t-1}(f_k)\}_{ji}} & \text{if } \sum_j \beta_j \sum_k p_k \{P^{t-1}(f_k)\}_{ji} \neq 0 \\ \text{arbitrary} & \text{if } \sum_j \beta_j \sum_k p_k \{P^{t-1}(f_k)\}_{ji} = 0, \end{cases}$$
is an optimal policy for problem (9.21).

7. STOP

Example 9.9
Consider the following MDP: $S = \{1, 2, 3\};\; A(1) = \{1, 2\},\; A(2) = \{1\},\; A(3) = \{1, 2\};$
$r_1(1) = r_1(2) = 0;\; r_2(1) = 1;\; r_3(1) = r_3(2) = 0;$
$p_{11}(1) = 0,\; p_{12}(1) = 1,\; p_{13}(1) = 0;\; p_{11}(2) = 0,\; p_{12}(2) = 0,\; p_{13}(2) = 1;$
$p_{21}(1) = 0,\; p_{22}(1) = 1;\; p_{23}(1) = 0;\; p_{31}(1) = 0,\; p_{32}(2) = 0;\; p_{33}(2) = 1;$
$p_{31}(2) = 0,\; p_{32}(2) = 1,\; p_{33}(2) = 0$. Take $\beta_1 = \frac{1}{4},\; \beta_2 = \frac{3}{16},\; \beta_3 = \frac{9}{16}$.
As constraint we have bounds for the value $x_{12}(\beta, R)$: $\frac{1}{4} \leq x_{12}(\beta, R) \leq \frac{1}{2}$.
If we apply Algorithm 9.5 we have to solve the following linear program:

$$\begin{array}{l} \textit{maximize } x_2(1) \textit{ subject to} \\
\begin{array}{rrrrrrrrrrrrrrrrcr}
 & x_1(1) & + & x_1(2) & & & & & & & & & & & & & = & 0 \\
- & x_1(1) & & & & & & & - & x_3(2) & & & & & & & = & 0 \\
 & & - & x_1(2) & & & & & + & x_3(2) & & & & & & & = & 0 \\
 & x_1(1) & + & x_1(2) & & & & & & & + & y_1(1) & + & y_1(2) & - & y_3(2) & = & \frac{1}{4} \\
 & & & & & x_2(1) & & & & & - & y_1(1) & & & + & y_3(2) & = & \frac{3}{16} \\
 & & & & & & & x_3(1) & + & x_3(2) & & & - & y_1(2) & & & = & \frac{9}{16} \\
 & & & & & x_2(1) & & & & & & & & & & & \leq & \frac{1}{2} \\
 & & & & - & x_2(1) & & & & & & & & & & & \leq & -\frac{1}{4}
\end{array} \\
x_1(1),\; x_1(2),\; x_2(1),\; x_3(1),\; x_3(2) \;\geq\; 0
\end{array}$$

with optimal solution $x_1(1) = 0$, $x_1(2) = 0$, $x_2(1) = \frac{1}{2}$, $x_3(1) = \frac{1}{2}$, $x_3(2) = 0$; $y_1(1) = 0$, $y_1(2) = \frac{1}{4}$, $y_3(2) = \frac{5}{16}$.

There are four deterministic policies:

$f_1(1) = 1,\; f_1(2) = 1,\; f_1(3) = 1;\; f_2(1) = 1,\; f_2(2) = 1,\; f_2(3) = 2;$

$f_3(1) = 2,\; f_3(2) = 1,\; f_3(3) = 1;\; f_4(1) = 2,\; f_4(2) = 1,\; f_4(3) = 2.$

The corresponding stationary matrices are:

$$P^*(f_1) = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix};\ P^*(f_2) = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 0 \end{pmatrix};\ P^*(f_3) = \begin{pmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix};\ P^*(f_4) = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 0 \end{pmatrix}.$$

The vectors $x^1,\ x^2,\ x^3,\ x^4$ are:

$x^1_{11} = 0;\ x^1_{12} = 0;\ x^1_{21} = \frac{7}{16};\ x^1_{31} = \frac{9}{16};\ x^1_{32} = 0.\ x^2_{11} = 0;\ x^2_{12} = 0;\ x^2_{21} = 1;\ x^2_{31} = 0;\ x^2_{32} = 0.$

$x^3_{11} = 0;\ x^3_{12} = 0;\ x^3_{21} = \frac{3}{16};\ x^3_{31} = \frac{13}{16};\ x^3_{32} = 0.\ x^4_{11} = 0;\ x^4_{12} = 0;\ x^4_{21} = 1;\ x^4_{31} = 0;\ x^4_{32} = 0.$

For the numbers $p_1,\ p_2,\ p_3,\ p_4 \geq 0$ such that $\sum_{k=1}^4 p_k = 1$ and $p_1x^1 + p_2x^2 + p_3x^3 + p_4x^4 = x$ we obtain: $p_1 = \frac{8}{9},\ p_2 = \frac{1}{9},\ p_3 = 0,\ p_4 = 0$.

Since $P^t(f_1) = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$ and $P^t(f_2) = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 0 \end{pmatrix}$ for all $t \in \mathbb{N}$, we obtain

$R = (\pi^1, \pi^2, \dots)$ with $\pi^t_{11} = 1,\ t \in \mathbb{N};\ \pi^t_{21} = 1,\ t \in \mathbb{N};\ \pi^t_{31} = \begin{cases} \frac{8}{9} & t = 1 \\ 1 & t \geq 2 \end{cases};\ \pi^t_{32} = \begin{cases} \frac{1}{9} & t = 1 \\ 1 & t \geq 2 \end{cases}.$

Remark

Algorithm 9.5 is unattractive for practical problems. The number of calculations is prohibitive. Moreover, the use of Markov policies is inefficient in practice. Therefore, in the next pages we discuss the problem of finding an optimal stationary policy, if one exists.

For any feasible solution (x, y) of (9.27) we define a stationary policy π^∞ by

$$\pi_{ia} := \begin{cases} \frac{x_i(a)}{x_i} & i \in S_x \\ \frac{y_i(a)}{y_i} & i \in S_y \\ \text{arbitrary} & \text{if } i \notin S_y \cup S_x \end{cases} \tag{9.28}$$

where $x_i := \sum_a x_i(a),\ y_i := \sum_a y_i(a),\ S_x := \{x \mid x_i > 0\}$ and $S_y := \{y \mid x_i = 0,\ y_i > 0\}$. Notice that, since $\beta_j = 0$ is allowed for one or more $j \in S$, it is possible that $S_x \cup S_y \neq S$.

Lemma 9.16

If (x, y) is an optimal solution of (9.27) and $x_i(a) = \pi_{ia} \cdot \{\beta^T P^*(\pi)\}_i,\ (i, a) \in S \times A$, where π is defined by (9.28), then π^∞ is an optimal solution of (9.21).

Proof

Since $c^k(\beta, \pi^\infty) = \beta^T P^*(\pi) c^k(\pi) = \sum_i \{\beta^T P^*(\pi)\}_i \sum_a c^k_i(a)\pi_{ia} = \sum_{(i,a)} c^k_i(a)x_i(a) \leq b_k$ for all $1 \leq k \leq m$, the stationary policy π^∞ is a feasible solution of (9.21). Moreover, by Theorem 9.23, part (2), we have $\phi(\beta, \pi^\infty) = \beta^T P^*(\pi) r(\pi) = \sum_{(i,a)} r_i(a)x_i(a) =$ optimum (9.27) = optimum (9.21), i.e. π^∞ is an optimal solution of (9.21). □

The next example shows that for an optimal solution (x, y) of (9.27), the policy π^∞, where π is defined by (9.28), is not an optimal solution of (9.21), even in the case that (9.21) has a stationary optimal policy.

Example 9.10

Consider the model of Example 9.9, but now with the constraint $x_{21}(\beta, R) \leq \frac{1}{4}$. The linear program (9.27) for this constrained problem is

$$\begin{array}{l}
\textit{maximize } x_2(1) \textit{ subject to} \\
\begin{array}{rrrrrrrrrrrrrrrr}
 & x_1(1) & + & x_1(2) & & & & & & & & & & & = & 0 \\
- & x_1(1) & & & & & - & x_3(2) & & & & & & & = & 0 \\
 & & - & x_1(2) & & & + & x_3(2) & & & & & & & = & 0 \\
 & x_1(1) & + & x_1(2) & & & & & + & y_1(1) & + & y_1(2) & - & y_3(2) & = & \frac{1}{4} \\
 & & & & x_2(1) & & & & - & y_1(1) & & & + & y_3(2) & = & \frac{3}{16} \\
 & & & & & x_3(1) & + & x_3(2) & & & - & y_1(2) & & & = & \frac{9}{16} \\
 & & & & x_2(1) & & & & & & & & & & \leq & \frac{1}{4}
\end{array} \\
x_1(1),\ x_1(2),\ x_2(1),\ x_3(1),\ x_3(2)\ \geq\ 0
\end{array}$$

with optimal solution $x_1(1) = 0,\ x_1(2) = 0,\ x_2(1) = \frac{1}{4},\ x_3(1) = \frac{3}{4},\ x_3(2) = 0;\ y_1(1) = 0,$ $y_1(2) = \frac{1}{4},\ y_3(2) = \frac{1}{16}$ and with optimum value $\frac{1}{4}$. The corresponding stationary policy π^∞ satisfies $\pi_{12} = \pi_{21} = \pi_{31} = 1$.

This policy is not optimal, because $\phi(\beta, \pi^\infty) = \frac{3}{16} < \frac{1}{4}$, the optimum of the linear program.

Consider the stationary policy with $\pi_{11} = \frac{1}{4},\ \pi_{12} = \frac{3}{4},\ \pi_{21} = \pi_{31} = 1$. For this policy we obtain $x_{21}(\beta, \pi^\infty) = \frac{1}{4}$ and $\phi(\beta, \pi^\infty) = \frac{1}{4}$, the optimum value of the linear program. So, this policy is feasible and optimal.

In order to apply Lemma 9.16 we have to compute the stationary matrix $P^*(\pi)$. The determination of the stationary matrix can be executed in polynomial time (see Algorithm 5.5). However, if $\frac{x_i(a)}{x_i} = \frac{y_i(a)}{y_i}$ for all $a \in A(i),\ i \in \{j \mid x_j > 0,\ y_j > 0\}$, which is for instance the case if $\{j \mid x_j > 0,\ y_j > 0\} = \emptyset$, then the policy π^∞, where π is defined by (9.28), is an optimal policy for problem (9.21) as the next lemma shows.

Lemma 9.17

If $\frac{x_i(a)}{x_i} = \frac{y_i(a)}{y_i}$ for all $a \in A(i),\ i \in \{j \mid x_j > 0,\ y_j > 0\}$, then the stationary policy π^∞, where π is defined by (9.28), is an optimal policy for problem (9.21).

Proof

The condition $\frac{x_i(a)}{x_i} = \frac{y_i(a)}{y_i}$ for all $a \in A(i),\ i \in \{j \mid x_j > 0,\ y_j > 0\}$ implies that $\frac{y_i(a)}{y_i} = \pi_{ia}$ for all $a \in A(i),\ i \in \{j \mid y_j > 0\}$, i.e. $y_i(a) = \pi_{ia} \cdot y_i$ for all $a \in A(i),\ i \in S$. Hence, we can write

$$\beta_j = x_j + \textstyle\sum_{(i,a)} \{\delta_{ij} - p_{ij}(a)\}\pi_{ia} \cdot y_i = x_j + \sum_i y_i\{\delta_{ij} - p_{ij}(\pi)\},\ j \in S.$$

So (x, y) satisfies, in vector notation, $x^T = x^T P(\pi)$ and $x^T + y^T\{I - P(\pi)\} = \beta^T$. Consequently, $x^T = x^T P^*(\pi)$ and $x^T P^*(\pi) = \beta^T P^*(\pi)$. So, $x^T = \beta^T P^*(\pi)$, i.e. x satisfies the conditions of Lemma 9.16. □

If the conditions of Lemma 9.17 are not satisfied, we can try to find - for the same x - another y, say $\overline{y}$, such that $(x, \overline{y})$ is feasible for (9.27) - and consequently also optimal - and satisfies the conditions of Lemma 9.17. To achieve this, we need $\frac{\overline{y}_i(a)}{\overline{y}_i} = \pi_{ia},\ a \in A(i),\ i \in \{j \mid x_j > 0,\ \overline{y}_j > 0\}$, which is equivalent to $\overline{y}_i(a) = \overline{y}_i \cdot \pi_{ia},\ a \in A(i),\ i \in \{j \mid \overline{y}_j > 0\}$. Hence, $\overline{y}$ has to satisfy the linear system in the y-variables (x is fixed)

$$\begin{cases} \sum_{i \notin S_x} \sum_a \{\delta_{ij} - p_{ij}(a)\}\overline{y}_i(a) + \sum_{i \in S_x} \{\delta_{ij} - p_{ij}(\pi)\}\overline{y}_i = \beta_j - x_j,\ j \in S \\ \overline{y}_i(a) \geq 0,\ i \notin S_x,\ a \in A(i);\ \overline{y}_i \geq 0,\ i \in S_x \end{cases} \tag{9.29}$$

The feasibility of system (9.29) can be checked by the so-called phase I of the simplex method.

Example 9.11

Consider the model of Example 9.10. The optimal solution does not satisfy $x_i(a)/x_i = y_i(a)/y_i$ for all $a \in A(i),\ i \in \{j \mid x_j > 0,\ y_j > 0\}$, because $x_3(2)/x_3 = 0$ and $y_3(2)/y_3 = 1$. The system (9.29) becomes $\overline{y}_1(1) + \overline{y}_1(2) = \frac{4}{16}$; $-\overline{y}_1(1) = -\frac{1}{16}$; $-\overline{y}_1(2) = -\frac{3}{16}$; $\overline{y}_1(1), \overline{y}_1(2) \geq 0$. This system has the solution $\overline{y}_1(1) = \frac{1}{16},\ \overline{y}_1(2) = \frac{3}{16}$. Hence, the stationary policy π^∞ with $\pi_{11} = \frac{1}{4},\ \pi_{12} = \frac{3}{4},\ \pi_{21} = \pi_{31} = 1$ is an optimal policy for problem (9.21).

Remark

If the x-part of problem (9.27) is unique and (9.29) is infeasible, then problem (9.21) has no optimal stationary policy, namely:

Suppose that (9.21) has an optimal stationary policy, say π^∞. Then, (x^π, y^π), defined by (5.35) and (5.36), is a feasible solution for problem (9.27) and $\sum_{(i,a)} r_i(a)x_i^\pi(a) = \beta^T P^*(\pi)r(\pi) =$ optimum (9.21). Hence, (x^π, y^π) is an optimal solution of problem (9.27). Consequently, $x^\pi = x$. Then, y^π is a feasible solution of system (9.29), which is contradictory to the assumption that (9.29) is infeasible.

Example 9.12

Consider the model of Example 9.9. We have seen that $x_1(1) = 0, x_1(2) = 0, x_2(1) = \frac{1}{2}, x_3(1) = \frac{1}{2}, x_{32} = 0$; $y_1(1) = 0,\ y_1(2) = \frac{1}{4},\ y_3(2) = \frac{5}{16}$ is an optimal solution. It can easily be verified that the x-part of the solution is unique. The system (9.21) is: $\overline{y}_1(1) + \overline{y}_1(2) = \frac{4}{16}$; $-\overline{y}_1(1) = -\frac{5}{16}$; $-\overline{y}_1(2) = \frac{1}{16}$; $\overline{y}_1(1), \overline{y}_1(2) \geq 0$. The system is infeasible and therefore the problem has no stationary optimal policy.

Unichain case

For this case we will show that $L(S) = Q$, which implies $L = L(M) = L(C) = L(S) = \overline{L(D)} = Q$. In order to show $L(S) = Q$ we need the following two lemmas.

Lemma 9.18

For every triple (j, a, R), where $j \in S,\ a \in A(j)$ and R a convergent policy, we have

$x_{ja}(\beta, R) = \lim_{\alpha \uparrow 1} (1 - \alpha) \sum_{t=1}^\infty \alpha^{t-1} \cdot \sum_i \beta_i \cdot \mathbb{P}_R\{X_t = j,\ Y_t = a \mid X_1 = i\}$.

Proof

Let R be a convergent policy and let $x(\beta, R) = \lim_{T\to\infty} x^T(\beta, R)$. Take a fixed pair $(j, a) \in S \times A$. Then, $x_{ja}(\beta, R) = \lim_{T\to\infty} \frac{1}{T} \sum_{t=1}^T w_t$, where $w_t := \sum_i \beta_i \cdot \mathbb{P}_R\{X_t = j,\ Y_t = a \mid X_1 = i\}$. Since $|w_t|$ is bounded by 1 for all t, the power series $\sum_{t=1}^\infty w_t\alpha^{t-1}$ has radius of convergence at least 1. The series $\sum_{t=1}^\infty \alpha^{t-1}$ has radius of convergence 1. Hence, we can write

$$(1-\alpha)^{-1} \cdot \sum_{t=1}^\infty w_t\,\alpha^{t-1} = \{\sum_{t=1}^\infty \alpha^{t-1}\} \cdot \{\sum_{t=1}^\infty w_t\,\alpha^{t-1}\} = \sum_{t=1}^\infty \{\sum_{s=1}^t w_s\}\,\alpha^{t-1}.$$

Since $(1-\alpha)^{-2} = \sum_{t=1}^\infty t\,\alpha^{t-1}$, we obtain

$$x_{ja}(\beta, R) - (1-\alpha)\sum_{t=1}^\infty \alpha^{t-1} \cdot \sum_i \beta_i \cdot \mathbb{P}\{X_t = j,\ Y_t = a \mid X_1 = i\} =$$
$$(1-\alpha)^2 \sum_{t=1}^\infty \left\{x_{ja}(\beta, R) - \tfrac{1}{t}\sum_{s=1}^t w_s\right\} t\,\alpha^{t-1}.$$

Choose $\varepsilon > 0$ arbitrary. Since $x_{ja}(\beta, R) = \lim_{T\to\infty} \frac{1}{T}\sum_{t=1}^T w_t$, there exists an integer T_ε such that $|x_{ja}(\beta, R) - \frac{1}{T}\sum_{t=1}^T w_t| \le \frac{1}{2}\varepsilon$ for all $T > T_\varepsilon$. Hence,

$$|(1-\alpha)^2 \sum_{t=1}^{T_\varepsilon} \left\{x_{ja}(\beta, R) - \tfrac{1}{t}\sum_{s=1}^t w_s\right\} t\,\alpha^{t-1}| \le (1-\alpha)^2 M \cdot \sum_{t=1}^{T_\varepsilon} T_\varepsilon\,\alpha^{t-1} \le \tfrac{1}{2}\varepsilon$$

for α sufficiently close to 1 and $M \ge \max_{1\le t\le T_\varepsilon} |x_{ja}(\beta, R) - \frac{1}{t}\sum_{s=1}^t w_s|$. Furthermore, we have

$$|(1-\alpha)^2 \sum_{t=T_\varepsilon+1}^\infty \left\{x_{ja}(\beta, R) - \tfrac{1}{t}\sum_{s=1}^t w_s\right\} t\,\alpha^{t-1}| \le (1-\alpha)^2 \sum_{t=T_\varepsilon+1}^\infty \tfrac{1}{2}\varepsilon\, t\,\alpha^{t-1} \le \tfrac{1}{2}\varepsilon.$$

Hence, $x_{ja}(\beta, R) = \lim_{\alpha\uparrow 1} (1-\alpha)\sum_{t=1}^\infty \alpha^{t-1} \cdot \sum_i \beta_i \cdot \mathbb{P}_R\{X_t = j,\ Y_t = a \mid X_1 = i\}$. □

Lemma 9.19

If $x(\beta, \pi^\infty)$ is continuous in π, then $L(S) = Q$.

Proof

Since $L(S) \subseteq L(C)$, it is sufficient to show that $L(C) \subseteq L(S)$. Take any $x(\beta, R) \in L(C)$. From Theorem 9.12 it follows that for any discount factor $\alpha \in [0, 1)$ there exists a stationary policy π_α^∞ such that $x^\alpha(\beta, R) = x^\alpha(\beta, \pi_\alpha^\infty)$. Choose a fixed pair $(j, a) \in S \times A$ and let the reward function r on $S \times A$ be defined by $r_i(b) := \begin{cases} 1 & \text{if } i = j \text{ and } b = a; \\ 0 & \text{elsewhere.} \end{cases}$

Then, $\beta^T v^\alpha(\pi_\alpha^\infty) = x_{ja}^\alpha(\beta, \pi_\alpha^\infty)$ and $\beta^T\phi(\pi_\alpha^\infty) = x_{ja}(\beta, \pi_\alpha^\infty)$ for all $\alpha \in [0, 1)$. Hence, we can write by Lemma 9.18

$$x_{ja}(\beta, R) = \lim_{\alpha\uparrow 1} (1-\alpha) \cdot x_{ja}^\alpha(\beta, R) = \lim_{\alpha\uparrow 1} (1-\alpha) \cdot x_{ja}^\alpha(\beta, \pi_\alpha^\infty) = \lim_{\alpha\uparrow 1} (1-\alpha) \cdot \beta^T v^\alpha(\pi_\alpha^\infty).$$

Consider a sequence $\{\alpha_k,\ k = 1, 2, \dots\}$ such that $\alpha_k \uparrow 1$ and $\pi_{\alpha_k} \to \pi$. Since for any $i \in S$ the sequence $\{(1-\alpha_k)v_i^{\alpha_k}(\pi_{\alpha_k}^\infty),\ k = 1, 2, \dots\}$ is dominated by the sequence $\{(1-\alpha_k)v_i^{\alpha_k},\ k = 1, 2, \dots\}$ and since $lim_{k\to\infty} \{(1-\alpha_k)v_i^{\alpha_k} = \phi_i$, there exists a limit point, say x, of the sequence of vectors $\{(1-\alpha_k)v_i^{\alpha_k}(\pi_{\alpha_k}^\infty),\ k = 1, 2, \dots\}$. Therefore, we may assume that

$$x_i = lim_{k\to\infty} (1-\alpha_k)v_i^{\alpha_k}(\pi_{\alpha_k}^\infty),\ i \in S, \tag{9.30}$$

implying, by Lemma 9.18,

$$\beta^T x = \sum_i \beta_i \cdot \lim_{k\to\infty} (1-\alpha_k)v_i^{\alpha_k}(\pi_{\alpha_k}^\infty) = \lim_{k\to\infty} (1-\alpha_k)\,\beta^T v^{\alpha_k}(\pi_{\alpha_k}^\infty) = x_{ja}(\beta, R). \tag{9.31}$$

Since $x(\beta, \pi^\infty)$ is continuous in π we can write for $\pi_{\alpha_k} \to \pi$,

$$x_{ja}(\beta,\pi^\infty) = \lim_{k\to\infty} x_{ja}(\beta,\pi^\infty_{\alpha_k}) \;=\; \lim_{k\to\infty} (1-\alpha_k)\{\textstyle\sum_{t=1}^\infty \alpha^{t-1}\} \cdot \beta^T P^*(\pi^\infty_{\alpha_k}) r(\pi_{\alpha_k}).$$

Because $P^* = P^* P^t$ for any stationary matrix P^* and any $t \in \mathbb{N}$, we obtain

$$\begin{aligned} x_{ja}(\beta,\pi^\infty) &= \lim_{k\to\infty} (1-\alpha_k)\{\textstyle\sum_{t=1}^\infty \alpha^{t-1}\} \cdot \beta^T P^*(\pi^\infty_{\alpha_k}) P^{t-1}(\pi^\infty_{\alpha_k}) r(\pi_{\alpha_k}) \\ &= \lim_{k\to\infty} \beta^T P^*(\pi^\infty_{\alpha_k})(1-\alpha_k) \textstyle\sum_{t=1}^\infty \alpha^{t-1} P^{t-1}(\pi^\infty_{\alpha_k}) r(\pi_{\alpha_k}) \\ &= \lim_{k\to\infty} \big\{x(\pi^\infty_{\alpha_k})\big\}^T (1-\alpha_k) v^{\alpha_k}(\pi^\infty_{\alpha_k}) \;=\; \big\{x(\pi^\infty)\big\}^T x = \beta^T P^*(\pi) x. \end{aligned}$$

Since $v^{\alpha_k}(\pi^\infty_{\alpha_k}) = r(\pi^\infty_{\alpha_k}) \;+\; \alpha P(\pi^\infty_{\alpha_k}) v^{\alpha_k}(\pi^\infty_{\alpha_k})$, we can also write

$$(1-\alpha_k) v^{\alpha_k}(\pi^\infty_{\alpha_k}) = (1-\alpha_k) r(\pi^\infty_{\alpha_k}) + \alpha P(\pi^\infty_{\alpha_k})(1-\alpha_k) v^{\alpha_k}(\pi^\infty_{\alpha_k}).$$

Letting $k \to \infty$, then - by (9.30) - we obtain $x = P(\pi)x$ and consequently, $x = P^*(\pi)x$. Finally, we have $x_{ja}(\beta,R) = \lim_{k\to\infty} (1-\alpha_k)\beta^T v^{\alpha_k}(\pi^\infty_{\alpha_k}) = \beta^T x = \beta^T P^*(\pi) x = x_{ja}(\beta,\pi)$. Since the stationary policy π is independent of the choice of the pair (j,a), we have shown $x(\beta,R) \in L(S)$.

□

Theorem 9.24

$L(S) = Q = Q_0$, where Q_0 is defined in Example 9.8.

Proof

In order to show $L(S) = Q$, by Lemma 9.19, it is sufficient to show that $x(\beta,\pi^\infty)$ is continuous in π. Let $\pi^\infty(k),\ k = 1,2,\dots$ and $\pi^\infty(0)$ be stationary policies such that $\pi(0) = \lim_{k\to\infty} \pi(k)$. By the unichain property the stationary distribution $p^*\big(\pi(k)\big)$ of the Markov chain $P\big(\pi(k)\big)$ is the unique solution of the linear system

$$\left\{ \begin{array}{lcl} \sum_i \{\delta_{ij} - p_{ij}\big(\pi(k)\big)\} x_i &=& 0 \\ \sum_i x_i &=& 1 \end{array} \right. \tag{9.32}$$

Since $\pi(k) \to \pi(0)$ for $k \to \infty$, we also have $P\big(\pi(k)\big) \to P\big(\pi(0)\big)$ for $k \to \infty$. Consequently, any limit point of $\{p^*\big(\pi(k)\big),\ k = 1,2,\dots\}$ is a solution of (9.32) with $k = 0$, i.e. is equal to $p^*\big(\pi(0)\big)$. Hence, $x_{ia}\big(\beta,\pi^\infty(k)\big) = p^*_i\big(\pi(k)\big) \cdot \pi_{ia}(k) \to p^*_i\big(\pi(0)\big) \cdot \pi_{ia}(0) = x_{ia}\big(\beta,\pi^\infty(0)\big)$, i.e. $x(\beta,\pi^\infty)$ is continuous in π.

Since $Q \subseteq Q_0$, for the proof of $Q = Q_0$, it is sufficient to show that $Q_0 \subseteq L(S)$. Take any $x \in Q_)$, i.e. $\sum_{(i,a)} \{\delta_{ij} - p_{ij}(a)\} x_{ia} = 0,\ j \in S,\ \sum_{(i,a)} x_{ia} = 1$ and $x_{ia} \geq 0,\ (i,a) \in S \times A$.

Set $\pi_{ia} := \left\{ \begin{array}{ll} \frac{x_{ia}}{\sum_a x_{ia}} & a \in A(i),\ i \in S_x := \{i \mid \sum_a x_{ia} > 0\}; \\ \text{arbitrary} & \text{otherwise.} \end{array} \right.$ Then, $x_{ia} = x_i \cdot \pi_{ia}$, where

$x_i := \sum_{(i,a)} x_{ia}$, for all $(i,a) \in S \times A$. Therefore, $\left\{ \begin{array}{lcl} \sum_i \{\delta_{ij} - p_{ij}(\pi)\} x_i &=& 0; \\ \sum_i x_i &=& 1. \end{array} \right.$

Hence, $x^T P(\pi) = x^T,\ x^T e = 1$ and $x \geq 0$, i.e. x is a stationary distribution of $P(\pi)$. By the unichain assumption the stationary distribution is unique, so $x_i = p^*_i(\pi)$ for all $i \in S$, and consequently, $x_{ia} = p^*_i(\pi) \cdot \pi_{ia}$ for all $(i,a) \in S \times A$. Therefore, $x \in L(S)$. □

By these results an optimal stationary policy for the CMDP in the unichain case can be computed by the following algorithm.

Algorithm 9.6 *Construction of a stationary optimal policy π^∞ for CMDP problem (9.21)*

Input: Instance of a unichain MDP, immediate costs $c_i^k(a),\ (i,a) \in S \times A$ and bounds b_k for $k = 1, 2, \dots, m$.

Output: Either the statement that (9.21) is infeasible or an optimal stationary policy π^∞. of problem (9.21).

1. Solve the linear program

$$max \left\{ \sum_{(i,a)} r_i(a)x_i(a) \;\middle|\; \begin{array}{lcl} \sum_{(i,a)} \{\delta_{ij} - p_{ij}(a)\}x_i(a) & = & 0,\ j \in S \\ \sum_{(i,a)} x_i(a) & = & 1 \\ \sum_{(i,a)} c_i^k(a)x_i(a) & \leq & b_k,\ k = 1, 2, \dots, m \\ x_i(a) \geq 0,\ (i,a) \in S \times A & & \end{array} \right\}. \tag{9.33}$$

2. **if** program (9.33) is infeasible **then begin** problem (9.21) is infeasible; **go to** step 4 **end**

   else begin if program (9.28) has optimal solution x **then go to** step 3 **end**

3. π^∞, where $\pi_{ia} := \begin{cases} \frac{x_i(a)}{\sum_a x_i(a)} & a \in A(i),\ i \in S_x := \{i \mid \sum_a x_i(a) > 0\} \\ \text{arbitrary} & \text{otherwise} \end{cases}$ is an optimal stationary policy.

4. STOP

Theorem 9.25

The stationary policy π^∞ obtained by Algorithm 9.6 is an optimal policy for problem (9.21).

Proof

From the proof of Theorem 9.24 it follows that $x_i = p_i^*(\pi),\ i \in S$. Consequently, $x = x(\beta, \pi^\infty)$. Therefore, π^∞ is feasible for (9.21). Moreover, $\phi(\beta, \pi^\infty) = \sum_{(i,a)} r_i(a)x_i(a) =$ optimum (9.33). From Theorem 9.24 it follows that there exists a stationary optimal policy of problem (9.21), say π_*^∞. Let $x^* = x(\beta, \pi_*^\infty)$. Then, x^* is a feasible solution of program (9.33) and consequently,

$$\text{optimum (9.21)} = \phi(\beta, \pi_*^\infty) = \textstyle\sum_{(i,a)} r_i(a)x_i^*(a) \leq \sum_{(i,a)} r_i(a)x_i(a) = \phi(\beta, \pi^\infty).$$

Hence, π^∞ obtained by Algorithm 9.6 is an optimal policy for problem (9.21). □

Remark

If the MDP is irreducible, then any solution of the system $\begin{cases} \sum_i \{\delta_{ij} - p_{ij}(\pi)\}x_i & = & 0 \\ \sum_i x_i & = & 1 \end{cases}$ satisfies $x_i > 0,\ i \in S$. Since any optimal extreme solution of the CMDP has at most $|S| + m$ positive variables, the optimal stationary policy π^∞ is nondeterministic in at most m states.

9.2.7 Constrained MDPs with sum of discounted rewards and different discount factors

This section deals with an MDP which has $m+1$ criteria. Each criterion is a sum of standard expected discounted total rewards over infinite horizon with different discount factors and different one-step rewards. We consider the problem of optimizing one criterion under inequality constraints on the other criteria. We prove that, given an initial state, if a feasible policy exists, then there exists an optimal *ultimately deterministic policy* $R = (\pi^1, \pi^2, \dots, \pi^T, f, f, \dots)$ such that, for $t = 1, 2, \dots, T$, the Markov decision rule π^t uses at most m actions more than a deterministic Markov decision rule. Such a policy is called an (m,T)*-policy.*

We will formulate a linear programming algorithm for the approximate solution of this constrained MDP. Furthermore, for the multiple criteria problem with the $m+1$ criteria, we show that any point on the boundary of the performance region can be reached by an (m,T)-policy. Since any Pareto optimal point belongs to the boundary, it follows that the performance of any Pareto optimal policy can be obtained by an equivalent (m,T)-policy. We also show that, given an initial state and a policy, there exists an $(m+1,T)$-policy with the same performance.

Several applications of MDPs in finance, project management, budget allocation and production lead to more than one criterion, each of them has with its own discount factor. In the next example we describe such application to a production system.

Example 9.13

Consider an unreliable production system consisting of two units, say 1 and 2. Unit k can fail at each epoch with probability p_k under the condition that it has been operating before (when a unit fails in a certain epoch it fails forever). The system operates if at least one of the units operates. Let $c_i^k(a)$, $k = 1$ or 2, be an operating cost for unit k, if its state is i and decision a is chosen. Let α be the discount factor. Then, the discounted reward for unit k at time t is $\alpha^{t-1} \cdot (1-p_k)^{t-1} \cdot c^k_{X_t}(Y_t) = \alpha_k^{t-1} \cdot c^k_{X_t}(Y_t)$ with $\alpha_k := \alpha \cdot (1-p_k)$ for $k = 1, 2$.

The problem of minimizing the total discounted costs under constraints on the corresponding costs for each unit is a constrained MDP with sum of discounted rewards and different discount factors.

A Markov decision rule π^t is of *order* m if $\pi^t_{ia} > 0$ for at most $N+m$ pairs $(i,a) \in S \times A$. A policy R is called an (m,T)*-policy* if $R = (\pi^1, \pi^2, \dots, \pi^T, f, f, \dots)$, where π^t is a Markov decision rule of order m for $t = 1, 2, \dots, T$ and f is a deterministic decision rule.

Let $u, v \in \mathbb{R}^{m+1}$. We say that u *dominates* v if $u - v \in \mathbb{R}^{m+1}_+$. Given a set $U \subseteq \mathbb{R}^{m+1}$, a point $u \in U$ is called *Pareto optimal in* U if there is no $v \in U$ which dominates u.

Given an initial state i and $m+1$ optimality criteria $v_i^0(R), v_i^1(R), \dots, v_i^m(R)$, let the $(m+1)$-dimensional vector $V(i,R) := \big(v_i^0(R), v_i^1(R), \dots, v_i^m(R)\big)$ characterize the performance of policy R. Let $U(i) := \{V(i,R) \mid R \in C\}$ be the *performance region.* A policy R is called *Pareto optimal* if $V(i,R)$ is Pareto optimal in $U(i)$. We say that policy R_1 dominates policy R_2 at i if $V(i,R_1)$ dominates $V(i,R_2)$ in $U(i)$.

We are interested in the solution of the following constrained optimization problem, given the numbers $b_1, b_2, \dots, b_m$ and initial state i, i.e. the problem

$$max\,\{v^0(i,R) \mid v^l(i,R) \geq b_l,\ l = 1,2,\dots,m\}, \tag{9.34}$$

where $v_i^l(R) = \sum_{k=1}^K \sum_{t=1}^\infty\ \alpha_k^{t-1} \sum_{j,a} \mathbb{P}_{i,R}\{X_t = j,\ Y_t = a\} \cdot r_j^{lk}(a),\ l = 0,1,\dots,m$ with $r_j^{lk}(a)$ for all $(j,a) \in S \times A$, the one-step rewards corresponding to criterion l and discount factor α_k. Notice that the unconstrained problem, i.e., $max\,\{v^0(i,R)\}$, was considered in section 7.13.

Lemma 9.20

The performance region $U(i)$ is a convex compact set.

Proof

Consider two elements of $U(i)$, say $V(i,R_1)$ and $V(i,R_2)$. Let $\lambda \in [0,1]$. By Theorem 1.1, there exists a Markov policy R_* such that

$$\mathbb{P}_{i,R}\{X_t = j,\ Y_t = a\} = \lambda \cdot \mathbb{P}_{i,R_1}\{X_t = j,\ Y_t = a\} + (1-\lambda) \cdot \mathbb{P}_{i,R_2}\{X_t = j,\ Y_t = a\}$$

for all $(j,a) \in S \times A$ and all $t \in \mathbb{N}$. This provides straightforward the convexity of $U(i)$.

Since $|v^l(i,R)| \leq \frac{K \cdot M}{1-\alpha}$ for all $l = 0,1,\dots,m$, with $M := max_{lk}\,\{max_{j,a}\, r_j^{lk}(a)\}$ and $\alpha := max_k\, \alpha_k$, the set $U(i)$ is bounded. Furthermore, we have for $l = 0,1,\dots,m$,

$$v^l(i,R) = \textstyle\sum_{k=1}^K\ v^{lk}(i,R), \text{ where } v^{lk}(i,R) := \sum_{t=1}^\infty\ \alpha_k^{t-1} \sum_{j,a}\ \mathbb{P}_{i,R}\{X_t = j,\ Y_t = a\} \cdot r_j^{lk}(a).$$

From the proof of Theorem 9.12 we know that for all k and l the set of $|S \times A|$-dimensional vectors $x^{kl}(R)$ with $x_{ja}^{kl}(R) := \sum_{t=1}^\infty\ \alpha_k^{t-1}\ \mathbb{P}_{i,R}\{X_t = j,\ Y_t = a\}$ is closed.

Since $v^l(i,R) = \sum_{k=1}^K \sum_{j,a}\ x_{ja}^{kl}(R) \cdot r_j^{lk}(a)$, the set $v^l(i,R)$ is also closed for all $l = 0,1,\dots,m$. Hence, $U(i)$ is a closed set, concluding the proof that $U(i)$ is a convex compact set. □

Remark

Since for a given $i \in S$ the set $U(i)$ is a compact set, problem (9.34) has an optimal solution if (9.34) is feasible. Since the set $U(i)$ is also convex, an optimal policy is either Pareto optimal in the set of feasible policies, or it is dominated by such a Pareto optimal policy.

We first consider a finite nonstationary horizon model with in period t and for the optimality criterion l $(0 \leq l \leq m)$ rewards $r^{l,t}(a),\ (j,a) \in S \times A$. The optimization problem in this case is

$$max\,\{v^0(i,R) \mid v^l(i,R) \geq b_l,\ l = 1,2,\dots,m\}, \tag{9.35}$$

where $v^l(i,R) := \sum_{t=1}^T \sum_{j,a}\ \mathbb{P}_{i,R}\{X_t = j,\ Y_t = a\} \cdot r_j^{l,t}(a),\ l = 0,1,\dots,m$. For this problem we consider the following linear program, where the variable x_{ja}^t can be interpreted as the state-action probability, i.e. $x_{ja}^t := \mathbb{P}\{X_t = j,\ Y_t = a \mid X_1 = i\}$:

$$max \left\{ \sum_{t=1}^T \sum_{(j,a)} r_j^{0,t}(a) x_{ja} \;\middle|\; \begin{array}{lcl} \sum_a x_{ja}^1 & = & \delta_{ij},\ j \in S \\ \sum_a x_{ja}^t - \sum_{l,a} p_{lj}(a) x_{la}^{t-1} & = & 0,\ j \in S,\ t = 2,3,\dots,T \\ \sum_{t=1}^T \sum_{j,a} r_j^{l,t}(a) x_{ja}^t & \geq & b_l,\ l = 1,2,\dots,m \\ x_{ja}^t & \geq & 0,\ (j,a) \in S \times A,\ t = 1,2,\dots,T \end{array} \right\}. \tag{9.36}$$

Theorem 9.26

(1) Problem (9.35) is feasible if and only if the corresponding LP problem is feasible.

(2) If x is an optimal basic solution of the linear program, then $R_* := (\pi^1, \pi^2, \dots, \pi^T)$, where

$$\pi^t_{ja} := \begin{cases} \frac{x^t_{ja}}{\sum_a x^t_{ja}} & \textit{if } \sum_a x^t_{ja} > 0 \\ 1 & \textit{if } \sum_a x^t_{ja} = 0 \textit{ and } a = a_j, \textit{ where } a_j \in A(j) \textit{ is arbitrarily chosen} \\ 0 & \textit{if } \sum_a x^t_{ja} = 0 \textit{ and } a \neq a_j \end{cases}$$

is an optimal Markov policy of order m.

Proof

In order to prove the theorem, we mention the following facts:

(1) a finite (non)stationary finite horizon model is equivalent to a transient nonstationary infinite horizon model (see section 2.3);

(2) a transient infinite horizon model is equivalent to a contracting infinite horizon model (see Theorem 4.8);

(3) a contracting infinite horizon model is equivalent to a discounted infinite horizon model (see the last part of section 4.7).

If we apply the above transformations, we obtain an MDP with state space $\overline{S}$, actions sets $\overline{A}$, transition probabilities $\overline{p}$ and one-step rewards $\overline{r}$, defined by:

(i) $\overline{S} := S \times \{1, 2, \dots, T \cup \{0\}$;

(ii) $\overline{A}(j,t) := A(j),\ j \in S,\ 1 \leq t \leq T;\ \overline{A}(0) := \{0\}$;

(iii) $\overline{p}_{(j,t)(k,t+1)}(a) := \frac{1}{\alpha} p_{jk}(a),\ j,k \in S,\ 1 \leq t \leq T-1,\ a \in A(j);\ \overline{p}_{(j,T)0}(a) :=$ ‘$1,\ j \in S,\ a \in A(j)$; $\overline{p}_{00}(0) := 1$: all other probabilities equal 0;

(iv) $\overline{r}^l_{(j,t)}(a) := r^{l,t}_j(a),\ j \in S,\ 1 \leq t \leq T,\ a \in A(j);\ \overline{r}^l_0(0) := 0$.

There is a natural one-to-one correspondence, given by $\pi^t_{ja} := \pi_{(j,t)}(a)$ for all j, a, t, between randomized Markov policies in the original finite horizon model and randomized stationary policies in the new infinite horizon discounted model. For every m this mapping is also a one-to-one correspondence between randomized Markov policies of order m in the original finite horizon model and randomized stationary policies of order m in the new infinite horizon discounted model. This correspondence preserves the values of the $m+1$ criteria.

The corresponding discounted optimization problem in the new discounted model is:

$$max\, \{\overline{v}^{0,\alpha}\big((i,1), R\big) \mid \overline{v}^{l,\alpha}\big((i,1), R\big) \geq b_l,\ l = 1, 2, \dots, m\}, \tag{9.37}$$

where $\overline{v}^{l,\alpha}\big((i,1), R\big) := \sum_{t=1}^\infty \alpha^{t-1} \sum_{(j,t)a} \overline{\mathbb{P}}_{(i,l),R}\{\overline{X}_t = (j,t),\ \overline{Y}_t = a\} \cdot \overline{r}^l_{(j,t)}(a),\ l = 0, 1, \dots, m$ for some $\alpha \in [0,1)$.

This optimization problem belongs to the model defined by (9.6) and can be solved by the linear program (9.9). The linear program (9.9) becomes in the setting of the new infinite horizon problem:

$$
max\left\{\sum_{(j,t),a}\overline{r}^0_{(j,t)}(a)\overline{x}_{j,t)}(a)\;\middle|\;\begin{array}{rcl}
\sum_a \overline{x}_{(j,1)}(a) & = & \delta_{ij},\ j\in S\\
\sum_a \overline{x}_{(j,t)}(a) - \alpha\sum_{(l,t-1),a}\overline{p}_{(l,t-1)(j,t)}(a)\overline{x}_{(l,t-1)}(a) & = & 0,\ j\in S,\ 2\le t\le T\\
\overline{x}_0(0) - \alpha\sum_{(l,T),a}\overline{p}_{(l,T)0}(a)\overline{x}_{(l,T)}(a) & = & 0\\
\sum_{(j,t),a}\overline{r}^l_{(j,t)}(a)\overline{x}_{(j,t)}(a) & \ge & b_l,\ l=1,2,\dots,m\\
\overline{x}_{(j,t)}(a)\ge 0,\ (j,a)\in S\times A,\ 1\le t\le T; & & \overline{x}_0(0)\ge 0
\end{array}\right\} \tag{9.38}
$$

If we write x^t_{ja} instead of $\overline{x}_{(j,t)}(a)$, then the objective function can be written as $\sum_{(j,t),a}\overline{r}^0_{(j,t)}(a)x^t_{ja}$ and the first set of constraints as $\sum_a x^1_{ja} = \delta_{ij},\ j\in S$. For the second set of constraints we obtain $\sum_a x^t_{ja} - \sum_{l,a} p_{lj}(a)x^{t-1}_{la} = 0,\ j\in S,\ t=2,3,\dots,T$. The next constraint of (9.38) gives $\overline{x}_0(0) - \alpha\sum_{la} x^T_{la} = 0$. Notice that the value of $\overline{x}_0(0)$ has no influence for the optimal solution,; so, this constraint may be omitted. The last set of constraints becomes $\sum_{t=1}^T\sum_{j,a} r^{l,t}_j(a)x^t_{ja}\ge b_l$ for $l=1,2,\dots,m$. Hence, program (9.38) is the same linear program as program (9.35) and from Theorem 9.13 and the second remark after the proof of Theorem 9.13 it follows that:
(1) Problem (9.35) is feasible if and only if the corresponding linear program (9.36) is feasible.
(2) If x is an optimal basic solution of the linear program, then $R_* := (\pi^1,\pi^2,\dots,\pi^T)$, where

$$
\pi^t_{ja} := \begin{cases} \frac{x^t_{ja}}{\sum_a x^t_{ja}} & \text{if } \sum_a x^t_{ja} > 0\\ 1 & \text{if } \sum_a x^t_{ja} = 0 \text{ and } a = a_j, \text{ where } a_j\in A(j) \text{ is arbitrarily chosen}\\ 0 & \text{if } \sum_a x^t_{ja} = 0 \text{ and } a\ne a_j\end{cases}
$$

is an optimal Markov policy of order m. □

Theorem 9.26 implies the correctness of the following algorithm for the computation of an optimal randomized Markov policy of order m for a finite horizon model with constraints.

Algorithm 9.7 *Construction of an optimal Markov policy of order m for the constrained finite horizon problem (9.35)*

Input: Instance of a nonstationary MDP, immediate rewards $c^{l,t}_j(a),\ (j,a)\in S\times A$, $l=0,1,\dots,m,\ 1\le t\le T$ and bounds b_l for $l=1,2,\dots,m$.

Output: Either the statement that (9.35) is infeasible or an optimal Markov policy of order m for problem (9.35)

```
1. Solve the linear program (9.36).

2. if program (9.36) is infeasible then begin problem (9.35) is infeasible; go to step 4 end
   else begin if program (9.36) has optimal solution $x$ then go to step 3 end

3. $R_* := (\pi^1,\pi^2,\dots,\pi^T)$, where $\pi^t_{ja}$ for $(j,a)\in S\times A,\ 1\le t\le T$ is defined by
   $\pi^t_{ja} := \begin{cases} \frac{x^t_{ja}}{\sum_a x^t_{ja}} & \text{if } \sum_a x^t_{ja} > 0\\ 1 & \text{if } \sum_a x^t_{ja} = 0 \text{ and } a = a_j, \text{ where } a_j\in A(j) \text{ is arbitrarily chosen}\\ 0 & \text{if } \sum_a x^t_{ja} = 0 \text{ and } a\ne a_j\end{cases}$
   is an optimal Markov policy of order $m$.

4. STOP
```

Next, we recall some properties of the unconstrained problem with sum of discounted rewards and different discount factors, as discussed in section 7.13. The unconstrained problem is

$$max_R \Big\{ \sum_{k=1}^{K} \sum_{t=1}^{\infty} (\alpha_k)^{t-1} \sum_{j,a} \mathbb{P}_{i,R}\{X_t = j, Y_t = a\} \cdot r_j^k(a) \Big\}. \tag{9.39}$$

Assume, without loss of generality, that the discount factors satisfy $\alpha_1 > \alpha_2 > \cdots > \alpha_K$. Let $v_i^k(R) := \sum_{t=1}^{\infty} (\alpha_k)^{t-1} \sum_{j,a} \mathbb{P}_{i,R}\{X_t = j, Y_t = a\} \cdot r_j^k(a),\ k = 1, 2, \dots, K$. Define the values $v_j^k, j \in S$, the action sets $A_k(j),\ j \in S$ and the policy spaces C_k for $k = 1, 2, \dots, K$ recursively as follows:

$v_j^1 := sup_R\, v_j^1(R),\ j \in S$, the value of the problem with discount factor α_1 and rewards $r_l^1(a)$;

$A_1(j) := \{a \in A(j) \mid v_j^1 = r_j^1(a) + \alpha_1 \cdot \sum_l p_{jl}(a) v_l^1\},\ j \in S;$

C_1 is the set of policies whose actions are in the sets $A_1(j),\ j \in S$.

Given the values $v_j^k,\ j \in S$, the action sets $A_k(j),\ j \in S$ and the policy space C_k, we define:

$v_j^{k+1} := sup_{R \in C_k}\, v_j^{k+1}(R),\ j \in S$, the value of the problem with discount factor α_{k+1} and rewards $r_l^{k+1}(a)$ and action sets $A_k(l)$;

$A_{k+1}(j) := \{a \in A_k(j) \mid v_j^{k+1} = r_j^{k+1}(a) + \alpha_{k+1} \cdot \sum_l p_{jl}(a) v_l^{k+1}\},\ j \in S.;$

C_{k+1} is the set of policies whose actions are in the sets $A_{k+1}(j),\ j \in S$.

We have seen in Theorem 7.24 of section 7.13 that there exists an ultimately deterministic optimal policy $R_* = (\pi^1, \pi^2, \dots, \pi^{T-1}, f, f, \dots)$ for some finite T, where the deterministic rule f can be chosen as arbitrary actions $f(j) \in A_K(j),\ j \in S$.

A set of Markov policies C_* is called a *funnel* if there exits a number T and action sets $A_t(j)$, $j \in S,\ t = 1, 2, \dots, T+1$ such that $R := (\pi^1, \pi^2, \dots) \in C_*$ if the following conditions hold:

(1) for $t = 1, 2, \dots, T$: if $\pi_{ja}^t > 0$ then $a \in A_t(j),\ j \in S$;

(2) for $t \geq T+1$: if $\pi_{ja}^t > 0$ then $a \in A_{T+1}(j),\ j \in S$.

Define a new MDP by:

- state space $\overline{S} := S \times \{1, 2, \dots, T\} \cup S$

- action sets $\overline{A}(z) := \begin{cases} A_t(j) & \text{if } z = (j,t) \text{ for } j \in S \text{ and } 1 \leq t \leq T \\ A_{T+1}(j) & \text{if } z = j \in S \end{cases}$

- transition probabilities $\overline{p}_{zz'}(a) := \begin{cases} p_{jl}(a) & \text{if } z = (j,t),\ z' = (l, t+1) \text{ for } j, l \in S \text{ and } 1 \leq t \leq T-1 \\ p_{jl}(a) & \text{if } z = (j,T),\ z' = l \text{ for } j, l \in S \\ p_{jl}(a) & \text{if } z = j \in S \text{ and } z' = l \in S \\ 0 & \text{otherwise} \end{cases}$

- rewards $\overline{r}_z^k(a) := \begin{cases} r_j^k(a) & \text{if } z = (j,t) \text{ for } j \in S \text{ and } 1 \leq t \leq T \\ r_j^k(a) & \text{if } z = j \in S \end{cases}$

Notice that the set of policies for this model coincides with the funnel C_*. For any subset C' of the set C of all policies we define the following sets:

$v^{lk}(i, C') := \{v^{lk}(i, R) \mid R \in C'\}$ for $l = 0, 1, \dots, m$ and $k = 1, 2, \dots, K$;

$v^l(i,C') := \{v^l(i,R) \mid R \in C'\}$ for $l = 0,1,\dots,m$;

$V(i,C') := \{V(i,R) \mid R \in C'\}$.

For the unconstrained problem ($m = 0$) we use the notation $v^k(i,C')$ instead of the double indexed $v^{0k}(i,C')$ and $v(i,C)'$ for $v^0(i,C)'$; in this case, the one-dimensional set $V(i,C')$ coincides with the set $v(i,C')$.

Lemma 9.21

Consider the unconstrained problem of section 7.13: $max_R\, v(i,R)$, where $v(i,R) := \sum_{k=1}^K\, v^k(i,R)$ with $v^k(i,R) := \sum_{t=1}^\infty (\alpha_k)^{t-1} \sum_{j,a} \mathbb{P}_{i,R}\{X_t = j, Y_t = a\} \cdot r_j^k(a),\ k = 1,2,\dots,K$.

Let C_* be a nonempty funnel and $C_*^0 := \{R_0 \in C_* \mid v(i,R_0) = sup_{R\in C_*}\, v(i,R)\}$.

Then, there is a nonempty funnel $C'_* \subseteq C_*$ such that:

(1) $v(i,R') = sup_{R\in C_*}\, v(i,R)$ for all $R' \in C'_*$.

(2) $\big(v^1(i,C'_*), v^2(i,C'_*),\dots,v^K(i,C'_*)\big) = \big(v^1(i,C_*^0), v^2(i,C_*^0),\dots,v^K(i,C_*^0)\big)$.

Proof

Consider the model $(\overline{S},\overline{A},\overline{p},\overline{r})$ as defined above. Since the set of policies for this model coincides with the funnel C_*, the value of this model with initial state $(i,1)$ equals $sup_{R\in C_*}\, v(i,R)$. From the results of section 7.13 applied to the new model $(\overline{S},\overline{A},\overline{p},\overline{r})$ it follows that there exists a $T' \geq T$ and action sets $A'_t(j),\ j \in S,\ t = 1,2,\dots,T'+1$ such that:

(a) $A'_t(j) \subseteq A_t(j)$ for $j \in S$ and $1 \leq t \leq T$;

(b) $A'_t(j) \subseteq A_{T+1}(j)$ for $j \in S$ and $t = T+1, T+2,\dots,T'+1$;

(c) $R = (\pi^1,\pi^2,\dots) \in C_*^0$ if and only if $\pi_{ja}^t > 0$ implies $a \in A'_t(j)$ for $t = 1,2,\dots,T'$ and $a \in A'_{T'+1}(j)$ for $t \geq T'+1$.

The number T' and the sets $A'_t(j),\ j \in S,\ 1 \leq t \leq T'+1$, define a funnel C'_* and, by (a) and (b), $C'_* \subseteq C_*^0$. Then, by (c), positive probabilities in decision rules correspond to actions from the sets $A'_t(j)$ for all j and t. Hence, $\big(v^1(i,C'_*), v^2(i,C'_*),\dots,v^K(i,C'_*)\big) = \big(v^1(i,C_*^0), v^2(i,C_*^0),\dots,v^K(i,C_*^0)\big)$.

□

The following lemma deals with the constrained problem (9.34), so that $V(i,R)$ is now a vector in R^{m+1}.

Lemma 9.22

For any funnel C_*, the set $V(i,C_*)$ is convex and compact.

Proof

For any funnel C_*, there exists an MDP such that there is a one-to-one correspondence between C_* and the set of policies in the new model. This model is similar to the model $(\overline{S},\overline{A},\overline{p},\overline{r})$ with the only difference that the reward functions are now depending on two indices: $\overline{r}_j^{lk}(a)$ instead of $\overline{r}_j^k(a)$. By Lemma 9.20, the performance region $V(i,C_*)$ is convex and compact. □

Next, we show that, if problem (9.34) has a feasible solution, then for some T there exists an optimal (m,T)-policy $R = (\pi^1, \pi^2, \dots, \pi^T, f, f, \dots)$.

We remind the reader some definitions and properties from convex analysis. A convex subset W of a convex set U is called *extreme* if any representation $u_3 = \lambda u_1 + (1-\lambda)u_2$ with $0 < \lambda < 1$, $u_1, u_2 \in U$ and $u_3 \in W$ is only possible if $u_1, u_2 \in W$. A subset W of U is called *exposed* if there is a supporting hyperplane H of U such that $W = H \cap U$. Extreme and exposed subsets other that U are called *proper*. Any exposed subset of a convex set is extreme, but the converse may not hold (see Stoer and Witzgall [284]).

Lemma 9.23

Let C_* be a funnel and let V be an exposed subset of $V(i, C_*)$. Then, there exists a funnel C'_* such that $V = V(i, C'_*)$.

Proof

Let $\sum_{l=0}^m b_m u_m = b$ be a supporting hyperplane of the convex compact set $V(i, C'_*)$ which contains V and let $\sum_{l=0}^m b_m u_m \leq b$ for every $u = (u_0, u_1, \dots, u_m) \in V(i, C_*)$. Then,

$$\begin{aligned} V &= \big\{u \in V(i, C_*) \mid \textstyle\sum_{l=0}^m b_m u_m = max\{\sum_{l=0}^m b_m u_m \mid u \in V(i, C_*)\}\big\} \\ &= \big\{u \in V(i, C_*) \mid \textstyle\sum_{l=0}^m b_m u_m = max\{\sum_{l=0}^m b_m V(i, R) \mid R \in C*\}\big\} \end{aligned}$$

Therefore, $u \in V$ if and only if $u = \sum_{l=0}^m b_m V(i, R_0)$, where R_0 is an optimal policy for the unconstrained problem with optimality criterion $\sum_{l=0}^m b_m V(i, R)$. By Lemma 9.21, $V = V(i, C'_*)$ for some funnel $C'_* \subseteq C_*$. □

Corollary 9.3

Let V be an exposed subset of $U(i)$. Then, there exists a funnel C_* such that $V = V(i, C'_*)$.

Proof

The set C of all policies is a funnel defined by $T := 0$ and $A_1(j) := A(j),\ j \in S$. □

Lemma 9.24

Let V be a proper extreme subset of $U(i)$. Then, there exists a funnel C_* such that $V = V(i, C_*)$.

Proof

The set $C^0 := C$ is clearly a funnel, defined by $T := 0$ and $A_1(j) := A(j),\ j \in S$, which satisfies $U(i) = V(i, C^0)$. $V^0 := V(i, C^0)$ and assume that, for some $j \in \mathbb{N}_0$, we have a funnel C^j such that V is a proper extreme subset of $V^j := V(i, C^j)$. By Lemma 9.22, the set V^j is convex and compact. Let V^{j+1} be is a proper extreme subset of the convex and compact set $V^j = V(i, C^j)$ such that $V \subseteq V^j$. Then, by Lemma 9.23, there exists a funnel C^{j+1} such that $V^{j+1} = V(i, C^{j+1})$. If $V^{j+1} = V$, the lemma is proved for $C_* := C^{j+1}$.

If $V^{j+1} \neq V$, we increase j by 1 and repeat the construction. By Proposition (3.6.5) and (3.6.3) of Stoer and Witzgall [284], we have $dim(V) \leq dim(V^{j+1}) < dim(V^j)$. Hence, after a finite number of steps we obtain $V^{j+1} = V$. □

Corollary 9.4

Let u be an extreme point of $U(i)$. Then, there exists an ultimately deterministic policy R such that $u = V(i,R)$.

Proof

If $U(i) = u$, then we have $u = V(i,R)$ for any policy R. If $U(i) \neq u$, then u is a proper extreme subset of $U(i)$. By Lemma 9.24, $u = V(i, C_*)$ for some funnel C_*. Let the funnel C_* be generated by T and the action sets $A_t(j),\ j \in S,\ t = 1,2,\dots,T+1$. Then, $u = V(i,R)$ for any policy $R \in C_*$. Since C_* contains an ultimately deterministic policy R, the corollary is proved. □

Define for two points $u = (u_0, u_1, \dots, u_m)$ and $v = (v_0, v_1, \dots, v_m)$ in $\mathbb{R}m+1$ the distance $d(u,v)$ by $d(u,v) := \sum_{l=0}^m |u_i - v_i|$ (L_1-norm).

Lemma 9.25

Let V be either an exposed subset or a proper extreme subset of $U(i)$. Then, there exists a policy $f^\infty \in C(D)$ with the following property: for every $\varepsilon > 0$ there exists $T \in \mathbb{N}$ such that for any $u \in V$ there exists an element $v \in V$ satisfying $d(u,v) \leq \varepsilon$ and $v = V(i,R)$ for some policy $R = (\pi^1, \pi^2, \dots, \pi^T, f, f, \dots)$.

Proof

By the lemmas 9.23 and 9.24, $V = V(i, C_*)$ for some funnel C_*. Let C_* be generated by $T_* \in \mathbb{N}_0$ and the action sets $A_t(j),\ j \in S,\ t = 1,2,\dots,T_*+1$ and let f^∞ be such that $f(j) \in A_{T_*+1}(j)$ for all $j \in S$.

Define $\alpha := max_{1 \leq k \leq K}\, \alpha_k$ and $M := max\,\{|r_j^{lk}(a)| \mid 0 \leq l \leq m;\ 1 \leq k \leq K;\ (j,a) \in S \times A$. Note that $\alpha \in [0,1)$. If two policies $R_1 = (\pi^1, \pi^2, \dots)$ and $R_2 = (\sigma^1, \sigma^2, \dots)$ are such that $\pi^t = \sigma^t$ for $t = 1,2,\dots,n$, then, $|V^l(i,R_1) - V^l(i,R_2)| \leq 2KM \cdot \frac{\alpha^n}{1-\alpha}$ for all $l = 0,1,\dots,m$. Hence, $|V(i,R_1) - V(i,R_2)| \leq 2(m+1)KM \cdot \frac{\alpha^n}{1-\alpha}$. Given $\varepsilon > 0$, choose $T \geq T_*$ such that $2(m+1)KM \cdot \frac{\alpha^T}{1-\alpha} \leq \varepsilon$. Then, for any two policies $R_1 = (\pi^1, \pi^2, \dots)$ and $R_2 = (\sigma^1, \sigma^2, \dots)$ such that $\pi^t = \sigma^t$ for $t = 1,2,\dots,T$, we have $d\big(V(i,R_1), V(i,R_2)\big) \leq \varepsilon$.

Take any $u \in V$ and consider a policy $R_1 = (\sigma^1, \sigma^2, \dots) \in C_*$ such that $u = V(i,R_1)$. Define the policy $R = (\pi^1, \pi^2, \dots)$ by $\pi^t := \sigma^t$ for $t = 1,2,\dots,T$ and $\pi^t := f$ for $t \geq T+1$, and define $v := V(i,R)$. Since $R \in C_*$, we have $v \in V$ and, furthermore, we have $d(u,v) \leq \varepsilon$. Therefore, the conditions (1) and (2) are satisfied. □

Theorem 9.27

Let V be either an exposed subset or a proper extreme subset of $U(i)$. Then, for any $u \in V$ there exist a Markov policy $R = (\pi^1, \pi^2, \dots)$, a deterministic policy f^∞ and an integer T such that $u = V(i,R)$ and $\pi^t = f$ for every $t \geq T+1$.

Proof

Take any $u \in V$. Since any intersection of extreme sets is an extreme set and any intersection of closed sets is a closed set, there exists a minimal closed extreme subset W of $U(i)$ containing

u. This set W is the intersection of all closed extreme subsets of $U(i)$ containing u. If V is an exposed set, it is extreme (see Stoer and Witzgall [284], but it is possible that $V = U(i)$.

Let $dim(W) = n \leq m+1$. By Caratheodorys theorem, u is a convex combination of $n+1$ extreme points $u^1, u^2, \dots, u^{n+1}$ of W. The minimality of W implies that the convex hull of $\{u^1, u^2, \dots, u^{n+1}\}$ is a simplex and u is an inner point of this simplex.

We select $\varepsilon > 0$ small enough so that if $\{v^1, v^2, \dots, v^{n+1}\} \subseteq W$ and each v^j belongs to the ε-neighborhood of u^j for $j = 1, 2, \dots, n+1$. Then, the following property holds: the convex hull of $\{v^1, v^2, \dots, v^{n+1}\}$ is a simplex and u belongs to this simplex, say $u = \sum_{j=1}^{n+1} \lambda_j v^j$ for some λ with $\lambda_j \geq 0$ for all j and $\sum_{j=1}^{n+1} \lambda_j = 1$.

W is either a proper extreme subset of $U(i)$ or $W = V = U(i)$ and W is an exposed subset. By Lemma 9.25, there exists a policy $f^\infty \in C(D)$, an integer T and policies $R_p = (\pi^{1p}, \pi^{2p}, \dots)$ such that $v^j = V(i, R_j)$ and $\pi^{tp} = f$ for all $t \geq T+1$ and $p = 1, 2, \dots, n+1$.

Hence, $u = \sum_{j=1}^{n+1} \lambda_j V(i, R_j)$. By Theorem 1.1, there exists a Markov policy $R = (\pi^1, \pi^2, \dots)$ such that $u = V(i, R)$ and, because the policies R_j have identical decision rules from stage $T+1$, $\pi^t = f$ for $t \geq T+1$. □

Corollary 9.5

Let u be a Pareto optimal point of $U(i)$. Then, there exists a Markov policy $R = (\pi^1, \pi^2, \dots)$, a deterministic policy f^∞ and an integer T such that $u = V(i, R)$ and $\pi^t = f$ for every $t \geq T+1$.

Proof

We consider two situations: (1) $dim\big(U(i)\big) \leq m$ and (2) $dim\big(U(i)\big) = m+1$.

If $dim\big(U(i)\big) \leq m$, then $U(i)$ is an exposed set and the result follows from Theorem 9.27.

If $dim\big(U(i)\big) = m+1$, then the Pareto optimal point u belongs to the boundary of $U(i)$ and, consequently, u belongs to some proper extremal subset of $U(i)$. Also in this case the result follows from Theorem 9.27. □

Theorem 9.28

If problem (9.34) is feasible, then there exists an optimal (m, T)-policy.

Proof

Assume that problem (9.34) is feasible. From the remark after Lemma 9.20 it follows that there exists an optimal policy, say R_*. Furthermore, it follows from this remark that there exists a Pareto optimal point $u \in U(i)$ such that either $u = V(i, R_*)$ or u dominates $V(i, R_*)$. Any policy R such that $V(i, R) = u$ is optimal. By Corollary 9.5, there exists a Markov policy a Markov policy $R = (\sigma^1, \sigma^2, \dots)$, a deterministic policy f^∞ and an integer T such that $u = V(i, R)$ and $\sigma^t = f$ for every $t \geq T+1$.

In order to find an optimal (m, T)-optimal one has to solve a finite horizon problem with nonstationary one-step rewards $\sum_{k=1}^K \alpha_k^{t-1} r^{lk}(a),\ (j, a) \in S \times A,\ 0 \leq l \leq m$, for $t = 1, 2, \dots, T-1$ and $\sum_{k=1}^K \{\alpha_k^{T-1} r^{lk}(a) + \alpha_k^T \sum_s p_{js}(a) v_s^{lk}(f^\infty)\},\ (j, a) \in S \times A,\ 0 \leq l \leq m$, for $t = T$.

Let $(\pi^1, \pi^2, \dots, \pi^T)$ be a randomized Markov policy of order m, optimal for this finite horizon model (see Algorithm 9.7 for the computation of such policy). Then, $R = (\pi^1, \pi^2, \dots, \pi^T, f, f, \dots)$ is an optimal (m, T)-policy. □

Next, we shall prove that, given any point u^* on the boundary of the performance set $U(i)$, there exists an (m, T)-policy R_* such that $V(i, R_*) = u^*$. This result implies that for any Pareto optimal policy there exists an equivalent (m, T)policy. We also show that for any policy there exists an equivalent (m, T)-policy. The proofs are based on Theorem 9.28 and the following lemma.

Lemma 9.26

Let $U \subseteq \mathbb{R}^{m+1}$ be convex and compact, and let u^* on the boundary of U. Then, there exist constants a_{lp}, $l, p = 0, 1, \dots, m$ and constants b_l, $l = 1, 2, \dots, m$, such that u^* is the unique solution of the problem

$$max \left\{ \sum_{p=0}^{m} a_{0p} u_p \;\middle|\; \sum_{p=0}^{m} a_{lp} u_p \geq b_l,\ l = 1, 2, \dots, m;\ (u_0, u_1, \dots, u_m) \in U \right\}. \tag{9.40}$$

Proof

Let $\sum_{p=0}^m d_p u_p = b$ be a supporting hyperplane H_0 of U which contains the point u^* and also satisfies $\sum_{p=0}^m d_p u_p \leq b$ for all $u = (u_0, u_1, \dots, u_m) \in U$. We consider for $l = 1, 2, \dots, m$ the hyperplanes H_l, defined by $\sum_{p=0}^m a_{lp} u_p = b_l$ such that $\cap_{l=0}^m H_l = \{u^*\}$. Then, u^* is a vertex of the polyhedron

$$U^* := \{u \mid \textstyle\sum_{p=0}^m a_{lp} u_p \geq b_l,\ l = 1, 2, \dots, m;\ \sum_{p=0}^m d_p u_p \leq b\}.$$

Let $\sum_{p=0}^m a_{0p} u_p = b_0$ be a hyperplane that supports U^* at u^* and satisfies $\sum_{p=0}^m a_{0p} u_p < b_0$ for all $u \in U^* \backslash \{u^*\}$ (this hyperplane exists because u^* is a vertex of U^*). Hence, u^* is the unique solution of (9.40). □

Theorem 9.29

Given any u^* on the boundary of $U(i)$, there exist an (m, T)-policy R_* such that $V(i, R_*) = u^*$.

Proof

Apply Lemma 9.26 with $U = U(i)$. Then, u^* is the unique solution of (9.40). Since we have $\sum_{p=0}^m a_{lp} v^p(i, R) = \sum_{k=1}^K \sum_{p=0}^m a_{lp} v^{pk}(i, R)$, also $\sum_{p=0}^m a_{lp} v^p(i, R) = \sum_{k=1}^K \overline{v}^{lk}(i, R) = \overline{v}^l(i, R)$, where $\overline{v}^{lk}(i, R)$ is the expected discounted reward with discount factor α_k and one-step rewards $\overline{r}_j^{lk}(a) := \sum_{p=0}^m a_{lp} r_j^{lk}(a)$, $(j, a) \in S \times A$ for all l, k. If we apply Theorem 9.27 to this new MDP model, we obtain the existence of an optimal (m, T)-policy R_*. By the uniqueness of u^*, we obtain $V(i, R_*) = u^*$. □

Corollary 9.6

If R is a Pareto optimal policy, there exists an optimal (m, T)-policy.

Proof
Any Pareto optimal policy of a convex and compact set belongs to the boundary. Now, the corollary follows from Theorem 9.29. □

Lemma 9.27
Let $U \subseteq \mathbb{R}^{m+1}$ be convex and compact. Then, for any $u^* \in U$ there exist constants a_{lp}, $l = 0, 1, \dots, m+1$, $p = 0, 1, \dots, m$ and constants b_l, $l = 1, 2, \dots, m+1$, such that u^* is the unique solution of the problem

$$max \left\{ \sum_{p=0}^{m} a_{0p}u_p \;\Big|\; \sum_{p=0}^{m} a_{lp}u_p \geq b_l,\; l = 1, 2, \dots, m+1;\; (u_0, u_1, \dots, u_m) \in U \right\}. \tag{9.41}$$

Proof
We consider the hyperplane H, defined by $\sum_{p=0}^{m} a_{m+1,p}u_p = b_{m+1}$, such that u^* belongs to this plane. Let $U^* := U \cap H$. Then, U^* is convex and compact and u^* belongs to the boundary of U^*. Apply Lemma 9.26 to the set U^* and the point u^*. □

Corollary 9.7
For any policy R there exists an optimal $(m+1, T)$-policy R_* with $V(i, R_*) = V(i, R)$.

Proof
The proof is similar to the proof of Theorem 9.29, but we apply Lemma 9.27 instead of Lemma 9.26. □

The following example illustrates that $m+1$ cannot be replaced with m in Collorally 9.7.

Example 9.14
Let $K = 1,\ m = 0,\ \alpha_1 = 0.5$; $S = \{1\}$; $A(1) = \{1, 2\}$; $p_{11}(1) = p_{11}(2) = 1$; $r_1^{01}(1) = 0,\ r_1^{01}(2) = 1$. Then, $U_1 = [0, 2]$. If R is a $(0, T)$-policy, then there all decision rules are deterministic and therefore $V_1(R)$ is a rational number for all $(0, T)$-policies. Hence, if $V_1(R)$ is an irrational number, at least one decision rule is randomized and we need a $(1, T)$-policy to obtain this performance.

We close this section with an algorithm for an approximate solution of problem (9.34). We say that, given $\varepsilon > 0$, a policy R_* is ε-optimal for problem (9.34) if this policy is feasible and $v^0(i, R_*) \geq v^0(i, R) - \varepsilon$ for all feasible policies R. A policy R_* is called *approximately ε-optimal* if R_* is ε-optimal and $v^l(i, R) \geq b_l - \varepsilon$ for all $l = 1, 2, \dots, m$. We remark that that an approximately ε-optimal policy may be infeasible. However, from a practical point of view, it is sufficient to find an approximately ε-optimal policy for some small positive ε.

Algorithm 9.8

Construction of an ε-optimal or approximately ε-optimal (m,T)-policy for problem (9.34)

Input: Instance of an MDP, a tolerance $\varepsilon > 0$, integers m and K, discount factors $\alpha_k,\ 1 \leq k \leq K$, bounds b_l for $l = 1,2,\dots,m$ and immediate rewards $r_j^{lk}(a),\ (j,a) \in S \times A,\ 0 \leq l \leq m$, $1 \leq k \leq K$.

Output: Either a ε-optimal or an approximately ε-optimal (m,T)-policy for problem (9.34).

1. Select an arbitrary policy $f^\infty \in C(D)$.

2. Select $T \in \mathbb{N}$ such that $\frac{K \cdot L \cdot \alpha^T}{1-\alpha} \leq \varepsilon$,
   where $\alpha := max_{1 \leq k \leq K}\, \alpha_k$ and $L := M - min\{r_j^{lk}\big(f(j)\big) \mid 0 \leq l \leq m;\ 1 \leq k \leq K;\ j \in S\}$ with $M := max\{|r_j^{lk}(a) \mid 0 \leq l \leq m;\ 1 \leq k \leq K;\ (j,a) \in S \times A\}$.

3. Apply Algorithm 9.7 to the finite horizon problem (9.35), where the one-step rewards are $\sum_{k=1}^K \alpha_k^{t-1} r^{lk}(a),\ (j,a) \in S \times A,\ 0 \leq l \leq m$, for $t = 1,2,\dots,T-1$ and
   $\sum_{k=1}^K \{\alpha_k^{T-1} r^{lk}(a) + \alpha_k^T \sum_s p_{js}(a) v_s^{lk}(f^\infty)\},\ (j,a) \in S \times A,\ 0 \leq l \leq m$, for $t = T$.

4. If the finite horizon problem is feasible, let $(\pi^1,\pi^2,\dots,\pi^T)$ be an optimal Markov policy of order m, obtained by Algorithm 9.7. The policy $R_* := (\pi^1,\pi^2,\dots,p^T,f,f,\dots)$ is an ε-optimal (m,T)-policy (STOP).

5. If the finite horizon problem is infeasible, consider a similar finite horizon problem with the constants b_l in the right-hand-side of (9.34) replaced by $b_l - \varepsilon$ for $l = 1,2,\dots,m$.

6. If the new finite horizon problem is feasible, let $(\pi^1,\pi^2,\dots,\pi^T)$ be an optimal Markov policy of order m, obtained by Algorithm 9.7 applied to this new problem. The policy $R_* := (\pi^1,\pi^2,\dots,p^T,f,f,\dots)$ is an approximately ε-optimal (m,T)-policy (STOP).

7. If the new finite horizon problem is infeasible, there does not exist an approximately ε-optimal (m,T)-policy (STOP).

9.2.8 Constrained discounted MDPs with two discount factors

In this section we consider an MDP, where the objectives are linear combinations of discounted rewards, each with a different discount factor. For the special case where a standard discounted reward function is to be maximized, subject to a constraint on another standard discounted reward function but with a different discount factor, we provide an implementable algorithm for computing an optimal policy.

Example 9.15

Consider the problem of managing a computer facility with many users. The objective is to provide acceptable service, while spending as little as possible. The state i encodes information about the number of users and about available computer resources that influence the performance of the system, such as RAM memory, computation power, disk space, etc.

Consider the simple case in which the only available decision is to add a gigabyte of disk space. Suppose that the cost per gigabyte of disk storage at time $t = 1$ is c. The price of disk space decreases over time and let $\alpha_1 \in [0, 1)$ denote the rate of decrease per time unit.

Let $c_i(a)$ be a combined measure of performance, when the state is i and action a is chosen. To model the fact that demands from computer performance as a whole increase at an exponential rate, we let $\alpha_2 \in [0, 1)$ be the ratio between required performances at consecutive decision epochs. Then, maintaining 'adequate performance' in the long run may be modeled by the requirement that, for some appropriate constant B, $\mathbb{E}_{i,R}\,\{\sum_{t=1}^{\infty}\,\alpha_2^{t-1}c_{X_t}(Y_t)\} \geq B$. Combining the different criteria we arrive at the following optimization problem

$$min\Big\{\,\mathbb{E}_{i,R}\,\{\sum_{t=1}^{\infty}\,\alpha_1^{t-1}c\cdot Y_t\}\;\Big|\;\mathbb{E}_{i,R}\,\{\sum_{t=1}^{\infty}\,\alpha_2^{t-1}c_{X_t}(Y_t)\} \geq B\Big\}. \tag{9.42}$$

In this section we are interested in solving the following constrained optimization problem, given some bound B:

$$max\Big\{\,\mathbb{E}_{i,R}\,\{\sum_{t=1}^{\infty}\,\alpha_1^{t-1}r^1_{X_t}(Y_t)\}\;\Big|\;\mathbb{E}_{i,R}\,\{\sum_{t=1}^{\infty}\,\alpha_2^{t-1}r^2_{X_t}(Y_t)\} \geq B\Big\}, \tag{9.43}$$

where $\alpha_1 \neq \alpha_2$ (for the case $\alpha_1 = \alpha_2$ we refer to section 9.2.2). Denote $\mathbb{E}_{i,R}\,\{\sum_{t=1}^{\infty}\,\alpha_1^{t-1}r^1_{X_t}(Y_t)\}$ and $\mathbb{E}_{i,R}\,\{\sum_{t=1}^{\infty}\,\alpha_1^{t-1}r^2_{X_t}(Y_t)\}$ by $v^1(i,R)$ and $v^2(i,R)$, respectively. Notice that in problem (9.43) the initial state i is fixed.

Suppose that problem (9.43) is feasible. The problem of optimizing $v^k(i,R)$ is for each $k = 1, 2$ a standard discounted MDP. So, checking $max_R\, v^2(i,R) \geq B$ can be easily verified by solving a discounted MDP. Let $A_k(j)$ for $k = 1$ or $k = 2$ be the set of conserving actions in state j for the corresponding problem; these are the actions that achieve the maximum in the optimality equation. We know from the general theory of discounted MDPs that $max_R\, v^k(i,R)$ is obtained for any policy that takes actions from $A_k(j),\ j \in S$. Call a policy R_* (1,2)-*lexicographic-optimal* at state i if $v^1(i,R_*) = max_R\, v^1(i,R)$ and $v^2(i,R_*) = max_{R\in C^1}\, v^2(i,R)$, where C^1 is the subset of policies that takes actions from $A_1(j),\ j \in S$. Similarly, we define the notion of (2,1)-*lexicographic-optimality.*

The computation of an (1,2)-lexicographic-optimal policy proceeds as follows. First, compute $A_1(j)$ for all $j \in S$. Then, solve the problem of maximizing $v^2(i,R)$ over C^1, i.e. the discounted MDP with discount factor α_2, action sets $A_1(j),\ j \in S$ and one-step rewards $r^2_j(a),\ (j,a) \in S \times A_1$. Let $A_{1,2}$ be the set of conserving actions in this restricted MDP.

Theorem 9.30

(1) If a (1,2)-lexicographic-optimal policy R_* is such that $v^2(i, R_*) \geq B$, then R_* is an optimal policy for problem (9.43).

(2) If $max_R\, v^2(i, R) = B$, then a (2,1)-lexicographic-optimal policy R_* is an optimal policy for problem (9.43).

Proof

(1) Since an optimal policy R_* for $max_R\, v^1(i, R)$ satisfies, by assumption, the constraint of problem (9.43), it is an optimal policy for problem (9.43).

(2) By the hypothesis of this part of the theorem, only policies with actions in $A_2(j),\ j \in S$, are feasible. By the properties of standard discounted MDPs a (2,1)-lexicographic-optimal policy R_* provides the maximum value among all feasible policies. □

We now consider the cases that not occur in Theorem 9.30. Therefore, we define for any $0 < \lambda < 1$ and any $R \in C$ the vector $w^\lambda(R)$ by

$$w^\lambda(R) := \lambda v^1(R) + (1 - \lambda) v^2(R). \tag{9.44}$$

Assume that $\alpha_1 > \alpha_2$. Then, as shown in section 7.13, there exists an ultimately deterministic optimal policy $R_* = (f_1, f_2, \dots, f_t, f, f, \dots)$ for the problem $max_R\, w^\lambda(R)$ (take as one-step-rewards $\lambda r^1_j(a) + (1 - \lambda) r^2_j(a)$) such that f^∞ is a (1,2)-lexicographic-optimal policy.

We recall from section 7.13 the following. Let v^k be the value vector of the discounted problem with discount factor α_k and one-step-rewards $r^k_j(a),\ (j, a) \in S \times A$. Let $\underline{v}^k$ be such that $v^k(R) \geq \underline{v}^k$ for all $R \in C$. Define S_1, ε and T as follows:

$S_1 := \{i \in S \mid A_1(i) \neq A(i)\}$;

$$\varepsilon := \begin{cases} \lambda \cdot min_{i \in S_1} \left\{ v^1_i - max_{a \in A(i) \setminus A_1(i)} \left\{ r^1_i(a) + \alpha_1 \cdot \sum_j p_{ij}(a) v^1_j \right\} \right\} & \text{if } S_1 \neq \emptyset; \\ 0 & \text{otherwise;} \end{cases}$$

$$T := \begin{cases} min \left\{ t \geq 1 \mid \left(\frac{\alpha_1}{\alpha_2}\right)^{t-1} \cdot max_i\, (v^2_i - \underline{v}^2_i) < \frac{\varepsilon}{1 - \lambda} \right\} & \text{if } \varepsilon > 0; \\ 1 & \text{if } \varepsilon = 0. \end{cases}$$

Then, we see that ε, and consequently also T, depends on λ, so we write $\varepsilon(\lambda)$ and $T(\lambda)$. Furthermore, we see that $\varepsilon(\lambda)$ is increasing in λ and $T(\lambda)$ decreasing in λ. The case $\alpha_1 < \alpha_2$ is similar. Therefore, we have established the following result.

Theorem 9.31

There exists an ultimately deterministic optimal policy $R_* = (f_1, f_2, \dots, f_{T(\lambda)}, f, f, \dots)$ for the problem $max_R\, w^\lambda(R)$ such that:

(1) if $\alpha_1 > \alpha_2$, then f^∞ is a (1,2)-lexicographic-optimal policy and $T(\lambda)$ is decreasing in λ;

(2) if $\alpha_1 < \alpha_2$, then f^∞ is a (2,1)-lexicographic-optimal policy and $T(\lambda)$ is increasing in λ.

As in section 9.2.7, define for a fixed initial state i the performance region $U(i)$ as follows: $U(i) := \{v^1(i,R), v^2(i,R) \mid R \in C\}$. Recall that $u = (u_1, u_2) \in R^2$ is called *Pareto optimal* in a set $U \in R^2$ if $v \in U$ and $v_i \geq u_i$ for $i = 1, 2$ imply $v = u$. We use the following well known lemma.

Lemma 9.28

Let $U \subseteq \mathbb{R}^2$ be convex and compact. Consider the following optimization problem

$$max\,\{b_1 v_1 + b_2 v_2 \mid v \in U\}. \tag{9.45}$$

(1) if $u \in U$ is Pareto optimal, then u is an optimal solution of (9.45) for some $b \in R^2_+$ with $b_1 + b_2 > 0$;

(2) if u is an optimal solution of (9.45) for some $b \in R^2_+$ with $b_1 > 0$ and $b_2 > 0$, then u is Pareto optimal.

If the conditions (1) and (2) of Theorem 9.30 do not hold, the convexity of $U(i)$ implies that the optimal solution of (9.43) is obtained at a point $v = \big(v^1(i,R), v^2(i,R)\big)$ with $v^2(i,R) = B$ and moreover, the slope of the normal to any tangent hyperplane to $U(i)$ at v is bounded away from 0 and ∞. But this implies that an optimal solution may be found by solving $max_R\, w^\lambda(R)$, where $w^\lambda(R) = \lambda v^1(R) + (1-\lambda)v^2(R)$ for some $0 < \lambda < 1$, say for $\lambda = \lambda_*$.

From Lemma 9.6 and Theorem 9.17 we also obtain the following results: (1) the sets $U(i)$ are compact for all $i \in S$; (2) if (9.43) is feasible, then there exists an optimal $(1,T)$ policy.

The presence of a single constraint implies a single randomization. Property (2) says that an optimal policy exists with in at most one state and at most one point in time a random action is necessary, and furthermore, at most two actions need to be chosen with positive probability at this time-state pair. Otherwise, the decision rules are deterministic.

Define $v_k(1,2)$ and $v_k(2,1)$ by $v_k(1,2) := max_R\,\{v^k(i,R) \mid R$is (1,2)-lexicographic-optimal$\}$ and $v_k(2,1) := max_R\,\{v^k(i,R) \mid R$is (2,1)-lexicographic-optimal$\}$, for $k = 1, 2$. We have already mentioned that the computation of each $v_k(1,2)$ and $v_k(2,1)$, for $k = 1, 2$, can be done by the solution of two standard discounted MDPs.

Lemma 9.29

If the condition of Theorem 9.30 (1) does not hold, i.e. $v^2(i,R) < B$ for any (1,2)-lexicographic-optimal policy, then if λ_0 is such that $(\lambda_0, 1-\lambda_0)$ is the normal to the line that connects the points $\big(v_1(1,2), B\big)$ and $\big(v_1(2,1), v_2(2,1)\big)$ in the (v_1, v_2)-space, is a lower bound on λ_*.

Proof

Since $v_2(1,2) < B$, the point $\big(v_1(1,2), B\big)$ is outside $U(i)$. The point $\big(v_1(2,1), v_2(2,1)\big)$ is on the boundary of $U(i)$. From the convexity of $U(i)$ it follows that the boundary of $U(i)$ must cross the horizontal line $v_2 = B$ at a point (v_1, v_2) with $v_1 < v_1(1,2)$. Let λ_0 be such that $(\lambda_0, 1-\lambda_0)$ is the normal to the line that connects the points $\big(v_1(1,2), B\big)$ and $\big(v_1(2,1), v_2(2,1)\big)$. Then, from the geometry in the (v_1, v_2)-space it follows that $\lambda_0 \leq \lambda_*$. □

The bound λ_0 suggest the following algorithm.

Algorithm 9.9 *Computation of a* $(1, T)$*-policy*

Input: Instance of an MDP, two discount factors α_1 and α_2, where $\alpha_1 \neq \alpha_2$, a bound B and two sets of immediate rewards $r^1_j(a)$ and $r^1_j(a)$, $(j, a) \in S \times A$.

Output: An optimal $(1, T)$-policy for problem (9.43).

1. Compute $v_1(1, 2)$, $v_2(1, 2)$, $v_1(2, 1)$ and $v_2(2, 1)$ by solving the corresponding MDPs.
2. Compute $\lambda_0 \in (0, 1)$ such that $(\lambda_0, 1 - \lambda_0)$ is the normal to the line that connects the points $\big(v_1(1, 2), B\big)$ and $\big(v_1(2, 1), v_2(2, 1)\big)$.
3. Compute $T = T(\lambda_0)$ as in the steps 1 until 4 of Algorithm 7.7.
4. Using Algorithm 9.7, compute an optimal randomized Markov policy $(\pi^1, \pi^2, \dots, \pi^T)$ of order 1.
5. Let $R_* := (\pi^1, \pi^2, \dots, \pi^T, f, f, \dots)$, where f^∞ is a (1,2)-lexicographic-optimal policy.

Theorem 9.32

If $\alpha_1 > \alpha_2$*, then the policy* R_**, obtained by Algorithm 9.9, is an optimal* $(1, T)$*-policy for problem (9.43).*

Proof

Since, by Theorem 9.31, $T(\lambda)$ is decreasing in λ, it follows from $\lambda_0 \leq \lambda_*$ that $T(\lambda_0) \geq T(\lambda_*)$. Therefore, policy R_*, obtained by Algorithm 9.9, is an optimal $(1, T)$-policy for problem (9.43).

□

For the case $\alpha_1 < \alpha_2$, we need an upper bound λ_1 on λ_*. The search algorithm below provides such upper bound.

Algorithm 9.10 *Computation of* $\lambda_1 \geq \lambda_*$

Input: Instance of an MDP, two discount factors α_1 and α_2, where $\alpha_1 \neq \alpha_2$, a bound B and two sets of immediate rewards $r^1_j(a)$ and $r^1_j(a)$, $(j, a) \in S \times A$.

Output: λ_1 satisfying $\lambda_1 \geq \lambda_*$.

1. Compute $v_1(1, 2)$, $v_2(1, 2)$, $v_1(2, 1)$ and $v_2(2, 1)$ by solving the corresponding MDPs.
2. Compute $\lambda_1 \in (0, 1)$ such that $(\lambda_1, 1 - \lambda_1)$ is the normal to the line that connects the points $\big(v_1(1, 2), v_2(1, 2)\big)$ and $\big(v_1(2, 1), v_2(2, 1)\big)$.
3. Compute an optimal policy f_1^∞ for the problem $max_R\, w^{\lambda_1}(R)$, where $w^{\lambda_1}(R)$ is defined in (9.44).
4. **if** $v^2(i, f^\infty) \geq B$ **then** STOP

   else begin $\lambda_1 := \frac{\lambda_1 + 1}{2}$; **go to** step 3 **end**

Remark

Algorithm 9.10 terminates in a finite number of steps. This follows from the property that, if $v^2(i, f_1^\infty) < B$, the slope of the normal to any tangent hyperplane to $U(i)$ at $\big(v_1(1,2), B\big)$ is bounded away from 0 and ∞. Since, by Theorem 9.31, $T(\lambda)$ is increasing in λ, it follows from $\lambda_1 \geq \lambda_*$ that $T(\lambda_1) \geq T(\lambda_*)$. Therefore, the policy R_*, obtained by Algorithm 9.9 with the adaptations to take λ_1 instead of λ_0 in steps 2 and 3 and in step 5 that f^∞ is a (2,1)-lexicographic-optimal policy, is an optimal $(1, T)$-policy for problem (9.43).

To conclude, we have established the following finite algorithm for computing an optimal $(1, T)$-policy for problem (9.43).

Algorithm 9.11 *Computation of an optimal* $(1, T)$*-policy for problem (9.43)*

Input: Instance of an MDP, two discount factors α_1 and α_2, where $\alpha_1 \neq \alpha_2$, a bound B and two sets of immediate rewards $r_j^1(a)$ and $r_j^1(a)$, $(j, a) \in S \times A$.

Output: An optimal $(1, T)$-policy for problem (9.43).

```
1. Compute a (1,2)-lexicographic-optimal policy $f_*^\infty \in C(D)$.
2. if $v^2(i, f_*^\infty) \geq B$ then begin $f_*^\infty$ is an optimal $(1, T)$-policy for problem (9.43); STOP end
   else go to step 3
3. Compute a (2,1)-lexicographic-optimal policy $f_*^\infty \in C(D)$.
4. if $v^2(i, f_*^\infty) = B$ then begin $f_*^\infty$ is an optimal $(1, T)$-policy for problem (9.43); STOP end
   else go to step 5
5. if $\alpha_1 > \alpha_2$ then
      begin compute an optimal $(1, T)$-policy for problem (9.43) by Algorithm 9.9; STOP end
   else go to step 6
6. Compute $\lambda_1 \geq \lambda_*$ by Algorithm 9.10.
7. Use Algorithm 9.9, with $\lambda_1$ instead of $\lambda_0$ in steps 2 and 3 and in step 5 that $f^\infty$ is a
   (2,1)-lexicographic-optimal policy, in order to compute an optimal $(1, T)$-policy for problem
   (9.43).
8. STOP.
```

9.3 Multiple objectives

For some problems we may have several sorts of rewards or costs, which we may not be able to optimize simultaneously. Assume that we want to maximize some utility for an m-tuple of immediate rewards, say utilities $u^k(R)$ and immediate rewards $r_i^k(a)$, $(i, a) \in S \times A$, for

$k = 1, 2, \ldots, m$. For each k one can find an optimal policy R_k, i.e. $u_i^k(R_k) \geq u_i^k(R),\ i \in S$, for all policies R. However, in general, $R_k \neq R_l$ if $k \neq l$, and there does not exist one policy which is optimal for all m rewards simultaneously for all starting states. Therefore, we consider the utility function with respect a given initial distribution β. Given this initial distribution β and a policy R, we denote the utilities by $u^k(\beta, R)$. The goal in multi-objective optimization is to find an *β-efficient solution*, i.e. a policyR_* such that there exists *no other policy R* satisfying

$$u^k(\beta, R) \geq u^k(\beta, R_*) \text{ for all } k \text{ and } u^k(\beta, R) > u^k(\beta, R_*) \text{ for at least one } k.$$

We shall consider multiple objectives for both discounted rewards and average rewards. In order to solve these problems, we shall use multi-objective linear programming. Therefore, we first present some properties of multi-objective linear programming.

9.3.1 Multi-objective linear programming

In this section we shall derive some result for the general multi-objective linear program

$$max\{Rx \mid Ax = b;\ x \geq 0\}, \tag{9.46}$$

where $R \in \mathbb{R}^{p \times n},\ x \in \mathbb{R}^n,\ A$ an $m \times n$ matrix and $b \in \mathbb{R}^m$. We define the following sets:

$X := \{x \in \mathbb{R}^n \mid Ax = b;\ x \geq 0\}$, the space of feasible solutions.

$Y := \{y = Rx \in \mathbb{R}^p \mid x \in X\}$, the space of the values of the objective functions.

For $x^1, x^2 \in X$ we say that x^1 is *dominated* by x^2 if $Rx^2 \geq Rx^1$ and $Rx^2 \neq Rx^1$. Let $D(x)$ be the subset of X consisting of the points that are dominated. We say that x^1 is *efficient* if there is no $x^2 \in X$ such that x^2 dominates x^1. The set of efficient points of X is denoted by $E(X)$. Obviously $D(X)$ and $E(X)$ is a partition of X, i.e. $D(X) \cup E(X) = X$ and $D(X) \cap E(X) = \emptyset$.

Lemma 9.30
Suppose that $x^1, x^2 \in X$ and $x^1 \in D(X)$. Then, the half-closed interval $[x^1, x^2) \in D(X)$, where $[x^1, x^2) := \{x \in X \mid x = \lambda x^1 + (1-\lambda)x^2,\ 0 < \lambda \leq 1\}$.

Proof
Since $x^1 \in D(X)$, there is an element $x^3 \in X$ such that $Rx^3 \geq Rx^1$ and $Rx^3 \neq Rx^1$. Let $x^4 \in [x^1, x^2)$, say $x^4 = \lambda_* x^1 + (1-\lambda_*)x^2$ for some $0 < \lambda_* \leq 1$. Define x^5 by $x^5 := \lambda_* x^3 + (1-\lambda_*)x^2$. Then, $x^4, x^5 \in X$ and $Rx^5 - Rx^4 = \lambda_*(Rx^3 - Rx^1) \geq 0$ and $Rx^5 - Rx^4 \neq 0$. □

Corollary 9.8
The set $D(X)$ is convex.

Theorem 9.33
Assume that X is bounded. Let $Ext(X)$ be the set of extreme efficient points of X. Then, $E(X) \subseteq \overline{Ext(X)}$, where $\overline{Ext(X)}$ is the closed convex hull of $Ext(X)$.

Proof

Suppose that x cannot be written as a convex combination of points of $Ext(X)$. It suffices to show that $x \in D(X)$.

Let X_{ext} be the finite set of extreme points of X. By the assumption there is at least one point $x^1 \in D(X) \cap X_{ext}$ and a scalar $\lambda_1 \in [0,1)$ such that $x = \sum_{k=1}^{r} \lambda_k x^k$ with $\sum_{k=1}^{r} \lambda_k = 1,\ \lambda_k \geq 0$, $1 \leq k \leq r$ and $x^k \in Ext(X),\ 1 \leq k \leq r$.

If $\lambda_k = 1$ for some $1 \leq k \leq r$, then $x = x_1$ and clearly $x \in D(X)$.

If $\lambda_k < 1$ for all $1 \leq k \leq r$, then $r \geq 2$ and $x = \lambda_1 x^1 + \sum_{j=2}^{r} \lambda_j \cdot \sum_{k=2}^{r} \frac{\lambda_k}{\sum_{j=2}^{r} \lambda_j} x^k$. Set $\lambda := \lambda_1,\ y^1 := x^1$ and $y^2 := \sum_{k=2}^{r} \frac{\lambda_k}{\sum_{j=2}^{r} \lambda_j} x^k$. Then. $x = \lambda y^1 + (1-\lambda) y^2$, $0 \leq \lambda < 1$, $y^1, y^2 \in X$ and $y^1 \in D(X)$. Hence, by Lemma 9.30, $x \in D(X)$. □

Theorem 9.34

$x^* \in E(X)$ if and only if the linear program

$$max\{e^T z \mid Ax = b;\ Rx - z = Rx^*;\ x, z \geq 0\} \tag{9.47}$$

has an optimal solution (x^0, z^0) with $z^0 = 0$.

Proof

A feasible solution of x^* of (9.46) is efficient if and only if there is no x with $Ax = b;\ x \geq 0$ and $Rx > Rx^*$. Hence, x^* is efficient if and only if there is no (x, z) with $Ax = b;\ x \geq 0,\ z > 0$ and $Rx - z = Rx^*$. Note that the system $Ax = b;\ x \geq 0,\ z \geq 0;\ Rx - z = Rx^*$ has a feasible solution $x = x^*,\ z = 0$. Therefore, x^* is efficient if and only if the system $Ax = b;\ x \geq 0,\ z \geq 0;$ $Rx - z = Rx^*$ has only feasible solutions (x, z) with $z = 0$, i.e. x^* is efficient if and only if the linear program (9.47) has an optimal solution (x^0, z^0) with $z^0 = 0$. □

The dual of the linear program (9.47) is

$$min\{b^T u + w^T Rx^* \mid A^T u + R^T w \geq 0;\ -w \geq e\}. \tag{9.48}$$

Therefore, we can state the following result.

Theorem 9.35

$x^* \in E(X)$ if and only if the linear program (9.48) has an optimal solution (u^*, w^*) with $b^T u^* + (w^*)^T Rx^* = 0$.

Theorem 9.36

$x^* \in E(X)$ if and only if there exists a $\lambda \in \mathbb{R}^p$ with $\sum_{k=1}^{p} \lambda_k = 1$ and $\lambda_k > 0,\ k = 1, 2, \ldots, p$ such that x^* is an optimal solution of

$$max\{\lambda^T Rz \mid Ax = b;\ x \geq 0\}. \tag{9.49}$$

Proof

$\Rightarrow$ Let x^* be a feasible and efficient solution of (9.46). By Theorem 9.35 there exists an optimal solution (u^*, w^*) of (9.48) with $b^T u^* = -(w^*)^T Rx^*$ and $-w^* \geq e$. Then, u^* is also an optimal solution of the linear program

$$min\{b^T u \mid A^T u \geq -R^T w^*\}. \tag{9.50}$$

The dual program of (9.50) is

$$max\{-(w^*)TRx \mid Ax = b;\ x \geq 0\}. \tag{9.51}$$

x^* is feasible for (9.49), u^* is feasible for (9.50) and $b^T u^* = -(w^*)^T Rx^*$, implying that x^* is an optimal solution of program (9.51). Set $\lambda_k := \frac{w_k}{\sum_{i=1}^p w_i}$,‘$1 \leq k \leq p$. Then, $\sum_{k=1}^p \lambda_k = 1$ and $\lambda_k > 0,\ 1 \leq k \leq p$ (because $-w_k \leq 1$ for all k). Obviously, x^* is also an optimal solution of (9.49).

$\Leftarrow$ Suppose that x^* is an optimal solution of (9.49) and not efficient. Then, there is a vector x with $Ax = b;\ x \geq 0$ and $Rx > Rx^*$. Hence, $\lambda^T Rx > \lambda^T Rx^*$, i.e. x^* is not an optimal solution of (9.49).

□

Remark

Since the linear problem (9.49) has an extreme optimal solution, the set X has an efficient basic solution.

9.3.2 Discounted rewards

The linear program usually associated with the discounted reward criterion for immediate rewards $r_i(a)$ is

$$max \left\{ \sum_{(i,a)} r_i(a)x_i(a) \;\middle|\; \begin{array}{rcl} \sum_{(i,a)} \{\delta_{ij} - \alpha p_{ij}(a)\}x_i(a) & = & \beta_j,\ j \in S \\ x_i(a) & \geq & 0,\ (i,a) \in S \times A \end{array} \right\} \tag{9.52}$$

with dual program

$$min \left\{ \sum_j \beta_j v_j \;\middle|\; \sum_j \{\delta_{ij} - \alpha p_{ij}(a)\}v_j \geq r_i(a) \text{for every } (i,a) \in S \times A \right\}. \tag{9.53}$$

Define the utility $v^\alpha(\beta, R)$ by

$$v^\alpha(\beta, R) := \sum_{t=1}^\infty \alpha^{t-1} \sum_{j \in S} \beta_j \cdot \sum_{(i,a)} \mathbb{P}_R\{X_t = i,\ Y_t = a \mid X_1 = j\} \cdot r_i(a). \tag{9.54}$$

A policy R_* is *β-optimal* if $v^\alpha(\beta, R_*) = max_R\, v^\alpha(\beta, R)$. Clearly, a discounted optimal policy is β-optimal, but not conversely.

Theorem 9.37

If x is an optimal solution of the linear program (9.52) and f^∞ is such that $x_i\big(f(i)\big) > 0,\ i \in S_x$, where $S_x := \{j \mid \sum_a x_j(a) > 0\}$, then f^∞ is a β-optimal policy.

Proof

Since v^α is the smallest α-superharmonic vector (see Theorem 3.16), v^α is an optimal solution of program (9.53) (the solution is not necessarily unique because $\beta_j = 0$ is allowed for some $j \in S$). By the complementary property of linear programming, we have

$$\sum_j \{\delta_{ij} - \alpha p_{ij}\big(f(i)\big)\} v_j^\alpha = r_i\big(f(i)\big),\ i \in S_x. \tag{9.55}$$

Next, we show that the set S_x is closed in the Markov chain $P(f)$. Suppose that S_x is not closed, i.e. $p_{kl}(f) > 0$ for some $k \in S_x$ and $l \notin S_x$. Since

$$0 = \textstyle\sum_a x_l(a) = \beta_l + \alpha \sum_{(k,a)} p_{kl}(a) x_k(a) \geq p_{kl}(f) x_k i\big(f(ki)\big) > 0,$$

we have a contradiction. Because S_x is closed and $\beta_l = 0,\ l \notin S_x$ (this follows from the relation $0 = \sum_a x_l(a) = \beta_l + \alpha \sum_{(k,a)} p_{kl}(a) x_k(a) \geq 0$), we may consider the process on S_x. Then, by (9.55), $\{I - \alpha P(f)\} v^\alpha = r(f)$, implying $v^\alpha(f^\infty) = \{I - \alpha P(f)\}^{-1} r(f) = v^\alpha$ on S_x. Hence,

$$v^\alpha(\beta, f^\infty) = \textstyle\sum_j \beta_j v_j^\alpha(f^\infty) = \sum_{j \in S_x} \beta_j v_j^\alpha(f^\infty) = \sum_{j \in S_x} \beta_j v_j^\alpha = \sum_j \beta_j v_j^\alpha \geq v^\alpha(\beta, R)$$

for all policies R. □

Define the utilities $v_k^\alpha(\beta, R)$ by

$$v_k^\alpha(\beta, R) := \sum_{t=1}^\infty \alpha^{t-1} \sum_{j \in S} \beta_j \cdot \sum_{(i,a)} \mathbb{P}_R\{X_t = i,\ Y_t = a \mid X_1 = j\} \cdot r_i^k(a).$$

Theorem 9.38

Take any $\lambda \in \mathbb{R}^m$ with $\lambda_k > 0,\ k = 1, 2, \dots, m$ and let x be an optimal solution of the linear program

$$max \left\{ \sum_{(i,a)} \Big\{ \sum_{k=1}^m \lambda_k r_i^k(a) \Big\} x_i(a) \;\middle|\; \begin{array}{rcl} \sum_{(i,a)} \{\delta_{ij} - \alpha p_{ij}(a)\} x_i(a) & = & \beta_j,\ j \in S \\ x_i(a) & \geq & 0,\ (i,a) \in S \times A \end{array} \right\}. \tag{9.56}$$

Take f^∞ such that $x_i\big(f(i)\big) > 0,\ i \in S_x$, then f^∞ a β-efficient policy.

Proof

From Theorem 9.37 it follows that f^∞ is a β-optimal policy with respect to the immediate rewards $r_i(a) = \sum_{k=1}^m \lambda_k r_i^k(a),\ (i,a) \in S \times A$, i.e. $v^\alpha(\beta, f^\infty) \geq v^\alpha(\beta, R)$ for all policies R. Since the discounted rewards are linear in the immediate rewards, we have $v^\alpha(\beta, R) = \sum_{k=1}^m v_k^\alpha(\beta, R)$. Suppose that f^∞ is not β-efficient. Then, there exists a policy R such that

$$\textstyle\sum_{k=1}^m \lambda_k v_k^\alpha(\beta, R) > \sum_{k=1}^m \lambda_k v_k^\alpha(\beta, f^\infty).$$

On the other hand we have $\sum_{k=1}^m \lambda_k v_k^\alpha(\beta, f^\infty) = v^\alpha(\beta, f^\infty) \geq v^\alpha(\beta, R) = \sum_{k=1}^m \lambda_k v_k^\alpha(\beta, R)$, implying a contradiction. □

Remark

Suppose that we want to maximize lexicographically the functions $v_k^\alpha(\beta,R)$ for $k=1,2,\dots,m$. A policy R^* which is lexicographically maximal with respect to $v_1^\alpha(\beta,R),\ v_2^\alpha(\beta,R),\dots,v_m^\alpha(\beta,R)$ is a *lexicographically efficient* policy.

To determine a lexicographically efficient policy, we compute an optimal solution, say x^1, of the linear program

$$max\left\{\sum_{(i,a)} r_i^1(a)x_i(a)\ \middle|\ \begin{array}{rcl}\sum_{(i,a)}\{\delta_{ij}-\alpha p_{ij}(a)\}x_i(a) & = & \beta_j,\ j\in S\\ x_i(a) & \geq & 0,\ (i,a)\in S\times A\end{array}\right\}. \tag{9.57}$$

Next, we solve the following linear program with one additional constraint

$$max\left\{\sum_{(i,a)} r_i^2(a)x_i(a)\ \middle|\ \begin{array}{rcl}\sum_{(i,a)}\{\delta_{ij}-\alpha p_{ij}(a)\}x_i(a) & = & \beta_j,\ j\in S\\ \sum_{(i,a)} r_i^1(a)x_i(a) & = & \sum_{(i,a)} r_i^1(a)x_i^1(a)\\ x_i(a) & \geq & 0,\ (i,a)\in S\times A\end{array}\right\}. \tag{9.58}$$

Continuing in this way we stop either when we find a unique optimal solution x^k for some $1\leq k\leq m$ or when we have solved all m linear programs. Let x^* be the finally obtained solution. Then, a lexicographically efficient solution is the stationary policy π^∞, defined by

$$\pi_{ia}:=\begin{cases} x_i^*(a)/x_i^* & \text{if } x_i^*>0\\ \text{arbitrary} & \text{if } x_i^*=0\end{cases} \quad \text{for all } (i,a)\in S\times A.$$

9.3.3 Average rewards

The average reward case is, as always, more cumbersome that the discounted reward case. The linear program usually associated with the average reward criterion for immediate rewards $r_i(a)$ is

$$max\left\{\sum_{(i,a)} r_i(a)x_i(a)\ \middle|\ \begin{array}{rcrcl}\sum_{(i,a)}\{\delta_{ij}-p_{ij}(a)\}x_i(a) & & & = & 0,\ j\in S\\ \sum_a x_j(a) & + & \sum_{(i,a)}\{\delta_{ij}-p_{ij}(a)\}y_i(a) & = & \beta_j,\ j\in S\\ & & x_i(a),y_i(a) & \geq & 0,\ (i,a)\in S\times A\end{array}\right\} \tag{9.59}$$

with dual program

$$min\left\{\sum_j \beta_j v_j\ \middle|\ \begin{array}{rcll}\sum_j\{\delta_{ij}-p_{ij}(a)\}v_j & \geq & 0 & \text{for every } (i,a)\in S\times A\\ v_i+\sum_j\{\delta_{ij}-p_{ij}(a)\}u_j & \geq & r_i(a) & \text{for every } (i,a)\in S\times A\end{array}\right\}. \tag{9.60}$$

Define the utility $\phi(\beta,R)$ by

$$\phi(\beta,R):=\liminf_{T\to\infty}\frac{1}{T}\sum_{t=1}^T\sum_{j\in S}\beta_j\cdot\sum_{(i,a)}\mathbb{P}_R\{X_t=i,\ Y_t=a\mid X_1=j\}\cdot r_i(a).$$

A policy R_* is *β-optimal* if $\phi(\beta,R_*)=max_R\,\phi(\beta,R)$. Clearly, an average optimal policy is β-optimal, but not conversely. For any feasible solution (x,y) of (9.59), we define $x_i,\ y_i,\ i\in S$ and $S_x,S_y\subseteq S$ by $x_i:=\sum_a x_i(a),\ y_i:=\sum_a y_i(a),\ S_x:=\{j\mid\sum_a x_j>0\},\ S_y:=\{j\mid x_j=0,\ y_j>0\}$.

In this subsection, for any $R \in C_1$, the set of convergent policies, we use for convenience the notation $x(R)$ instead of $x(\beta, R)$, i.e. for $R \in C_1$, $x(R) := \lim_{T\to\infty} x^{\beta,T}(R)$, where $x^{\beta,T}(R)$ is defined in (9.22).

Theorem 9.39

If $R_* \in C_1$ is a β-optimal policy, then $\big(x(R_*), y_*\big)$ is an optimal solution of (9.59) for every y_* such that $\big(x(R_*), y_*\big)$ is a feasible solution of (9.59).

Proof

Take any feasible solution (x, y) of (9.59). Since, by Theorem 9.22, $L = Q$, we have $x = x(R)$ for some $R \in C_1$ and there exists a vector y such that $\big(x(R), y\big)$ is a feasible solution of (9.59). Furthermore, $\phi(\beta, R) = \sum_{(i,a)} r_i(a)x_{ia}(R) = \sum_{(i,a)} r_i(a)x_i(a)$. Because $R_* \in C_1$, there also exists a vector y_* such that $\big(x(R_*), y_*\big)$ is a feasible solution of (9.59) and $\phi(\beta, R_*) = \sum_{(i,a)} r_i(a)x_{ia}(R_*)$. Because R_* is a β-optimal policy, $\phi(\beta, R_*) \geq \phi(\beta, R)$, implying that $\big(x(R_*), y_*\big)$ is an optimal solution of (9.59) for every y_* such that $\big(x(R_*), y_*\big)$ is a feasible solution of (9.59). □

With a given feasible solution (x, y) of (9.59), we can associate a stationary policy $\pi^\infty(x, y)$ defined by

$$\pi_{ia}(x,y) := \begin{cases} \frac{x_i(a)}{x_i} & i \in S_x \\ \frac{y_i(a)}{y_i} & i \in S_y \\ \text{arbitray} & i \notin S_x \cup S_y \end{cases} \tag{9.61}$$

where $S_x := \{i \mid \sum_i x_i(a) > 0\}$, $S_y := \{i \mid \sum_i x_i(a) = 0 \ \sum_i y_i(a) > 0\}$, $x_i := \sum_i x_i(a)$ and $y_i := \sum_i y_i(a)$.

Theorem 9.40

If (x, y) is an optimal solution of (9.59) and $\pi^\infty(x, y)$ is defined by (9.61), then $\pi^\infty(x, y)$ is a β-optimal policy.

Proof

Since the value vector ϕ is the smallest superharmonic vector, $(v = \phi, u)$ is an optimal solution of program (9.60) for some vector u. By adding the second set of constraints of (9.59), we obtain $\sum_{(j,a)} x_j(a) = \sum_j \beta = 1$, implying $S_x \neq \emptyset$. Let $A^+(i) := a \in A(i) \mid \pi_{ia}(x, y) > 0\}$, $i \in S$. From the complementary slackness property of linear programming it follows that

$$\begin{aligned} \phi_i + \sum_j \{\delta_{ij} - p_{ij}(a)\}u_j &= r_i(a),\ i \in S_x,\ a \in A^+(i); && (9.62) \\ \sum_j \{\delta_{ij} - p_{ij}(a)\}\phi_j &= 0,\ i \in S_y,\ a \in A^+(i). && (9.63) \end{aligned}$$

The linear program (9.60) implies $\sum_j \{\delta_{ij} - p_{ij}(a)\}\phi_j \geq 0$, $(i, a) \in S \times A$. Suppose that for some $k \in S_x$ and some $a_k \in A^+(k)$, we have $\sum_j \{\delta_{kj} - p_{kj}(a_k)\}\phi_j > 0$.

Since $\pi_{ka_k}(x, y) > 0$, also $x_k(a_k) > 0$, so $\sum_j \{\delta_{kj} - p_{kj}(a_k)\}\phi_j \cdot x_k(a_k) > 0$. Furthermore, $\sum_j \{\delta_{ij} - p_{ij}(a)\}\phi_j \cdot x_i(a) \geq 0$, $(i, a) \in S \times A$. Hence, $\sum_{i,a} \sum_j \{\delta_{ij} - p_{ij}(a)\}\phi_j \cdot x_i(a) > 0$.

On the other hand, this result is contradictory to the constraints of program (9.60) from which follows that $\sum_{i,a} \sum_j \{\delta_{ij} - p_{ij}(a)\}\phi_j \cdot x_i(a) = \sum_j \big\{ \sum_{i,a} \{\delta_{ij} - p_{ij}(a)\}x_i(a)\big\}\phi_j = 0$.
This contradiction implies that $\sum_j \{\delta_{ij} - p_{ij}(a)\}\phi_j = 0$ for all $i \in S_x,\ a \in A^+(i)$. With (9.64), it follows that

$$\sum_j \{\delta_{ij} - p_{ij}(a)\}\phi_j = 0 \text{ for all } i \in S_x \cup S_y,\ a \in A^+(i). \tag{9.64}$$

Next, we show that S_x is closed under $P\big(\pi(x,y)\big)$, i.e. $p_{ij}(\pi(x,y) = 0,\ i \in S_x,\ j \notin S_x$. Suppose that $p_{kl}\big(\pi(x,y)\big) > 0$ for some $k \in S_x,\ l \notin S_x$. Since $p_{kl}\big(\pi(x,y)\big) = \sum_a p_{kl}(a)\pi_{ka}(x,y)$, there exists an action a_k such that $p_{kl}(a_k) > 0$ and $\pi_{ka_k}(x,y) > 0$. From the constraints of program (9.59) it follows that $0 = \sum_a x_l(a) = \sum_{(i,a)} p_{il}(a)x_i(a) \geq p_{kl}(a_k)x_k(a_k) > 0$, implying a contradiction.
Consider the Markov chain on S_x. From (9.64) and (9.62) it follows that on S_x we obtain

$$\phi = P\big(\pi(x,y)\big)\phi, \text{ implying } \phi = P^*\big(\pi(x,y)\big)\phi, \text{ and } \phi + \{I - P\big(\pi(x,y)\big)\}u = r\big(\pi(x,y)\big).$$

Hence, on S_x we have

$$\phi = P^*\big(\pi(x,y)\big)\big\{r\big(\pi(x,y)\big) - \{I - P\big(\pi(x,y)\big)\}u\big\} = P^*\big(\pi(x,y)\big)r\big(\pi(x,y)\big) = \phi\big(\pi^\infty(x,y)\big).$$

So, we have shown that

$$\phi_i = \phi_i\big(\pi^\infty(x,y)\big),\ i \in S_x. \tag{9.65}$$

We shall now show that $S_x \cup S_y$ is also closed under $P\big(\pi(x,y)\big)$. Since S_x is closed, it is sufficient to show that $p_{kl}\big(\pi(x,y)\big) = 0,\ k \in S_y,\ l \notin S_x \cup S_y$. Suppose that $p_{kl}\big(\pi(x,y)\big) > 0$ for some $k \in S_y$ and $l \notin S_x \cup S_y$. Then.

$$\begin{aligned} 0 &= \textstyle\sum_a x_l(a) + \sum_a y_l(a) = \beta_l + \sum_{(i,a)} p_{il}(a)y_i(a) \geq \sum_a p_{kl}(a)y_k(a) \\ &= \textstyle\sum_a p_{kl}(a)\pi_{ka}(x,y)y_k = p_{kl}(a)\big(\pi(x,y)\big)y_k > 0, \end{aligned}$$

So, we have a contradiction, which establishes that $S_x \cup S_y$ is closed.
Since $S_x \cup S_y$ is closed and since $\beta_j = 0$ for all $j \notin S_x \cup S_y$, the stochastic process with β as initial distribution, will never enter any of the states outside $S_x \cup S_y$. Hence, it is sufficient to consider the process on the closed set $S_x \cup S_y$. By the relation (9.64), we have $\phi = P\big(\pi^\infty(x,y)\big)\phi$, which implies

$$\phi = P^*\big(\pi^\infty(x,y)\big)\phi. \tag{9.66}$$

Assume that S_1 is an ergodic set outside S_x, i.e. $S_1 \subseteq S_y$. Adding the second set of the constraints of (9.59) that correspond to S_1 yields

$$\begin{aligned} \textstyle\sum_{j \in S_1} \beta_j &= \textstyle\sum_{j \in S_1} y_j - \sum_{j \in S_1} \sum_{(i,a)} p_{ij}(a)y_i(a) \\ &= \textstyle\sum_{j \in S_1} y_j - \sum_{j \in S} \sum_{i,a} p_{ij}(a)y_i(a) + \sum_{j \notin S_1} \sum_{(i,a)} p_{ij}(a)y_i(a) \\ &= \textstyle\sum_{j \in S_1} y_j - \sum_{j \in S} \sum_{i \in S_1} \sum_a p_{ij}(a)y_i(a) - \sum_{j \in S} \sum_{i \notin S_1} \sum_a p_{ij}(a)y_i(a) \\ &\qquad \textstyle+ \sum_{j \notin S_1} \sum_{i \in S_1} \sum_a p_{ij}(a)y_i(a) + \sum_{j \notin S_1} \sum_{i \notin S_1} \sum_a p_{ij}(a)y_i(a) \\ &= \textstyle\sum_{j \in S_1} y_j - \sum_{j \in S} \sum_{i \in S_1} \sum_a p_{ij}(a)y_i(a) - \sum_{j \in S} \sum_{i \notin S_1} \sum_a p_{ij}(a)y_i(a) \\ &\qquad \textstyle+ \sum_{j \notin S_1} \sum_{i \notin S_1} \sum_a p_{ij}(a)y_i(a) \\ &= \textstyle\sum_{j \in S_1} y_j - \sum_{i \in S_1} y_i - \sum_{j \in S} \sum_{i \notin S_1} \sum_a p_{ij}(a)y_i(a) + \sum_{j \notin S_1} \sum_{i \notin S_1} \sum_a p_{ij}(a)y_i(a) \\ &= \textstyle-\sum_{j \in S_1} \sum_{i \notin S_1} \sum_a p_{ij}(a)y_i(a) \leq 0. \end{aligned}$$

Hence, $\beta_i = 0$ for all $i \in R_y$, where $R_y := \{i \in S_y \mid i \text{ is recurrent in the Markov chain } P\big(\pi(x,y)\big)\}$, and $\sum_a p_{ij}(a)y_i(a) = 0$ for all $i \notin R_y$ and $j \in R_y$. Then, for $i \in S_y \backslash R_y$ and $j \in R_y$, we have $p_{ij}\big(\pi(x,y)\big) = 0$, which implies that $p^*_{ij}(\big(\pi(x,y)\big) = 0$ for all $i \notin R_y$ and $j \in R_y$. Consequently, for all for all $i \in S_y \backslash R_y$,

$$\begin{array}{rcl}\phi_i\big(\pi^\infty(x,y)\big) & = & \sum_j p^*_{ij}\big(\pi(x,y)\big)r_j\big(\pi(x,y)\big) \;=\; \sum_{j\in S_x} p^*_{ij}\big(\pi(x,y)\big)r_j\big(\pi(x,y)\big) \\ & = & \sum_{j\in S_x} p^*_{ij}\big(\pi(x,y)\big)\{\phi_j + \sum_k\{\delta_{jk} - p_{jk}\big(\pi(x,y)\big)\}u_k\} \\ & = & \sum_{j\in S_x} p^*_{ij}\big(\pi(x,y)\big)\phi_j \;=\; \phi_i.\end{array}$$

So, we have shown that

$$\phi_i = \phi_i\big(\pi^\infty(x,y)\big),\ i \in S_y \backslash R_y. \tag{9.67}$$

Because $\beta_i = 0$ for all $i \in R_y$ and by (9.65) and (9.67), we obtain

$$\sum_i \beta_i \cdot \phi_i\big(\pi^\infty(x,y)\big) = \sum_{i\notin R_y} \beta_i \cdot \phi_i\big(\pi^\infty(x,y)\big) = \sum_{i\notin R_y} \beta_i \cdot \phi_i = \sum_i \beta_i \cdot \phi_i,$$

i.e. $\pi^\infty(x,y)$ is β-optimal. □

Theorem 9.41

If (x,y) is an extreme optimal solution of the linear program (9.59) and f^∞ is such that $x_i\big(f(i)\big) > 0,\ i \in S_x;\ y_i\big(f(i)\big) > 0,\ i \in S_y$, then f^∞ is a β-optimal policy.

Proof

Similarly to the corresponding part in the proof of Theorem 5.18 (the proof is left to the reader as Exercise 9.6) it can be shown that

$$\begin{array}{rcll}\phi_i \;+\; \sum_j \{\delta_{ij} - p_{ij}\big(f(i)\big)\}u_j & = & r_i\big(f(i)\big) & ,\ i \in S_x \\ \sum_j \{\delta_{ij} - p_{ij}\big(f(i)\big)\}\phi_j & = & 0 & ,\ i \in S_x \cup S_y\end{array} \tag{9.68}$$

We first show that S_x is closed in the Markov chain $P(f)$. Suppose that $p_{kl}\big(f(k)\big) > 0$ for some $k \in S_x,\ l \notin S_x$. From the constraints of (9.59) it follows that

$$0 = \sum_a x_l(a) = \sum_{i,a} p_{il}(a)x_i(a) \geq p_{kl}\big(f(k)\big)x_k\big(f(k)\big) > 0,$$

implying a contradiction.

We now show that $S_x \cup S_y$ is also closed. Suppose that $p_{kl}(f) > 0$ for some $k \in S_x \cup S_y$ and $l \notin S_x \cup S_y$. Then,

if $k \in S_x$: $0 = \sum_a x_l(a) = \sum_{i,a} p_{il}(a)x_i(a) \geq p_{kl}(f)x_k\big(f(k)\big) > 0$;

if $k \in S_y$: $0 = \sum_a x_l(a) + \sum_a y_l(a) = \beta_l + \sum_{i,a} p_{il}(a)y_i(a) \geq p_{kl}(f)y_k\big(f(l)\big) > 0$.

In both cases we have a contradiction: $S_x \cup S_y$ is closed in the Markov chain $P(f)$.

Next, we show that the states of S_y are transient in the Markov chain $P(f)$. Suppose that S_y has an ergodic state. Since S_x and $S_x \cup S_y$ are closed, the set S_y contains an ergodic class, say $J = \{j_1, j_2, \dots, j_m\}$. Since (x,y) is an extreme solution and $y_j\big(f(j)\big) > 0,\ j \in J$, the corresponding columns in (9.59) are linearly independent. Because these columns have zeros in the first N rows, the second parts of these vectors are also independent vectors. Since for $j \in J$

and $k \notin J$, we have $\delta_{jk} - p_{jk}\big(f(j)\big) = 0 - 0 = 0$, the vectors $b^i,\ 1 \le i \le m$, where b^i has components $\delta_{j_ik} - p_{j_ik}\big(f(j_i)\big),\ k \in J$, are also linear independent.
However, $\sum_{k=1}^m b_k^i = \sum_{k=1}^m \{\delta_{j_ij_k} - p_{j_ij_k}\big(f(j_i)\big)\} = 1 - 1 = 0,\ i = 1, 2, \dots, m$, which contradicts the independence of $b^1, b^2, \dots, b^m$.
Consider the Markov chain on the closed $S_x \cup S_y$. From (9.68) it follows that $\phi = P(f)\phi$, and consequently we have $\phi = P^*(f)\phi$. Since that states of S_y are transient, the columns of $P^*(f)$ corresponding to S_y are zero-vectors. Hence, also by (9.68),

$$\begin{array}{rcl} \{P^*(f)r(f)\}_i & = & \sum_j p_{ij}^*(f)r_j(f) \ = \ \sum_{j\in S_x} p_{ij}^*(f)r_j(f) \\ & = & \sum_{j\in S_x} p_{ij}^*(f)\{\phi_j + u_j - \{P(f)u\}_j\} \\ & = & \big\{P^*(f)\{\phi + u - P(f)u\}\big\}_i = \{P^*(f)\phi\}_i \ = \ \phi_i,\ i \in S_x \cup S_y. \end{array}$$

Hence,

$$\begin{array}{rcl} \phi(\beta, f^\infty) & = & \sum_i \beta_i\{P^*(f)r(f)\}_i \ = \ \sum_{i\in S_x\cup S_y} \beta_i\{P^*(f)r(f)\}_i \ = \ \sum_{i\in S_x\cup S_y} \beta_i\phi_i \\ & = & \sum_i \beta_i\phi_i = \phi(\beta, \phi) \ \ge \ \phi(\beta, R) \text{ for all policies } R. \end{array}$$

□

Let $\phi_k(\beta, R)$ be the average reward for immediate rewards $r_i^k(a),\ (i, a) \in S \times A,\ 1 \le k \le m$, given initial distribution β and policy R, i.e.

$$\phi_k(\beta, R) := \liminf_{T\to\infty} \frac{1}{T}\sum_{t=1}^T\sum_{j\in S} \beta_j \cdot \sum_{(i,a)} \mathbb{P}_R\{X_t = i,\ Y_t = a \mid X_1 = j\} \cdot r_i^k(a). \tag{9.69}$$

Let $E(C)$ be the set of all β-efficient policies, i.e.

$$E(C) := \left\{ R_* \ \middle| \ \begin{array}{l} \text{there does not exist a policy } R \text{ such that } \phi_k(\beta, R) \ge \phi_k(\beta, R_*) \\ \text{for } k = 1, 2, \dots, m \text{ and } \phi_k(\beta, R) > \phi_k(\beta, R_*) \text{ for at least one } k \end{array} \right\}. \tag{9.70}$$

Our aim is to characterize $E(C)$ and, equally importantly, the set $E_0(C)$ which is the image in the objective space, i.e.

$$E_0(C) := \{(\phi^1, \phi^2, \dots, \phi^m) \mid \text{there exists a policy } R \in E(C) \text{ such that } \phi_k(\beta, R) \ge \phi_k,\ 1 \le k \le m\}. \tag{9.71}$$

Note that for every $R \in C_1$ we know from Theorem 9.21 and Theorem 9.22 that there exists a $x \in Q$ such that $x(R) = x$. Hence, we have for every $k = 1, 2, \dots, m$ and every $R \in C_!$,

$$\phi_k(\beta, R) = \sum_{(i,a)} r_i^k(a)x_{ia}(R) = \sum_{(i,a)} r_i^k(a)x_{ia} \text{ for some } x \in Q. \tag{9.72}$$

Consider the $m \times n$-dimensional matrix R, where $n := \sum_{i=1}^N |A(i)|$, whose rows are $r^1, r^2, \dots, r^m$, treated as n-dimensional vectors with components $r_i^k(a)$, the rewards of the m single-objective MDPs. With the above multi-objective MDP we shall associate the following multi-objective linear program (MOLP)

$$max\{Rx \mid x \in Q\}. \tag{9.73}$$

By (9.72), the values $\sum_{(i,a)} r_i^k(a)x_{ia},\ 1 \le k \le m$ of the objectives of the multi-objective linear program correspond to the payoffs $\phi_k(\beta, R),\ \le k \le m$, where R and x satisfy $x(R) = x$, of the multi-objective MDP.

For every $\lambda \in \mathbb{R}^m_+$ with $\sum_{k=1}^m \lambda_k = 1$ we can associate a weighted MDP, denoted by MDP(λ), which is identical to the original MDP, but whose rewards are $r_i^\lambda(a) := \sum_{k=1}^m \lambda_k r_i^k(a),\ (i,a) \in S \times A$. The average rewards in MDP(λ), given initial distribution β, are denoted as $\phi^\lambda(\beta, R$. The linear program for MDP (λ) becomes

$$max\{\lambda^T Rx \mid x \in Q\}. \tag{9.74}$$

Let $XY := \{(x,y) \mid (x,y)$ is feasible for (9.59)$\}$ and let $E(XY)$ be the set of all efficient points for MOLP. Also define X and $E(X)$ by $X := \{x \mid (x,y)$ is feasible for (9.59) for some $y\}$ and $E(X) := \{x \mid (x,y) \in E(XY)$ for some $y\}$. By Theorem 9.34, we have the following result.

Theorem 9.42

$(x^*, y^*) \in E(XY)$ if and only if there exists $\lambda \in \mathbb{R}^m$ with $\sum_{k=1}^m \lambda_k = 1$ and $\lambda_k > 0,\ k = 1,2,\dots,m$ such that (x^*, y^*) is an optimal solution of (9.74).

Lemma 9.31

Let $\lambda \in \mathbb{R}^m$ with $\sum_{k=1}^m \lambda_k = 1$ and $\lambda_k > 0,\ k = 1,2,\dots,m$, and let R_* be a β-optimal policy of $MDP(\lambda)$. Then, $R_* \in E(C)$.

Proof

Suppose $R_* \notin E(C)$, i.e. there exists a policy R such that $\phi_k(\beta, R) \geq \phi_k(\beta, R_*)$ for all k and with a strict inequality for at least one k. Hence, $\sum_{k=1}^m \lambda_k \phi_k(\beta, R) > \sum_{k=1}^m \lambda_k \phi_k(\beta, R_*)$. From Theorem 9.22 it follows that there exists $x(R) \in Q$ such that $\phi_k(\beta, R) = \sum_{(i,a)} r_i^k(a) x_{ia}(R),\ 1 \leq k \leq m$. A similar result holds for R_*. Therefore, we can write

$$\phi^\lambda(\beta, R) = \sum_{k=1}^m \lambda_k \phi_k(\beta, R) > \sum_{k=1}^m \lambda_k \phi_k(\beta, R_*) = \phi^\lambda(\beta, R_*),$$

which leads to a contradiction of the β-optimality of policy R_* in $MDP(\lambda)$. □

Corollary 9.9

Let $\lambda \in \mathbb{R}^m$ with $\sum_{k=1}^m \lambda_k = 1$ and $\lambda_k > 0,\ k = 1,2,\dots,m$, and let (x,y) be an extreme optimal solution of the linear program (9.59). Then, the policy f^∞ satisfying $x_i\big(f(i)\big) > 0,\ i \in S_x$; $y_i\big(f(i)\big) > 0,\ i \in S_y$, is a β-efficient policy.

Proof

From Theorem 9.41 it follows that f^∞ is a β-optimal policy for $MDP(\lambda)$. Applying Lemma 9.31 yields the result. □

Theorem 9.43

(1) If $(x^*, y^*) \in E(XY)$ and $x^* = x(R_*)$ for some $R_* \in C_1$, then $R_* \in E(C)$.

(2) If $R_* \in E(C) \cap C_1$, then $\big(x(R_*), y\big) \in E(XY)$ for all y such that $\big(x(R_*), y\big) \in E(XY)$.

Proof

(1) By Theorem 9.42, there exists $\lambda \in \mathbb{R}^m$ with $\sum_{k=1}^m \lambda_k = 1$ and $\lambda_k > 0,\ k = 1, 2, \dots, m$, such that $\lambda^T Rx^* \geq \lambda^T Rx$ for all $x \in X$. Hence, by (9.72), $\phi^\lambda(\beta, R_) \geq \phi^\lambda(\beta, R)$ for all $R \in C_1$, i.e. R_* is β-optimal in $MDP(\lambda)$. By Lemma 9.31, $R_* \in E(C)$.

(2) Since, by Theorem 9.21 and Theorem 9.22, $x(R_*) \in X$, there exists y with $\big(x(R_*), y\big) \in XY$. Suppose that $\big(x(R_*), y\big) \notin E(XY)$ for some y such that $\big(x(R_*), y\big) \in XY$. Then, there exists $(\overline{x}, \overline{y}) \in XY$ such that $R\overline{x} \geq Rx(R_*)$, i.e. $\sum_{(i,a)} r_i^k(a)\overline{x}_i(a) \geq \sum_{(i,a)} r_i^k(a)x_{ia}(R_*)$ for $k = 1, 2, \dots, m$ with strict inequality for at least one k. Again, by Theorem 9.21 and Theorem 9.22, there exists a policy $\overline{R} \in C_1$ such that $\overline{x} = x(\overline{R})$. Hence, by (9.72), $\phi_k(\beta, \overline{R}) \geq \phi_k(\beta, R_*)$ for $k = 1, 2, \dots, m$ with strict inequality for at least one k. This implies $R_* \notin E(C)$, which yields the desired contradiction. □

The next lemma shows that the relation between stationary policies and feasible solutions of (9.59) preserves the property of efficiency.

Lemma 9.32

(1) If $(x^*, y*)^{\in} E(XY)$, then $\pi^\infty(x^*, y^*) \in E(C)$, where $\pi^\infty(x^*, y^*)$ is defined in (9.61).

(2) If $\pi^\infty \in E(C) \cap C(S)$, then $\big(x(\pi), y(\pi)\big) \in E(XY)$, where $\big(x(\pi), y(\pi)\big)$ is defined in (5.35) and (5.36).

Proof

(1) By Theorem 9.42, there exists $\lambda \in \mathbb{R}^m$ with $\sum_{k=1}^m \lambda_k = 1$ and $\lambda_k > 0,\ k = 1, 2, \dots, m$ such that (x^*, y^*) is an optimal solution of (9.74). By Theorem 9.40, $\pi^\infty(x^*, y^*)$ is a β-optimal solution of $MDP(\lambda)$. Then, by Lemma 9.31, $\pi^\infty(x^*, y^*) \in E(C)$.

(2) This part follows directly from part (2) of Theorem 9.43). □

Lemma 9.33

For every $R \in E(C)$ there exists a policy $R_1 \in C_1 \cap C(M)$ such that $\phi_k(\beta, R) = \phi_k(\beta, R_1)$ for $k = 1, 2, \dots, m$.

Proof

Note that, by (9.22) and Theorem 9.22, there exists $x(R) \in X$ such that

$$\begin{array}{rcl} \phi_k(\beta, R) & = & \liminf_{T\to\infty} \frac{1}{T} \sum_{t=1}^T \sum_{j\in S} \beta_j \cdot \sum_{(i,a)} \mathbb{P}_R\{X_t = i,\ Y_t = a \mid X_1 = j\} \cdot r_i^k(a) \\ & = & \liminf_{T\to\infty} \sum_{(i,a)} x_{ia}^{\beta,T}(R) \cdot r_i^k(a) \ \leq\ \sum_{(i,a)} x_{ia}(R) \cdot r_i^k(a). \end{array}$$

Since $x(R) = x(R_1)$ for some $R_1 \in C_1 \cap C(M)$ (see Theorem 1.1 and Theorem 9.21, we have

$$\phi_k(\beta, R) \leq \sum_{(i,a)} x_{ia}(R_1) \cdot r_i^k(a) = \phi_k(\beta, R_1).$$

Because $R \in E(C)$ a strict inequality is impossible. Therefore, we have $\phi_k(\beta, R) = \phi_k(\beta, R_1)$ for $k = 1, 2, \dots, m$. □

The next theorem characterizes the set $E(C)$ of efficient policies.

Theorem 9.44
A policy $R_* \in E(C)$ if and only if there exists $\lambda \in \mathbb{R}^m$ with $\sum_{k=1}^m \lambda_k = 1$ and $\lambda_k > 0$ for $k = 1, 2, \dots, m$ such that R_* is a β-optimal policy for $MDP(\lambda)$.

Proof

⇒ Take any policy $R_* \in E(C)$. By Lemma 9.33, we may assume $R_* \in E(C) \cap C_1$. Then, by Theorem 9.43 $\big(x(R_*), y\big) \in E(XY)$ for all y such that $\big(x(R_*), y\big) \in XY$. Now, Theorem 9.42 implies that there exists $\lambda \in \mathbb{R}^m$ with $\sum_{k=1}^m \lambda_k = 1$ and $\lambda_k > 0,\ k = 1, 2, \dots, m$ such that $\big(x(R_*), y\big)$ is optimal for (9.74). Hence, $\lambda^T Rx(R_*) \geq \lambda^T Rx$ for all $x \in X$. Take any policy R. Then, by an argument as in the proof of Lemma 9.33, $\phi_k(\beta, R) \leq \sum_{(i,a)} r_i^k(a) x_{ia}(R)$ for $k = 1, 2, \dots, m$ for some $x(R) \in X$. Therefore, by (9.72),

$$\phi^\lambda(\beta, R_*) = \lambda^T Rx(R_*) \geq \lambda^T Rx(R) \geq \phi^\lambda(\beta, R),$$

i.e. R_* is β-optimal for $MDP(\lambda)$.

⇐ This part follows directly from Lemma 9.31.

Corollary 9.10
If $\beta_j^* > 0,\ j \in S$, and $R_* \in E(C)$ with respect to $\phi_k(\beta^*, R_*),\ 1 \leq k \leq m$, then $R_* \in E(C)$ with respect to $\phi_k(\beta, R_*),\ 1 \leq k \leq m$, for all initial distributions β.

Proof

By Theorem 9.44, R_* is β^*-optimal for $MDP(\lambda)$, where $\lambda \in \mathbb{R}^m$ with $\sum_{k=1}^m \lambda_k = 1$ and $\lambda_k > 0$ for $k = 1, 2, \dots, m$. Since $\beta_j^* > 0,\ j \in S$, it follows from the general theory of MDPs with average rewards that R_* is average optimal for all initial states simultaneously, i.e. $\phi_i^\lambda(R_*) \geq \phi_i^\lambda(R)$ for all $i \in S$ and all $R \in C$. Hence, for all initial distributions β, $\phi^\lambda(\beta, R_*) \geq \phi^\lambda(\beta, R)$ for all $R \in C$. Then, again by Theorem 9.44, $R_* \in E(C)$ with respect to $\phi_k(\beta, R_*),\ 1 \leq k \leq m$, for all initial distributions β. □

Let $Ext(XY) := \{(x^l, y^l,\ 1 \leq l \leq p\}$ be the set of efficient extreme points of XY. For any point $(x, y) \in XY$ define the family $C_{xy}(D)$ of all deterministic policies f^∞ such that

$$x_i\big(f(i)\big) > 0,\ i \in S_x;\ y_i\big(f(i)\big) > 0,\ i \in S_y;\ f(i) \text{ an arbitrary action if } i \notin S_x \cup S_y.$$

Now, let $F_{ext} := \bigcup_{l=1}^p C_{x^l y^l}(D)$. The next theorem shows that F_{ext} constitutes the set of all efficient deterministic policies.

Theorem 9.45
$F_{ext} = E(C) \cap C(D)$.

Proof

Let $f^\infty \in F_{ext}$. By definition, $f^\infty \in C(D)$ and $f^\infty \in C_{x^l y^l}(D)$ for some $1 \leq l \leq p$. Because $(x^l, y^l) \in Ext(XY)$, by Theorem 9.42, (x^l, y^l) is an optimal solution of (9.74) for some $\lambda \in \mathbb{R}^m$ with $\sum_{k=1}^m \lambda_k = 1$ and $\lambda_k > 0$ for $k = 1, 2, \dots, m$. Now, by Theorem 9.41, f^∞ is β-optimal for $MDP(\lambda)$. Hence, by Theorem 9.44, $f^\infty \in E(C)$. So, we have shown that $F_{ext} \subseteq E(C) \cap C(D)$.

Next, take any $f^\infty \in E(C) \cap C(D)$. In order to prove that $f^\infty \in F_{ext}$ we have to show that there exists an efficient extreme point (x^l, y^l) with $x^l_i\big(f(i)\big) > 0,\ i \in S_{x^l}$ and $y^l_i\big(f(i)\big) > 0,\ i \in S_{y^l}$. Let $\big(x(f), y(f)\big)$ be constructed by (5.35) and (5.36). Then, by Theorem 5.21, $\big(x(f), y(f)\big)$ is is an extreme point of XY (it can easily be checked that the proof of Theorem 5.21 remains valid for any $\beta \geq 0$). Now, we shall prove that $\big(x(f), y(f)\big) \in E(XY)$. Suppose not, then there exists $(x, y) \in XY$ such that $\sum_{(i,a)} r^k_i(a) x_{ia} \geq \sum_{(i,a)} r^k_i(a) x_{ia}(f)$ for all $1 \leq k \leq m$ with a strict inequality for at least one k. Then, for any $\lambda \in \mathbb{R}^m$ with $\sum_{k=1}^m \lambda_k = 1$ and $\lambda_k > 0$ for $k = 1, 2, \dots, m$, we obtain $\lambda^T Rx > \lambda^T Rx(f)$. Since, by Theorem 9.22, $x = x(R)$ for some $R \in C_1$, we have

$$\phi^\lambda(\beta, R) = \lambda^T Rx(R) > \lambda^T Rx(f) = \phi^\lambda(\beta, f^\infty),$$

contradicting the hypothesis that $f^\infty \in E(C)$. Since $x_{ia}(f)$ and $y_{ia}(f)$ are strictly positive only if $a = f(i),\ f^\infty \in F_{ext}$. □

Lemma 9.34

Let $Ext(X)$ be the set of efficient extreme points of X. Then, if $x^* \in Ext(X)$, there exists $f^\infty_* \in F_{ext}$ such that $x^* = x(f_*)$.

Proof

By Theorem 9.21 and Theorem 9.22, $x = x(f_*)$ for some $f^\infty \in C(D)$. By Theorem 9.45, we have to show that $f \in E(C)$. Since any $\big(x(f_*), y\big) \in XY$ has a value of the objective function that is independent of y, $\big(x(f_*), y(f_*)\big) \in E(XY)$. Then, by Theorem 9.43 part (1), $f^\infty \in E(C)$. □

Given a point $x^* \in Ext(X)$, Lemma 9.34 guarantees the existence of a deterministic policy $f^\infty_* \in F_{ext}$. How can we construct such f^∞_*? We know that $x^* = x(f_*) = \beta^T P^*(f_*)$.

Since $x^*_i(a) = \begin{cases} x^*_i\big(f_*(i)\big), & i \in S_{x^*},\ a = f_*(i) \\ 0 & i \in S_{x^*},\ a \neq f_*(i) \end{cases}$ the choice of $f_*(i)$ on S_{x^*} is obvious. In order to find $f_*(i)$ for $i \notin S_{x^*}$, we have to examine the set $F_* := \{f^\infty \in C(D) \mid f(i) = f_*(i),\ i \in S_{x^*}\}$. If $\{\beta^T P^*(f)\}_i = 0$ for all $i \notin S_{x^*}$ for some $f^\infty \in F_*$, then f^∞ is a policy that satisfies $x^* = x(f)$. Let $Ext(X) = \{x^1, x^2, \dots, x^p\}$. For each x^k let f^∞_k be found via the above construction. Define $F^*_{ext} := \{f^\infty_1, f^\infty_2, \dots, f^\infty_p\} \subseteq F_{ext}$; F^*_{ext} is called the set of *basic efficient policies.*

Theorem 9.46

Let $F^*_{ext} = \{x^1, x^2, \dots, x^p\}$ be the set of basic efficient policies. Take $(\phi^1, \phi^2, \dots, \phi^p\} \in E_0(C)$ arbitrarily, where $E_0(C)$ is defined in (9.71). Then, there exists $\mu_j,\ 1 \leq j \leq p$, with $\sum_{j=1}^p \mu_j = 1$ and $\mu_j \geq 0$ for $j = 1, 2, \dots, p$, such that $\phi^k = \sum_{j=1}^p \mu_j \phi_k(\beta, f^\infty_j)$ for $k = 1, 2, \dots, m$.

Proof

By Lemma 9.33, the definition of $E_0(C)$ and (9.72), there exists $R_1 \in C_1 \cap E(C)$ such that

$$\phi^k = \phi_k(\beta, R_1) = \textstyle\sum_{(i,a)} r^k_i(a) x_{ia}(R_1) \text{ for } k = 1, 2, \dots, m.$$

Hence, by Theorem 9.44 part (2), we obtain $\big(x(R_1), y\big) \in E(XY)$ for all y with $\big(x(R_1), y\big) \in XY$. Now, by Theorem 9.33,

$x(R_1) = \sum_{j=1}^{p} \mu_j x(f_j)$ for some $\mu \in \mathbb{R}^p$ with $\mu_j,\ 1 \leq j \leq p$, and $\mu_j \geq 0$ for $j = 1, 2, \ldots, p$. Therefore,

$$\phi^k = \textstyle\sum_{j=1}^{p} \mu_j \sum_{(i,a)} r_i^k(a) x_{ia}(f_j) = \sum_{j=1}^{p} \mu_j \phi_k(\beta, f_j^\infty) \text{ for } k = 1, 2, \ldots, m. \qquad \square$$

We shall also introduce the concept of *uniform efficient policies.* A policy R_* is called uniform efficient if there does not exists a policy R such that $\phi_j^k(R) \geq \phi_j^k(R_*)$ for all $j \in S$ and all $1 \leq k \leq m$, with strict inequality holding for some k and some j. The following theorem shows that a uniform efficient deterministic policy always exists.

Theorem 9.47
Let f_*^∞ be an optimal policy for $MDP(\lambda)$ with λ such that $\sum_{k=1}^{m} \lambda_k = 1$ and $\lambda_k > 0$ for $k = 1, 2, \ldots, m$. Then, f_*^∞ is a uniform efficient policy.

Proof
Suppose that there exists a policy R such that $\phi_j^k(R) \geq \phi_j^k(f_*^\infty)$ for all $j \in S$ and all $1 \leq k \leq m$, with strict inequality holding for some k_* and some j_*. Take $\beta_j > 0$ for all $j \in S$ and $\sum_j \beta_j = 1$. Then, we have

$$\sum_{k=1}^{m} \lambda_k \phi^k(\beta, R) = \sum_{k=1}^{m} \lambda_k \sum_j \beta_j \phi_j^k(R) > \sum_{k=1}^{m} \lambda_k \sum_j \beta_j \phi_j^k(f_*^\infty) = \sum_{k=1}^{m} \lambda_k \phi^k(f_*^\infty). \qquad (9.75)$$

By Theorem 1.1, we may assume that $R \in C_1$. Then, $\phi^k(\beta, R) = \sum_{(i,a)} x_{ia}(R) r_i^k(a)$, and consequently,

$$\phi^\lambda(\beta, R) = \sum_{(i,a)} x_{ia}(R) r_i^\lambda(a) > \sum_{(i,a)} x_{ia}(f_*) r_i^\lambda(a) = \phi^\lambda(\beta, f_*^\infty). \qquad (9.76)$$

which contradicts the optimality of f_*^∞ for $MDP(\lambda)$. □

Lemma 9.35
Let R_* be β-efficient for some β with $\beta_j > 0$ for all $j \in S$ and $\sum_j \beta_j = 1$. Then, R_* is uniform efficient.

Proof
Suppose that there exists a policy R such that $\phi_j^k(R) \geq \phi_j^k(f_*^\infty)$ for all $j \in S$ and all $1 \leq k \leq m$, with strict inequality holding for some k_* and some j_*. Take $\beta_j > 0$ for all $j \in S$ and $\sum_j \beta_j = 1$. Then, we have

$$\phi^k(\beta, R) \geq \phi^k(\beta, R_*) \text{ for all } 1 \leq k \leq m \text{ and with strict inequality holding for } k_*. \qquad (9.77)$$

which implies that R_* is not a β-efficient policy. □

The shall present two examples. The first examples shows that if $\beta_j = 0$ for some $j \in S$, then Lemma 9.35 no longer holds. The second example shows that a uniform policy is not always β-efficient, even when $\beta_j > 0$ for all $j \in S$.

Example 9.16

Consider the following multi-objective MDP with $k = 2$.
Let $S = \{1, 2\};\ A(1) = \{1, 2\},\ A(2) = \{1\};\ p_{11}(1) = 1,\ p_{12}(1) = 0;\ p_{11}(2) = 1,\ p_{12}(2) = 0;$ $p_{21}(1) = 0,\ p_{22}(1) = 1;\ r_1^1(1) = 3,\ r_1^1(2) = 3,\ r_2^1(1) = 2;\ r_1^2(1) = 1,\ r_1^2(2) = 3,\ r_2^2(1) = 2.$
Take $\beta_1 = 0,\ \beta_2 = 1$. There are two deterministic policies: f_1^∞ with $f_1(1) = 1,\ f_1(2) = 1$ and f_2^∞ with $f_2(1) = 2,\ f_2(2) = 1$. The average rewards are:

$$\phi_1^1(f_1^\infty) = 3;\quad \phi_2^1(f_1^\infty) = 2;\quad \phi_1^2(f_1^\infty) = 1;\quad \phi_2^2(f_1^\infty) = 2;\quad \phi_1(\beta, f_1^\infty) = 2;\quad \phi_2(\beta, f_1^\infty) = 2;$$

$$\phi_1^1(f_2^\infty) = 3;\quad \phi_2^1(f_2^\infty) = 2;\quad \phi_1^2(f_2^\infty) = 1;\quad \phi_2^2(f_2^\infty) = 2;\quad \phi_1(\beta, f_2^\infty) = 2;\quad \phi_2(\beta, f_2^\infty) = 2.$$

Hence, f_1^∞ is β-efficient, because $\phi_k(\beta, R) = 2$ for $k = 1, 2$ and for all policies R, but not uniform efficient, because $\phi_j^k(f_2^\infty) \geq \phi_j^k(f_1^\infty)$ for all $j \in S$ and for $k = 1, 2$ and $\phi_1^2(f_2^\infty > \phi_1^k(f_1^\infty$.

Example 9.17

Consider the following multi-objective MDP with $k = 2$.
Let $S = \{1, 2\};\ A(1) = A(2) = \{1, 2\};\ p_{11}(1) = 1,\ p_{12}(1) = 0;\ p_{11}(2) = 1,\ p_{12}(2) = 0;$ $p_{21}(1) = 0,\ p_{22}(1) = 1;\ p_{21}(1) = 0,\ p_{22}(1) = 1;\ r_1^1(1) = 5,\ r_1^1(2) = 4,\ r_2^1(1) = 5,\ r_2^1(2) = 7;$ $r_1^2(1) = 5,\ r_1^2(2) = 7,\ r_2^2(1) = 7,\ r_2^2(2) = 5$. Take $\beta_1 = \beta_2 = \frac{1}{2}$.
Consider the $MDP(\lambda)$ with $\lambda_1 = \lambda_2 = \frac{1}{2}$. Then. $r_1^\lambda(1) = 5,\ r_1^\lambda(2) = \frac{11}{2},\ r_2^\lambda(1) = 6,\ r_2^\lambda(2) = 6$.
It is easy to verify that the deterministic policy f_1^∞ with $f_1(1) = 1,\ f_1(2) = 1$ is an optimal policy for $MDP(\lambda)$. Hence, by Theorem 9.47, f_1^∞ is uniform efficient.
Let f_2^∞ be such that $f_2(1) = 2,\ f_2(2) = 2$. Then, $\phi_1(\beta, f_2^\infty) = \frac{11}{2}$ and $\phi_2(\beta, f_2^\infty) = 6$. Because $\phi_1(\beta, f_1^\infty) = 5$ and $\phi_2(\beta, f_1^\infty) = \frac{9}{2}$, f_1^∞ is not β-efficient. Therefore, a uniform policy is not always β-efficient, even when $\beta_j > 0$ for all $j \in S$.

In the unichain case, the next theorem shows that the concept uniform efficient is equivalent with β-efficient for any initial distribution β.

Theorem 9.48

In the unichain case, a policy R_* is uniform efficient if and only if R_* is β-efficient for any initial distribution β.

Proof

In the unichain case the average reward is independent of the initial distribution. Therefore, we use the notation $\phi_k(R)$ in stead of $\phi_k(\beta, R)$ for the average reward with respect to rewards $r_i^k(a)$. Hence, the concept of a β-efficient policy is independent of β, i.e. a policy R_* is β-efficient is there does not exists a policy R such that $\phi_k(R) \geq \phi_k(R_*),\ 1 \leq k \leq m$ with strict inequality for some k. A policy is uniform efficient if there does not exists a policy R such that $\phi_j^k(R) \geq \phi_j^k(R_*)$ for $j \in S$ and all $k = 1, 2, \ldots, m$, with strict inequality holding for some k and some j. Because, in the unichain case, the average reward is independent of the initial state, the two definition are equal.

□

Remark

Suppose that we want to maximize lexicographically the functions $\phi^k(\beta, R)$ for $k = 1, 2, \dots, m$. A policy R^* which is lexicographically maximal with respect to $\phi^1(\beta, R),\ \phi^2(\beta, R), \dots, \phi^m(\beta, R)$ is a *lexicographically efficient* policy.

To determine a lexicographically efficient policy, we compute an optimal solution, say (x^1, y^1) of the linear program

$$max \left\{ \sum_{(i,a)} r_i^1(a)x_i(a) \;\middle|\; \begin{array}{rcl} \sum_{(i,a)} \{\delta_{ij} - p_{ij}(a)\}x_i(a) & = & 0,\ j \in S \\ \sum_a x_j(a) \quad + \sum_{(i,a)} \{\delta_{ij} - p_{ij}(a)\}y_i(a) & = & \beta_j,\ j \in S \\ x_i(a), y_i(a) & \geq & 0,\ (i,a) \in S \times A \end{array} \right\}. \tag{9.78}$$

Next, we solve the following linear program with one additional constraint

$$max \left\{ \sum_{(i,a)} r_i^2(a)x_i(a) \;\middle|\; \begin{array}{rcl} \sum_{(i,a)} \{\delta_{ij} - p_{ij}(a)\}x_i(a) & = & 0,\ j \in S \\ \sum_a x_j(a) \quad + \sum_{(i,a)} \{\delta_{ij} - p_{ij}(a)\}y_i(a) & = & \beta_j,\ j \in S \\ \sum_{(i,a)} r_i^1(a)x_i(a) & = & \sum_{(i,a)} r_i^1(a)x_i^1(a) \\ x_i(a), y_i(a) & \geq & 0,\ (i,a) \in S \times A \end{array} \right\}. \tag{9.79}$$

Continuing in this way we stop either when we find for some $1 \leq k \leq m$ an optimal solution (x^k, y^k) in which x^k is unique or when we have solved all m linear programs. Let (x, y) be the finally obtained solution. Then, as shown in Section 9.2.6, we can construct a convergent Markov policy R such that $x(R) = x$. This policy is obviously a lexicographically efficient solution.

9.4 The linear program approach for average rewards revisited

We consider in this section linear programming for MDPs with average rewards and with respect to an arbitrary fixed initial distribution β. So, we allow $\beta_j = 0$ for some states j. We first present an interpretation of the y-variables of the linear program. We then show for constrained MDPs that the linear program can be obtained from an equivalent unconstrained Lagrange formulation of the optimization problem. This shows the connection between the linear program approach.

For any policy R the average reward with respect to an arbitrary fixed initial distribution β is denoted by $\phi(\beta, R)$ and defined by

$$\phi(\beta, R) := \liminf_{T \to \infty} \frac{1}{T} \sum_{t=1}^{T} \sum_{j \in S} \beta_j \cdot \sum_{(i,a)} \mathbb{P}_R\{X_t = i,\ Y_t = a \mid X_1 = j\} \cdot r_i(a).$$

The optimization problem is to find the value $\phi(\beta)$ and a *β-optimal policy* R^*, where $\phi(\beta)$ and R^* have to satisfy

$$\phi(\beta) = \sup_R \phi(\beta, R) = \phi(\beta, R^*) = \sum_i \beta_i \phi_i. \tag{9.80}$$

In order to find the value and a β-optimal policy we consider the dual pair of linear programs (9.59) and (9.60). Note that, because $\beta_j = 0$ is allowed for some states j, ϕ is an optimal solution

of (9.60), but not necessarily unique. Furthermore, since it may occur that in states j with $\beta_j = 0$, we have $\sum_a x_j(a) = \sum_a y_j(a) = 0$, for a feasible solution (x, y) of (9.59), and consequently also $\sum_{(i,a)} p_{ij}(a)y_i(a) = 0$.
For any $\pi^\infty \in C(S)$ we define $x(\pi)$ and $y(\pi)$, as in (5.35) and (5.36), by

$$\begin{cases} x_{ia}(\pi) & := & \{\beta^T P^*(\pi)\}_i \cdot \pi_{ia}, \;\; (i,a) \in S \times A \\ y_{ia}(\pi) & := & \{\beta^T D(\pi) + \gamma^T P^*(\pi)\}_i \cdot \pi_{ia}, \;\; (i,a) \in S \times A \end{cases} \tag{9.81}$$

where γ is defined as in (5.36). Similar as in Theorem 5.19 it can be shown that $\big(x(\pi), y(\pi)\big)$ is a feasible solution of (9.59). Given a feasible solution (x, y) of (9.59), we define a stationary policy $\pi^\infty(x, y)$ by (9.61).

Theorem 9.49

The correspondence between the stationary policies and the feasible solutions of program (9.59) preserves the optimality property , i.e.

(1) If π^∞ is a β-optimal policy, then $\big(x(\pi), y(\pi)\big)$ is an optimal solution of (9.59) and the optimal value of (9.59) equals $\phi(\beta)$.

(2) If (x, y) is an optimal solution of (9.59), then the stationary policy $\pi^\infty(x, y)$ is a β-optimal policy.

Proof

(1) If π^∞ is a β-optimal policy, then,

$\sum_{(i,a)} r_i(a)x_{ia}(\pi) = \sum_{(i,a)} r_i(a)\{\beta^T P^*(\pi)\}_i \cdot \pi_{ia} = \{\beta^T P^*(\pi)\}_i r_i(\pi) = \beta^T \phi(\pi^\infty) = \phi(\beta)$,

i.e. part (1) of the theorem holds.

(2) This result is shown in Theorem 9.40. □

Given some $x(R) \in L$ (for a definition of L see section 9.2.6) and, we define the *biased total occupation* $y^T(R)$ for any $T \in \mathbb{N}$ by

$$y^T_{ja}(R) := \sum_{t=1}^{T} \big\{ \sum_{i \in S} \beta_i \cdot \mathbb{P}\{X_t = j, \; Y_t = a \mid X_1 = i\} - x_{ja}(R)\big\}, \; (j,a) \in S \times A. \tag{9.82}$$

Define the *average biased occupation* $\overline{y}^T_{ja}(R)$ by

$$\overline{y}^T_{ja}(R) := \frac{1}{T} \sum_{t=1}^{T} y^t_{ja}(R), \; (j,a) \in S \times A. \tag{9.83}$$

Let $\{T_n\}$ be a subsequence of $\{1, 2, \ldots\}$ along which $x_{ja}(R) = \lim_{n\to\infty} x^{T_n}_{ja}(R), \; (j,a) \in S \times A$. Pick a further subsequence $\{t_n\}$ of $\{T_n\}$ along which some (possibly infinite) limit $y(R)$ of $\{\overline{y}^{t_n}(R)\}$ exists, i.e. $y_{ja}(R) = \lim_{n\to\infty} \overline{y}^{t_n}_{ja}(R)$ for all $(j,a) \in S \times A$. Let $Y\big(x(R)\big)$ denote the set of all such limit points. We call any $y(R) \in Y(R)$ a *deviation measure*. The following lemma relates the quantities $x(R)$ and $y(R)$ to the decision variables of the linear program (9.59).

Lemma 9.36

Given $x(R) \in L$ and $y(R) \in Y\big((x(R)\big)$ such that $y_{ja}(R)$ is finite for all $(j,a) \in S \times A$, then $\big(x(R), y(R)\big)$ satisfies the constraints of the linear program (9.59), except the requirement $y_i(a) \geq 0$ for all $(i,a) \in S \times A$.

Proof

$\sum_a \mathbb{P}_{\beta,R}\{X_t = j, Y_t = a\} = \mathbb{P}_{\beta,R}\{X_t = j\} = \sum_{(i,a)} \mathbb{P}_{\beta,R}\{X_{t-1} = i, Y_{t-1} = a\} \cdot p_{ij}(a)$ for $t \geq 2$.

By averaging we obtain

$\frac{1}{T-1} \sum_{t=2}^T \sum_a \mathbb{P}_{\beta,R}\{X_t = j, Y_t = a\} = \frac{1}{T-1} \sum_{t=2}^T \sum_{(i,a)} \mathbb{P}_{\beta,R}\{X_{t-1} = i, Y_{t-1} = a\} \cdot p_{ij}(a),\ T \geq 2.$

Taking a sequence $\{T_n\}$ such that $x_{ja}(R) = \lim_{n\to\infty} x_{ja}^{T_n}(R$ for all $(j,a) \in S \times A$, gives

$$\sum_a x_{ja}(R) = \sum_{(i,a)} p_{ij}(a) x_{ia}(R) \text{ for all } j \in S,$$

i.e. $x(R)$ satisfies $\sum_{(i,a)} \{\delta_{ij} - p_{ij}(a)\} x_{ia}(R) = 0$ for all $j \in S$, the first set of equalities of (9.59).

Furthermore, we can write for $T \geq 2$ and all $j \in S$,

$$\begin{aligned}
&\sum_{t=2}^T \sum_a \big\{ \mathbb{P}_{\beta,R}\{X_t = j, Y_t = a\} - x_{ja}(R) \big\} \\
&\quad = \sum_{t=2}^T \sum_{(i,a)} \big\{ \mathbb{P}_{\beta,R}\{X_{t-1} = i, Y_{t-1} = a\} \cdot p_{ij}(a) - \sum_a x_{ja}(R) \big\} \\
&\quad = \sum_{t=2}^T \sum_{(i,a)} \big\{ \mathbb{P}_{\beta,R}\{X_{t-1} = i, Y_{t-1} = a\} \cdot p_{ij}(a) - \sum_{(i,a)} p_{ij}(a) x_{ia}(R) \big\} \\
&\quad = \sum_{t=2}^T \sum_{(i,a)} p_{ij}(a) \big\{ \mathbb{P}_{\beta,R}\{X_{t-1} = i, Y_{t-1} = a\} - x_{ia}(R) \big\} \\
&\quad = \sum_{(i,a)} p_{ij}(a) y_{ia}^{T-1}(R).
\end{aligned}$$

For $t = 1$, we have $\sum_a \big\{ \mathbb{P}_{\beta,R}\{X_t = j, Y_t = a\} - x_{ja}(R) \big\} = \beta_j - \sum_a x_{ja}(R)$. Hence, for $T \geq 2$, we obtain

$$\sum_{t=1}^T \sum_a \big\{ \mathbb{P}_{\beta,R}\{X_t = j, Y_t = a\} - x_{ja}(R) \big\} = \beta_j - \sum_a x_{ja}(R) + \sum_{(i,a)} p_{ij}(a) y_{ia}^{T-1}(R).$$

Therefore, for $T = 2, 3, \ldots$ and all $j \in S$,

$$\sum_a x_{ja}(R) + \sum_a \frac{1}{T-1} \sum_{t=2}^T y_{ja}^t(R) - \sum_{(i,a)} p_{ij}(a) \cdot \frac{1}{T-1} \sum_{t=2}^T y_{ia}^{t-1}(R) = \beta_j.$$

Taking a subsequence for which the average biased occupation converges to $y(R)$, we obtain

$$\sum_a x_{ja}(R) + \sum_a y_{ja}(R) - \sum_{(i,a)} p_{ij}(a) y_{ia}(R) = \beta_j \text{ for all } j \in S,$$

i.e. $\big(x(R), y(R)\big)$ satisfies the set of equalities of (9.59). Clearly, $x_{ja}(R) \geq 0$ for all $(j,a) \in S \times A$.

□

The next corollary certifies the name deviation measure for an element $y(R) \in Y(R)$.

Corollary 9.11

Given $x(R) \in L$ and $y(R) \in Y\big(x(R)\big)$ such that $y_{ja}(R)$ is finite for all $(j,a) \in S \times A$, then $\sum_{(j,a)} y_{ja}(R) = 0$.

Proof

$$\begin{aligned}
\sum_{(j,a)} y_{ja}(R) &= \sum_{(j,a)} \Big\{ \lim_{n\to\infty} \frac{1}{T_n} \sum_{t=1}^{T_n} \sum_{s=1}^t \big\{ \mathbb{P}_{\beta,R}\{X_t = j, Y_t = a\} - x_{ja}(R) \big\} \Big\} \\
&= \lim_{n\to\infty} \frac{1}{T_n} \sum_{t=1}^{T_n} \sum_{s=1}^t \big\{ \sum_{(j,a)} \mathbb{P}_{\beta,R}\{X_t = j, Y_t = a\} - \sum_{(j,a)} x_{ja}(R) \big\} \\
&= \lim_{n\to\infty} \frac{1}{T_n} \sum_{t=1}^{T_n} \sum_{s=1}^t (1 - 1) = 0.
\end{aligned}$$

□

For $\big(x(R), y(R)\big)$ to be a feasible solution of (9.59), we need to satisfy $y_{ja}(R) \geq 0$ for all (j,a). If $x_{ja}(R) = 0$, then, by (9.82), $y^T_{ja}(R) \geq 0$ for all $T \geq 1$, and consequently, $\overline{y}^T_{ja}(R) \geq 0$ for all $T \geq 1$, implying $y_{ja}(R) \geq 0$.
If $x_{ja}(R) > 0$, then, set $\hat{y}_{ja}(R) := y_{ja}(R) - c \cdot x_{ja}(R)$, where $c := min_{i,a}\big\{\frac{y_{ia}(R)}{x_{ia}(R)} \mid x_{ia}(R) > 0\big\}$.
Then, $\hat{y}_{ja}(R) \geq 0$ for all $(i,a) \in S \times A$ and $\big(x(R), \hat{y}(R)\big)$ satisfies the constraints of (9.59), because

$$\begin{array}{rcl} \sum_{(i,a)} \{\delta_{ij} - p_{ij}(a)\}\hat{y}_{ia}(R) & = & \sum_{(i,a)} \{\delta_{ij} - p_{ij}(a)\}y_{ia}(R) - c \cdot \sum_{(i,a)} \{\delta_{ij} - p_{ij}(a)\}x_{ja}(R) \\ & = & \sum_{(i,a)} \{\delta_{ij} - p_{ij}(a)\}y_{ia}(R) - c \cdot 0 = \sum_{(i,a)} \{\delta_{ij} - p_{ij}(a)\}y_{ia}(R). \end{array}$$

□

<u>Remark</u>

Given a feasible solution (x,y) of (9.59), we can construct, by Algorithm 9.5, a policy R such that $x = x(R)$ and $R \in L(M) \cap L(C)$. Consequently, $\big(x(R) = x, \hat{y}(R)\big)$ is a feasible solution of (9.59). However, in general $y \neq y(R)$; even $y = y(R) + \gamma \cdot x(R)$ for some γ is not true, in general, as the next example shows.

Example 9.9 (continued)

The constraints of linear programming problem are

$$\begin{array}{rrrrrrrrrrrrrrrr} & x_1(1) & + & x_1(2) & & & & & & & & & & & = & 0 \\ - & x_1(1) & & & & & - & x_3(2) & & & & & & & = & 0 \\ & & - & x_1(2) & & & + & x_3(2) & & & & & & & = & 0 \\ & x_1(1) & + & x_1(2) & & & & & + & y_1(1) & + & y_1(2) & - & y_3(2) & = & \frac{1}{4} \\ & & & & x_2(1) & & & & - & y_1(1) & & & + & y_3(2) & = & \frac{3}{16} \\ & & & & & x_3(1) & + & x_3(2) & & & - & y_1(2) & & & = & \frac{9}{16} \end{array}$$

$$x_1(1),\ x_1(2),\ x_2(1),\ x_3(1),\ x_3(2)\ \geq\ 0$$

Consider the feasible solution (x,y), where $x_{11} = 0,\ x_{12} = 0,\ x_{21} = \frac{1}{2},\ x_{31} = \frac{1}{2},\ x_{32} = 0;\ y_{11} = 0,\ y_{12} = \frac{1}{4},\ y_{21} = 0,\ y_{31} = 0,\ y_{32} = \frac{5}{16}$. By Algorithm 9.5 we obtain the policy $R = (\pi^1, f^\infty)$, where $\pi^1_{11} = 1,\ \pi^1_{12} = 0,\ \pi^1_{21} = 1,\ \pi^1_{31} = \frac{8}{9},\ \pi^1_{32} = \frac{1}{9}$ and $f(1) = f(2) = f(3) = 1$.
It is easy to verify that $x(R) = x$ and $y_{11}(R) = \frac{1}{4},\ y_{12}(R) = 0,\ y_{21}(R) = -\frac{5}{16},\ y_{31}(R) = 0,\ y_{32}(R) = \frac{1}{16}$. Notice that $\sum_{(i,a)} y_{ia}(R) = 0$. The constant $c = -\frac{5}{16}$ and $\hat{y}_{11}(R) = \frac{1}{4},\ \hat{y}_{12}(R) = 0,\ \hat{y}_{21}(R) = -\frac{5}{16},\ \hat{y}_{31}(R) = 0,\ \hat{y}_{32}(R) = \frac{1}{16}$. It is easy to see that $\big(x(R) = x, \hat{y}(R)\big)$ is a feasible solution of the linear program, but $\hat{y}(R) + \gamma \cdot x(R) \neq y$ for every scalar γ.

Unlike the state-action frequencies $x_{ja}(R)$, which sum up to 1 under any policy R, the deviation measures are not bounded, in general. This is demonstrated in the following simple example with only one state and two actions.

Example 9.18

Let $S = \{1\};\ A(1) = \{1,2\};\ p_{11}(1) = 1,\ p_{11}(2) = 1$.
Let R_t be the policy that takes action 1 at time $1, 2, ;t$ and action 2 at time $t+1, t+2, \ldots$.
Hence, $x_{11}(R_t) = 0$ and $x_{12}(R_t) = 1$ for any fixed t.
Since $\mathbb{P}_{\beta,R_t}\{X_s = 1,\ Y_s = 1\} = \begin{cases} 1 & 1 \leq s \leq t \\ 0 & s \geq t+1 \end{cases}$, we obtain $y_{11}(R_t) = \begin{cases} T & 1 \leq T \leq t \\ t & T \geq t+1 \end{cases}$.
Hence, $\overline{y}^T_{11}(R_t) \to t$ for $T \to \infty$, implying $y_{11}(R_t) = t$. Therefore, $y_{11}(R_t)$is in general unbounded.

Lemma 9.37

Assume that the MDP is unichained. Let R_1 and R_2 be two policies such that

(1) $x(R_1) = x(R_2)$ *for some* $x(R_1) \in X(R_1)$ *and* $x(R_2) \in X(R_2)$*;*

(2) $\frac{y_{ja}(R_1)}{\sum_a y_{ja}(R_1)} = \frac{y_{ja}(R_2)}{\sum_a y_{ja}(R_2)}$, $(j,a) \in S \times A$ *for some finite* $y(R_1) \in Y\big(x(R_1)\big)$ *and for some finite* $y(R_2) \in Y\big(x(R_2)\big)$.

Then, $y(R_1) = y(R_2)$.

Proof

Let $\pi_{ja} := \frac{y_{ja}(R_1)}{\sum_a y_{ja}(R_1)} = \frac{y_{ja}(R_2)}{\sum_a y_{ja}(R_2)}$, $(j,a) \in S \times A$. Then, $P(\pi)$ is unichained. Let $x_j(R_k)$ and $y_j(R_k)$ be defined by $x_j(R_k) := \sum_a x_{ja}(R_k)$ and $y_j(R_k) := \sum_a y_{ja}(R_k)$ for $j \in S$ and $k = 1,2$. Hence, $\{x(R_1)\}^T + \{y(R_1)\}^T\{I - P(\pi)\} = \beta^T$ and $\{x(R_2)\}^T + \{y(R_2)\}^T\{I - P(\pi)\} = \beta^T$. By subtraction and because $x(R_1) = x(R_2)$, we have for $y := y(R_1) - y(R_2)$, $y^T\{I - P(\pi)\} = 0$, implying $y^T = y^T P(\pi)^*$. Since $P(\pi)^*$ has identical rows, say π^*, we obtain, $y = (\sum_j y_j) \cdot \pi^*$. Because $\sum_j y_j = \sum_j y_j(R_1) - \sum_j y_j(R_2) = \sum_{j,a} y_{ja}(R_1) - \sum_{j,a} y_{ja}(R_2) = 0 - 0 = 0$, we have shown that $y(R_1) = y(R_2)$. □

Next, we will consider the constrained MDP problem by a Lagrange approach. The constrained MDP problem was formulated in (9.21) as

$$sup_R\,\{\phi(\beta,R) \mid c^k(\beta,R) \le b_k,\ k = 1,2,\dots,m\}. \tag{9.84}$$

where $\phi(\beta,R) := \liminf_{T\to\infty} \frac{1}{T}\sum_{t=1}^T \sum_{j\in S} \beta_j \cdot \sum_{(i,a)} \mathbb{P}_R\{X_t = i,\ Y_t = a \mid X_1 = j\} \cdot r_i(a)$ and

$$\begin{array}{rcl} c^k(\beta,R) & := & \liminf_{T\to\infty} \frac{1}{T}\sum_{t=1}^T \sum_{j\in S} \beta_j \cdot \sum_{(i,a)} \mathbb{P}_R\{X_t = i,\ Y_t = a \mid X_1 = j\} \cdot c_i^k(a) \\ & = & -\liminf_{T\to\infty} \frac{1}{T}\sum_{t=1}^T \sum_{j\in S} \beta_j \cdot \sum_{(i,a)} \mathbb{P}_R\{X_t = i,\ Y_t = a \mid X_1 = j\} \cdot \{-c_i^k(a)\} \\ & = & -\phi^k(\beta,R), \end{array}$$

where $\phi^k(\beta,R)$ is, given initial distribution β and policy R, the average reward with respect to immediate rewards $r_j^k(a) := -c_j^k(a)$ $(j,a) \in S \times A$. Hence, an equivalent formulation of (9.84) is

$$sup_R\,\{\phi(\beta,R) \mid -\phi^k(\beta,R) \le b_k,\ k = 1,2,\dots,m\}. \tag{9.85}$$

Since $L = L(C)$ (see Theorem 9.21), the problem is equivalent to

$$sup_{x(R)\in L(C)}\Big\{\sum_{(i,a)} r_j(a)x_{ja}(R)\ \Big|\ -\sum_{(i,a)} r_j^k(a)x_{ja}(R) \le b_k,\ k = 1,2,\dots,m\Big\}. \tag{9.86}$$

The Lagrange function for problem (9.86) becomes for any $\lambda \in \mathbb{R}_+^M$

$$\begin{array}{rcl} L(\beta,R,\lambda) & = & \sum_{(j,a)} r_j(a)x_{ja}(R) - \sum_{k=1}^m \lambda_k \cdot \{-\sum_{(i,a)} r_j^k(a)x_{ja}(R) - b_k\} \\ & = & \sum_{(j,a)} \{r_j(a) + \sum_{k=1}^m \lambda_k r_j^k(a)\}x_{ja}(R) + \sum_{k=1}^m \lambda_k b_k \\ & = & \sum_{(j,a)} \big\{r_j(a) + \sum_{k=1}^m \lambda_k\{r_j^k(a) + b_k\}\big\}x_{ja}(R), \end{array}$$

the last equality because $\sum_{(i,a)} x_{ja}(R) = 1$. Therefore, $sup_R\,L(\beta,R,\lambda)$ is the value of the MDP with immediate rewards $\overline{r}_j(a) := r_j(a) + \sum_{k=1}^m \lambda_k\{r_j^k(a) + b_k\}$, $(j,a) \in S \times A$. Hence, $sup_R\,L(\beta,R,\lambda)$ is the optimum value of the following linear program

$$min_{u,v}\left\{\sum_j \beta_j v_j \;\middle|\; \begin{array}{llll} \sum_j\{\delta_{ij}-p_{ij}(a)\}v_j & \geq & 0, & (i,a)\in S\times A \\ v_i \; + \; \sum_j\{\delta_{ij}-p_{ij}(a)\}u_j & \geq & \overline{r}_i(a), & (i,a)\in S\times A \end{array}\right\}.$$

i.e. the linear program $min_{u,v}\sum_j \beta_j v_j$ under the constraints

$$\begin{array}{llll} \sum_j\{\delta_{ij}-p_{ij}(a)\}v_j & \geq & 0, & (i,a)\in S\times A \\ v_i \; + \; \sum_j\{\delta_{ij}-p_{ij}(a)\}u_j - \sum_{k=1}^m \lambda_k\{r_j^k(a)+b_k\} & \geq & r_i(a), & (i,a)\in S\times A \end{array}$$

Therefore, $min_{\lambda>0}\, sup_R\, L(\beta,R,\lambda)$ is the optimum value of the linear program $min_{u,v,\lambda}\sum_j \beta_j v_j$ under the constraints

$$\begin{array}{llll} \sum_j\{\delta_{ij}-p_{ij}(a)\}v_j & \geq & 0, & (i,a)\in S\times A \\ v_i \; + \; \sum_j\{\delta_{ij}-p_{ij}(a)\}u_j - \sum_{k=1}^m \lambda_k\{r_j^k(a)+b_k\} & \geq & r_i(a), & (i,a)\in S\times A \end{array}$$

The dual program of this LP becomes

$$max\left\{\sum_{(i,a)} r_i(a)x_i(a) \;\middle|\; \begin{array}{lll} \sum_{(i,a)}\{\delta_{ij}-p_{ij}(a)\}x_i(a) & = & 0,\ j\in S \\ \sum_a x_j(a) + \sum_{(i,a)}\{\delta_{ij}-p_{ij}(a)\}y_i(a) & = & \beta_j,\ j\in S \\ -\sum_{(i,a)}\{r_j^k(a)+b_k\}x_i(a) & \leq & 0,\ k=1,2,\dots,m \\ x_i(a), y_i(a)\geq 0,\ (i,a)\in S\times A & & \end{array}\right\} \tag{9.87}$$

which is (again using $\sum_{(i,a)} x_i(a)=1$) exactly the linear program (9.27) for solving the constrained MDP, namely

$$max\left\{\sum_{(i,a)} r_i(a)x_i(a) \;\middle|\; \begin{array}{lll} \sum_{(i,a)}\{\delta_{ij}-p_{ij}(a)\}x_i(a) & = & 0,\ j\in S \\ \sum_a x_j(a) \; + \; \sum_{(i,a)}\{\delta_{ij}-p_{ij}(a)\}y_i(a) & = & \beta_j,\ j\in S \\ \sum_{(i,a)} c_i^k(a)x_i(a) & \leq & b_k,\ k=1,2,\dots,m \\ x_i(a), y_i(a)\geq 0,\ (i,a)\in S\times A & & \end{array}\right\}. \tag{9.88}$$

We close this section by showing the following minimax result for the function $L(\beta,R,\lambda)$.

Theorem 9.50

$min_{\lambda>0}\, sup_R\, L(\beta,R,\lambda) = sup_R\, min_{\lambda>0}\, sup_R\, L(\beta,R,\lambda)$.

Proof

Since $L=L(C)$, we may restrict the policy space to C_1, the set for which $X(R)$ consists of one element. For $R\in C_1$, we have $L(\beta,R,\lambda)=\sum_{(j,a)}\big\{r_j(a)+\sum_{k=1}^m \lambda_k\{r_j^k(a)+b_k\}\big\}x_{ja}(R)$ with $x(R)\in L(C)=Q$, where Q is a convex polytope, and $\lambda\in\big\{\mathbb{R}_+\cup\{\infty\}\big\}^m$, which set is convex and a compactified space. It follows from the Minimax Theorem (see e.g. [187], p. 208) that the min and sup can be interchanged, i.e. $min_{\lambda>0}\, sup_R\, L(\beta,R,\lambda) = sup_R\, min_{\lambda>0}\, sup_R\, L(\beta,R,\lambda)$. □

Remark

The derivation, by the Lagrange approach, of the linear program for the constrained MDP can easily be applied for the discounted reward criterion and the total reward criterion with for the last criterion as policy space the set of transient policies.

9.5 Mean-variance tradeoffs

9.5.1 Formulations of the problem

In many areas of application, a decision maker may wish to incorporate his attitude toward risk or variability when choosing a policy. One measure of risk is the variance of the rewards generated by a policy. Frequently one considers tradeoffs between return and risk. Examples of this include a dynamic investment model in which the investor may accept a lower than optimal return to achieve reduced variability in return, and a queueing control model, in which the controller might prefer a policy which results in greater but less variable waiting times. These mean-variance tradeoffs may be analyzed in an MDP using the long-run state-action frequencies.
Given an initial distribution β and a policy R the *long-run variance* $V(\beta, R)$ is defined by

$$\begin{array}{rcl} V(\beta,R) & = & \limsup_{T\to\infty} \frac{1}{T}\sum_{t=1}^T \sum_i \beta_i \cdot \mathbb{E}_{i,R}\{r_{X_t}(Y_t) - \phi(\beta,R)\}^2 \\ & = & \limsup_{T\to\infty} \frac{1}{T}\sum_{t=1}^T \sum_i \beta_i \cdot \sum_{j,a} \mathbb{P}_R\{X_t=j,\ Y_t=a \mid X_1=i\}\{r_j(a)-\phi(\beta,R)\}^2 \end{array} \tag{9.89}$$

If $R \in C_1$ the long-run state-action frequencies are unique and the long-run variance can be written as

$$\begin{array}{rcl} V(\beta,R) & = & \sum_{j,a} x_{ja}(R)\{r_j(a)-\phi(\beta,R)\}^2 \\ & = & \sum_{j,a} x_{ja}(R)r_j^2(a) - 2\sum_{j,a} x_{ja}(R)r_j(a)\phi(\beta,R) + \sum_{j,a} x_{ja}(R)\phi(\beta,R)^2 \\ & = & \sum_{j,a} x_{ja}(R)r_j^2(a) - \phi(\beta,R)^2 \\ & = & \sum_{j,a} x_{ja}(R)r_j^2(a) - \{\sum_{j,a} x_{ja}(R)r_j(a)\}^2. \end{array} \tag{9.90}$$

Example 9.19
Let $S=\{1,2,3\}$; $A(1)=A(2)=\{1\}$; $A(3)\{1,2\}$; $p_{11}(1)=p_{12}(1)=0,\ p_{13}(1)=1$;
$p_{21}(1)=p_{22}(1)=0,\ p_{23}(1)=1$; $p_{31}(1)=1,\ p_{32}(1)=p_{33}(1)=0$; $p_{31}(2)=0$,
$p_{32}(2)=1,\ p_{33}(2)=0$; $r_1(1)=0$; $r_2(1)=2$; $r_3(1)=8$; $r_1(1)=4$. $\beta_1=\beta_2=\frac{1}{4},\ \beta_3=\frac{1}{2}$.
There are two deterministic policies f_1^∞ with $f(3)=1$ and f_2^∞ with $f(3)=2$.
For these policies we obtain:

$$\begin{array}{l} x_{11}(f_1)=\frac{1}{2};\quad x_{21}(f_1)=0;\quad x_{31}(f_1)=\frac{1}{2};\quad x_{32}(f_1)=0;\quad \phi(\beta,f_1^\infty)=4;\quad V(\beta,f_1^\infty)=16. \\ x_{11}(f_2)=0;\quad x_{21}(f_2)=\frac{1}{2};\quad x_{31}(f_2)=0;\quad x_{32}(f_2)=\frac{1}{2};\quad \phi(\beta,f_1^\infty)=3;\quad V(\beta,f_1^\infty)=1. \end{array} .$$

Observe that f_1^∞ is average optimal but has a considerably larger variance than f_2^∞, so that a risk averse decision maker may prefer $f_2^\infty$to f_1^∞.

There are several ways to consider the mean-variance tradeoffs. Sobel ([278]) proposed to maximize the *mean-standard deviation ratio* with upper and lower bounds on the mean. This is equivalent to minimizing the ratio of the variance and the square of the mean under the same constraints. In policy space this concept is

$$min\left\{\frac{V(\beta,R)}{\phi(\beta,R)^2}\ \middle|\ L \le \phi(\beta,R) \le U\right\}.$$

Using the state action frequencies, problem (9.91) becomes

$$min \left\{ \frac{\sum_{j,a} r_j^2(a)x_j(a) - \{\sum_{j,a} r_j(a)x_j(a)\}^2}{\{\sum_{j,a} r_j(a)x_j(a)\}^2} \;\middle|\; x \in Q;\; L \leq \sum_{j,a} r_j(a)x_j(a) \leq U \right\},$$

with polyhedron Q defined by

$$Q := \left\{ x \;\middle|\; \begin{array}{rcrcl} \sum_{(i,a)}\{\delta_{ij} - p_{ij}(a)\}x_{ia} & & & = & 0,\ j \in S \\ \sum_a x_{ja} & + & \sum_{(i,a)}\{\delta_{ij} - p_{ij}(a)\}y_{ia} & = & \beta_j,\ j \in S \\ & & x_{ia},\ y_{ia} & \geq & 0,\ (i,a) \in S \times A \end{array} \right\}.$$

This minimization program is equivalent to

$$max \left\{ \frac{-\sum_{j,a} r_j^2(a)x_j(a)}{\{\sum_{j,a} r_j(a)x_j(a)\}^2} \;\middle|\; x \in Q;\; L \leq \sum_{j,a} r_j(a)x_j(a) \leq U \right\}. \tag{9.91}$$

Kawai ([165]) has considered the problem of minimizing the *variance* subject a lower bounds on the mean, i.e.

$$min\{V(\beta, R) \mid \phi(\beta, R) \geq L\}.$$

This problem becomes in the state-action space

$$min \left\{ \sum_{j,a} r_j^2(a)x_j(a) - \Big\{ \sum_{j,a} r_j(a)x_j(a) \Big\}^2 \;\middle|\; x \in Q;\; \sum_{j,a} r_j(a)x_j(a) \geq L \right\},$$

which is equivalent to

$$max \left\{ -\sum_{j,a} r_j^2(a)x_j(a) + \Big\{ \sum_{j,a} r_j(a)x_j(a) \Big\}^2 \;\middle|\; x \in Q;\; \sum_{j,a} r_j(a)x_j(a) \geq L \right\}. \tag{9.92}$$

Filar, Kallenberg and Lee ([93]) proposed a *variance-penalized* version, i.e.

$$max\{\phi(\beta, R) - \lambda \cdot V(\beta, R)\} \text{ for some fixed penalty } \gamma \geq 0,$$

or in the x-space

$$max \left\{ \sum_{j,a} r_j(a)x_j(a) - \lambda \Big\{ \sum_{j,a} r_j^2(a)x_j(a) - \{ \sum_{j,a} r_j(a)x_j(a) \}^2 \Big\} \;\middle|\; x \in Q \right\}. \tag{9.93}$$

9.5.2 A unifying framework

In [137] Huang and Kallenberg presented a framework that unifies and extends the approaches posed above. This framework is formulated as a nonlinear program which can be solved by a parametric linear programming problem. This solution method is at least as good as any known method for the particular problems (9.91), (9.92) and (9.93). This unifying framework considers the nonlinear program

$$max \left\{ \frac{\sum_{j,a} B_j(a)x_j(a)}{D\Big(\sum_{j,a} R_j(a)x_j(a)\Big)} + C\Big(\sum_{j,a} R_j(a)x_j(a)\Big) \;\Bigg|\; x \in Q;\; L \le \sum_{j,a} R_j(a)x_j(a) \le U \right\}, \tag{9.94}$$

with the following assumptions:

(A1) the functions $D(\cdot)$ and $C(\cdot)$ are convex;

(A2) either $D(\cdot)$ is a positive constant; or $D(\cdot)$ is positive and nondecreasing, $C(\cdot)$ is nondecreasing and $\sum_{j,a} B_j(a)x_j(a) \le 0$ for every $x \in Q$.

We now show that (9.91), (9.92) and (9.93) are special cases of 9.94).

(9.91): set $B_j(a) := -r_j^2(a),\ R_j(a) := r_j(a),\ C(y) := 0,\ D(y) := y^2$.

(9.92): set $B_j(a) := -r_j^2(a),\ R_j(a) := r_j(a),\ C(y) := y^2,\ D(y) := 1$ and $U := \infty$.

(9.93): set $B_j(a) := r_j(a) - \lambda r_j^2(a),\ R_j(a) := r_j(a)\sqrt{\lambda},\ C(y) := y^2,\ D(y) := 1$, $L := -\infty$ and $U := \infty$.

9.5.3 Determination of an optimal solution

In order to solve (9.94)) we consider a parametric version of the linear program for average rewards. The parametric objective function is $\sum_{i,a} \{B_i(a) + \lambda R_i(a)\}x_j(a)$. Hence, the parametric linear program is

$$\begin{array}{lrcl}
maximize\{\sum_{i,a} \{B_i(a) + \lambda R_i(a)\}x_i(a)\} & & & \\
\text{subject to} & & & \\
\sum_{(i,a)} \{\delta_{ij} - p_{ij}(a)\}x_i(a) & & = & 0,\ j \in S \\
\sum_a x_j(a) & + \sum_{(i,a)} \{\delta_{ij} - p_{ij}(a)\}y_i(a) & = & \beta_j,\ j \in S \\
& x_i(a), y_i(a) & \ge & 0,\ (i,a) \in S \times A
\end{array} \tag{9.95}$$

It is well known (see e.g. Zoutendijk [341], p.165) that the optimum objective function of a parametric linear program is a piecewise linear convex function of the parameter λ, and that on each interval of this piecewise linear convex function an optimal solution exists with is an extreme point of the polytope of the constraints. Thus, there exists $\lambda_0 \equiv -\infty < \lambda_1 < \cdots < \lambda_{m-1} < \lambda_m \equiv +\infty$ and extreme optimal solutions (x^n, y^n) for $n = 1, 2, \dots, m$. Let $k+1$ and $j+1$ be respectively the smallest integers among $1, 2, \dots, m$ such that

$$\sum_{i,a} R_i(a)x_i^{k+1}(a) > U \text{ and } \sum_{i,a} R_i(a)x_i^{j+1}(a) \ge L.$$

Furthermore, let $\mu, \nu \in [0, 1]$ be such that

$$x^U = \mu x^k + (1-\mu)x^{k+1} \text{ and } x^L = \nu x^j + (1-\nu)x^{j+1}$$

satisfying $\sum_{i,a} R_i(a)x_i^U(a) = U$ and $\sum_{i,a} R_i(a)x_i^L(a) = L$.

Set $G(x) := \sum_{i,a} B_i(a)x_i(a),\ g(x) := \sum_{i,a} R_i(a)x_i(a)$ and $V(x) := \frac{G(x)}{D\big(g(x)\big)} + C\big(g(x)\big)$ for $x \in X$, and set $G_n := G(x^n),\ g_n := g(x^n)$ and $V^n := V(x^n)$ for $n = 1, 2, \dots, m$.

Theorem 9.51

(1) The nonlinear program (9.94) is feasible if and only if $g_m \geq L$ and $g_1 \leq U$.

(2) If program (9.94) is feasible with optimum value V_{opt} and optimal solution x_{opt}, then

$$V_{opt} = max\Big\{max_{j+1 \leq n \leq k}\, V\big(x^n\big), V\big(x^L\big), V\big(x^U\big)\Big\} \text{ and } x_{opt} = \begin{cases} x^n \text{ if } V\big(x^n\big) = V_{opt} \\ x^L \text{ if } V\big(x^L\big) = V_{opt} \\ x^U \text{ if } V\big(x^U\big) = V_{opt} \end{cases}$$

Proof

Part (1)

Since x^n is optimal for (9.94) for $\lambda_{n-1} \leq \lambda \leq \lambda_n$, we have

$$G_n + \lambda g_n \geq G(x) + \lambda g(x),\ \lambda_{n-1} \leq \lambda \leq \lambda_n,\ x \in X \text{ for } n = 1, 2, \dots, m. \tag{9.96}$$

For $n = 1$, we obtain $G_1 + \lambda g_1 \geq G(x) + \lambda g(x),\ -\infty < \lambda \leq \lambda_1,\ x \in X$. Hence, $g_1 \leq g(x),\ x \in X$. Similarly, for $n = m$, we have $G_m + \lambda g_m \geq G(x) + \lambda g(x),\ \lambda_{m-1} \leq \lambda < +\infty$, and consequently $g_m \geq g(x),\ x \in X$. Therefore, $g_m < L$ or $g_1 > U$ implies infeasibility of the problem.
Conversely, if program (9.94) is feasible, we have $g_1 \leq U$ and $g_m \geq L$.

Part (2)

We first show that $g_1 \leq g_{opt} \leq g_m$, where $g_{opt} = g(x_{opt})$. Let $G_{opt} = G(x_{opt})$. Again, specifying (9.96) to the cases $n = 1$ and $n = m$ gives for $x = x_{opt}$:

$$G_1 + \lambda g_1 \geq G_{opt} + \lambda g_{opt},\ -\infty < \lambda \leq \lambda_1;\ G_m + \lambda g_m \geq G_{opt} + \lambda g_{opt},\ \lambda_{m-1} \leq \lambda < +\infty.$$

Letting $\lambda \to -\infty$ in the first inequality and $\lambda \to +\infty$ in the second estableshes

$$g_1 \leq g_{opt} \leq g_m. \tag{9.97}$$

Also by (9.96)

$$\begin{cases} G_{n+1} + \lambda_{n+1} g_{n+1} \geq G_n + \lambda_{n+1} g_n \\ G_{n+1} + \lambda_n g_{n+1} = G_n + \lambda_n g_n \end{cases} \rightarrow\ (\lambda_{n+1} - \lambda_n)(g_{n+1} - g_n) \geq 0,$$

implying

$$g_{n+1} \geq g_n,\ n = 1, 2, \dots, m-1. \tag{9.98}$$

From $g_m \geq L,\ g_1 \leq U$, (9.97) and (9.98) it follows that there exists an index $1 \leq p \leq m-1$ such that $g_p \leq g_{opt} \leq g_{p+1}$, such that exactly one of the following is true:

(a) $L \leq g_p \leq g_{opt} \leq g_{p+1} \leq U$; (b) $g_p < g\big(x^L\big) = L \leq g_{opt} \leq g_{p+1} \leq U$;

(c) $L \leq g_p \leq g_{opt} \leq U = g\big(x^U\big) < g_{p+1}$; (d) $g_p < g\big(x^L\big) = L \leq g_{opt} \leq U = g\big(x^U\big) < g_{p+1}$.

Case d

In this case, we have $j = k = p$, and therefore $V_{opt} = max\Big\{V\big(x^L\big),\ V\big(x^U\big)\Big\}$. By (9.96) for $n = p$, we obtain $G_{p+1} + \lambda_p g_{p+1} = G_p + \lambda_p g_p \geq G(x) + \lambda_p g(x),\ x \in X$. Since x^L and x^U are convex combinations of x^p and x^{p+1}, we also have

$$G\big(x^L\big) + \lambda_p g\big(x^L\big) = G\big(x^U\big) + \lambda_p g\big(x^U\big) = G_{p+1} + \lambda_p g_{p+1} = G_p + \lambda_p g_p \geq G(x) + \lambda_p g(x),\ x \in X. \tag{9.99}$$

For two distinct real numbers y and z, let $c(y,z) = \frac{C(y)-C(z)}{y-z}$ and $d(y,z) = \frac{D(y)-D(z)}{y-z}$. We claim that

$$D\big(g(x^U)\big)c\big(g(x^L), g(x^U)\big) - \frac{G(x^L)}{D\big(g(x^L)\big)}d\big(g(x^L), g(x^U)\big) \geq \lambda_p \tag{9.100}$$

if and only if

$$D\big(g(x^L)\big)c\big(g(x^L), g(x^U)\big) - \frac{G(x^U)}{D\big(g(x^U)\big)}d\big(g(x^L), g(x^U)\big) \geq \lambda_p. \tag{9.101}$$

From (9.99) it follows that $G(x^U) = G(x^L) + \lambda_p\{g(x^L) - g(x^U)\}$. Since $V(x) = \frac{G(x)}{D\big(g(x)\big)} + C\big(g(x)\big)$, we have

$$\begin{aligned}
V(x^U) - V(x^L) &= \tfrac{G(x^U)}{D\big(g(x^U)\big)} + C\big(g(x^U)\big) - \tfrac{G(x^L)}{D\big(g(x^L)\big)} - C\big(g(x^L)\big) \\
&= \tfrac{1}{D\big(g(x^U)\big)}\Big\{G(x^U) + D\big(g(x^U)\big)C\big(g(x^U)\big) - \tfrac{G(x^L)D\big(g(x^U)\big)}{D\big(g(x^L)\big)} - D\big(g(x^U)\big)C\big(g(x^L)\big)\Big\} \\
&= \tfrac{g(x^U)-g(x^L)}{D\big(g(x^U)\big)}\Big\{\tfrac{G(x^U)}{g(x^U)-g(x^L)} + D\big(g(x^U)\big)\cdot\tfrac{C(g(x^U))-C(g(x^L))}{g(x^U)-g(x^L)} - \tfrac{G(x^L)D(g(x^U))}{D(g(x^L))\{g(x^U)-g(x^L)\}}\Big\}.
\end{aligned}$$

Since by (9.99) $G\big(x^U\big) = G\big(x^L\big) - \lambda_p\{g\big(x^U\big) - g\big(x^L\big)\}$, we can write $\frac{G(x^U)}{g(x^U)-g(x^L)} = \frac{G(x^L)}{g(x^U)-g(x^L)} - \lambda_p$. Substituting this expression yields

$$\begin{aligned}
V(x^U) - V(x^L) &= \tfrac{g(x^U)-g(x^L)}{D\big(g(x^U)\big)}\Big\{\tfrac{G(x^L)}{g(x^U)-g(x^L)} - \lambda_p + D\big(g(x^U)\big)c\big(g(x^L), g(x^U)\big) - \tfrac{G(x^L)D(g(x^U))}{D(g(x^L))\{g(x^U)-g(x^L)\}}\Big\} \\
&= \tfrac{g(x^U)-g(x^L)}{D\big(g(x^U)\big)}\Big\{D\big(g(x^U)\big)c\big(g(x^L), g(x^U)\big) - \tfrac{G(x^L)}{D(g(x^L))}d\big(g(x^L), g(x^U)\big) - \lambda_p\Big\}.
\end{aligned}$$

Similarly, we can write

$$\begin{aligned}
V(x^U) - V(x^L) &= \tfrac{G(x^U)}{D\big(g(x^U)\big)} + C\big(g(x^U)\big) - \tfrac{G(x^L)}{D\big(g(x^L)\big)} - C\big(g(x^L)\big) \\
&= \tfrac{1}{D\big(g(x^L)\big)}\Big\{\tfrac{G(x^U)D\big(g(x^L)\big)}{D\big(g(x^U)\big)} + D\big(g(x^L)\big)C\big(g(x^U)\big) - G(x^L) - D\big(g(x^L)\big)C\big(g(x^L)\big)\Big\} \\
&= \tfrac{g(x^U)-g(x^L)}{D\big(g(x^L)\big)}\Big\{\tfrac{G(x^U)D(g(x^L))}{D(g(x^U))\{g(x^U)-g(x^L)\}} + D\big(g(x^L)\big)\cdot\tfrac{C(g(x^U))-C(g(x^L))}{g(x^U)-g(x^L)} - \tfrac{G(x^L)}{g(x^U)-g(x^L)}\Big\}.
\end{aligned}$$

Substituting, again by (9.99), $G\big(x^L\big) = G\big(x^U\big) + \lambda_p\{g\big(x^U\big) - g\big(x^L\big)\}$, gives

$$\begin{aligned}
V(x^U) - V(x^L) &= \tfrac{g(x^U)-g(x^L)}{D\big(g(x^L)\big)}\Big\{\tfrac{G(x^U)D(g(x^L))}{D(g(x^U))\{g(x^U)-g(x^L)\}} + D\big(g(x^L)\big)c\big(g(x^L), g(x^U)\big) - \tfrac{G(x^U)}{g(x^U)-g(x^L)} - \lambda_p\Big\} \\
&= \tfrac{g(x^U)-g(x^L)}{D\big(g(x^L)\big)}\Big\{D\big(g(x^L)\big)c\big(g(x^L), g(x^U)\big) - \tfrac{G(x^U)}{D(g(x^U))}d\big(g(x^L), g(x^U)\big) - \lambda_p\Big\}.
\end{aligned}$$

Hence, (9.100) and (9.101) are equivalent if and only if $D\big(g(x^U)\big)$ and $D\big(g(x^L)\big)$ have the same sign. By assumption (A2) this is true.

Next, we establish that $V_{opt} = max\big\{V\big(x^L\big), V\big(x^U\big)\big\}$ and $x_{opt} = \left\{ \begin{array}{l} x^L \text{ if } V\big(x^L\big) = V_{opt}; \\ x^U \text{ if } V\big(x^U\big) = V_{opt}. \end{array} \right.$.

We distinguish between two cases:

(1) $D\big(g(x^L)\big)c\big(g(x^L), g(x^U)\big) - \frac{G(x^U)}{D(g(x^U))}d\big(g(x^L), g(x^U)\big) \geq \lambda_p$;

(2) $D\big(g(x^L)\big)c\big(g(x^L), g(x^U)\big) - \frac{G(x^U)}{D(g(x^U))}d\big(g(x^L), g(x^U)\big) < \lambda_p$.

Case (1):

We can write, using (9.99),

$$\begin{array}{rcl} 0 \ \leq \ V(x_{opt}) - V(x^U) & = & \frac{G(x_{opt})}{D\big(g(x_{opt})\big)} + C\big(g(x_{opt})\big) - \frac{G(x^U)}{D\big(g(x^U)\big)} - C\big(g(x^U)\big) \\ & \leq & C\big(g(x_{opt})\big) - C\big(g(x^U)\big) + \frac{G(x^U)+\lambda_p\{(g(x^U)-g(x_{opt})\}}{D\big(g(x_{opt})\big)} - \frac{G(x^U)}{D\big(g(x^U)\big)} \\ & = & \frac{g(x_{opt})-g(x^U)}{D\big(g(x_{opt})\big)}\Big\{D\big(g(x_{opt})\big)c\big(g(x_{opt}), g(x^U)\big) - \frac{G(x^U)}{D\big(g(x^U)\big)}d\big(g(x_{opt}), g(x^U)\big) - \lambda_p\Big\}. \end{array}$$

Case (1a): D is a constant, i.e. $d(\cdot, \cdot) \equiv 0$, and by Case (1), $c\big(g(x^L), g(x^U)\big) \geq \frac{\lambda_p}{D}$.

The above inequality becomes: $0 \leq V(x_{opt}) - V(x^U) \leq \{g(x_{opt}) - g(x^U)\}\big\{c\big(g(x_{opt}), g(x^U)\big) - \frac{\lambda_p}{D}\big\}$.

The convexity of C implies $c(y, z) \leq c(x, z)$ for all x, y, z with $x \leq y \leq z$. Since we consider Case d, we have $g\big(x^L\big) \leq g(x_{opt}) \leq g\big(x^U\big)$ and therefore, $c\big(g(x_{opt}), g(x^U)\big) \leq c\big(g(x^L), g(x^U)\big)$.

Consequently

$$0 \leq V(x_{opt}) - V(x^U) \leq \{g(x_{opt})) - g(x^U)\}\big\{c\big(g(x^L), g(x^U)\big) - \frac{\lambda_p}{D}\big\} \leq 0,$$

implying $V(x_{opt}) = V(x^U)$ and $x_{opt} = x^U$.

Case (1b): D is not a constant and $D\big(g(x^L)\big)c\big(g(x^L), g(x^U)\big) - \frac{G(x^U)}{D(g(x^U))}d\big(g(x^L), g(x^U)\big) \geq \lambda_p$.

We have seen that

$$0 \leq V(x_{opt}) - V(x^U) \leq \frac{g(x_{opt})-g(x^U)}{D\big(g(x_{opt})\big)}\Big\{D\big(g(x_{opt})\big)c\big(g(x_{opt}), g(x^U)\big) - \frac{G(x^U)}{D\big(g(x^U)\big)}d\big(g(x_{opt}), g(x^U)\big) - \lambda_p\Big\}.$$

Because C is convex and D is nondecreasing and convex:

$$c\big(g(x_{opt}), g(x^U)\big) \geq c\big(g(x^L), g(x^U)\big);\ D\big(g(x_{opt})\big) \geq D\big(g(x^L);\ d\big(g(x_{opt}), g(x^U)\big) \geq d\big(g(x^L), g(x^U)\big).$$

Since $G(x^U) \leq 0$, we obtain

$$0 \leq V(x_{opt}) - V(x^U) \leq \frac{g(x_{opt})-g(x^U)}{D\big(g(x_{opt})\big)}\Big\{D\big(g(x^L)\big)c\big(g(x^L), g(x^U)\big) - \frac{G(x^U)}{D\big(g(x^U)\big)}d\big(g(x^L), g(x^U)\big) - \lambda_p\Big\}.$$

On the other hand, $g(x_{opt}) \leq g(x^U)$ and $D\big(g(x^L)\big)c\big(g(x^L), g(x^U)\big) - \frac{G(x^U)}{D(g(x^U))}d\big(g(x^L), g(x^U)\big) \geq \lambda_p$.

So,

$$\frac{g(x_{opt})-g(x^U)}{D\big(g(x_{opt})\big)}\Big\{D\big(g(x^L)\big)c\big(g(x^L), g(x^U)\big) - \frac{G(x^U)}{D\big(g(x^U)\big)}d\big(g(x^L), g(x^U)\big) - \lambda_p\Big\} \leq 0.$$

Hence, $V(x_{opt}) = V(x^U)$ and $x_{opt} = x^U$.

Case (2):

Similarly as in Case (1) we can write, using (9.99),

$$\begin{array}{rcl} 0 \ \leq \ V(x_{opt}) - V(x^L) & = & \frac{G(x_{opt})}{D\big(g(x_{opt})\big)} + C\big(g(x_{opt})\big) - \frac{G(x^L)}{D\big(g(x^L)\big)} - C\big(g(x^L)\big) \\ & \leq & C\big(g(x_{opt})\big) - C\big(g(x^L)\big) + \frac{G(x^L)+\lambda_p\{(g(x^L)-g(x_{opt})\}}{D\big(g(x_{opt})\big)} - \frac{G(x^L)}{D\big(g(x^L\big)} \\ & = & \frac{g(x_{opt})-g(x^L)}{D\big(g(x_{opt})\big)}\Big\{D\big(g(x_{opt})\big)c\big(g(x_L), g(x_{opt})\big) - \frac{G(x^L)}{D\big(g(x^L)\big)}d\big(g(x_L), g(x_{opt})\big) - \lambda_p\Big\}. \end{array}$$

Case (2a): D is a constant, i.e. $d(\cdot,\cdot) \equiv 0$, and by Case (2), $c\big(g(x^L), g(x^U)\big) < \frac{\lambda_p}{D}$.
The above inequality becomes: $0 \leq V(x_{opt}) - V(x^L) \leq \{g(x_{opt}) - g(x^L)\}\big\{c\big(g(x_L), g(x_{opt})\big) - \frac{\lambda_p}{D}\big\}$.
The convexity of C and $g\big(x^L\big) \leq g(x_{opt}) \leq g\big(x^U\big)$ imply, $c\big(g(x^L), g(x_{opt})\big) \leq c\big(g(x^L), g(x^U)\big)$.
Consequently

$$0 \leq V(x_{opt}) - V(x^L) \leq \{g(x_{opt})) - g(x^L)\}\big\{c\big(g(x^L), g(x^U)\big) - \tfrac{\lambda_p}{D}\big\} \leq 0,$$

implying $V(x_{opt}) = V(x^L)$ and $x_{opt} = x^L$.
Case (2b): D is not a constant and $D\big(g(x^L)\big)c\big(g(x^L), g(x^U)\big) - \frac{G(x^U)}{D(g(x^U)}d\big(g(x^L), g(x^U)\big) < \lambda_p$.
We have seen that

$$0 \leq V(x_{opt}) - V(x^U) \leq \frac{g(x_{opt}) - g(x^L)}{D\big(g(x_{opt}\big)}\Big\{D\big(g(x_{opt})\big)c\big(g(x^L), g(x_{opt})\big) - \frac{G(x^L)}{D\big(g(x^L)\big)}d\big(g(x^L), g(x_{opt})\big) - \lambda_p\Big\}.$$

Since C is convex and D is nondecreasing and convex:

$$c\big(g(x^L), g(x_{opt})\big) \leq c\big(g(x^L), g(x^U)\big);\ D\big(g(x_{opt})\big) \leq D\big(g(x^U);\ c\big(g(x^L), g(x_{opt})\big) \leq c\big(g(x^L), g(x^U)\big).$$

Since $G(x^L) \leq 0$, we obtain

$$0 \leq V(x_{opt}) - V(x^U) \leq \frac{g(x_{opt}) - g(x^L)}{D\big(g(x_{opt}\big)}\Big\{D\big(g(x^U)\big)c\big(g(x^L), g(x^U)\big) - \frac{G(x^L)}{D\big(g(x^L)\big)}d\big(g(x^L), g(x^U)\big) - \lambda_p\Big\}.$$

On the other hand, $g(x_{opt}) \geq g(x^L)$ and $D\big(g(x^U)\big)c\big(g(x^L), g(x^U)\big) - \frac{G(x^L)}{D(g(x^L))}d\big(g(x^L), g(x^U)\big) < \lambda_p$, the last inequility by the equivalence of (9.100) and (9.101), So,

$$\frac{g(x_{opt}) - g(x^L)}{D\big(g(x_{opt}\big)}\Big\{D\big(g(x^U)\big)c\big(g(x^L), g(x^U)\big) - \frac{G(x^L)}{D\big(g(x^L)\big)}d\big(g(x^L), g(x^U)\big) - \lambda_p\Big\} \leq 0.$$

Hence, $V(x_{opt}) = V(x^L)$ and $x_{opt} = x^L$.

The proofs for the cases (a), (b) and (c) can be obtained in a similar way. Instead of x^L and x^U we take: in case (a): x^p and x^{p+1}; in case (b): x^L and x^{p+1}; in case (c): x^p and x^U. □

9.5.4 Determination of an optimal policy

Theorem 9.51 provides an optimal solution for program (9.94), but it does not provide a procedure to construct an optimal policy for the mean-variance problem.

Theorem 9.52
Let (x, y) be an extreme optimal solution for program (9.95) for all λ in an open interval I. Then, there exists a policy $f^\infty \in C(D)$ whose limiting state-action frequencies vector $x(f)$ satisfies $\sum_{(i,a)} B_i(a)x_{ia}(f) = \sum_{(i,a)} B_i(a)x_i(a)$, $\sum_{(i,a)} R_i(a)x_{ia}(f) = \sum_{(i,a)} R_i(a)x_i(a)$, $V\big(x(f)\big) = V(x)$.

Proof
Let f^∞ be a policy satisfying $x_i\big(f(i)\big) > 0,\ i \in S_x;\ y_i\big(f(i)\big) > 0,\ i \in S_y$ and $f(i)$ arbitrarily chosen for $i \notin S_x \cup S_y$. From Theorem 9.41 it follows that f^∞ is β-optimal for all $\lambda \in I$. Define the policy f^∞ by (5.35). Then, for $r_i(a) := B_i(a) + \lambda R_i(a),\ (i,a) \in S \times A$, we have

$$\textstyle\sum_{(i,a)} r_i(a)x_{ia}(f) = \sum_i r_i(f) \cdot \sum_j \beta_j\{P^*(f)\}_{ji} \sum_k \beta_j \cdot \sum_i \{P^*(f)\}_{ji} r_i(f) = \phi(\beta, f^\infty),$$

i.e. $\big(x(f), y(f)\big)$ is also an optimal solution of (9.95). Therefore,

$$\textstyle\sum_{(i,a)}\{B_i(a) + \lambda R_i(a)\}x_i(a) = \sum_{(i,a)}\{B_i(a) + \lambda R_i(a)\}x_{ia}(f) \text{ for all } \lambda \in I.$$

Hence, $\sum_{(i,a)}\{B_i(a)x_i(a) = \sum_{(i,a)} B_i(a)x_{ia}(f)$ and $\sum_{(i,a)} R_i(a)\}x_i(a) = \sum_{(i,a)} R_i(a)\}x_{ia}(f)$. Since $g(x) = R_i(a)\}x_i(a)$, $G(x) = \sum_{(i,a)}\{B_i(a)x_i(a)$ and $V(x) = \frac{G(x)}{D\big(g(x)\big)} + C\big(g(x)\big)$, we have $g(x) = g\big(x(f)\big)$, $G(x) = G\big(x(f)\big)$, implying $V(x) = V\big(x(f)\big)$. □

Theorem 9.53

If program (9.94) is feasible, then either $x_{opt} = x^n$ for some $j+1 \leq n \leq k$ and there exists an optimal deterministic policy, or $x_{opt} = x^L$ (or x^U) and an initial randomization of two deterministic policies is optimal for the mean-variance tradeoffs problem.

Proof

Suppose that $x_{opt} = x^n$ for some $j+1 \leq n \leq k$. Since x^n is optimal for all $\lambda \in [\lambda_{n-1}, \lambda_n]$ and $\lambda_{n-1} < \lambda_n$, by Theorem 9.52, there exists a policy f^∞ whose limiting state-action frequencies vector $x(f)$ satisfies $\sum_{(i,a)} B_i(a)x_{ia}(f) = \sum_{(i,a)} B_i(a)x_i^n(a)$, $\sum_{(i,a)} R_i(a)x_{ia}(f) = \sum_{(i,a)} R_i(a)x_i^n(a)$ and $V\big(x(f)\big) = V(x^n)$. Because $x(f)$ also satisfies the constraint $L \leq \sum_{(i,a)} R_i(a)x_i(a) \leq U$, f^∞ is an optimal policy.

Next, suppose that $x_{opt} = x^L$, where $x^L = \nu x^j + (1-\nu)x^{j+1}$ and $\sum_{(i,a)} R_i(a)x_i^L(a) = L$ (the case $x_{opt} = x^U$ can be shown similarly). By Theorem 9.52, corresponding to x^j and x^{j+1}, there are policies $f_j^\infty, f_{j+1}^\infty \in C(D)$ whose limiting state-action frequencies vectors $x(f_j)$ and $x(f_{j+1})$ satisfy $\sum_{(i,a)} B_i(a)x_{ia}(f_j) = \sum_{(i,a)} B_i(a)x_i^j(a)$, $\sum_{(i,a)} R_i(a)x_{ia}(f_j) = \sum_{(i,a)} R_i(a)x_i^j(a)$ and $\sum_{(i,a)} B_i(a)x_{ia}(f_{j+1}) = \sum_{(i,a)} B_i(a)x_i^{j+1}(a)$, $\sum_{(i,a)} R_i(a)x_{ia}(f_{j+1}) = \sum_{(i,a)} R_i(a)x_i^{j+1}(a)$, respectively. Then, setting $x^* = \nu x(f_j) + (1-\nu)x(f_{j+1})$, we obtain

$$\begin{aligned}\textstyle\sum_{(i,a)} R_i(a)x_i^*(a) &= \textstyle\sum_{(i,a)} R_i(a)\{\nu x_i^j(a) + (1-\nu)x_i^{j+1}(a)\} \\ &= \textstyle\sum_{(i,a)} R_i(a)\{\nu x_i^j(a) + (1-\nu)x_i^{j+1}(a)\} \ = \ \sum_{(i,a)} R_i(a)x_i^L(a) = L\end{aligned}$$

and

$$\begin{aligned}\textstyle\sum_{(i,a)} B_i(a)x_i^*(a) &= \textstyle\sum_{(i,a)} B_i(a)\{\nu x_i^j(a) + (1-\nu)x_i^{j+1}(a)\} \\ &= \textstyle\sum_{(i,a)} B_i(a)\{\nu x_i^j(a) + (1-\nu)x_i^{j+1}(a)\} = \sum_{(i,a)} B_i(a)x_i^L(a).\end{aligned}$$

Hence, $V(x_{opt}) = V(x^L) = V(x^*)$. From Theorem 1.1 it follows that the policy R_* which initially randomizes between f_j^∞ and f_{j+1}^∞ with coefficients ν and $1-\nu$ yields as state-action frequencies vector $x(R_*) = \nu x(f_j) + (1-\nu)x(f_{j+1}) = x^*$. Therefore, R_* is an optimal policy for the mean-variance tradeoffs problem. □

Corollary 9.12

For an unconstrained problem, i.e. without the constraint $L \leq \sum_{(i,a)} R_i(a)x_i(a) \leq U$, there exists a deterministic optimal policy. Hence, the variance-penalized version of the mean-variance tradeoff problem has a deterministic policy.

Example 9.19 (continued)

Consider the model of Example 9.19 for the variance-penalized version of with penalty $\gamma = 1$, i.e. $max_R\,\{\phi(\beta, R) - V(\beta, R)\}$. The corresponding quadratic program is:

$max\{-2x_{21} - 56x_{31} - 12x_{32} + (2x_{21} + 8x_{31} + 4x_{32})^2\}$

subject to

$$\begin{array}{rrrrrrrrrrrrrrrrrl}
 & x_{11} & & & - & x_{31} & & & & & & & & & & & = & 0 \\
 & & & x_{21} & & & - & x_{32} & & & & & & & & & = & 0 \\
- & x_{11} & - & x_{21} & + & x_{31} & + & x_{32} & & & & & & & & & = & 0 \\
 & x_{11} & & & & & & & + & y_{11} & & & - & y_{31} & & & = & \frac{1}{4} \\
 & & & x_{21} & & & & & & & + & y_{21} & & & - & y_{32} & = & \frac{1}{4} \\
 & & & & & x_{31} & + & x_{32} & - & y_{11} & - & y_{21} & + & y_{31} & + & y_{32} & = & \frac{1}{2} \\
 & \multicolumn{15}{l}{x_{11},\ x_{21},\ x_{31},\ x_{32},\ y_{11},\ y_{21},\ y_{31},\ y_{32}} & \geq & 0
\end{array}$$

The parametric linear program is:

$max\{-2x_{21} - 56x_{31} - 12x_{32} + \lambda \cdot (2x_{21} + 8x_{31} + 4x_{32})\}$

subject to

$$\begin{array}{rrrrrrrrrrrrrrrrrl}
 & x_{11} & & & - & x_{31} & & & & & & & & & & & = & 0 \\
 & & & x_{21} & & & - & x_{32} & & & & & & & & & = & 0 \\
- & x_{11} & - & x_{21} & + & x_{31} & + & x_{32} & & & & & & & & & = & 0 \\
 & x_{11} & & & & & & & + & y_{11} & & & - & y_{31} & & & = & \frac{1}{4} \\
 & & & x_{21} & & & & & & & + & y_{21} & & & - & y_{32} & = & \frac{1}{4} \\
 & & & & & x_{31} & + & x_{32} & - & y_{11} & - & y_{21} & + & y_{31} & + & y_{32} & = & \frac{1}{2} \\
 & \multicolumn{15}{l}{x_{11},\ x_{21},\ x_{31},\ x_{32},\ y_{11},\ y_{21},\ y_{31},\ y_{32}} & \geq & 0
\end{array}$$

The extreme optimal solutions are:

$$\begin{array}{lllllllll}
\lambda \geq 21: & x^1_{11} = \frac{1}{2} & x^1_{21} = 0 & x^1_{31} = \frac{1}{2} & x^1_{32} = 0 & y^1_{11} = 0 & y^1_{21} = \frac{1}{4} & y^1_{31} = \frac{1}{4} & y^1_{32} = 0. \\
\lambda \leq 21: & x^2_{11} = 0 & x^2_{21} = \frac{1}{2} & x^2_{31} = 0 & x^2_{32} = \frac{1}{2} & y^2_{11} = \frac{1}{4} & y^2_{21} = 0 & y^2_{31} = 0 & y^2_{32} = \frac{1}{4}. \\
\lambda = 21: & x^3_{11} = \frac{1}{4} & x^3_{21} = \frac{1}{4} & x^3_{31} = \frac{1}{4} & x^3_{32} = \frac{1}{4} & y^3_{11} = 0 & y^3_{21} = 0 & y^3_{31} = 0 & y^3_{32} = 0.
\end{array}$$

The first two extreme optimal solutions, (x^1, y^1) and (x^2, y^2), are optimal in an open interval. So, according to Theorem 9.52, they correspond to deterministic policies, namely f_1^∞ and f_2^∞, respectively. Notice that the last extreme optimal solution (x^3, y^3) is not optimal in an open interval; the corresponding policy is the stationary policy π^∞ with $\pi_{31} = \pi_{32} = \frac{1}{2}$.

In order to determine the optimal policy for the variance-penalized problem, we evaluate the nonlinear objective function $-2x_{21} - 56x_{31} - 12x_{32} + (2x_{21} + 8x_{31} + 4x_{32})^2$ for x_1 and x^2. For x^1 we obtain the value $-28 + 42 = -12$; for x^2, the value is $-7 + 32 = 2$. Hence, f_2^∞ is the optimal deterministic policy for the variance-penalized problem. Notice that for x^3 the value is $-\frac{35}{2} + (\frac{7}{2})^2 = -\frac{21}{4}$.

<u>Remark</u>

If we have a multichain MDP, then it is possible that the optimal solution x_{opt} of the nonlinear program (9.94) does not correspond to a deterministic or stationary policy. In that case we can find, using Algorithm 9.5, a convergent Markov policy R_{opt} with $x(R_{opt}) = x_{opt}$. For unichain MDPs, as always, the analysis can be simplified. This is the subject of the next section.

9.5.5 The unichain case

In the unichain case the state-action frequencies are independent of the initial distribution. Furthermore, by Theorem 9.24, $L = L(M) = L(C) = L(S) = \overline{L(D)} = Q = Q_0$. Hence, in this case the parametric linear program (9.95) can be simplified to

$$maximize \left\{ \sum_{i,a} \{B_i(a) + \lambda R_i(a)\} x_i(a) \;\middle|\; \begin{array}{rcl} \sum_{i,a} \{\delta_{ij} - p_{ij}(a)\} x_i(a) & = & 0,\ j \in S \\ \sum_{i,a} x_i(a) & = & 1 \\ x_i(a) & \geq & 0,\ (i,a) \in S \times A \end{array} \right\}. \tag{9.102}$$

After the construction of an optimal solution $\overline{x}$ of the nonlinear problem (9.94) one can construct an optimal stationary policy $\overline{\pi}^\infty$ by

$$\overline{\pi}_{ia} := \begin{cases} \frac{\overline{x}_i(a)}{\overline{x}_i} & a \in A(i),\ i \in S_{\overline{x}} \\ \text{arbitrary} & \text{otherwise} \end{cases} \quad \text{where } \overline{x}_i := \sum_a \overline{x}_i(a) \text{ and } S_{\overline{x}} := \{i \mid \overline{x}_i > 0\}. \tag{9.103}$$

<u>Remark</u>
When $L < \sum_{j,a} R_j(a)\overline{x}_j(a) < U$, then the extreme optimal solution $\overline{x}$ is an extreme point of Q. Hence, the corresponding policy $\overline{\pi}^\infty$ is a deterministic policy. If $\sum_{j,a} R_j(a)\overline{x}_j(a) = L$ or $\sum_{j,a} R_j(a)\overline{x}_j(a) = U$, then the corresponding policy $\overline{\pi}^\infty$ is deterministic in all but one state, and in that state it randomizes between at most two actions.

Example 9.19 (continued)
Consider the model of Example 9.19 for the variance-penalized version with penalty $\gamma = 1$. Since this is a unichain model the parametric linear programming problem (9.102) becomes:

The parametric linear program is:

$max\{-2x_{21} - 56x_{31} - 12x_{32} + \lambda \cdot (2x_{21} + 8x_{31} + 4x_{32})\}$

subject to

$$\begin{array}{rcrcrcrcl} & x_{11} & & & - & x_{31} & & & = & 0 \\ & & & x_{21} & & & - & x_{32} & = & 0 \\ - & x_{11} & - & x_{21} & + & x_{31} & + & x_{32} & = & 0 \\ & x_{11} & + & x_{21} & + & x_{31} & + & x_{32} & = & 1 \\ & \multicolumn{7}{l}{x_{11},\ x_{21},\ x_{31},\ x_{32}} & \geq & 0 \end{array}$$

The extreme optimal solutions are:

$\lambda \geq 21: \quad x^1_{11} = \frac{1}{2};\quad x^1_{21} = 0;\quad x^1_{31} = \frac{1}{2};\quad x^1_{32} = 0.$

$\lambda \leq 21: \quad x^2_{11} = 0;\quad x^2_{21} = \frac{1}{2};\quad x^2_{31} = 0;\quad x^2_{32} = \frac{1}{2}.$

In comparison with the previous parametric linear programming problem with also the y-variables, in the present program without the y-variables is the solution x^3 is not an extreme solution because $x^3 = \frac{1}{2}(x^1 + x^2)$. The two extreme optimal solutions. In order to determine the optimal policy for the variance-penalized problem, we evaluate the nonlinear objective function for x^1 and x^2. We have already observed That for x^1 and x^2 we obtain the values -12 and 2, respectively. Hence, f_2^∞ is the optimal deterministic policy for the variance-penalized problem.

Next, we consider the problem of minimizing the variance subject to the constraint $\phi(\beta,R) \geq \frac{7}{2}$:

$min\{4x_{21} + 64x_{31} + 16x_{32} - (2x_{21} + 8x_{31} + 4x_{32})^2 \mid x \in Q_0;\ 2x_{21} + 8x_{31} + 4x_{32} \geq \frac{7}{2}\}$.

The corresponding parametric linear program is:

$max\{-4x_{21} - 64x_{31} - 16x_{32} + \lambda \cdot (2x_{21} + 8x_{31} + 4x_{32})\}$

subject to

$$\begin{array}{rrrrrrrrrrrcl}
 & x_{11} & & & - & x_{31} & & & & & = & 0 \\
 & & & x_{21} & & & - & x_{32} & & & = & 0 \\
- & x_{11} & - & x_{21} & + & x_{31} & + & x_{32} & & & = & 0 \\
 & x_{11} & + & x_{21} & + & x_{31} & + & x_{32} & & & = & 1 \\
 & & & 2x_{21} & + & 8x_{31} & + & 4x_{32} & - & y_5 & = & \frac{7}{2} \\
 & & & \multicolumn{7}{l}{x_{11},\ x_{21},\ x_{31},\ x_{32},\ y_5} & \geq & 0
\end{array}$$

The extreme optimal solutions are:

$\lambda \geq 22:\quad x^1_{11} = \frac{1}{2};\quad x^1_{21} = 0;\quad x^1_{31} = \frac{1}{2};\quad x^1_{32} = 0;\quad y_5 = \frac{1}{2}.$

$\lambda \leq 22:\quad x^2_{11} = \frac{1}{4};\quad x^2_{21} = \frac{1}{4};\quad x^2_{31} = \frac{1}{4};\quad x^2_{32} = \frac{1}{4};\quad y_5 = 0.$

In order to determine the optimal policy for this problem, we evaluate the nonlinear objective function $4x_{21} + 64x_{31} + 16x_{32} - (2x_{21} + 8x_{31} + 4x_{32})^2$ for x^1 and x^2. For x^1 we obtain the value 16 and for x^2 the value $\frac{35}{4}$. Hence, x^2 is the optimal solution and π^∞ with $\pi_{11} = 1, \pi\ _{21} = 1,\ \pi_{31} = \pi_{32} = \frac{1}{2}$ is an optimal stationary policy.

Finally, we consider the mean-standard deviation ratio problem under the constraint $\phi(\beta,R) \geq \frac{7}{2}$:

$$max\Big\{\frac{-4x_{21}-64x_{31}-16x_{32}}{(2x_{21}+8x_{31}+4x_{32})^2}\ \Big|\ x \in Q_0;\ 2x_{21} + 8x_{31} + 4x_{32} \geq \frac{7}{2}\Big\}.$$

The parametric program for this problem is the same as for minimum variance problem. Therefore, we only have to evaluate for x^1 and x^2 the nonlinear function $\frac{-4x_{21}-64x_{31}-16x_{32}}{(2x_{21}+8x_{31}+4x_{32})^2}$. For x^1 we obtain the value -2 and for x^2 the value $\frac{84}{49}$. Hence, we have the same optimal solution x^2 as in het minimum variance problem and also the same optimal policy π^∞ with $\pi_{11} = 1, \pi\ _{21} = 1$, $\pi_{31} = \pi_{32} = \frac{1}{2}$.

9.5.6 Finite horizon variance-penalized MDPs

We consider a finite horizon MDP with T periods and with nonstationary transition probabilities $p^t_{ij}(a),\ 1 \leq t \leq T,\ i,j \in S$ and $a \in A(i)$. We assume that there are only terminal rewards r_i when state i is reached at the end of the horizon. We also assume a fixed initial distribution β at $t = 1$. For a general policy R, we let $x_i(\beta,R)$ denote the probability of being in state i at the end of the horizon, i.e. when $t = T+1$. Then, the variance-penalized reward $v^\gamma(\beta,R)$, where $\gamma > 0$ is a fixed penalty, is defined by

$$v^\gamma(\beta,R) := \mathbb{E}_{\beta,R}\{r_{T+1}\} - \gamma \cdot Var_{\beta,R}\{r_{T+1}\}. \tag{9.104}$$

Therefore, $v^\gamma(\beta,R) = \sum_i r_i x_i(\beta,R) - \gamma \cdot \Big\{\sum_i r_i^2 x_i(\beta,R) - \{\sum_i r_i x_i(\beta,R)\}^2\Big\}$.

The variance-penalized problem is $max_R\, v^\gamma(\beta, R)$. The corresponding nonlinear program is:

$$max\Big\{ \sum_i r_i x_{i,T+1} - \gamma \cdot \sum_i r_i^2 x_{i,T+1} + \gamma \cdot \{\sum_i r_i x_{i,T+1}\}^2 \Big\}.$$

subject to the constraints

$$\begin{array}{rcrcl} \sum_a x_{j,1}(a) & & & = & \beta_j,\ j \in S \\ \sum_a x_{j,t}(a) & - & \sum_{(i,a)} p_{ij}^{t-1} x_{j,t-1}(a) & = & 0,\ j \in S,\ 2 \le t \le T \\ x_{j,T+1} & - & \sum_{(i,a)} p_{ij}^{T} x_{j,t-1}(a) & = & 0,\ j \in S \\ & & x_{i,t}(a) & \ge & 0,\ (i,a) \in S \times A,\ 1 \le t \le T \\ & & x_{i,T+1} & \ge & 0,\ j \in S \end{array}$$

Notice that the objective function is a convex function (the Hessian $H = 2\gamma \cdot rr^T$, so we have $x^T H x = 2\gamma \cdot (x^T r)^2 \ge 0$ for all $x \in \mathbb{R}^N$). It is well known that the maximum of a convex function over linear constraints is achieved at some vertex of the set of the linear constraints. This convex program can be solved by the method described in Section 9.5.3. In the way described in Section 9.5.4 a Markov deterministic policy can be determined.

We will also present another approach. Define the vector sets $K, K(M), K(MD)$ and P, with components $i \in S$, by

$$\begin{array}{rcl} K & := & \{x(\beta,R) \mid R \text{ is an arbitrary policy}\}; \\ K(M) & := & \{x(\beta,R) \mid R \text{ is a Markov policy}\}; \\ K(MD) & := & \{x(\beta,R) \mid R \text{ is a deterministic Markov policy}\}; \\ P & := & \Big\{ x \;\Big|\; \begin{array}{l} x_i = x_{i,T+1},\ i \in S, \text{ where } x_{i,T+1} \text{ is (part of) a feasible solution} \\ \text{of the above convex optimization problem} \end{array} \Big\}. \end{array}$$

Theorem 9.54
$K = K(M) = \overline{K(MD)} = P$, where $\overline{K(MD)}$ is the closed convex hull of the finite set of vectors $K(MD)$.

Proof
The proof is similar to the proof of Theorem 9.12 □

Consider the convex function $h(x) := \sum_i r_i x_i - \gamma \cdot \sum_i r_i^2 x_i + \gamma \cdot \{\sum_i r_i x_i\}^2$. The objective is to find a distribution x^* which maximizes $h(x)$ over K and to find the corresponding deterministic Markov policy R_*. The approach we will take is based on a geometrical characterization of K. Let $vert(K)$ denote the finite set of vertices of K. Each point in $vert(K)$ is a point in $K(MD)$ corresponding to the final distribution $x_{i,T+1},\ i \in S$, for some deterministic Markov policy.
Given a direction $d \in \mathbb{R}^N$, we say a policy R is a *best response* in the direction of d if:
(1) $x(\beta, R) \in vert(K)$;
(2) $y^T d \le \{x(\beta,R)\}^T d$ for all $y \in K$.
Whenever there is no possibility of confusion, we often refer $x(\beta, R)$ itself as a best response. A geometric interpretation of a best response can given as follows. Given some $d \in \mathbb{R}^N$ and $z \in \mathbb{R}$, let the half-spaces $H^-(d,z)$ and $H^+(d,z)$ be defined by $H^-(d,z) := \{x \in \mathbb{R}^N \mid x^T d \le z\}$,

$H^+(d,z) := \{x \in \mathbb{R}^N \mid x^Td \geq z\}$. Furthermore, let the hyperplane $H(d,z)$ be defined by $H(d,z) := \{x \in \mathbb{R}^N \mid x^Td = z\}$. From the definition of best response, we have that R is a best response if and only if $x(\beta,R) \in vert(K)$ and $K \subseteq H^-\big(d, x(\beta,R)^Td\big)$, which is equivalent to $x(\beta,R) \in vert(K)$ and $x(\beta,R) \in H\big(d, x(\beta,R)^Td\big)$.

Let $Q = H\big(d, x(\beta,R)^Td\big) \cap K$. So, the set of best responses in the direction of d corresponds to the set $vert(Q)$. The situation where Q contains a single point corresponds to the case where there is a unique best response, which is a deterministic Markov policy whose final distribution maximizes x^Td over $x \in K$. Similarly, the situation that Q contains more than one point corresponds to the case where there is more than one best response, i.e. there is more than one deterministic Markov policy whose final distribution maximizes x^Td over $x \in K$. However, in this case we need to note that there may also be points in $Q \cup K(MD)$ which are not vertices. There points corresponds to deterministic Markov policies R for which the final distribution $x(\beta,R)$ maximizes x^Td over $x \in K$, but for which $x(\beta,R) \notin vert(K)$.

We now derive an algorithm which enables us to compute a best response in a given direction.

Algorithm 9.12 *Computation of a best response in a given direction*
Input: Instance of an MDP over a finite horizon T and a direction d.
Output: A best response R_* for the given direction d.

```
1. for all $i \in S$ do $v_i^{T+1} := d_i$.
2. for all $i \in S$ do
      for $t = T,\ T-1$ until 1 compute
         begin $v_i^t := max_a\{\sum_j p_{ij}(a)v_j^{t+1}\};\ A(i,t) := \{a \mid v_i^t = \sum_j p_{ij}(a)v_j^{t+1}\}$ end
3. if $|A(i,t)| = 1$ and $A(i,t) = \{a(i,t)\}$ for all $i \in S$ and all $1 \leq t \leq T$ then
   begin
      (a) $R_* := (f_1, f_2, \dots, f_T)$ with $f_t(i) = a(i,t)$ for all $i \in S$ and $1 \leq t \leq T$
      (b) compute the corresponding final distribution $x(\beta,R)$ by working forward using the
          known initial distribution $\beta$
   end
   else
   begin
      (a) determine the distinct deterministic Markov policies, say $f_1, f_2, \dots, f_m$, corresponding
          to the distinct choices of the elements of $A(i,t),\ i \in S,\ 1 \leq t \leq T$
      (b) determine $x^1, x^2, \dots, x^m$, the corresponding final distributions by working forward us-
          ing the known initial distribution $\beta$
```

(c) select some arbitrary strictly convex function g and determine $g(x^1), g(x^2), \ldots, g(x^m)$

(d) select $x^* \in argmax\{(g(x^1), g(x^2), \ldots, g(x^m)\}$ and let R_* be the corresponding deterministic Markov policy

end

Lemma 9.38
Algorithm 9.12 is correct.

Proof
We have to show that, given direction d, the algorithm identifies a deterministic Markov policy R_* such that $x(\beta, R_*)$ is a vertex of K and maximizes $y^T d$ over K. Define the function w on S by taking $w(i) := d_i,\ i \in S$. Then, $y^T d = \sum_i y_i w(i) = \mathbb{E}_y\{w(X_{T+1})\}$, where $\mathbb{E}_y$ denotes the expectation of X_{T+1}, given the distribution y. Therefore, the problem $max\{y^T d \mid y \in K\}$ corresponds to the problem of $max\, \mathbb{E}_{x(R)}\{w(X_{T+1})\}$ over all policies R.

Standard MDP theory now says that there is at least one deterministic Markov policy which maximizes $max\, \mathbb{E}_{x(R)}\{w(X_{T+1})\}$ over all policies R and that Algorithm 9.12 will identify all distinct deterministic optimal Markov policies.

The algorithm identifies m deterministic Markov policies, $f_1, f_2, \ldots, f_m$, and the corresponding final distributions $x_1, x_2, \ldots, x^m$. There may be final distributions which are not vertices of K. To verify that $x_* \in argmax\{g(x^1), g(x^2), \ldots, g(x^m)\}$ is a vertex of K, consider the closed, bounded, convex set $Q = H\big(d, x(\beta, R)^T d\big) \cap K$. Since g is a strictly convex function, it achieves its maximum over Q at a vertex of Q, and any point maximizing g over Q must be a vertex of Q. However, the set $\{x^1, x^2, \ldots, x^m\}$ contains all the points in $K(MD) \cap Q$, and in particular all the vertices of Q. Thus, x^* must be a vertex of Q and hence, R_* is indeed a best response. □

We will describe a vertex identification algorithm for finding an optimal final distribution x^* and a corresponding deterministic Markov policy R_*. The algorithm generates a sequence of polytopes $P_1, P_2, \ldots$ which is used to iteratively identify vertices of K. The algorithm eventually identifies all the vertices of K and hence finds x^*.

Before we describe the algorithm we present some definitions and properties of convex polytopes. A *convex polytope* may be defined as the convex hull of a finite set of points or as a bounded intersection of a finite set of half-spaces. Let P be an n-dimensional polytope. For a real n-vector d and a real number b, the linear inequality $y^T d \leq b$ is called valid for P if $y^T d \leq b$ holds for all $y \in P$. A subset F of a polyhedron P is called a *face* of P if it is represented as $F = P \cap \{y \mid y^T d = b\}$ for some *valid inequality* $d^T y \leq b$.

By this definition, both the empty set $\emptyset$ and the whole set P are faces. These two faces are called *improper faces* while the other faces are called *proper faces.* The faces of dimension $0, 1$ and $n-1$ are called the *vertices, edges and facets*, respectively.

For each face F, let *aff*(F) denote the intersection of all affine subspaces of $\mathbb{R}^n$ containing F. If F is a facet, then *aff*(F) corresponds to a hyperplane $H = \{y \mid y^T d = b\}$, where d is the outward normal with respect to P. If $F_1, F_2, \dots, F_m$ are the facets of P with corresponding hyperplanes $y^T d^i = b_i,\ 1 \le i \le m$, then $P = \{y \mid y^T d^i \le b_i,\ 1 \le i \le m\}$.

The algorithm works as follows. At each iteration $k = 1, 2, \dots$ let P_k be the $(N-1)$-dimensional polytope defined by the currently identified vertices, and let x^k be a vertex which maximizes the strictly convex function $g(x)$ over the vertices of P_k. We classify a facet F as a *non-active facet* of P_k if the algorithm has already identified F as a supporting hyperplane of K; otherwise, we classify an unchecked facet F as an *active facet* of P_k. Notice that, on checking, an active facet may turn out to be a supporting hyperplane of K.

The algorithm chooses an active facet F of P_k and then uses the best response method to (try to) identify a new vertex of K not in P_k which can be used to construct the next polytope. The polytopes $P_1 \subseteq P_2 \subseteq \cdots$ successively approximate K from within. Since K has only a finite number of vertices, the algorithm generates a finite sequence of points $x^1, x^2, \dots, x^M$ and a corresponding sequence of deterministic Markov policies $R_1, R_2, \dots, R_M$ such that we have $h(x^1) \le h(x^2) \le \cdots \le h(x^M)$. We set $x^* := x^M$ and $R_* := R_M$.

We formally summarize the basic algorithm below, and comment on the steps involved and the correctness of the algorithm.

Algorithm 9.13 *Computation of a variance-penalized optimal policy*

Input: Instance of an MDP over a finite horizon T and with only terminal rewards $r_i,\ i \in S$ and a penalty γ for the variance.

Output: A variance-penalized optimal policy R_*.

1. Set the real function h on $\mathbb{R}^N$ by $h(x) := \sum_i r_i x_i - \gamma \cdot \sum_i r_i^2 x_i + \gamma \cdot \{\sum_i r_i x_i\}^2$.

2. Select some arbitrary real strictly convex function g on $\mathbb{R}^N$.

3. *Initialization*

   (a) Generate an $(N-1)$-dimensional polytope P_1 with $V_1 := vert(P_1) \subseteq vert(K)$.

   (b) Set x^1 to be a vertex of P_1 which maximizes $g(x)$ over V_1 and set R_1 to be the corresponding deterministic Markov policy.

4. *Iteration*

   (a) Let V_k be the set of vertices of K identified after the k-th iteration of the algorithm.

   (b) Let P_k the convex hull of the vertices of V_k.

   (c) Let x^k be a vertex which maximizes $h(x)$ over V_k and let R_k be the corresponding deterministic Markov policy.

   (d) **if possible** choose an active facet F of P_k
   else go to step 7

```
5. Identification
   (a) Compute the best response $x$ in the direction $d$ of the outward normal of $aff(F)$ relative
       to $P_k$ and compute the corresponding deterministic Markov policy $R$.
   (b) if $x \in V_k$ then go to step 6(a)
       else go to step 6(b)
6. Updating
   (a) begin
         set $V_{k+1} := V_k$;  $P_{k+1} := P_k$;  $x^{k+1} := x^k$;  $R_{k+1} := R_k$;
         classify $F$ as non-active; return to step 4
       end
   (b) begin
         set $V_{k+1} := V_k \cup x$; let $P_{k+1}$ be the convex hull of the vertices of $V_{k+1}$;
         choose $x^{k+1} \in argmax\{h(x), h(x^k)\}$;
         set $R_{k+1}$ be the corresponding deterministic Markov policy;
         identify the faces of $P_{k+1}$, noting which are non-active and which are active;
         return to step 4
       end
7. Termination
   The deterministic Markov policy $R_k$ is a variance-penalized optimal policy (STOP).
```

Initialization
If $|K(MD)|$ is small, we can evaluate all the deterministic Markov policies directly. Therefore, assume $|vert(K)| > N$. Identify N independent vertices of K, say $v^1, v^2, \dots, v^N$, and the N policies for which they are the corresponding final distributions. There vertices can be found by finding the best response in N independent directions. Set $V_1 := \{v^1, v^2, \dots, v^N\}$ and set P_1 the convex hull of V_1.

Identification
The following lemma shows that step 5 of Algorithm 9.13 either identifies a new vertex of K not in P_k (if $x \notin V_k$) or identifies $aff(F)$ as a non-active facet which is a supporting hyperplane of K (if $x \in V_k$).

Lemma 9.39
Let F and x be respectively the active facet of P_k chosen in step 4(d) and the vertex x computed in step 5(a). Then,
(1) If $x/notinV_k$, then x is a new vertex of K not in P_k.
(2) If $x \in V_k$, then aff (F) *is a non-active facet which is a supporting hyperplane of K.*

Proof

(1) By the definition of best response, $x \in vert(K)$. Since $V_k = P_k \cap vert(K)$, $x \notin V_k$ implies $x \notin P_k$. Hence, x is a new vertex of K not in P_k.

(2) Let $\mathit{aff}(F) = \{y \mid y^T d = b\}$, where d is the outward normal relative to P_k, and let $b^* := x^T d$. Then, $\{y \mid y^T d = b^*\}$, the hyperplane parallel to $\mathit{aff}(F)$ through x, is - by the definition of best response - a supporting hyperplane of K. Since $x \in V_k$, $\mathit{aff}(F) = \{y \mid yTd = b^*\}$. Therefore, $\mathit{aff}(F)$ is a non-active facet which is a supporting hyperplane of K. □

Updating

If $x \in V_k$, then $V_{k+1} = V_k$, so the polytope formed by these vertices stays the same except that F is classified as non-active.

If $x \notin V_k$, then $V_{k+1} \neq V_k$. In this case every facet of P_k, except F, is also a facet of P_{k+1}, and the new facets of P_{k+1} are formed from the appropriate combinations of x with the vertices of F.

Lemma 9.40

Algorithm 9.13 terminates after a finite number of iterations with a vertex x^m which satisfies $h(x^m) = h(x^*)$ and with corresponding deterministic Markov policy R_m which is a variance-penalized optimal policy.

Proof

Since $K(MD)$ is finite and $vert(K) \subseteq K(MD)$, $vert(K)$ is also finite. Thus there are only a finite number of iterations at which a new vertex is actually identified. In the intervening iterations the vertex set and the approximating polytope stay constant and each iteration uses a different facet of this polytope to define the direction of search. Since the number of facets of each polytope is finite, the number of intervening iterations each time is also finite and so the overall number of iterations is finite.

Let P_m be the terminal polytope and let x^m be the terminal vertex generated by the algorithm. By construction, $vert(P_m) \subseteq vert(K)$. Let $F_1, F_2, \dots, F_r$ be the finite set of facets of P_m, and let $\mathit{aff}(F_r) = \{y \mid y^T d^i = b_i\}$, where d^i is the outward normal relative to P_m. Then, we have $P_m = \{y \mid y^T d^i \leq b^i,\ 1 \leq i \leq r\}$.

Since P_m is the terminal polytope, each $\{y \mid y^T d^i = b^i\}$ is a supporting hyperplane of K, implying $K \subseteq \{y \mid y^T d^i \leq b^i\}$ for $i = 1, 2, \dots, r$. Therefore, we obtain $K \subseteq \{y \mid y^T d^i \leq b^i,\ 1 \leq i \leq r\} = P_m$ and hence $vert(K) \subseteq vert(P_m)$. Combining the above gives $vert(K) = vert(P_m)$.

Furthermore, we have $h(x^*) = max_{x \in vert(K)}\, h(x) = max_{x \in vert(P_m)}\, h(x) = h(x^m)$, and the corresponding deterministic Markov policy R_m is a variance-penalized optimal policy. □

We will also describe a modified algorithm for finding x^* which uses vertex elimination to avoid having to explicitly check all the vertices of K. Let P_k be the current approximation of K, let F be the chosen active facet of P_k, and let $\mathit{aff}(F) = \{y \mid y^T d = b\}$, where d is the outward normal relative to P_k. Furthermore, let x be the best response in the direction d.

Assume, on checking, it turns out that $\mathit{aff}(F)$ is not a supporting hyperplane of K. Then, F divides K into two regions K_F and $K \backslash K_F$, where $K_F := K \cap \{y \mid y^T d \geq b\}$ and where $P_k \subseteq K \cap \{y \mid y^T d \geq b\}$.

In principle the maximum of $h(x)$ over K could be at any of the vertices of K and, in the absence of other information, the algorithm would not terminate until it had found and evaluated all the currently unknown vertices of K lying in K_F. However, Algorithm 9.13 has provided extra useful information. Let $u^1, u^2, \dots, u^s$ be the known vertices of F, where u^i was identified as the best response in some known direction d^i. Let $b_i := (u^i)^T d^i,\ 1 \leq i \leq s$. Then, by the definition of best response, K is contained in each known half-space $\{y \mid y^T d^i \leq b_i\}$. Hence, $K_F \subseteq \big\{ \bigcap_{i=1}^s \{y \mid y^T d^i \leq b_i\} \big\} \cap \{y \mid y^T d \geq b\}$.

Let Q_F be the known polyhedral set $\big\{ \bigcap_{i=1}^s \{y \mid y^T d^i \leq b_i\} \big\} \cap \{y \mid y^T d \geq b\}$. If Q_F is bounded, we can find the finite set $vert(Q_F)$ and the value $h_F := max_{y \in QF}\, h(y) = max_{y \in vert(Q_F)}\, h(y)$. Since h is a convex function and $K_F \subseteq Q_F$, the value h_F is an upper bound of $h(y)$ over K_F.

If $h_F \leq h(x^k)$, where x^k is the best vertex in our current approximating polytope P_k, then no vertex of K in K_F can be better than x^k and we can eliminate all vertices of K_F from further consideration. Note that it is only worth computing h_F if $h(x) \leq h(x^k)$, because otherwise $h(x^k) < h(x) \leq h_F$, the last inequality since $x \in Q_F$.

The above motivates the following modification of the algorithm in two respects. Firstly, step 5 is replaced by the following.

```
Identification
 (a)  Compute the best response $x$ in the direction $d$ of the outward normal of $\mathit{aff}(F)$ relative
      to $P_k$ and compute the corresponding deterministic Markov policy $R$.
 (b)  **if** $x \in V_k$ **then go to** step 6(a)
      **else go to** step 6(b)
 (c)  Compute $h(x)$.
 (d)  **if** $h(x) > h(x^k)$ or bf if $Q_F$ is bounded **then go to** step 6(b)
      **else go to** step 5(e)
 (e)  Compute $h_F$.
 (f)  **if** $h_F \leq h(x^k)$ **then go to** step 6(a)
      **else go to** step 6(b)
```

Secondly, at step 6(b), we now classify a facet F as a non-active facet of P_k if F is identified as a supporting hyperplane of K or if $h_F \leq h(x^k)$.

The number of iterations required by the modified algorithm is potentially smaller than by Algorithm 9.13, but more effort is required for the additional computation of h_F at some iterations.

We close this section by a comparison of the two approaches: on one hand the solution of the nonlinear program and on the other hand the geometric, linear algebra approach of Algorithm 9.13. For simplicity in comparing the two approaches, assume the same number of actions in each state: $|A(i)| = M$ for all $i \in S$.

Then, the nonlinear programming formulation involves of the order of $T \times N \times M$ variables $x_{j,t}(a)$ with $T \times N$ equality constraints and $T \times N \times M$ non-negativity constraints. In contrast, the geometrical, linear algebra approach splits the problem up into an iterative sequence, where each iteration involves the solution of a dynamic programming problem (the best response algorithm) and a polytope updating problem. The size of the dynamic programming computation is linear in T and M and at most quadratic in N, while each polytope updating problem is solved in a space of dimension N, so independent of T and M. However, the number of iterations is at most $|vert(K)| \leq |K(MD)| \leq M^{T \times N}$.

9.6 Deterministic MDPs

9.6.1 Introduction

An MDP is said to be deterministic if each action uniquely determines the next state of the process. In other words, the probability distribution associated with each action assigns probability 1 to one of the states. Deterministic Markov decision problems are denoted as DMDPs. A DMDP can be conveniently represented as a network, i.e. a directed graph with weights on the arcs.

The vertices of the graph correspond to the states of the DMDP and the arcs correspond to the actions. If in state i action $a \in A(i)$ is chosen which has a transition with probability 1 to state j, then the graph has an arc from state i to state j with as weight the cost $c_i(a)$ (we assume in this section that we have costs instead of rewards, which can be assumed without loss of generality by taking $c_i(a) := -r_i(a)$.

For the limiting average cost criterion the DMDP is strongly related to the well-known problem of finding a minimum mean weight cycle in a directed graph. This problem is analyzed in [158] and for this problem a polynomial time algorithm with complexity $\mathcal{O}(NM)$ is known, where N is the number of states and M the number of action, i.e. $M := \sum_{i=1}^N |A(i)|$. Solving discounted DMDPs seems to be somewhat harder, but also in this case an $\mathcal{O}(NM)$ algorithm can be obtained.

9.6.2 Average costs

Let f^∞ be a deterministic policy. Then, f^∞ induces in each state i exactly one outgoing arc $\big(i, f(i)\big)$ with weight $c_i(f)$. Hence, for each starting state i the policy f^∞ generates an infinite path $i \rightarrow f(i) \rightarrow f\big(f(i)\big) \rightarrow \cdots$. Define for $k = 0, 1, 2, \dots$ the state $f^k(i)$ recursively by $f^0(i) := i$ and $f^k(i) := f\big(f^{k-1}(i)\big)$ for $k = 1, 2, \dots$. With this notation, given starting point i and policy f^∞, we obtain the path $f^0(i) \rightarrow f^1(i) \rightarrow f^2(i) \rightarrow \cdots$ with weights $c_{f^0(i)}\big(f^1(i)\big), c_{f^1(i)}\big(f^2(i)\big), c_{f^2(i)}\big(f^3(i)\big), \dots$.

Since the number of states is finite, after at most N steps the path meets a state, say state $f^{k_2}(i)$, which was already in this path, say as state $f^{k_1}(i)$, i.e. $f^{k_2}(i) = f^{k_1}(i)$. From that point, the cycle $C := \{f^{k_1}(i), f^{k_2+1}(i), \cdots, f^{k_2}(i) = f^{k_1}(i)\}$ is repeated infinitely, implying that $\phi_i(f^\infty)$, the limiting average costs of policy f^∞ given starting state i, equals the mean-weight of cycle C.

Therefore, an optimal policy f^∞ for the minimum average costs can be determined as follows:

1. Find the cycle C_1 with minimum mean-weight and let $f(i),\ i \in C_1$, be the actions in cycle C_1.
2. Find $S_1 := \{j \notin C_1 \mid C_1$ is reachable from $j\}$ and let $f(j),\ j \in S_1$, be the action that create a path from j to C_1.
3. Repeat the steps 1 and 2 in the graph in which the states $C_1 \cap S_1$ are removed from S.

In the next subsection we present two algorithms to find a cycle with minimum mean-weight. These algorithms are based on shortest paths and linear programming, respectively.

Minimum mean-weight cycles

Let $D = (V, A)$ be a directed graph with weight function $w : A \to \mathbb{R}$, and let $N = |V|$. We define the mean-weight $w(C)$ of a cycle $C = \{a_1, a_2, \dots, a_k\}$ of arcs $a_i \in A$ by $w(C) := \frac{1}{k} \sum_{i=1}^k w(a_i)$. Let $w^* := min_C\, w(C)$, where C ranges over all directed cycles in D. A cycle C^* for which $w(C^*) = w^*$ is called a minimum mean-weight cycle.

Minimum mean-weight cycles via shortest paths

Assume that every vertex $v_j \in V,\ j \neq 1$, is reachable from a source vertex v_1. This assumption is without loss of generality: otherwise, add a vertex v_0 and arcs $(v_0, v_j),\ 1 \leq j \leq N$, with weight 0, and since v_0 is in none of the cycles, the problem does not change. Let $F(j)$ be the length, i.e. the weight, of the shortest path from v_1 to v_j and let $F_k(j)$ be the weight of the shortest path from v_1 to v_j with exactly k arcs. If there is no path from v_1 to v_j with exactly k arcs, then $F_k(j) := \infty$.

Lemma 9.41

If $w^* = 0$, then $F(j) = min_{0 \leq k \leq N-1}\, F_k(j)$ and $max_{0 \leq k \leq N-1}\, \frac{F_N(j) - F_k(j)}{N-k} \geq 0$ for all $j \in V$.

Proof

Since $w^* = 0$ there are no negative cycles. Hence, there is a shortest path from v_1 to v_j without any cycle, so with at most $N-1$ arcs, and consequently, $F(j) = min_{0 \leq k \leq N-1}\, F_k(j)$ for all $j \in V$. Assume that there exists a vertex v_j for which $max_{0 \leq k \leq N-1}\, \frac{F_N(j) - F_k(j)}{N-k} < 0$.
Then, $F_N(j) - F_k(j) < 0$ for $k = 0, 1, \dots, N-1$. The shortest path from v_1 to v_j with exactly N arcs consists of a path from v_1 to some v_i, a cycle C from v_i to v_i with $N - k \geq 1$ arcs, and a path from v_i to v_j. Since $F_N(j) < F_k(j)$, the weight $w(C) < 0$, which contradicts $w^* = 0$. □

Lemma 9.42

If $w^* = 0$, then $max_{0 \leq k \leq N-1}\, \frac{F_N(j) - F_k(j)}{N-k} = 0$ for some $j \in V$.

Proof

Since $w^* = 0$, there exists a cycle C^* with $w(C^*) = 0$. Let v_i be a vertex on C^*. Take a shortest path from v_1 to v_i and then extend the path going along the cycle C^* so many times that we end with a path P from v_1 to v_i with at least N arcs. Then, $w(P) = F(i) = F_k(i)$ for some $0 \leq k \leq N-1$. Let P^* be the first N arcs of P, ending in v_j on C^*. Since a part of a shortest path is also a shortest path, we have $w(P^*) = F(j) = F_N(j)$. Because also $F_N(j) = F(j) \leq F_k(j)$ for all $k \geq 0$, we obtain $max_{0\leq k\leq N-1} \frac{F_N(j)-F_k(j)}{N-k} \leq 0$. Using Lemma 9.41, we conclude $max_{0\leq k\leq N-1} \frac{F_N(j)-F_k(j)}{N-k} = 0$ for some $j \in V$. □

Corollary 9.13
If $w^* = 0$, then $min_{j\in V}\, max_{0\leq k\leq N-1} \frac{F_N(j)-F_k(j)}{N-k} = 0$.

Theorem 9.55
$w^* = min_{j\in V}\, max_{0\leq k\leq N-1} \frac{F_N(j)-F_k(j)}{N-k}$.

Proof
Adding c to each arc increases w^* by c. It also increases $F_N(j)$ by Nc and $F_k(j)$ by kc. Hence, $\frac{F_N(j)-F_k(j)}{N-k}$ is also increased by c. Thus, for a general w^*, by taking $c := -w^*$, we have the case with $w^* = 0$ and, by Corollary 9.13, we obtain $w^* = min_{j\in V}\, max_{0\leq k\leq N-1} \frac{F_N(j)-F_k(j)}{N-k}$. □

Lemma 9.43
The minimum mean-weight cycle can be computed by an algorithm with complexity $\mathcal{O}(NM)$, where M is the total number of arcs.

Proof
We first add - if necessary - a vertex, say v_1, such that any other vertex is reachable from v_1. This can be executed with complexity $\mathcal{O}(N)$. Then, compute $F_k(j)$ for $k = 0, 1, \ldots, N$ in $\mathcal{O}(NM)$ time by evaluating the recurrence $F_{k+1}(j) = min_i \{F_k(i) + w\big((i,j)\big)\}$. Finally, in $\mathcal{O}(N^2)$ time, determine $min_{j\in V}\, max_{0\leq k\leq N-1} \frac{F_N(j)-F_k(j)}{N-k}$. Because $M \geq N$, the overall complexity is $\mathcal{O}(NM)$. The minimum mean-weight cycle C^* can be found by bookkeeping, in the recurrence computation $F_N(j) = min_i \{F_{N-1}(i) + w\big((i,j)\big)\}$ the vertex, say j_N, with $F_N(j) = F_{N-1}(j_N) + w\big((j_N, j)\big)$. Then, determine the vertex j^* such that $w^* = max_{0\leq k\leq N-1} \frac{F_N(j^*)-F_k(j^*)}{N-k}$. Finally, the minimum mean-weight cycle C^* can be found by following the sequence $j^* \leftarrow j^*_N \leftarrow (j^*_N)_N \leftarrow$ until we are back in j^*. This does not increases the complexity of the method. □

Algorithm 9.14 *Computation of a minimum mean-weight cycle via shortest paths*
Input: Instance of a directed graph with weights on the arcs and such that every vertex $v_j \in V,\ j \neq 1$ is reachable from a source vertex v_1.
Output: A mean-weight cycle C^*.

```
1. $F_0(1) := 0$; for all $j \neq 1$ do $F_0(j) := \infty$;
   for $k = 1$ until $N$ do
       for all $j \neq 1$ do $F_k(j) := min_i \{F_{k-1}(i) + \big((i,j)\big)\}$.
2. for all $j \neq 1$ do determine $j_N$ such that $F_N(j) = F_{N-1}(j_N) + w\big((j_N, j)\big)$.
3. for all $j \neq 1$ do determine $max_{0\leq k\leq N-1} \frac{F_N(j)-F_k(j)}{N-k}$.
```

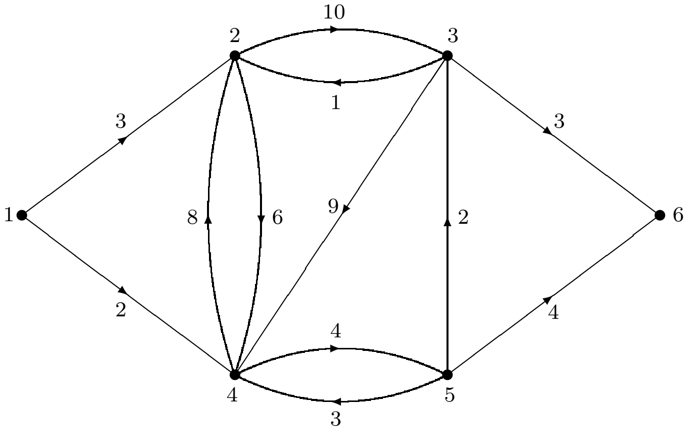
4. Determine w^* and j^* such that $w^* = min_{j\in V}\, max_{0\le k\le N-1}\, \frac{F_N(j)-F_k(j)}{N-k} = max_{0\le k\le N-1}\, \frac{F_N(j^*)-F_k(j^*)}{N-k}$.

5. Set $i_1 := j^*$; $l := 1$; **while** $i_l \ne j^*$ **do begin** $l := l+1$; $i_l := (i_{l-1})_N$ **end**

   $C^* := \{i_l, i_{l-1}, \dots, i_1\}$.

Example 9.20

Consider the directed graph below. The weights are placed next to the arcs.



Notice that all vertices are reachable from vertex v_1 and v_1 is not in any cycle.

Step 1: Computation of $F_k(j)$ for all k and j:

$k=0:$	$F_0(1)=0;$	$F_0(2)=\infty;$	$F_0(3)=\infty;$	$F_0(4)=\infty;$	$F_0(5)=\infty;$	$F_0(6)=\infty.$
$k=1:$		$F_1(2)=3;$	$F_1(3)=\infty;$	$F_1(4)=2;$	$F_1(5)=\infty;$	$F_1(6)=\infty.$
$k=2:$		$F_2(2)=10;$	$F_2(3)=13;$	$F_2(4)=9;$	$F_2(5)=6;$	$F_2(6)=\infty.$
$k=3:$		$F_3(2)=14;$	$F_3(3)=8;$	$F_3(4)=9;$	$F_3(5)=13;$	$F_3(6)=10.$
$k=4:$		$F_4(2)=9;$	$F_4(3)=15;$	$F_4(4)=16;$	$F_4(5)=13;$	$F_4(6)=11.$
$k=5:$		$F_5(2)=16;$	$F_5(3)=15;$	$F_5(4)=15;$	$F_5(5)=20;$	$F_5(6)=17.$
$k=6:$		$F_6(2)=16;$	$F_6(3)=22;$	$F_6(4)=22;$	$F_6(5)=19;$	$F_6(6)=18.$

Step 2: The determination of the nodes i_6:

$2_6 = 3;\ 3_6 = 5;\ 4_6 = 2;\ 5_6 = 4;\ 6_6 = 3.$

Step 3: Computation of $max_{1\le k\le N-1}\, \frac{F_N(j)-F_k(j)}{N-k}$ for all $j \ge 2$:

$j=2: max_{0\le k\le 5}\, \frac{F_6(2)-F_k(2)}{6-k} = max\{\frac{16-3}{5}, \frac{16-10}{4}, \frac{16-14}{3}, \frac{16-9}{2}, \frac{16-16}{1} = \frac{7}{2}.$

$j=3: max_{0\le k\le 5}\, \frac{F_6(3)-F_k(3)}{6-k} = max\{\frac{22-\infty}{5}, \frac{22-13}{4}, \frac{22-8}{3}, \frac{22-15}{2}, \frac{22-15}{1} = 7.$

$j=4: max_{0\le k\le 5}\, \frac{F_6(4)-F_k(4)}{6-k} = max\{\frac{22-2}{5}, \frac{22-9}{4}, \frac{22-9}{3}, \frac{22-16}{2}, \frac{22-15}{1} = 7.$

$j=5: max_{0\le k\le 5}\, \frac{F_6(5)-F_k(5)}{6-k} = max\{\frac{19-\infty}{5}, \frac{19-6}{4}, \frac{19-13}{3}, \frac{19-13}{2}, \frac{19-20}{1} = \frac{13}{4}.$

$j=6: max_{0\le k\le 5}\, \frac{F_6(6)-F_k(6)}{6-k} = max\{\frac{18-\infty}{5}, \frac{18-\infty}{4}, \frac{18-10}{3}, \frac{18-11}{2}, \frac{18-17}{1} = \frac{7}{2}.$

Step 4: Computation of w^* and j^*:

$w^* = min_{j\in V}\, max_{0\le k\le N-1}\, \frac{F_N(j)-F_k(j)}{N-k} = min\{\frac{7}{2}, 7, 7, \frac{13}{4}, \frac{7}{2}\} = \frac{13}{4};\ j^* = 5.$

Step 5: Determination of the minimum mean-weight cycle:

$i_1 = 5;\ l = 1;\ l = 2;\ i_2 = 5_6 = 4;\ l = 3;\ i_3 = 4_6 = 2;\ l = 4;\ i_4 = 2_6 = 3;\ l = 5;\ i_5 = 3_6 = 5.$

$C^* = \{5, 3, 2, 4, 5\}.$

Minimum mean-weight cycles via linear programming

Associate the variable x_{ij} to each arc $(i,j) \in A$ and denote the cost of this arc by c_{ij}. Then, the linear programming formulation for the minimum mean-weight cycle problem is:

$$min\left\{\sum_{(i,j)} c_{ij}x_{ij} \;\middle|\; \begin{array}{rcl} \sum_j x_{ij} - \sum_j x_{ji} & = & 0,\ i \in V \\ \sum_{(i,j)} x_{ij} & = & 1 \\ x_{ij} & \geq & 0,\ (i,j) \in A \end{array}\right\}. \qquad (9.105)$$

A solution x of (9.105) determines in D some circulation with constant sum, which equals 1. The next lemma shows that if x is an extreme solution of (9.105), then $\{(i,j) \mid x_{ij} > 0\}$ represents an elementary cycle.

Lemma 9.44

If x is an extreme solution of (9.105), then $A_x := \{(i,j) \mid x_{ij} > 0\}$ represents an elementary cycle, say C, and the value of the objective function equals the mean-weight of C.

Proof

We first prove that A_x contains a cycle. Since $\sum_{(i,j)} x_{ij} = 1$, $A_x \neq \emptyset$. Let $(k,l) \in A_x$. Then, $\sum_j x_{kj} > 0$ and, by the circulation, $\sum_j x_{jk} > 0$. Hence, $x_{jk} > 0$ for some $j \in V$, i.e. $(j,k) \in A_x$ for some $j \in V$. Let $V_x := \{k \in V \quad \sum_l x_{kl} > 0\}$. Then, in any vertex $k \in V_x$ there is an outgoing and an ingoing arc from A_x. As D is a finite graph, crossing the vertices from V_x we shall come to the first vertex from which we started. Thus, there exists a cycle in A_x.

Next, we show that A_x is an elementary cycle. Therefore, choose an elementary cycle C from A_x. Assume that $C = \{(i_1,i_2),(i_2,i_3),\ldots,(i_n,i_{n+1}) = (i_n,i_1)\}$ and that $x = (x^1,x^2,0)$, where x^1 and x^2 are the subvectors corresponding to C and $A_x \backslash C$, respectively. Furthermore, let $\theta := min_{(i,j)\in C}\, x^1_{ij} > 0$.

Consider the following two solutions: $y^1 := \frac{1}{1-n\theta} \cdot (x^1 - \theta e_n, x^2, 0)$ and $y^2 := \frac{1}{n\theta} \cdot (\theta e_n, 0, 0)$. It is easy to verify that y^1 and y^2 are feasible solutions, and that $x = (1-n\theta)\cdot y^1 + (n\theta)\cdot y^2$. Hence, x is not an extreme solution, showing that A_x represents an elementary cycle C and let $C := \{(i_1,i_2),(i_2,i_3),\ldots,(i_n,i_{n+1}) = (i_n,i_1)\}$. Then, by the circulation of an elementary cycle, $x = (x^1,0)$ with $x^1 = \frac{1}{n}\cdot e_n$. This implies that $\sum_{(i,j)} c_{ij}x_{ij} = \frac{1}{n}\cdot\sum_{(i,j)\in C} c_{ij}$, i.e. the mean-weight of C. □

Corollary 9.14

If x is an extreme optimal solution of (9.105), then $A_x := \{(i,j) \mid x_{ij} > 0\}$ represents a minimum mean-weight cycle.

Algorithm 9.15 *Computation of a minimum mean-weight cycle via linear programming*

Input: Instance of a directed graph $D = (S,A)$ with weights c_{ij} on the arcs.

Output: A mean-weight cycle C^*.

1. Determine an extreme optimal solution x^* of liner program (9.105).

2. The arcs of $C^* := \{(i,j) \mid x_{ij} > 0\}$ represent a minimum mean-weight cycle.

Example 9.20 (continued)

The linear program for this example is:

$min\{3x_{12} + 2x_{14} + 10x_{23} + 6x_{24} + x_{32} + 9x_{34} + 3x_{36} + 8x_{42} + 4x_{45} + 2x_{53} + 3x_{54} + 4x_{56}\}$

subject to

$$\begin{array}{rrrrrrrrrrrrl}
x_{12} & +\,x_{14} & & & & & & & & & & & = 0 \\
-\,x_{12} & & +\,x_{23} & +\,x_{24} & -\,x_{32} & +\,x_{34} & +\,x_{36} & & -\,x_{42} & & & & = 0 \\
 & & -\,x_{23} & & +\,x_{32} & -\,x_{34} & & -\,x_{53} & & & & & = 0 \\
 & -\,x_{14} & & -\,x_{24} & & & & & +\,x_{42} & +\,x_{45} & -\,x_{54} & & = 0 \\
 & & & & & & & x_{53} & & -\,x_{45} & +\,x_{54} & +\,x_{56} & = 0 \\
 & & & & & & -\,x_{36} & & & & & -\,x_{56} & = 0 \\
x_{12} & +\,x_{14} & +\,x_{23} & +\,x_{24} & +\,x_{32} & +\,x_{34} & +\,x_{36} & +\,x_{53} & +\,x_{42} & +\,x_{45} & +\,x_{54} & +\,x_{56} & = 0
\end{array}$$

$$x_{12},\ x_{14},\ x_{23},\ x_{24},\ x_{32},\ x_{34},\ x_{36},\ x_{53},\ x_{42},\ x_{45},\ x_{54},\ x_{56}\ \geq\ 0$$

The optimal solution is: $x_{12} = 0,\ x_{14} = 0,\ x_{23} = 0,\ x_{24} = \frac{1}{4},\ x_{32} = \frac{1}{4},\ x_{34} = 0,\ x_{36} = 0,\ x_{53} = \frac{1}{4},$ $x_{42} = 0,\ x_{45} = \frac{1}{4},\ x_{54} = 0,\ x_{56} = 0$. The value of the objective function is 13 , which is the minimum mean-weight of the cycles.

9.6.3 Discounted costs

Let $D = (S, A)$ be the directed graph corresponding to the DMDP. A deterministic policy f^∞ and a starting state i induce an infinite path $f^0(i) \rightarrow f^1(i) \rightarrow f^2(i) \rightarrow \cdots$ with α-discounted cost $v^\alpha(f^\infty) = \sum_{t=1}^\infty \alpha^{t-1} c_{f^{t-1}(i)}\big(f^t(i)\big)$. From the general theory of discounted MDPs we obtain the following results for the value vector $v^\alpha := min_{f^\infty} v^\alpha(f^\infty)$.

Theorem 9.56

(1) $v_i^\alpha = min_{\{j \mid (i,j) \in A\}} \{c_{ij} + \alpha v_j^\alpha\},\ i \in S.$

(2) If $v_i^\alpha = c_{ik(i)} + \alpha v_{k(i)}^\alpha,\ i \in S$, *then* f^∞, *where* $f(i) := k(i),\ i \in S$, *is an optimal policy.*

(3) v^α *is the optimal solution of the linear program* $max\{\sum_j v_j \mid v_i - \alpha v_j \leq c_{ij},\ (i,j) \in A\}$.

Remark

The linear program in part (3) of Theorem 9.56 has a special form. Firstly, it contains two variables per inequality. Secondly, in each inequality the coefficient of one variable is positive, while the coefficient of the other variable is negative. Strongly polynomial-time algorithms for checking the feasibility of linear programs with two variables per inequality were obtained by Cohen and Megiddo ([44]) and Hochbaum and Naor ([118]).

An arc $(i,j) \in A$ for which $v_i^\alpha = c_{ij} + \alpha v_j^\alpha$ is said to be *optimal.* Notice that each vertex i has at least one optimal outgoing arc. An *optimal path (cycle)* is a path (cycle) composed of optimal arcs. It follows from the finiteness of S that for every $i \in S$ there exists an optimal path P from i to some vertex j and an optimal cycle C that passes through j such that the infinite path $PC^\infty := PCC\ldots$ has cost v^α. The infinite path PC^∞ corresponds to a deterministic policy f^∞. If path P is of length k, where $0 \leq k \leq N-1$, and C is a cycle of length l, where $1 \leq l \leq N$, then we have $v_i^\alpha = C(P) + \frac{\alpha^k}{1-\alpha^l} \cdot c(C)$, with $c(P) = \sum_{t=1}^k \alpha^{t-1} c_{f^{t-1}(i)}\big(f^t(i)\big)$ and $c(C) = \sum_{t=1}^l \alpha^{t-1} c_{f^{t-1}(i)}\big(f^t(i)\big)$.

Minimum discounted-weight infinite path via Bellman-Ford

We shall present an $\mathcal{O}(NM)$ algorithm for solving discounted DMDPs. The algorithm is inspired by the algorithm for finding a minimum mean-weight cycle via shortest paths. We have seen that an optimal policy corresponds to an optimal path of type PC^∞. The algorithm starts by computing, for each vertex i and each $k = 0, 1, \dots, N-1$, the weight $U_k(i)$ of the shortest discounted path of k arcs that starts at vertex i. This can easily be done in $\mathcal{O}(NM)$ time using an algorithm based on the classical Bellman-ford algorithm[2] for computing single-source shortest paths.

Step 1 of the algorithm is as follows:

```
  for each $i \in S$ do $U_0(i) := 0$;
  for $k = 1$ step 1 until $N$ do
      for each $i \in S$ do $U_k(i) := min_{\{j \mid (i,j) \in A\}} \{c_{ij} + \alpha U_{k-1}(j)\}$
```

It is obvious that this algorithm has complexity $\mathcal{O}(NM)$. Then, for each $j \in S$, the algorithm computes $G_0(j) := max_{0 \le k \le N-1} \frac{U_N(j) - \alpha^{N-k} U_k(j)}{1 - \alpha^{N-k}}$. The ratio $\frac{U_N(j) - \alpha^{N-k} U_k(j)}{1 - \alpha^{N-k}}$ is the analog of the nondiscounted ratio $\frac{F_N(j) - F_k(j)}{N-k}$ in the mean-weight cycle. To understand the intuition behind the ratio $\frac{U_N(j) - \alpha^{N-k} U_k(j)}{1 - \alpha^{N-k}}$, note that if the optimal policy corresponds to the optimal path PC^∞, then for starting vertices j on the cycle C, the optimal path is C^∞ with discounted weight v_j^α. Let C be a cycle with $N-k$ arcs, where $0 \le k \le N-1$. Then, the discounted weight of cycle C is $U_N(j) - \alpha^{N-k} U_k(j)$. Hence, the discounted weight of C^∞ is

$$\{U_N(j) - \alpha^{N-k} U_k(j)\} \cdot \{1 + \alpha^{N-k} + \left(\alpha^{N-k}\right)^2 + \cdots\} = \frac{U_N(j) - \alpha^{N-k} U_k(j)}{1 - \alpha^{N-k}}.$$

We show shortly (see Lemma 9.45) that $G_0(j)$, the maximum over $0 \le k \le N-1$ of these ratios, is an upper bound on the value v_j^α for all $j \in S$, and we also will show that $G_0(j) = v_j^\alpha$ for some $j \in S$.

In the nondiscounted case, the expression $w^* = min_{j \in V} \, max_{0 \le k \le N-1} \frac{F_N(j) - F_k(j)}{N-k}$ is the minimum mean-weight of a cycle in the graph. In the discounted case, things are more complicated. In particular, to compute the correct values of all vertices j, we have to perform a second Bellman-Ford stage, which is related to an infinite path PC^∞. For every $j \in S$ and $k = 0, 1, \dots N-1$, $G_k(j)$ will be a k-arc path that starts in j, where the discounted cost now takes into account the $G_0(j)$-value of the last vertex on the path. Finally, the algorithm computes $min_{0 \le k \le N-1} \, G_k(j)$. We shall also show that $v_j^\alpha = min_{0 \le k \le N} \, G_k(j)$. The whole algorithm is as follows.

Algorithm 9.16 *Computation of the discounted value vector of an DMDP*
Input: Instance of a directed graph $D = (S, A)$ with weights c_{ij} on the arcs.
Output: The discounted value vector v^α.

```
1. for each $i \in S$ do $U_0(i) := 0$.
   for $k = 1$ step 1 until $N$ do
       for each $i \in S$ do $U_k(i) := min_{\{j \mid (i,j) \in A\}} \{c_{ij} + \alpha U_{k-1}(j)\}$.
```

[2] see http://en.wikipedia.org/wiki/Bellman-Ford_algorithm

```
2. for each $j \in S$ do $G_0(j) := max_{0 \leq k \leq N-1} \frac{U_N(j) - \alpha^{N-k} U_k(j)}{1-\alpha^{N-k}}$.
3. for $k = 1$ step 1 until $N-1$ do
       for each $i \in S$ do $G_k(i) := min_{\{j \mid (i,j) \in A\}} \{c_{ij} + \alpha G_{k-1}(j)\}$.
4. for each $i \in S$ do $v_i^\alpha := min_{0 \leq k \leq N} G_k(j)$.
```

For the proof of the correctness of algorithm 9.16, not for the algorithm itself, we use the graph G^* with costs c^*_{ij}, obtained from G by adding an auxiliary vertex s (the sink) and arcs from all $j \in S$ to the sink s with cost $c^*_{js} := 0$ (the other costs are unchanged, i.e. $c^*_{ij} := c_{ij}$ for all $(i,j) \in A$). Any k-arc path $P = \{j_0, j_1, \dots, j_k\}$ in G can be extended to a $(k+1)$-arc path P^*, where $P^* := \{j_0, j_1, \dots, j_k, j_{k+1} = s\}$ in G^* and $c^*(P^*) = c(P)$. Let $S^* := S \cup \{s\}$ and $A^* := A \cup \{(i,s) \mid i \in S\}$. For every potential function $v : S^* \to \mathbb{R}$, we define modified arc costs $c^*(v)$ by $c^*_{ij}(v) := c^*_{ij} - v_i + \alpha v_j$ for all $(i,j) \in A^*$.

It is easy to see that if $P = \{j_0, j_1, \dots, j_k\}$ is a k-arc path in G with corresponding $(k+1)$-arc path $P^* = \{j_0, j_1, \dots, j_k, s\}$ in G^*, then $c^*(v; P^*) = c^*(P^*) - v_{j_0} + \alpha^{k+1} v_s = c(P) - v_{j_0} + \alpha^{k+1} v_s$. Hence, P is a k-arc path in G, starting in j with minimal discounted costs with respect to c in G if and only if the corresponding path P^* in G^* is a $(k+1)$-arc path P^* in G^* from j to s with minimal discounted costs with respect to $c^*(v)$ in G^*.

Consider the potential function $v_j := v^\alpha$, $j \in S$ (the value v_s will be set in a way to be explained in the proof of Lemma 9.46). With this potential function we have $c^*_{ij}(v) = c_{ij} - v^\alpha_i + \alpha v^\alpha_j \geq 0$ for all $(i,j) \in A$ and $c^*_{ij}(v) = 0$ for all optimal arcs $(i,j) \in A$. The only arcs that can have negative costs $c^*(v)$ are arcs to the sink, i.e. arcs (i,s) for which $c^*_{is}(v) = c^*_{is} - v^\alpha_i + \alpha v_s = -v^\alpha_i + \alpha v_s$.

Let $U^*_k(j)$ be the minimal discounted cost with respect to a $(k+1)$-arc path from j to s in G^* with cost $c^*(v)$ in G^*. Clearly, $U^*_k(j) = U_k(j) - v^\alpha_j + \alpha^{k+1} v_s$.

Lemma 9.45
$v^\alpha_j \leq G_0(j)$ for all $j \in S$.

Proof
Since $G_0(j) := max_{0 \leq k \leq N-1} \frac{U_N(j) - \alpha^{N-k} U_k(j)}{1-\alpha^{N-k}}$, we have to show that for every $j \in S$ there exists a $0 \leq k \leq N-1$ such that $U_N(j) - \alpha^{N-k} U_k(j) \geq (1-\alpha^{N-k}) v^\alpha_j$. For every $j \in S$ and $0 \leq k \leq N-1$, we have

$$\begin{aligned} U_N(j) - \alpha^{N-k} U_k(j) &= \{U^*_N(j) + v^\alpha_j - \alpha^{N+1} v_s\} - \alpha^{N-k} \{U^*_k(j) + v^\alpha_j - \alpha^{k+1} v_s\} \\ &= \{U^*_N(j) - \alpha^{N-k} U^*_k(j)\} + v^\alpha_j \{1 - \alpha^{N-k}\}. \end{aligned}$$

Therefore, it is enough to show that for every $j \in S$ there exists a $0 \leq k \leq N-1$ such that $U^*_N(j) \geq \alpha^{N-k} U^*_k(j)$. Take any $j \in S$. Consider an N-arc path P in G, starting in j, that attains the value $U_N(j)$. The corresponding path P^* then attains the value $U^*_N(j)$. As P is composed of N arcs, it must contain a cycle. Let $P = P_1 P_2 P_3$, where P_2 is a cycle. Let $N_1, N_2, N_3 \geq 0$ be the number of arcs on P_1, P_2 and P_3, respectively. Note that $N_2 \geq 1$. Let $P_4 := P_1 P_3$ be the path with $N - N_2$ arcs obtained from P by removing the cycle P_2. For every arc (i,j) on P we have $c^*_{ij}(v) \geq 0$. Hence, $c^*(v; P_1)$, $c^*(v; P_2)$, $c^*(v; P_3) \geq 0$. Thus,

$$\begin{aligned} U_N^*(j) &= c^*(v;P_1) + \alpha^{N_1}c^*(v;P_2) + \alpha^{N_1+N_2}c^*(v;P_3) \\ &\geq \alpha^{N_2}c^*(v;P_1) + \alpha^{N_1}c^*(v;P_2) + \alpha^{N_1+N_2}c^*(v;P_3) \\ &\geq \alpha^{N_2}\{c^*(v;P_1) + \alpha^{N_1}c^*(v;P_3)\} \\ &= \alpha^{N_2}c^*(v;P_4). \end{aligned}$$

For $k := N - N_2$ we have $0 \leq k \leq N-1$ and $U_N^*(j) \geq \alpha^{N-k}c^*(v;P_4) \geq \alpha^{N-k}U_k^*(j)$, the last inequality because P_4 corresponds to a $(k+1)$-arc path P_4^* from j to s in G^* with cost $c^*(v;P_4)$ and $U_k^*(j)$ is the minimal discounted cost of the $(k+1)$-arcs paths from j to s in G^*. □

Lemma 9.46

On every optimal cycle in G, i.e. a cycle with optimal arcs, there is at least one vertex j for which $v_j^\alpha = G_0(j)$.

Proof

Since, by Lemma 9.45, $v_j^\alpha \leq G_0(j)$ for all $j \in S$, it is enough to show that for every optimal cycle in G there is at least one vertex j for which $v_j^\alpha \geq max_{0\leq k\leq N-1}\, \frac{U_N(j)-\alpha^{N-k}U_k(j)}{1-\alpha^{N-k}}$, i.e.

$$v_j^\alpha(1-\alpha^{N-k}) \geq U_N(j) - \alpha^{N-k}U_k(j) \text{ for every } 0 \leq k \leq N-1.$$

As we have seen in the proof of Lemma 9.45, for every $j \in S$ and every $0 \leq k \leq N-1$, we have

$$U_N(j) - \alpha^{N-k}U_k(j) = \{U_N^*(j) - \alpha^{N-k}U_k^*(j)\} + v_j^\alpha(1-\alpha^{N-k}).$$

Therefore, we have to show that on every optimal cycle in G there is at least one vertex j for which $U_N^*(j) \leq \alpha^{N-k}U_k^*(j)$ for every $0 \leq k \leq N-1$.

Let C be an optimal cycle in G. For every $(i,j) \in C$ we have $c_{ij}^*(v) = 0$. Let i be an arbitrary, but fixed, vertex on C. Choose v_s such that

$$min_{0\leq k\leq N-1}\, U_k^*(i) = 0, \text{ i.e. } min_{0\leq k\leq N-1}\, \{U_k(i) - v_i^\alpha + \alpha^{k+1}v_s\} = 0,$$

which is equivalent to $v_s := max_{0\leq k\leq N-1}\, \frac{1}{\alpha^{k+1}} \cdot \{v_i^\alpha - U_k(i)\}$.

Suppose that $U_l^*(i) = 0$. Let P_1 be the l-arc path in G, starting in i, such that for the corresponding path P^* in G^* from i to s we have $c^*(v;P_1^*) = 0$. Extend P_1 to an N-arc path P_2 in G by adding to its beginning arcs from the optimal cycle C. Let vertex $j \in C$ be the starting point of P_2. As all arcs added to P_2 have a $c^*(v)$-cost of 0, we clearly have $c^*(v;P_2^*) = 0$ and thus $U_N^*(j) \leq 0$. We claim that for every $0 \leq k \leq N-1$, we have $U_k^*(j) \geq 0$, and thus $U_N^*(j) \leq 0 \leq \alpha^{k-1}U_k^*(j)$ for every $0 \leq k \leq N-1$.

It remains to show that $U_k^*(j) \geq 0$ for every $0 \leq k \leq N-1$. Suppose, for the sake of contradiction, that $U_k^*(j) < 0$ for some $0 \leq k \leq N-1$. Let P_3 be a k-arc path of G, starting in j such that the corresponding path P_3^* in G^* from j to s is such that $c^*(v;P_3^*) < 0$. Extend P_3, by adding to its beginning arcs from C, to a path P_4, that starts in i, the vertex i chosen before. Recall that i and j are both on C. As all arcs of C has $c^*(v)$-cost of 0, we have $c^*(v;P_4^*) = c^*(v;P_3^*) < 0$.

If P_4 contains N or more arcs, then it contains a cycle in G. By removing such a cycle, we get a path P_5 with $c^*(v;P_5^*) \leq c^*(v;P_4^*) < 0$. By repeating removing cycles, we get a path P_6 that starts at i, contains at most $N-1$ arcs and has $c^*(v;P_6^*) < 0$, a contradiction to the property that $min_{0\leq k\leq N-1}\, U_k^*(i) = 0$. □

Lemma 9.47

$v_i^\alpha \leq G_k(i)$ for every $i \in S$ and every $0 \leq k \leq N-1$.

Proof

We apply induction on k. For $k = 0$ the result is shown in Lemma 9.45. Suppose that $v_i^\alpha \leq G_k(i)$ for every $i \in S$ and some $1 \leq k \leq N-2$. We can write

$$G_{k+1}(i) = min_{\{j \mid (i,j) \in A\}} \{c_{ij} + \alpha G_k(j)\} \geq min_{\{j \mid (i,j) \in A\}} \{c_{ij} + \alpha v_j^\alpha\} = v_i^\alpha, \ i \in S. \qquad \square$$

Lemma 9.48

For every $i \in S$ there exists some $0 \leq k \leq N-1$ such that $v_i^\alpha = G_k(i)$.

Proof

Every $i \in S$ has at least one optimal outgoing arc. Take any $i \in S$ and let $P = \{i = i_0, i_1, \dots, i_N\}$ be a path with optimal arcs, starting in i. As P contains N arcs, it must contain an optimal cycle C. By Lemma 9.46, there exists a vertex i_l on C for which $v_{i_l}^\alpha = G_0(i_l)$. As the arcs (i_k, i_{k+1}) of P are optimal arcs, we get that $v_{i_k}^\alpha = c_{i_k, i_{k+1}} + \alpha v_{i_{k+1}}^\alpha$. We shall show by induction on k that $G_{l-k}(i_k) \leq v_{i_k}^\alpha$ for $k = l, l-1, \dots, 0$. For $k = l$, we have seen that $G_0(i_l) = v_{i_l}^\alpha$. Suppose that $G_{l-k}(i_k) \leq v_{i_k}^\alpha$ for some $l-1 \geq k \geq 0$. Then, we can write

$$\begin{aligned} G_{l-k+1}(i_{k-1}) &= min_{\{j \mid (i,j) \in A\}} \{c_{i_{k-1}j} + \alpha G_{l-k}(j)\} \\ &\leq c_{i_{k-1}i_k} + \alpha G_{l-k}(i_k) \leq c_{i_{k-1}i_k} + \alpha v_{i_k}^\alpha = v_{i_{k-1}}^\alpha. \end{aligned}$$

As $i = i_0$, we get in particular $G_l(i) \leq v_i^\alpha$. Combined with Lemma 9.47, we get $G_l(i) = v_i^\alpha$ for some $0 \leq l \leq N-1$, as required. $\square$

Theorem 9.57

Algorithm 9.16 computes the discounted value vector in $\mathcal{O}(NM)$ time.

Proof

Step 1 and step 3 of the algorithm have complexity $\mathcal{O}(NM)$. The steps 2 and 4 have complexity $\mathcal{O}(N^2) \leq \mathcal{O}(NM)$. Hence, the overall complexity is $\mathcal{O}(NM)$.

By Lemma 9.45, we get $v_i^\alpha \leq G_0(i)$ for every $i \in S$, and, by Lemma 9.46, on every optimal cycle in G there is at least one vertex j for which $v_j^\alpha = G_0(j)$. The algorithm next computes $G_k(i) := min_{\{j \mid (i,j) \in A\}} \{c_{ij} + \alpha G_{k-1}(j)\}$ for every $i \in S$ and $k = 1, 2, \dots, N-1$. By Lemma 9.47, we get $v_i^\alpha \leq G_k(i)$ for every $i \in S$ and every $0 \leq k \leq N-1$.

If (i,j) is an optimal arc, i.e. $v_i^\alpha = c_{i,j} + \alpha v_j^\alpha$, if furthermore $v_j^\alpha = G_{k-1}(j)$, then

$$v_i^\alpha = c_{i,j} + \alpha v_j^\alpha = v_i^\alpha = c_{i,j} + \alpha G_{k-1}(j) \geq min_{\{j \mid (i,j) \in A\}} \{c_{ij} + \alpha G_{k-1}(j)\} = G_k(i) \geq v_i^\alpha,$$

implying $G_k(i) = v_i^\alpha$. Take any $i \in S$. Let P_1 be a path of N optimal arcs and starting in i. P_1 must contain a cycle C and, by Lemma 9.46, there exists a vertex j on C for which $v_j^\alpha = G_0(j)$. Let P_2 be the subpath of P_1 leading from i to the first occurrence of j on C and let l be the number of arcs in P_2 $(0 \leq l \leq N-1)$. As all arcs of P_2 are optimal, we get $v_i^\alpha = G_l(i)$. Since, by Lemma 9.47, $v_i^\alpha \leq G_k(i)$ for every $0 \leq k \leq N-1$, we have $v_i^\alpha = min_{0 \leq k \leq N-1} G_k(i)$, which is computed in Algorithm 9.16. $\square$

Remark

From the value vector v^α one obtains easily an optimal deterministic policy: take in each state i an action $f(i)$ such that $v^\alpha_i = c_{ij} + \alpha v^\alpha_j$, where j is the state corresponding to the transition from state i if action $f(i)$ is chosen. This can be done in $\mathcal{O}(M)$ time.

Example 9.21

Consider the following DMDP with $\alpha = \frac{1}{2}$. $S = \{1,2,3,4,5,6\}$; $A(1) = A(2) = \{1,2\}$; $A(3) = \{1,2,3\}$; $A(4) = \{1,2,\}$; $A(5) = \{1,2,3\}$ and $A(6) = \{1\}$. $p_{12}(1) = p_{14}(2) = p_{23}(1) = 1$; $p_{24}(2) = 1$; $p_{32}(1) = p_{34}(2) = p_{36}(3) = p_{45}(1) = p_{42}(2) = p_{54}(1) = p_{56}(2) = p_{53}(3) = p_{66}(1) = 1$. $c_1(1) = 3$; $c_1(2) = 2$; $c_2(1) = 10$; $c_2(2) = 6$; $c_3(1) = 1$; $c_3(2) = 9$; $c_3(3) = 3$; $c_4(1) = 4$; $c_4(2) = 8$; $c_5(1) = 3$; $c_5(2) = 4$; $c_5(3) = 2$; $c_6(1) = 7$.

Note that the graph of this DMDP is the same as in Example 9.20, but with a loop in state 6.

Step 1:

$k = 0:$ $U_0 = (0,0,0,0,0,0)$.

$k = 1:$ $U_1 = (2,6,1,4,2,7)$.

$k = 2:$ $U_2(1) = min\{3 + \frac{1}{2}\cdot 6, 2 + \frac{1}{2}\cdot 4\} = 4$; $U_2(2) = min\{10 + \frac{1}{2}\cdot 1, 6 + \frac{1}{2}\cdot 4\} = 8$;
$U_2(3) = min\{1 + \frac{1}{2}\cdot 6, 9 + \frac{1}{2}\cdot 4, 3 + \frac{1}{2}\cdot 7\} = 4$; $U_2(4) = min\{4 + \frac{1}{2}\cdot 2, 8 + \frac{1}{2}\cdot 6\} = 5$;
$U_2(5) = min\{3 + \frac{1}{2}\cdot 4, 4 + \frac{1}{2}\cdot 7, 2 + \frac{1}{2}\cdot 1\} = \frac{5}{2}$; $U_2(6) = 7 + \frac{1}{2}\cdot 7 = \frac{21}{2}$.
$U_2 = (4, 8, 4, 5, \frac{5}{2}, \frac{21}{2})$.

$k = 3:$ $U_3(1) = min\{3 + \frac{1}{2}\cdot 8, 2 + \frac{1}{2}\cdot 5\} = \frac{9}{2}$; $U_3(2) = min\{10 + \frac{1}{2}\cdot 4, 6 + \frac{1}{2}\cdot 5\} = \frac{17}{2}$;
$U_3(3) = min\{1 + \frac{1}{2}\cdot 8, 9 + \frac{1}{2}\cdot 5, 3 + \frac{1}{2}\cdot \frac{21}{2}\} = 5$; $U_3(4) = min\{4 + \frac{1}{2}\cdot \frac{5}{2}, 8 + \frac{1}{2}\cdot 8\} = \frac{21}{4}$;
$U_3(5) = min\{3 + \frac{1}{2}\cdot 5, 4 + \frac{1}{2}\cdot \frac{21}{5}, 2 + \frac{1}{2}\cdot 4\} = 4$; $U_3(6) = 7 + \frac{1}{2}\cdot \frac{21}{2} = \frac{49}{4}$.
$U_3 = (\frac{9}{2}, \frac{17}{2}, 5, \frac{21}{4}, 4, \frac{49}{4})$.

$k = 4:$ $U_4(1) = min\{3 + \frac{1}{2}\cdot \frac{17}{2}, 2 + \frac{1}{2}\cdot \frac{21}{4}\} = \frac{37}{8}$; $U_4(2) = min\{10 + \frac{1}{2}\cdot 5, 6 + \frac{1}{2}\cdot \frac{21}{4}\} = \frac{69}{8}$;
$U_4(3) = min\{1 + \frac{1}{2}\cdot \frac{17}{2}, 9 + \frac{1}{2}\cdot \frac{21}{4}, 3 + \frac{1}{2}\cdot \frac{49}{4}\} = \frac{21}{4}$; $U_4(4) = min\{4 + \frac{1}{2}\cdot 4, 8 + \frac{1}{2}\cdot \frac{17}{2}\} = 6$;
$U_4(5) = min\{3 + \frac{1}{2}\cdot \frac{21}{4}, 4 + \frac{1}{2}\cdot \frac{49}{4}, 2 + \frac{1}{2}\cdot 5\} = \frac{9}{2}$; $U_4(6) = 7 + \frac{1}{2}\cdot \frac{49}{4} = \frac{105}{8}$.
$U_4 = (\frac{37}{8}, \frac{69}{8}, \frac{21}{4}, 6, \frac{9}{2}, \frac{105}{8})$.

$k = 5:$ $U_5(1) = min\{3 + \frac{1}{2}\cdot \frac{69}{8}, 2 + \frac{1}{2}\cdot 6\} = 5$; $U_5(2) = min\{10 + \frac{1}{2}\cdot \frac{21}{4}, 6 + \frac{1}{2}\cdot 6\} = 9$;
$U_5(3) = min\{1 + \frac{1}{2}\cdot \frac{69}{8}, 9 + \frac{1}{2}\cdot 6, 3 + \frac{1}{2}\cdot \frac{105}{8}\} = \frac{85}{16}$; $U_5(4) = min\{4 + \frac{1}{2}\cdot \frac{9}{2}, 8 + \frac{1}{2}\cdot \frac{69}{8}\} = \frac{25}{4}$;
$U_5(5) = min\{3 + \frac{1}{2}\cdot 6, 4 + \frac{1}{2}\cdot \frac{105}{8}, 2 + \frac{1}{2}\cdot \frac{21}{4}\} = \frac{37}{8}$; $U_5(6) = 7 + \frac{1}{2}\cdot \frac{105}{8} = \frac{217}{16}$.
$U_5 = (5, 9, \frac{85}{16}, \frac{25}{4}, \frac{37}{8}, \frac{217}{16})$.

$k = 6:$ $U_6(1) = min\{3 + \frac{1}{2}\cdot 9, 2 + \frac{1}{2}\cdot \frac{25}{4}\} = \frac{41}{8}$; $U_6(2) = min\{10 + \frac{1}{2}\cdot \frac{85}{16}, 6 + \frac{1}{2}\cdot \frac{25}{4}\} = \frac{73}{8}$;
$U_6(3) = min\{1 + \frac{1}{2}\cdot 9, 9 + \frac{1}{2}\cdot \frac{25}{4}, 3 + \frac{1}{2}\cdot \frac{217}{16}\} = \frac{11}{2}$; $U_6(4) = min\{4 + \frac{1}{2}\cdot \frac{37}{8}, 8 + \frac{1}{2}\cdot 9\} = \frac{101}{16}$;
$U_6(5) = min\{3 + \frac{1}{2}\cdot \frac{25}{4}, 4 + \frac{1}{2}\cdot \frac{217}{16}, 2 + \frac{1}{2}\cdot \frac{85}{16}\} = \frac{149}{32}$; $U_6(6) = 7 + \frac{1}{2}\cdot \frac{217}{16} = \frac{441}{32}$.
$U_6 = (\frac{41}{8}, \frac{73}{8}, \frac{11}{2}, \frac{101}{16}, \frac{149}{32}, \frac{441}{32})$.

Step 2:

In the next table we have computed the values $\frac{U_N(j) - \alpha^{N-k}U_k(j)}{1-\alpha^{N-k}}$ for $j \in S$ and $0 \leq k \leq N-1$.

	$k=0$	$k=1$	$k=2$	$k=3$	$k=4$	$k=5$
$j=1$	5.206	5.226	5.2	5.214	5.292	5.25
$j=2$	9.270	9.226	9.2	9.214	9.292	9.25
$j=3$	5.587	5.645	5.6	5.571	5.583	5.688
$j=4$	6.413	6.387	6.4	6.464	6.417	6.375
$j=5$	4.730	4.742	4.7	4.75	4.708	4.688
$j=6$	14	14	14	14	14	14

Hence, $G_0 = (5.292, 9.292, 5.688, 6.464, 4.8, 14)$.

Step 3:

$k=1:$ $G_1(1) = min\{3+\frac{1}{2}\cdot 9.292, 2+\frac{1}{2}\cdot 6.464\} = 5.323;$
$G_1(2) = min\{10+\frac{1}{2}\cdot 5.688, 6+\frac{1}{2}\cdot 6.464\} = 9.232;$
$G_1(3) = min\{1+\frac{1}{2}\cdot 9.292, 9+\frac{1}{2}\cdot 6.464, 3+\frac{1}{2}\cdot 14\} = 5.6464;$
$G_1(4) = min\{4+\frac{1}{2}\cdot 4.8, 8+\frac{1}{2}\cdot 9.292\} = 6.4;$
$G_1(5) = min\{3+\frac{1}{2}\cdot 6.646, 4+\frac{1}{2}\cdot 14, 2+\frac{1}{2}\cdot 5.688\} = 4.844;$
$G_1(6) = 7+\frac{1}{2}\cdot 14 = 14.$
$G_1 = (5.232, 9.232, 5.464, 6.4, 4.844, 14).$

$k=2:$ $G_2(1) = min\{3+\frac{1}{2}\cdot 9.232, 2+\frac{1}{2}\cdot 6.4\} = 5.2;$
$G_2(2) = min\{10+\frac{1}{2}\cdot 5.646, 6+\frac{1}{2}\cdot 6.4\} = 9.2;$
$G_2(3) = min\{1+\frac{1}{2}\cdot 9.232, 9+\frac{1}{2}\cdot 6.4, 3+\frac{1}{2}\cdot 14\} = 5.616;$
$G_2(4) = min\{4+\frac{1}{2}\cdot 4.844, 8+\frac{1}{2}\cdot 9.232\} = 6.422;$
$G_2(5) = min\{3+\frac{1}{2}\cdot 6.4, 4+\frac{1}{2}\cdot 14, 2+\frac{1}{2}\cdot 5.646\} = 4.823;$
$G_2(6) = 7+\frac{1}{2}\cdot 14 = 14.$
$G_2 = (5.2, 9.2, 5.616, 6.422, 4.823, 14).$

$k=3:$ $G_3(1) = min\{3+\frac{1}{2}\cdot 9.2, 2+\frac{1}{2}\cdot 6.422\} = 5.211;$
$G_3(2) = min\{10+\frac{1}{2}\cdot 5.616, 6+\frac{1}{2}\cdot 6.422\} = 9.211;$
$G_3(3) = min\{1+\frac{1}{2}\cdot 9.2, 9+\frac{1}{2}\cdot 6.422, 3+\frac{1}{2}\cdot 14\} = 5.6;$
$G_3(4) = min\{4+\frac{1}{2}\cdot 4.823, 8+\frac{1}{2}\cdot 9.2\} = 6.4115;$
$G_3(5) = min\{3+\frac{1}{2}\cdot 6.422, 4+\frac{1}{2}\cdot 14, 2+\frac{1}{2}\cdot 5.616\} = 4.808;$
$G_3(6) = 7+\frac{1}{2}\cdot 14 = 14.$
$G_3 = (5.211, 9.211, 5.6, 6.4115, 4.808, 14).$

$k=4:$ $G_4(1) = min\{3+\frac{1}{2}\cdot 9.211, 2+\frac{1}{2}\cdot 6.4115\} = 5.20575;$
$G_4(2) = min\{10+\frac{1}{2}\cdot 5.6, 6+\frac{1}{2}\cdot 6.4115\} = 9.20575;$
$G_4(3) = min\{1+\frac{1}{2}\cdot 9.211, 9+\frac{1}{2}\cdot 6.4115, 3+\frac{1}{2}\cdot 14\} = 5.6057;$
$G_4(4) = min\{4+\frac{1}{2}\cdot 4.808, 8+\frac{1}{2}\cdot 9.211\} = 6.404;$
$G_4(5) = min\{3+\frac{1}{2}\cdot 6.4115, 4+\frac{1}{2}\cdot 14, 2+\frac{1}{2}\cdot 5.6\} = 4.8;$
$G_4(6) = 7+\frac{1}{2}\cdot 14 = 14.$
$G_4 = (5.20575, 9.20575, 5.6057, 6.404, 4.8, 14).$

$k=5:$ $G_5(1) = min\{3+\frac{1}{2}\cdot 9.20575, 2+\frac{1}{2}\cdot 6.404\} = 5.202;$
$G_5(2) = min\{10+\frac{1}{2}\cdot 5.60575, 6+\frac{1}{2}\cdot 6.404\} = 9.202;$
$G_5(3) = min\{1+\frac{1}{2}\cdot 9.20575, 9+\frac{1}{2}\cdot 6.404, 3+\frac{1}{2}\cdot 14\} = 5.602875;$

$G_5(4) = min\{4 + \frac{1}{2} \cdot 4.8, 8 + \frac{1}{2} \cdot 9.20575\} = 6.4;$
$G_5(5) = min\{3 + \frac{1}{2} \cdot 6.404, 4 + \frac{1}{2} \cdot 14, 2 + \frac{1}{2} \cdot 5.6057\} = 4.80285;$
$G_5(6) = 7 + \frac{1}{2} \cdot 14 = 14.$
$G_5 = (5.202, 9.202, 5.602875, 6.4, 4.80285, 14).$

Step 4:
$v_1^\alpha = min\{5.292, 5.232, 5.2, 5.211, 5.20575, 5.202\} = 5.2.$
$v_2^\alpha = min\{9.292, 9.232, 9.2, 9.211, 9.20575, 9.202\} = 9.2.$
$v_3^\alpha = min\{5.688, 5.646, 5.616, 5.6, 5.6057, 5.602875\} = 5.6.$
$v_4^\alpha = min\{6.464, 6.4, 6.422, 6.4115, 6404, 6.4\} = 6.4.$
$v_5^\alpha = min\{4.8, 4.844, 4.823, 4.808, 4.8, 4.8085\} = 4.8.$
$v_6^\alpha = min\{14, 14, 14, 14, 14, 14\} = 14.$
Hence, $v^\alpha = (5.2, 9.2, 5.6, 6.4, 4.8, 14)$ and the optimal policy is: $f(1) = 2,\ f(2) = 2,\ f(3) = 1,$ $f(4) = 1,\ f(5) = 3,\ f(6) = 1.$

9.7 Semi-Markov decision processes

9.7.1 Introduction

In the models studied in previous chapters, the decision maker could choose actions only at a discrete set of time points. However, some applications, particularly in queueing control, are more natural modeled by allowing decision time points at random times. We have seen some examples in Chapter 8.

In this section we consider semi-Markov decision processes. They generalizes MDPs by:

(1) Allowing the decision maker to choose actions whenever the state of the system changes.
(2) Allowing the time spent in a particular state to follow an arbitrary probability distribution.
(3) Modeling the system evolution in continuous-time.

We have seen in Section 8.4 that stochastic systems with exponential time distributions can be transformed to standard MDPs by the method of *uniformization*. By semi-Markov decision processes (SMDPs) we can also analyze systems with nonexponential distributions.

In SMDPs action choice determines the joint probability distribution of the subsequent state and the time between decision epochs. In its simplest form, the system evolves by remaining in a state for a random amount of time and then jumping to a different state. In greater generality, the system state may change several times between decision epochs; however, only the state at the decision epochs is relevant to the decision maker. We refer to these models as semi-Markov because the distribution time to the next decision epoch and the state at that time depend on the past only through the the state and action chosen at the current decision epoch and because the time between transitions may follow an arbitrary probability distribution.

We also study *continuous-time Markov decision processes* (CTMDPs). This model may be viewed as a special case of an SMDP because the intertransition times are exponentially distributed and the actions are chosen at every transition.

We restrict our attention to models in which decision epochs may only occur after a distinguished set of transitions. Such models are also called *Markov renewal programs.* We also restrict to infinite horizon models with discounted and average reward optimality criteria and assume the all models have *time-homogeneous* rewards and transition probabilities, i.e., the rewards and transition probabilities are independent on the time.

9.7.2 Model formulation

Defining the state system requires some care. In queueing control models the system state may vary between decision epochs. However, we only allow action choice to depend on the system content at points of time when actions may be implemented. From the perspective of this model, what happens between decision epochs provides no relevant information to the decision maker. To this end we distinguish between the *natural process* and the *semi-Markov decision process.* The natural process models the state evolution of the system as if it were observed continually throughout time, while the SMDP represents the evolution of the system at decision points only. The two processes agree at decision epochs. We also refer an SMDP as an *embedded Markov decision process.*

For example, in a queueing admission control model, the semi-Markov decision process describes the system state at arrival times only, while the natural process describes the system state at all time points. To determine rewards we need information about the queue size at all times. This is described by the natural process. To determine whether to admit a job, we need only know the number of customers in the queue when a job enters the system. This is described by the SMDP.

Let S denote the finite *state space. Decision epochs* occur at random points of time determined by the specific model description. If at some decision epoch the system occupies state i, the decision maker must choose an *action* a from a finite set $A(i)$. If at some decision epoch the state is i and action a is chosen, then $Q_{ij}(a,t)$ denotes the probability that the subsequent decision epoch is at most t time units later and the state at that decision epoch equals j. In most applications $Q_{ij}(a,t)$ is not provided directly. Instead, the basic model quantities are the *sojourn time* $T_{ij}(a)$ and the *transition probabilities* $p_{ij}(a)$. Conditional to the events that at the current decision epoch the state is i and action a is chosen and that at the subsequent decision epoch is state j, the sojourn time is the random time between the current and the subsequent decision epoch. We assume that the function $F_{ij}(a,t) := \mathbb{P}\{T_{ij}(a) \leq t\}$ is known. By the transition probabilities $p_{ij}(a)$, we denote that, given the state i and the action a at the current decision epoch, the state at the subsequent decision epoch is state j. Hence,

$$Q_{ij}(a,t) = p_{ij}(a) \cdot F_{ij}(a,t) \text{ for all } (i,a) \in S \times A,\ j \in S \text{ and } t \geq 0. \tag{9.106}$$

If $F_{ij}(a,t)$ is independent of the state j, then we denote the sojourn time by $F_i(a,t)$. In order to ensure that an infinite number of transitions does not occur in a finite interval, we shall assume throughout that the following condition holds.

Assumption 9.3

There exist $\delta > 0$ and $\varepsilon > 0$ such that $\sum_j Q_{ij}(a, \delta) \leq 1 - \varepsilon$ for every $(i, a) \in S \times A$.

Or, in other words, this condition states that for every state i and every action $a \in A(i)$, there is a positive probability of at least ε that the transition time will be greater than δ. Consequently, the expected number of decisions epochs in a finite interval is finite.

We have the following reward structure. When the decision maker chooses action a in state i a lump sum $r_i(a)$ is earned immediately; further, a reward at *rate* $s_i(a)$ is imposed until the next decision epoch: if the next decision epoch occurs after $t_i(a)$ units of time, then the reward during this period is $r_i(a) + t_i(a) \cdot s_i(a)$. We shall transform the model into one with an expected reward which depends only on the state of the semi-Markov decision process at a decision epoch and the action chosen. We shall denote these rewards by $r^*_i(a),\ (i, a) \in S \times A$.

We now describe the evolution of the SMDP. At the start, the system occupies state i_1 and the decision maker chooses action a_1. As a consequence the system remains in state i_1 for t_1 units of time at which point the system changes to state i_2 and the next decision epoch occurs. The decision maker chooses action a_2 and the system remains in state i_2 for t_2 units of time at which point the system changes to state i_3, and so on. Let $h_n := (i_1, a_1, t_1, \dots, i_{n-1}, a_{n-1}, t_{n-1}, i_n)$ denote the history of the process up to nth decision epoch. In contrast to discrete-time models, the history also contains the sojourn times. Equivalently, we can view discrete MDPs as special cases of SMDPs in which $t_n = 1$ for $n = 1, 2, \dots$.

As in discrete-time models, we consider several classes of decision rules. They may be either deterministic or randomized and Markovian or history dependent. Note that history dependent decision rules are defined in terms of the above expanded notion of history which includes the sojourn times. We use the same notations for the classes of policies as in MDPs: $C, C(M), C(S)$ and $C(D)$ are the sets of general, Markov, stationary and deterministic policies, respectively.

9.7.3 Examples

Example 9.22 *A two-state semi-Markov decision process*

Let $S = \{1, 2\},\ A(1) = \{1, 2\}$ and $A(2) = \{1\}$. We assume the following timing of events. After choosing an action in a given state, the system remains there for an action-dependent random period of time. Then a transition occurs and the next action can be chosen. Since the transitions occur only at the end of a sojourn in a state, we specify transition probabilities for the embedded Markov decision process.

Let $p_{11}(1) = p_{12}(1) = 0.5;\ p_{11}(2) = 0,\ p_{12}(2) = 1;\ p_{21}(1) = 0.1,\ p_{22}(1) = 0.9$.

We assume that the sojourn times are uniformly distributed and independent of the states at the subsequent decision epoch. Let $F_1(1, t) = U[0, 2],\ F_1(2, t) = U[0, 4]$ and $F_2(1, t) = U[0, 3]$.

The lump sum rewards are given by $r_1(1) = 0,\ r_1(2) = -1$ and $r_2(1) = 0$; the continuous reward rates are: $s_1(1) = 5,\ s_1(2) = 10$ and $s_2(1) = -1$.

From (9.106) it follows that

$$Q_{11}(1,t) = Q_{12}(1,t) = \begin{cases} 0.5 \cdot \frac{t}{2} & 0 \leq t \leq 2 \\ 0.5 & t > 2 \end{cases};\ Q_{11}(2,t) = 0,\ t \geq 0;\ Q_{12}(2,t) = \begin{cases} 1 \cdot \frac{t}{4} & 0 \leq t \leq 4 \\ 1 & t > 4 \end{cases}$$

$$Q_{21}(1,t) = \begin{cases} 0.1 \cdot \frac{t}{t} & 0 \leq t \leq 3 \\ 0.1 & t > 3 \end{cases};\ Q_{22}(1,t) = \begin{cases} 0.9 \cdot \frac{t}{t} & 0 \leq t \leq 3 \\ 0.9 & t > 3 \end{cases}$$

Example 9.23 *Admission control for a $G/M/1$ queueing system*
In a $G/M/1$ queueing system, interarrival times are independent and follow an arbitrary distribution, service times at the single server are independent and exponentially distributed. A controller regulates the system load by rejecting ($a = 0$) or accepting ($a = 1$) arriving jobs.
Let the state space for the natural process denote the number of jobs in a system with capacity N: $S = \{0, 1, \ldots, N\}$. We denote the interarrival time distribution by $G(\cdot)$ and its density by $g(\cdot)$. Further, we assume an exponential service rate with parameter μ, independent of the number of jobs in the system. Each arriving job contributes r units of reward and the system incurs a holding cost at rate $s(i)$ per time unit whenever there are i jobs in the system. Hence,

$$r_i(a) = \begin{cases} 0, \ i \in S, \ a = 0 \\ r, \ i \in S, \ a = 1 \end{cases} \text{ and } s_i(a) = -s(i), \ (i, a) \in S \times A.$$

Decisions are required only when jobs enter the system. The embedded Markov decision process models the system at these time points. We set $A(i) = \{0, 1\}, \ 0 \leq i \leq N - 1$ and $A(N) = \{0\}$. Action 0 denotes rejecting an arrival, while action 1 corresponds to accepting an arrival. Since decisions are made only at arrival time, we have $F_{ij}(a, t) = G(t)$ for all $(i, a) \in S \times A, \ j \in S$ and $t \geq 0$. In between arrivals, the natural state may change because of service completion. From elementary probability, the number of service completion in t units of time follows a Poisson distribution with parameter μt. Consequently, $\frac{(\mu t)^k}{k!} \cdot e^{-\mu t}$ is the probability of k departures during t units of time. Hence,

$$Q_{ij}(0, t) = \begin{cases} \int_0^t \frac{(\mu s)^(i-j)}{(i-j)!} \cdot e^{-\mu s} \cdot g(s) ds & 1 \leq j \leq i \leq N \\ \int_0^t \sum_{k \geq i} \frac{(\mu s)^k}{k!} \cdot e^{-\mu s} \cdot g(s) ds & j = 0 \\ 0 & j > i \end{cases}$$

$$Q_{ij}(1, t) = \begin{cases} \int_0^t \frac{(\mu s)^(i+1-j)}{(i+1-j)!} \cdot e^{-\mu s} \cdot g(s) ds & 1 \leq j \leq i + 1 \leq N \\ \int_0^t \sum_{k \geq i+1} \frac{(\mu s)^k}{k!} \cdot e^{-\mu s} \cdot g(s) ds & j = 0 \\ 0 & j > i + 1 \end{cases}$$

Example 9.24 *Service rate control in an $M/G/1$ queueing system*
An $M/G/1$ queueing system has a single server, independent exponential interarrival times, and independent service times which follow an arbitrary distribution. In the controlled model, the controller regulates the system by varying the service rate, where faster servers are more expensive. We assume that interarrival times are exponential with parameter λ, and that service distributions $G_a(\cdot)$, with densities $g_a(\cdot)$, where a can be drawn from a finite set $A = \{1, 2, \cdot, M\}$. In some applications, a will represent a scale parameter. For example, we might specify G_a to be exponential with parameter μ_a. Further, we assume that the controller may change the service rate only upon completion of a service, or on the arrival of a job to an empty system. Costs include a fixed cost K for changing the service rate, a holding cost rate $h(i)$ when there are i customers in the system and a cost rate $c(a)$ if action a is chosen.
We denote the state of the natural process by (i, a), where i denote the number of jobs in the system and a the index of the service distribution in use: $S = \{(i, a) \mid i = 0, 1, \ldots; \ a = 1, 2, \ldots, M\}$. The semi-Markov decision process describes these quantities at decision epochs.
In this model, the sojourn time distribution explicitly depends on both the state of the system and the chosen action. For $i \geq 1$, the next decision epoch occurs upon completion of a service:

$$F_{(i,a)}(b, t) = G_b(t) \text{ for all } i \geq 1, \ a, b \in A \text{ and } t \geq 0.$$

If $i = 0$, the next opportunity to change the service rate occurs when a job arrives:

$$F_{(0,a)}(b, t) = 1 - e^{-\lambda t} \text{ for all } a, b \in A \text{ and } t \geq 0.$$

We now provide the transition probabilities in the natural process. When the queue is empty, the next transition occurs at an arrival, so that

$$p_{(0,a)(1,b)}(b) = 1 \text{ for all } a, b \in A \text{ (all other transition probabilities are zero).}$$

If $i \geq 1$, the next opportunity to change the service rate occurs when a job is completed. If the service time is t, then the probability that during these t units of time k new jobs arrive is $\frac{(\lambda t)^k}{k!} \cdot e^{-\lambda t}$ for $k = 0, 1, \dots$. Therefore,

$$p_{(i,a)(i-1+k,b)}(b) = \int_0^\infty \tfrac{(\lambda t)^k}{k!} \cdot e^{-\lambda t} \cdot g_b(t)dt \text{ for } i \geq 1,\ a, b \in A \text{ and } k = 0, 1, \dots.$$

(all other transition probabilities are zero). For the rewards, we obtain:

$$r_{(i,a)}(b) = \begin{cases} 0 & i \geq 0,\ a \in A,\ b = a \\ -K & i \geq 0,\ a \in A,\ b \neq a \end{cases} ;\ s_{(i,a)}(b) = -h(i) - c(b),\ i \geq 0,\ a, b \in A$$

9.7.4 Discounted rewards

We assume continuous-time discounting at rate $\lambda > 0$. This means that the present value of one unit received t time units in the future equals $e^{-\lambda t}$. By setting $e^{-\lambda} := \alpha$, this corresponds with the discrete-time discount factor $\alpha \in (0, 1)$.

For a policy R, let $v_i^\lambda(R)$ denote the expected infinite-horizon discounted reward, given that the process occupies state i at the first decision epoch, defined by

$$v_i^\lambda(R) := \mathbb{E}_R \Big\{ \sum_{n=1}^\infty e^{-\lambda(T_1+T_2+\cdots+T_{n-1})} \cdot \{ r_{X_n}(Y_n) + s_{X_n}(Y_n) \cdot \int_0^{T_n} e^{-\lambda t}dt \} \ \Big|\ X_1 = i \Big\}. \tag{9.107}$$

In this expression $X_n,\ Y_n$ denote random variables of the state and action at the nth decision epoch; T_n denotes the random variable of the time between decision epoch n and decision epoch $n + 1$, where $T_1 + T_2 + \cdots + T_{n-1} := 0$ for $n = 1$.

Lemma 9.49

Let $\pi_{ij}(a, n, R, t) := \mathbb{P}_R\{X_n = j,\ Y_n = a,\ T_1 + T_2 + \cdots + T_{n-1} = t \mid X_1 = i\}$ *for* $i, j \in S,\ a \in A(i),\ n \in \mathbb{N}$, $R \in C$ *and* $t \geq 0$, *and let* $r_j^*(a) := r_j(a) + s_j(a) \cdot \sum_k p_{jk}(a) \int_0^\infty \{ \int_0^t e^{-}\lambda s ds\} dF_{jk}(a, t)$.
Then, $v_i^\lambda(R) = \sum_{n=1}^\infty \sum_{j,a} r_j^*(a) \cdot \int_0^\infty e^{-\lambda t} d\pi_{ij}(a, n, R, t),\ i \in S,\ R \in C$.

Proof

First, we observe that

$$\mathbb{E}_R \left\{ r_{X_n}(Y_n) + s_{X_n}(Y_n) \cdot \int_0^{T_n} e^{-\lambda t}dt \ \middle|\ X_n = j,\ Y_n = a \right\} =$$
$$\sum_k \mathbb{E}_R \left\{ r_{X_n}(Y_n) + s_{X_n}(Y_n) \cdot \int_0^{T_n} e^{-\lambda t}dt \ \middle|\ X_n = j,\ Y_n = a,\ X_{n+1} = k \right\} \cdot$$
$$\mathbb{P}_R\{X_{n+1} = k \mid X_n = j,\ Y_n = a\} =$$
$$\sum_k p_{jk}(a) \cdot \left\{ r_j(a) + s_j(a) \cdot \int_0^\infty \{\int_0^t e^{-\lambda s}ds\} dF_{jk}(a, t) \right\} =$$
$$r_j(a) + s_j(a) \cdot \sum_k p_{jk}(a) \int_0^\infty \{\int_0^t e^{-\lambda s}ds\} dF_{jk}(a, t) \ = \ r_j^*(a),\ (j, a) \in S \times A.$$

Since the random variables $T_1 + T_2 + \cdots + T_{n-1}$ and T_n are independent, given X_n and Y_n, we obtain

$$\mathbb{E}_R \left\{ e^{-\lambda(T_1+T_2+\cdots+T_{n-1})} \cdot \{ r_{X_n}(Y_n) + s_{X_n}(Y_n) \cdot \int_0^{T_n} e^{-\lambda t}dt\} \ \middle|\ X_1 = i \right\} =$$
$$\sum_{j,a} \int_0^\infty e^{-\lambda t} \cdot \left\{ \mathbb{E}_R\{ r_{X_n}(Y_n) + s_{X_n}(Y_n) \cdot \int_0^{T_n} e^{-\lambda t}dt \mid X_n = j,\ Y_n = a\} \right\} \cdot$$
$$d\,\mathbb{P}_R\{X_n = j,\ Y_n = a,\ T_1 + T_2 + \cdots + T_{n-1} \leq t \mid X_1 = i\} =$$
$$\sum_{j,a} r_j^*(a) \cdot \int_0^\infty e^{-\lambda t} d\,\mathbb{P}_R\{X_n = j,\ Y_n = a,\ T_1 + T_2 + \cdots + T_{n-1} \leq t \mid X_1 = i\} =$$
$$\sum_{j,a} r_j^*(a) \cdot \int_0^\infty e^{-\lambda t} d\pi_{ij}(a, n, R, t).$$

$\int_0^\infty e^{-\lambda t} d\pi_{ij}(a,n,R,t)$ may be interpreted as the expected discounted probability that, given $X_1 = i$, we have $X_n = j,\ Y_n = a$. By conditioning to the state and action at epoch $n-1$, we obtain the recursion

$$\sum_a \int_0^\infty e^{-\lambda t} d\pi_{ij}(a,n,R,t) = \sum_{l,b} \Big\{ \big\{ \int_0^\infty e^{-\lambda t} d\pi_{il}(b,n{-}1,R,t) \big\} \cdot p_{lj}(b) \cdot \big\{ \int_0^\infty e^{-\lambda s} dF_{lj}(b,s) \big\} \Big\}. \quad (9.108)$$

Define $w_n,\ M$ and ρ by

$$w_n := \sum_{j,a} \int_0^\infty e^{-\lambda t} d\pi_{ij}(a,n,R,t);\ M := max_{i,a} \Big\{ |r_i(a)| + \frac{|s_i(a)|}{\lambda} \Big\};\ \rho := max_{i,j,a} \int_0^\infty e^{-\lambda t} dF_{ij}(a,t). \quad (9.109)$$

Then, (9.108) and (9.109) imply

$$\begin{array}{rcl} w_n & = & \sum_j \Big\{ \sum_{l,b} \big\{ [\int_0^\infty e^{-\lambda t} d\pi_{il}(b,n-1,R,t)] \cdot p_{lj}(b) \cdot [\int_0^\infty e^{-\lambda s} dF_{lj}(b,s)] \big\} \Big\} \\ & = & \sum_{l,b} \Big\{ [\int_0^\infty e^{-\lambda t} d\pi_{il}(b,n-1,R,t)] \cdot \sum_j p_{lj}(b) \cdot [\int_0^\infty e^{-\lambda s} dF_{lj}(b,s)] \Big\} \\ & \leq & \sum_{l,b} [\int_0^\infty e^{-\lambda t} d\pi_{il}(b,n-1,R,t)] \cdot \rho \\ & = & \rho \cdot w_{n-1} \leq \cdots \leq \rho^{n-1} \cdot w_1 = \rho^{n-1}, \end{array}$$

because $w_1 = \sum_{j,a} \int_0^\infty e^{-\lambda t} d\pi_{ij}(a,1,R,t) = \int_0^\infty e^{-\lambda t} d\mathbb{R}_R\{X_1 = i,\ t \geq 0 \mid X_1 = i\} = 1$.

Furthermore, we have

$$\begin{array}{rcl} |r_j^*(a)| & \leq & |r_j(a)| + |s_j(a)| \cdot \sum_k p_{jk}(a) \cdot \int_0^\infty \big\{ \int_0^t e^{-\lambda s} ds \big\} dF_{jk}(a,t) \\ & = & |r_j(a)| + |s_j(a)| \cdot \sum_k p_{jk}(a) \cdot \int_0^\infty \frac{1}{\lambda} \big(1 - e^{-\lambda t}\big) dF_{jk}(a,t) \\ & \leq & |r_j(a)| + \frac{|s_j(a)|}{\lambda} \cdot \sum_k p_{jk}(a) \cdot \int_0^\infty \big(1 - e^{-\lambda t}\big) dF_{jk}(a,t) \\ & \leq & |r_j(a)| + \frac{|s_j(a)|}{\lambda} \cdot \sum_k p_{jk}(a) \cdot \int_0^\infty dF_{jk}(a,t) \\ & \leq & |r_j(a)| + \frac{|s_j(a)|}{\lambda}\ \leq\ M. \end{array}$$

Consequently, also noting that $\rho < 1$, we obtain

$$\sum_{n=1}^\infty \sum_{j,a} \int_0^\infty |r_j^*(a)| \cdot e^{-\lambda t} d\pi_{ij}(a,n,R,t)\ \leq M\ \cdot \sum_{n=1}^\infty w_n\ \leq\ \frac{M}{1-\rho} < \infty.$$

Hence, we may interchange the expectation operator and the infinite summation in the expression below:

$$\begin{array}{rcl} v_i^\lambda(R) & = & \mathbb{E}_R \Big\{ \sum_{n=1}^\infty e^{-\lambda(T_1+T_2+\cdots+T_{n-1})} \cdot \{ r_{X_n}(Y_n) + s_{X_n}(Y_n) \cdot \int_0^{T_n} e^{-\lambda t} dt \} \ \Big| \ X_1 = i \Big\} \\ & = & \sum_{n=1}^\infty \mathbb{E}_R \Big\{ e^{-\lambda(T_1+T_2+\cdots+T_{n-1})} \cdot \{ r_{X_n}(Y_n) + s_{X_n}(Y_n) \cdot \int_0^{T_n} e^{-\lambda t} dt \} \ \Big| \ X_1 = i \Big\} \\ & = & \sum_{n=1}^\infty \sum_{j,a} r_j^*(a) \cdot e^{-\lambda t} d\pi_{ij}(a,n,R,t). \end{array}$$

□

The *value vector* v^λ of a discounted SMDP is defined by $v_i^\lambda := sup_R\, v_i^\lambda(R),\ i \in S$. A policy R^* is an *optimal policy* if $v_i^\lambda(R^*) = v_i^\lambda,\ i \in S$. From the proof of Lemma 9.49 it follows that

$$|v_i^\lambda(R)| \leq \sum_{n=1}^\infty \sum_{j,a} r_j^*(a) \cdot e^{-\lambda t} d\pi_{ij}(a,n,R,t) \leq \frac{M}{1-\rho},\ i \in S.$$

A vector $v \in \mathbb{R}^N$ is λ-*superharmonic* if

$$v_i \geq r_i^*(a) + \sum_j p_{ij}^*(a) v_j,\ (i,a) \in S \times A,\ \text{where } p_{ij}^*(a) := p_{ij}(a) \cdot \int_0^\infty e^{-\lambda t} dF_{ij}(a,t). \quad (9.110)$$

Notice that $\sum_j p_{ij}^*(a) = \sum_j p_{ij}(a) \cdot \int_0^\infty e^{-\lambda t} dF_{ij}(a,t) \leq \rho \cdot \sum_j p_{ij}(a) = \rho < 1,\ (i,a) \in S \times A$.

Theorem 9.58

The value vector is the (componentwise) smallest λ-superharmonic vector.

Proof

Choose $\varepsilon > 0$ arbitrarily. Take policies $R_j,\ j \in S$, such that $v^\lambda_j(R_j) \ge v^\lambda_j - \varepsilon$. Let $a_i \in A(i)$ be such that

$$r^*_i(a_i) + \sum_j p^*_{ij}(a_i)v^\lambda_j = max_a\,\{r^*_i(a) + \sum_j p^*_{ij}(a)v^\lambda_j\},\ i \in S. \tag{9.111}$$

We denote by R^* the policy that chooses at $t = 0$ action a_i, given that the state of the system is state i at $t = 0$, and then follows policy R_j if the next state is state j, while the process is considered as starting in state j. Then, we obtain

$$\begin{array}{lllll} v^\lambda_i & \ge & v^\lambda_i(R^*) \;=\; r^*_i(a_i) + \sum_j p^*_{ij}(a_i)v^\lambda_j(R_j) \;\ge\; r^*_i(a_i) + \sum_j p^*_{ij}(a_i)v^\lambda_j - \varepsilon \cdot \sum_j p^*_{ij}(a_i) \\ & \ge & r^*_i(a_i) + \sum_j p^*_{ij}(a_i)v^\lambda_j - \varepsilon \cdot \rho \;\ge\; max_a\,\{r^*_i(a) + \sum_j p^*_{ij}(a)v^\lambda_j\} - \varepsilon. \end{array}$$

Since ε is arbitrarily chosen, it follows that

$$v^\lambda_i \ge max_a\,\{r^*_i(a) + \sum_j p^*_{ij}(a)v^\lambda_j\},\ i \in S, \tag{9.112}$$

i.e. v^λ is λ-superharmonic. Let $R := (\pi^1, \pi^2, \dots)$ be an arbitrary policy. Then, we can write

$$v^\lambda_i(R) = \sum_a \pi^1_{ia} \cdot \Big\{r^*_i(a) + \sum_j p^*_{ij}(a) \cdot \{\, \int_0^\infty e^{-\lambda t} dF_{ij}(a,t)\} \cdot u^\lambda_j(R)\Big\},\ i \in S,$$

where $u^\lambda_j(R)$ represents the expected discounted reward earned from the second decision epoch, given that the state at the second decision epoch is state j. Therefore, $u^\lambda_j(R) \le v^\lambda_j,\ j \in S$. Hence,

$$\begin{array}{lll} v^\lambda_i(R) & \ge & \sum_a \pi^1_{ia} \cdot \{r^*_i(a) + \sum_j p^*_{ij}(a)v^\lambda_j\} \;\le\; \sum_a \pi^1_{ia} \cdot max_a\,\{r^*_i(a) + \sum_j p^*_{ij}(a)v^\lambda_j\} \\ & = & max_a\,\{r^*_i(a) + \sum_j p^*_{ij}(a)v^\lambda_j\},\ i \in S. \end{array}$$

Since R is arbitrarily chosen, we obtain

$$v^\lambda_i \le max_a\,\{r^*_i(a) + \sum_j p^*_{ij}(a)v^\lambda_j\},\ i \in S. \tag{9.113}$$

Combining (9.112) and (9.113) yields

$$v^\lambda_i = max_a\,\{r^*_i(a) + \sum_j p^*_{ij}(a)v^\lambda_j\},\ i \in S. \tag{9.114}$$

Suppose that $v \in \mathbb{R}^N$ is also λ-superharmonic. Let $a_i,\ i \in S$, again satisfy (9.111). Then, we have

$$\begin{array}{lll} v_i - v^\lambda_i(R) & \ge & \{r^*_i(a_i) + \sum_j p^*_{ij}(a_i)v_j\} - \{r^*_i(a_i) + \sum_j p^*_{ij}(a_i)v^\lambda_j\} \\ & = & \sum_j p^*_{ij}(a_i)(v_j - v^\lambda_j\},\ i \in S. \end{array}$$

Let P be the $N \times N$-matrix with elements $p_{ij} := p^*_{ij}(a_i),\ i,j \in S$. Then, we may write in vector notation $v - v^\lambda \ge P(v - v^\lambda) \ge \cdots \ge P^n(v - v^\lambda),\ n \in N$. The matrix P satisfies $\|P\|_\infty = max_i \sum_j p^*_{ij}(a_i) \le \rho < 1$. Consequently, $\lim_{n\to\infty} P^n = 0$, implying $v - v^\lambda \ge \lim_{n\to\infty} P^n(v - v^\lambda) = 0$. This completes the proof that the value vector is the smallest λ-superharmonic vector. □

For any deterministic policy f^∞, let $P(f) := \big(p^*_{ij}(f)\big)$ and let $r^*(f) := \big(r^*(f)\big)$.

Theorem 9.59

$v^\lambda(f^\infty)$ is the unique solution of the equation $r^*(f) + P(f)x = x$ and $v^\lambda(f^\infty) = \{I - P(f)\}^{-1}r^*(f)$.

Proof

The matrix $P(f)$ satisfies $\|P(f)\|_\infty = max_i \sum_j p^*_{ij}\big(f(i)\big) \le \rho < 1$. Hence, $\lim_{n\to\infty} P^n(f) = 0$, implying $\{I - P(f)\}$ is nonsingular. Therefore, it is sufficient to show that $v^\lambda(f^\infty) = r^*(f) + P(f)v^\lambda(f^\infty)$.

Notice that

$\mathbb{E}_{i,f^\infty}\ \{r_{X_1}(Y_1) + s_{X_1}(Y_1) \cdot \int_0^{T_1} e^{-\lambda t}dt\} = r_i(f) + s_i(f) \cdot \sum_j p_{ij}(f) \cdot \int_0^\infty \{\int_0^t e^{-\lambda s}ds\} \cdot dF_{ij}(f,t) = r_i^*(f)$.

Hence, by conditioning on X_2 and T_1,

$$\mathbb{E}_{i,f^\infty}\left\{\sum_{n=2}^\infty e^{-\lambda(T_1+T_2+\cdots+T_{n-1})} \cdot \{r_{X_n}(Y_n) + s_{X_n}(Y_n) \cdot \int_0^{T_n} e^{-\lambda t}dt\}\right\} =$$
$$\sum_j p_{ij}(f) \cdot \{\int_0^\infty e^{-\lambda t}dF_{ij}(f,t)\} \cdot v_j^\lambda(f^\infty) = \sum_j p_{ij}^*(f)v_j^\lambda(f^\infty).$$

In this way, we can write

$$\begin{aligned} v_i^\lambda(f^\infty) &= \mathbb{E}_{i,f^\infty}\left\{\sum_{n=1}^\infty e^{-\lambda(T_1+T_2+\cdots+T_{n-1})} \cdot \{r_{X_n}(Y_n) + s_{X_n}(Y_n) \cdot \int_0^{T_n} e^{-\lambda t}dt\}\right\} \\ &= r_i^*(f) + \sum_j p_{ij}^*(f)v_j^\lambda(f^\infty),\ i \in S, \end{aligned}$$

implying that $v^\lambda(f^\infty)$ satisfies $r^*(f) + P(f)x = x$. □

We have seen that the optimality equation for a discounted semi-Markov decision problem has the form (9.114). In fact it can be shown similarly to the discrete-time model with discounted rewards that the operator $U : R^N \to R^N$, defined by

$$(Ux)_i := max_a\,\{r_i^*(a) + \sum_j p_{ij}^*(a)xa_j\},\ i \in S, \tag{9.115}$$

is a contraction (for details see Denardo [56]). Consequently, we summarize the results in the following theorem.

Theorem 9.60
(1) The operator U, defined in (9.115), is a contraction.
(2) The value vector v^λ is the unique solution of the optimality equation $Ux = x$.
(3) If $f^\infty \in C(D)$ satisfies $r^*(f) + P(f)v^\lambda = v^\lambda$, then f^∞ is an optimal policy.

The data of a discounted SMDP are given by the state space S, the action sets $A(i),\ i \in S$, the transition probabilities $p_{ij}(a),\ (i,a) \in S \times A,\ j \in S$, the immediate rewards $r_i(a),\ (i,a) \in S \times A$, the reward rates $s_i(a),\ (i,a) \in S \times A$ and the sojourn time distribution functions $F_{ij}(a,t)$ for all $(i,a) \in S \times A,\ j \in S$. From these quantities we compute the transition numbers $p_{ij}^*(a),\ (i,a) \in S \times A,\ j \in S$, and the rewards $r_i^*(a),\ (i,a) \in S \times A$.

Example 9.22 (continued)
We assume that $\lambda = 0.1$. We first compute the numbers $\int_0^\infty e^{-\lambda t}dF_i(a,t)$ for all $i \in S$ and $a \in A(i)$ (notice that we write $dF_i(a,t)$ because $F_{ij}(a,t)$ is independent of j).

$\int_0^\infty e^{-\lambda t}dF_1(1,t) = \frac{1}{2} \cdot \int_0^2 e^{-0.1t}dt = \frac{1}{2} \cdot \frac{1}{0.1} \cdot \left(1 - e^{-0.2}\right) = 0.906.$

$\int_0^\infty e^{-\lambda t}dF_1(2,t) = \frac{1}{4} \cdot \int_0^4 e^{-0.1t}dt = \frac{1}{4} \cdot \frac{1}{0.1} \cdot \left(1 - e^{-0.4}\right) = 0.824.$

$\int_0^\infty e^{-\lambda t}dF_2(1,t) = \frac{1}{3} \cdot \int_0^3 e^{-0.1t}dt = \frac{1}{3} \cdot \frac{1}{0.1} \cdot \left(1 - e^{-0.3}\right) = 0.864.$

Hence,

$p_{11}^*(1) = 0.453,\ p_{12}^*(1) = 0.453;\ p_{11}^*(2) = 0,\ p_{12}^*(2) = 0.824;\ p_{21}^*(1) = 0.086,\ p_{22}^*(1) = 0.778.$

For the rewards $r_i^*(a) = r_i(a) + s_i(a) \cdot \int_0^\infty \{\int_0^t e^{-\lambda s}ds\}dF_j(a,t)$, we obtain

$r_1^*(1) = 0 + 5 \cdot \int_0^\infty \{\int_0^t e^{-\lambda s}ds\}dF_1(1,t) = 5 \cdot \frac{1}{0.2}\int_0^2 \left(1 - e^{-0.1t}\right)dt = 4.683.$

$r_1^*(2) = -1 + 10 \cdot \int_0^\infty \{\int_0^t e^{-\lambda s}ds\}dF_1(2,t) = -1 + 10 \cdot \frac{1}{0.4}\int_0^4 \left(1 - e^{-0.1t}\right)dt = 16.580.$

$r_2^*(1) = 0 - 1 \cdot \int_0^\infty \{\int_0^t e^{-\lambda s}ds\}dF_2(1,t) = - \cdot \frac{1}{0.3}\int_0^3 \left(1 - e^{-0.1t}\right)dt = -1.361.$

Let f_1^∞ and f_2^∞ be the deterministic policies with $f_1(1) = 1$ and $f_2(1) = 2$. Then, we have

$$P(f_1) = \begin{pmatrix} 0.453 & 0.453 \\ 0.086 & 0.778 \end{pmatrix},\ r^*(f_1) = \begin{pmatrix} 4.683 \\ -1.361 \end{pmatrix},\ P(f_2) = \begin{pmatrix} 0 & 0.824 \\ 0.086 & 0.778 \end{pmatrix},\ r^*(f_2) = \begin{pmatrix} 16.580 \\ -1.361 \end{pmatrix}.$$

$$v^\lambda(f_1^\infty) = \{I - P(f_1)\}^{-1} r^*(f_1) = \begin{pmatrix} 5.141 \\ -4.139 \end{pmatrix} \text{ and } v^\lambda(f_2^\infty) = \{I - P(f_2)\}^{-1} r^*(f_2) = \begin{pmatrix} 16.934 \\ -1.361 \end{pmatrix}.$$

The optimality equation is:

$v_1^\lambda = max\{4.683 + 0.453 \cdot v_1^\lambda + 0.453 \cdot v_2^\lambda\}$ and $v_2^\lambda = -1.361 + 0.086 \cdot v_1^\lambda + 0.778 \cdot v_2^\lambda$ with solution $v_1^\lambda = 16.934$ and $v_2^\lambda = 0.429$.

The methods of policy iteration, linear programming, value iteration and modified policy iteration for discounted MDPs can be applied directly to the discounted SMDPs when we replace $\alpha\, p_{ij}(a)$ and $r_i(a)$ by $p^*_{ij}(a)$ and $r^*_i(a)$ for all $(i,a) \in S \times A$ and $j \in S$. We shall make this statement explicit for the linear programming method. Since, by Theorem 9.60, v^λ is the smallest superharmonic vector, v^λ is the unique optimal solution of the linear program

$$min\{\sum_j \beta_j v_j \mid \sum_{i,a} \{\delta_{ij} - p^*_{ij}(a)\} v_j \ge r^*_i(a),\ (i,a) \in S \times A\}, \tag{9.116}$$

where $\beta_j,\ j \in S$, are arbitrary, but strictly positive, numbers. The dual of (9.116) becomes

$$max\left\{ \sum_{i,a} r^*_i(a) x_i(a) \;\middle|\; \begin{array}{rcl} \sum_{i,a} \{\delta_{ij} - p^*_{ij}(a)\} x_i(a) & = & \beta_j,\ j \in S \\ x_i(a) & \ge & 0,\ (i,a) \in S \times A \end{array} \right\}. \tag{9.117}$$

Theorem 9.61
Let x^* be an optimal solution of the linear program (9.117). Then, any deterministic policy f_*^∞ such that $x^*_i\big(f_*(i)\big) > 0,\ i \in S$, is an optimal policy.

Proof
Since v^λ is the unique optimal solution of the linear program (9.116), the dual program (9.117) has also a finite optimal solution, say x^*. Notice that

$$\textstyle\sum_a x^*_j(a) = \beta_j + \sum_{i,a} p^*_{ij}(a) x^*_i(a) \ge \beta_j > 0,\ j \in S.$$

Hence, the policy f_*^∞ is well-defined. Further, from the complementary slackness property of linear programming, we obtain

$$x^*_i\big(f_*(i)\big) \cdot \big\{ \textstyle\sum_j \{\delta_{ij} - p^*_{ij}(f_*)\} v^\lambda_j - r^*_i(f_*) \big\} = 0,\ i \in S,$$

implying $\sum_j \{\delta_{ij} - p^*_{ij}(f_*)\} v^\lambda_j = r^*_i(f_*),\ i \in S$, i.e. $v^\lambda(f_*^\infty) = v^\lambda$.

Algorithm 9.17 *Linear programming algorithm for a discounted SMDP*
Input: Instance of a discounted SMDP.
Output: The value vector v^λ and an optimal policy f_*^∞.

```
1. for each $(i,a) \in S \times A$ do
       for each $j \in S$ do $p^*_{ij}(a) := p_{ij}(a) \cdot \int_0^\infty e^{-\lambda t} dF_{ij}(a,t)$.
2. for each $(i,a) \in S \times A$ do $r^*_i(a) := r_i(a) + s_i(a) \cdot \sum_j p_{ij}(a) \int_0^\infty \{ \int_0^t e^-\lambda s ds \} dF_{ij}(a,t)$.
3. Select $\beta_j \in \mathbb{R}^N$ such that $\beta_j > 0,\ j \in S$.
```

4. Compute optimal solutions v^* and x^* of the dual pair of linear programs

$$min\{\, \textstyle\sum_j \beta_j v_j \;|\, \sum_{i,a} \{\delta_{ij} - p^*_{ij}(a)\} v_j \geq r^*_i(a),\ (i,a) \in S \times A\}$$

   and

$$max \left\{ \textstyle\sum_{i,a} r^*_i(a) x_i(a) \;\middle|\; \begin{array}{rcl} \sum_{i,a} \{\delta_{ij} - p^*_{ij}(a)\} x_i(a) & = & \beta_j,\ j \in S \\ x_i(a) & \geq & 0,\ (i,a) \in S \times A \end{array} \right\}.$$

5. **for all** $i \in S$ **do** select $f_*(i) \in A(i)$ such that $x^*_i\big(f_*(i)\big) > 0$.

6. f^∞_* is an optimal policy and v^* is the value vector v^λ (STOP).

Example 9.22 (continued)

Step 1:

$p^*_{11}(1) = 0.453,\ p^*_{12}(1) = 0.453;\ p^*_{11}(2) = 0,\ p^*_{12}(2) = 0.824;\ p^*_{21}(1) = 0.086,\ p^*_{22}(1) = 0.778.$

Step 2:

$r^*_1(1) = 4.683;\ r^*_1(2) = 16.580;\ r^*_2(1) = -1.361.$

Step 3:

Set $\beta_1 := \beta_2 := 0.5$.

Step 4:

The primal and dual linear programs are:

$$min \left\{ 0.5v_1 + 0.5v_2 \;\middle|\; \begin{array}{rcrcr} 0.547v_1 & - & 0.453v_2 & \geq & 4.683 \\ v_1 & - & 0.824v_2 & \geq & 16.580 \\ -0.086v_1 & + & 0.222v_2 & \geq & -1.361 \end{array} \right\} \text{ and}$$

$$max \left\{ \begin{array}{l} 4.683x_1(1) + 16.580x_1(2) \\ 1.361x_2(1) \end{array} \;\middle|\; \begin{array}{rcrcrcl} 0.547x_1(1) & + & x_1(2) & - & 0.086x_2(1) & = & 0.5 \\ -0.453x_1(1) & - & 0.824x_1(2) & + & 0.222x_2(1) & = & 0.5 \\ & & & & x_1(1),\ x_1(2),\ x_2(1) & \geq & 0 \end{array} \right\}$$

with optimal solutions $v^*_1 = 16.934,\ v^*_2 = 0.429$ and $x^*_1(1) = 0,\ x^*_1(2) = 1.0189,\ x^*_2(1) = 6.034.$

Step 5:

$f_*(1) = 2,\ f_*(2) = 1.$

Step 6:

f^∞_* is an optimal policy and $v^* = (16.934, 0.429$ is the value vector.

Remark

Consider the MDP model (S, A, p^*, r^*) with the total reward criterion. Denote the total rewards by $v^*(R)$ for policy R. Since $\sum_j p^*_{ij}(a) = \sum_j p_{ij}(a) \cdot \int_0^\infty e^{-\lambda t} dF_{ij}(a,t) \leq \rho \cdot \sum_j p_{ij}(a) = \rho < 1$ for all $(i,a) \in S \times A$, this model is contracting with $\mu := e$ and $\alpha := \rho$. It can also easily be verified that $v^\lambda(\pi^\infty) = v^*(\pi^\infty)$ for every policy $\pi^\infty \in C(S)$. Therefore, the MDP model (S, A, p^*, r^*) with total rewards may be considered as equivalent to the discounted SMDP model.

9.7.5 Average rewards - general case

We can define the average expected reward in two ways. We startn with the most natural definition. Let $Z(t)$ denote the total reward generated by the process up to time t. The first definition of the expected reward, $\chi^1(R)$, is defined by

$$\chi^1(R) := \liminf_{t \to \infty} \mathbb{E}_{i,R} \left\{ \frac{Z(t)}{t} \right\},\ i \in S. \tag{9.118}$$

For the second definition, $\chi^2(R)$, we use the random variables τ_n, where τ_n is the time between decision epoch n and decision epoch $n+1$:

$$\chi^2(R) := \liminf_{M\to\infty} \frac{\mathbb{E}_{i,R}\left\{\sum_{n=1}^M \{r_{X_n}(Y_n) + \tau_n \cdot s_{X_n}(Y_n)\}\right\}}{\mathbb{E}_{i,R}\left\{\sum_{n=1}^M \tau_n\right\}},\ i \in S. \tag{9.119}$$

However, while $\chi^1(R)$ is clearly a more natural criterion, it turns out that it is easier to work with $\chi^2(R)$. Fortunately, it turns out that under certain conditions, both criteria are equal. If a stationary policy is employed, then the process $\{X(t),\ t \geq 0\}$ is a semi-Markov process, where $X(t)$ represents the state at time t. Roughly speaking, for any stationary policy π^∞, a sufficient condition for $\chi^1(\pi^\infty) = \chi^2(\pi^\infty)$ is that the resultant semi-Markov process $\{X(t),\ t \geq 0\}$ is a regenerative process with finite expected cycle length. Let

$$T := min\,\{T > 0 \mid X(t) = i,\ X(0) = i\} \text{ and } N := min\,\{N \geq 1 \mid X_{n+1} = i,\ X_1 = i\}. \tag{9.120}$$

Hence, T is the time of the first return to state i, and N is the number of transitions that it takes before this occurs. We suppress in the notation T and N the dependency of state i.

Lemma 9.50
If $\mathbb{E}_{\pi^\infty}\{T\} < \infty$, *then* $\mathbb{E}_{\pi^\infty}\{N\} < \infty$ *and* $T = \sum_{n=1}^N \tau_n$.

Proof
By the definition of T and N, it follows that $T \geq= \sum_{n=1}^N \tau_n$, with equality holding if $\mathbb{E}_{\pi^\infty}\{N\} < \infty$. Let $\delta > 0$ and $\varepsilon > 0$ are such as in Assumption 9.3 and let

$$\overline{\tau}_n := \begin{cases} 0 & \text{if } \tau_n \leq \delta \\ \delta \text{ with probability } \frac{\varepsilon}{1-\sum_j Q_{kj}(a,\delta)} & \text{if } \tau_n > \delta,\ X_n = k,\ Y_n = a \\ 0 \text{ with probability } 1 - \frac{\varepsilon}{1-\sum_j Q_{kj}(a,\delta)} & \text{if } \tau_n > \delta,\ X_n = k,\ Y_n = a \end{cases}$$

If $X_n = k$, then $\mathbb{P}_{\pi^\infty}\{\tau_n > \delta\} = 1 - \sum_j Q_{kj}(\pi,\delta)$. Further, $\overline{\tau}_n,\ n = 1,2,\dots$ are independent and identically distributed with $\mathbb{P}_{\pi^\infty}\{\overline{\tau}_n = \delta\} = \mathbb{P}_{\pi^\infty}\{\tau_n > \delta\} \cdot \frac{\varepsilon}{1-\sum_j Q_{kj}(\pi,\delta)} = \varepsilon = 1 - \mathbb{P}_{\pi^\infty}\{\overline{\tau} = 0\}$.

Now, from Walds equation (see [236]), it follows that if $\mathbb{E}_{\pi^\infty}\{N\} = \infty$, then $\mathbb{E}_{\pi^\infty}\{\sum_{n=1}^N \overline{\tau}_n\} = \infty$, and hence $\mathbb{E}_{\pi^\infty}\{T\} \geq \mathbb{E}_{\pi^\infty}\{\sum_{n=1}^N \tau_n\} \geq \mathbb{E}_{\pi^\infty}\{\sum_{n=1}^N \overline{\tau}_n\} = \infty$ (since $\tau_n \geq \overline{\tau}_n$ for all n). Therefore, if $\mathbb{E}_{\pi^\infty}\{T\} < \infty$, then $\mathbb{E}_{\pi^\infty}\{N\} < \infty$ and $T = \sum_{n=1}^N \tau_n$. □

Theorem 9.62
If $\mathbb{E}_{\pi^\infty}\{T\} < \infty$, *then* $\chi_i^1(\pi^\infty) = \chi_i^2(\pi^\infty) = \frac{\mathbb{E}_{i,\pi^\infty}\{Z(T)\}}{\mathbb{E}_{i,\pi^\infty}\{T\}},\ i \in S$.

Proof
Take any starting state $i \in S$. Now, it is easily seen that, under a stationary policy, the semi-Markov process $\{X(t),\ t \geq 0\}$ is a regenerative process with regeneration time T. Hence, the process $\{Z(t),\ t \geq 0\}$ may be regarded as a renewal reward process, and thus (by Theorem 3.16 in [236]),

$$\chi_i^1(\pi^\infty) = \lim_{t\to\infty} \mathbb{E}_{i,\pi^\infty}\left\{\frac{Z(T)}{T}\right\} = \frac{\mathbb{E}_{i,\pi^\infty}\{Z(T)\}}{\mathbb{E}_{i,\pi^\infty}\{T\}}. \tag{9.121}$$

It is also easy to see that $\{X_n,\ n = 1,2,\dots\}$ is a discrete time regenerative process with regeneration time N. Hence, by regarding $Z_1 + Z_2 + \cdots + Z_N$ as the reward during the first cycle of this process, it follows (by Lemma 9.36 and by Theorem 3.16 in [236]), that

$$\lim_{M\to\infty} \mathbb{E}_{i,\pi^\infty}\left\{\frac{1}{M}\sum_{n=1}^M Z_n\right\} = \frac{\mathbb{E}_{i,\pi^\infty}\{\sum_{n=1}^N Z_n)\}}{\mathbb{E}_{i,\pi^\infty}\{N\}}. \tag{9.122}$$

However, if we regard $\tau_1 + \tau_2 + \cdots + \tau_N$ as the reward during the first cycle of this process, it follows similarly that

$$\lim_{M\to\infty} \mathbb{E}_{i,\pi^\infty} \left\{ \frac{1}{M} \sum_{n=1}^{M} \tau_n \right\} = \frac{\mathbb{E}_{i,\pi^\infty} \{ \sum_{n=1}^{N} \tau_n)\}}{\mathbb{E}_{i,\pi^\infty} \{N\}}. \tag{9.123}$$

By combining (9.112) and (9.112), we obtain

$$\chi_i^2(\pi^\infty) = \liminf_{M\to\infty} \frac{\mathbb{E}_{i,\pi^\infty} \left\{ \frac{1}{M} \sum_{n=1}^{M} Z_n \right\}}{\mathbb{E}_{i,\pi^\infty} \left\{ \frac{1}{M} \sum_{n=1}^{M} \tau_n \right\}} = \frac{\mathbb{E}_{i,\pi^\infty} \left\{ \sum_{n=1}^{N} Z_n \right\}}{\mathbb{E}_{i,\pi^\infty} \{N\}} \cdot \frac{\mathbb{E}_{i,\pi^\infty} \{N\}}{\mathbb{E}_{i,\pi^\infty} \left\{ \sum_{n=1}^{N} \tau_n \right\}} = \frac{\mathbb{E}_{i,\pi^\infty} \left\{ \sum_{n=1}^{N} Z_n \right\}}{\mathbb{E}_{i,\pi^\infty} \left\{ \sum_{n=1}^{N} \tau_n \right\}}. \tag{9.124}$$

Since $\mathbb{E}_{\pi^\infty}\{T\} < \infty$, we have by Lemma 9.50, $\mathbb{E}_{\pi^\infty}\{N\} < \infty$ and $T = \sum_{n=1}^{N} \tau_n$. Because $\mathbb{E}_{\pi^\infty}\{N\} < \infty$, we also have $\sum_{n=1}^{N} Z_n = Z(T)$. Hence,

$$\chi_i^2(\pi^\infty) = \frac{\mathbb{E}_{i,\pi^\infty} \left\{ Z(T) \right\}}{\mathbb{E}_{i,\pi^\infty} \left\{ T \right\}}. \tag{9.125}$$

The result of the theorem follows from (9.121) and (9.125). □

As a consequence of the above theorem, we shall write $\chi(\pi^\infty)$ for the average reward of a stationary policy π^∞. The *value vector* χ of an undiscounted SMDP is defined by $\chi_i := sup_R\, \chi_i^1(R),\ i \in S$. A policy R_* is an *optimal policy* if $\chi_i^1(R_*) = \chi_i,\ i \in S$.
We now introduce some additional notation by letting

$$\tau_i(a) := \sum_j p_{ij}(a) \cdot \int_0^\infty t\, dF_{ij}(a,t) \text{ and } r_i^*(a) := r_i(a) + \tau_i(a) \cdot s_i(a),\ (i,a) \in S \times A. \tag{9.126}$$

In other words, $\tau_i(a)$ is the expected time until a transition occurs when action a is taken in state i, and $r_i^*(a)$ is the expected reward incurred during such a transition interval. By $T(\pi)$ we denote the diagonal matrix with elements $\{T(\pi)\}_{ij} := \delta_{ij} \cdot \sum_a \tau_i(a)\pi_{ia}$. Throughout the remaining part of this section we shall assume that the following condition holds.

Assumption 9.4
$0 < \int_0^\infty t^2\, dF_{ij}(a,t) < \infty$ for every $i,j \in S$ and $a \in A(i)$.

Let $v_i^t(R)$ denote the expected total reward generated by the process up to time t, given that the system occupies state i at time $t = 0$. The next theorem gives a deep result, which is based on renewal theory, on Laplace-Stieltjes transforms (the discounted rewards $\int_0^\infty e^{\lambda t}\, dv_i^t(R)$ may be viewed as a Laplace-Stieljes transform of the total rewards $v_i^t(R)$ and on Abelian and Tauberian theorems to the behavior of $v_i^t(R)$ as $t \to \infty$. The results are generalizations of results obtained by Blackwell ([29]) and Miller and Veinott ([199]) for the discrete-time model.

Theorem 9.63
Let π^∞ be any stationary policy. Then, we have the following properties.
(1) $\chi(\pi^\infty)$ is the unique solution of the following system of linear equations:

$$\left\{ \begin{array}{rcl} \{I - P(\pi)\}x & = & 0 \\ P^*(\pi)T(\pi)x & = & P^*(\pi)r^*(\pi) \end{array} \right. \tag{9.127}$$

(2) $v^\lambda(\pi^\infty) = \frac{1}{\lambda}\chi(\pi^\infty) + w(\pi^\infty) + \varepsilon(\lambda)$, where $\lim_{\lambda\downarrow 0} \varepsilon(\lambda) = 0$.
(3) $w(\pi^\infty)$ is a solution of the linear system $\{I - P(\pi)\}y = r^*(\pi) - T(\pi)\chi(\pi^\infty)$.
(4) There exists a deterministic Blackwell optimal policy f_*^∞, i.e. $v^\lambda(f_*^\infty) = v^\lambda$ for all $\lambda \in (0, \lambda_0]$.

Proof

For the proof, which is complicated, we refer to Denardo's paper ([60]). □

Lemma 9.51

Suppose that x satisfies $\begin{cases} \{I-P(\pi)\}x & \geq & 0 \\ P^*(\pi)T(\pi)x & \geq & P^*(\pi)r^*(\pi) \end{cases}$ *. Then, $x \geq \chi(\pi^\infty)$.*

Proof

Suppose that the Markov chain $P(\pi)$ and the stationary matrix $P^*(\pi)$ have the standard form (5.4) and (5.7), respectively. Let $R(\pi)$ and $F(\pi)$ be the set of states which are recurrent and transient, respectively, under $P(\pi)$. Further, let i_k be an arbitrary state in the kth ergodic set of the Markov chain $P(\pi)$. Denote the vector $\{I-P(\pi)\}x$ by a. Then, $a \geq 0$ and $P^*(\pi)a = 0$, implying $a_i = 0$ for all $i \in R(\pi)$, i.e. $x_i = \{P(\pi)x\}_i,\ i \in R(\pi)$. Consequently, $x_i = \{P^*(\pi)x\}_i,\ i \in R(\pi)$. Hence, $x_i = x_{i_k}$ if i belongs to the kth ergodic set of $P(\pi)$. Therefore, we can write $x_i \geq \frac{\{P^*(\pi)r^*(\pi)\}_i}{\{P^*(\pi)T(\pi)e\}_i}$ for all $i \in R(\pi)$.

By (9.127) we have with the same arguments, $\chi_i(\pi^\infty) = \frac{\{P^*(\pi)r^*(\pi)\}_i}{\{P^*(\pi)T(\pi)e\}_i},\ i \in R(\pi)$, implying $x_i \geq \chi_i(\pi^\infty)$ for all $i \in R(\pi)$. Let x_F be the vector with as components the transient states of x. Then, because $x \geq P(\pi)x$, we can write $x_F \geq \sum_{k=1}^m x_{i_k} \cdot A_k(\pi)e + Q(\pi)x_F \geq \sum_{k=1}^m \chi_{i_k}(\pi) \cdot A_k(\pi)e + Q(\pi)x_F$.

Since $\{I-Q(\pi)\}$ is nonsingular and $\{I-Q(\pi)\}^{-1} \geq 0$, we obtain $x_F \geq \sum_{k=1}^m \chi_{i_k}(\pi^\infty)\cdot\{I-Q(\pi)\}^{-1}A_k(\pi)e$.

With the same arguments, we obtain by (9.127), $\chi_F(\pi^\infty) = \sum_{k=1}^m \chi_{i_k}(\pi^\infty)\cdot\{I-Q(\pi)\}^{-1}A_k(\pi)e$.

Therefore, $x_F \geq \chi_F(\pi^\infty)$, completing the proof that $x \geq \chi(\pi^\infty)$. □

Lemma 9.52

$\liminf_{\lambda\downarrow 0} \lambda \cdot v_i^\lambda(R) \geq \chi_i^1(R)$ *for every $i \in S$ and every policy R.*

Proof

Since $v_i^\lambda(R) = \int_0^\infty e^{\lambda t}\, dv_i^t(R)$, the proof follows from an Abelian theorem (see Widder [[334]). □

Theorem 9.64

Any deterministic Blackwell optimal policy f_0^∞ is also an average optimal policy.

Proof

Let f_0^∞ be a Blackwell optimal policy. Take any arbitrary policy R. Then, Lemma 9.52 and Theorem 9.63 part (2) imply $\chi_i^1(R) \leq \liminf_{\lambda\downarrow 0} \lambda \cdot v_i^\lambda(R) \leq \liminf_{\lambda\downarrow 0} \lambda \cdot v_i^\lambda(f_0^\infty) = \chi_i^1(f_0^\infty),\ i \in S$. Consequently, $\chi_i^1(f_0^\infty) = \chi_i^1,\ i \in S$. i.e. f_0^∞ is an average optimal policy. □

From Theorem 9.64 it follows that for the determination of an average optimal policy, we may restrict ourselves to the deterministic policies. Consider a deterministic policy f^∞. Then, (9.127) implies that $\chi_i^1(f^\infty)$ depends on the rewards $r_i^*(a)$, the transition probabilities $p_{ij}(a)$ and the transition times $\tau_i(a)$. Hence, it is sufficient to know the transition times $\tau_i(a)$ instead of explicit knowledge about the probability distributions $F_{ij}(a,t)$. By the same argument, we may assume

$$F_{ij}(a,t) = \begin{cases} 0 & \text{if } t < \tau_i(a) \\ 1 & \text{if } t \geq \tau_i(a) \end{cases} \quad \text{for all } i,j \in S \text{ and all } a \in A(i) \tag{9.128}$$

A vector $v \in \mathbb{R}^N$ is called *average superharmonic* if there exists a vector w such that the pair (v,w) satisfies the following system of inequalities

$$\begin{cases} v_i & & \geq & \sum_j p_{ij}v_j & \text{for every } (i,a) \in S \times A \\ \tau_i(a)v_i & + \quad w_i & \geq & r_i^*(a) + \sum_j p_{ij}w_j & \text{for every } (i,a) \in S \times A \end{cases} \tag{9.129}$$

Theorem 9.65

The value vector χ is the (componentwise) smallest average superharmonic vector.

Proof

Let f_0^∞ be any deterministic Blackwell optimal policy (f_0^∞ exists by Theorem 9.63 part (4)), i.e. for every $\lambda \in (0,\lambda_0]$, $v^\lambda(f_0^\infty) = v^\lambda$. Since v^λ is λ-superharmonic, we have for all $(i,j) \in S \times A$ and all $\lambda \in (0,\lambda_0]$
$v_i^\lambda(f_0^\infty) \geq r_i(a) + s_i(a) \cdot \sum_j p_{ij}(a) \int_0^\infty \{ \int_0^t e^{-\lambda s}\, ds\} dF_{ij}(a,t) + \sum_j p_{ij}(a) \cdot \{ \int_0^\infty e^{-\lambda t} dF_{ij}(a,t)\} v_j^\lambda(f_0^\infty)$.

Since we may assume, by formula (9.128), that $F_{ij}(a,t) = \begin{cases} 0 & \text{if } t < \tau_i(a) \\ 1 & \text{if } t \geq \tau_i(a) \end{cases}$ for all $i,j \in S$ and all $a \in A(i)$, we obtain $\int_0^\infty \{ \int_0^t e^{-\lambda s}\, ds\} dF_{ij}(a,t) = \int_0^{\tau_i(a)} e^{-\lambda s}\, ds = \frac{1}{\lambda} \cdot \big(1 - e^{-\lambda \tau_i(a)}\big)$ and $\int_0^\infty e^{-\lambda t} dF_{ij}(a,t) = e^{-\lambda \tau_i(a)}$.
Therefore, we have for all $(i,j) \in S \times A$ and all $\lambda \in (0,\lambda_0]$

$$v_i^\lambda(f_0^\infty) \geq r_i(a) + s_i(a) \cdot \tfrac{1}{\lambda} \cdot \{1 - e^{-\lambda \tau_i(a)}\} + e^{-\lambda \tau_i(a)} \cdot \textstyle\sum_j p_{ij}(a) v_j^\lambda(f_0^\infty).$$

Using the expansion $e^{-\lambda \tau_i(a)} = \sum_{n=1}^\infty \frac{(-\lambda \tau_i(a))^n}{n!}$, we obtain for all $(i,j) \in S \times A$ and all $\lambda \in (0,\lambda_0]$

$$\begin{array}{rcl} v_i^\lambda(f_0^\infty) & \geq & r_i(a) + s_i(a) \cdot \tau_i(a) + \{1 - \lambda \tau_i(a))\} \cdot \sum_j p_{ij}(a) v_j^\lambda(f_0^\infty) + o(\lambda) \\ & = & r_i^*(a) + \sum_j p_{ij}(a) v_j^\lambda(f_0^\infty) - \lambda \tau_i(a) \cdot \sum_j p_{ij}(a) v_j^\lambda(f_0^\infty) + o(\lambda), \end{array}$$

where a function $h(\lambda) = o(\lambda)$ if $\lim_{\lambda \to \infty} \frac{h(\lambda}{\lambda} = 0$. Using Theorem 9.63 part (2), we can write

$$\begin{array}{l} \frac{1}{\lambda}\chi_i(f_0^\infty) + w_i(f_0^\infty) + \varepsilon(\lambda) \geq r_i^*(a) + \sum_j p_{ij}(a)\{\frac{1}{\lambda}\chi_j(f_0^\infty) + w_j(f_0^\infty) + \varepsilon(\lambda)\} \\ \qquad\qquad - \lambda \tau_i(a) \cdot \sum_j p_{ij}(a)\{\frac{1}{\lambda}\chi_j(f_0^\infty) + w_j(f_0^\infty) + \varepsilon(\lambda)\} + o(\lambda) \end{array}$$

for all $(i,j) \in S \times A$ and all $\lambda \in (0,\lambda_0]$. Hence, since $\chi(f_0^\infty) = \chi$ for every $\lambda \in (0,\lambda_0]$,

$$\tfrac{1}{\lambda}\{\chi_i - \textstyle\sum_j p_{ij}(a)\chi_j\} \geq r_i^*(a) - \{w_i(f_0^\infty) - \sum_j p_{ij}(a) w_j(f_0^\infty)\} - \tau_i(a) \cdot \sum_j p_{ij}(a)\chi_j + \varepsilon(\lambda)$$

for all $(i,j) \in S \times A$. Therefore,

$$\chi_i \geq \sum_j p_{ij}(a)\chi_j \text{ for all } (i,a) \in S \times A \tag{9.130}$$

and $w_i(f_0^\infty)\} \geq r_i^*(a) + \sum_j p_{ij}(a) w_j(f_0^\infty) - \tau_i(a) \cdot \chi_i,\ i \in S,\ a \in A(i,\chi) := \{a \in A(i) \mid \chi_i = \sum_j p_{ij}(a)\chi_j\}$.
Similarly is in Theorem 5.17 we can prove that

$$\tau_i(a) \cdot \chi_i + w_i \geq r_i^*(a) + \sum_j p_{ij}(a) w_j \text{ for every } (i,a) \in S \times A, \tag{9.131}$$

where $w_i := w_i(f_0^\infty) - M \cdot \chi_i$, and $M := min\left\{ \frac{\tau_i(a)\cdot\chi_i - r_i^*(a) + w_i(f_0^\infty) - \sum_j p_{ij}(a) w_j(f_0^\infty)}{\chi_i - \sum_j p_{ij}(a)\chi_j} \,\middle|\, a \in A^*(i),\ i \in S \right\}$
with $A^*(i) := \{a \in A(i) \mid \tau_i(a) \cdot \chi_i - r_i^*(a) + w_i(f_0^\infty) - \sum_j p_{ij}(a) w_j(f_0^\infty) < 0\},\ i \in S$.

If $A^*(i) = \emptyset$, then we set $w_i := w_i(f_0^\infty)$. Consequently, (9.130) and (9.131) imply that the value vector χ is average superharmonic.

Suppose that v is also an average superharmonic vector with corresponding vector w. Then, $\{I - P(f_0)\}v \geq 0$ and $T(f_0)v + \{I - P(f_0)\}w \geq r^*(f_0)$. Consequently, $P^*(f_0)T(f_0)v \geq P^*(f_0)r^*(f_0)$.

Since $\{I - P(f_0)\}v \geq 0$ and $P^*(f_0)T(f_0)v \geq P^*(f_0)r^*(f_0)$, we have by Lemma 9.51, $v \geq \chi(f_0^\infty) = \chi$, the last equality because f_0^∞ is an average optimal policy (Theorem 9.64). So, we have shown that the value vector χ is the (componentwise) smallest average superharmonic vector. □

Since the value vector χ is the smallest average superharmonic vector, any optimal solution (v^*, w^*) of the following linear program satisfies $v^* = \chi$.

$$min \left\{ \sum_j \beta_j v_j \;\middle|\; \begin{array}{rcrcl} \sum_j \{\delta_{ij} - p_{ij}(a)\} v_j & & & \geq & 0 \quad \text{for every } (i,a) \in S \times A \\ \tau_i(a) v_i & + & \sum_j \big(\delta_{ij} - p_{ij}(a)\big) u_j & \geq & r_i^*(a) \quad \text{for every } (i,a) \in S \times A \end{array} \right\}, \tag{9.132}$$

where $\beta_j > 0,\ j \in S$, is arbitrarily chosen. The dual linear program of (9.132) is

$$max \left\{ \sum_{(i,a)} r_i^*(a) x_i(a) \;\middle|\; \begin{array}{rcrcl} \sum_{(i,a)} \{\delta_{ij} - p_{ij}(a)\} x_i(a) & & & = & 0,\ j \in S \\ \sum_a \tau_j(a) x_j(a) & + & \sum_{(i,a)} \{\delta_{ij} - p_{ij}(a)\} y_i(a) & = & \beta_j,\ j \in S \\ & & x_i(a), y_i(a) & \geq & 0,\ (i,a) \in S \times A \end{array} \right\}. \tag{9.133}$$

Theorem 9.66

Let (x^*, y^*) be an extreme optimal solution of (9.133). Then, any deterministic policy f_*^∞, where $x_i^*\big(f_*(i)\big) > 0$ if $\sum_a x_i^*(a) > 0$ and $y_i^*\big(f_*(i)\big) > 0$ if $\sum_a x_i^*(a) = 0$ is an average optimal policy.

Proof

Let (v^*, w^*) be an optimal optimal solution of (9.132). Then, $v^* = \chi$. Analogously to the proof of Theorem 5.18 it can be shown that:

1. f_*^∞ is well-defined.
2. $\sum_j \{\delta_{ij} - p_{ij}(f_*)\}\chi_j = 0$ for all $i \in S$.
3. $\tau_i(f_*)\chi_i + \sum_j \{\delta_{ij} - p_{ij}(f_*)\} w_j^* = 0$ for all $i \in S_{x^*} := \{j \mid \sum_a x_j^*(a) > 0\}$.
4. The states of are transient in the Markov chain induced by $P(f_*)$.

From the above properties it follows that $\left\{ \begin{array}{rcl} \{I - P(f_*)\}\chi & = & 0 \\ P^*(f_*)T(f_*)\chi & = & P^*(f_*)r^*(f_*) \end{array} \right.$

Hence, by Theorem 9.63, $\chi(f_*^\infty) = \chi$, i.e. f_*^∞ is an average optimal policy. □

Algorithm 9.18 *Linear programming algorithm for an undiscounted SMDP*

Input: Instance of an undiscounted SMDP.

Output: The value vector χ and an optimal policy f_*^∞.

```
1. Select $\beta_j \in \mathbb{R}^N$ such that $\beta_j > 0,\ j \in S$.

2. Use the simplex method to compute optimal solutions $(v^*, w^*)$ and $(x^*, y^*)$ of the dual pair of linear
   programs (9.132) and (9.133), respectively.

3. for all $i \in S$ do select $f_*(i) \in A(i)$ such that
       begin if $\sum_a x_i^*(a) > 0$ then $x_i^*\big(f_*(i)\big) > 0$
             else $y_i^*\big(f_*(i)\big) > 0$
       end

4. $f_*^\infty$ is an average optimal policy and $v^*$ is the value vector $\chi$ (STOP).
```

Example 9.22 (continued)

In this example we have the following data:

Transition probabilities: $p_{11}(1) = p_{12}(1) = 0.5;\ p_{11}(2) = 0,\ p_{12}(2) = 1;\ p_{21}(1) = 0.1,\ p_{22}(1) = 0.9.$

Immediate rewards: $r_1(1) = 0,\ r_1(2) = -1$ and $r_2(1) = 0$.

Continuous reward rates are: $s_1(1) = 5,\ s_1(2) = 10$ and $s_2(1) = -1$.
Expected sojourn times: $\tau_1(1) = 1,\ \tau_1(2) = 2$ and $\tau_2(1) = 1.5$.
Hence, $r_1^*(1) = 0 + 1 \cdot 5 = 5,\ r_1^*(2) = -1 + 2 \cdot 10 = 19$ and $r_1^*(1) = 0 + 1.5 \cdot (-1) = -1.5$.
The primal and dual linear programs are:

$$\min \left\{ 0.5v_1 - 0.5v_2 \;\middle|\; \begin{array}{rcrcrcrcr} 0.5v_1 & - & 0.5v_2 & & & & & & 0 \\ v_1 & - & v_2 & & & & & & 0 \\ -0.1v_1 & + & 0.1v_2 & & & & & & 0 \\ v_1 & & & + & 0.5w_1 & - & 0.5w_2 & \geq & 5 \\ 2v_1 & & & + & w_1 & - & w_2 & \geq & 19 \\ & & 1.5v_2 & - & 0.1w_1 & + & 0.1w_2 & \geq & -1.5 \end{array} \right\} \text{ and}$$

$$max \left\{ \begin{array}{l} 5x_1(1)\ + \\ 19x_1(2)\ - \\ 1.5x_2(1) \end{array} \;\middle|\; \begin{array}{rcrcrcrcrcrcr} 0.5x_1(1) & + & x_1(2) & - & 0.1x_2(1) & & & & & & & = & 0 \\ -0.5x_1(1) & - & x_1(2) & - & 0.1x_2(1) & & & & & & & = & 0 \\ x_1(1) & + & 2x_1(2) & & & + & 0.5y_1(1) & + & 0.5y_1(2) & - & 0.1y_2(1) & = & 0.5 \\ & & & & 1.5x_2(1) & - & 0.5y_1(1) & - & y_1(2) & + & 0.1y_2(1) & = & 0.5 \\ \multicolumn{11}{r}{x_1(1),\ x_1(2),\ x_2(1),\ y_1(1),\ y_1(2),\ y_2(1)} & \geq & 0 \end{array} \right\}.$$

with optimal solutions $v_1^* = v_2^* = \frac{4}{17};\ w_1^* = \frac{315}{17},\ w_2^* = 0$ and $x_1^*(1) = 0,\ x_1^*(2) = \frac{1}{17},\ x_2^*(1) = \frac{10}{17}$.
$f_*(1) = 2,\ f_*(2) = 1;\ \chi = (\frac{4}{17}, \frac{4}{17})$ is the value vector and f_*^∞ is the optimal policy.

Let τ be such that

$$0 < \tau \leq min_{i,a} \Big\{ \frac{\tau_i(a)}{1 - p_{ii}(a)} \;\Big|\; p_{ii}(a) \neq 1 \Big\}. \tag{9.134}$$

Further, let $\overline{p}_{ij}(a) := \delta_{ij} - \{\delta_{ij} - p_{ij}(a)\} \cdot \frac{\tau}{\tau_i(a)}$ for all $i, j \in S$ and $a \in A(i)$. Therefore, we also have $\delta_{ij} - \overline{p}_{ij}(a) = \{\delta_{ij} - p_{ij}(a)\} \cdot \frac{\tau}{\tau_i(a)}$ for all $i, j \in S$ and $a \in A(i)$. Then, one can easily verify that $\overline{p}_{ij}(a) \geq 0$ for all $i, j \in S$ and $a \in A(i)$, and $\sum_j \overline{p}_{ij}(a) = 1$ for all $(i, a) \in S \times A$. Let $\overline{r}_i(a) := \frac{1}{\tau_i(a)} \cdot r_i^*(a)$ for all $(i, a) \in S \times A$. Then, we obtain for all $(i, a) \in S \times A$ and with $\overline{v}_j := v_j$ and $\overline{w}_j := \frac{1}{\tau} \cdot w_j$ for all $j \in S$

$$\begin{aligned} \textstyle\sum_j \{\delta_{ij} - p_{ij}(a)\} v_j \leq 0 \quad &\Leftrightarrow \quad \textstyle\sum_j \{\delta_{ij} - p_{ij}(a)\} v_j \cdot \frac{\tau}{\tau_i(a)} \geq 0 \\ &\Leftrightarrow \quad \textstyle\sum_j \{\delta_{ij} - \overline{p}_{ij}(a)\} v_j \geq 0 \\ &\Leftrightarrow \quad \textstyle\sum_j \{\delta_{ij} - \overline{p}_{ij}(a)\} \overline{v}_j \geq 0 \end{aligned}$$

and

$$\begin{aligned} \tau_i(a) v_i + \textstyle\sum_j \{\delta_{ij} - p_{ij}(a)\} w_j \leq r_i^*(a) \quad &\Leftrightarrow \quad v_i + \textstyle\sum_j \{\delta_{ij} - p_{ij}(a)\} w_j \cdot \frac{1}{\tau_i(a)} \geq r_i^*(a) \cdot \frac{1}{\tau_i(a)} \\ &\Leftrightarrow \quad v_i + \textstyle\sum_j \{\delta_{ij} - \overline{p}_{ij}(a)\} w_j \cdot \frac{1}{\tau} \geq \overline{r}_i(a) \\ &\Leftrightarrow \quad \overline{v}_i + \textstyle\sum_j \{\delta_{ij} - \overline{p}_{ij}(a)\} \overline{w}_j \geq \overline{r}_i(a) \end{aligned}$$

Hence, the linear program (9.132) is equivalent to the linear program

$$min \left\{ \sum_j \beta_j v_j \;\middle|\; \begin{array}{lcllcl} \sum_j \{\delta_{ij} - \overline{p}_{ij}(a)\} v_j & & & \geq & 0 & \text{for every } (i, a) \in S \times A \\ \qquad\qquad v_i & + & \sum_j \big(\delta_{ij} - \overline{p}_{ij}(a)\big) w_j & \geq & \overline{r}_i(a) & \text{for every } (i, a) \in S \times A \end{array} \right\}, \tag{9.135}$$

which is the linear program (5.28) for the MDP, which is derived from the SMDP by taking transition probabilities $\overline{p}_{ij}(a),\ i, j \in S,\ a \in A(i)$ and immediate rewards $\overline{r}_i(a),\ i \in S,\ a \in A(i)$. Therefore, the SMDP is equivalent to the MDP $(S, A, \overline{p}, \overline{r})$, and also the methods policy iteration, value iteration and modified policy iteration can be used to find an average optimal or ε-optimal policy for the SMDP.

9.7.6 Average rewards - special cases

In this section we present linear programming algorithms for the weak unichain case, the unichain case and the irreducible case. These algorithms are a direct consequence of related results for corresponding special cases of MDPs. Since for these special cases the value vector components $\chi_i,\ i \in S$, are independent of the starting state i, we can use the following dual pair of linear programs:

$$min\Big\{v \ \Big|\ \tau_i(a)v + \sum_j \{\delta_{ij} - p_{ij}(a)\}w_j \geq r_i^*(a) \text{ for every } (i,a) \in S \times A\Big\} \tag{9.136}$$

and

$$max\left\{\sum_{(i,a)} r_i^*(a)x_i(a) \left| \begin{array}{rcl} \sum_{(i,a)}\{\delta_{ij} - p_{ij}(a)\}x_i(a) & = & 0,\ j \in S \\ \sum_{(i,a)} \tau_i(a)x_i(a) & = & 1 \\ x_i(a) & \geq & 0,\ (i,a) \in S \times A \end{array}\right.\right\}. \tag{9.137}$$

Furthermore, the optimality equation becomes

$$y_i = max_a\{r_i^*(a) + p_{ij}(a)y_j - \tau_i(a)x\},\ i \in S. \tag{9.138}$$

Algorithm 9.19 *Linear programming algorithm for a weak unichained undiscounted SMDP*
Input: Instance of a weak unichained undiscounted SMDP.
Output: The value χ and an optimal policy f_*^∞.

```
1. Use the simplex method to compute optimal solutions $(v^*, w^*)$ and $x^*$ of the dual pair of
   linear programs (9.136) and (9.137), respectively.
2. Set $S_{x^*} := \{j \in S \mid \sum_a x_j^*(a) > 0\}$.
3. for all $i \in S_{x^*}$ do select $f_*(i) \in A(i)$ such that $x_i^*\big(f_*(i)\big) > 0$.
4. Set $S_0 := S_{x^*}$.
5. if $S_0 := S$ then begin $v^*$ is the value $\chi$; $f_*^\infty$ is an average optimal policy (STOP) end
   else go to step 6
6. Select a triple $(i, a_i, j)$ such that $i \in S\backslash S_0,\ a_i \in A(i),\ j \in S_0$ and $p_{ij}(a_i) > 0$.
7. Set $f_*(i) := a_i,\ S_0 := S_0 \cup \{i\}$; go to step 5.
```

Algorithm 9.20 *Linear programming algorithm for a unichained undiscounted SMDP*
Input: Instance of a unichained undiscounted SMDP.
Output: The value χ and an optimal policy f_*^∞.

```
1. Use the simplex method to compute optimal solutions $(v^*, w^*)$ and $x^*$ of the dual pair of
   linear programs (9.136) and (9.137), respectively.
```

2. Set $S_{x^*} := \{j \in S \mid \sum_a x^*_j(a) > 0\}$.

3. **for all** $i \in S_{x^*}$ **do** select $f_*(i) \in A(i)$ such that $x^*_i\big(f_*(i)\big) > 0$.

4. **for all** $i \in S \backslash S_{x^*}$ **do** select $f_*(i) \in A(i)$ arbitrarily.

5. v^* is the value χ and f_*^∞ is an average optimal policy (STOP).

Algorithm 9.21 *Linear programming algorithm for an irreducible undiscounted SMDP*
Input: Instance of an irreducible undiscounted SMDP.
Output: The value χ and an optimal policy f_*^∞.

1. Use the simplex method to compute optimal solutions (v^*, w^*) and x^* of the dual pair of linear programs (9.136) and (9.137), respectively.

2. **for all** $i \in S$ **do** select $f_*(i) \in A(i)$ such that $x^*_i\big(f_*(i)\big) > 0$.

3. v^* is the value χ and f_*^∞ is an average optimal policy (STOP).

Example 9.22 (continued)
This example is an irreducible model. The linear programs (9.136) and (9.137) are:

$$min \left\{ v \left| \begin{array}{rcrcrr} v & + & 0.5w_1 & - & 0.5w_2 & \geq 5 \\ 2v & + & w_1 & - & w_2 & \geq 19 \\ 1.5v & - & 0.1w_1 & + & 0.1w_2 & \geq -1.5 \end{array} \right. \right\}$$

and

$$max \left\{ \begin{array}{l} 5x_1(1) + \\ 19x_1(2) - \\ 1.5x_2(1) \end{array} \left| \begin{array}{rcrcrcl} 0.5x_1(1) & + & x_1(2) & - & 0.1x_2(1) & = & 0 \\ -0.5x_1(1) & - & x_1(2) & + & 0.1x_2(1) & = & 0 \\ x_1(1) & + & 2x_1(2) & + & 1.5x_2(1) & = & 1 \\ & & x_1(1), & x_1(2), & x_2(1) & \geq & 0 \end{array} \right. \right\}.$$

with optimal solutions $v^* = \frac{4}{17}$; $w^*_1 = \frac{315}{17}$, $w^*_2 = 0$ and $x^*_1(1) = 0$, $x^*_1(2) = \frac{1}{17}$, $x^*_2(1) = \frac{10}{17}$. $f_*(1) = 2$, $f_*(2) = 1$; $\chi = \left(\frac{4}{17}, \frac{4}{17}\right)$ is the value vector and f_*^∞ is an optimal policy.

One can also use policy iteration and value iteration for SMDPs under the unichain (or irreducible) assumption. The proofs are similar to the proofs for unichained MDPs and are left to the reader. Below, we present the algorithms.

Algorithm 9.22 *Policy iteration algorithm for a unichained undiscounted SMDP*
Input: Instance of a unichained undiscounted SMDP.
Output: The value χ and an optimal policy f_*^∞.

1. Select an arbitrary policy $f^\infty \in C(D)$.

2. Determine the unique solution (x, y) of the system

$$\left\{ \begin{array}{rcl} x \cdot T(f)e + \{I - P(f)\}y & = & r^*(f) \\ y_1 & = & 0 \end{array} \right.$$

```
3. for all $i \in S$ do $B(i,f) := \{a \in A(i) \mid r_i^*(a) + \sum_j p_{ij}(a)y_j > \tau_i(a)x + y_i\}$.
4. if $B(i,f) = \emptyset$ for all $i \in S$ then
        begin $x$ is the value $\chi$; $f_*^\infty := f^\infty$ is an average optimal policy (STOP) end
   else go to step 5.
5. Select $g$ such that for every $i \in S$,
        $r_i^*(g) + \sum_j p_{ij}(g)y_j - \tau_i(g)x = max_a \{r_i^*(a) + \sum_j p_{ij}(a)y_j - \tau_i(a)x\}$.
6. $f := g$; return to step 2.
```

Example 9.22 (continued)

Iteration 1

We start with $f(1) = f(2) = 1$. The linear system becomes:

$$\begin{array}{rcrcrcr} x & + & 0.5y_1 & - & 0.5y_2 & = & 5 \\ 1.5x & - & 0.1y_1 & + & 0.1y_2 & = & -1.5 \\ & & y_1 & & & = & 0 \end{array} \quad \text{with solution } x = -\tfrac{5}{17},\ y_1 = 0,\ y_2 = -\tfrac{180}{17}.$$

$B(1,f) = \{2\},\ B(2,f) = \emptyset;\ g(1) = 2,\ g(2) = 1.$

Iteration 2

$f(1) = 2,\ f(2) = 1$. The linear system becomes:

$$\begin{array}{rcrcrcr} 2x & + & y_1 & - & y_2 & = & 19 \\ 1.5x & - & 0.1y_1 & + & 0.1y_2 & = & -1.5 \\ & & y_1 & & & = & 0 \end{array} \quad \text{with solution } x = \tfrac{4}{17},\ y_1 = 0,\ y_2 = -\tfrac{315}{17}.$$

$B(1,f) = B(2,f) = \emptyset$. The value $\chi = \frac{4}{17}$; $f(1) = 2,\ f(2) = 1$ is an optimal policy.

For the semi-Markov decision model the formulation of a value iteration algorithm is not straightforward. The usual relation is

$$v^{n+1} := Tv^n \text{ where } (Tx)_i := max_a \{r_i^*(a) + \sum_j p_{ij}(a)x_j\},\ i \in S. \tag{9.139}$$

This recursion relation does not take into account the non-identical transition times. However, we can use the data transformation which uses transition probabilities $\overline{p}_{ij}(a)$ and rewards $\overline{r}_i(a)$, defined by

$$\overline{p}_{ij}(a) := \delta_{ij} - \{\delta_{ij} - p_{ij}(a)\} \cdot \frac{\tau}{\tau_i(a)},\ i,j \in S,\ a \in A(i);\ \overline{r}_i(a) := \frac{1}{\tau_i(a)} \cdot r_i^*(a),\ i \in S,\ a \in A(i), \tag{9.140}$$

where τ is defined by (9.134).

Lemma 9.53

For any $f^\infty \in C(D)$ such that $P(f)$ is a unichain Markov chain, we have the following property: y is the stationary distribution of the Markov chain $P(f)$ if and only if x, defined by $x_i := \frac{\tau_i(f)}{\sum_j \tau_j(f)y_j} \cdot y_i,\ i \in S$, is the stationary distribution of the Markov chain $\overline{P}(f)$.

Proof

Let x be the stationary distribution of the Markov chain $\overline{P}(f)$, i.e. $x^T\overline{P}(f) = x^T$ and $x^Te = 1$. Hence,

$$\begin{array}{rcl} x_j & = & \sum_i x_i\,\overline{p}_{ij}(f) = \sum_i x_i\big\{\delta_{ij} - \{\delta_{ij} - p_{ij}(a)\} \cdot \frac{\tau}{\tau_i(a)}\big\} \\ & = & x_j - x_j \cdot \frac{\tau}{\tau_j(a)} + \sum_i x_i\,p_{ij}(a) \cdot \frac{\tau}{\tau_i(a)},\ j \in S, \end{array}$$

implying $\frac{x_j}{\tau_j(f)} = \sum_i \frac{x_i}{\tau_i(f)} \cdot p_{ij}(f),\ j \in S$. Since $P(f)$ is a unichain Markov chain, any solution of $z^T = z^TP(f)$ equals $z = c{\cdot}y$, where y is the stationary distribution of $P(f)$. Therefore, $\frac{x_j}{\tau_j(f)} = c{\cdot}y_i$ for all $j \in S$. Because $\sum_j x_j = 1$, we have $c = \frac{1}{\sum_j \tau_j(f)y_j}$ and consequently, $x_i := \frac{\tau_i(f)}{\sum_j \tau_j(f)y_j} \cdot y_i$ for all $i \in S$.

Conversely, let y be the stationary distribution $P(f)$ and define x by $x_i := \frac{\tau_i(f)}{\sum_k \tau_k(f)y_k} \cdot y_i,\ i \in S$. Then, $\sum_i x_i = 1$ and

$$\begin{array}{rcl} \sum_i x_i\,\overline{p}_{ij}(f) & = & \frac{1}{\sum_k \tau_k(f)y_k} \cdot \sum_i \tau_i(f)y_i \cdot \big\{\delta_{ij} - \{\delta_{ij} - p_{ij}(f)\} \cdot \frac{\tau}{\tau_i(f)}\big\} \\ & = & \frac{1}{\sum_k \tau_k(f)y_k} \cdot \big\{\tau_j(f)y_j - \tau \cdot y_j + \tau \cdot \sum_i y_i\,p_{ij}(f)\big\} \\ & = & \frac{1}{\sum_k \tau_k(f)y_k} \cdot \big\{\tau_j(f)y_j - \tau \cdot \{y_j - \sum_i y_i\,p_{ij}(f)\}\big\} \\ & = & \frac{1}{\sum_k \tau_k(f)y_k} \cdot \tau_j(f)y_j = x_j,\ j \in S, \end{array}$$

implying x is the stationary distribution of the Markov chain $\overline{P}(f)$. □

Corollary 9.15

$\chi(f^\infty) = \overline{\phi}(f^\infty)$, where $\overline{\phi}(f^\infty$ is the average reward in $MDP(S, A, \overline{p}, \overline{r})$.

Proof

Let $\overline{\pi}$ and π be the stationary distributions of the Markov chains $\overline{P}(f)$ and $P(f)$, respectively. Then, $\overline{\phi}(f^\infty) = \sum_i \overline{\pi}_i\,\overline{r}_i = \sum_i \frac{\tau_i(f)}{\sum_k \tau_k(f)\pi_k}\,\pi_i \cdot \frac{r_i^*(f)}{\tau_i(f))} = \frac{\sum_i \pi_i r_i^*(f)}{\sum_k \pi_k\tau_k(f)} = \chi(f^\infty)$, the last equality by (9.127), part (1). □

The Markov chain $\overline{P}(f)$ is unichain and aperiodic. Thus the semi-Markov model can be solved by applying the value iteration algorithm 5.10, which yields the following algorithm.

Algorithm 9.23 *Value iteration algorithm for a unichained undiscounted SMDP*

Input: Instance of a unichained undiscounted SMDP and some scalar $\varepsilon > 0$.

Output: An ε-optimal deterministic policy f^∞ and a $\frac{1}{2}\varepsilon$-approximation of the value χ.

```
1. Select $\tau$ be such that $0 < \tau \le min_{i,a}\Big\{\frac{\tau_i(a)}{1-p_{ii}(a)}\ \Big|\ p_{ii}(a) \neq 1\Big\}$.

2. Select $v \in \mathbb{R}^N$ arbitrarily; $v_N := 0$.

3. Compute $r_i^*(a) := r_i(a) + \tau_i(a) \cdot s_i(a),\ (i,a) \in S \times A$.

4. (a) for all $(i,a) \in S \times A$ do $y_i(a) := \frac{r_i^*(a)}{\tau_i(a)} + \frac{\tau}{\tau_i(a)} \cdot \sum p_{ij}(a)v_j + \{1 - \frac{\tau}{\tau_i(a)}\} \cdot v_i$.
   (b) $g := max_{a \in A(N)}\,y_N(a)$.
   (c) for all $i \in S$ do $w_i := max_{a \in A(i)}\,y_i(a) - g$.
```

```
    (d) Select $f$ such that $w = y(f) - g \cdot e$.
    (e) $u := max_i\,(w_i - v_i)$; $l := min_i\,(w_i - v_i)$.
5. if $u - l \le \varepsilon$ then
       begin $f^\infty$ is an $\varepsilon$-optimal policy; $\frac{1}{2}(u + l) + g$ is an $\frac{1}{2}\varepsilon$-approximation of $\phi_0$ (STOP)
       end
   else begin $v := w$; return to step 4 end.
```

Example 9.25 *The streetwalkers dilemma*
Consider a prostitute and suppose that potential customers arrive in accordance with a Poisson process with rate λ. Each potential customer makes an offer consisting of the pair (i, t_i), where i is the amount of money offered and t_i is the mean time spent to this customer. The successive offers are assumed independent and the offer (i, t_i) occurs with probability $p_i > 0$, where $\sum_{i=1}^N p_i = 1$ (we assume that there are N possible offers of the type (i, t_i)).
If the offer is rejected, then the arrival leaves and the prostitute waits for the next potential customer. If the offer is accepted, then all potential customers, who arrive while the prostitute is busy, are assumed lost. The prostitute dilemma is to choose the customers so as to maximize the long-run return.
The above problem may be viewed as a two action SMDP with $S = \{1, 2, \dots, N\}$, where state i means that the prostitute has received an offer of (i, t_i). Let action 1 be the accept and action 2 the reject action. The other parameters of the process are given by:

$$p_{ij}(1) = p_{ij}(2) = p_j,\ i, j \in S;\ r_i^*(1) = i,\ r_i^*(2) = 0,\ i \in S;\ \tau_i(1) = t_i + \tfrac{1}{\lambda},\ \tau_i(2) = \tfrac{1}{\lambda},\ i \in S.$$

This model is an irreducible SMDP. The optimality equation for this model becomes

$$y_i = max\Big\{i + \sum_{j=1}^N p_j y_j - \big\{t_i + \frac{1}{\lambda}\big\} \cdot x, \sum_{j=1}^N p_j y_j - \frac{1}{\lambda} \cdot x\Big\},\ i \in S. \tag{9.141}$$

We know from the general theory that the x-part of (9.141) is the value χ and that in state i action 1 is optimal if and only if $i + \sum_{j=1}^N p_j y_j - \big\{t_i + \frac{1}{\lambda}\big\} \cdot \chi \ge \sum_{j=1}^N p_j y_j - \frac{1}{\lambda} \cdot x$. Hence the optimal policy f_*^∞ satisfies

$$f_* = 1 \text{ if and only if } \frac{i}{t_i} \ge \chi. \tag{9.142}$$

So, the structure of the policy is determined. Note that $P(f)$ is, independently of the policy f^∞, the fixed matrix P with identical rows $(p_1, p_2, \dots, p_N)$. Consequently, $\pi := (p_1, p_2, \dots, p_N)$ is also the stationary distribution of any $P(f)$. By Theorem 9.63, part (1), $\chi(f^\infty)$ is the unique solution of $\pi^T T(f)x = \pi^T r^*(f)$. For any policy f^∞, let $S_1(f) := \{a \mid f(i) = 1\}$. Then, it is straightforward that

$$\chi(f^\infty) = \frac{\sum_{j \in S_1(f)} p_j \cdot j}{\frac{1}{\lambda} + \sum_{j \in S_1(f)} p_j \cdot t_j}. \tag{9.143}$$

The right hand of (9.143) can be interpreted as the ratio of the expected return and the expected time between arrivals. This is exactly in accordance with the renewal theory. For the value χ this expression has to be maximized over the policies. When the value χ is known, then the optimal policy f_*^∞ follows from (9.142).

Example 9.26 *Post office*

Suppose that letters arrive at a post office in accordance with a Poisson process with rate λ. At any time, the postmaster may, at a cost of K, summon a truck to pick up all letters presently in the post office. We assume that the truck arrives instantaneously. Suppose also that the post office occurs a cost at a rate of c_i when there are i letters waiting to be picked up, where c_i is an nondecreasing function. The problem is to select a policy which minimizes the long-run average cost per unit time.

This problem may be viewed as a two action SMDP, where state i means that there are i letters waiting to be picked up. Action 1 is summon a truck and action 2 dont summon a truck. As state space we take $S = \{1, 2, \dots, N\}$. Note that since a truck would never be summoned if there were no letters in the post office, we need not have a state 0. Further, we assume that there exists a number N such that a truck is always summoned when there are N letters. The other parameters of the problem are for all $i \in S$:

$$p_{i1}(1) = 1;\ \tau_i(1) = \tfrac{1}{\lambda};\ c_i^*(1) = K + \tfrac{c_0}{\lambda};\ p_{i,i+1}(2) = 1;\ \tau_i(2) = \tfrac{1}{\lambda};\ c_i^*(2) = \tfrac{c_i}{\lambda}.$$

This SMDP is obviously a unichained model. Take $\tau = 1$. Then the SMDP model is equivalent to the MDP model with costs $\overline{c}_i(1) = \lambda K + c_0$, $\overline{c}_i(2) = c_i$ and transitions $\overline{p}_{i1}(1) = 1$, $\overline{p}_{i,i+1}(2) = 1$. From Section 6.2, it follows that the optimality equation for this model is

$$x + y_i = min\,\{\lambda K + c_0 + y_1,\ c_i + y_{i+1}\},\ i \in S, \tag{9.144}$$

where $y_{N+1} := \infty$. Further, we have seen that any solution (x^*, y^*) of this equation satisfies $x^* = \chi$, the value of the MDP, and that the policy which chooses the minimal actions is optimal.

Theorem 9.67

Let $(x^* = \chi, y^*)$ be a solution of the optimality equation (9.144). Let the index i_* be such that $i_* = min\{i \mid c_i + y_{i+1}^* > \lambda K + c_0 + y_1^*\}$. Then, the optimal control-limit policy is to summon the truck whenever the number of letters in the post office is at least i_*.

Proof

From the definition of the index i_* it follows that action 1 is optimal in state i_*. Notice that

$$x + y_N^* = \lambda K + c_0 + y_1^* \geq min\,\{\lambda K + c_0 + y_1^*, c_{N-1} + y_N^*\} = x + y_{N-1}^*,$$

implying $y_{N-1}^* \leq y_N^*$. Then, by backward induction and the property that c_i is nondecreasing in i, it follows from (9.144) that the function y_i^* is also nondecreasing in i. Hence, we can write for all $i \geq i_*$: $c_i + y_{i+1}^* \geq c_{i_*} + y_{i_*+1}^* > \lambda K + c_0 + y_1^*$. Further, for $i \leq i_* - 1$, we have: $c_i + y_{i+1}^* \leq c_{i_*-1} + y_{i_*}^* \leq \lambda K + c_0 + y_1^*$, i.e. in these states it is optimal not to summon the truck. □

Consider the optimal control limit policy f_*^∞, as defined in Theorem 9.67. It is easy to see that the stationary matrix $P^*(f_*)$ has identical rows π^* with elements $\pi^*_i = \begin{cases} \frac{1}{i_*}, & 1 \le i \le i_* \\ 0, & i \ge i_* + 1 \end{cases}$

The linear system $P^*(f_*)T(f_*)x = P^*(f_*)r(f_*)$, which has the unique solution $x = \chi$, becomes for this model $\frac{1}{\lambda}x = \sum_{i=1}^{i_*-1} \frac{1}{i_*} \cdot \frac{c_i}{\lambda} + \frac{1}{i_*}\big(K + \frac{c_0}{\lambda}\big)$, implying $\chi = \frac{1}{i_*} \cdot \{\lambda K + \sum_{i=0}^{i_*-1} c_i\}$. Knowing that an optimal control-limit policy exists, the optimal i_* can be found as the value for which the function $h(i) := \frac{1}{i} \cdot \{\lambda K + \sum_{k=1}^{i} c_k\}$ is minimal.

For example, if $c_i = c \cdot i$, we have $h(i) = \frac{\lambda K}{i} + \frac{i-1}{2} \cdot c$, and by treating i as a continuous variable, we obtain by differential calculus that the optimal i_* is one of the two integers adjacent to $\sqrt{\frac{2\lambda K}{c}}$.

Example 9.27 *Optimal sharing of memory between processors*

In computer networks an important problem is the allocation of memory to several types of users. Suppose two processors share a common memory that is able to accommodate a total of M messages. The messages are distinguished by the processor destinations: a message of type k is destined for processor k and arrives according to a Poisson process with rate λ_k ($k = 1$ or $k = 2$). When a message arrives a decision to accept or reject that message must be made. A message that is rejected has no further influence on the system. If a message is accepted it stays in the memory until completion of service. The time required to process a message of type k is exponentially distributed with mean $\frac{1}{\mu_k}$, $k = 1, 2$. The processor k handles only messages of type k and is able to serve only one message at a time.

The measure of system performance is minimizing the average weighted sum of the rejections of the messages 1 and 2, where the respective weights are given by γ_1 and γ_2. Note that the special case $\gamma_1 = \gamma_2 = 1$ is the minimization of the average rejections which is equivalent to the maximization of the average throughput. This sharing problem can be modeled as a semi-Markov decision problem.

A straightforward formulation takes the arrival epochs as the only decision epochs. In such a formulation the determination of the transition probabilities is rather complicated and the vectors $\{p_{ij}(a),\ i \in S\}$, with components $(j, a) \in S \times A$, have many nonzero entries. By the nature of the value iteration algorithm it is computationally burdensome to have many nonzero transition probabilities.

In our specific problem this difficulty can be circumvented by including the service completion epochs as fictitious decision epochs in addition to the real decision epochs, being the arrival epochs of the messages. The fictitious decision at the service completion epochs is to leave the system unchanged. Note that the inclusion of these fictitious decision epochs does not change the Markovian nature of the decision process, since the times between state transitions are exponentially distributed and thus have the memoryless property.

It will appear that the inclusion of fictitious decision epochs simplifies not only the formulation of the value iteration algorithm, but also reduces the computational effort as compared with a straightforward formulation. The inclusion of the service completion epochs as fictitious decision epochs has a consequence that the state space must be enlarged.

We take as state space $S = \{(i_1, i_2, k) \mid i_1, i_2 = 0, 1, \ldots, M;\ i_1 + i_2 \leq M;\ k = 0, 1, 2\}$. State (i_1, i_2, k) with $k = 1$ or $k = 2$ corresponds to the situation in which a message of type k arrives and finds i_1 messages of type 1 and i_2 messages of type 2 being present in the common waiting area. The state $(i_1, i_2, 0)$ corresponds to the situation in which the service of a message is completed and i_1 messages of type 1 and i_2 messages of type 2 are left behind in the common waiting area. For the states (i_1, i_2, k) with $k = 1$ or $k = 2$ the possible actions a are 0 or 1, where $a = 0$ corresponds to rejection and $a = 1$ to acceptance, with the stipulation that action 0 is the only action when $i_1 + i_2 = M$. For the states $(i_1, i_2, 0)$ the only decision $(a = 0)$ is leaving the system unchanged.

Thanks to the fictitious decisions, each transition from a given state is to one of the four neighboring states, corresponding to arrival of a message of type 1 or 2, or completion of message of type 1 or 2. In other words, most of the one-step transition probabilities are zero. Further, the nonzero transition probabilities are easy to specify. To find these probabilities, we use the basic properties of the exponential distribution.

Let $\lambda(i_1, i_2) = \lambda_1 + \lambda_2 + \mu_1\delta(i_1) + \mu_2\delta(i_2)$, where $\delta(x)$ is defined by $\delta(x) := \begin{cases} 0 & \text{if } x = 0 \\ 1 & \text{if } x \geq 1 \end{cases}$.

For action $a = 0$ in any state $s = (i_1, i_2, k)$, we obtain

$$\tau_s(a) := \tfrac{1}{\lambda(i_1,i_2)} \text{ and } p_{ss'} := \begin{cases} \frac{\lambda_1}{\lambda(i_1,i_2)} & = & \lambda_1 \cdot \tau_s(a) & \text{if } s' = (i_1, i_2, 1) \\ \frac{\lambda_2}{\lambda(i_1,i_2)} & = & \lambda_2 \cdot \tau_s(a) & \text{if } s' = (i_1, i_2, 2) \\ \frac{\mu_1 \cdot \delta(i_1)}{\lambda(i_1,i_2)} & = & \mu_1 \cdot \delta(i_1) \cdot \tau_s(a) & \text{if } s' = (i_1 - 1, i_2, 0) \\ \frac{\mu_2 \cdot \delta(i_2)}{\lambda(i_1,i_2)} & = & \mu_2 \cdot \delta(i_2) \cdot \tau_s(a) & \text{if } s' = (i_1, i_2 - 1, 0) \end{cases}.$$

For action $a = 1$ in any state $s = (i_1, i_2, 1)$, we obtain

$$\tau_s(a) := \tfrac{1}{\lambda(i_1+1,i_2)} \text{ and } p_{ss'} := \begin{cases} \frac{\lambda_1}{\lambda(i_1+1,i_2)} & = & \lambda_1 \cdot \tau_s(a) & \text{if } s' = (i_1 + 1, i_2, 1) \\ \frac{\lambda_2}{\lambda(i_1+1,i_2)} & = & \lambda_2 \cdot \tau_s(a) & \text{if } s' = (i_1 + 1, i_2, 2) \\ \frac{\mu_1}{\lambda(i_1+1,i_2)} & = & \mu_1 \cdot \tau_s(a) & \text{if } s' = (i_1, i_2, 0) \\ \frac{\mu_2 \cdot \delta(i_2)}{\lambda(i_1+1,i_2)} & = & \mu_2 \cdot \delta(i_2) \cdot \tau_s(a) & \text{if } s' = (i_1 + 1, i_2 - 1, 0) \end{cases}.$$

For action $a = 2$ in any state $s = (i_1, i_2, 1)$, we obtain

$$\tau_s(a) := \tfrac{1}{\lambda(i_1,i_2+1)} \text{ and } p_{ss'} := \begin{cases} \frac{\lambda_1}{\lambda(i_1,i_2+1)} & = & \lambda_1 \cdot \tau_s(a) & \text{if } s' = (i_1, i_2 + 1, 1) \\ \frac{\lambda_2}{\lambda(i_1,i_2+1)} & = & \lambda_2 \cdot \tau_s(a) & \text{if } s' = (i_1, i_2 + 1, 2) \\ \frac{\mu_1 \cdot \delta(i_1)}{\lambda(i_1,i_2+1)} & = & \mu_1 \cdot \delta(i_1) \cdot \tau_s(a) & \text{if } s' = (i_1 - 1, i_2 + 1, 0) \\ \frac{\mu_2}{\lambda(i_1,i_2+1)} & = & \mu_2 \cdot \tau_s(a) & \text{if } s' = (i_1, i_2, 0) \end{cases}.$$

For the costs $c_s^*(a)$, we have $c_s^*(a) := \begin{cases} \gamma_1 & \text{if } s = (i_1, i_2, 1) \text{ and } a = 0 \\ \gamma_2 & \text{if } s = (i_1, i_2, 2) \text{ and } a = 0 \\ 0 & \text{otherwise} \end{cases}$.

Now, having specified the basic elements of this semi-Markov decision model, we are in a position to formulate the value iteration algorithm for the computation of an ε-optimal policy. In the data transformation, we take $\tau := \frac{1}{\lambda_1+\lambda_2+\mu_1+\mu_2}$. The value iteration scheme becomes quite simple.

For the states $(i_1, i_2, 0)$ we have

$$v^{n+1}_{(i_1,i_2,0)} = \tau\lambda_1 v^n_{(i_1,i_2,1)} + \tau\lambda_2 v^n_{(i_1,i_2,2)} + \tau\mu_1 v^n_{(i_1-1,i_2,0)} + \tau\mu_2 v^n_{(i_1,i_2-1,0)} + \{1-\tau\lambda(i_1,i_2)\}\, v^n_{(i_1,i_2,0)},$$

with the convention that $v^n_{(i_1,i_2,0)} = 0$ when $i_1 = -1$ or $i_2 = -1$.

For the states $(i_1, i_2, 1)$ we have

$$\begin{aligned} v^{n+1}_{(i_1,i_2,1)} \;=\;& min\{\big[\gamma_1\lambda(i_1,i_2) + \tau\lambda_1 v^n_{(i_1,i_2,1)} + \tau\lambda_2 v^n_{(i_1,i_2,2)} + \tau\mu_1 v^n_{(i_1-1,i_2,0)} + \tau\mu_2 v^n_{(i_1,i_2-1,0)} + \\ & \{1-\tau\lambda(i_1,i_2)\}\, v^n_{(i_1,i_2,1)}\big], \big[\tau\lambda_1 v^n_{(i_1+1,i_2,1)} + \tau\lambda_2 v^n_{(i_1+1,i_2,2)} + \tau\mu_1 v^n_{(i_1,i_2,0)} + \\ & \tau\mu_2 v^n_{(i_1+1,i_2-1,0)} + \{1-\tau\lambda(i_1+1,i_2)\}\, v^n_{(i_1,i_2,1)}\big]\}, \end{aligned}$$

with the convention that $v^n_{(i_1,i_2,1)} = \infty$ when $i_1 + i_2 = M + 1$.

For the states $(i_1, i_2, 2)$ we have

$$\begin{aligned} v^{n+1}_{(i_1,i_2,2)} \;=\;& min\{\big[\gamma_2\lambda(i_1,i_2) + \tau\lambda_1 v^n_{(i_1,i_2,1)} + \tau\lambda_2 v^n_{(i_1,i_2,2)} + \tau\mu_1 v^n_{(i_1-1,i_2,0)} + \tau\mu_2 v^n_{(i_1,i_2-1,0)} + \\ & \{1-\tau\lambda(i_1,i_2)\}\, v^n_{(i_1,i_2,2)}\big], \big[\tau\lambda_1 v^n_{(i_1,i_2+1,1)} + \tau\lambda_2 v^n_{(i_1,i_2+1,2)} + \tau\mu_1 v^n_{(i_1-1,i_2+1,0)} + \\ & \tau\mu_2 v^n_{(i_1,i_2,0)} + \{1-\tau\lambda(i_1,i_2+1)\}\, v^n_{(i_1,i_2,2)}\big]\}, \end{aligned}$$

with the convention that $v^n_{(i_1,i_2,1)} = \infty$ when $i_1 + i_2 = M + 1$.

The value iteration algorithm with the fictitious decision epoch requires the extra states $(i_1, i_2, 0)$. However the number of additions and multiplications per iteration is of the order M^2 rather than the order M^4 as in a straightforward value iteration algorithm.

Numerical investigations (see [288] p. 229) indicate that for $\gamma_1 = \gamma_2 = 1$ and $\mu_1 = \mu_2$ the optimal sharing rule has the intuitively reasonable property that the acceptance of a message of type 1 in state (i_1, i_2) implies the acceptance of a message of type 1 in state $(i_1 - 1, i_2)$; similarly, the acceptance of a message of type 2 in state (i_1, i_2) implies the acceptance of a message of type 2 in state $(i_1, i_2 - 1)$. A control rule of this type is characterized by two nonincreasing sequences $a_0 \geq a_1 \geq \cdots \geq a_{M-1}$ and $b_0 \geq b_1 \geq \cdots \geq b_{M-1}$. A message of type 1 finding upon arrival (i_1, i_2) as the state of the system is accepted only when $i_1 < a_{i_2}$ and $i_1 + i_2 \leq M - 1$. Similarly, a message of type 2 finding upon arrival (i_1, i_2) as the state of the system is accepted only when $i_2 < b_{i_1}$ and $i_1 + i_2 \leq M - 1$.

In a numerical example with $M = 15,\ \lambda_1 = 1.2,\ \lambda_2 = 1,\ \mu_1 = \mu_2 = 1$ and $\gamma_1 = \gamma_2 = 1$, we find $a_0 = a_1 = 11, a_2 = a_3 = 10,\ a_4 = 9,\ a_5 = a_6 = 8,\ a_7 = a_8 = 7,\ a_9 = 6,\ a_{10} = 5,\ a_{11} = 4,$ $a_{12} = 3,\ a_{13} = 2,\ a_{14} = 1$ and $b_0 = b_1 = 12,\ b_2 = b_3 = 11,\ b_4 = b_5 = 10,\ b_6 = 9,\ b_7 = 8,\ b_8 = 7,$ $b_9 = 6,\ b_{10} = 5,\ b_{11} = 4,\ b_{12} = 3,\ b_{13} = 2,\ b_{14} = 1.$ The minimal average lost is 0.348.

A challenging open problem is to find a theoretical proof that there exists an optimal policy with this structure.

Example 9.28 *Optimal control of a service system*

A service system has s identical channels available for providing service, where the number of channels in operation can be controlled by turning channels *on* or *off*. For example, the service channels could be checkouts in a supermarket or production machines in a factory.

Requests for service are sent to the service facility according to a Poisson process with rate λ. Each arriving request is allowed to enter the system and waits in line until an operating channel is provided. The service time of each request is exponentially distributed with mean $\frac{1}{\mu}$. It is assumed that the average arrival rate λ is less than the maximum service rate $s\mu$.

A channel that is turned on can handle only one request at the time. At any time, channels can be turned on and off depending on the number of service request in the system. A switching cost $K(a,b) \geq 0$ is incurred when adjusting the number of channels turned on from a to b. For each channel turned on there is an operating costs at a rate $r > 0$ per unit of time. Also, for each request a holding cost $h > 0$ is incurred for each unit of time the message is in the system until the service is completed. The objective is to find a rule for controlling the number of channels turned on such that the long run average cost per unit of time is minimal. The decision epochs are the epochs at which a new request for service arrives or the service of a request is completed.

Since the Poisson process and the exponential distribution are memoryless, the state of the system can be described by the pair (i,k), where i is the number of service requests present, and k is the number of channels turned on. In principle, the number of service requests in the system is unbounded. It is intuitively obvious that under each reasonable control rule all of the s channels will be turned on when the number of requests in the system is sufficiently large. In other words, choosing a sufficiently large integer $M \geq s$, it is from a practical point of view no restriction to assume that in the states (i,k) with $i \geq M$ the only feasible action is to turn on all of the s channels. However, this implies that we can restrict the control of the system only to those arrival epochs and service completion epochs at which no more that M service requests remain in the system.

By doing so, we obtain an SMDP with state space $S = \{(i,k) \mid 0 \leq i \leq M;\ 0 \leq k \leq s\}$. The action sets are $A(i,k)$, where $A(i,k) := \begin{cases} \{0,1,\dots,s\}, & 0 \leq i \leq M-1;\ 0 \leq k \leq s \\ \{s\}, & i = M,\ 0 \leq k \leq s \end{cases}$, where action a in state (i,k) means that the number of channels turned on is adjusted from k to a.

If action $a = s$ is taken in state (M,k), then the next decision epoch is defined as the first service completion epoch at which either $M-1$ (when there are no arrivals in the meantime) or M (when there are arrivals in the meantime) service requests are left behind. The first possibility has a probability of $\frac{s\mu}{\lambda+s\mu}$ and the second possibility has a probability of $\frac{\lambda}{\lambda+s\mu}$. Denote by the random variable $t(M,s)$ the time until the next decision epoch when action s is taken in state (M,k). The random variable $t(M,s)$ is the sum of two components. The first component is the time until the next service completion or the next arrival, whichever occurs first. This first component is exponentially distributed with expectation $\frac{1}{\lambda+s\mu}$. The second component is zero if a service completion occurs first, which has probability $\frac{s\mu}{\lambda+s\mu}$; otherwise, which has probability $\frac{\lambda}{\lambda+s\mu}$, it is the time needed to reduce the number of service requests from $M+1$ to M. Whenever M or more requests are in the system, we can imagine that a single 'superchannel' is servicing requests one at a time at an exponential rate of $s\mu$. Hence, from the properties of the $M/M/1$ queue we know that, if an arrival occurs first, the expectation of the second component of $t(M,s)$ is $\frac{1}{s\mu-\lambda}$. Therefore, we obtain the one-step expected transition time $\tau_{(M,k)}(s) = \frac{1}{\lambda+s\mu} + \frac{\lambda}{\lambda+s\mu} \cdot \frac{1}{s\mu-\lambda} = \frac{s\mu}{(\lambda+s\mu)(s\mu-\lambda)}$.

Next, we will compute the one-step expected costs $c^*_{(M,k)}(s)$. These costs consists of several terms:

- switching costs $K(k,s)$;
- operating costs: $rs \cdot \tau_{(M,k)}(s) = \frac{rs^2\mu}{(\lambda+s\mu)(s\mu-\lambda)}$;
- holding costs: $hM \cdot \tau_{(M,k)}(s) + h \cdot \frac{\lambda}{\lambda+s\mu} \cdot \frac{1}{s\mu-\lambda} \cdot (1+L)$, where L is the average number of requests that enter the system during the second component of $t(M,s)$.

From the the properties of the $M/M/1$ queue we know that $L = \frac{\lambda}{s\mu-\lambda}$. Hence, the holding costs are $hM \cdot \tau_{(M,k)}(s) + h \cdot \frac{\lambda}{\lambda+s\mu} \cdot \frac{1}{s\mu-\lambda} \cdot \frac{s\mu}{s\mu-\lambda}$. Therefore, we obtain for the one-step expected cost:

$$c^*_{(M,k)}(s) = K(k,s) + rs \cdot \frac{s\mu}{(\lambda+s\mu)(s\mu-\lambda)} + hM \cdot \frac{s\mu}{(\lambda+s\mu)(s\mu-\lambda)} + h \cdot \frac{s\mu}{(\lambda+s\mu)(s\mu-\lambda)} \cdot \frac{\lambda}{s\mu-\lambda}.$$

Finally, we have for the transition probabilities in state (M,k):

$$p_{(M,k)(M-1,s)}(s) = \frac{s\mu}{\lambda+s\mu};\ p_{(M,k)(M,s)}(s) = \frac{\lambda}{\lambda+s\mu} \text{ for } k = 0,1,\dots,s.$$

For the other states, the basic elements of the SMDP are:

$$\tau_{(i,k)}(a) = \frac{1}{\lambda+\mu\cdot \mathit{min}(i,a)},\ 0 \le i \le M-1;\ 0 \le a \le s;$$
$$c^*_{(i,k)}(a) = K(k,a) + \frac{h\cdot i + r\cdot a}{\lambda+\mu\cdot \mathit{min}(i,a)},\ 0 \le i \le M-1;\ 0 \le a \le s;$$
$$p_{(i,a)(i+1,a)}(a) = \frac{\lambda}{\lambda+\mu\cdot \mathit{min}(i,a)};\ p_{(i,a)(i-1,a)}(a) = \frac{\mu\cdot \mathit{min}(i,a)}{\lambda+\mu\cdot \mathit{min}(i,a)}.$$

Note that this model is a unichained SMDP. We set $\tau = \frac{1}{\lambda+s\mu}$. The value iteration scheme becomes:

$$v^{n+1}_{(i,k)} = \mathit{min}_{0\le a\le s}\Big\{\{\lambda+\mu\cdot \mathit{min}(i,a)\}\cdot K(k,a) + h\cdot i + r\cdot a + \frac{\lambda}{\lambda+s\mu}\cdot v^n_{(i+1,a)} + \frac{\mu\cdot \mathit{min}(i,a)}{\lambda+s\mu}\cdot v^n_{(i-1,a)} + \big\{1 - \frac{\lambda+\mu\cdot \mathit{min}(i,a)}{\lambda+s\mu}\big\}\cdot v^n_{(i,k)}\Big\}$$

for states (i,k) with $0 \ge i \le M-1$ and $0 \le k \le s$ and with the convention $v^n_{(-1,k)} = 0$.

For the states $(M,k),\ 0 \le k \le s$, we obtain

$$v^{n+1}_{(M,k)} = \frac{1}{s\mu}\cdot(\lambda+s\mu)(s\mu-\lambda)\cdot K(k,s) + h\cdot M + r\cdot s + \frac{s\mu-\lambda}{\lambda+s\mu}\cdot v^n_{(M-1,s)} + \frac{\lambda}{\lambda+s\mu}\cdot\frac{s\mu}{\lambda+s\mu}\cdot v^n_{(M,s)} + \big\{1 - \frac{s\mu-\lambda}{s\mu}\big\}\cdot v^n_{(M,k)}.$$

9.7.7 Continuous-time Markov decision processes

In continuous-time Markov decision processes (CTMDPs), the intertransition times are exponentially distributed. These times may depend on the state and the chosen action. Hence, when the current state is state i and action $a \in A(i)$ is chosen, the sojourn times $F_i(a,t)$ are given by

$$F_i(a,t) = 1 - e^{-\beta(i,a)t},\ t \ge 0 \tag{9.145}$$

for some parameter $\beta(i,a)$.

Consider a fixed stationary policy f^∞. The corresponding stochastic original process remains in state i for a period of time determined by an exponential distribution with parameter $\beta(i,f)$, and then jumps to state j with probability $p_{ij}(f)$. This process is a *continuous stationary Markov chain*. We may summarize the probabilistic behavior of the process in terms of its *infinitesimal generator*. By the infinitesimal generator we mean an $(N \times N)$-matrix $Q(f)$ with components

$$q_{ij}(f) := \begin{cases} -\{1-p_{ii}(f)\}\cdot\beta(i,f) & j = i \\ p_{ij}(f)\beta(i,f) & j \ne i \end{cases} \quad i,j \in S.$$

Continuous Markov chains

In the continuous Markov chain $\{X(t),\ t \geq 0\}$, induced by policy f^∞ and with sojourn time $T_i(f)$ in state i, we have for all states i, j with $j \neq i$, for all $t \geq 0$ and for h sufficiently small

$$\begin{aligned}\mathbb{P}\{X(t+h) = i \mid X(t) = i\} &= \mathbb{P}\{T_i(f) \geq h\} + \mathbb{P}\{T_i(f) \leq h\} \cdot p_{ii}(f) \\ &= e^{-\beta(i,f)h} + \{1 - e^{-\beta(i,f)h}\} \cdot p_{ii}(f) \\ &= \{1 - \beta(i,f)h\} + \{\beta(i,f)h \cdot p_{ii}(f)\} + o(h) \\ &= 1 + q_{ii}(f)h + o(h)\end{aligned}$$

and for every $j \neq i$,

$$\begin{aligned}\mathbb{P}\{X(t+h) = j \mid X(t) = i\} &= \mathbb{P}\{T_i(f) \leq h\} \cdot p_{ij}(f) \\ &= \{1 - e^{-\beta(i,f)h}\} \cdot p_{ij}(f) \\ &= \beta(i,f)h \cdot p_{ij}(f)\} + o(h) \\ &= q_{ij}(f)h + o(h)\end{aligned}$$

where $o(h)$ for a function $g(h)$ means $\lim_{h\to 0} \frac{g(h)}{h} = 0$. Note that one might argue that within the next h time units state j could be reached from state i by first jumping from state i to some state k and next jumping from state k to state j. However, the probability of two or more state transitions in a small interval h is of $o(h)$.

Let $P(t)$ be the $N \times N$-matrix defined by $\{P(t)\}_{ij} := \mathbb{P}\{X(t) = j \mid X(0) = i\}$ for $i, j \in S$. Then, the following results are well known from the theory of continuous Markov chains.

Lemma 9.54 *Chapman-Kolmogorov equations*
$P(t+s) = P(t)P(s)$ for every $s, t > 0$.

Proof
Take any $i, j \in S$. Then, we may write

$$\begin{aligned}\{P(t+s)\}_{ij} &= \mathbb{P}\{X(t+s) = j \mid X(0) = i\} = \textstyle\sum_k \mathbb{P}\{X(t+s) = j,\ X(t) = k \mid X(0) = i\} \\ &= \textstyle\sum_k \mathbb{P}\{X(t+s) = j \mid X(t) = k,\ X(0) = i\} \cdot \mathbb{P}\{X(t) = k \mid X(0) = i\} \\ &= \textstyle\sum_k \mathbb{P}\{X(t+s) = j \mid X(t) = k\} \cdot \mathbb{P}\{X(t) = k \mid X(0) = i\} \\ &= \textstyle\sum_k p_{ik}(t)\, p_{kj}(s) = \{P(t)P(s)\}_{ij}\}.\end{aligned}$$

□

Lemma 9.55 *Kolmogorovs forward differential equations*
$P'(t) = P(t)Q(f)$ for every $t > 0$ and every $f^\infty \in C(D)$.

Proof
From Lemma 9.54, we obtain $P(t+h) = P(t)P(h)$ for every $t, h > 0$. Therefore, for every $i, j \in S$,

$$p_{ij}(t+h) = \textstyle\sum_{k\neq j} p_{ik}(t)p_{kj}(h) + p_{ij}(t)p_{jj}(h) = \sum_{k\neq j} p_{ik}(t)q_{kj}(f)h + p_{ij}(t)\{1+q_{jj}(f)h\} + o(h).$$

Hence,

$$\frac{p_{ij}(t+h)-p_{ij}(t)}{h} = \textstyle\sum_{k\neq j} p_{ik}(t)q_{kj}(f) + p_{ij}(t)q_{jj}(f) + \frac{o(h)}{h} = \sum_k p_{ik}(t)q_{kj}(f) + \frac{o(h)}{h}.$$

Letting $h \downarrow 0$, we obtain $p'_{ij}(t) = \sum_k p_{ik}(t)q_{kj}(f)$. □

Lemma 9.56 *Kolmogorovs backward differential equations*
$P'(t) = Q(f)P(t)$ *for every* $t > 0$ *and every* $f^\infty \in C(D)$.

Proof
From Lemma 9.54, we obtain $P(t+h) = P(h)P(t)$ for every $t, h > 0$. Therefore, for every $i, j \in S$,

$$p_{ij}(t+h) = \sum_{k \neq j} p_{ik}(h)p_{kj}(t) + p_{ij}(t)p_{jj}(h) = \sum_{k \neq i} q_{ik}(f)hp_{kj}(t) + \{1 + q_{ii}(f)h\}p_{ij}(t) + o(h).$$

Hence,

$$\frac{p_{ij}(t+h) - p_{ij}(t)}{h} = \sum_{k \neq i} q_{ik}(f)p_{kj}(t) + q_{ii}(f)p_{ij}(t) + \frac{o(h)}{h} = \sum_k q_{ik}(f)p_{kj}(t) + \frac{o(h)}{h}.$$

Letting $h \downarrow 0$, we obtain $p'_{ij}(t) = \sum_k q_{ik}(f)p_{kj}(t)$. □

Since $P(0) = I$, we obtain from the above lemmata that the infinitesimal generator determines the probability distribution of the system. $P'(t) = P(t)Q(f)$ implies $P(t) = e^{tQ(f)}$, where $e^{tQ(f)}$ is defined by $e^{tQ(f)} := \sum_{k=0}^\infty \frac{\{tQ(f)\}^k}{k!}$. Consequently, processes with the same infinitesimal generator have identical distributions, provided they have the same initial distribution.

Uniformization

Uniformization is a powerful technique which transforms the original continuous-time process with nonidentical transition times into an equivalent continuous-time process with identical transition times. This technique was already used at the end of Section 9.7.5. Take the constant c such that

$$\{1 - p_{ii}(a)\} \cdot \beta(i, a) \leq c \text{ for all } (i, a) \in S \times A.$$

Remark that c can be taken as $\frac{1}{\tau}$, where τ is defined in (9.134). Let, also as in Section 9.7.5,

$$\overline{p}_{ij}(a) := \delta_{ij} - \{\delta_{ij} - p_{ij}(a)\} \cdot \frac{\beta(i, a)}{c} \text{ for all } i, j \in S \text{ and } a \in A(i).$$

Note that $\overline{p}_{ij}(a) \geq 0$ for all $(i, a) \in S \times A,\ j \in S$, and $\sum_j \overline{p}_{ij}(a) = 1$ for all $(i, a) \in S \times A$.

Consider the stochastic decision process $\{\overline{X}(t),\ t \geq 0\}$ with uniform exponential sojourn times with parameter $\overline{\beta}(i, a) = c$ for all $(i, a) \in S \times A$, and with transition probabilities $\overline{p}_{ij}(a)$ for all $(i, a) \in S \times A$ and $j \in S$. The corresponding infinitesimal generator $\overline{Q}(f)$ satisfies

$$\overline{q}_{ij}(f) = \begin{cases} -\{1 - \overline{p}_{ii}(f)\}\overline{\beta}(i, f) & = & -\Big\{1 - \{1 - \frac{\{1 - p_{ii}(f)\}\beta(i,f)}{c}\}\Big\} \cdot c & = & -\{1 - p_{ii}(f)\}\beta(i, f), & j = i; \\ \overline{p}_{ij}(f)\overline{\beta}(i, f) & = & \frac{p_{ij}(f)\}\beta(i,f)}{c} \cdot c & = & p_{ij}(f)\beta(i, f), & j \neq i. \end{cases}$$

Hence, given a deterministic policy f^∞, the stochastic processes $\{X(t),\ t \geq 0\}$ and $\{\overline{X}(t),\ t \geq 0\}$ have the same infinitesimal generator, so that they are equal in distribution. Notice also that $Q(f) = \overline{Q}(f) = c \cdot \{P(f) - I\}$. Since for every deterministic policy f^∞ the matrices $P(t),\ t \geq 0$, are completely determined by the infinitesimal generator $Q(f)$ via $P(t) = e^{tQ(f)}$, it follows from $Q(f) = \overline{Q}(f)$ that $\overline{P}(t) = P(t)$ for all $t \geq 0$, i.e. the original continuous-time process with nonidentical transition times is equivalent to a continuous-time process with identical transition times.

Let $\overline{P}^n(f)$ be the n-step transition probabilities of the discrete-time Markov chain $\overline{P}(f)$. Using the fact that the probability of exactly n state transitions of the process $\overline{X}(t)$ during a given time t equals the Poisson probability $e^{-ct} \cdot \frac{(ct)^n}{n!}$, it follows by conditioning that

$$\overline{p}_{ij}(t) = \sum_{n=0}^{\infty} \{\overline{P}^n(f)\}_{ij} \cdot e^{-ct} \cdot \frac{(ct)^n}{n!} \text{ for } t > 0 \text{ and } i, j \in S. \tag{9.146}$$

For any fixed time t and starting state i, the probabilities $\overline{p}_{ij}(t),\ j \in S$ can be computed by $\overline{p}_{ij}(t) = \sum_{n=0}^{\infty} z_j(n)$, where $z_j(n) := \{\overline{P}^n(f)\}_{ij} \cdot e^{-ct} \cdot \frac{(ct)^n}{n!}$ for $n = 0, 1, \ldots$ and all $j \in S$. The numbers $z_j(n)$ can be calculated by applying the recursion scheme

$$z_j(n) := \begin{cases} 0 & n = 0 \text{ and } j \neq i; \\ e^{-ct} & n = 0 \text{ and } j = i; \\ \frac{ct}{n} \cdot \sum_k z_k(n-1) \cdot \overline{p}_{kj}(f) & n \geq 1 \text{ and } j \in i. \end{cases} \tag{9.147}$$

Example 9.29 *Uniformization*

Consider a continuous Markov chain with $S = \{1, 2\};\ p_{11} = 0,\ p_{12} = 1;\ p_{21} = 1,\ p_{22} = 0;$ $\beta_1 = 2,\ \beta_2 = 0.8$. Take $c = 4$ and note that $(1 - p_{ii}) \cdot \beta_i \leq c$ for all $i \in S$).

Then, $P = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$ and $Q = \begin{pmatrix} -2 & 2 \\ 0.8 & -0.8 \end{pmatrix}$. For the Markov chain $\overline{P}$ we obtain $\overline{P} = \begin{pmatrix} 0.5 & 0.5 \\ 0.2 & 0.8 \end{pmatrix}$.

It is easy to see that the corresponding infinitesimal generator $\overline{Q} = \begin{pmatrix} -2 & 2 \\ 0.8 & -0.8 \end{pmatrix} = Q$.

Discounted rewards

Assume the same reward structure as in the previous sections. Then, we have

$$\begin{aligned} r_j^*(a) &= r_j(a) + s_j(a) \cdot \textstyle\int_0^\infty \{\int_0^t e^{-\lambda s} ds\} \cdot \beta(j,a) \cdot e^{-\beta(j,a)t} dt \\ &= r_j(a) + s_j(a) \cdot \tfrac{\beta(j,a)}{\lambda} \cdot \textstyle\int_0^\infty \{1 - e^{-\lambda t}\} \cdot \beta(j,a) \cdot e^{-\beta(j,a)t} dt \\ &= r_j(a) + s_j(a) \cdot \tfrac{\beta(j,a)}{\lambda} \cdot \{\textstyle\int_0^\infty e^{-\beta(j,a)t} dt - \int_0^\infty e^{-\{\lambda + \beta(j,a)\}t} dt\} \\ &= r_j(a) + s_j(a) \cdot \tfrac{\beta(j,a)}{\lambda} \cdot \{\tfrac{1}{\beta(j,a)} - \tfrac{1}{\lambda + \beta(j,a)}\} \\ &= r_j(a) + s_j(a) \cdot \tfrac{1}{\lambda + \beta(j,a)},\ (j,a) \in S \times A \end{aligned}$$

and

$$\begin{aligned} p_{ij}^*(a) &= p_{ij}(a) \cdot \textstyle\int_0^\infty e^{-\lambda t} \cdot \beta(i,a) \cdot e^{-\beta(j,a)t} dt \\ &= p_{ij}(a) \cdot \beta(i,a) \cdot \textstyle\int_0^\infty e^{-\{\lambda + \beta(i,a)\}t} dt \\ &= p_{ij}(a) \cdot \tfrac{\beta(i,a)}{\lambda + \beta(i,a)},\ (i,a) \in S \times A,\ j \in S. \end{aligned}$$

From Theorem 9.59 it follows that

$$v_i^\lambda(f^\infty) = r_i^*(f) + \textstyle\sum_j p_{ij}^*(f) v_j^\lambda(f^\infty) = r_i^*(f) + \tfrac{\beta(i,a)}{\lambda + \beta(i,a)} \cdot \sum_j p_{ij}(a) v_j^\lambda(f^\infty),\ i \in S.$$

If $\beta(i,f) = \beta$ for all $(i,a) \in S \times A$, then $v_i^\lambda(f^\infty) = r_i^*(f) + \alpha \cdot \sum_j p_{ij}(a) v_j^\lambda(f^\infty),\ i \in S$, with $\alpha := \frac{\beta}{\lambda + \beta} \in (0,1)$. Hence, in this case we have a discrete MDP with discount factor $\alpha = \frac{\beta}{\lambda + \beta} \in (0,1)$.

Next, we consider the CTMDP obtained by the technique of uniformization and with as rewards $\overline{r}_i(a) := r^*_i(a)\cdot\frac{\lambda+\beta(i,a)}{\lambda+c}$, $(i,a) \in S\times A$. The following result relates the original and the uniformized model.

Theorem 9.68
$\overline{v}^\lambda(f^\infty) = v^\lambda(f^\infty)$ for every $f^\infty \in C(D)$.

Proof
From Theorem 9.59 it follows that $v^\lambda(f^\infty)$ and $\overline{v}^\lambda(f^\infty)$ are the unique solutions of the linear systems

$$r^*_i(f) + \sum_j p_{ij}\cdot\frac{\beta(i,f)}{\lambda+\beta(i,f)}\cdot x_j = x_i,\ i \in S \tag{9.148}$$

and

$$\overline{r}_i(f) + \sum_j \overline{p}_{ij}\cdot\frac{\overline{\beta}(i,f)}{\lambda+\overline{\beta}(i,f)}\cdot y_j = y_i,\ i \in S, \tag{9.149}$$

respectively. System (9.149) can be rewritten as

$$r^*_i(a)\cdot\tfrac{\lambda+\beta(i,a)}{\lambda+c} + \textstyle\sum_{j\neq i} p_{ij}\cdot\frac{\beta(i,f)}{c}\cdot\frac{c}{\lambda+c}\cdot y_j + \big\{1 - \frac{\{1-p_{ii}(f)\}\cdot\beta(i,f)}{c}\big\}\cdot\frac{c}{\lambda+c}\cdot y_i = y_i,\ i \in S,$$

which is equivalent to

$$r^*_i(a)\cdot\{\lambda+\beta(i,a)\} + \textstyle\sum_{j\neq i} p_{ij}\cdot\beta(i,f)\cdot y_j + \big\{c - \{1-p_{ii}(f)\}\cdot\beta(i,f)\big\}\cdot y_i = (\lambda+c)\cdot y_i,\ i \in S,$$

or

$$r^*_i(a)\cdot\{\lambda+\beta(i,a)\} + \textstyle\sum_j p_{ij}\cdot\beta(i,f)\cdot y_j = \{\lambda+\beta(i,f)\}\cdot y_i,\ i \in S,$$

This last equation can be written as $r^*_i(a) + \sum_j p_{ij}\cdot\frac{\beta(i,f)}{\lambda+\beta(i,a)}\cdot y_j = y_i,\ i \in S$, which is system (9.148) (with y instead of x). □

From the above analysis we may consider the discounted CTMDP as a discrete discounted MDP $(S, A, \overline{p}, \overline{r})$ with discount factor $\alpha = \frac{c}{\lambda+c}$. Both models have, by Theorem 9.68, the same value vector. Note that by uniformization all results from Chapter 3 also are applicable to a discounted CTMDP.

Example 9.29 (continued)
Let $\lambda = 0.1$ and $r_1 = 3,\ r_2 = 5;\ s_1 = 2$ and $s_2 = 1$. Since $c = 4$, we have $\alpha = \frac{c}{\lambda+c} = \frac{40}{41}$.
Further, $r^*_1 = r_1 + s_1\frac{1}{\lambda+\beta_1} = \frac{83}{21};\ r^*_2 = r_2 + s_2\frac{1}{\lambda+\beta_2} = \frac{55}{9};\ \overline{r}_1 = r^*_1\cdot\frac{\lambda+\beta_1}{\lambda+c} = \frac{83}{41};\ \overline{r}_2 = r^*_2\cdot\frac{\lambda+\beta_2}{\lambda+c} = \frac{55}{41}$.
The value vector $\overline{v}^\alpha$ is the unique solution of the system.

$$\begin{cases} \overline{v}^\alpha_1 = \overline{r}_1 + \alpha\cdot\{\overline{p}_{11}\overline{v}^\alpha_1 + \overline{p}_{12}\overline{v}^\alpha_2\} = \frac{83}{41} + \frac{40}{41}\cdot\{0.5\overline{v}^\alpha_1 + 0.5\overline{v}^\alpha_2\} = \frac{83}{41} + \frac{20}{41}\cdot\overline{v}^\alpha_1 + \frac{20}{41}\cdot\overline{v}^\alpha_2; \\ \overline{v}^\alpha_2 = \overline{r}_2 + \alpha\cdot\{\overline{p}_{21}\overline{v}^\alpha_1 + \overline{p}_{22}\overline{v}^\alpha_2\} = \frac{55}{41} + \frac{40}{41}\cdot\{0.2\overline{v}^\alpha_1 + 0.8\overline{v}^\alpha_2\} = \frac{55}{41} + \frac{8}{41}\cdot\overline{v}^\alpha_1 + \frac{42}{41}\cdot\overline{v}^\alpha_2. \end{cases}$$

The solution of this system gives: $\overline{v}^\alpha_1 = 62\frac{21}{69}$ and $\overline{v}^\alpha_2 = 63\frac{20}{69}$

Average rewards - unichain case

In this subsection we consider a CTMDP with average rewards under the assumption that for all deterministic policies f^∞ the transition matrix $P(f)$ is unichain. For average rewards we have $r^*_i(a) = r_i(a) + \frac{s_i(a)}{\beta(i,a)}$ for all $(i,a) \in S \times A$. Also in this case we consider the uniformized model with rewards $\overline{r}_i(a) = r^*_i(a) \cdot \beta(i,a) = r_i(a) \cdot s_i(a)$ for all $(i,a) \in S \times A$. The relation between the original model and the uniformized model is explained in the following theorem.

Theorem 9.69
$\overline{\chi}(f^\infty) = c \cdot \chi(f^\infty)$ *for every* $f^\infty \in C(D)$.

Proof
From Algorithm 9.22 it follows that $\chi(f^\infty)$ and $\overline{\chi}(f^\infty)$ are the unique solutions x and w of the linear systems

$$x \cdot T(f)e + \{I - P(f)\}y = r^*(f); y_1 = 0 \qquad (9.150)$$

and

$$w \cdot \overline{T}(f)e + \{I - \overline{P}(f)\}z = \overline{r}(f); z_1 = 0, \qquad (9.151)$$

respectively. We also have the following relations:

$\overline{P}(f) = I - B(f)\{I - P(f)\}$, where $B(f)$ is a diagonal matrix with elements $\beta(i,f),\ i \in S$.

$\overline{r}(f) = B(f)r^*(f);\ T(f) = \{B(f)\}^{-1};\ \overline{T}(f) = \frac{1}{c} \cdot I$.

System (9.151) can be rewritten as $\frac{1}{c} \cdot \overline{\chi}(f^\infty) \cdot e + B(f)\{I - P(f)\}z = B(f)r^*(f);\ z_1 = 0$, what is equivalent to $\frac{1}{c} \cdot \overline{\chi}(f^\infty) \cdot T(f)e + \{I - P(f)\}z = r^*(f);\ z_1 = 0$. Now, it follows from (9.150) and that $\overline{\chi}(f^\infty) = c \cdot \chi(f^\infty)$. □

Example 9.29 (continued)
$r^*_1 = r_1 + s_1 \cdot \frac{1}{\beta_1} = 4;\ r^*_2 = r_2 + s_2 \cdot \frac{1}{\beta_2} = \frac{25}{4};\ \overline{r}_1 = r^*_1 \cdot \beta_1 = 8;\ \overline{r}_2 = r^*_2 \cdot \beta_2 = 5.$

$T = \begin{pmatrix} 0.5 & 0 \\ 0 & 1.25 \end{pmatrix};\ \overline{T} = \begin{pmatrix} 0.25 & 0 \\ 0 & 0.25 \end{pmatrix};\ P = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix};\ \overline{P} = \begin{pmatrix} 0.5 & 0.5 \\ 0.2 & 0.8 \end{pmatrix}.$

For the value χ we solve the system

$$\left\{ \begin{array}{rcrcrcl} \frac{1}{2}x & + & y_1 & - & y_2 & = & 4 \\ \frac{5}{4}x & - & y_1 & + & y_2 & = & \frac{25}{4} \\ & & y_1 & & & = & 0 \end{array} \right. \rightarrow x = \chi = \tfrac{41}{7},\ y_1 = 0,\ y_2 = -\tfrac{15}{14}.$$

For the value $\overline{\chi}$ we solve the system

$$\left\{ \begin{array}{rcrcrcl} \frac{1}{4}w & + & \frac{1}{2}z_1 & - & \frac{1}{2}z_2 & = & 8 \\ \frac{1}{4}w & - & \frac{1}{5}z_1 & + & \frac{1}{5}z_2 & = & 2 \\ & & z_1 & & & = & 0 \end{array} \right. \rightarrow w = \overline{\chi} = \tfrac{164}{7},\ z_1 = 0,\ y_2 = -\tfrac{30}{7}.$$

From the above analysis we may consider a unichain CTMDP with average rewards as a discrete unichain MDP $(S, A\overline{p}, \overline{r})$ with average rewards. Therefore, we may apply all results from Chapter 6. There are a lot of applications, particularly in queueing theory, that are successfully analyzed by applying uniformization. As an example we mention the admission control of an $M/M/1$-queue as discussed in Section 8.4.

9.8 Bibliographic notes

As the notion of computational complexity emerged, there were tremendous efforts in analyzing the complexity of MDPs and its solution methods. On the positive side, since it can be formulated as a linear program, the MDP can be solved in polynomial time by either the ellipsoid method (e.g. Khachiyan ([168]) or the interior-point method (e.g. Karmarkar ([157]).

The first results particularly for MDPs are due to Papadimitriou and Tsitsiklis ([211]) who showed for the variants finite horizon MDPs, discounted MDPs and undiscounted MDPs the following:

(1) these decision problems are complete for $\mathcal{P}$, and therefore most likely cannot be solved very fast by parallel algorithms;

(2) the deterministic cases of all these variants, the DMDPs, are in $\mathcal{NC}$, and therefore can be solved very fast in parallel.

Tseng ([290]) showed that the value-iteration method generates an optimal policy in polynomial time. Mansour and Singh ([195]) gave the upper bound $\frac{1}{N} \cdot 2^N$ on the number of iterations for the policy iteration method when each state has two actions (note that 2^N is the total number of policies, so that this result is not much better than complete enumeration). In 2005, Ye ([336]) developed a strongly polynomial-time combinatorial interior-point algorithm (CIPA).

In terms of the worst-case complexity bound on the number of arithmetic operations the best results (within a constant factor) are summarized in the following table, when there are exactly k actions in each of the N states and for L the total bit-size of the input data (see also Littman et al. [183]). Notice that the discount factor α is a fixed constant, no parameter.

Value iteration	Policy iteration	Linear programming	CIPA
$kN^2L \cdot \frac{log\{1/(1-\alpha)\}}{1-\alpha}$	$kN^3L \cdot \frac{log\{1/(1-\alpha)\}}{1-\alpha}$	k^2N^3L	$k^4N^4\,log\{N/(1-\alpha)\}$

For general linear programming, Klee and Minty ([169]) showed that the classic simplex method, with as pivot column the column of the most negative reduced cost, necessarily takes an exponential number of iterations in the worst case. In 1994, Melekopoglou and Condon ([197]) showed that a special policy iteration algorithm, where only the action in the largest state which has an improving action is updated, needs an exponential number of iterations.

Finally, Ye ([337]) showed that the classic simplex method is indeed a strongly polynomial-time algorithm for discounted MDPs. He proved that the number of iterations is bounded by $\frac{(k-1)N^2}{1-\alpha} \cdot log\{N^2/(1-\alpha)\}$, and that each each iteration uses at most $\mathcal{O}(kN^2)$ aritmetic operations. Since the policy iteration method with the all-negative-reduced-cost pivoting rule (in terms of a simplex method with block pivots) is at least good as the policy iteration method with only one new action per iteration (the action of the most negative reduced-cost), the policy iteration method is also a strongly polynomial-time algorithm with the same iteration complexity bound. Therefore, the worst case operation complexity $\mathcal{O}(k^2N^4\,log\,N)$ is actually superior to the complexity $\mathcal{O}(k^4N^4\,log\,N)$ of Yes combinatorial interior-point algorithm.

The first reference on MDPs with additional constraints is the paper of Derman and Klein ([70]). Derman was the first who presented a comprehensive treatment to analyze a constrained MDP ([[69], chapter 7). He introduced the state-action frequency approach for the analysis of these problems, and developed its relationship to linear programming. Derman and Veinott ([73]) analyzed CMDPs by applying the Dantzig-Wolfe decomposition principle.

Hordijk and Kallenberg ([129]) have derived results for transient MDPs with additional constraints. These results imply the treatment of discounted MDPs with additional constraints. Our presentation of the material on monotone optimal policies draws from a working paper by Serin ([265]). The section on finite horizon and additional constraints is due to Kallenberg ([147]).

Kallenberg ([148]), and Hordijk and Kallenberg ([129]) developed further properties of the sets of limiting state action frequencies, and extended the linear programming approach for MDPs with average rewards to manage also constrained multichain models. Altman and Spieksma ([5]) have have shown that the linear program for constrained MDPs can be obtained from an equivalent unconstrained Lagrange formulation of this problem. Altman, Hordijk and Kallenberg studied the value function for constrained discounted MDPs ([3]).

Constrained semi-Markov decision processes with with average rewards was studied by Feinberg ([83]). He considered two average reward criteria: time-average rewards and ratio-average rewards (ratio of total rewards during the first n steps and duration of first n steps as $n \to \infty$). Optimal policies exist for both criteria, but may be different for each of these criteria, even for unichain problems.

Section 9.2.7 on constrained MDPs with sum of discounted rewards and different discount factors is based on a paper of Feinberg and Shwartz ([85]), which uses results from Kallenberg ([147]). In Section 9.2.8 we consider the special case where a standard discounted reward function is to be maximized, subject to a constraint on another standard discounted reward function but with a different discount factor. For this case we provide an easier implementable algorithm for computing an optimal policy. This section is also based on a paper of Feinberg and Shwartz ([86])

The sensitivity of CMDPs was considered by Altman and Shwartz ([4]). White ([326]) and Beutler and Ross ([24]) used Lagrange multipliers to analyze constrained models. A more recent comprehensive survey of constrained MDPs with an emphasis on the Lagrange approach is Altmans book ([2]). We also mention some papers on CMDPs written by Ross and Varadarajan ([240], [241], [242]).

The results for multiobjective linear programming are based on the papers by Iserman ([140]), and Yu and Zeleny ([338]). The treatment of MDPs with multiple objective for the average reward criterion is based on the papers by Durinovic, Lee, Kathehakis and Filar ([76]), and Hordijk and Kallenberg ([129]).

Section 9.4, in which the linear program for unconstrained and constrained MDPs under the average reward criterion is revisited, is based on Altman and Spieksma paper ([5]).

Sobel ([278]) and Chung ([41], [42], [43]) considered the mean-variance ratio with a lower bound on the mean. Kawai ([165]) investigated the minimization of the variance with a lower bound on the mean. White ([329]) surveyed various models with mean-variance criteria and reviewed the importance of and relationship between the limiting state action frequencies in different classes of models. Filar, Kallenberg and Lee ([93]) and White ([330], [331]) analyzed the variance penalized model. Other contributions to the literature on mean-variance tradeoffs are Kawai and Katoh ([166]), Bayal-Gürsoy and Ross ([15]) and Sobel ([279]). Huang and Kallenberg ([137]) presented a framework that unifies and extends most of these approaches. The geometrical, linear algebra approach for the finite horizon variance-penalized problem is due to Collins ([45]).

Deterministic MDPs with average rewards or costs is based on the minimum mean-weight cycle in a directed graph. This minimum mean-weight cycle problem via shortest paths has been studied by several authors, among Karp ([158]). The approach via linear programming is borrowed from Lozovanu and Petric ([186]). The results for deterministic MDPs with discounted costs are developed by Madani, Thorup and Zwick ([190]).

Semi-Markov decision processes, also called Markov renewal programs, were introduced by Jewell ([142],[143], Howard ([135]), De Cani ([50]) and Schweitzer ([254]). Ross ([236]) introduced Assumption 9.3, which appears to be fundamental. The examples 9.22, 9.23 and 9.24 are taken from Puterman ([227]). Lemma 9.49 and the linear programming approach for discounted SMDPs were developed by Wessels and Van Nunen ([324]) and by Kallenberg ([148]). The contraction property of the operator U, defined in (9.115), is due to Denardo ([56]).

The proof that $\chi^1(\pi^\infty) = \chi^2(\pi^\infty)$ for stationary policies π^∞ is based on Ross ([237]). The fundamental Theorem 9.63 was derived by Denardo ([60]). The linear programming method for undiscounted SMDPs is due to Denardo and Fox ([64]) and to Kallenberg ([148]). The important data transformation, the uniformization technique, which converts SMDPs to equivalent MDPs was established by Schweitzer ([255]). The examples 9.25 and 9.25 are taken from Ross ([236]), and the examples 9.26 and 9.27 from Tijms ([288]).

The material of Section 9.7.7 is taken from Puterman ([227]) and Tijms ([288]).

9.9 Exercises

Exercise 9.1

Consider the following MDP model: $S = \{1,2\};\ A(1) = \{1,2\},\ A(2) = \{1\};$
$p_{11}(1) = 1,\ p_{12}(1) = 0;\ p_{11}(2) = 0,\ p_{12}(2); = 1;\ p_{21}(1) = 0,\ p_{22}(1) = 1;\ \beta_1 = \beta_2 = \frac{1}{2}.$
Determine the set Q of the long-run average state-action frequencies.

Exercise 9.2

Show by a counterexample that in the multichain case $x(\beta, \pi^\infty)$ is in general not continuous in π.

Exercise 9.3

Consider the inventory model with backlogging of Example 1.1. The state represent the inventory on hand and negative states represent backlogged orders. Suppose that we are interested in maximizing the long-run average profit, subject to the requirement that the average probability that there is out of stock is at most γ. Formulate the constraint of this optimization problem.

Exercise 9.4

Consider the following irreducible MDP model: $S = \{1, 2\};\ A(1) = \{1\},\ A(2) = \{1, 2, 3\};$
$p_{11}(1) = 0.4,\ p_{12}(1) = 0.6;\ p_{21}(1) = 1,\ p_{22}(1) = 0;\ p_{21}(2) = 0.8,\ p_{22}(2) = 0.2;$
$p_{21}(3) = 0.3,\ p_{22}(3) = 0.7.$

a. Determine an average optimal deterministic policy f^∞ by linear programming.

b. Add the constraint that the limiting state-action frequencies in state 2 is no more than 0.4 and solve the constrained model, i.e. determine an optimal stationary policy π^∞.

Exercise 9.5

Consider a unichain multi-objective MDP with immediate rewards $r_i^k(a),\ k = 1, 2, \dots, m$.
Let x be an optimal solution of the linear program

$$max \left\{ \sum_{(i,a)} r_i(a)x_i(a) \;\middle|\; \begin{array}{rcl} \sum_{(i,a)} \{\delta_{ij} - p_{ij}(a)\}x_i(a) & = & 0,\ j \in S \\ \sum_{(i,a)} x_i(a) & = & 1 \\ x_i(a) & \geq & 0,\ (i,a) \in S \times A \end{array} \right\},$$

where $r_i(a) = \sum_{k=1}^m \lambda_k r_i^k(a)$ for some $\lambda \in \mathbb{R}^m$ with $\lambda_k > 0,\ k = 1, 2, \dots, m$.

Define the stationary policy π^∞ by $\pi_{ia} = \begin{cases} x_i(a)/x_i & \text{if } x_i > 0; \\ \text{arbitrary} & \text{if } x_i = 0. \end{cases}$

Show that policy π^∞ is a β-efficient solution for any initial distribution β.

Exercise 9.6

Prove Lemma 9.53.

Exercise 9.7

Consider the following irreducible MDP model: $S = \{1, 2\};\ A(1) = \{1, 2\},\ A(2) = \{1\}.$
$p_{11}(1) = 0.8,\ p_{12}(1) = 0.2;\ p_{11}(2) = 0,\ p_{12}(2) = 1;\ p_{21}(1) = 1,\ p_{22}(1) = 0.$
$r_1(1) = 1,\ r_1(2) = 0,\ r_2(1) = 3.$

a. Determine for the two deterministic stationary policies the average reward and the variance.

b. Consider the mean-variance tradeoffs problem $min\{V(R) \mid \phi(R) \geq \frac{17}{12}\}$.

   (1) Formulate the parametric linear program for this problem and solve it.

   (2) Determine the optimal solution x_{opt} of problem (9.94) and the optimum value $V(x_{opt})$.

   (3) Determine an optimal policy according to the proof of Theorem 9.53.

   (4) Show that if π^∞ is the stationary policy which randomizes in state 1 between the two actions with the same randomization as the optimal policy uses between the two deterministic policies, then π^∞ is not an optimal policy for the mean-variance tradeoffs problem.

   (5) Try to find a stationary policy π^∞ with the same average reward and variance as the policy of part (3).

Chapter 10

Stochastic Games

10.1 Introduction

10.1.1 The model

In this chapter we first consider *two-person zero-sum stochastic games*. As in MDPs a stochastic game is a dynamic system that evolves along discrete time points. The state of the system at every time point is assumed to be one of the finite set $S = \{1, 2, \dots, N\}$. At these discrete time points each of the two players has the possibility to earn rewards and to influence the course of the system by choosing, independently of the choice of the other player, an action out of a finite action set. Let $A(i)$ and $B(i)$ be the action sets of player 1 and player 2, respectively, in state $i,\ i \in S$. If in state i player 1 chooses action $a \in A(i)$ and player 2 action $b \in B(i)$ then two things happen:

(1) Player 1 earns an immediate reward $r_i(a, b)$ from player 2 (*zero-sum game*);

(2) The next state is determined by a transitions which depend on the actions a and b, i.e. the state of the next decision time point is state j with probability $p_{ij}(a, b),\ j \in S$, where $\sum_j p_{ij}(a, b) = 1$ for every $i \in S,\ a \in A(i)$ and $b \in B(i)$.

Consider the Cartesian product

$$S \times A \times B := \{(i, a, b) \mid i \in S,\ a \in A(i),\ b \in B(i)\}$$

and let H_t denote the set of the possible *histories* of the system up to time point t, i.e.

$$H_t := \{h_t = (i_1, a_1, b_1, \dots, i_{t-1}, a_{t-1}, b_{t-1}, i_t \mid (i_k, a_k, b_k) \in S \times A \times B,\ 1 \leq k \leq t-1;\ i_t \in S\}.$$

A decision rule π^t at time point t for player 1 is a function on H_t which prescribes the action to be taken at time t as a transition probability from H_t into A, i.e.

$$\pi^t_{h_t a_t} \geq 0 \text{ for every } a_t \in A(i_t) \text{ and } \sum_{a_t} \pi^t_{h_t a_t} = 1 \text{ for every } h_t \in H_t.$$

A policy R_1 for player 1 is a sequence of decision rules: $R_1 = (\pi^1, \pi^2, \dots, \pi^t, \dots)$, where π^t is the decision rule at time point $t,\ t = 1, 2, \dots$. Similarly, the concept of a decision rule and a policy for player 2 is defined. As in the MDP model we distinguish between Markov, stationary and deterministic policies.

For stationary policies π^∞ and ρ^∞ for player 1 and 2, respectively, the transition matrix $P(\pi, \rho)$ and the reward vector $r(\pi, \rho)$ are defined by

$$p_{ij}(\pi, \rho) := \sum_{a,b} p_{ij}(a, b)\pi_{ia}\rho_{ib} \text{ for every } (i, j) \in S \times S; \tag{10.1}$$

$$r_i(\pi, \rho) := \sum_{a,b} r_i(a, b)\pi_{ia}\rho_{ib} \text{ for every } i \in S. \tag{10.2}$$

Furthermore, we define for all $i, j \in S$ and all $a \in A(i)$ and $b \in B(i)$:

$$p_{ij}(a, \rho) := \sum_b p_{ij}(a, b)\rho_{ib};\ r_i(a, \rho) := \sum_b r_i(a, b)\rho_{ib}; \tag{10.3}$$

$$p_{ij}(\pi, b) := \sum_a p_{ij}(a, b)\pi_{ia};\ r_i(\pi, b) := \sum_a r_i(a, b)\pi_{ia}. \tag{10.4}$$

10.1.2 Optimality criteria

Let $X_t,\ Y_t,\ Z_t$ be random variables denoting the observed state, the action chosen by player 1 and the action chosen by player 2, respectively, at time point t. For any two policies R_1 and R_2 for player 1 and player 2, respectively, and initial state i, we denote the *total expected discounted reward* and the *average expected reward* by $v_i^\alpha(R_1, R_2)$ and $\phi_i(R_1, R_2)$, defined by

$$v_i^\alpha(R_1, R_2) := \sum_{t=1}^{\infty} \alpha^{t-1} \sum_{j,a,b} \mathbb{P}_{i,R_1,R_2}\{X_t = j, Y_t = a, Z_t = b\} \cdot r_j(a, b). \tag{10.5}$$

and

$$\phi_i(R_1, R_2) := \liminf_{T\to\infty} \frac{1}{T} \sum_{t=1}^{T} \sum_{j,a,b} \mathbb{P}_{i,R_1,R_2}\{X_t = j, Y_t = a, Z_t = b\} \cdot r_j(a, b). \tag{10.6}$$

The *total expected reward*, given initial state i and the policies R_1 and R_2 is denoted by $v_i(R_1, R_2)$ and defined by

$$v_i(R_1, R_2) := \sum_{t=1}^{\infty} \sum_{j,a,b} \mathbb{P}_{i,R_1,R_2}\{X_t = j, Y_t = a, Z_t = b\} \cdot r_j(a, b), \tag{10.7}$$

under the following assumptions:

(1) The model is *substochastic*, i.e. $\sum_j p_{ij}(a, b) \leq 1$ for all $(i, a, b) \in S \times A \times B$.

(2) For any initial state i and any two policies R_1, R_2 the expected total reward $v_i(R_1, R_2)$ is well-defined (possibly $\pm\infty$).

Under the assumption that the model is *transient*, i.e. $\sum_{t=1}^{\infty} \mathbb{P}_{i,R_1,R_2}\{X_t = j, Y_t = a, Z_t = b\} < \infty$ for all i, j and a, b, it can be shown that, with $\alpha = 1$, most properties of the discounted model are valid for the total reward.

10.1.3 Matrix games

For the solution of stochastic games we sometimes make use of properties of matrix games. Therefore we present in this section a number of concepts and results in the theory of matrix games.[1] A *two-person zero-sum matrix game* can be represented by an $m \times n$-matrix $A = (a_{ij})$, the *game matrix* or *payoff matrix.* The actions of player 1 correspond to the rows and the actions of player 2 to the columns of A. When player 1 chooses row i and player 2 column j, player 2 has to pay the amount a_{ij} to player 1. If player 1 chooses row i, he will get at least $min_j\, a_{ij}$. Hence, by an optimal choice of row i, he can achieve $\underline{w}(A) := max_i min_j\, a_{ij}$. Similarly, player 2 can obtain a payoff of at most $\overline{w}(A) := min_j max_i\, a_{ij}$. It is well known that

$$\overline{w}(A) = min_i max_j\, a_{ij} \geq max_i min_j\, a_{ij} = \underline{w}(A).$$

Let us now allow the choice of a strategy by a player to be random. The set of *mixed strategies* of player 1 is the simplex

$$X := \big\{(x_1, x_2, \dots, x_m) \mid x_i \geq 0,\ 1 \leq i \leq m;\ \sum_{i=1}^{m} x_i = 1\big\}.$$

[1] For a comprehensive survey of matrix games we refer to Owen, G.: *Game Theory*, Academic Press, 1982.

An element $x \in X$ is the probability on the set of rows of A. Similarly, the set of mixed strategies of player 2 is the simplex

$$Y := \{(y_1, y_2, \ldots, y_n) \mid y_j \geq 0,\ 1 \leq j \leq n;\ \sum_{j=1}^{n} y_j = 1\}.$$

If player 1 uses $x \in X$ and player 2 $y \in Y$, the (average) payoff is $x^T Ay = \sum_{i=1}^m \sum_{j=1}^n x_i a_{ij} y_j$. Note that the *pure strategy* for player 1 of choosing row i may be represented as the mixed strategy e_i, the unit vector with a 1 in the i-th position and 0's elsewhere. Similarly, the pure strategy for player 2 of choosing column j may be represented as the mixed strategy e_j.

It is natural to consider the mixed *maxmin* and *minmax*, namely $\underline{v}(A) := max_{x\in X} min_{y\in Y}\, x^T Ay$ and $\overline{v}(A) := min_{y\in Y} max_{x\in X}\, x^T Ay$. Since $min_{y\in Y}\, x^T Ay = min_j\, x^T Ae_j$, we can write

$$\underline{v}(A) = max_{x\in X} min_{y\in Y}\, x^T Ay = max_{x\in X} min_j\, x^T Ae_j \geq max_i min_j\, a_{ij} = \underline{w}(A).$$

Similarly, $\overline{w}(A) \geq \overline{v}(A)$, implying $\overline{v}(A) - \underline{v}(A) \leq \overline{w}(A) - \underline{w}(A)$, i.e. mixed strategies reduce the 'duality gap'. Since $max_{x\in X}\, x^T Ay \geq max_{x\in X} min_{y\in Y}\, x^T Ay$ for all $y \in Y$, we obtain

$$\overline{v}(A) = min_{y\in Y} max_{x\in X}\, x^T Ay \geq max_{x\in X} min_{y\in Y}\, x^T Ay = \underline{v}(A).$$

The matrix game with payoff matrix A has a *value* $val(A)$ if $val(A) = \overline{v}(A) = \underline{v}(A)$. The policy $x^* \in X$ is an *optimal policy for player 1* if

$$(x^*)^T Ay \geq \overline{v}(A) \text{ for all } y \in Y.$$

The policy $y^* \in Y$ is an *optimal policy for player 2* if

$$x^T Ay^* \leq \underline{v}(A) \text{ for all } x \in X.$$

The basic Minmax Theorem for two-person zero-sum matrix games proves that the game has a value and that both players have optimal mixed strategies.

Theorem 10.1 *Minmax theorem*

Two-person zero-sum matrix games have a value and both players have optimal mixed strategies.

Proof

Consider the linear programming problem

$$min \left\{ y_0 \;\middle|\; y_0 \geq \sum_{j=1}^{n} a_{ij} y_j,\ 1 \leq i \leq m;\ \sum_{j=1}^{n} y_j = 1;\ y_j \geq 0,\ 1 \leq j \leq n \right\} \tag{10.8}$$

with corresponding dual program

$$max \left\{ x_0 \;\middle|\; x_0 \leq \sum_{i=1}^{m} a_{ij} x_i, 1 \leq j \leq n;\ \sum_{i=1}^{m} x_i = 1;\ x_i \geq 0, 1 \leq i \leq m \right\}. \tag{10.9}$$

Let (y_0^*, y^*) and (x_0^*, x^*) be optimal solutions of (10.8) and (10.9), respectively. Take any $x \in X$ and $y \in Y$. Then, we can write

$$y_0^* = \sum_{i=1}^m x_i y_0^* \geq \sum_{i=1}^m x_i \sum_{j=1}^n a_{ij} y_j^* = x^T A y^*$$

and

$$x_0^* = \sum_{j=1}^n y_j x_0^* \leq \sum_{j=1}^n y_j \sum_{i=1}^m a_{ij} x_i^* = (x^*)^T A y.$$

Hence, $x^T A y^* \leq y_0^* = x_0^* \leq (x^*)^T A y$ for all $x \in X$ and $y \in Y$. Therefore,

$$x_0^* = y_0^* = (x^*)^T A y^* \text{ and } (x^*)^T A y^* = max_{x \in X}\, x^T A y^* \text{ and } (x^*)^T A y^* = min_{y \in Y}\, (x^*)^T A y,$$

implying

$$\begin{array}{rcl} \underline{v}(A) & = & max_{x \in X} min_{y \in Y}\, x^T A y \;\geq\; min_{y \in Y}\, (x^*)^T A y \;=\; (x^*)^T A y^* \;=\; max_{x \in X}\, x^T A y^* \\ & \geq & min_{y \in Y} max_{x \in X}\, x^T A y = \overline{v}(A). \end{array}$$

Since we also have $\overline{v}(A) \geq \underline{v}(A)$, we have shown that $\overline{v}(A) = \underline{v}(A) = val(A)$ and $(x^*)^T A y \geq val(A)$ for all $y \in Y$ and $x^T A y^* \leq val(A)$, i.e. x^* and y^* are optimal policies for player 1 and 2, respectively. □

The simplest case of all occurs if a *saddle point* exists, i.e. there exists an entry a_{kl} which is both the maximum entry in its column and the minimum entry in its row. In this case the pure strategies row k for player 1 and column l for player 2 are optimal strategies as the following lemma shows.

Lemma 10.1

If $a_{kl} \geq a_{il}$ for all i and $a_{kl} \leq a_{kj}$ for all j, then $x = e_k$ and $y = e_l$ are optimal pure strategies for player 1 and 2, respectively, and a_{kl} is the value of the game.

Proof

The result follows immediately from the following observation.

$a_{kl} = max_i\, a_{il} \geq min_j max_i\, a_{ij} = \overline{w}(A) \geq val(A) \geq \underline{w}(A) = max_i min_j\, a_{ij} \geq min_j\, a_{kj} = a_{kl}$. □

Suppose that player 1 has the pure optimal strategy e_k. From $(e_k)^T A y \geq val(A) = (e_k)^T A y^*$ for all $y \in Y$ and some $y^* \in Y$ it follows that $a_{kj} \geq val(A) = (e_k)^T A y^*$ for $j = 1, 2, \dots, n$. Therefore, e_l is an optimal pure strategy for player 2, where l satisfies $a_{kl} = min_j\, a_{kj}$. Since e_l is an optimal pure strategy for player 2, we also have $x^T A e_l \leq val(A) = a_{kl}$ for all $x \in X$, implying $a_{il} \leq a_{kl}$ for $i = 1, 2, \dots, m$. Hence, A has a saddle point a_{kl} and we obtain the following result.

Lemma 10.2

If one of the players has a pure optimal strategy, both players have optimal pure strategies and the game has a saddle point.

Lemma 10.3

(1) For any $c \in \mathbb{R}$ and any $m \times n$-matrix A, $val(A + cJ) = val(A) + c$, where J is the $m \times n$-matrix with each entry equal to 1.

(2) For any two $m \times n$-matrices A and B with $a_{ij} \leq b_{ij}$ for all (i, j), we have $val(A) \leq val(B)$.

(3) For any two $m \times n$-matrices A and B, $|val(A) - val(B)| \leq max_{(k,l)}\, |a_{kl} - b_{kl}|$.

Proof

(1) and (2): Since $x^T(A+cJ)y = x^TAy + c$ and $x^TAy \leq x^TBy$, it is straightforward that $val(A+cJ) = val(A) + c$ and $val(A) \leq val(B)$.

(3) Notice that $a_{ij} - max_{(k,l)}\,|a_{kl} - b_{kl}| \leq b_{ij} \leq a_{ij} + max_{(k,l)}\,|a_{kl} - b_{kl}|$ for all (i,j). Hence, by (1) and (2), $val(A) \leq val(B) + max_{(k,l)}\,|a_{kl} - b_{kl}|$ and $val(B) \leq val(A) + max_{(k,l)}\,|a_{kl} - b_{kl}|$, implying $|val(A) - val(B)| \leq max_{(k,l)}\,|a_{kl} - b_{kl}|$. □

2 × 2 games

Suppose we are given the 2 × 2 matrix game $\begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix}$. It may be that this game has a saddle point; if so, this entry is the value and provides the optimal strategies which are pure. Suppose that the game has no saddle point. Then, by Lemma 10.2 both players have completely mixed optimal strategies x and y, i.e. $x_1 > 0,\ x_2 > 0,\ y_1 > 0$ and $y_2 > 0$. For the value of the game we have $val(A) = x_1\{a_{11}y_1 + a_{12}y_2\} + x_2\{a_{21}y_1 + a_{22}y_2\}$. The two terms between brackets are at most $val(A)$ (see the linear program (10.8)), we have $val(A) = a_{11}y_1 + a_{12}y_2 = a_{21}y_1 + a_{22}y_2$. Similarly, it can been seen that $val(A) = a_{11}x_1 + a_{21}x_2 = a_{12}x_1 + a_{22}x_2$. In vector notation, $v = Ay$ and $v = A^Tx$, where $v := \begin{pmatrix} val(A) \\ val(A) \end{pmatrix}$.

If A is nonsingular, we can write

$$\begin{array}{lll} v = A^Tx & \rightarrow & (A^{-1})^Tv = x \ \rightarrow\ v^TA^{-1}e = \{(A^{-1})^Tv\}^Te = x^Te = 1 \ \rightarrow\ val(A)\cdot e^TA^{-1}e = 1 \\ & \rightarrow & val(A) = \frac{1}{e^TA^{-1}e}. \\ v = Ay & \rightarrow & y = A^{-1}v = val(A)\cdot A^{-1}e = \frac{A^{-1}e}{e^TA^{-1}e}. \\ v = A^Tx & \rightarrow & x = (A^{-1})^Tv = val(A)\cdot(A^{-1})^Te = \frac{(A^{-1})^Te}{e^TA^{-1}e}. \end{array}$$

If A is singular, this is of course meaningless. Then, it can be shown that

$$val(A) = \tfrac{|A|}{e^TA^*e} = \tfrac{a_{11}a_{22}-a_{12}a_{21}}{a_{11}+a_{22}-a_{12}-a_{21}},\ y = \tfrac{A^*e}{e^TA^*e} = \left(\tfrac{a_{22}-a_{21}}{a_{11}+a_{22}-a_{12}-a_{21}}, \tfrac{a_{11}-a_{12}}{a_{11}+a_{22}-a_{12}-a_{21}}\right),$$

$$x = \tfrac{A^*e}{e^TA^*e} = \left(\tfrac{a_{22}-a_{12}}{a_{11}+a_{22}-a_{12}-a_{21}}, \tfrac{a_{11}-a_{21}}{a_{11}+a_{22}-a_{12}-a_{21}}\right).$$

where $|A|$ the determinant of A and A^* is the adjoint of A. Note that the formulas in the nonsingular case coincide with the above formulas, because $A^*A = AA^* = |A|\cdot I$. For the details on the adjoint of A and the property $A^*A = AA^* = |A|\cdot I$ we refer to text books on linear algebra.

10.1.4 Bimatrix games

A pair of matrices (M^1, M^2) constitutes a *bimatrix game* when the sizes of M^1 and M^2 are equal. Let M^1 and M^2 be $m \times n$ matrices. The rows correspond to pure actions of player 1 and the columns to pure actions of player 2. Given a pair of *pure actions* (i,j), the payoff for player 1 can be found in the corresponding entry of the matrix M^1 and the payoff for player 2 in the corresponding entry of the matrix M^2. We allow mixed strategies, i.e., the players are allowed to make a convex combination of pure actions. These mixed strategies are represented by probability vectors x and y for player 1 and 2, respectively.

A bimatrix game is a generalization of a matrix game, because a bimatrix game (M^1, M^2) with $M^2 = -M^1$ is equivalent to a matrix game. For bimatrix games we use the notion of equilibrium points. A pair (x^*, y^*) is an equilibrium point if and only if

$$\begin{aligned}(x^*)^T M^1 y^* &\geq x^T M^1 y^* \text{ for all mixed strategies } x \text{ for player 1;} &(10.10)\\ (x^*)^T M^2 y^* &\geq (x^*)^T M^2 y \text{ for all mixed strategies } y \text{ for player 2.} &(10.11)\end{aligned}$$

The following result, due to Nash ([202]), is well known.

Theorem 10.2
Each bimatrix game has at least one equilibrium point.

Consider the following associated quadratic program (quadratic objective function and and linear constraints):

$$max \left\{ x^T M^1 y + x^T M^2 y - z^1 - z^2 \,\middle|\, \begin{array}{ll} \sum_{j=1}^n m^1_{ij} y_j \leq z^1, & 1 \leq i \leq m \\ \sum_{i=1}^m m^2_{ij} x_i \leq z^2, & 1 \leq j \leq n \\ \sum_{j=1}^n y_j = 1;\; y_j \geq 0, & 1 \leq j \leq n \\ \sum_{i=1}^m x_i = 1;\; x_i \geq 0, & 1 \leq i \leq m \end{array} \right\}. \qquad (10.12)$$

From the first two sets of linear constraints it follows that for any feasible solution (x, y, z^1, z^2) of the quadratic program (10.12) we have $x^T M^1 y + x^T M^2 y - z^1 - z^2 \leq 0$, i.e. the optimum of the quadratic program is at most 0.

Remark
Consider the special case of a matrix game, i.e. $M^1 = -M^2$. Then the quadratic program becomes the linear program

$$max \left\{ -z^1 - z^2 \,\middle|\, \begin{array}{ll} \sum_{j=1}^n m^1_{ij} y_j \leq z^1, & 1 \leq i \leq m \\ -\sum_{i=1}^m m^1_{ij} x_i \leq z^2, & 1 \leq j \leq n \\ \sum_{j=1}^n y_j = 1;\; y_j \geq 0, & 1 \leq j \leq n \\ \sum_{i=1}^m x_i = 1;\; x_i \geq 0, & 1 \leq i \leq m \end{array} \right\}.$$

From Section 10.1.3 we know that $\overline{x}$ and $\overline{y}$ are optimal mixed strategies for player 1 and 2, respectively, if and only if $(\overline{x}, \overline{x}_0)$ and $(\overline{y}, \overline{y}_0)$ are feasible solutions of (10.9) and (10.8) with $\overline{x}_0 = \overline{y}_0 = (\overline{x})^T M^1 \overline{y}$. Take $\overline{z}^1 = \overline{y}_0$ and $\overline{z}^2 = -\overline{x}_0$, then $(\overline{x}, \overline{y}, \overline{z}^1, \overline{z}^2)$ is an optimal solution of the above linear program with value 0. Hence, $(\overline{x}, \overline{y}, \overline{z}^1, \overline{z}^2)$ is an optimal solution of the above linear program with value 0 if and only if $\overline{x}$ and $\overline{y}$ are optimal mixed strategies, $\overline{z}^1 = (\overline{x})^T M^1 \overline{y}$ and $\overline{z}^2 = -(\overline{x})^T M^1 \overline{y}$. This result can be generalized to bimatrix games.

Theorem 10.3
The following two assertions are equivalent:
(1) $(\overline{x},\overline{y},\overline{z}^1,\overline{z}^2)$ is an optimal solution of the quadratic program (10.12) with value 0.
(2) $(\overline{x},\overline{y})$ is an equilibrium point, $\overline{z}^1=(\overline{x})^TM^1\overline{y}$ and $\overline{z}^2=-(\overline{x})^TM^1\overline{y}$.

Proof
First, suppose that $(\overline{x},\overline{y},\overline{z}^1,\overline{z}^2)$ is an optimal solution of the quadratic program (10.12) with value 0. Since the value equals 0, we have $\overline{z}^1=(\overline{x})^TM^1\overline{y}$ and $\overline{z}^2=-(\overline{x})^TM^1\overline{y}$. Furthermore, from the first two sets of the constraints of (10.12) it follows that $x^TM^1\overline{y}\leq(\overline{x})^TM^1\overline{y}$ and $\overline{x}^TM^2y\leq(\overline{x})^TM^2\overline{y}$, i.e. $(\overline{x},\overline{y})$ is an equilibrium point.
Next, assume that $(\overline{x},\overline{y})$ is an equilibrium point and that $\overline{z}^1=(\overline{x})^TM^1\overline{y}$ and $\overline{z}^2=-(\overline{x})^TM^1\overline{y}$. By taking for $i=1,2,\dots,n$, the pure strategy $x=e_i$, the ith unit vector, it follows that $\sum_{j=1}^n m^1_{ij}\overline{y}_j\leq\overline{z}^1$, i.e. $(\overline{y},\overline{z}^1)$ satisfies the first set of the constraints of (10.12). Similarly, $(\overline{x},\overline{z}^2)$ satisfies the second set of the constraints of (10.12). □

By Theorem 10.3, the quadratic program (10.12) is equivalent to the equilibrium point question. Since, by Theorem 10.2, every bimatrix game has an equilibrium point, the quadratic program (10.12) has an optimal solution with value 0.

Remark
The objective function is not concave in general. However, it is known a priori that its global maximum is zero. Hence, the lack of concavity is not a handicap, since in most computational schemes, concavity of the objective function is invoked mainly to exclude local maxima. Here, a local maximum, if any, will be immediately discarded upon finding that its value is less than zero.

10.2 Discounted rewards

10.2.1 Value and optimal policies

A policy R_1^* is *optimal for player 1* if $v^\alpha(R_1^*,R_2)\geq inf_{R_2}\,sup_{R_1}\,v^\alpha(R_1,R_2)$ for all policies R_2.
A policy R_2^* is *optimal for player 2* if $v^\alpha(R_1,R_2^*)\leq sup_{R_1}\,inf_{R_2}\,v^\alpha(R_1,R_2)$ for all policies R_1.
The stochastic discounted game has a *value* if $inf_{R_2}\,sup_{R_1}\,v^\alpha(R_1,R_2)=sup_{R_1}\,inf_{R_2}\,v^\alpha(R_1,R_2)$.
A policy R_1^* is *ε-optimal for player 1* if $v^\alpha(R_1^*,R_2)\geq inf_{R_2}sup_{R_1}v^\alpha(R_1,R_2)-\varepsilon$ for all policies R_2.
A policy R_2^* is *ε-optimal for player 2* if $v^\alpha(R_1,R_2^*)\leq sup_{R_1}inf_{R_2}v^\alpha(R_1,R_2)+\varepsilon$ for all policies R_1.

Theorem 10.4
If the policies R_1^* and R_2^* satisfy $v^\alpha(R_1,R_2^*)\leq v^\alpha(R_1^*,R_2^*)\leq v^\alpha(R_1^*,R_2)$ for all policies R_1 and R_2, the game has a value and R_1^* and R_2^* are optimal policies.

Proof

We can write

$$\begin{aligned} sup_{R_1}\, inf_{R_2}\, v^\alpha(R_1,R_2) &\geq inf_{R_2}\, v^\alpha(R_1^*,R_2) \geq v^\alpha(R_1^*,R_2^*) \\ &\geq sup_{R_1}\, v^\alpha(R_1,R_2^*) \geq inf_{R_2}\, sup_{R_1}\, v^\alpha(R_1,R_2). \end{aligned}$$

On the other hand, $inf_{R_2}\, sup_{R_1}\, v^\alpha(R_1,R_2) \geq inf_{R_2}\, v^\alpha(R_1,R_2)$ for all policies R_1, implying $inf_{R_2}\, sup_{R_1}\, v^\alpha(R_1,R_2) \geq sup_{R_1} inf_{R_2}\, v^\alpha(R_1,R_2)$. Hence, we have shown that $inf_{R_2}\, sup_{R_1}\, v^\alpha(R_1,R_2) = sup_{R_1} inf_{R_2}\, v^\alpha(R_1,R_2) = v^\alpha(R_1^*,R_2^*)$ i.e. the game has a value. Since $v^\alpha(R_1^*,R_2) \geq inf_{R_2}\, sup_{R_1}\, v^\alpha(R_1,R_2)$ for all R_2 and $v^\alpha(R_1,R_2^*) \leq sup_{R_1}\, inf_{R_2}\, v^\alpha(R_1,R_2)$ for all R_1, i.e. R_1^* and R_2^* are optimal policies. □

We will show in this section that the game has a value and that there exist stationary optimal policies for both players. Furthermore, we present algorithms to approximate the value and stationary optimal policies arbitrarily close. Let Π and Γ be the set of stationary policies for player 1 and 2, respectively. Define for any $x \in \mathbb{R}^N$ the mapping $T : \mathbb{R}^N \to \mathbb{R}^N$ by

$$(Tx)_i = inf_{\rho^\infty \in \Gamma}\, sup_{\pi^\infty \in \Pi}\, \{r_i(\pi,\rho) + \alpha \sum_j p_{ij}(\pi,\rho)x_j\},\ i \in S. \tag{10.13}$$

$(Tx)_i$ is the value of a matrix game with matrix $M_x[i]$. The matrix $M_x[i]$ has $m = \#A(i)$ rows and $n = \#B(i)$ columns and the payoff, if player 1 chooses row a and player 2 column b, is $r_i(a,b) + \alpha \sum_j p_{ij}(a,b)x_j$. We will show in the next theorem that T is a monotone contraction.

Theorem 10.5

The mapping T, defined in (10.13), is a monotone contraction with respect to the supremum norm $\|\cdot\|_\infty$ with contraction factor α and fixed point $v^\alpha = \inf_{R_2} \sup_{R_1}\, v^\alpha(R_1,R_2) = \sup_{R_1} \inf_{R_2}\, v^\alpha(R_1,R_2)$.

Proof

Let $x,y \in \mathbb{R}^N$ with $x \leq y$. Take any $i \in S$. Then, $\{M_x[i]\}_{ab} \leq \{M_y[i]\}_{ab}$ for all (a,b). By Lemma 10.3 part (2), $(Tx)_i = val(M_x[i]) \leq val(M_y[i]) = (Ty)_i$, proving the monotonicity.
$\|Tx - Ty\|_\infty = \max_i |(Tx)_i - (Ty)_i|$. Notice that $|(Tx)_i - (Ty)_i| = |val(M_x[i]) - val(M_y[i])|$.
By Lemma 10.3 part (3), we can write,

$$\begin{aligned} |val(M_x[i]) - val(M_y)[i]| &\leq \max_{(a,b)} |\{r_i(a,b) + \alpha \textstyle\sum_j p_{ij}(a,b)x_j\{-\{r_i(a,b) + \alpha \sum_j p_{ij}(a,b)y_j\}| \\ &= \alpha \cdot \max_{(a,b)} |\textstyle\sum_j p_{ij}(a,b)(x_j - y_j)| \leq \alpha \cdot \|x-y\|_\infty, \end{aligned}$$

implying that T is a contraction with contraction factor α.
Hence, T has a unique fixed point, say v^α. We now show that there exist stationary policies $(\pi^*)^\infty$ and $(\rho^*)^\infty$ such that $v^\alpha\big(\pi^\infty,(\rho^*)^\infty\big) \leq v^\alpha \leq v^\alpha\big((\pi^*)^\infty,\rho^\infty\big)$ for every $\pi^\infty \in \Pi$ and $\rho^\infty \in \Gamma$. Let π^* be such that $\pi_{ia}^*,\ a \in A(i)$, is an optimal mixed strategy in the matrix game with matrix $\{r_i(a,b) + \alpha \sum_j p_{ij}(a,b)v_j^\alpha\}$, which - because of the fixed point property - has value $v_i^\alpha,\ i \in S$. So, $r(\pi^*,\rho) + \alpha P(\pi^*,\rho)v^\alpha \geq v^\alpha$ for all $\rho^\infty \in \Gamma$, implying $v^\alpha\big((\pi^*)^\infty,\rho^\infty\big) \geq v^\alpha$ for every $\rho^\infty \in \Gamma$. Similarly, it can be shown that $v^\alpha\big(\pi^\infty,(\rho^*)^\infty\big) \leq v^\alpha$ for every $\pi^\infty \in \Pi$. Therefore

$$v^\alpha\big(\pi^\infty,(\rho^*)^\infty\big) \leq v^\alpha \leq v^\alpha\big((\pi^*)^\infty,\rho^\infty\big) \text{ for every } \pi^\infty \in \Pi \text{ and } \rho^\infty \in \Gamma. \tag{10.14}$$

As in the proof of Theorem 10.4, we obtain from these inequalities

$$v^\alpha = v^\alpha\big((\pi^*)^\infty, (\rho^*)^\infty\big) = \inf_{\rho^\infty \in \Gamma} \sup_{\pi^\infty \in \Pi} v^\alpha(\pi^\infty, \rho^\infty) = \sup_{\pi^\infty \in \Pi} \inf_{\rho^\infty \in \Gamma} v^\alpha(\pi^\infty, \rho^\infty). \tag{10.15}$$

Finally we show that $v^\alpha = \inf_{R_2} \sup_{R_1} v^\alpha(R_1, R_2) = \sup_{R_1} \min_{R_2} v^\alpha(R_1, R_2)$.
Since $\sup_{R_1} v^\alpha(R_1, R_2) \geq \sup_{R_1} \min_{R_2} v^\alpha(R_1, R_2)$ for all policies R_2, we have

$$\inf_{R_2} \sup_{R_1} v^\alpha(R_1, R_2) \geq \sup_{R_1} \min_{R_2} v^\alpha(R_1, R_2).$$

Take any fixed policy ρ^∞ for player 2. This induces an MDP, so we obtain

$$\sup_{R_1} v^\alpha(R_1, \rho^\infty) = \max_{\pi^\infty \in \Pi} v^\alpha(\pi^\infty, \rho^\infty) \text{ for any fixed } \rho^\infty \in \Gamma$$

and similarly $\inf_{R_2} v^\alpha(\pi^\infty, R_2) = \min_{\rho^\infty \in \Gamma} v^\alpha(\pi^\infty, \rho^\infty)$ for any fixed $\pi^\infty \in \Pi$.

Because

$$\begin{array}{rcl} \sup_{R_1} \inf_{R_2} v^\alpha(R_1, R_2) & \geq & \sup_{\pi^\infty \in \Pi} \inf_{R_2} v^\alpha(\pi^\infty, R_2) = \sup_{\pi^\infty \in \Pi} \inf_{\rho^\infty \in \Gamma} v^\alpha(\pi^\infty, \rho^\infty) \\ & = & v^\alpha = \inf_{\rho^\infty \in \Gamma} \sup_{\pi^\infty \in \Pi} v^\alpha(\pi^\infty, \rho^\infty) = \inf_{\rho^\infty \in \Gamma} \sup_{R_1} v^\alpha(R_1, \rho^\infty) \\ & \geq & \inf_{R_2} \sup_{R_1} v^\alpha(R_1, R_2), \end{array}$$

we have shown that $v^\alpha = \inf_{R_2} \sup_{R_1} v^\alpha(R_1, R_2) = \sup_{R_1} \min_{R_2} v^\alpha(R_1, R_2)$. □

Corollary 10.1
The game has a value v^α, which satisfies $v^\alpha_i = val(M_{v^\alpha}[i]),\ i \in S$. Furthermore, there are stationary optimal policies for both players.

Proof
From the last line of the proof of Theorem 10.5 we obtain that v^α is value of the game. Since v^α is the unique fixed point of T, we have $v^\alpha_i = val(M_{v^\alpha}[i]),\ i \in S$. Furthermore, we can write,

$$v^\alpha\big((\pi^*)^\infty, R_2\big) \geq \inf_{R_2} v^\alpha\big((\pi^*)^\infty, R_2\big) = \inf_{\rho^\infty \in \Gamma} v^\alpha\big((\pi^*)^\infty, \rho^\infty\big) = v^\alpha,$$

the last equality by (10.14), i.e. $(\pi^*)^\infty$ is an optimal policy for player 1. Similarly, we have

$$v^\alpha\big(R_1, (\rho^*)^\infty\big) \leq \sup_{R_1} v^\alpha\big(R_1, (\rho^*)^\infty\big) = \sup_{\pi^\infty \in \Pi} v^\alpha\big((\pi^\infty, (\rho^*)^\infty\big) = v^\alpha,$$

i.e. $(\rho^*)^\infty$ is an optimal policy for player 2. □

Example 10.1
$S = \{1, 2\};\ A(1) = B(1) = \{1, 2\},\ A(2) = B(2) = \{1\};\ \alpha = \frac{1}{2}$.
$r_1(1,1) = \frac{1}{2},\ r_1(1,2) = 1,\ r_1(2,1) = 3,\ r_1(2,2) = \frac{3}{2},\ r_2(1,1) = 1$.
$p_{11}(1,1) = \frac{1}{3},\ p_{12}(1,1) = \frac{2}{3};\ p_{11}(1,2) = 0,\ p_{12}(1,2) = 1;\ p_{11}(2,1) = 0,\ p_{12}(2,1) = 1;$
$p_{11}(2,2) = \frac{1}{2},\ p_{12}(2,2) = \frac{1}{2};\ p_{21}(1,1) = 0,\ p_{22}(1,1) = 1$.
Consider the fixed point equation $x = Tx$, i.e.

$$x_1 = val \begin{pmatrix} \frac{1}{2} + \frac{1}{6}x_1 + \frac{1}{3}x_2 & 1 + \frac{1}{2}x_2 \\ 3 + \frac{1}{2}x_2 & \frac{3}{2} + \frac{1}{4}x_1 + \frac{1}{4}x_2 \end{pmatrix};\ x_2 = val\big(1 + \tfrac{1}{2}x_2\big).$$

Hence, $v_2^\alpha = x_2 = 2$ and $x_1 = val \begin{pmatrix} \frac{5}{6} + \frac{1}{6}x_1 & 2 \\ 4 & 2 + \frac{1}{4}x_1 \end{pmatrix}$. Since the maximum reward is $\frac{3}{2}$, the total expected discounted reward is at most $\frac{3/2}{1-\alpha} = 3$. Therefore the second row of the matrix dominates the first one and player 1 and 2 will both choose the second action:
$x_1 = 2 + \frac{1}{4}x_1 \ \rightarrow \ v_1^\alpha = x_1 = \frac{8}{3}$.

Perfect information

A stochastic game is said to be a game of *perfect information* if the state space S can be divided into two disjoint sets S_1 and S_2 such that $|A(i)| = 1$ for $i \in S_1$ and $|B(i)| = 1$ for $i \in S_2$. Then, the matrices in the matrix game with matrix M_x has either one row (if $i \in S_1$) or one column (if $i \in S_2$). Hence, the optimal policies are pure, i.e. nonrandomized, and we obtain the following result.

Corollary 10.2

In a discounted stochastic game with perfect information, both players possess optimal deterministic policies.

Lemma 10.4

A pair of deterministic policies (f^∞, g^∞) is optimal if and only if $v^\alpha(f^\infty, g^\infty) = v^\alpha$, the value of the stochastic game.

Proof

The if-part is obvious, because the optimality of (f^∞, g^∞) implies $v^\alpha(f^\infty, h_2^\infty) \geq v^\alpha \geq v^\alpha(h_1^\infty, g^\infty)$ for every pair h_1^∞, h_2^∞, and consequently, $v^\alpha(f^\infty, g^\infty) = v^\alpha$.
Let the pair (f^∞, g^∞) be such that $v^\alpha(f^\infty, g^\infty) = v^\alpha$, and let (f_*^∞, g_*^∞) be a pair of deterministic optimal policies. When players 2 policy is fixed at g^∞, we are in an MDP situation with one-step rewards $r_i(a, g_*)$ and transition probabilities $p_{ij}(a, g_*)$, and f^∞ is an optimal policy for this MDP. Thus, for any deterministic policy h^∞ for player 1, we have

$$r_i(h_1, g_*) + \alpha \sum_j p_{ij}(h_1, g_*) v_j^\alpha(f_*^\infty, g_*^\infty) \leq v_i^\alpha(f_*^\infty, g_*^\infty), \ i \in S. \tag{10.16}$$

In the states $i \in S_2$, player 2 has only one action, so $g(i) = g_*(i), \ i \in S_2$. Furthermore, $v^\alpha(f_*^\infty, g_*^\infty) = v^\alpha = v^\alpha(f^\infty, g^\infty)$. Hence, we obtain for any deterministic policy h_1^∞ for player 1, and also using (10.16),

$$r_i(h_1, g) + \alpha \sum_j p_{ij}(h_1, g) v_j^\alpha(f^\infty, g^\infty) \leq v_i^\alpha(f^\infty, g^\infty), \ i \in S_2. \tag{10.17}$$

For the states $i \in S_1$, player 1 has only one action. Hence, for all $i \in S_1, \ r_i(h_1, g) = r_i(f, g)$ and $p_{ij}(h_1, g) = p_{ij}(f, g), \ j \in S$. Because also $v_i^\alpha(f^\infty, g^\infty) = r_i(f, g) + \alpha \sum_j p_{ij}(f, g) v^\alpha(f^\infty, g^\infty)$ for all $i \in S$, we can write,
$v_i^\alpha(f^\infty, g^\infty) = r_i(f, g) + \alpha \sum_j p_{ij}(f, g) v^\alpha(f^\infty, g^\infty) = r_i(h_1, g) + \alpha \sum_j p_{ij}(h_1, g) v^\alpha(f^\infty, g^\infty), \ i \in S_1$.

Therefore, using (10.17), we have

$$r_i(h_1, g) + \alpha \sum_j p_{ij}(h_1, g) v_j^\alpha(f^\infty, g^\infty) \leq v_i^\alpha(f^\infty, g^\infty),\ i \in S. \tag{10.18}$$

From MDP we know that the map $L_{h_1,g}x := r(h_1, g) + \alpha P(h_1, g)x$ is a monotone contraction with fixed point $v^\alpha(h_1^\infty, g^\infty)$. By (10.18), we have $L_{h_1,g}v^\alpha(f^\infty, g^\infty) \leq v^\alpha(f^\infty, g^\infty)$, implying

$$v^\alpha(h_1^\infty, g^\infty) \leq v^\alpha(f^\infty, g^\infty) \text{ for all deterministic policies } h_1^\infty \text{ for player 1.} \tag{10.19}$$

Similarly, we can show for any policy h_2^∞ for player 2, we have $v^\alpha(f^\infty, h_2^\infty) \geq v^\alpha(f^\infty, g^\infty)$. Hence, $v^\alpha(h_1^\infty, g^\infty) \leq v^\alpha(f^\infty, g^\infty) \leq v^\alpha(f^\infty, h_2^\infty)$ for all pairs of deterministic policies (h_1^∞, h_2^∞), implying the optimality of the pair $(f\infty, g\infty)$. □

We shall give a policy improvement type algorithm to find optimal deterministic policies for discounted stochastic games with perfect information. The algorithm uses a certain lexicographic search in the policy improvement process. At each iteration one players policy is fixed and the other players policy changes just at one state. Two deterministic policies that differ just in one state are called *adjacent*. Because of this adjacent property we can compare the vectors $v^\alpha(f_1^\infty, g^\infty)$ and $v^\alpha(f_2^\infty, g^\infty)$, where f_1^∞ and f_2^∞ are adjacent: either $v^\alpha(f_2^\infty, g^\infty) > v^\alpha(f_1^\infty, g^\infty)$, i.e. $v_i^\alpha(f_2^\infty, g^\infty) \geq v_i^\alpha(f_1^\infty, g^\infty)$ for all $i \in S$ and $v_i^\alpha(f_2^\infty, g^\infty) > v_i^\alpha(f_1^\infty, g^\infty)$ for at least one i, or $v^\alpha(f_2^\infty, g^\infty) \leq v^\alpha(f_1^\infty, g^\infty)$. The next lemma shows this property.

Lemma 10.5
Let g^∞ be a deterministic policy for player 2, and let f_1^∞ and f_2^∞ be two adjacent deterministic policies for player 1. Then, either $v^\alpha(f_2^\infty, g^\infty) > v^\alpha(f_1^\infty, g^\infty)$ or $v^\alpha(f_2^\infty, g^\infty) \leq v^\alpha(f_1^\infty, g^\infty)$.

Proof
Consider the MDP induced by the fixed policy g^∞ for player 2 and assume that the policies f_1^∞ and f_2^∞ differ only in state k. Then,

$r_i(f_2, g) + \alpha \sum_j p_{ij}(f_2, g) v_j^\alpha(f_1^\infty, g^\infty) = r_i(f_1, g) + \alpha \sum_j p_{ij}(f_1, g) v_j^\alpha(f_1^\infty, g^\infty) = v_i^\alpha(f_1^\infty, g^\infty),\ i \neq k$.

If $r_k(f_2, g) + \alpha \sum_j p_{kj}(f_2, g) v_j^\alpha(f_1^\infty, g^\infty) > v_k^\alpha(f_1^\infty, g^\infty)$, then $L_{f_2,g}v^\alpha(f_1^\infty, g^\infty) > v^\alpha(f_1^\infty, g^\infty)$. Since $L_{f_2,g}$ is a monotone contraction with fixed point $v^\alpha(f_2^\infty, g^\infty)$, we obtain the strict inequality $v^\alpha(f_2^\infty, g^\infty) > v^\alpha(f_1^\infty, g^\infty)$.

Otherwise, we have $r_k(f_2, g) + \alpha \sum_j p_{kj}(f_2, g) v_j^\alpha(f_1^\infty, g^\infty) \leq v_k^\alpha(f_1^\infty, g^\infty)$, which implies that $L_{f_2,g}v^\alpha(f_1^\infty, g^\infty) \leq v^\alpha(f_1^\infty, g^\infty)$. In this case, we get the inequality $v^\alpha(f_2^\infty, g^\infty) \leq v^\alpha(f_1^\infty, g^\infty)$. □

The key property of lexicographic improvements are based on effectively going back and forth between the policies of both players in various iterations. Although the payoffs may be increasing some times and decreasing at other times, old deterministic policies are never revisited and hence there is no cycling. Since there are finitely many deterministic policies, the algorithm terminates in finite steps with an optimal pair of policies.

By Theorem 10.4 and Corollary 10.2, we only need to find a pair (f_*^∞, g_*^∞) of deterministic policies such that $v^\alpha(f^\infty, g_*^\infty) \leq v^\alpha(f_*^\infty, g_*^\infty) \leq v^\alpha(f_*^\infty, g^\infty)$ for all deterministic policies f^∞ and g^∞ player 1 and 2, respectively.

For any pair (f^∞, g^∞) of deterministic policies, we say that (h^∞, g^∞) is an *adjacent improvement of type 1* of (f^∞, g^∞) if:
(1) h differs from f in exactly one state; (2) $v^\alpha(h^\infty, g^\infty) > v^\alpha(f^\infty, g^\infty)$.
Similarly, we say that (f^∞, h^∞) is an *adjacent improvement of type 2* of (f^∞, g^∞) if:
(1) h differs from g in exactly one state; (2) $v^\alpha(f^\infty, h^\infty) < v^\alpha(f^\infty, g^\infty)$.
A pair of policies (f_2^∞, g_2^∞) will be called an *improvement* of (f_1^∞, g_1^∞) if it is an adjacent improvement of either type 1 or type 2.

Remark

Given a pair (f^∞, g^∞) of deterministic policies, we can find an adjacent improvement (h^∞, g^∞) of type 1 as follows. Consider the MDP induced by the fixed policy g^∞. Then, determine the sets $A(i, f),\ i \in S_2$, of improving actions as defined in (3.18).
If $A(i, f) = \emptyset$ for all $i \in S_2$, then there is no adjacent improvement of type 1.
Otherwise, take k such that $A(k, f) \neq \emptyset$ and set $h(i) := \begin{cases} a \in A(k, f) & \text{if } i = k; \\ \quad f(i) & \text{if } i \neq k. \end{cases}$
Similarly, an adjacent improvement (f^∞, h^∞) of type 2 can be found, if such improvement exists. We can also use linear programming, because an adjacent vertex corresponds with an adjacent improvement (see also Theorem 3.19).

Algorithm 10.1 *Policy iteration for discounted games with perfect information*
Input: Instance of a two-person stochastic game with perfect information.
Output: A pair (f_*^∞, g_*^∞) of deterministic optimal policies and the value vector.

```
1. Select an arbitrarily pair $(f_0^\infty, g_0^\infty)$ of deterministic policies; set $k := 0$.

2. Search lexicographically for an improvement of $(f_k^\infty, g_k^\infty)$, where lexicographically means
   looking first for an adjacent improvement of type 1 and, if such improvement does not
   exist, then for an adjacent improvement of type 2.

3. if an adjacent improvement $(h^\infty, g_k^\infty)$ of type 1 is found then
         begin $f_{k+1} := h$; $g_{k+1} := g_k$; $k := k + 1$; go to step 2 end
   else go to step 4.

4. if an adjacent improvement $(f_k^\infty, h^\infty)$ of type 2 is found then
         begin $f_{k+1} := f_k$; $g_{k+1} := h$; $k := k + 1$; go to step 2 end
   else go to step 5.

5. $(f_*^\infty, g_*^\infty) := (f_k^\infty, g_k^\infty)$ is a pair of deterministic optimal policies and $v^\alpha := v^\alpha(f_*^\infty, g_*^\infty)$ is
   the value vector of the game.
```

For what follows we require some notation. Let $s \in S$ be a fixed state and assume $s \in S_2$. For any nonempty action subset $A_1 \subseteq A(s)$ we write $\Gamma^s_{A_1}$ for the subgame in which only the actions in A_1 are allowed in state s. The value vector of the subgame $\Gamma^s_{A_1}$ is denoted by $v^\alpha(\Gamma^s_{A_1})$.

Lemma 10.6

Let $s \in S_2$ and let A_1 and A_2 be nonempty subsets of $A(s)$ with $A_1 \cap A_2 = \emptyset$. Then, either $v^\alpha(\Gamma^s_{A_1}) \geq v^\alpha(\Gamma^s_{A_2})$ or $v^\alpha(\Gamma^s_{A_1}) \leq v^\alpha(\Gamma^s_{A_2})$. We also have $v^\alpha_i(\Gamma^s_{A_1 \cup A_2}) = max\,\{v^\alpha_i(\Gamma^s_{A_1}), v^\alpha_i(\Gamma^s_{A_2})\}$ for all $i \in S$.

Proof

An optimal policy f^∞_* for player 1 in the game $\Gamma^s_{A_1 \cup A_2}$ will have either $f_*(s) \in A_1$ or $f_*(s) \in A_2$. Suppose $f_*(s) \in A_1$. Then, since $f_*(s) \in A1 \subset A_1 \cup A_2$, we have

$$v^\alpha(f^\infty_*, g^\infty) \geq v^\alpha(\Gamma^s_{A_1 \cup A_2}) \geq v^\alpha(\Gamma^s_{A_1}) \text{ for any policy } g^\infty \text{ for player 2,}$$

where the last inequality $v^\alpha(\Gamma^s_{A_1 \cup A_2}) \geq v^\alpha(\Gamma^s_{A_1})$ is trivial, because player 1 has in state s more actions in $\Gamma^s_{A_1 \cup A_2}$ than in $\Gamma^s_{A_1}$ and player 2 has in state s only one action. Also, any optimal policy g^∞_* for player 2 in the game $\Gamma^s_{A_1 \cup A_2}$ is found in $\Gamma^s_{A_1}$. Hence, $v^\alpha(\Gamma^s_{A_1}) = v^\alpha(\Gamma^s_{A_1 \cup A_2}) \geq v^\alpha(\Gamma^s_{A_2})$. In case $f_*(s) \in A_2$, we obtain similarly $v^\alpha(\Gamma^s_{A_2}) = v^\alpha(\Gamma^s_{A_1 \cup A_2}) \geq v^\alpha(\Gamma^s_{A_1})$. Therefore, we also have shown $v^\alpha_i(\Gamma^s_{A_1 \cup A_2}) = max\,\{v^\alpha_i(\Gamma^s_{A_1}), v^\alpha_i(\Gamma^s_{A_2})\}$ for all $i \in S$. □

An obvious analogy for player 2 exists and is formulated in the following lemma.

Lemma 10.7

Let $s \in S_1$ and let B_1 and B_2 be nonempty subsets of $B(s)$ with $B_1 \cap B_2 = \emptyset$. Then, either $v^\alpha(\Gamma^s_{B_1}) \geq v^\alpha(\Gamma^s_{B_2})$ or $v^\alpha(\Gamma^s_{B_1}) \leq v^\alpha(\Gamma^s_{B_2})$. We also have $v^\alpha_i(\Gamma^s_{B_1 \cup B_2}) = max\,\{v^\alpha_i(\Gamma^s_{B_1}), v^\alpha_i(\Gamma^s_{B_2})\}$ for all $i \in S$.

Theorem 10.6

Algorithm 10.1 terminates with an optimal pair of deterministic policies.

Proof

We need to find a saddle point, i.e. a pair of deterministic policies (f^∞_*, g^∞_*) such that

$$v^\alpha(f^\infty, g^\infty_*) \leq v^\alpha(f^\infty_*, g^\infty_*) \leq v^\alpha(f^\infty_*, g^\infty) \text{ for all deterministic policies } f^\infty \text{ and } g^\infty).$$

The proof is by induction on $n := \sum_{i=1}^N \{|A(i)|+|B(i)|\}$, the total number of actions in all states. Notice that $n \geq 2n$. If $n = 2N$, both players have only one policy, say f^∞_* and g^∞_*, and the pair (f^∞_*, g^∞_*) is obviously an optimal pair of deterministic policies.

By induction we shall assume that the algorithm terminates at a saddle point whenever $n \leq k$ and let $n = k + 1$. Let s be the largest value of i for which one player, say player 1, has more than one action. Then $s \in S_2$ and player 2 has one action in state s. It is sufficient to prove the theorem for this case as the proof for the case $s \in S_1$ is identical.

We now split the game at state s. The algorithm will pass through a sequence $a_1, a_2, \dots, a_m$ of actions in state s. Let $f_0(s) := a_1$, the first action in state s. Let $A_i \subseteq A(s)$ be defined by

$A_i := \{a_1, a_2, \dots, a_i\}$ for $i = 1, 2, \dots, m$. By the induction assumption the algorithm will reach a pair $(f^\infty_{n_1}, g^\infty_{n_1})$ which has no improvements in the subgame $\Gamma^s_{A_1}$ and which is an optimal pair for $\Gamma^s_{A_1}$, i.e.

$$v^\alpha(f^\infty, g^\infty_{n_1}) \le v^\alpha(f^\infty_{n_1}, g^\infty_{n_1}) \le v^\alpha(f^\infty_{n_1}, g^\infty) \text{ for all } f^\infty \text{ with } f(s) = a_1, \text{ and all } g^\infty.$$

Consider the MDP induced on the original game, when player 2 restricts to $g^\infty_{n_1}$. If no adjacent policy of $f^\infty_{n_1}$ gives a strict improvement in any state, then

$$v^\alpha(f^\infty, g^\infty_{n_1}) \le v^\alpha(f^\infty_{n_1}, g^\infty_{n_1}) \le v^\alpha(f^\infty_{n_1}, g^\infty) \text{ for all } f^\infty \text{ and all } g^\infty,$$

implying that the pair $(f^\infty_{n_1}, g^\infty_{n_1})$ is an optimal pair of deterministic policies in the original game. An adjacent policy of $(f^\infty_{n_1}$ which changes the action at a state other than state s is a policy available for player 1 in $\Gamma^s_{A_1}$ and is not better than $f^\infty_{n_1}$. Thus the only way any strict improvement occurs via some adjacent policy of $f^\infty_{n_1}$ has to be one which changes the action in state s. Let $f_{n_1+1}(s) := a_2$ be such action. Then,

$$v^\alpha(f^\infty_{n_1+1}, g^\infty_{n_1+1}) = v^\alpha(f^\infty_{n_1+1}, g^\infty_{n_1}) > v^\alpha(f^\infty_{n_1}, g^\infty_{n_1}).$$

After this improvement the algorithm continues and, by induction hypothesis, we shall reach a saddle point $(f^\infty_{n_2}, g^\infty_{n_2})$ in the subgame $\Gamma^s_{A_2\backslash A_1}$. By Lemma 10.6, we get $v^\alpha(f^\infty_{n_2}, g^\infty_{n_2}) = v^\alpha(\Gamma^s_{A_2})$. Lemma 10.4 we can conclude that there are no improvements of $(f^\infty_{n_2}, g^\infty_{n_2})$ in $\Gamma^s_{A_2}$, and further by Lemma 10.6that $v^\alpha(f^\infty_{n_1}, g^\infty_{n_1}) \le v^\alpha(f^\infty_{n_2}, g^\infty_{n_2})$ with strict inequality in some coordinate. If there are no improvements in the original game, then $(f^\infty_{n_2}, g^\infty_{n_2})$ is a saddle point of the original game and the algorithm terminates. Otherwise an improvement in state s to an action outside A_2 is available. By repeating the same arguments as before, we obtain a saddle point $(f^\infty_{n_3}, g^\infty_{n_3})$ in subgame $\Gamma^s_{A_3\backslash(A_1\cup A_2)}$, and we get $v^\alpha(f^\infty_{n_3}, g^\infty_{n_3}) = v^\alpha(\Gamma^s_{A_3})$. We can also conclude that there are no improvements of $(f^\infty_{n_3}, g^\infty_{n_3})$ in $\Gamma^s_{A_3}$ and that $v^\alpha(f^\infty_{n_2}, g^\infty_{n_2}) \le v^\alpha(f^\infty_{n_3}, g^\infty_{n_3})$ with strict inequality in some coordinate. Since there only a finite number of actions in state s, we end the algorithm with a saddle point $(f^\infty_{n_m}, g^\infty_{n_m})$ in subgame $\Gamma^s_{A_m\backslash(A_1\cup A_2\cdots A_{m-1})}$ for which there are no improvements of $(f^\infty_{n_m}, g^\infty_{n_m})$ in the original game. Hence, the pair $(f^\infty_{n_m}, g^\infty_{n_m})$ is a saddle point of the original game.

□

Example 10.2

$S = \{1, 2, 3, 4, 5\}$. $A(1) = A(2) = \{1, 2, 3\},\ A(3) = A(4) = A(5) = \{1\}$.
$B(1) = B(2) = \{1\}$, $B(3) = B(4) = \{1, 2, 3\}$, $B(5) = \{1\}$.
$r_1(1,1) = 5,\ r_1(2,1) = 4,\ r_1(3,1) = 3;\ r_2(1,1) = 6,\ r_2(2,1) = 1,\ r_2(3,1) = 0;$
$r_3(1,1) = 4,\ r_3(1,2) = 2,\ r_3(1,3) = 0;\ r_4(1,1) = 2,\ r_4(1,2) = 2,\ r_4(1,3) = 3;\ r_5(1,1) = 0.$
$p_{15}(1,1) = 1;\ p_{13}(2,1) = 0.2,\ p_{15}(2,1) = 0.8;\ p_{13}(3,1) = 0.6,\ p_{15}(3,1) = 0.4;$.
$p_{34}(1,1) = 0.9,\ p_{35}(1,1) = 0.1;\ p_{31}(1,2) = 1;\ p_{31}(1,3) = 0.3,\ p_{33}(1,3) = 0.2,\ p_{34}(1,3) = 0.5.$
$p_{42}(1,1) = 0.1;\ p_{43}(1,1) = 0.6,\ p_{44}(1,1) = 0.3;\ p_{41}(1,2) = 0.2,\ p_{43}(1,2) = 0.4,\ p_{44}(1,2) = 0.4.$
$p_{44}(1,3) = 0.9;\ p_{45}(1,3) = 0.1;\ p_{52}(1,1) = 0.1,\ p_{53}(1,1) = 0.2,\ p_{53}(1,1) = 0.3;\ p_{55}(1,1) = 0.4.$
The other transition probabilities are 0. $\alpha = 0.999$.

For player 1 only in the states 1 and 2 there are more actions; player 2 has only in the states 3 and 4 a choice of actions. Therefore, we specify only the actions $f(1),\ f(2),\ g(3)$ and $g(4)$.

We start with $f_0(1) = f_0(2) = 1$ and $g_0(3) = g_0(4) = 1$.
The vector $v^\alpha(f_0^\infty, g_0^\infty) = (25623.8,\ 25624.8,\ 25626.4,\ 25625.3,\ 25621.3)$.
Iteration 1:
We have an improvement of type 1 which yields $f_1(1) = 3,\ f_1(2) = 1$ and $g_1(3) = g_1(4) = 1$.
The vector $v^\alpha(f_1^\infty, g_1^\infty) = (25624.8,\ 25624.8,\ 25626.4,\ 25625.3,\ 25621.3)$.
Iteration 2:
We have an improvement of type 2 which gives $f_2(1) = 3,\ f_2(2) = 1$ and $g_2(3) = 2,\ g_2(4) = 1$.
The vector $v^\alpha(f_2^\infty, g_2^\infty) = (19259.9,\ 19261.2,\ 19260.0,\ 19260.2,\ 19257.1)$.
Iteration 3:
We have an improvement of type 1: $f_3(1) = 1,\ f_3(2) = 1$ and $g_3(3) = 2,\ g_3(4) = 1$.
The vector $v^\alpha(f_3^\infty, g_3^\infty) = (19771.2,\ 19772.2,\ 19771.3,\ 19771.4,\ 19768.2)$.
Iteration 4:
We have an improvement of type 2: $f_4(1) = 1,\ f_4(2) = 1$ and $g_4(3) = 2,\ g_4(4) = 2$.
The vector $v^\alpha(f_4^\infty, g_4^\infty) = (19601.2,\ 19602.2,\ 19601.2,\ 19601.2,\ 19598.1)$.
Iteration 5:
We have an improvement of type 2: $f_5(1) = 1,\ f_5(2) = 1$ and $g_5(3) = 3,\ g_5(4) = 2$.
The vector $v^\alpha(f_5^\infty, g_5^\infty) = (15060.1,\ 15061.1,\ 15057.7,\ 15059.4,\ 15057.0)$.
Iteration 6:
We have again an improvement of type 2: $f_6(1) = 1,\ f_6(2) = 1$ and $g_6(3) = 3,\ g_6(4) = 1$.
The vector $v^\alpha(f_6^\infty, g_6^\infty) = (14128.3,\ 14129.3,\ 14125.8,\ 14127.2,\ 14124.7)$.
Iteration 7:
There are no improvements, so the pair (f_6^∞, g_6^∞) is an optimal pair of policies and the value vector of the game is $(14128.3,\ 14129.3, 14125.8,\ 14127.2,\ 14124.7)$.

Next, we shall discuss Blackwell optimality for stochastic games with perfect information. The approach is similar to the analysis in section 7.7.

A policy R_1^* is *Blackwell optimal* for player 1 if $v^\alpha(R_1^*, R_2) \geq inf_{R_2}\, sup_{R_1}\, v^\alpha(R_1, R_2)$ for all policies R_2 and for all $\alpha \in [\alpha_1, 1)$ for some α_1. Similarly, a policy R_2^* is *Blackwell optimal* for player 2 if $v^\alpha(R_1, R_2^*) \leq sup_{R_1}\, inf_{R_2}\, v^\alpha(R_1, R_2)$ for all policies R_1 and for all $\alpha \in [\alpha_2, 1)$ for some α_2.

Let $F(R)$ be the completely ordered field of rational functions with real coefficients. The ordering is induced by $\frac{p(x)}{q(x)} >_l 0$ if and only if $d(p)d(q) > 0$, where the dominating coefficient $d(p)$ of a polynomial $p(x) = a_0 + a_1x + \cdots + a_nx^n$ is the coefficient a_k with $k := \min\{i \mid a_i \neq 0\}$. Two rational functions $\frac{p(x)}{q(x)}$ and $\frac{r(x)}{s(x)}$ are identical, i.e. $\frac{p}{q} =_l \frac{r}{s}$ if and only if $p(x)s(x) = r(x)q(x)$ for all $x \in \mathbb{R}$.

Let π^∞ and σ^∞ be two stationary policies for player 1 and 2, respectively. Then,

$$v^\alpha(\pi^\infty, \sigma^\infty) = r(\pi, \sigma) + \alpha P(\pi, \sigma) v^\alpha(\pi^\infty, \sigma^\infty). \tag{10.20}$$

Instead of the discount factor α we can also use the interest rate ρ, where $\alpha(1+\rho) = 1$, and write $v^\rho(\pi^\infty, \sigma^\infty)$ instead of $v^\alpha(\pi^\infty, \sigma^\infty)$. Hence, we have

$$(1+\rho)v^\rho(\pi^\infty, \sigma^\infty) = (1+\rho)r(\pi, \sigma) + P(\pi, \sigma)v^\rho(\pi^\infty, \sigma^\infty). \tag{10.21}$$

By solving the system $(1+\rho)x = (1+\rho)r(\pi,\sigma) + P(\pi,\sigma)x$ by Cramers rule, it is evident that

$$v_i^\rho(\pi^\infty,\sigma^\infty) \in F(\mathbb{R}) \text{ for all states } i \in S. \tag{10.22}$$

Lemma 10.8

Let π_*^∞ and σ_*^∞ be stationary Blackwell optimal policies for player 1 and 2. Then, there exists a vector v^ρ with $v_i^\rho \in F(\mathbb{R}),\ i \in S$, such that $v^\rho(R_1,\sigma_*^\infty) \leq_l v^\rho(\rho_*^\infty,\sigma_*^\infty) =_l v^\rho \leq_l v^\rho(\rho_*^\infty,R_2)$ for all policies R_1 and R_2 for player 1 and 2, respectively.

Proof

By hypothesis, there exists $\rho_0 > 0$ such that π_*^∞ and σ_*^∞ is an optimal pair of policies for all interest rates $\rho \in (0,\rho_0]$. Hence, $v^\rho(R_1,\sigma_*^\infty) \leq v^\rho \leq v^\rho(\rho_*^\infty,R_2)$ for all policies R_1 and R_2 and for all interest rates $\rho \in (0,\rho_0]$, where v^ρ is some vector. Therefore, $v^\rho(\rho_*^\infty,\sigma_*^\infty) = v^\rho$ for all interest rates $\rho \in (0,\rho_0]$. By (10.22), $v_i^\rho(\pi^\infty,\sigma^\infty) \in F(\mathbb{R})$ for all states $i \in S$. This produces $v^\rho(R_1,\sigma_*^\infty) \leq_l v^\rho(\rho_*^\infty,\sigma_*^\infty) =_l v^\rho \leq_l v^\rho(\rho_*^\infty,R_2)$ and $v_i^\rho \in F(\mathbb{R}),\ i \in S$. □

Remark

Generally, the components of the value vector of a discounted stochastic game are no elements of $F(\mathbb{R})$, but belong to the field of Puiseux series for ρ sufficiently small (see [25]). A vector $v^\rho \in F(\mathbb{R})$, satisfying $v^\rho(R_1,\sigma_*^\infty) \leq v^\rho \leq v^\rho(\rho_*^\infty,R_2)$ for all policies R_1 and R_2 and for all interest rates $\rho \in (0,\rho_0]$ is called the *Blackwell value vector* of the game.

The following property holds, whose proof is analogous to the proof of Lemma 10.5.

Lemma 10.9

Let g^∞ be a deterministic policy for player 2, and let f_1^∞ and f_2^∞ be two adjacent deterministic policies for player 1. Then, either $v^\rho(f_2^\infty,g^\infty) >_l v^\rho(f_1^\infty,g^\infty)$ or $v^\rho(f_2^\infty,g^\infty) \leq_l v^\rho(f_1^\infty,g^\infty)$, which means that the two vectors are partially ordered.

Lemma 10.9 allows us to give the following definition. Let (f^∞,g^∞) be a pair of deterministic policies for player 1 and 2, respectively. We call h^∞ a *Blackwell adjacent improvement of type 1* for player 1 if and only if:
(1) h differs from f only in one state; (2) $v^\rho(h^\infty,g^\infty) >_l v^\rho(f^\infty,g^\infty)$.
Similarly, h^∞ is a *Blackwell adjacent improvement of type 2* for player 2 if and only if:
(1) h differs from g only in one state; (2) $v^\rho(f^\infty,h^\infty) <_l v^\rho(f^\infty,g^\infty)$.
From this definition the following property holds.

Lemma 10.10

A pair of deterministic policies (f_*^∞,g_*^∞) is Blackwell optimal if and only if no Blackwell adjacent improvements is possible for both players.

From the result of Lemma 10.4 we can derive the analogous property in the ordered field $F(\mathbb{R})$.

Lemma 10.11

A pair of deterministic policies (f_*^∞, g_*^∞) is Blackwell optimal if and only if $v^\rho(f_*^\infty, g_*^\infty) =_l v^\rho$, where v^ρ is the Blackwell value vector of the game.

Proof

For the if-part assume that $v^\rho(f_*^\infty, g_*^\infty) =_l v^\rho$, where v^ρ is the Blackwell value vector of the game. Then, there exists a $\rho^* > 0$ such that $v^\rho(f_*^\infty, g_*^\infty) = v^\rho$ for all $\rho \in (0, \rho^*)$. Hence, by Lemma 10.4, (f_*^∞, g_*^∞) is a pair of optimal policies for all $\rho \in (0, \rho^*)$, which means that they are Blackwell optimal.

Conversely, let (f_*^∞, g_*^∞) be a pair of deterministic Blackwell optimal policies. Then, we have $v^\rho(f_*^\infty, R_2) \geq v^\rho \geq v^\rho(R_1, g_*^\infty)$ for all policies R_1, R_2 and for all $\rho \in (0, \rho^*)$ for some $\rho^* > 0$. Therefore, $v^\rho(f_*^\infty, g_*^\infty) = v^\rho$ for all $\rho \in (0, \rho^*)$, i.e. $v^\rho(f_*^\infty, g_*^\infty) =_l v^\rho$, where v^ρ is the Blackwell value vector of the game. □

The proof of next lemma is analogous to the proof of Lemma 10.6 and is the version in the field $F(\mathbb{R})$.

Lemma 10.12

Let $s \in S_2$ and let A_1 and A_2 be nonempty subsets of $A(s)$ with $A_1 \cap A_2 = \emptyset$. Then, either $v^\alpha(\Gamma^s_{A_1}) \geq_l v^\alpha(\Gamma^s_{A_2})$ or $v^\alpha(\Gamma^s_{A_1}) \leq_l v^\alpha(\Gamma^s_{A_2})$. We also have $v_i^\alpha(\Gamma^s_{A_1 \cup A_2}) =_l max\, \{v_i^\alpha(\Gamma^s_{A_1}), v_i^\alpha(\Gamma^s_{A_2})\}$ for all $i \in S$.

We shall present an algorithm to find Blackwell optimal policies for both players and the Blackwell value vector of the game.

Algorithm 10.2 *Blackwell optimality for discounted games with perfect information*
Input: Instance of a two-person stochastic game with perfect information.
Output: A pair (f_*^∞, g_*^∞) of deterministic Blackwell optimal policies and the Blackwell vector vector.

1. Select an arbitrarily deterministic policy g_*^∞.
2. Determine in the field $F(\mathbb{R})$ an optimal solution $x^*(\rho)$ of the program
$$max \left\{ \textstyle\sum_{(i,a)} r_i(a, g_*) \cdot x_{ia}(\rho) \;\middle|\; \begin{array}{rcl} \sum_{(i,a)}\{(1+\rho)\delta_{ij} - p_{ij}(a, g_*)\} \cdot x_{ia}(\rho) & =_l & 1,\ j \in S \\ x_{ia}(\rho) & \geq_l & 0,\ (i,a) \in S \times A \end{array} \right\}.$$
3. Take f_*^∞ such that $x^*_{i,f_*(i)}(\rho) > 0$ for all $i \in S$.
4. Determine in the field $F(\mathbb{R})$ the simplex tableau corresponding to the program
$$max \left\{ -\textstyle\sum_{(i,b)} r_i(f_*, b) \cdot y_{ib}(\rho) \;\middle|\; \begin{array}{rcl} \sum_{(i,b)}\{(1+\rho)\delta_{ij} - p_{ij}(f_*, b)\} \cdot y_{ib}(\rho) & =_l & 1,\ j \in S \\ y_{ib}(\rho) & \geq_l & 0,\ (i,b) \in S \times B \end{array} \right\}.$$
with as basic variables $y_{ig_*(i)}(\rho),\ i \in S$.

5. **if** all shadow prices, corresponding to the variables $y_{ib}(\rho)$, in this tableau are $\geq_l 0$,

   then go to step 7

   else go to step 6.

6. Determine an adjacent improvement h^∞ of g_*^∞; $g_* := h$; **go to** step 2.

7. (f_*^∞, g_*^∞) is a pair of deterministic optimal policies and $v^\rho(f_*^\infty, g_*^\infty)$ is the value vector of the game.

Remark

The Blackwell value vector $v^\rho(f_*^\infty, g_*^\infty)$ can be found in the simplex tableau of step 4 in the last iteration. By the same argument as in the discounted case with a fixed discount factor, we have the property that the element $v_i^\rho(f_*^\infty, g_*^\infty)$ is the shadow price of the ith artificial variable of the program. So, no additional calculation is required to obtain the value vector.

Theorem 10.7

Algorithm 10.2 terminates with a pair of Blackwell optimal deterministic policies and with the Blackwell value vector.

Proof

The proof follows the lines analogous the one in the real field $\mathbb{R}$ (see Theorem 10.6). It proceeds by induction on n, where $n := \sum_{i=1}^N \{|A(i)| + |B(i)|\}$, the total number of actions in all states, and exploits Lemmas 10.10 and 10.12. The main difference with Algorithm 10.1 is that the policy for player 1 is not constructed by adjacent improvements of type 1, but that does not effect the correctness of the proof. □

Similar to MDPs, Blackwell optimal policies are also optimal for the criterion of average rewards as the next theorem shows.

Theorem 10.8

A pair of Blackwell optimal deterministic policies is also optimal for the criterion of average rewards.

Proof

Let (f_*^∞, g_*^∞) be a pair of Blackwell optimal deterministic policies. Then, we have for all deterministic policies f^∞ and g^∞: $v^\alpha(f^\infty, g_*^\infty) \leq v^\alpha(f_*^\infty, g_*^\infty) \leq v^\alpha(f_*^\infty, g^\infty)$ for all $\alpha \in (\alpha_0, 1)$ for some α_0. Hence, we also have $(1-\alpha)v^\alpha(f^\infty, g_*^\infty) \leq (1-\alpha)v^\alpha(f_*^\infty, g_*^\infty) \leq (1-\alpha)v^\alpha(f_*^\infty, g^\infty)$ for all deterministic policies f^∞ and g^∞, and for all $\alpha \in (\alpha_0, 1)$ for some α_0. Taking $\alpha \uparrow 1$ and using $\lim\,(1-\alpha)v^\alpha(f^\infty, g^\infty) = \phi(f^\infty, g^\infty)$, we get $\phi(f^\infty, g_*^\infty) \leq \phi(f_*^\infty, g_*^\infty) \leq \phi(f_*^\infty, g^\infty)$ for all deterministic policies f^∞ and g^∞, i.e. (f_*^∞, g_*^∞) is a pair of average optimal deterministic policies.

□

Remark

Let SP, in the two optimal tableaux obtained in the steps 2 and 4 of the last iteration of Algorithm 10.2, be the set of shadow prices corresponding to the variables $x_{ia}(\rho)$ and $y_{ib}(\rho)$, respectively. Let $\rho^* > 0$ the smallest positive simple root of the elements of SP. Then, the pair (f_*^∞, g_*^∞) of Blackwell optimal deterministic policies obtained in step 7 of the algorithm, is a pair of discounted optimal deterministic policies for all discount factors $\alpha \in [\alpha_*, 1)$, where $\alpha_* := \frac{1}{1+\rho^*}$.

10.2.2 Mathematical programming

A vector $v \in \mathbb{R}^N$ is called *superharmonic* if there exists a policy $\rho^\infty \in \Gamma$ such that

$$v_i \geq r_i(a,\rho) + \alpha \textstyle\sum_j p_{ij}(a,\rho)v_j,\ a \in A(i),\ i \in S.$$

A vector $v \in \mathbb{R}^N$ is called *subharmonic* if there exists a policy $\pi^\infty \in \Pi$ such that

$$v_i \leq r_i(\pi,b) + \alpha \textstyle\sum_j p_{ij}(\pi,\rho)v_j,\ b \in B(i),\ i \in S.$$

Theorem 10.9

(1) The value vector v^α is the smallest superharmonic vector.

(2) The value vector v^α is the largest subharmonic vector.

Proof

Let $(\pi^*)^\infty$ and $(\rho^*)^\infty$ be the policies mentioned in Theorem 10.5. If player 2 uses policy $(\rho^*)^\infty$, then the game becomes an MDP. We know from Theorem 3.16 that $x := \sup_{R_1} v^\alpha\big(R_1, (\rho^*)^\infty\big)$ is the smallest superharmonic vector. Since $v^\alpha = v^\alpha\big((\pi^*)^\infty, (\rho^*)^\infty\big)$, we have $x \geq v^\alpha$. On the other hand, it follows from the proof of Corollary 10.1 that $x = sup_{R_1} v^\alpha\big(R_1, (\rho^*)^\infty\big) \leq v^\alpha$.

The proof of part (2) is analogous to the proof of part (1). □

Consider the two nonlinear programs

$$min\left\{\sum_i v_i \;\middle|\; \begin{array}{rrl} \sum_j\{\delta_{ij} - \alpha\sum_b p_{ij}(a,b)\rho_{ib}\}v_j & - \quad \sum_b r_i(a,b)\rho_{ib} & \geq 0,\ a \in A(i),\ i \in S \\ & \sum_b \rho_{ib} & = 1,\ i \in S \\ & \rho_{ib} & \geq 0,\ b \in B(i),\ i \in S \end{array}\right\} \tag{10.23}$$

and

$$max\left\{\sum_i w_i \;\middle|\; \begin{array}{rrl} \sum_j\{\delta_{ij} - \alpha\sum_a p_{ij}(a,b)\pi_{ia}\}w_j & - \quad \sum_a r_i(a,b)\pi_{ia} & \leq 0,\ b \in B(i),\ i \in S \\ & \sum_a \pi_{ia} & = 1,\ i \in S \\ & \pi_{ia} & \geq 0,\ a \in A(i),\ i \in S \end{array}\right\}. \tag{10.24}$$

Theorem 10.10

The nonlinear programs (10.23) and (10.24) have both optimal solutions, say (v^*, ρ^*) and (w^*, π^*). Furthermore, $v^* = w^* = v^\alpha$ and $(\pi^*)^\infty$ and $(\rho^*)^\infty$ are optimal policies for player 1 and player 2.

Proof

From Theorem 10.9 it follows that both nonlinear programs have optimal solutions and that $v^* = w^* = v^\alpha$. The constraints of the programs imply

$$r(\pi, \rho^*) + \alpha P(\pi, \rho^*)v^\alpha \leq v^\alpha \leq r(\pi^*, \rho) + \alpha P(\pi^*, \rho)v^\alpha \text{ for all } \pi \text{ and } \rho.$$

Therefore, $\{I - \alpha P(\pi, \rho^*)\}v^\alpha \geq r(\pi, \rho^*)$ and $\{I - \alpha P(\pi^*, \rho)\}v^\alpha \leq r(\pi^*, \rho)$ for all π and ρ. Hence, $v^\alpha(\pi^\infty, (\rho^*)^\infty) = \{I - \alpha P(\pi, \rho^*)\}^{-1}r(\pi, \rho^*) \leq v^\alpha \leq \{I - \alpha P(\pi^*, \rho)\}^{-1}r(\pi^*, \rho) = v^\alpha((\pi^*)^\infty, \rho^\infty)$ for all $\pi^\infty \in \Pi$ and $\rho^\infty \in \Gamma$. Then, by the proof of Corollary 10.1, it follows that $(\pi^*)^\infty$ and $(\rho^*)^\infty$ are optimal policies for player 1 and player 2. □

10.2.3 Iterative methods

Since T is a contraction with fixed point the value vector v^α, it follows that the value iteration algorithm stated below approximately computes v^α.

Algorithm 10.3 *Value Iteration for discounted games*

Input: Instance of a two-person stochastic game and some $\varepsilon > 0$.

Output: An ε-approximation of the value vector v^α and a pair $\big((\pi^*)^\infty, (\rho^*)^\infty\big)$ of stationary 2ε-optimal policies.

```
1. Select $x \in \mathbb{R}^N$ arbitrarily.

2. for all $i \in S$ do
     begin compute the matrix $M_x[i]$ with entries $r_i(a,b) + \alpha \sum_j p_{ij}(a,b)x_j,\ a \in A(i),\ b \in B(i)$;
         $y_i := val(M_x[i])$
     end

3. if $\|y - x\|_\infty > (1-\alpha)\alpha^{-1}\varepsilon$ then begin $x := y$; go to step 2 end
   else for each $i \in S$ do
       begin
         determine an optimal strategy $\pi^*_{ia},\ a \in A(i)$, for player 1 in the matrix game $M_x[i]$;
         determine an optimal strategy $\rho^*_{ib},\ b \in B(i)$, for player 2 in the matrix game $M_x[i]$
       end

4. $y$ is an $\varepsilon$-approximation of the value vector $v^\alpha$ and $(\pi^*)^\infty$ and $(\rho^*)^\infty$ are $2\varepsilon$-optimal policies
   for player 1 and 2, respectively (STOP).
```

Theorem 10.11

Algorithm 10.3 is correct.

Proof

Since T is a monotone contraction with contraction factor α and fixed point v^α, it follows from Corollary 3.1 that $\|v^\alpha - y\|_\infty \leq \alpha(1-\alpha)^{-1}\|y-x\|_\infty \leq \varepsilon$, i.e. y is a ε-approximation of the value vector v^α. For any two policies $\pi^\infty \in \Pi$ and $\rho^\infty \in \Gamma$, we define the operator $L_{\pi,\rho} : \mathbb{R}^N \to \mathbb{R}^N$ by

$$L_{\pi,\rho}x := r(\pi,\rho) + \alpha P(\pi,\rho)x.$$

It is straightforward to show that $L_{\pi,\rho}$ is a monotone contraction with contraction factor α and fixed point $v^\alpha(\pi^\infty,\rho^\infty)$. Because $(\pi^*)^\infty$ is an optimal policy in the matrix games of step 2 of Algorithm 10.3, which have values $y_i,\ i \in S$, we can write

$$L_{\pi^*,\rho}x = r(\pi^*,\rho) + \alpha P(\pi^*,\rho)x \geq y = x + (y-x) \geq x - \|y-x\|_\infty \cdot e \geq x - \frac{1-\alpha}{\alpha}\varepsilon \cdot e \quad (10.25)$$

Hence, applying $L_{\pi^*,\rho}$ to (10.25), $L^2_{\pi^*,\rho}x \geq L_{\pi^*,\rho}\{x - \frac{1-\alpha}{\alpha}\varepsilon\cdot e\} = L_{\pi^*,\rho}x - (1-\alpha)\varepsilon\cdot e \geq y - (1-\alpha)\varepsilon\cdot e$. By iterating (10.25), we obtain $L^n_{\pi^*,\rho}x \geq y - (1-\alpha)\{1+\alpha+\cdots+\alpha^{n-2}\}\varepsilon\cdot e$. Taking the limit for $n\to\infty$ yields $v^\alpha\big((\pi^*)^\infty,\rho\big) \geq y - \varepsilon\cdot e \geq v^\alpha - 2\varepsilon\cdot e$. Since the fixed stationary policy $(\pi^*)^\infty$ induces an MDP, we also have $v^\alpha\big((\pi^*)^\infty, R_2\big) \geq v^\alpha - 2\varepsilon\cdot e$, i.e. $(\pi^*)^\infty$ is an 2ε-optimal policy for player 1. Similarly, it can be shown that $(\rho^*)^\infty$ is an 2ε-optimal policy for player 2. □

Example 10.1 (continued)

We apply Algorithm 10.3 with $\varepsilon = 0.2$ $\big((1-\alpha)\alpha^{-1}\varepsilon = 0.2\big)$ and starting value $x = (2,2)$.

Iteration 1:

$i=1$: $M_x[1] = \begin{pmatrix} \frac{3}{2} & 2 \\ 4 & \frac{5}{2} \end{pmatrix}$; $y_1 = val\,M_x[1] = \frac{5}{2}$; $i=2:$ $M_x[2]=(2)$; $y_2 = val\,M_x[2] = 2$;

$\|y-x\|_\infty = 0.5 > 0.2$; $x = \big(\frac{5}{2},2\big)$.

Iteration 2:

$i=1$: $M_x[1] = \begin{pmatrix} \frac{19}{12} & 2 \\ 4 & \frac{21}{8} \end{pmatrix}$; $y_1 = val\,M_x[1] = \frac{21}{8}$; $i=2:$ $M_x[2]=(2)$; $y_2 = val\,M_x[2] = 2$;

$\|y-x\|_\infty = 0.125 \leq 0.2$;

$i=1$: $\pi^*_{11}=0,\ \pi^*_{12}=1$; $\rho^*_{11}=0,\ \rho^*_{12}=1$; $i=2$: $\pi^*_{21}=1$; $\rho^*_{21}=1$.

$\big(\frac{21}{8},2\big)$ is a 0.2-approximation of the value vector v^α; f_*^∞ with $f(1)=2,\ f(2)=1$ is a 0.4-optimal policy for player 1 and g_*^∞ with $g(1)=2,\ g(2)=1$ is a 0.4-optimal policy for player 2.

Algorithm 10.3 does not utilize the information contained in the optimal strategies of the matrix games at each iteration. The next algorithm attempts to improve the basis scheme of Algorithm 10.3 by using these optimal strategies. This algorithm iterates in both value space and policy space.

Algorithm 10.4 *Value Iteration for discounted games (Modification 1)*

Input: Instance of a two-person stochastic game and some $\varepsilon > 0$.

Output: An ε-approximation of the value vector v^α and a pair $\big((\pi^*)^\infty, (\rho^*)^\infty\big)$ of stationary 2ε-optimal policies.

```
1. Select a stationary policy $(\rho^*)^\infty$ for player 2.
2. Compute the value vector $x$ of the MDP induced by the policy $(\rho^*)^\infty$,
   i.e. $x := \max_{f^\infty \in C(D)} v^\alpha\big(f^\infty, (\rho^*)^\infty\big)$.
3. for all $i \in S$ do
     begin compute the matrix $M_x[i]$ with entries $r_i(a,b)+\alpha\sum_j p_{ij}(a,b)x_j,\ a \in A(i),\ b \in B(i)$;
           determine an optimal stationary policy $\rho^*$ for player 2 in the matrix game $M_x[i]$;
           $y_i := val(M_x[i])$
     end
4. if $\|y - x\|_\infty > (1-\alpha)\alpha^{-1}\varepsilon$ then go to step 2
5. else for each $i \in S$ do
           determine an optimal strategy $\pi^*_{ia},\ a \in A(i)$, for player 1 in the matrix game $M_x[i]$.
6. $y$ is an $\varepsilon$-approximation of the value vector $v^\alpha$ and $(\pi^*)^\infty$ and $(\rho^*)^\infty$ are $2\varepsilon$-optimal policies
   for player 1 and 2, respectively (STOP).
```

Example 10.1 (continued)

We apply Algorithm 10.4 with $\varepsilon = 0.2$ and starting policy $\rho^*_{11} = \rho^*_{12} = \frac{1}{2};\ \rho^*_{21} = 1$.

Iteration 1:

$r_1(1,\rho^*) = \frac{3}{4},\ r_1(2,\rho^*) = \frac{5}{4},\ r_2(1,\rho^*) = 1.$

$p_{11}(1,\rho^*) = \frac{1}{6},\ p_{12}(1,\rho^*) = \frac{5}{6};\ p_{11}(2,\rho^*) = \frac{1}{4},\ p_{12}(2,\rho^*) = \frac{3}{4};\ p_{21}(2,\rho^*) = 0,\ p_{22}(2,\rho^*) = 1.$

$x = \big(\frac{16}{7}, 2\big).$

$i = 1$: $M_x[1] = \begin{pmatrix} \frac{50}{21} & 2 \\ 4 & \frac{18}{7} \end{pmatrix}$; $\rho^*_{11} = 0,\ \rho^*_{12} = 1;\ y_1 = val\, M_x[1] = \frac{18}{7}.$

$i = 2:\ M_x[2] = (2);\ \rho^*_{21} = 1;\ y_2 = val(2) = 2.$

$\|y - x\|_\infty = \frac{2}{7} > 0.2.$

Iteration 2:

$r_1(1,\rho^*) = 1,\ r_1(2,\rho^*) = \frac{3}{2},\ r_2(1,\rho^*) = 1.$

$p_{11}(1,\rho^*) = \frac{1}{6},\ p_{12}(1,\rho^*) = \frac{5}{6};\ p_{11}(2,\rho^*) = \frac{1}{2},\ p_{12}(2,\rho^*) = \frac{1}{2};\ p_{21}(2,\rho^*) = 0,\ p_{22}(2,\rho^*) = 1.$

$x = \big(\frac{8}{3}, 2\big).$

$i = 1$: $M_x[1] = \begin{pmatrix} \frac{29}{18} & 2 \\ 4 & \frac{8}{3} \end{pmatrix}$; $\rho^*_{11} = 0,\ \rho^*_{12} = 1;\ y_1 = val\, M_x[1] = \frac{8}{3}.$

$i = 2:\ M_x[2] = (2);\ \rho^*_{21} = 1;\ y_2 = val(2) = 2.$

$\|y - x\|_\infty = 0 \leq 0.2$.
$i = 1: \; \pi^*_{11} = 0, \; \pi^*_{12} = 1; \; i = 2: \; \pi^*_{21} = 1$.
$\left(\frac{8}{3}, 2\right)$ is a 0.2-approximation of the value vector v^α; f_*^∞ with $f(1) = 2, \; f(2) = 1$ is a 0.4-optimal policy for player 1 and g_*^∞ with $g(1) = 2, \; g(2) = 1$ is a 0.4-optimal policy for player 2.

Theorem 10.12
Algorithm 10.4 is correct.

Proof
Let x^n and y^n be the values of x and y in iteration n; let f_n^∞ be the optimal policy obtained in step 2 in iteration n; let π^n and ρ^n be the optimal mixed strategies of the two players obtained in the steps 4 and 3, respectively, in iteration n. Then,

$$x^n = r(f_n, \rho^{n-1} + \alpha P(f_n, \rho^{n-1})x^n \geq r(\pi, \rho^{n-1}) + \alpha P(\pi, \rho^{n-1})x^n, \; \pi^\infty \in \Pi. \tag{10.26}$$

and

$$r(\pi^n, \rho) + \alpha P(\pi^n, \rho)x^n \geq r(\pi^n, \rho^n) + \alpha P(\pi^n, \rho^n)x^n = y^n \geq r(\pi, \rho^n) + \alpha P(\pi, \rho^n)x^n, \; \pi^\infty \in \Pi, \; \rho^\infty \in \Gamma. \tag{10.27}$$

Hence, $y^n \leq r(\pi^n, \rho^{n-1}) + \alpha P(\pi^n, \rho^{n-1})x^n \leq x^n$. From (10.26) and the monotonicity of $L_{\pi,\rho}$ it follows that $y^n \geq L_{f_{n+1},\rho^n} x^n \geq L_{f_{n+1},\rho^n} x^n$, implying $y^n \geq v^\alpha\big(f_{n+1}^\infty, (\rho^n)^\infty\big)x^n = x^{n+1}$.
So, we obtain the sequence $x^0 \geq y^0 \geq x^1 \geq y^1 \geq \cdots \geq x^n \geq y^n \geq \cdots$, bounded below by $\frac{-1}{1-\alpha} \cdot max_{i,a,b} |r_i(a, b)| \cdot e$. Therefore, $\lim_{n\to\infty} x^n = \lim_{n\to\infty} y^n = x^*$ for some $x^* \in \mathbb{R}^N$.
Since the sets Π and Γ are compact, there are subsequences $\{n_k\}_{k=1}^\infty$ such that $\pi^{n_k} \to \pi^*$ and $\rho^{n_k} \to \rho^*$ for some $(\pi^*)^\infty \in \Pi$ and $(\rho^*)^\infty \in \Gamma$. From (10.26) it follows that

$$r(\pi^*, \rho) + \alpha P(\pi^*, \rho)x^* \geq x^* \geq r(\pi, \rho^*) + \alpha P(\pi, \rho^*)x^*, \; \pi^\infty \in \Pi, \; \rho^\infty \in \Gamma,$$

implying $v^\alpha\big((\pi^*)^\infty, \rho^\infty\big) \geq x^* \geq v^\alpha\big(\pi^\infty, (\rho^*)^\infty\big), \; \pi^\infty \in \Pi, \; \rho^\infty \in \Gamma$. Hence, x^* is the value, and $(\pi^*)^\infty$ and $(\rho^*)^\infty$ are optimal policies and the algorithm terminates.
Let x and y be the vectors at termination of the algorithm. Then, we can write

$$\|y - v^\alpha\|_\infty = \|Tx - Tv^\alpha\|_\infty \leq \alpha\|x - v^\alpha\|_\infty \leq \alpha\|x - y\|_\infty + \alpha\|y - v^\alpha\|_\infty.$$

Therefore, $\|y - v^\alpha\|_\infty \leq \alpha(1-\alpha)^{-1}\|x - y\|_\infty < \varepsilon$ at termination, i.e. y is an ε-approximation of the value vector v^α. Similarly as in the proof of Theorem 10.11 we can show that the policies $(\pi^*)^\infty$ and $(\rho^*)^\infty$, defined in the steps 4a and 3b, respectively, are 2ε-optimal policy for the players. □

In the next algorithm the optimal mixed strategies of the two players obtained by the matrix game $M_x[i]$ are used in another way.

Algorithm 10.5 *Value Iteration for discounted games (Modification 2)*
Input: Instance of a two-person stochastic game and some $\varepsilon > 0$.
Output: An ε-approximation of the value vector v^α and a pair $\big((\pi^*)^\infty, (\rho^*)^\infty\big)$ of stationary $\big\{1 + \frac{2}{\beta(1+\alpha)}\big\}\, \varepsilon$-optimal policies, where $\beta := \frac{\alpha}{1-\alpha} \cdot max_i\big\{ \sum_j \{max_{(a,b)}\, p_{ij}(a, b) - min_{(a,b)}\, p_{ij}(a, b)\}\big\}$.

```
1. Select $x \in \mathbb{R}^N$ arbitrarily; $\beta := \frac{\alpha}{1-\alpha} \cdot max_i\big\{\sum_j \{max_{(a,b)}\, p_{ij}(a,b) - min_{(a,b)}\, p_{ij}(a,b)\}\big\}$.

2. for all $i \in S$ do
     begin
       compute the matrix $M_x[i]$ with entries $r_i(a,b) + \alpha \sum_j p_{ij}(a,b)x_j,\ a \in A(i),\ b \in B(i)$;
       determine an optimal stationary policy $\pi^*$ for player 1 in the matrix game $M_x[i]$;
       determine an optimal stationary policy $\rho^*$ for player 2 in the matrix game $M_x[i]$;
       $y_i := val(M_x[i])$
     end

3. $z := v^\alpha\big((\pi^*)^\infty, (\rho^*)^\infty\big)$.

4. if $\|z - x\|_\infty > \frac{1-\alpha}{(1+\alpha)\beta}\,\varepsilon$ then begin $x := z$; go to step 2 end
   else go to step 5.

5. $z$ is an $\varepsilon$-approximation of the value vector $v^\alpha$ and $(\pi^*)^\infty$ and $(\rho^*)^\infty$ are $\big\{1 + \frac{2}{\beta(1+\alpha)}\big\}\,\varepsilon$-optimal policies for player 1 and 2, respectively (STOP).
```

The next example shows that Algorithm 10.5 does not converge in general.

Example 10.3

$S = \{1,2\};\ A(1) = B(1) = \{1,2\},\ A(2) = B(2) = \{1\};\ \alpha = \frac{3}{4}$.
$r_1(1,1) = 3,\ r_1(1,2) = 6,\ r_1(2,1) = 2,\ r_1(2,2) = 1,\ r_2(1,1) = 0$.
$p_{11}(1,1) = 1,\ p_{12}(1,1) = 0,\ p_{11}(1,2) = \frac{1}{3};\ p_{12}(1,2) = \frac{2}{3},\ p_{11}(2,1) = 1,\ p_{12}(2,1) = 0;$
$p_{11}(2,2) = 1,\ p_{12}(2,2) = 0;\ p_{21}(1,1) = 0,\ p_{22}(1,1) = 1$.
Take $\varepsilon = 0.2$ (then $\beta = 4$ and $\frac{1-\alpha}{(1+\alpha)\beta}\,\varepsilon = \frac{1}{140}$) and select $x = (0,0)$.

Iteration 1:

$i = 1:\ M_x[1] = \begin{pmatrix} 3 & 6 \\ 2 & 1 \end{pmatrix};\ \pi^*_{11} = 1,\ \pi^*_{11} = 0;\ \rho^*_{11} = 1,\ \rho^*_{11} = 0;\ y_1 = val\, M[1] = 3.$

$i = 2:\ M_x[2] = (0);\ \pi^*_{21} = 1;\ \rho^*_{21} = 1;\ y_2 = val\, M[2] = 0.$

$z = v^\alpha\big((\pi^*)^\infty, (\rho^*)^\infty\big) = (12, 0);\ \|z - x\|_\infty = 12 > \frac{1}{140};\ x = (12, 0).$

Iteration 2:

$i = 1:\ M_x[1] = \begin{pmatrix} 12 & 9 \\ 11 & 10 \end{pmatrix};\ \pi^*_{11} = 0,\ \pi^*_{11} = 1;\ \rho^*_{11} = 0,\ \rho^*_{11} = 1;\ y_1 = val\, M[1] = 10.$

$i = 2:\ M_x[2] = (0);\ \pi^*_{21} = 1;\ \rho^*_{21} = 1;\ y_2 = val\, M[2] = 0.$

$z = v^\alpha\big((\pi^*)^\infty, (\rho^*)^\infty\big) = (4, 0);\ \|z - x\|_\infty = 8 > \frac{1}{140};\ x = (4, 0).$

Iteration 3:

$i = 1:\ M_x[1] = \begin{pmatrix} 6 & 7 \\ 5 & 4 \end{pmatrix};\ \pi^*_{11} = 1,\ \pi^*_{11} = 0;\ \rho^*_{11} = 1,\ \rho^*_{11} = 0;\ y_1 = val\, M[1] = 6.$

$i = 2:\ M_x[2] = (0);\ \pi^*_{21} = 1;\ \rho^*_{21} = 1;\ y_2 = val\, M[2] = 0.$

$z = v^\alpha\big((\pi^*)^\infty, (\rho^*)^\infty\big) = (12, 0)$; $\|z - x\|_\infty = 8 > \frac{1}{140}$; $x = (12, 0)$.
Hence, we are in the same situation as at the start of iteration 2 and there is no convergence.

Since the mapping T is a contraction, it is a continuous mapping. In case $\frac{\partial (Tx)_i}{\partial x_j}$, the partial derivative in x, exists, then $\frac{\partial (Tx)_i}{\partial x_j} = \alpha p_{ij}(\pi, \rho)$, because $(Tx)_i = r_i(\pi, \rho) + \alpha \sum_j p_{ij}(\pi.\rho)x_j$, where π and ρ are optimal mixed strategies in the matrix game $M_x[i]$. Let $F : \mathbb{R}^N \to \mathbb{R}^N$ be defined by $Fx := Tx - x$. Then, the problem of finding the value vector of the stochastic game is the same as solving the nonlinear equation $Fx = 0$. We will show that Algorithm 10.5 is equivalent to Newton's method for solving $Fx = 0$. From Algorithm 10.5 we obtain

$$\begin{aligned}
x^{n+1} &= v^\alpha\big((\pi^n)^\infty, (\rho^n)^\infty\big) = x^n + v^\alpha\big((\pi^n)^\infty, (\rho^n)^\infty\big) - x^n \\
&= x^n + \big\{I - \alpha P(\pi^n, \rho^n)\big\}^{-1} r(\pi^n, \rho^n) - x^n \\
&= x^n + \big\{I - \alpha P(\pi^n, \rho^n)\big\}^{-1} r(\pi^n, \rho^n) - \big\{I - \alpha P(\pi^n, \rho^n)\big\}^{-1}\big\{I - \alpha P(\pi^n, \rho^n)\big\}x^n \\
&= x^n - \big\{\alpha P(\pi^n, \rho^n) - I\big\}^{-1}\{r(\pi^n, \rho^n) + \alpha P(\pi^n, \rho^n)x^n - x^n\}.
\end{aligned}$$

Because $\left\{\frac{\partial (Fx)_i}{\partial x_j}\right\}_{x=x^n} = \alpha p_{ij}(\pi^n, \rho^n) - \delta_{ij}$ and $r(\pi^n, \rho^n) + \alpha P(\pi^n, \rho^n)x^n - x^n = Tx^n - x^n = Fx^n$, we have

$$x^{n+1} = x^n - \{\nabla F x^n\}^{-1} F x^n. \tag{10.28}$$

i.e. Algorithm 10.5 is Newton's method for solving $Fx = 0$.

Let $\Delta x_n := x^{n+1} - x^n$, $\Delta T_n := Tx^{n+1} - Tx^n$ and $\Delta F_n := Fx^{n+1} - Fx^n$. Then,
$\Delta F_n = (Tx^{n+1} - x^{n+1}) - (Tx^n - x^n) = \Delta T_n - \Delta x_n$; $Fx^{n+1} = Fx^n + \Delta F_n = Fx^n + \Delta T_n - \Delta x_n$.
Similarly as in the proof of Lemma 10.3 it can be shown that

$$\alpha \cdot \textstyle\sum_j \{min_{(a,b)} p_{ij}(a,b)\}(\Delta x^n)_j \le (\Delta T_n)_i \le \alpha \cdot \sum_j \{max_{(a,b)} p_{ij}(a,b)\}(\Delta x^n)_j.$$

Then, $(\Delta T_n)_i$ is a convex combination of the upper and lower bound, i.e. $(\Delta T_n)_i = \sum_j q_{ij}(n)(\Delta x_n)_j$, where $q_{ij}(n) = \alpha\{\lambda \cdot max_{(a,b)}\, p_{ij}(a,b) + (1-\lambda) \cdot min_{(a,b)}\, p_{ij}(a,b)\}$ for some $\lambda \in [0,1]$. Hence,

$$Fx^{n+1} = Fx^n - \{I - Q(n)\}\Delta x_n. \tag{10.29}$$

From (10.29) and (10.28) it follows that

$$\begin{aligned}
Fx^{n+1} &= Fx^n + \{I - Q(n)\}\{\nabla F x^n\}^{-1} F x^n \\
&= Fx^n - \{I - Q(n)\}\{I - \alpha P(\pi^n, \rho^n)\}^{-1} F x^n \\
&= \big\{I - \{I - Q(n)\}\{I - \alpha P(\pi^n, \rho^n)\}^{-1}\big\} F x^n \\
&= \big\{I - \{I - \alpha P(\pi^n, \rho^n)\}^{-1} + Q(n)\{I - \alpha P(\pi^n, \rho^n)\}^{-1}\big\} F x^n \\
&= \big\{ - \alpha P(\pi^n, \rho^n)\{I - \alpha P(\pi^n, \rho^n)\}^{-1} + Q(n)\{I - \alpha P(\pi^n, \rho^n)\}^{-1}\big\} F x^n \\
&= \{Q(n) - \alpha P(\pi^n, \rho^n)\}\{I - \alpha P(\pi^n, \rho^n)\}^{-1} F x^n.
\end{aligned}$$

Therefore,

$$\begin{array}{rcl}(Fx^{n+1})_i & = & \sum_j \big\{\{Q(n) - \alpha P(\pi^n,\rho^n)\}\{I - \alpha P(\pi^n,\rho^n)\}^{-1}\big\}_{ij}(Fx^n)_j \\ & = & \sum_j \big\{\sum_k \{Q(n) - \alpha P(\pi^n,\rho^n)\}_{ik}\{\{I - \alpha P(\pi^n,\rho^n)\}^{-1}\}_{kj}\big\}(Fx^n)_j \\ & = & \sum_k \{Q(n) - \alpha P(\pi^n,\rho^n)\}_{ik} \cdot \sum_j \big\{\{I - \alpha P(\pi^n,\rho^n)\}^{-1}\}_{kj}(Fx^n)_j \\ & = & \sum_k \{Q(n) - \alpha P(\pi^n,\rho^n)\}_{ik} \cdot \frac{1}{1-\alpha} \cdot \|Fx^n\|.\end{array}$$

Notice that

$$\begin{array}{rcl}|\sum_k \{Q(n) - \alpha P(\pi^n,\rho^n)\}_{ik}| & = & \alpha \sum_k |\lambda \cdot max_{(a,b)}\, p_{ik}(a,b) + (1-\lambda) \cdot min_{(a,b)}\, p_{ik}(a,b) - p_{ik}(\pi^n,\rho^n)| \\ & \leq & \alpha \cdot max_i \big\{\sum_k \{max_{(a,b)}\, p_{ik}(a,b) - min_{(a,b)}\, p_{ik}(a,b)\}\big\} = (1-\alpha)\beta.\end{array}$$

Hence, $\|Fx^{n+1}\|_\infty \leq \beta \cdot \|Fx^n\|_\infty$, i.e. the process converges if $\beta < 1$.

Remark

The condition $\beta < 1$ is very restrictive. However, for problems that do not satisfy $\beta < 1$ the algorithm terminates in most cases.

Theorem 10.13

Assume that $\beta < 1$. Then, Algorithm 10.5 is correct.

Proof

For $\beta < 1$, we have shown that $Fx^n \to 0$ for $n \to \infty$, implying that $\|x^{n+1} - x^n\| \to 0$ for $n \to \infty$, i.e. the algorithm terminates. At termination with $z = x^{n+1}$ and $x = x^n$, we have

$$\begin{array}{rcl}\|x^{n+1} - v^\alpha\|_\infty & = & \|Tx^{n+1} - Fx^{n+1} - Tv^\alpha\|_\infty \leq \|Tx^{n+1} - Tv^\alpha\|_\infty + \|Fx^{n+1}\|_\infty \\ & \leq & \alpha \cdot \|x^{n+1} - v^\alpha\|_\infty + \beta \cdot \|Fx^n\|_\infty.\end{array}$$

Hence,

$$\begin{array}{rcl}\|x^{n+1} - v^\alpha\|_\infty & \leq & \frac{\beta}{1-\alpha} \cdot \|Fx^n|_\infty = \frac{\beta}{1-\alpha} \cdot \|\nabla Fx^n(x^{n+1} - x^n)\|_\infty \\ & \leq & \frac{\beta}{1-\alpha} \cdot \|I - \alpha P(\pi^n,\rho^n)\|_\infty \cdot \|x^{n+1} - x^n)\|_\infty \\ & \leq & \frac{1+\alpha}{1-\alpha} \cdot \beta \cdot \|x^{n+1} - x^n)\|_\infty \leq \varepsilon.\end{array}$$

Let $\gamma := \frac{\varepsilon}{\beta} \cdot \frac{1-\alpha}{1+\alpha}$, then $-\gamma \cdot e \leq x^{n+1} - x^n \leq \gamma \cdot e$. We can also write for any $\rho^\infty \in \Gamma$,

$$\begin{array}{rcl}L_{\pi^n,\rho}x^n & = & r(\pi^n,\rho) + \alpha P(\pi^n,\rho)x^n \geq r(\pi^n,\rho^n) + \alpha P(\pi^n,\rho^n)x^n \\ & = & r(\pi^n,\rho^n) + \alpha P(\pi^n,\rho^n)x^{n+1} + \alpha P(\pi^n,\rho^n)(x^n - x^{n+1}) \\ & = & x^{n+1} + \alpha P(\pi^n,\rho^n)(x^n - x^{n+1}) \\ & \geq & x^{n+1} - \alpha\gamma P(\pi^n,\rho^n)e = x^{n+1} - \alpha\gamma \cdot e \geq x^n - (1+\alpha)\gamma \cdot e = x^n - \delta \cdot e,\end{array}$$

with $\delta := (1+\alpha)\gamma$. The monotonicity of $L_{\pi^n,\rho}$ yields $L^k_{\pi^n,\rho}x^n \geq x^n - \delta(1+\alpha+\cdots+\alpha^k) \cdot e,\ k \in \mathbb{N}$, implying

$$\begin{array}{rcl}v^\alpha\big((\pi^n)^\infty,\rho^\infty\big) & \geq & x^n - \delta(1-\alpha)^{-1} \cdot e = x^{n+1} + (x^n - x^{n+1}) - (1-\alpha)^{-1}\delta \cdot e \\ & \geq & v^\alpha - \varepsilon \cdot e - \frac{1+\alpha}{1-\alpha}\gamma \cdot e = v^\alpha - \{1 + \frac{2}{\beta(1+\alpha)}\varepsilon \cdot e.\end{array}$$

From this result it follows that $(\pi^n)^\infty$ is a $\{1 + \frac{2}{\beta(1+\alpha)}\}\varepsilon$-optimal policy for player 1. Similarly it can be shown that $(\rho^n)^\infty$ is a $\{1 + \frac{2}{\beta(1+\alpha)}\}\varepsilon$-optimal policy for player 2. □

The last method in this section uses an integer k, where $1 \leq k \leq \infty$. For $k = 1$ we obtain Algorithm 10.3 and for $k = \infty$ Algorithm 10.5. So, this algorithm is of the type of modified policy iteration as analyzed in Section 3.8 for the MDP model.

Algorithm 10.6 *Modified policy iteration for discounted games*
Input: Instance of a two-person stochastic game, some $\varepsilon > 0$ and some integer $1 \leq k \leq \infty$.
Output: An ε-approximation of the value vector v^α and a pair $\big((\pi^*)^\infty, (\rho^*)^\infty\big)$ of stationary $\frac{1}{\alpha}\varepsilon$-optimal

policies.

```
1. Select $x \in \mathbb{R}^N$ such that $Tx \leq x$.
2. for all $i \in S$ do
     begin
       compute the matrix $M_x[i]$ with entries $r_i(a,b) + \alpha \sum_j p_{ij}(a,b)x_j,\ a \in A(i),\ b \in B(i)$;
       determine an optimal stationary policy $\pi^*$ for player 1 in the matrix game $M_x[i]$;
       determine an optimal stationary policy $\rho^*$ for player 2 in the matrix game $M_x[i]$;
       $y_i := val(M_x[i])$
     end
3. $z := U^k(\rho^*)x$, where $U(\rho^*)x$ is defined by $\{U(\rho^*)x\}_i := max_a \{r_i(a,\rho^*) + \alpha \sum_j p_{ij}(a,\rho^*)x_j\}$.
4. if $\|z - x\|_\infty > \frac{1-\alpha}{\alpha)}\,\varepsilon$ then begin $x := z$; go to step 2 end
   else go to step 5.
5. $z$ is an $\varepsilon$-approximation of the value vector $v^\alpha$ and $(\pi^*)^\infty$ and $(\rho^*)^\infty$ are $\frac{1}{\alpha}\varepsilon$-optimal policies
   for player 1 and 2, respectively (STOP).
```

We denote the vectors x, y, z, the strategies π^* and ρ^* and the operator $U(\rho^*)$ in the n-th iteration by $x^n, y^n, z^n, \pi^n, \rho^n$ and U_n, respectively. For any fixed $\rho^\infty \in \Gamma$ and $x \in \mathbb{R}^N$, we have the property

$$U(\rho)x = max_\pi\,\{r(\pi,\rho) + \alpha P(\pi,\rho)x\} \geq max_\pi min_\rho\,\{r(\pi,\rho) + \alpha P(\pi,\rho)x\} = Tx, \qquad (10.30)$$

implying that $U^m(\rho)x \geq T^m x$ for all $\rho^\infty \in \Gamma,\ x \in \mathbb{R}^N$ and $m \in \mathbb{N}$. Furthermore, notice that $y^n = Tx^n = U_n x^n$ for all n.

Lemma 10.13
$x^n \geq Tx^n \geq x^{n+1} \geq v^\alpha$ *for* $n = 0, 1, \dots$.

Proof
We apply induction on n.
For $n = 0$, we have $x^0 \geq Tx^0 = y^0 = U_0x^0$ (the first inequality by step 1 of the algorithm).

Since U_0 is monotone and $x^0 \geq U_0x^0$, we obtain $x^1 = U_0^kx^0 \leq U_0x^0 = Tx^0 \leq x^0$.
From $x^1 \leq Tx^0 \leq x^0$ and the monotonicity of T it follows that $T^mx^1 \leq x^0$ for $m = 0, 1, 2, \dots$. Hence, $v^\alpha = \lim_{m\to\infty} T^mx^1 \leq x^0$. Therefore, $x^1 = U_0^kx^0 \geq T^kx^0 \geq Tv^\alpha = v^\alpha$, and we have shown that $x^n \geq Tx^n \geq x^{n+1} \geq v^\alpha$ for $n = 0$.
Suppose that $x^n \geq Tx^n \geq x^{n+1} \geq v^\alpha$. Now, we will show that $x^{n+1} \geq Tx^{n+1} \geq x^{n+2} \geq v^\alpha$. We have, $U_nx^{n+1} = Tx^{n+1} = T\{U_n^kx^n\} \leq U_n^{k+1}x^n \leq U_n^kx^n = x^{n+1}$, the last inequality since $U_nx^n = Tx^n \leq x^n$ and the monotonicity of U_n. From $U_nx^{n+1} \leq x^{n+1}$ follows

$$x^{n+2} = U_{n+1}^kx^{n+1} \leq U_{n+1}^{k-1}x^{n+1} \leq \cdots \leq U_{n+1}x^{n+1} = Tx^{n+1}.$$

Since $x^{n+1} \geq v^\alpha$, we obtain $x^{n+2} = U_{n+1}^kx^{n+1} \geq T^kx^{n+1} \geq T^kv^\alpha = v^\alpha$. □

Corollary 10.3
$\lim_{n\to\infty} x^n = v^\alpha$.

Proof
From Lemma 10.13 it follows that $v^\alpha \leq x^n \leq Tx^{n-1} \leq T^2x^{n-2} \leq \cdots \leq T^{n-1}x^1 \leq T^nx^0$ for $n = 0, 1, 2, \dots$. Since $\lim_{n\to\infty} T^nx^0 = v^\alpha$, we also have $\lim_{n\to\infty} x^n = v^\alpha$. □

Theorem 10.14
Algorithm 10.6 is correct.

Proof
Because $\lim_{n\to\infty} x^n = v^\alpha$, the algorithm terminates. Let x^n and z^n be the vectors x and z in the final iteration. Since $0 \leq x^{n+1} - v^\alpha \leq Tx^n - v^\alpha$, we obtain

$$\begin{array}{rcl} \|x^{n+1} - v^\alpha\|_\infty & \leq & \|Tx^n - v^\alpha\|_\infty = \|Tx^n - Tv^\alpha\|_\infty \leq \alpha \cdot \|x^n - v^\alpha\|_\infty \\ & \leq & \alpha \cdot \|x^n - x^{n+1}\|_\infty + \alpha \cdot \|x^{n+1} - v^\alpha\|_\infty. \end{array}.$$

Hence, $\|z^n - v^\alpha\|_\infty = \|x^{n+1} - v^\alpha\|_\infty \leq \frac{\alpha}{1-\alpha} \cdot \|x^n - x^{n+1}\|_\infty = \frac{\alpha}{1-\alpha} \cdot \|z^n - x^n\|_\infty < \varepsilon$, i.e. z^n is an ε-approximation of the value vector. Furthermore, we have for any $\rho^\infty \in \Gamma$,

$$L_{\pi^n,\rho}x^n = r(\pi^n, \rho) + \alpha P(\pi^n, \rho)x^n \geq Tx^n \geq x^{n+1} \geq x^n - \|x^n - x^{n+1}\|_\infty \cdot e \geq x^n - \tfrac{1-\alpha}{\alpha}\varepsilon \cdot e.$$

Hence, $L_{\pi^n,\rho}^mx^n \geq x^n - \{1 + \alpha + \cdots + \alpha^{m-1}\}\frac{1-\alpha}{\alpha}\varepsilon \cdot e$ for $m = 1, 2, \dots$. Therefore, we obtain $v^\alpha\big((\pi^n)^\infty, \rho^\infty\big) = \lim_{m\to\infty} L_{\pi^n,\rho}^mx^n \geq x^n - \frac{1}{\alpha}\varepsilon \cdot e \geq v^\alpha - \frac{1}{\alpha}\varepsilon \cdot e$. From this result it follows that $(\pi^n)^\infty$ is a $\frac{1}{\alpha}\varepsilon$-optimal policy for player 1. Similarly it can be shown that $(\rho^n)^\infty$ is a $\frac{1}{\alpha}\varepsilon$-optimal policy for player 2. □

10.2.4 Finite methods

In general, solutions to stochastic games lack an important algebraic property, which suggests that effectively solving is essentially more difficult than solving matrix games. This is illustrated by the following example.

Example 10.4
$S=\{1,2\};\ A(1)=B(1)=\{1,2\},\ A(2)=B(2)=\{1\};\ \alpha=\frac{1}{2}$.
$r_1(1,1)=1,\ r_1(1,2)=0,\ r_1(2,1)=0,\ r_1(2,2)=3,\ r_2(1,1)=0$.
$p_{11}(1,1)=1,\ p_{12}(1,1)=0;\ p_{11}(1,2)=0,\ p_{12}(1,2)=1;\ p_{11}(2,1)=0,\ p_{12}(2,1)=1;$
$p_{11}(2,2)=1,\ p_{12}(2,2)=0;\ p_{21}(1,1)=0,\ p_{22}(1,1)=1$.
Consider the fixed point equation $x=Tx$, i.e.

$$x_1 = val\begin{pmatrix} 1+\frac{1}{2}x_1 & 0+\frac{1}{2}x_2 \\ 0+\frac{1}{2}x_2 & 3+\frac{1}{4}x_1 \end{pmatrix};\ x_2 = val\big(0+\tfrac{1}{2}x_2\big) \ \rightarrow\ v_2^\alpha = x_2 = 0.$$

$$x_1 = val\begin{pmatrix} 1+\frac{1}{2}x_1 & 0 \\ 0 & 3+\frac{1}{2}x_1 \end{pmatrix} \ \rightarrow\ x_1 = \frac{(1+\frac{1}{2}x_1)(3+\frac{1}{2}x_1)}{(1+\frac{1}{2}x_1)+(3+\frac{1}{2}x_1)} \ \rightarrow\ v_1^\alpha = x_1 = \tfrac{2}{3}\{-2+\sqrt{13}\}.$$

The optimal policies are for both players $\left(\frac{7+\sqrt{13}}{8+2\sqrt{13}}, \frac{1+\sqrt{13}}{8+2\sqrt{13}}\right)$.

The above example shows that while all the data defining the stochastic game (the rewards, the transition probabilities and the discount factor) are rational, the value vector has irrational entries. Thus the data and the solution are not in the same ordered Archimedean field. This phenomenon is called lack of the *ordered field property*. It essentially eliminates the possibility of solving discounted stochastic games by performing only finitely many arithmetic operations. Note that since linear programs solve a general matrix game, and since an optimal basis of that program can be found via finitely many pivots of the simplex method, matrix games possess the ordered field property.

One line of research that has evolved from the preceding considerations is focussed on identifying those natural classes of stochastic games for which the ordered field property holds, and on developing algorithms for their solution. We will consider the following special games:
(1) The single-controller stochastic game.
(2) The switching-controller stochastic game.
(3) The separable reward - state independent transitions (SER-SIT) stochastic game.
(4) The additive reward - additive transitions (ARAT) stochastic game.

Single-controller stochastic game

In the single-controller stochastic game is player 1 the 'single-controller'. This means that the transition probabilities $p_{ij}(a,b)$ are independent of b. Therefore, we denote these probabilities as $p_{ij}(a)$. Under this assumption the nonlinear program (10.23) becomes the following linear program

$$min\left\{\sum_i v_i \ \middle|\ \begin{array}{rcl} \sum_j\{\delta_{ij}-\alpha p_{ij}(a)\}v_j & - & \sum_b r_i(a,b)\rho_{ib} \ \geq 0,\ a\in A(i),\ i\in S \\ & & \sum_b \rho_{ib} \ = 1,\ i\in S \\ & & \rho_{ib} \ \geq 0,\ b\in B(i),\ i\in S \end{array}\right\}. \qquad (10.31)$$

The dual program is

$$max \left\{ \sum_i z_i \;\middle|\; \begin{array}{rcrcl} \sum_{(i,a)}\{\delta_{ij} - \alpha p_{ij}(a)\}x_i(a) & & & = & 1,\ j \in S \\ -\sum_a r_i(a,b)x_i(a) & + & z_i & \leq & 0,\ (i,b) \in S \times B \\ & & x_i(a) & \geq & 0,\ (i,a) \in S \times A \end{array} \right\}. \tag{10.32}$$

The following theorem shows that the value vector and optimal stationary policies for both players can be obtained from the optimal solutions of the dual pair of linear programs.

Theorem 10.15

Let (v^*, ρ^*) and (x^*, z^*) be optimal solutions of the linear programs (10.31) and (10.32), respectively. Define the stationary policy $(\pi^*)^\infty$ by $\pi^*_{ia} := \frac{x^*_i(a)}{\sum_a x^*_i(a)}$, $(i,a) \in S \times A$. Then, v^* is the value vector and $(\pi^*)^\infty$ and $(\rho^*)^\infty$ are optimal stationary policies for player 1 and 2, respectively.

Proof

Theorem 10.9 implies that v^* is the value vector of the stochastic game. Since

$$\textstyle\sum_a x^*_i(a) = 1 + \alpha \sum_{(i,a)} p_{ij}(a)x_i(a) > 0,\ j \in S,$$

the stationary policy $(\pi^*)^\infty$ is well defined. From the constraints of program (10.31) it follows that $\{I - \alpha P(\pi)\}v^* \geq r(\pi, \rho^*)$ for every $\pi^\infty \in \Pi$. Therefore,

$$v^* \geq \{I - \alpha P(\pi)\}^{-1} r(\pi, \rho^*) = v^\alpha\big(\pi^\infty, (\rho^*)^\infty\big) \text{ for every } \pi^\infty \in \Pi. \tag{10.33}$$

From the complementary slackness property of linear programming it follows that

$$x^*_i(a) \cdot \Big\{ \textstyle\sum_j \{\delta_{ij} - \alpha p_{ij}(a)\}v^*_j - \sum_b r_i(a,b)\rho^*_{ib} \Big\} = 0 \text{ for all } (i,a) \in S \times A.$$

Since $x^*_i(a) > 0$ if and only if $\pi^*_{ia} > 0$, we also have

$$\pi^*_{ia} \cdot \Big\{ \textstyle\sum_j \{\delta_{ij} - \alpha p_{ij}(a)\}v^*_j - \sum_b r_i(a,b)\rho^*_{ib} \Big\} = 0 \text{ for all } (i,a) \in S \times A.$$

Therefore, $\sum_a \pi^*_{ia} \cdot \Big\{ \sum_j \{\delta_{ij} - \alpha p_{ij}(a)\}v^*_j - \sum_b r_i(a,b)\rho^*_{ib} \Big\} = 0$ for all $i \in S$, implying

$$\textstyle\sum_j \{\delta_{ij} - \alpha p_{ij}(\pi^*)\}v^*_j = r_i(\pi^*, \rho^*) \text{ for all } i \in S, \text{ i.e. } \{I - \alpha P(\pi^*)\}v^* = r(\pi^*, \rho^*).$$

So, $v^* = \{I - \alpha P(\pi^*)\}^{-1} r(\pi^*, \rho^*) = v^\alpha\big((\pi^*)^\infty, (\rho^*)^\infty\big)$. Since the optimum values of (10.31) and (10.32) are equal, we can write

$$\sum_j v^\alpha_j\big((\pi^*)^\infty, (\rho^*)^\infty\big) = \sum_i z^*_i. \tag{10.34}$$

Since $z^*_i \leq \sum_a r_i(a,b)x^*_i(a)$ for all $b \in B(i)$, we also have $z^*_i \leq \sum_a r_i(a,\rho)x^*_i(a)$ for all $\rho^\infty \in \Gamma$. From the constraints of (10.32) it follows that, with $x^*_i := \sum_a x^*_i(a),\ i \in S$,

$$1 = \textstyle\sum_{(i,a)} \{\delta_{ij} - \alpha p_{ij}(a)\}\pi^*_{ia} \cdot x^*_i = \sum_i \{\delta_{ij} - \alpha p_{ij}(\pi^*)\} \cdot x^*_i,$$

or, in vector notation, $e^T = (x^*)^T\{I - \alpha P(\pi^*)\}$, implying $(x^*)^T = e^T\{I - \alpha P(\pi^*)\}^{-1}$. Then, because $z^*_i \leq \sum_a r_i(a,\rho)x^*_i(a)$,

$$\begin{array}{rcl} \sum_i z^*_i & \leq & \sum_{(i,a)} r_i(a,\rho)x^*_i(a) = \sum_{(i,a)} r_i(a,\rho)\pi^*_{ia}x^*_i \\ & = & \sum_i r_i(\pi^*,\rho)x^*_i = (x^*)^T r(\pi^*,\rho) = e^T\{I\ \alpha P(\pi^*)\}^{-1} r(\pi^*,\rho) \\ & = & e^T v^\alpha\big((\pi^*)^\infty, \rho^\infty\big) = \sum_j v^\alpha_j\big((\pi^*)^\infty, \rho^\infty\big). \end{array}$$

With (10.34) we obtain $\sum_j v_j^\alpha\big((\pi^*)^\infty,\rho^\infty\big) \geq \sum_j v_j^\alpha\big((\pi^*)^\infty,(\rho^*)^\infty\big)$ for all $\rho^\infty \in \Gamma$. Hence, $(\rho^*)^\infty$ is an optimal policy for player 2 in the MDP induced by policy $(\pi^*)^\infty$. Therefore,

$$v^\alpha\big((\pi^*)^\infty,\rho^\infty\big) \geq v^* = v^\alpha\big((\pi^*)^\infty,(\rho^*)^\infty\big) \text{ for all } \rho^\infty \in \Gamma. \tag{10.35}$$

Hence, by (10.33 and (10.35, we have

$$v^\alpha\big((\pi^*)^\infty,\rho^\infty\big) \geq v^* = v^\alpha\big((\pi^*)^\infty,(\rho^*)^\infty\big) \geq v^\alpha\big(\pi^\infty,(\rho^*)^\infty\big),\ \pi^\infty \in \Pi, \rho^\infty \in \Gamma,$$

i.e. $(\pi^*)^\infty$ and $(\rho^*)^\infty$ are optimal stationary policies for player 1 and 2, respectively. □

Algorithm 10.7 *Single-controller game with discounting*

Input: Instance of a two-person single-controller stochastic game

Output: The value vector v^α and a pair $\big((\pi^*)^\infty,(\rho^*)^\infty\big)$ of stationary optimal policies.

1. Compute optimal solutions (v^*,ρ^*) and (x^*,z^*) of the linear programs (10.31) and (10.32).
2. Define the stationary policy $(\pi^*)^\infty$ by $\pi^*_{ia} := \frac{x_i^*(a)}{\sum_a x_i^*(a)}$, $(i,a) \in S \times A$.
3. v^* is the value vector and $(\pi^*)^\infty$ and $(\rho^*)^\infty$ optimal stationary policies for player 1 and 2 (STOP).

Example 10.5

$S = \{1,2\};\ A(1) = \{1,2\},\ B(1) = \{1,2,3\},\ A(2) = \{1,2,3\},\ B(2) = \{1,2\};\ \alpha = \frac{1}{2}$.
$r_1(1,1) = 5,\ r_1(1,2) = 1,\ r_1(1,3) = 6,\ r_1(2,1) = 4,\ r_1(2,2) = 6,\ r_1(2,3) = 2;$
$r_2(1,1) = 6,\ r_2(1,2) = 0,\ r_2(2,1) = 3,\ r_2(2,2) = 4,\ r_2(3,1) = 0,\ r_2(3,2) = 6.$
$p_{11}(1) = 1,\ p_{12}(1) = 0;\ p_{11}(2) = 0,\ p_{12}(2) = 1;\ p_{21}(1) = 1,\ p_{22}(1) = 0;$
$p_{21}(2) = 0,\ p_{22}(2) = 1;\ p_{21}(3) = 1,\ p_{22}(3) = 0.$
The linear programs (10.31) and (10.32) are

$$min \left\{ v_1 + v_2 \;\middle|\; \begin{array}{rrrrrrrr} \frac{1}{2}v_1 & & -\ 5\rho_{11} & -\ \rho_{12} & -\ 6\rho_{13} & & & \geq 0 \\ v_1 & -\ \frac{1}{2}v_2 & -\ 4\rho_{11} & -\ 6\rho_{12} & -\ 2\rho_{13} & & & \geq 0 \\ -\ \frac{1}{2}v_1 & +\ v_2 & & & & -\ 6\rho_{21} & & \geq 0 \\ & \frac{1}{2}v_2 & & & & -\ 3\rho_{21} & -\ 4\rho_{22} & \geq 0 \\ -\ \frac{1}{2}v_1 & +\ v_2 & & & & & -\ 6\rho_{22} & \geq 0 \\ \multicolumn{8}{r}{\rho_{11}+\rho_{12}+\rho_{13} = 1;\ \rho_{21}+\rho_{22} = 1;\ \rho_{11},\rho_{12},\rho_{13},\rho_{21},\rho_{22} \geq 0} \end{array} \right\}$$

and

$$max \left\{ z_1 + z_2 \;\middle|\; \begin{array}{rrrrrrr} \frac{1}{2}x_{11} & +\ x_{12} & -\ \frac{1}{2}x_{21} & & -\ \frac{1}{2}x_{23} & & = 1 \\ & -\ \frac{1}{2}x_{12} & +\ x_{21} & +\ \frac{1}{2}x_{22} & +\ x_{23} & & = 1 \\ -\ 5x_{11} & -\ 4x_{12} & & & & +\ z_1 & \leq 0 \\ -\ x_{11} & -\ 6x_{12} & & & & +\ z_1 & \leq 0 \\ -\ 6x_{11} & -\ 2x_{12} & & & & +\ z_1 & \leq 0 \\ & & -\ 6x_{21} & -\ 3x_{22} & & +\ z_2 & \leq 0 \\ & & & -\ 4x_{22} & -\ 6x_{23} & +\ z_2 & \leq 0 \\ \multicolumn{7}{r}{x_{11},x_{12},x_{21},x_{22},x_{23} \geq 0} \end{array} \right\}$$

The optimal solutions are:
$v_1^* = 7.327,\ v_2^* = 6.916;\ \rho_{11}^* = 0,\ \rho_{12}^* = 0.467,\ \rho_{13}^* = 0.533,\ \rho_{21}^* = 0.542,\ \rho_{22}^* = 0.458$ and
$z_1^* = 5.720,\ z_2^* = 8.523;\ x_{11}^* = 0.673,\ x_{12}^* = 0.841,\ x_{21}^* = 0.355,\ x_{22}^* = 2.131,\ x_{23}^* = 0.$
The optimal policy for player 1 is: $\pi_{11}^* = 0.446,\ \pi_{12}^* = 0.554,\ \pi_{21}^* = 0.856,\ \pi_{22}^* = 0.144,\ \pi_{23}^* = 0.$

Switching-controller stochastic game

In a switching-controller stochastic game we assume that the set of states is the union of two disjoint sets S_1 and S_2 such that player 1 controls the transitions in S_1 and player 2 in S_2. Notice that a game with perfect information and the single-controller stochastic game are special cases of the switching-controller stochastic game.

Denote the transitions by $p_{ij}(a,b) = \begin{cases} p_{ij}(a),\ i \in S_1,\ a \in A(i),\ b \in B(i),\ j \in S; \\ p_{ij}(b),\ i \in S_2,\ a \in A(i),\ b \in B(i),\ j \in S. \end{cases}$

It appears that to solve such a game by a finite algorithm, a finite sequence of linear programs and matrix games needs to be solved instead of only a single one. The linear programs are the programs of the type of linear programs for single-controller stochastic games.

Suppose that player 2 fixes his strategy ρ^∞ in the states of S_2. Then, we denote the corresponding single-controller stochastic game by $SCSG(\rho)$ with data

$$r_i(a,b) = \begin{cases} \qquad r_i(a,b) & ,\ i \in S_1,\ a \in A(i),\ b \in B(i) \\ r_i(a,\rho) = \sum_b r_i(a,b)\rho_{ib} & ,\ i \in S_2,\ a \in A(i),\ b \in B(i) \end{cases}$$

and

$$p_{ij}(a,b) = \begin{cases} \qquad p_{ij}(a) & ,\ i \in S_1,\ a \in A(i),\ b \in B(i),\ j \in S \\ p_{ij}(\rho) = \sum_b p_{ij}(b)\rho_{ib} & ,\ i \in S_2,\ a \in A(i),\ b \in B(i),\ j \in S. \end{cases}$$

Notice that the transitions in the states of S_2 are independent of any choice of the players and the rewards depend only on the action taken by player 1. So, in the states of S_2 player 2 is a dummy and the first player will choose the action which maximizes $r_i(a,\rho)$ over the action set $A(i)$. Let $a[i,\rho]$ be that action, i.e. $a[i,\rho] := argmax_{a \in A(i)}\, r_i(a,\rho),\ i \in S_2$. The linear program for the single-controller stochastic game by $SCSG(\rho)$ is:

$$min \left\{ \sum_i v_i \ \middle|\ \begin{array}{lcrl} \sum_j \{\delta_{ij} - \alpha p_{ij}(a)\}v_j & - & \sum_b r_i(a,b)\rho_{ib} & \geq 0,\ a \in A(i),\ i \in S_1 \\ \sum_j \{\delta_{ij} - \alpha p_{ij}(\rho)\}v_j & - & r_i(a,\rho) & \geq 0,\ a \in A(i),\ i \in S_2 \\ & & \sum_b \rho_{ib} & = 1,\ i \in S_1 \\ & & \rho_{ib} & \geq 0,\ b \in B(i),\ i \in S_1 \end{array} \right\}. \tag{10.36}$$

The inequalities for $i \in S_2$ can be written as $v_i \geq \alpha \sum_j p_{ij}(\rho)v_j + r_i(a,\rho),\ a \in A(i),\ i \in S_2$, and are equivalent to a single inequality for each $i \in S_2$, namely $v_i \geq \alpha \sum_j p_{ij}(\rho)v_j + r_i(a[i,\rho])$. Therefore, program (10.36) is equivalent to the program

$$min \left\{ \sum_i v_i \ \middle|\ \begin{array}{lcrl} \sum_j \{\delta_{ij} - \alpha p_{ij}(a)\}v_j & - & \sum_b r_i(a,b)\rho_{ib} & \geq 0,\ a \in A(i),\ i \in S_1 \\ \sum_j \{\delta_{ij} - \alpha p_{ij}(\rho)\}v_j & - & r_i(a[i,\rho]) & \geq 0,\ i \in S_2 \\ & & \sum_b \rho_{ib} & = 1,\ i \in S_1 \\ & & \rho_{ib} & \geq 0,\ b \in B(i),\ i \in S_1 \end{array} \right\}. \tag{10.37}$$

Note

For different choices of ρ in S_2, the linear programs (10.37) only differ in the inequalities for the states S_2. This property can be used, e.g. by using the dual simplex method for the solution of subsequent programs (10.37).

Example 10.6

$S = \{1,2\};\ S_1 = \{1\},\ S_2 = \{2\};\ A(1) = B(1) = A(2) = B(2) = \{1,2\};\ \alpha = \frac{1}{2}.$
$r_1(1,1) = 3,\ r_1(1,2) = 1,\ r_1(2,1) = 1,\ r_1(2,2) = 4;$
$r_2(1,1) = 4,\ r_2(1,2) = 6,\ r_2(2,1) = 7,\ r_2(2,2) = 5.$
$p_{11}(1) = 0,\ p_{12}(1) = 1;\ p_{11}(2) = 1,\ p_{12}(2) = 0;\ p_{21}(1) = 1,\ p_{22}(1) = 0;\ p_{21}(2) = 0,\ p_{22}(2) = 1.$
Let player 2 choose in state 2 both action 1 and 2 with probability $\frac{1}{2}$.
The rewards and probabilities in state 2 are $r_2(1,\rho) = \frac{1}{2} \cdot 4 + \frac{1}{2} \cdot 6 = 5;\ r_2(2,\rho) = \frac{1}{2} \cdot 7 + \frac{1}{2} \cdot 5 = 6.$
$p_{21}(\rho) = \frac{1}{2} \cdot 1 + \frac{1}{2} \cdot 0 = \frac{1}{2};\ p_{22}(\rho) = \frac{1}{2} \cdot 0 + \frac{1}{2} \cdot 1 = \frac{1}{2}.$
Program (10.36) becomes

$$min \left\{ v_1 + v_2 \;\middle|\; \begin{array}{rcrcrcrl} v_1 & - & \frac{1}{2}v_2 & - & 3\rho_{11} & - & \rho_{12} & \geq 0 \\ \frac{1}{2}v_1 & & & - & \rho_{11} & - & 4\rho_{12} & \geq 0 \\ -\frac{1}{4}v_1 & + & \frac{3}{4}v_2 & & & & & \geq 5 \\ -\frac{1}{4}v_1 & + & \frac{3}{4}v_2 & & & & & \geq 6 \\ & & & & \rho_{11} & + & \rho_{12} & = 1 \\ & & & & & & \rho_{11},\ \rho_{12} & \geq 0 \end{array} \right\},$$

which is equivalent to

$$min \left\{ v_1 + v_2 \;\middle|\; \begin{array}{rcrcrcrl} v_1 & - & \frac{1}{2}v_2 & - & 3\rho_{11} & - & \rho_{12} & \geq 0 \\ \frac{1}{2}v_1 & & & - & \rho_{11} & - & 4\rho_{12} & \geq 0 \\ -\frac{1}{4}v_1 & + & \frac{3}{4}v_2 & & & & & \geq 6 \\ & & & & \rho_{11} & + & \rho_{12} & = 1 \\ & & & & & & \rho_{11},\ \rho_{12} & \geq 0 \end{array} \right\}.$$

The solution of this program is: $v_1 = 6.57,\ v_2 = 10.19,\ \rho_{11} = 0.24,\ \rho_{12} = 0.76.$

Denote the value vector of the single controller stochastic game $SCSG(\rho)$ by v^ρ. This value vector satisfies the fixed point equation $x = T^\rho x$, i.e. $x_i = val\big(M_x^\rho[i]\big),\ i \in S$, where $M_x^\rho[i]$ has the elements $\begin{cases} r_i(a,b) + \alpha \sum_j p_{ij}(a)x_j & ,\ i \in S_1; \\ r_i(a,\rho) + \alpha \sum_j p_{ij}(\rho)x_j & ,\ i \in S_2. \end{cases}$

If it turns out that $x = Tx$, with $(Tx)_i := val\big(M_x[i]\big),\ i \in S$, where $M_x[i]$ has the elements $\begin{cases} r_i(a,b) + \alpha \sum_j p_{ij}(a)x_j & ,\ i \in S_1 \\ r_i(a,b) + \alpha \sum_j p_{ij}(b)x_j & ,\ i \in S_2 \end{cases}$, then x is the value vector of the original game.

Therefore, we compute $val\big(M_{v^\rho}[i]\big),\ i \in S_2$, and check whether $v_i^\rho = val\big(M_{v^\rho}[i]\big),\ i \in S_2$. If this is the case, we have found the value vector and the corresponding optimal stationary policies of the two players; if not, our 'guess' for $\rho_{ib},\ i \in S_2,\ b \in B(i)$, was not optimal and we try another ρ for the states in S_2, namely the ρ's we found in the matrix games $M_{v^\rho}[i],\ i \in S_2$.

For a matrix game it is well known that the optimal strategy spaces are polytopes. We need for our algorithm extreme optimal strategies. If we use linear programming to compute the value and optimal strategies of the game we find extreme optimal strategies. The algorithm for the switching-controller game with discounting is as follows.

Algorithm 10.8 *Switching-controller game with discounting*
Input: Instance of a two-person switching-controller stochastic game
Output: The value vector v^α and a pair $\big((\pi^*)^\infty, (\rho^*)^\infty\big)$ of stationary optimal policies.

```
1. $n := 0$; select an arbitrary $x^0 \in \mathbb{R}^N$;
   for all $i \in S_2$ do
     determine an optimal extreme stationary policy $\rho^0$ for player 2 in the matrix game $M_{x^0}[i]$
     with entries $r_i(a,b) + \alpha \sum_j p_{ij}(b)x_j,\ a \in A(i),\ b \in B(i)$.
2. $n := n + 1$; solve the single-controller stochastic game $SCSG(\rho^{n-1})$, i.e. solve the linear
   program (10.37) and denote the value vector by $x^n$.
3. for all $i \in S_2$ do
     begin
     compute the matrix $M_{x^n}[i]$ with entries $r_i(a,b) + \alpha \sum_j p_{ij}(b)x_j,\ a \in A(i),\ b \in B(i)$.
     determine an optimal extreme policy $\rho^n$ for player 2 in the matrix game $M_{x^n}[i]$;
     $y_i^n := val(M_{x^n}[i])$
     end
4. if $y_i^n = x_i^n$ for all $i \in S_2$ then go to step 5
   else return to step 2
5. $v^\alpha := x^n$ is the value vector and $\rho^*_{ib} := \rho^n_{ib},\ i \in S_2,\ b \in B(i)$ is part of an optimal policy
   for player 2; the optimal actions for player 1 in the states $i \in S_2$ follow from an optimal
   extreme policy $\pi^n$ for player 1 in the matrix game $M_{x^n}[i]$; the optimal actions for player 2
   and 1 in the states $i \in S_1$ follow from the linear program (10.37) and its dual, respectively
   (STOP).
```

Example 10.6 (continued)
Start:
Note that $M_x[2] = \begin{pmatrix} 4 + \frac{1}{2}x_1 & 6 + \frac{1}{2}x_2 \\ 7 + \frac{1}{2}x_1 & 5 + \frac{1}{2}x_2 \end{pmatrix}$.

$n = 0$; choose $x^0 = (12, 0)$; $M_{x^0}[2] = \begin{pmatrix} 10 & 6 \\ 13 & 5 \end{pmatrix}$ with $val\big(M_{x^0}[2]\big) = 6$ and $\rho^0_{21} = 0,\ \rho^0_{22} = 1$.

Iteration 1:

$n = 1$; $r_2(1, \rho^0) = 6,\ r_2(2, \rho^0) = 5,\ p_{21}(\rho^0) = 0,\ p_{22}(\rho^0) = 1$.

The linear program for $SCSG(\rho^0)$ is

$$min \left\{ v_1 + v_2 \;\middle|\; \begin{array}{rcrcrcrl} v_1 & - & \frac{1}{2}v_2 & - & 3\rho_{11} & - & \rho_{12} & \geq 0 \\ \frac{1}{2}v_1 & & & - & \rho_{11} & - & 4\rho_{12} & \geq 0 \\ & & \frac{1}{2}v_2 & & & & & \geq 6 \\ & & & & \rho_{11} & + & \rho_{12} & = 1 \\ & & & & & & \rho_{11},\ \rho_{12} & \geq 0 \end{array} \right\}.$$

The solution of this program is: $v_1 = \frac{29}{4},\ v_2 = 12,\ \rho_{11} = \frac{1}{8},\ \rho_{12} = \frac{7}{8}$ and $x^1 = (7.25, 12)$.

$i = 2$: $M_{x^1}[2] = \begin{pmatrix} 7\frac{5}{8} & 12 \\ 10\frac{5}{8} & 11 \end{pmatrix}$ with $val\big(M_{x^1}[2]\big) = 10\frac{5}{8}$ and $\rho^1_{21} = 1,\ \rho^1_{22} = 0$; $y^1_2 = 10\frac{5}{8}$.

Iteration 2:

$n = 2$; $r_2(1, \rho^1) = 4,\ r_2(2, \rho^1) = 7,\ p_{21}(\rho^1) = 1,\ p_{22}(\rho^1) = 0$.

The linear program for $SCSG(\rho^1)$ is

$$min \left\{ v_1 + v_2 \;\middle|\; \begin{array}{rcrcrcrl} v_1 & - & \frac{1}{2}v_2 & - & 3\rho_{11} & - & \rho_{12} & \geq 0 \\ \frac{1}{2}v_1 & & & - & \rho_{11} & - & 4\rho_{12} & \geq 0 \\ -\frac{1}{2}v_1 & + & v_2 & & & & & \geq 7 \\ & & & & \rho_{11} & + & \rho_{12} & = 1 \\ & & & & & & \rho_{11},\ \rho_{12} & \geq 0 \end{array} \right\}$$

with solution: $v_1 = 6.62,\ v_2 = 10.31,\ \rho_{11} = 0.23,\ \rho_{12} = 0.77$ and $x^2 = (6.62, 10.31)$.

$i = 2$: $M_{x^2}[2] = \begin{pmatrix} 7.31 & 11.15 \\ 10.31 & 10.15 \end{pmatrix}$ with $val\big(M_{x^2}[2]\big) = 10.19$ and $\rho^2_{21} = \frac{1}{4},\ \rho^2_{22} = \frac{3}{4}$; $y^2_2 = 10.19$.

Iteration 3:

$n = 3$; $r_2(1, \rho^2) = 5\frac{1}{2},\ r_2(2, \rho^2) = 5\frac{1}{2},\ p_{21}(\rho^2) = \frac{1}{4},\ p_{22}(\rho^2) = \frac{3}{4}$.

The linear program for $SCSG(\rho^2)$ is

$$min \left\{ v_1 + v_2 \;\middle|\; \begin{array}{rcrcrcrl} v_1 & - & \frac{1}{2}v_2 & - & 3\rho_{11} & - & \rho_{12} & \geq 0 \\ \frac{1}{2}v_1 & & & - & \rho_{11} & - & 4\rho_{12} & \geq 0 \\ -\frac{1}{8}v_1 & + & \frac{5}{8}v_2 & & & & & \geq 5\frac{1}{2} \\ & & & & \rho_{11} & + & \rho_{12} & = 1 \\ & & & & & & \rho_{11},\ \rho_{12} & \geq 0 \end{array} \right\}$$

with solution: $v_1 = 6.54,\ v_2 = 10.11,\ \rho_{11} = \frac{1}{4},\ \rho_{12} = \frac{3}{4}$ and $x^3 = (6.54, 10.11)$.

$i = 2$: $M_{x^3}[2] = \begin{pmatrix} 7.27 & 11.05 \\ 10.27 & 10.05 \end{pmatrix}$ with $val\big(M_{x^3}[2]\big) = 10.11$ and $\rho^3_{21} = \frac{1}{4},\ \rho^3_{22} = \frac{3}{4}$; $y^3_2 = 10.11$.

Since $y^3_2 = x^3_2 = 10.11$, we have found the optimal solution: $v^\alpha = (6.54, 10.11),\ \rho^*_{21} = \rho^3_{21} = \frac{1}{4}$, $\rho^*_{22} = \rho^3_{22} = \frac{3}{4}$. From the optimal solution of the matrix game $M_{x^3}[2]$ we also obtain $\pi^*_{21} = 0.055$, $\pi^*_{22} = 0.945$. The optimal solution of the linear program $SCSG(\rho^2)$ provides $\rho^*_{11} = \rho_{11} = \frac{1}{4}$, $\rho^*_{12} = \rho_{22} = \frac{3}{4}$. In order to find π^*_{11} and π^*_{12} we have to solve the dual of $SCSG(\rho^2)$, i.e. the linear program

$$max \left\{ 5\tfrac{1}{2}y + z \;\middle|\; \begin{array}{rcrcrcrl} x_{11} & + & \frac{1}{2}x_{12} & - & \frac{1}{8}y & & & = 1 \\ -\frac{1}{2}x_{11} & & & + & \frac{5}{8}y & & & = 1 \\ -3x_{11} & - & x_{12} & & & + & z & \leq 0 \\ -x_{11} & - & 4x_{12} & & & + & z & \leq 0 \\ & & & & & & \multicolumn{2}{r}{x_{11},\ x_{12},\ y \geq 0} \end{array} \right\}.$$

The optimal solution of this program is: $x_{11} = 0.97,\ x_{12} = 0.65,\ y = 2.38$ and $z = 3.57$. Hence, $\pi^*_{11} = \frac{0.97}{1.62} = 0.6$ and $\pi^*_{12} = \frac{0.65}{1.62} = 0.4$.

Lemma 10.14

For $n = 1, 2, \ldots,$ we have $x^{n+1} \leq x^n$. Furthermore, if $val\big(M_{x^n}[i]\big) \neq x^n_i$ for some $i \in S_2$, then $x^{n+1} < x^n$, i.e. $x^{n+1}_i \leq x^n_i,\ i \in S$, with at least one strict inequality.

Proof

x^n is the value vector of the single-controller stochastic game $SCSG(\rho^{n-1})$. Therefore, we have $x^n_i = val\big(M^{\rho^{n-1}}_{x^n}[i]\big),\ i \in S$. Since $M^{\rho^{n-1}}_{x^n}[i] = M_{x^n}[i],\ i \in S_1$, and ρ^n is optimal policy for player 2 in the matrix game $M_{x^n}[i]$ for all $i \in S_2$, we have

$$x^n_i = val\big(M_{x^n}[i]\big),\ i \in S_1 \text{ and } x^n_i = max_a\,\{r_i(a, \rho^n) + \alpha \sum_j p_{ij}(\rho^n)x^n_j\},\ i \in S_2. \tag{10.38}$$

Let $\{\rho^n_{ib},\ i \in S_1,\ b \in B(i)\}$ be the optimal strategy for player 2 in the matrix games $M_{x^n}[i],\ i \in S_1$. From (10.38) and the definition of ρ^n it follows that

$$x^n_i \geq r_i(a, \rho^n) + \alpha \sum_j p_{ij}(\rho^n)x^n_j,\ i \in S_1,\ a \in A(i). \tag{10.39}$$

By (10.38) and (10.39), we obtain

$$x^n \geq r(f, \rho^n) + \alpha P(f, \rho^n)x^n \text{ for all } f^\infty \in C(D), \tag{10.40}$$

from which it follows that $x^n \geq v^\alpha\big(f^\infty, (\rho^n)^\infty\big)$ for all $f^\infty \in C(D)$. Hence, we can write

$$\begin{array}{rcl} x^n & \geq & max_{f^\infty \in C(D)}\, v^\alpha\big(f^\infty, (\rho^n)^\infty\big) = max_{\pi^\infty \in \Pi}\, v^\alpha\big(\pi^\infty, (\rho^n)^\infty\big) \\ & \geq & max_{\pi^\infty \in \Pi}\, inf_{\{\rho^\infty \in \Gamma \,|\, \rho_{ib} = \rho^n_{ib},\ i \in S_2,\ b \in B(i)\}}\, v^\alpha\big(\pi^\infty, (\rho^n)^\infty\big) \\ & = & \text{value vector of } SCSG(\rho^n) = x^{n+1}, \end{array}$$

which proves the first part of the lemma. If $val\big(M_{x^n}[i]\big) \neq x^n_i$ for some $i \in S_2$, then (10.40) holds with a strict inequality for at least one $i \in S_2$, i.e. $x^n > r(f, \rho^n) + \alpha P(f, \rho^n)x^n$ for all $f^\infty \in C(D)$, implying $x^n > x^{n+1}$. □

Lemma 10.15

Let $C = (a_{ij} + b_j)$ be a square and nonsingular matrix with $a_{ij} > 0$ for all i, j. Furthermore, let $Cx = \gamma \cdot e$ have a nonnegative solution x with $\sum_i x_i = 1$. Then, the matrix $A = (a_{ij})$ is nonsingular and $Ax = \delta \cdot e$ for some scalar d.

Proof

Assume that A is singular. Then, there exists a $y \neq 0$ with $Ay = 0$. Therefore, we have $\sum_j c_{ij}y_j = \sum_j a_{ij}y_j + \sum_j b_jy_j = \sum_j b_jy_j$ for all i. Hence, $Cy = \beta \cdot e$ with $\beta := \sum_j b_jy_j$, implying $y = \beta \cdot C^{-1}e$. Because $y \neq 0$, also $\beta \neq 0$. Furthermore, we have $x = \gamma \cdot C^{-1}e \neq 0$. Thus, $y = \frac{\beta}{\gamma} \cdot x$ and $\sum_j y_j = \frac{\beta}{\gamma} \neq 0$. Therefore, we may assume that y is such that $Ay = 0$ and $\sum_j y_j = 1$. From $\sum_j y_j = \frac{\beta}{\gamma} = 1$ it follows that $\beta = \gamma$, so $y = x$ and $Ax = 0$. However, since A has positive entries and x is a probability vector, $Ax = 0$ is impossible and we have shown that A is nonsingular. Since $\sum_j a_{ij}y_j = \sum_j c_{ij}y_j - \sum_j b_jy_j = \gamma - \sum_j b_jy_j$, which is independent of i, we obtain $Ax = \delta \cdot e$ for some scalar d. □

Without loss of generality, we may assume that $r_i(a, b) > 0$ for all i, a, b (otherwise add a positive scalar $c > -min_{i,a,b}\, r_i(a, b)$ to all these rewards). In that case all elements of the matrix game $M_{x^n}[i]$ are also strictly positive if $x^n \geq 0$. From a theorem by Shapley and Snow ([268]), and also from the linear programming approach of matrix games, we know that optimal strategies can be found in a submatrix game, where this submatrix is square and nonsingular. For any $i \in S_2$, the matrix game $M_{x^n}[i]$, has elements $r_i(a, b) + \alpha \sum_j p_{ij}(b)x_j$. These elements are of the type of the elements of C in Lemma 10.15 (a and b in the elements of $M_{x^n}[i]$ play the role of i and j in C). Therefore, Lemma 10.15 implies that an extreme optimal strategy for player 2 in the matrix game $M_{x^n}[i]$ is also an extreme optimal action in some square and nonsingular submatrix of a matrix with elements $r_i(a, b)$, $(a, b) \in A(i) \times B(i)$ (the matrix with elements $r_i(a, b)$ corresponds to matrix A in Lemma 10.15. Since there are only a finite number of submatrices from the matrix $\big(r_i(a, b)\big)$, we see that for all $i \in S_2$ and $n \in \mathbb{N}$, the extreme optimal strategy ρ^n_{ib}, $b \in B(i)$, is chosen from a finite set.

Theorem 10.16

Algorithm 10.8 is finite and correct.

Proof

Assume that the algorithm is not finite. Then, by Lemma 10.14, $x^1 > x^2 > \cdots > x^n > \cdots$. Hence, the subsequent extreme strategies ρ^n, $n = 1, 2, \cdots$ are different. Since, in step 3 of Algorithm 10.8 the extreme optimal strategies $\rho^n_{ib}, b \in B(i)$, are chosen from a finite set (see above), this yields a contradiction. Hence, the algorithm is finite.

Let the algorithm terminate at the nth iteration, i.e. $x^n = val(M_{x^n}[i])$ for all $i \in S_2$. Since we always have $x^n = val(M_{x^n}[i])$ for all $i \in S_1$ (see (10.38)), we have $x^n = val(M_{x^n})$ implying that x^n is the value vector v^α of the discounted stochastic game.

The optimal stationary strategies in the matrix games $M_{x^n}[i]$, $i \in S$, say π^n_{ia}, $a \in A(i)$ and ρ^n_{ib}, $b \in B(i)$, are optimal policies in the stochastic game, because $r(\pi, \rho^n) + \alpha P(\pi, \rho^n)v^\alpha \leq v^\alpha$ implies $v^\alpha\big(\pi^\infty, (\rho^n)^\infty\big) = \{I - \alpha P(\pi, \rho^n)\}^{-1}r(\pi, \rho^n) \leq v^\alpha$ for all $\pi^\infty \in \Pi$. Similarly, we derive $v^\alpha\big((\pi^n)^\infty, \rho^\infty\big) \geq v^\alpha$ for all $\rho^\infty \in \Gamma$. □

Remark

The correctness of Algorithm 10.8 provides also the proof that the value vector and the optimal policies lie in the same ordered field as the data: linear programming is used and the data of the stochastic games $SCSG(\rho^n)$ are also in the same field. Hence, the ordered field property holds also for switching control stochastic games.

SER-SIT stochastic game

In this game we assume that the rewards are *separable*, i.e. $r_i(a,b) = s_i + t(a,b)$ for all $i.a, b$ (SER property), and the transitions are *state independent*, i.e. $p_{ij}(a,b) = p_j(a,b),\ j \in S$ for all $i.a, b$ (SIT property). Note that the above is meaningful if the set $\{(a,b)\}$ is independent of the states. Therefore, we assume that $|A(i)| = m$ and $|B(i)| = n$ for all $i \in S$. Thus a fixed pair of actions (a,b) determines the same transition law, $p_j(a,b),\ j \in S$, in every $i \in S$. In addition, the SER property implies that all rewards are a sum of a contribution due the current state (s_i) and a contribution due to the action pair selected ($t(a,b)$).
Let $s = (s_1, s_2, \dots, s_N)^T$ and define the $m \times n$ matrix $M = (m_{ab})$ by $m_{ab} := t(a,b) + \alpha \sum_j p_j(a,b)s_j$, $1 \le a \le m,\ 1 \le b \le n$, which is, unlike the matrix $M_x[i]$ in the previous section, independent of the state i.

Lemma 10.16
Let $\pi = (\pi_1, \pi_2, \dots, \pi_m)$ and $\rho = (\rho_1, \rho_2, \dots, \rho_n)$ be an arbitrary pair of mixed strategies of the matrix game with matrix M. Then, $v^\alpha(\pi^\infty, \rho^\infty) = s + (1-\alpha)^{-1}\pi^T M\rho \cdot e$.

Proof
Since $v^\alpha(\pi^\infty, \rho^\infty) = r(\pi,\rho) + \alpha P(\pi,\rho)v^\alpha(\pi^\infty,\rho^\infty) = s + t(\pi,\rho)\cdot e + \alpha P(\pi,\rho)v^\alpha(\pi^\infty,\rho^\infty)$, we have

$$v_i^\alpha(\pi^\infty,\rho^\infty) - s_i = t(\pi,\rho) + \alpha \textstyle\sum_j p_j(\pi,\rho)s_j + \alpha \sum_j p_j(\pi,\rho)\{v_j^\alpha(\pi^\infty,\rho^\infty) - s_j\},\ i \in S.$$

In vector notation: $\{I - \alpha P(\pi,\rho)\}\{v^\alpha(\pi^\infty,\rho^\infty) - s\} = t(\pi,\rho)\cdot e + \alpha P(\pi,\rho)s = \pi^T M\rho \cdot e$. Hence, $v^\alpha(\pi^\infty,\rho^\infty) - s = \pi^T M\rho \cdot \{I - \alpha P(\pi,\rho)\}^{-1}\cdot e = \pi^T M\rho \cdot \sum_{t=0}^\infty \{\alpha P(\pi,\rho)\}^t \cdot e = \pi^T M\rho\cdot(1-\alpha)^{-1}\cdot e$, i.e. $v^\alpha(\pi^\infty,\rho^\infty) = s + (1-\alpha)^{-1}\pi^T M\rho \cdot e$. □

Corollary 10.4
Let $v^* := val(M)$ and let $\pi^* = (\pi_1^*, \pi_2^*, \dots, \pi_m^*)$ and $\rho^* = (\rho_1^*, \rho_2^*, \dots, \rho_n^*)$ be a pair of optimal mixed strategies of the matrix game with matrix M. Then the value vector of the stochastic game $v^\alpha = s + \frac{1}{1-\alpha}v^* \cdot e$ and $(\pi^*)^\infty$ and $(\rho^*)^\infty$ are optimal policies for player 1 and player 2, respectively.

Proof
Since π^* and ρ^* are optimal strategies for the matrix game with matrix M, we have for all strategies π and ρ: $\pi^T M\rho^* \le (\pi^*)^T M\rho^* \le (\pi^*)^T M\rho$. Therefore, by Lemma 10.16, for all $\pi^\infty \in \Pi$ and all $\rho^\infty \in \Gamma$: $v^\alpha\big(\pi^\infty, (\rho^*)^\infty\big) \le v^\alpha\big((\pi^*)^\infty, (\rho^*)^\infty\big) \le v^\alpha\big((\pi^*)^\infty, \rho^\infty\big)$. Hence, by Theorem 10.4, the value vector of the stochastic game is $s + \frac{1}{1-\alpha}v^* \cdot e$ and $(\pi^*)^\infty$ and $(\rho^*)^\infty$ are optimal policies for player 1 and player 2, respectively. □

Algorithm 10.9 *SER-SIT game with discounting*

Input: Instance of a two-person SER-SIT stochastic game

Output: The value vector v^α and a pair $\big((\pi^*)^\infty, (\rho^*)^\infty\big)$ of optimal stationary policies.

1. Compute the matrix M with entries $m_{ab} := t(a,b) + \alpha \sum_j p_j(a,b)s_j,\ a \in A(i),\ b \in B(i)$.
2. Determine the value v^* and optimal mixed strategies π^* and ρ^* of the matrix game with matrix M.
3. $v^\alpha := s + \frac{1}{1-\alpha} v^* \cdot e$ is the value vector; $(\pi^*)^\infty$ and $(\rho^*)^\infty$ are optimal stationary policies for player 1 and player 2, respectively (STOP).

Remark

Since the value v^* and the optimal stationary strategies π^* and ρ^* can be computed by linear programming, *SER-SIT* games possess the ordered field property.

Example 10.7

Consider the following example, which is a *SIT* game but no *SER* game.
$S = \{1,2,3\}$; $A(1) = A(2) = A(3) = \{1,2\}$; $B(1) = B(2) = B(3) = \{1,2\}$.
$r_1(1,1) = 0,\ r_1(1,2) = 0,\ r_1(2,1) = 0,\ r_1(2,2) = 1;\ r_2(1,1) = -1,\ r_2(1,2) = -1;$
$r_2(2,1) = -1,\ r_2(2,2) = -1,\ r_3(1,1) = -2,\ r_3(1,2) = -2;\ r_3(2,1) = 2,\ r_3(2,2) = 1.$
$p_1(1,1) = 1,\ p_2(1,1) = 0,\ p_3(1,1) = 0;\ p_1(1,2) = 0,\ p_2(1,2) = 0,\ p_3(1,2) = 1;$
$p_1(2,1) = 0,\ p_2(2,1) = 1,\ p_3(2,1) = 0;\ p_1(2,2) = 1,\ p_2(2,2) = 0,\ p_3(2,2) = 0.$
For the value vector v^α, we have

$$v_1^\alpha = val\begin{pmatrix} \alpha v_1^\alpha & \alpha v_3^\alpha \\ \alpha v_2^\alpha & \alpha v_1^\alpha \end{pmatrix};\ v_2^\alpha = val\begin{pmatrix} -1+\alpha v_1^\alpha & -1+\alpha v_3^\alpha \\ -1+\alpha v_2^\alpha & -1+\alpha v_1^\alpha \end{pmatrix};\ v_3^\alpha = val\begin{pmatrix} -2+\alpha v_1^\alpha & -2+\alpha v_3^\alpha \\ -1+\alpha v_2^\alpha & -1+\alpha v_1^\alpha \end{pmatrix}.$$

The matrix of v_1^α has entries which are all 1 larger than the entries of the matrix of v_2^α: $v_1^\alpha = v_2^\alpha + 1$. Furthermore, we see in the matrix of v_3^α that the entry at position (2,1), i.e. $-1+\alpha v_2^\alpha$, is the largest in the first column and the smallest in the second row. So, this entry is a saddle point, i.e. $v_3^\alpha = -1 + \alpha v_2^\alpha$. Hence, we obtain from the equation for v_1^α,

$$\tfrac{1}{\alpha} v_1^\alpha = val\begin{pmatrix} v_1^\alpha & v_3^\alpha \\ v_2^\alpha & v_1^\alpha \end{pmatrix} = val\begin{pmatrix} v_1^\alpha & \alpha v_1^\alpha - 1 - \alpha \\ v_1^\alpha - 1 & v_1^\alpha \end{pmatrix}.$$

Hence, $\frac{1}{\alpha} v_1^\alpha = \frac{v_1^\alpha \cdot v_1^\alpha - (v_1^\alpha - 1)\cdot(\alpha v_1^\alpha - 1 - \alpha)}{v_1^\alpha + v_1^\alpha - (v_1^\alpha - 1) - (\alpha v_1^\alpha - 1 - \alpha)} = \frac{(1-\alpha)\big((v_1^\alpha)^2 + v_1^\alpha(1+2\alpha) - (1+\alpha)}{(1-\alpha)v_1^\alpha + 2 + \alpha}$.

The solution of this quadratic equation yields $v_1^\alpha = \frac{-(1+\alpha)+\sqrt{(1+\alpha)}}{1-\alpha}$. Let $\alpha = \frac{1}{2}$, then $v_1^\alpha = -3+\sqrt{6}$. So, we conclude that the *SIT* game without the *SER* property does not possess the ordered field property.

Remark

It can also be shown (see Exercise 10.9) that a *SER* game without the *SIT* property does not possess the ordered field property.

SER-SIT/SC stochastic game

We have seen that both the Switching-controller and the *SER-SIT* stochastic game have the ordered field property. A natural generalization of the above types of games is a game where in some states the law of transition is as in switching control and in the rest of the states the game is *SER-SIT*. We call such games *SER-SIT/SC* stochastic games.

A zero-sum stochastic game is a *SER-SIT/SC* stochastic game if:

(1) $S = S1 \cup S_2 \cup S_3$, where $S_1 \cap S_2 = S_1 \cap S_3 = S_2 \cap S_3 = \emptyset$;

(2) $p_{ij}(a) = \begin{cases} p_{ij}(a),\ i \in S_1, a \in A(i),\ b \in B(i),\ j \in S \\ p_{ij}(b),\ i \in S_2, a \in A(i),\ b \in B(i),\ j \in S \end{cases}$

(3a) $r_i(a,b) = s_i + t(a,b),\ i \in S_3,\ a \in A(i),\ b \in B(i)$;

(3b) $p_{ij}(a,b) = p_j(a,b),\ i \in S_3,\ a \in A(i),\ b \in B(i),\ j \in S$;

(3c) $|A(i)| = m$ and $|B(i)| = n$ for all $i \in S_3$.

A first question is: *Do these games also possess the ordered field property?* Unfortunately the answer is no as the next example shows.

Example 10.4 (continued)
Take $S_1 = 2,\ S_2 = \emptyset,\ S_3 = \{1\}$. Notice that this is trivially a *SER-SIT/SC* stochastic game. Since $v^\alpha = \frac{2}{3}\{-2 + \sqrt{13}\}$, the ordered field property does not hold.

Remark

One might wonder whether there exists a subclass which has the ordered field property. Sinha ([272]) claims that under the assumption that $\sum_{j \in S_3} p_j(a,b)$ is constant for all $(a,b) \in A(i) \times B(i)$, $i \in S_3$ this game has the ordered field property. Furthermore, he presents for both the discounted and the undiscounted case finite algorithms. These algorithms contain a finite sequence of linear programs and matrix games as in the switching-controller stochastic games.

ARAT stochastic game

An additive reward and additive transition (*ARAT*) stochastic game is defined by the property that the rewards as well as the transitions can be written as the sum of a term determined by player 1 and a term determined by player 2: $r_i(a,b) = r_i^1(a) + r_i^2(b),\ i \in S,\ a \in A(i),\ b \in B(i)$ and $p_{ij}(a,b) = p_{ij}^1(a) + p_{ij}^2(b),\ i,j \in S,\ a \in A(i),\ b \in B(i)$.

Theorem 10.17
(1) Both players have optimal deterministic and stationary policies.
(2) The ordered field property holds.

Proof

(1) By the additivity of $r_i(a,b)$ and $p_{ij}(a,b)$, the matrix $M_x[i]$, with entries $r_i(a,b) + \alpha \sum_j p_{ij}(a,b)x_j$ can be written as the sum of two matrices: $M_x[i] = A_x[i] + B_x[i]$, where $A_x[i]$ and $B_x[i]$ have

elements $r_i^1(a) + \alpha \sum_j p_{ij}^1(a)x_j$ and $r_i^2(b) + \alpha \sum_j p_{ij}^2(b)x_j$, respectively. The matrix $A_x[i]$ has identical columns and matrix $B_x[i]$ has identical rows. Consider the equation $x_i = val(M_x[i])$, which has as unique solution v_i^α. Effectively, this means that player 1 is only interested in the matrix $A_{v^\alpha}[i]$ with identical columns, and player 2 is only interested in the matrix $B_{v^\alpha}[i]$ with identical rows. Hence, in each state i, both players possess deterministic optimal strategies. Hence, the stochastic game has optimal deterministic and stationary policies.

(2) Let f_*^∞ and g_*^∞ be optimal deterministic and stationary optimal policies for player 1 and 2, respectively. Then, the value vector $v^\alpha = v^\alpha(f_*^\infty, g_*^\infty) = \{I - \alpha P(f,g)\}^{-1} r(f,g)$, which shows the ordered field property. □

Since there are only a finite number of deterministic and stationary policies, there is a finite algorithm. The next algorithm is a special version of Algorithm 10.4 with $\varepsilon = 0$.

Algorithm 10.10 *ARAT game with discounting*
Input: Instance of a two-person *ARAT* stochastic game
Output: The value vector v^α and a pair (f_*^∞, g_*^∞) of deterministic optimal policies.

```
1. Select any deterministic policy $g_*^\infty$ for player 2.

2. Solve the MDP induced by the policy $g_*^\infty$: $x := max_{f^\infty \in C(D)} v^\alpha(f^\infty, g^\infty)$.

3. for all $i \in S$ do
   begin determine the matrix $M_x[i]$ with entries $r_i(a,b) + \alpha \sum_j p_{ij}(a,b)x_j,\ a \in A(i),\ b \in B(i)$;
         compute $y_i := val(M_x[i])$;
         determine a deterministic optimal strategy $g_*(i)$ for player 2 in the matrix $M_x[i]$
   end

4. if $\|y - x\|_\infty = 0$ then go to step 5
   else return to step 2.

5. for all $i \in S$ do
      determine an optimal deterministic strategy $f_*(i)$ for player 1 in the matrix $M_x[i]$.

6. $v^\alpha := y$.

7. $v^\alpha$ is the value vector and $(f_*^\infty, g_*^\infty)$ is a pair of deterministic optimal policies (STOP).
```

10.3 Total rewards

10.3.1 Value and optimal policies

We make the following assumptions:

(1) The model is *substochastic*, i.e. $\sum_j p_{ij}(a,b) \leq 1$ for all $(i,a,b) \in S \times A \times B$.

(2) The model is *transient*, i.e. for any initial state i and any two policies R_1, R_2 the expected total reward $v_i(R1, R2)$ is finite.

A policy R_1^* is optimal for player 1 if $v(R_1^*, R_2) \geq \inf_{R_2} \sup_{R_1} v(R_1, R_2)$ for all policies R_2.
A policy R_2^* is optimal for player 2 if $v(R_1, R_2^*) \leq \sup_{R_1} \inf_{R_2} v(R_1, R_2)$ for all policies R_1.
The stochastic game with total rewards has a *value* if $\inf_{R_2} \sup_{R_1} v(R_1, R_2) = \sup_{R_1} \inf_{R_2} v(R_1, R_2)$.
The value vector is denoted by v.

Most of the results for transient Markov games are similar to the results for discounted Markov games. Below we give an overview of these results; for the proofs we often refer to the section discounted rewards.

Theorem 10.18
If the policies R_1^* and R_2^* satisfy $v(R_1, R_2^*) \leq v(R_1^*, R_2^*) \leq v(R_1^*, R_2)$ for all policies R_1 and R_2, then the game has a value, and R_1^* and R_2^* are optimal policies.

Proof
The proof is analogous to the proof of Theorem 10.4. □

Theorem 10.19
The game has a value and both players have optimal policies.

Proof
We shall use the following properties:

(1) When one of the players uses a stationary policy, then the Markov game is an MDP for which the other player is the decision maker.

(2) An MDP has an optimal deterministic policy.

(3) $v(\pi^\infty, \rho^\infty)$ is finite and equal to $\{I - P(\pi, \rho)\}^{-1} r(\pi, \rho)$ for every $\pi^\infty \in C_1(S)$ and every $\rho^\infty \in C_2(S)$.

Let $w := \inf_{\rho^\infty \in \Gamma} \sup_{\pi^\infty \in \Pi} v(\pi^\infty, \rho^\infty)$. Define for any $x \in \mathbb{R}$ the mapping $T : \mathbb{R}^N \to \mathbb{R}^N$ by

$$(Tx)_i := \inf_{\rho^\infty \in \Gamma} \sup_{\pi^\infty \in \Pi} \{r_i(\pi, \rho) + \sum_j p_{ij}(\pi, \rho) x_j\}, \ i \in S. \tag{10.41}$$

$(Tx)_i$ is the value of a matrix game with pay-off matrix $M_x[i]$. The matrix $M_x[i]$ has $m = |A(i)|$ rows and $n = |B(i)|$ columns and the payoff, if player 1 chooses row a and player 2 column b, is $r_i(a,b) + \sum_j p_{ij}(a,b) x_j$.

Let $w_i(\rho^\infty) := \sup_{\pi^\infty \in \Pi} v(\pi^\infty), \rho^\infty)$ and $w_i(\pi^\infty) := inf_{\rho^\infty \in Gamma} v(\pi^\infty, \rho^\infty)$ for every $\rho^\infty \in \Gamma$ and every $\pi^\infty \in \Pi$. Obviously, $w \le w(\rho^\infty)$ for every $\rho^\infty \in \Gamma$. If we fix $\rho^\infty \in \Gamma$ as policy for player 2, the game becomes a transient MDP. From the results of Section 4.7 it follows that $w(\rho^\infty)$ is the unique solution of the equation $x = \sup_{\pi^\infty} \{r(\pi, \rho) + P(\pi, \rho)x\}$. Hence,

$$\begin{aligned} Tw &= \inf_{\rho^\infty \in \Gamma} \sup_{\pi^\infty \in \Pi} \{r(\pi, \rho) + P(\pi, \rho)w\} \\ &\le \inf_{\rho^\infty \in \Gamma} \sup_{\pi^\infty \in \Pi} \{r(\pi, \rho) + P(\pi, \rho)w(\rho^\infty)\} \qquad (10.42) \\ &= \inf_{\rho^\infty \in \Gamma} w(\rho^\infty) = \inf_{\rho^\infty \in \Gamma} \sup_{\pi^\infty \in \Pi} v(\pi^\infty, \rho^\infty) = w. \end{aligned}$$

Since $(Tw)_i$ is the value of the matrix game $M_w[i]$, there are optimal strategies $\pi^*_{ia},\ a \in A(i)$, and $\rho^*_{ib},\ b \in B(i)$, such that

$$r_i(\pi, \rho^*) + \textstyle\sum_j p_{ij}(\pi, \rho^*)w_j \le (Tw)_i = r_i(\pi^*, \rho^*) + \sum_j p_{ij}(\pi^*, \rho^*)w_j \le r_i(\pi^*, \rho) + \sum_j p_{ij}(\pi^*, \rho)w_j$$

for all strategies $\pi_{ia},\ a \in A(i)$ and $\rho_{ib},\ b \in B(i)$. In vector notation,

$$r(\pi, \rho^*) + P(\pi, \rho^*)w \le Tw = r(\pi^*, \rho^*) + P(\pi^*, \rho^*)w \le r(\pi^*, \rho) + P(\pi^*, \rho)w,\ \pi^\infty \in \Pi,\ \rho^\infty \in \Gamma. \qquad (10.43)$$

Suppose that $Tw \ne w$. Then, it follows from (10.42) and (10.43) that

$$w_i \ge (Tw)_i \ge \{r(\pi, \rho^*) + P(\pi, \rho^*)w\}_i,\ i \in S,\ \pi^\infty \in \Pi, \qquad (10.44)$$

where the first inequality is strict for at least one i, say for $i = k$. By iterating (10.44), we obtain

$$w_k > \{\textstyle\sum_{t=1}^\infty P^{t-1}(\pi, \rho^*)r(\pi, \rho^*)\}_k = v_k\big(\pi^\infty, (\rho^*)^\infty\big) \text{ for every } \pi^\infty \in \Pi.$$

Then, it follows that

$$w_k > max_{\pi^\infty \in \Pi} v_k\big(\pi^\infty, (\rho^*)^\infty\big) = \sup_{\pi^\infty \in \Pi} v_k\big(\pi^\infty, (\rho^*)^\infty\big) \ge \inf_{\rho^\infty \in \Gamma} \sup_{\pi^\infty \in \Pi} v_k(\pi^\infty, \rho^\infty) = w_k,$$

implying a contradiction. Hence, $Tw = w$ and $r(\pi, \rho^*) + P(\pi, \rho^*)w \le w \le r(\pi^*, \rho) + P(\pi^*, \rho)w$ for every $\pi^\infty \in \Pi$ and $\rho^\infty \in \Gamma$. Consequently,

$$v\big(\pi^\infty, (\rho^*)^\infty\big) \le v\big((\pi^*)^\infty, (\rho^*)^\infty\big) \le v\big((\pi^*)^\infty, \rho^\infty\big) \text{ for every } \pi^\infty \in \Pi \text{ and } \rho^\infty \in \Gamma. \qquad (10.45)$$

Since in any Markov decision problem an optimal policy can be found in the class of stationary policies, we also have $v\big(R_1, (\rho^*)^\infty\big) \le v\big((\pi^*)^\infty, (\rho^*)^\infty\big) \le v\big((\pi^*)^\infty, R_2\big)$ for every pair (R_1, R_2) of policies for player 1 and 2, respectively. By Theorem 10.18, the game has a value and both players have stationary optimal policies. □

10.3.2 Mathematical programming

A vector $v \in \mathbb{R}^N$ is called *superharmonic* if there exists a policy $\rho^\infty \in \Gamma$ such that

$$v_i \ge r_i(a, \rho) + \textstyle\sum_j p_{ij}(a, \rho)v_j,\ a \in A(i),\ i \in S.$$

A vector $v \in \mathbb{R}^N$ is called *subharmonic* if there exists a policy $\pi^\infty \in \Pi$ such that

$$v_i \le r_i(\pi, b) + \textstyle\sum_j p_{ij}(\pi, b)v_j,\ b \in B(i),\ i \in S.$$

Theorem 10.20

(1) The value vector v is the smallest superharmonic vector.

(2) The value vector v is the largest subharmonic vector.

Proof

The proof is analogous to the proof of Theorem 10.9. □

Consider the two nonlinear programs

$$min \left\{ \sum_i v_i \;\middle|\; \begin{array}{rrl} \sum_j \{\delta_{ij} - \sum_b p_{ij}(a,b)\rho_{ib}\} v_j & - \quad \sum_b r_i(a,b)\rho_{ib} & \geq 0, \; a \in A(i), \; i \in S \\ & \sum_b \rho_{ib} & = 1, \; i \in S \\ & \rho_{ib} & \geq 0, \; b \in B(i), \; i \in S \end{array} \right\} \tag{10.46}$$

and

$$max \left\{ \sum_i w_i \;\middle|\; \begin{array}{rrl} \sum_j \{\delta_{ij} - \sum_a p_{ij}(a,b)\pi_{ia}\} w_j & - \quad \sum_a r_i(a,b)\pi_{ia} & \leq 0, \; b \in B(i), \; i \in S \\ & \sum_a \pi_{ia} & = 1, \; i \in S \\ & \pi_{ia} & \geq 0, \; a \in A(i), \; i \in S \end{array} \right\}. \tag{10.47}$$

Theorem 10.21

The nonlinear programs (10.46) and (10.47) have both optimal solutions, say (v^*, ρ^*) and (w^*, π^*). Furthermore, $v^* = w^* = v$, the value vector, and $(p^*)^\infty$ and $(\rho^*)^\infty$ are optimal policies for player 1 and player 2, respectively.

Proof

The proof is analogous to the proof of Theorem 10.10. □

Consider for a given initial distribution β, i.e. $\beta_j \geq 0, \; j \in S$, and $\sum_j \beta_j = 1$, the nonlinear system

$$\left\{ \begin{array}{l} \sum_a \pi_{ia} = 1, \; i \in S; \; \pi_{ia} \geq 0, \; a \in A(i), \; i \in S \\ \sum_b \rho_{ib} = 1, \; i \in S; \; \rho_{ib} \geq 0, \; b \in B(i), \; i \in S \\ \sum_i \{\delta_{ij} - p_{ij}(\pi,\rho)\} x_i = \beta_j, \; j \in S \end{array} \right. \tag{10.48}$$

Any solution (π, ρ, x) of this system satisfies $x^T\{I - P(\pi,\rho)\} = \beta^T$. By iterating this equality, we obtain $x^T = \beta^T \sum_{t=1}^\infty P^{t-1}(\pi,\rho)$, from which it follows that x_i is the expected number of times that the process visits state i, given initial distribution β and the stationary policies π^∞ and ρ^∞ for player 1 and 2, respectively. Furthermore, we have

$$\beta^T v(\pi^\infty, \rho^\infty) = \beta^T \sum_{t=1}^\infty P^{t-1}(\pi,\rho) r(\pi,\rho) = x^T r(\pi,\rho) = \sum_i \sum_a \sum_b r_i(a,b) x_i \pi_{ia} \rho_{ib}.$$

Conversely, any pair of decision rules (π, ρ) gives a solution of (10.48) with $x^T = \beta^T \sum_{t=1}^\infty P^{t-1}(\pi,\rho)$. Inequality (10.45) implies

$$\beta^T v = \min_{\rho^\infty \in \Gamma} \max_{\pi^\infty \in \Pi} \beta^T v(\pi^\infty, \rho^\infty) = \min_{\rho^\infty \in \Gamma} \max_{\pi^\infty \in \Pi} \sum_i \sum_a \sum_b r_i(a,b) x_i \pi_{ia} \rho_{ib}.$$

Hence, we can state the following result.

Theorem 10.22
$\beta^T v$ is the value of the following minimax game problem:
$min_\rho \max_\pi \{\sum_i \sum_a \sum_b r_i(a,b)x_i\pi_{ia}\rho_{ib} \mid (\pi,\rho,x)$ is a feasible solution of (10.48)$\}$.

10.3.3 Single-controller stochastic game: the transient case

In the single-controller stochastic game is player 1 the single-controller. This means that the transition probabilities $p_{ij}(a,b)$ are independent of b. Therefore, we denote these probabilities as $p_{ij}(a)$. Under this assumption the nonlinear program (10.46) (with objective function $\sum_j \beta_j v_j$ instead of $\sum_j v_j$, where $\beta_j > 0,\ j \in S$) becomes the following linear program

$$min\left\{\sum_i \beta v_i \;\middle|\; \begin{array}{rcl} \sum_j\{\delta_{ij}-p_{ij}(a)\}v_j \; - \; \sum_b r_i(a,b)\rho_{ib} & \geq & 0,\ a\in A(i),\ i\in S \\ \sum_b \rho_{ib} & = & 1,\ i\in S \\ \rho_{ib} & \geq & 0,\ b\in B(i),\ i\in S \end{array}\right\}. \tag{10.49}$$

The dual program is

$$max\left\{\sum_i z_i \;\middle|\; \begin{array}{rcl} \sum_{(i,a)}\{\delta_{ij}-p_{ij}(a)\}x_i(a) & = & \beta_j,\ j\in S \\ -\sum_a r_i(a,b)x_i(a) \; + \; z_i & \leq & 0,\ (i,b)\in S\times B \\ x_i(a) & \geq & 0,\ (i,a)\in S\times A \end{array}\right\}. \tag{10.50}$$

The following theorem shows that the value vector and optimal stationary policies for both players can be obtained from the optimal solutions of the dual pair of linear programs.

Theorem 10.23
Let (v^*,ρ^*) and (x^*,z^*) be optimal solutions of the linear programs (10.49) and (10.50), respectively. Define the stationary policy $(\pi^*)^\infty$ by $\pi^*_{ia} := \frac{x^*_i(a)}{\sum_a x^*_i(a)}$, $(i,a)\in S\times A$. Then, v^* is the value vector and $(\pi^*)^\infty$ and $(\rho^*)^\infty$ are optimal stationary policies for player 1 and 2, respectively.

Proof
The proof is analogous to the proof of Theorem 10.15. □

Algorithm 10.11 *Single-controller game (transient case)*
Input: Instance of a two-person single-controller transient stochastic game
Output: The value vector v^* and a pair $\big((\pi^*)^\infty,(\rho^*)^\infty\big)$ of stationary optimal policies.

1. Compute optimal solutions (v^*,ρ^*) and (x^*,z^*) of the linear programs (10.49) and (10.50).
2. Define the stationary policy $(\pi^*)^\infty$ by $\pi^*_{ia} := \frac{x^*_i(a)}{\sum_a x^*_i(a)}$, $(i,a)\in S\times A$.
3. v^* is the value vector and $(\pi^*)^\infty$ and $(\rho^*)^\infty$ optimal stationary policies for player 1 and 2 (STOP).

Additional constraints

We assume that the constraints are imposed on the expected total state-action frequencies for the player who controls the transitions (player 1). For the additional constraints we assume that, besides the immediate rewards, there are for $k = 1, 2, \ldots, m$ also certain immediate costs $c_i^k(a),\ (i, a) \in S \times A$. The constraints are:

$c_k(R_1) := \sum_j \beta_j \cdot \sum_{t=1}^{\infty} \sum_{i,a} \mathbb{P}\{X_t = i,\ Y_t = a \mid X_1 = j\} \cdot c_i^k(a) \leq b_k$ for $k = 1, 2, \ldots, m$,

for some real numbers $b_1, b_2, \ldots, b_m$ and some initial distribution β with $\beta_j > 0,\ j \in S$.
Let C^1, C^2 be the set of policies for player 1 and 2, respectively. For any policy $R_1 \in C^1$ we denote the total expected number of times of being in state i and choosing action a by

$$x_{ia}(R_1) = \sum_j \beta_j \cdot \sum_{t=1}^{\infty} \sum_{i,a} \mathbb{P}\{X_t = i,\ Y_t = a \mid X_1 = j\}. \qquad (10.51)$$

Let $C^1(S), C^2(S)$ be the set of stationary policies for player 1 and 2, respectively. We define the vector sets $K, K(S)$ and P, with components $(i, a) \in S \times A$, by

$$\begin{array}{rcl} K & = & \{x(R_1) \mid R_1 \in C^1\}; \\ K(S) & = & \{x(R_1) \mid R_1 \in C^1(S)\}; \\ P & = & \left\{ x \;\middle|\; \begin{array}{rcl} \sum_{(i,a)} \{\delta_{ij} - p_{ij}(a)\} x_{ia} & = & \beta_j,\ j \in S \\ x_{ia} & = & 0,\ (i,a) \in S \times A \end{array} \right\}. \end{array}$$

Theorem 10.24
$K = K(S) = P$.

Proof
The result was shown in Theorem 9.18. □

Let $C_0^1 := \{R_1 \in C^1 \mid c_k(R_1) \leq b_k$ for $k = 1, 2, \ldots, m\}$ the set of feasible solutions for player 1. A policy R_1^* is *optimal* for player 1 in the constrained Markov game if $R^* \in C_0^1$ and

$$\inf_{R_2 \in C^2} \sum_j \beta_j v_j(R_1^*, R_2) = \sup R_1 \in C_0^1 \inf_{R_2 \in C^2} \sum_j \beta_j v_j(R_1, R_2). \qquad (10.52)$$

A policy R_2^* is *optimal* for player 2 in the constrained Markov game if

$$\sup_{R_1 \in C_0^1} \sum_j \beta_j v_j(R_1, R_2^*) = \inf_{R_2 \in C^2} \sup_{R_1 \in C_0^1} \sum_j \beta_j v_j(R_1, R_2). \qquad (10.53)$$

The constrained Markov game has a *value* if

$$\sup_{R_1 \in C_0^1} \inf_{R_2 \in C^2} \sum_j \beta_j v_j(R_1, R_2) = \inf_{R_2 \in C^2} \sup_{R_1 \in C_0^1} \sum_j \beta_j v_j(R_1, R_2). \qquad (10.54)$$

From Theorem 10.24 it follows that for any $R_1 \in C^1$ there exists $x \in P$ such that $x = x(R_1)$. Since $c_k(R_1) = \sum_{i,a} x_{ia}(R_1)c_i^k(a)$ for $k = 1, 2, \dots, m$, the constrained Markov game can be converted in the following polyhedral game

$$\sup_{P_0} \inf_{\rho^\infty \in \Gamma} \sum_{i,a} \sum_b r_i(a,b)x_{ia}\rho_{ib} \text{ where } P_0 := \{x \in P \mid \sum_{i,a} c_i^k(a)x_{ia} \le b_k,\ 1 \le k \le m\}. \quad (10.55)$$

Theorem 10.25

Let (x^*, z^*) and (v^*, w^*, ρ^*) be optimal solutions of the following dual pair of linear programs

$$max \left\{ \sum_i z_i \;\middle|\; \begin{array}{rcrcl} \sum_{(i,a)}\{\delta_{ij} - p_{ij}(a)\}x_i(a) & & & = & \beta_j,\ j \in S \\ -\sum_a r_i(a,b)x_i(a) & + & z_i & \le & 0,\ (i,b) \in S \times B \\ \sum_{(i,a)} c_i^k(a,b)x_i(a) & & & \le & b_k,\ k = 1, 2, \dots, m \\ & & x_i(a) & \ge & 0,\ (i,a) \in S \times A \end{array} \right\} \quad (10.56)$$

and

$$min \left\{ \sum_j \beta_j v_j + \sum_k b_k w_k \;\middle|\; \begin{array}{rcl} \sum_j \{\delta_{ij} - p_{ij}(a)\}v_j - \sum_b r_i(a,b)\rho_{ib} + \sum_k c_i^k(a)w_k & \ge & 0,\ a \in A(i),\ i \in S \\ \sum_b \rho_{ib} & = & 1,\ i \in S \\ \rho_{ib} & \ge & 0,\ b \in B(i),\ i \in S \\ w_k & \ge & 0,\ k = 1, 2, \dots, m \end{array} \right\} \quad (10.57)$$

Define the stationary policy $(\pi^*)^\infty$ by $\pi^*_{ia} := \frac{x_i^*(a)}{\sum_a x_i^*(a)}$, $(i,a) \in S \times A$. Then, $(\pi^*)^\infty$ and $(\rho^*)^\infty$ are optimal stationary policies for the constrained Markov game and $\sum_i z_i^* = \sum_j \beta_j v_j^* + \sum_k b_k w_k^*$ is the value of the constrained game.
If program (10.56) is infeasible, then $C_0^1 = \emptyset$.

Proof

If program (10.56) is infeasible, then obviously $C_0^1 = \emptyset$.
If program (10.56) is feasible, then - since P is a compact set - (10.56) has a finite optimal solution, say (x^*, z^*). Consequently, (10.57) has also a finite optimal solution, say (v^*, w^*, ρ^*).
We have to show that for every $R_1 \in C_0^1$ and every $R_2 \in C^2$

$$\sum_j \beta_j v_j\big(R_1, (\rho^*)^\infty\big) \le \sum_j \beta_j v_j\big((\pi^*)^\infty, (\rho^*)^\infty\big) = \sum_j \beta_j v_j^* + \sum_k b_k w_k^* = \sum_i z_i^* \le \sum_j \beta_j v_j\big((\pi^*)^\infty, R_2\big).$$

We have for every $x \in P_0$,

$$\begin{array}{rcl} \sum_{(i,a)} \sum_b r_i(a,b)\rho_{ib}^* x_i(a) & \le & \sum_{(i,a)} \{ \sum_j \{\delta_{ij} - p_{ij}(a)\}v_j^* + \sum_k c_i^k(a)w_k^*\}x_i(a) \\ & = & \sum_j \{ \sum_{(i,a)} \{\delta_{ij} - p_{ij}(a)\}v_j^*\}x_i(a) + \sum_k \{ \sum_{(i,a)} c_i^k(a)w_k^*\}x_i(a) \\ & = & \sum_j \{ \sum_{(i,a)} \{\delta_{ij} - p_{ij}(a)\}x_i(a)\}v_j^* + \sum_k \{ \sum_{(i,a)} c_i^k(a)x_i(a)\}w_k^* \\ & \le & \sum_j \beta_j v_j^* + \sum_k b_k w_k^* = \sum_i z_i^*. \end{array}$$

Furthermore, we obtain for every $\rho^\infty \in C^2(S)$

$$\begin{array}{rcl} \sum_{(i,a)} \sum_b r_i(a,b)\rho_{ib}x_i^*(a) & = & \sum_{(i,b)} \rho_{ib}\{\sum_a r_i(a,b)x_i^*(a)\} \\ & \ge & \sum_{(i,b)} \rho_{ib}z_i^* = \sum_i\{\sum_b \rho_{ib}\}z_i^* = \sum_i z_i^*. \end{array}$$

Hence, we have

$$\sum_{(i,a)} \sum_b r_i(a,b)\rho^*_{ib}x_i(a) \leq \sum_j \beta_j v^*_j + \sum_k b_k w^*_k = \sum_i z^*_i \leq \sum_{(i,a)} \sum_b r_i(a,b)\rho_{ib}x^*_i(a)$$

for every $x \in P_0$ and every $\rho^\infty \in C^2(S)$. Consequently,

$$\sum_{(i,a,b)} r_i(a,b)\rho^*_{ib}x_i(a) \leq \sum_j \beta_j v^*_j + \sum_k b_k w^*_k = \sum_{(i,a,b)} r_i(a,b)\rho^*_{ib}x^*_i(a) = \sum_i z^*_i \leq \sum_{(i,a,b)} r_i(a,b)\rho_{ib}x^*_i(a) \tag{10.58}$$

for every $x \in P_0$ and every $\rho^\infty \in C^2(S)$. Take any $R_1 \in C^1_0$ and any $\rho^\infty \in C^2(S)$. Let $x \in P_0$ be such that $x = x(R_1)$. Then,

$$\begin{array}{rcl} \sum_{(i,a,b)} r_i(a,b)\rho^*_{ib}x_i(a) & = & \sum_b \rho^*_{ib}\{\sum_{(i,a)} r_i(a,b)x_{ia}(R_1)\} \\ & = & \sum_{(i,a)} r_i(a,\rho^*)x_{ia}(R_1) = \sum_j \beta_j v_j\big(R_1,\rho^\infty\big) \end{array}$$

and

$$\begin{array}{rcl} \sum_{(i,a,b)} r_i(a,b)\rho_{ib}x^*_i(a) & = & \sum_b \rho_{ib}\{\sum_{(i,a)} r_i(a,b)\pi^*_{ia}x^*_i\} \\ & = & \sum_i r_i(\pi^*,\rho)\{\beta^T\big(I - P(\pi^*)\big)^{-1}\}_i \\ & = & \beta^T\big(I - P(\pi^*)\big)^{-1}r(\pi^*,\rho) = \sum_j \beta_j v_j\big((\pi^*)^\infty,\rho^\infty\big). \end{array}$$

Therefore, these equalities and (10.58) imply

$$\begin{array}{rcl} \sum_j \beta_j v_j\big(R_1,(\rho^*)^\infty\big) & = & \sum_{(i,a,b)} r_i(a,b)\rho^*_{ib}x_i(a) \\ & \leq & \sum_{(i,a,b)} r_i(a,b)\rho_{ib}x^*_i(a) \\ & = & \sum_j \beta_j v_j\big((\pi^*)^\infty,\rho^\infty\big) \end{array}$$

for every $R_1 \in C^1_0$ and every $\rho^\infty \in C^2(S)$. Since the game becomes an MDP if player 1 uses the stationary policy $(\pi^*)^\infty$, we also have

$$\sum_j \beta_j v_j\big(R_1,(\rho^*)^\infty\big) \leq \sum_j \beta_j v_j\big((\pi^*)^\infty,(\rho^*)^\infty\big) \leq \sum_j \beta_j v_j\big((\pi^*)^\infty, R_2$$

for every $R_1 \in C^1_0$ and every $R_2 \in C^2$, i.e. $(\pi^*)^\infty$ and $(\rho^*)^\infty$ are optimal stationary policies for the constrained Markov game. Furthermore, $\sum_i z^*_i = \sum_j \beta_j v^*_j + \sum_k b_k w^*_k = \sum_j \beta_j v_j\big((\pi^*)^\infty,(\rho^*)^\infty\big)$ is the value of the constrained game. □

Algorithm 10.12 *Single-controller constrained Markov game (transient case)*
Input: Instance of a two-person single-controller constrained transient stochastic game.
Output: The value and a pair $(\pi^*)^\infty$ and $(\rho^*)^\infty$ of stationary optimal policies (in case the constrained game is feasible).

1. Solve the linear programs (10.56) and (10.57), respectively.

2. **if** (10.56) is infeasible **then** the constrained game is infeasible (STOP).

3. Let (x^*, z^*) and (v^*, w^*, ρ^*) be optimal solutions of (10.56) and (10.57), respectively.

4. Define the stationary policy $(\pi^*)^\infty$ by $\pi^*_{ia} := \frac{x^*_i(a)}{\sum_a x^*_i(a)}$, $(i,a) \in S \times A$.

5. $\sum_i z^*_i$ is the value and $(\pi^*)^\infty$ and $(\rho^*)^\infty$ are optimal stationary policies for player 1 and 2, respectively (STOP).

Remark 1

Since the discounted Markov game is a special case of a transient Markov game, Algorithm 10.12 can also be used for discounted Markov games with constraints.

Remark 2

Consider a *two-person zero-sum discounted semi-Markov game* in which player 1 controls the transitions. This model can be described as follows:
- state space S;
- action sets $A(i)$ and $B(i),\ i \in S$, for player 1 and 2, respectively;
- transition probabilities $p_{ij}(a),\ (i,a) \in S \times A,\ j \in S$, which depend only on the actions chosen by player 1;
- immediate rewards $r_i(a,b),\ (i,a,b) \in S \times A \times B$;
- reward rates $s_i(a,b),\ (i,a) \in S \times A \times B$;
- sojourn time distributions $F_{ij}(a,t),\ (i,a) \in S \times A,\ j \in S$, which depend only on the actions chosen by player 1.

From these quantities we compute the transition numbers $p^*_{ij}(a),\ (i,a) \in S \times A,\ j \in S$, and the rewards $r^*(a,b),\ (i,a) \in S \times A \times B,\ j \in S$, by:

$p^*_{ij}(a) := p_{ij}(a) \cdot \int_0^\infty e^{-\lambda t} dF_{ij}(a,t)$ for every $(i,a) \in S \times A$ and $j \in S$;

$r^*_i(a,b) := r_i(a,b) + s_i(a,b) \cdot \sum_j p_{ij}(a) \int_0^\infty \{\int_0^t e^{-\lambda s} ds\}\, dF_{ij}(a,t)$ for every $(i,a,b) \in S \times A \times B$.

Analogously to the analysis in Section 9.7.4 it can straightforward be shown that this discounted semi-Markov game is contracting and equivalent to a transient Markov game (S,A,B,p^*,r^*) with total rewards. Therefore, the results of a single-controller transient Markov game are also applicable to a discounted single-controller semi-Markov game.

10.3.4 Single-controller stochastic game: the general case

In this subsection we drop the assumption that the model is transient; so, there may be nontransient policies. We relax the assumption of transiency to the following.

Assumption 10.1

For any initial state i and any two policies R_1, R_2 the expected total reward $v_i(R_1,R_2)$ exists, possibly $+\infty$ or $-\infty$.

Since the transition probabilities are controlled by player 1, the concept of a transient policy is only significant for policies $R_1 \in C_1$. Let C^1_T be the set of transient policies for player 1, i.e.

$$C^1_T := \Big\{ R_1 \in C^1 \;\Big|\; \sum_{t=1}^\infty \sum_{i,a} \mathbb{P}\{X_t = i,\ Y_t = a \mid X_1 = j\} < \infty \text{ for all } j \in S \Big\}. \tag{10.59}$$

We shall discuss the problem of finding the best policies R_1 and R_2 with $R_1 \in C^1_T$ and $R_2 \in C^2$. Let β be an initial distribution with $\beta_j > 0,\ j \in S$. We define the vector sets $L, L(S)$ and P, with components $(i,a) \in S \times A$, by:

$$\begin{array}{rcl} L & = & \{x(R_1) \mid R_1 \in C^1_T\}; \\ L(S) & = & \{x(R_1) \mid R_1 \in C^1_T \cap C^1(S)\}; \\ P & = & \left\{ x \,\middle|\, \begin{array}{rcl} \sum_{(i,a)} \{\delta_{ij} - p_{ij}(a)\} x_{ia} & = & \beta_j,\ j \in S \\ x_{ia} & = & 0,\ (i,a) \in S \times A \end{array} \right\}. \end{array}$$

Then, by Theorem 9.16, $L = L(S) = P$. For $x \in P$ we define a stationary policy $\pi^\infty(x)$ by (4.16). Conversely, let π^∞ be an arbitrary transient stationary policy. Then, define the vector $x(\pi)$ by (4.18). By Theorem 4.7, we know that the mapping (4.18) is a bijection between the set of transient stationary policies and the set of feasible solutions of (4.15) with (4.16) as the inverse mapping.

Theorem 10.26

Consider the dual pair of linear programs (10.49) and (10.50).

(1) If (10.50) is infeasible, then $C^1_T = \emptyset$.

(2) If (10.50) is unbounded, then there does not exist a finite value of the stochastic game.

(3) If (x^*, z^*) and (v^*, ρ^*) are optimal solutions of (10.49) and 10.50, respectively, then $\pi^\infty(x^*)$ and $(\rho^*)^\infty$ are optimal transient policies for the two players and v^* is the value of this game, i.e. $v^* = \sup_{R_1 \in C^1_T} \inf_{R_2 \in C^2} v(R_1, R_2) = \inf_{R_2 \in C^2} \sup_{R_1 \in C^1_T} v(R_1, R_2)$.

Proof

(1) Assume that $R_1 \in C^1_T$. Then, $x(R_1) \in L = P$. So, there exists $x \in P$ such that $x = x(R_1)$. Let $z_i := \min_b \sum_a r_i(a,b) x_i(a),\ i \in S$. Then, (x,z) is a feasible solution of (10.50), which causes a contradiction. Hence, $C^1_T = \emptyset$.

(2) For any feasible solution (x,z) of (10.50), we have $\sum_i z_i \le \sum_i \sum_a \sum_b r_i(a,b) x_i(a) \rho_{ib}$ for every $\rho^\infty \in C^2(S)$. Hence, $\sum_i z_i \le \inf_{\rho^\infty \in C^2(S)} \sum_i \sum_a r_i(a,\rho) x_i(a)$. Because (10.50) is unbounded, we obtain $\sup_{x \in P} \inf_{\rho^\infty \in C^2(S)} \sum_i \sum_a r_i(a,\rho) x_i(a) = +\infty$.
Since $\sum_i \sum_a r_i(a,\rho) x_i(a) = \sum_j \beta_j v_j\big(\pi^\infty(x), \rho^\infty\big)$, we obtain

$$\begin{array}{rcl} \inf_{R_2 \in C^2} \sup_{R_1 \in C^1_T} \sum_j v_j(R_1,R_2) & \ge & \sup_{R_1 \in C^1_T} \inf_{R_2 \in C^2} \sum_j v_j(R_1,R_2) \\ & \ge & \sup_{\pi^\infty \in C^1_T \cap C^1(S)} \inf_{R_2 \in C^2} \sum_j v_j(\pi^\infty, R_2) \\ & = & \sup_{\pi^\infty \in C^1_T \cap C^1(S)} \inf_{\rho^\infty \in C^2(S)} \sum_j v_j(\pi^\infty, \rho^\infty) \\ & = & \sup_{\pi^\infty \in C^1_T \cap C^1(S)} \inf_{\rho^\infty \in C^2(S)} \sum_i \sum_a r_i(a,\rho) x_i(a) = +\infty. \end{array}$$

Hence, there does not exist a finite value vector.

(3) The proof can be given analogously to the proof of Theorem 10.25). □

Algorithm 10.13 *Single-controller constrained Markov game (general case)*

Input: Instance of a two-person single-controller constrained stochastic game.

Output: The value vector and a pair $(\pi^*)^\infty$ and $(\rho^*)^\infty$ of stationary optimal policies (in case the linear program (10.50) is feasible and bounded).

1. Solve the dual pair of linear programs (10.49) and (10.50), respectively.
2. **if** (10.50) is infeasible **then** $C^1_T = \emptyset$ (STOP).
3. **if** (10.50) is unbounded **then** there does not exist a finite value of the stochastic game (STOP).
4. Let (v^*, ρ^*) and (x^*, z^*) and be optimal solutions of (10.49) and (10.50), respectively.
5. Define the stationary policy $(\pi^*)^\infty$ by $\pi^*_{ia} := \frac{x^*_i(a)}{\sum_a x^*_i(a)}$, $(i,a) \in S \times A$.
6. v^* is the value and $(\pi^*)^\infty$ and $(\rho^*)^\infty$ are optimal stationary policies for player 1 and 2, respectively (STOP).

Additional constraints

Let us consider the constrained Markov game. As before, we assume that the constraints are imposed on the expected total state-action frequencies for player 1. We also assume that, besides the immediate rewards $r_i(a)$, there are also certain immediate costs $c^k_i(a),\ (i,a) \in S \times A$, for $k = 1, 2, \dots, m$. The constraints are:

$$c_k(R_1) := \sum_j \beta_j \cdot \sum_{t=1}^\infty \sum_{i,a} \mathbb{P}\{X_t = i,\ Y_t = a \mid X_1 = j\} \cdot c^k_i(a) \leq b_k \text{ for } k = 1, 2, \dots, m,$$

for some real numbers $b_1, b_2, \dots, b_m$ and some initial distribution β with $\beta_j > 0,\ j \in S$. The policies for player 1 are restricted to to the set C^1_*, where

$$C^1_* := \{R_1 \in C^1_T \mid c_k(R_1) \leq b_k,\ k = 1, 2, \dots, m\}.$$

Then, with similar arguments as used in Theorem 10.25 and Theorem 10.26 the following result can be shown.

Theorem 10.27

Consider the dual pair of linear programs (10.56) and (10.57).

(1) If (10.56) is infeasible, then $C^1_* = \emptyset$.

(2) If (10.56) is unbounded, then there does not exist a finite value of the stochastic game.

(3) If (x^*, z^*) and (v^*, w^*, ρ^*) are optimal solutions of (10.56) and 10.57, respectively, then $\pi^\infty(x^*)$ and $(\rho^*)^\infty$ are optimal transient policies for the two players and the value of this constrained game is $\sum_i z^*_i = \sum_j \beta_j v^*_j + \sum_k b_k w^*_k$.

Algorithm 10.14 *Single-controller constrained Markov game (general case)*

Input: Instance of a two-person single-controller constrained stochastic game.

Output: The value and a pair $(\pi^*)^\infty$ and $(\rho^*)^\infty$ of stationary optimal policies (in case the the linear program (10.56) is feasible and bounded).

1. Solve the linear programs (10.56) and (10.57), respectively.
2. **if** (10.56) is infeasible **then** the constrained game is infeasible (STOP).
3. **if** (10.56) is unbounded **then** there does not exist a finite value of the constrained game (STOP).
4. Let (x^*, z^*) and (v^*, w^*, ρ^*) be optimal solutions of (10.56) and (10.57), respectively.
5. Define the stationary policy $(\pi^*)^\infty$ by $\pi^*_{ia} := \frac{x^*_i(a)}{\sum_a x^*_i(a)}$, $(i, a) \in S \times A$.
6. $\sum_i z^*_i$ is the value and $(\pi^*)^\infty$ and $(\rho^*)^\infty$ are optimal stationary policies for player 1 and 2, respectively (STOP).

10.4 Average rewards

10.4.1 Value and optimal policies

A policy R_1^* is *optimal for player 1* if $\phi(R_1^*, R_2) \geq inf_{R_2}\, sup_{R_1}\, \phi(R_1, R_2)$ for all policies R_2.

A policy R_2^* is *optimal for player 2* if $\phi(R_1, R_2^*) \leq sup_{R_1}\, inf_{R_2}\, \phi(R_1, R_2)$ for all policies R_1.

The stochastic undiscounted game has a *value* if $inf_{R_2}\, sup_{R_1}\, \phi(R_1, R_2) = sup_{R_1}\, inf_{R_2}\, \phi(R_1, R_2)$.

The value vector of an undiscounted stochastic game is denoted by ϕ.

A policy R_1^* is *ε-optimal for player 1* if $\phi(R_1^*, R_2) \geq inf_{R_2} sup_{R_1} \phi(R_1, R_2) - \varepsilon$ for all policies R_2.

A policy R_2^* is *ε-optimal for player 2* if $\phi(R_1, R_2^*) \leq sup_{R_1} inf_{R_2} \phi(R_1, R_2) + \varepsilon$ for all policies R_1.

Theorem 10.28
If the policies R_1^* and R_2^* satisfy $\phi(R_1, R_2^*) \leq \phi(R_1^*, R_2^*) \leq \phi(R_1^*, R_2)$ for all policies R_1 and R_2, the game has a value and R_1^* and R_2^* are optimal policies.

Proof
The proof is analogous to the proof of Theorem 10.4. □

We have seen in Chapter 5 that, for Markov decision processes, the average reward criterion is considerably more difficult to analyze than the discounted reward criterion. Nonetheless, after overcoming a number of technical difficulties, results of qualitative strength and generality were established for both the discounted and the average reward criterium. For instance, we have seen that in both cases there are optimal deterministic and stationary optimal policies and that they can be found by policy iteration, linear programming and value iteration methods. Consequently, one might think that in the case of stochastic games (perhaps at the cost of extra analysis) it might be possible to obtain qualitatively the same results in the average and the discounted case. Unfortunately, this is not the case. In the next section we shall illustrate some of the problems that arise.

10.4.2 The Big Match

The seemingly simple example described below can be used to illustrate many of the difficulties arising in the analysis of stochastic games under the average reward criterion.

Example 10.8 *The Big Match*
$S = \{1,2,3\};\ A(1) = B(1) = \{1,2\},\ A(2) = B(2) = A(3) = B(3) = \{1\}.$
$r_1(1,1) = 1,\ r_1(1,2) = 0,\ r_1(2,1) = 0,\ r_1(2,2) = 1;\ r_2(1,1) = 0;\ r_3(1,1) = 1.$
$p_{11}(1,1) = 1,\ p_{12}(1,1) = 0,\ p_{13}(1,1) = 0;\ p_{11}(1,2) = 1,\ p_{12}(1,2) = 0,\ p_{13}(1,2) = 0;$
$p_{11}(2,1) = 0,\ p_{12}(2,1) = 1,\ p_{13}(2,1) = 0;\ p_{11}(2,2) = 0,\ p_{12}(2,2) = 0,\ p_{13}(2,2) = 1.$
$p_{21}(1,1) = 0,\ p_{22}(1,1) = 1,\ p_{23}(1,1) = 0;\ p_{31}(1,1) = 0,\ p_{32}(1,1) = 0,\ p_{33}(1,1) = 1.$
The states 2 and 3 are absorbing: $\phi_2(R_1, R_2) = 0$ and $\phi_3(R_1, R_2) = 1$ for all policies R_1 and R_2. However, it seems that the structure of the transition data makes the choice for player 1 in state 1 extremely difficult. While the choice of the first action leads to a repetition of the same game, the choice of the second action absorbs the game either in state 2 or state 3, depending on the choice of player 2. Thus the consequence of the second choice is so permanent and with such different payoffs that it is a risky action.
To make the above point more precise, suppose that player 1 uses a stationary policy π^∞ with probability p for action 1 and probability $1-p$ for action 2 in state 1, and that player 2 uses a stationary policy ρ^∞ with probability q for action 1 and probability $1-q$ for action 2 in state 1. Let $P[p,q]$ and $r[p,q]$ be the corresponding transition matrix and reward vector.
There are now two cases: $p = 1$ and $0 \le p < 1$.
<u>Case 1: $p = 1$</u>

$$P[p,q] = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix};\ r[p,q] = (q,0,1)^T \ \rightarrow\ P^*[p,q] = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \ \rightarrow\ \phi_1(\pi^\infty, \rho^\infty) = q.$$

<u>Case 2: $0 \le p < 1$</u>

$$P[p,q] = \begin{pmatrix} p & (1-p)q & (1-p)(1-q) \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix};\ r[p,q] = (pq + (1-p)(1-q), 0, 1)^T.$$

$$P^*[p,q] = \begin{pmatrix} 0 & q & 1-q \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \ \rightarrow\ \phi_1(\pi^\infty, \rho^\infty) = 1 - q.$$

Hence,

$$\max\nolimits_{0\le p\le 1}\ \phi_1(\pi^\infty,\rho^\infty) = \max\nolimits_{0\le q\le 1}(q, 1-q) \ \rightarrow\ \min\nolimits_{0\le q\le 1}\max\nolimits_{0\le p\le 1}\ \phi_1(\pi^\infty,\rho^\infty) = \tfrac{1}{2}$$

and

$$\min\nolimits_{0\le q\le 1}\ \phi_1(\pi^\infty,\rho^\infty) = \min\nolimits_{0\le q\le 1}(q, 1-q) = 0 \ \rightarrow\ \max\nolimits_{0\le p\le 1}\min\nolimits_{0\le q\le 1}\ \phi_1(\pi^\infty,\rho^\infty) = 0.$$

Therefore,

$$\max_{\pi^\infty\in\Pi}\ \min_{\rho^\infty\in\Gamma}\ \phi_1(\pi^\infty,\rho^\infty) = 0 < \frac{1}{2} = \min_{\rho^\infty\in\Gamma}\ \max_{\pi^\infty\in\Pi}\ \phi_1(\pi^\infty,\rho^\infty). \qquad (10.60)$$

Of course, the above strict inequality implies that optimal stationary policies do not exist in the Big Match. Since we alway have $sup_{R_1}\, inf_{R_2}\, \phi(R_1, R_2) \leq inf_{R_2}\, sup_{R_1}\, \phi(R_1, R_2)$, it is sufficient for the existence of the value of this stochastic game to show that there exitst a policy R_1^* for player 1 such that $\phi_1(R_1^*, R_2) \geq \frac{1}{2} - \varepsilon$ for every $\varepsilon > 0$ and every policy R_2 for player 2, namely: In that case we have:

$$\begin{array}{rcl} sup_{R_1}\, inf_{R_2}\, \phi_1(R_1, R_2) & \geq & inf_{R_2}\, \phi_1(R_1^*, R_2) \\ & \geq & \frac{1}{2} = min_{\rho^\infty \in \Gamma}\, max_{\pi^\infty \in \Pi}\, \phi_1(\pi^\infty, \rho^\infty) \\ & = & min_{\rho^\infty \in \Gamma}\, sup_{R_1}\, \phi_1(R_1, \rho^\infty) \\ & \geq & inf_{R_2}\, sup_{R_1}\, \phi_1(R_1, R_2). \end{array}$$

Thus, in order to show that the Big Match has a value vector, it is sufficient to show that for any $M \in \mathbb{N}$ there exists a policy R_1^M for player 1 such that, for any realization $Z = (Z_1, Z_2, \dots)$ of the actions of player 2, player 1 has an average reward of at least $\frac{1}{2} \cdot \frac{M}{M+1}$.
At decision point $t+1$ player 1 knows the realizations of $Z_1, Z_2, \dots, Z_t$, say $b_1, b_2, \dots, b_t$, where $b_i \in \{1, 2\}$, $1 \leq i \leq t$. Let k_t^1 be the number of 1's and k_t^2 be the number of 2's in $\{b_1, b_2, \dots, b_t\}$, and let $k_t = k_t^1 - k_t^2$ for $t = 1, 2, \dots$. The policy R_1^M is history-dependent, but depends only on the numbers k_t. Let $\pi^{t+1}(k_t)$ be the probability that player 1 chooses action 2 in state 1 at time point $t+1$, given k_t. Then, in policy R_1^M, we take

$$\pi^{t+1}(k_t) := \frac{1}{(k_t + M + 1)^2} \text{ for } t = 0, 1, 2, \dots, \text{ where } k_0 \equiv 0.$$

Intuitively, when k_t is positive and large, i.e. player 2 seems more willing for action 1, which leads - when player 1 chooses action 2 - to the for player 1 bad state 2, then the probability that player 1 chooses action 2 is very small; when k_t is negative and tends to $-M$, i.e. player 2 seems more willing for action 2, which leads - when player 1 chooses action 2 - to the for player 1 good state 3, then the probability that player 1 chooses action 2 is increasing to 1.

Lemma 10.17
Define the events E_m by $E_m := \{Y_1 = Y_2 = \cdots = Y_m = 1$ or $Y_n = Z_n = 2$ for some $1 \leq n \leq m\}$ for $m = 1, 2, \dots$. Then, $\mathbb{P}_{R_1^M}\{E_m\} \geq \frac{1}{2} \cdot \frac{M}{M+1}$ for all $m, M \in \mathbb{N}$.

Proof
The proof is inductively on m.
If $m = 1$, then $E_m = \{Y_1 = 1$ or $Y_1 = Z_1 = 2\}$.
Hence, $\mathbb{P}_{R_1^M}\{E_m\} \geq \mathbb{P}_{R_1^M}\{Y_1 = 1\} = 1 - \frac{1}{(M+1)^2} \geq \frac{1}{2} \cdot \frac{M}{M+1}$ for all $M \in \mathbb{N}$.
Next, assume that $\mathbb{P}_{R_1^M}\{E_m\} \geq \frac{1}{2} \cdot \frac{M}{M+1}$ for all $M \in \mathbb{N}$ and consider E_{m+1}.
We distinguish between the two possibilities for Z_1.
Case 1: $Z_1 = 1$, i.e. $k_1 = 1$

$$\begin{array}{rcl} E_{m+1} & = & \{Y_1 = Y_2 = \cdots = Y_{m+1} = 1 \text{ or } Y_n = Z_n = 2 \text{ for some } 2 \leq n \leq m+1\} \\ & = & \big\{Y_1 = 1 \text{ or } \{Y_2 = \cdots = Y_{m+1} = 1 \text{ or } Y_n = Z_n = 2 \text{ for some } 2 \leq n \leq m+1\}\big\}. \end{array}$$

If $Y_1 = 2$, then the next state is state 2 and $Y_n = Z_n = 2$ for some $2 \leq n \leq m+1$ is impossible.

Therefore,
$\mathbb{P}_{R_1^M}\{E_{m+1}\} = \mathbb{P}_{R_1^M}\{Y_1 = 1\}\cdot\mathbb{P}_{R_1^M}\{Y_2 = \cdots = Y_{m+1} = 1 \text{ or } Y_n = Z_n = 2 \text{ for some } 2 \leq n \leq m+1\}$.
Because $k_1 = 1$,
$\mathbb{P}_{R_1^M}\{Y_2 = \cdots = Y_{m+1} = 1 \text{ or } Y_n = Z_n = 2 \text{ for some } 2 \leq n \leq m+1\} = \mathbb{P}_{R_1^{M+1}}\{E_m\}$.
Hence,
$\mathbb{P}_{R_1^M}\{E_{m+1}\} = \{1 - \frac{1}{(M+1)^2}\}\cdot\mathbb{P}_{R_1^{M+1}}\{E_m\} = \frac{M(M+2)}{(M+1)^2}\cdot\mathbb{P}_{R_1^{M+1}}\{E_m\} \geq \frac{M(M+2)}{(M+1)^2}\cdot\frac{1}{2}\cdot\frac{M+1}{M+2} = \frac{1}{2}\cdot\frac{M}{M+1}$.

Case 2: $Z_1 = 2$, i.e. $k_1 = -1$

$$\begin{aligned} E_{m+1} &= \{Y_1 = Y_2 = \cdots = Y_{m+1} = 1 \text{ or } Y_n = Z_n = 2 \text{ for some } 1 \leq n \leq m+1\} \\ &= \big\{Y_1 = Z_1 = 2 \text{ or } \{Y_1 = Y_2 = \cdots = Y_{m+1} = 1 \text{ or } Y_n = Z_n = 2 \text{ for some } 2 \leq n \leq m+1\}\big\}. \end{aligned}$$

Since $Z_1 = 2$, we have $\mathbb{P}_{R_1^M}\{Y_1 = Z_1 = 2\} = \mathbb{P}_{R_1^M}\{Y_1 = 2\} = \frac{1}{(M+1)^2}$.
If $Y_1 = 2$, then the next state is state 3 and $Y_n = Z_n = 2$ for some $2 \leq n \leq m+1$ is impossible.
Therefore,
$\mathbb{P}_{R_1^M}\{E_{m+1}\} = \frac{1}{(M+1)^2} + \mathbb{P}_{R_1^M}\{Y_1 = 1\}\,\mathbb{P}_{R_1^M}\{Y_2 = \cdots = Y_{m+1} = 1 \text{ or } Y_n = Z_n = 2 \text{ for some } 2 \leq n \leq m+1\}$.
Because $k_1 = -1$,
$\mathbb{P}_{R_1^M}\{Y_2 = \cdots = Y_{m+1} = 1 \text{ or } Y_n = Z_n = 2 \text{ for some } 2 \leq n \leq m+1\} = \mathbb{P}_{R_1^{M-1}}\{E_m\}$.
Hence,

$$\begin{aligned} \mathbb{P}_{R_1^M}\{E_{m+1}\} &= \frac{1}{(M+1)^2} + \Big\{1 - \frac{1}{(M+1)^2}\Big\}\cdot\mathbb{P}_{R_1^{M-1}}\{E_m\} \geq \frac{1}{(M+1)^2} + \frac{M(M+2)}{(M+1)^2}\cdot\frac{1}{2}\cdot\frac{M-1}{M} \\ &= \frac{1}{(M+1)^2}\cdot\{1 + \frac{1}{2}\cdot(M-1)(M+2)\} = \frac{1}{2}\cdot\frac{M}{M+1}. \end{aligned}$$

So, we have shown that $\mathbb{P}_{R_1^M}\{E_m\} \geq \frac{1}{2}\cdot\frac{M}{M+1}$ for all $m, M \in \mathbb{N}$. □

Lemma 10.18
$\phi_1(R_1^M, R_2) \geq \frac{1}{2}\cdot\frac{M}{M+1}$ for all $M \in \mathbb{N}$ and all policies R_2.

Proof

Take any $M \in \mathbb{N}$ and any policy R_2 with realization $Z_1, Z_2, \ldots$.

Consider the cases $k_t > -M$ for all t and $k_t = -M$ for some t separately.

Case 1: $k_t = -M$ for some t

In this case player 1 chooses at time $t+1$ (or earlier) action 2. Let m be the smallest time point for which $Y_m = 2$. Then, we have $\phi_1(R_1^M, R_2) = \mathbb{P}\{Z_m = 2\} = \mathbb{P}_{R_1^M}\{E_m\} \geq \frac{1}{2}\cdot\frac{M}{M+1}$ for any policy R_2 of player 2.

Case 2: $k_t > -M$ for all t

Let t be the smallest time point for which $Y_{t+1} = 2$ (if $Y_n = 1$ for all n, then $t = \infty$). Define for $m \geq 2$: $\lambda(m) = \mathbb{P}\{t < m \text{ and } Z_{t+1} = 1\}$ and $\mu(m) = \mathbb{P}\{t < m \text{ and } Z_{t+1} = 2\}$. Then, the sequences $\{\lambda(m)\}$ and $\{\mu(m)\}$ are nondecreasing. Let $\lambda = lim_{m\to\infty}\,\lambda(m)$ and $\mu = lim_{m\to\infty}\,\mu(m)$: λ is the probability that the game ends in state 2, μ the probability that the game ends in state 3, and $1 - \lambda - \mu$ is the probability that the game never leaves state 1. Since $k_t^1 + k_t^2 = t$ and $k_t = k_t^1 - k_t^2 > -M$ for all t, we have $k_t^1 > \frac{1}{2}(t - M)$: $\frac{k_t^1}{t} > \frac{1}{2}\big(1 - \frac{M}{t}\big)$ for all t, implying $\liminf_{t\to\infty}\frac{k_t^1}{t} \geq \frac{1}{2}$. Hence, $\phi_1(R_1^M, R_2) \geq \mu + (1 - \lambda - \mu)\cdot\frac{1}{2} = \frac{1}{2}(1 - \lambda + \mu)$.

Finally, we have to show that $\frac{1}{2}(1-\lambda+\mu) \geq \frac{M}{M+1}$. Therefore, consider the following policy for player 2: first he plays according to $Z_1, Z_2, \ldots, Z_m$ and thereafter he uses a fain coin, i.e. with probability $\frac{1}{2}$ he chooses action 1 and action 2. Then, the expected average reward for player 1 is: the probability to move from state 1 to state 3 during the first m time point plus the probability to be at time point $m+1$ in state 1 multiplied with the average reward from time point $m+1$. This yields $\mu(m) + \{1-\lambda(m)-\mu(m)\} \cdot \frac{1}{2}$.

On the other hand, any realization of this policy will, with probability 1, reach $k_t = -M$ for some t. Hence, by case 1 of the lemma, $\mu(m) + \{1-\lambda(m)-\mu(m)\} \cdot \frac{1}{2} \geq \frac{M}{M+1}$ for all m. Letting $m \to \infty$ completes the proof that $\frac{1}{2}(1-\lambda+\mu) \geq \frac{M}{M+1}$. □

Theorem 10.29

The Big Match has the following properties:

(1) There exists a value vector ϕ and $\phi = \big(\frac{1}{2}, 0, 1\big)^T$.

(2) Player 2 has an optimal stationary policy ρ^∞ with $\rho_{11} = \rho_{12} = \frac{1}{2}$.

(3) For any $\varepsilon > 0$ player 1 has a ε-optimal policy: R_1^M with $M = \frac{1}{2}\{\frac{1}{\varepsilon} - 2\}$.

(4) There is no optimal policy for plyer 1.

Proof

(1) We have shown above that this game has a value vector ϕ and that $\phi = \big(\frac{1}{2}, 0, 1\big)^T$.

(2) Take for player 2 the stationary policy with $\rho_{11} = \rho_{12} = \frac{1}{2}$. Then, in state 1 in each period player 1 earns $\frac{1}{2}$ independent of his strategy: $\phi_1(R_1, \rho^\infty) = \frac{1}{2} = \phi_1$ for all policies R_1, i.e. ρ^∞ is an optimal policy for player 2.

(3) We have shown (Lemma 10.18) that $\phi_1(R_1^M, R_2) \geq \frac{1}{2} \cdot \frac{M}{M+1}$ for all $M \in \mathbb{N}$ and all policies R_2. Hence, with $M = \frac{1}{2}\{\frac{1}{\varepsilon} - 2\}$, we obtain $\phi_1(R_1^M, R_2) \geq \frac{1}{2} - \varepsilon$, i.e. R_1^M with $M = \frac{1}{2}\{\frac{1}{\varepsilon} - 2\}$ is an optimal policy for player 1.

(4) Suppose that player 1 has an optimal policy, say $R_1^* = (\pi^1, \pi^2, \ldots)$, i.e. $\phi_1(R_1^*, R_2) \geq \frac{1}{2}$ for all R_2. The game is only interesting in state 1, i.e. as long as player 1 uses action 1.

We distinguish between two cases.

Case 1: $R_1^* = f_*^\infty$ with $f_*(1) = 1$

Take $R_2 = g^\infty$ with $g(1) = 2$. Then, $\phi_1(R_1^*, R_2) = 0 < \frac{1}{2} = \phi_1$: R_1^* is not optimal for player 1.

Case 2: $R_1^* \neq f_*^\infty$ with $f_*(1) = 1$

Suppose that $\pi^t_{h_t 2} = \varepsilon > 0$ for some t and some history h_t. Let t be the smallest time point for which this case holds and suppose that $b_1, b_2, \cdots, b_{t-1}$ is the sequence of actions for player 2 in h_t. Take $R_2 = (\rho^1, \rho^2, \ldots)$, where
$$\rho^n_{h_n b} := \begin{cases} 1 & 1 \leq n \leq t-1 \quad b = b_n \\ 0 & 1 \leq n \leq t-1 \quad b \neq b_n \\ 1 & \quad n = t \qquad\quad b = 1 \\ 0 & \quad n = t \qquad\quad b = 2 \\ \frac{1}{2} & \quad n \geq t \qquad\quad b = 1,2 \end{cases}$$

Then, $\phi_1(R_1^*, R_2) = \varepsilon \cdot 0 + (1-\varepsilon) \cdot \frac{1}{2} < \frac{1}{2} = \phi_1$: R_1^* is not optimal for player 1. □

The above lack of a solution in the space of stationary policies naturally gives reason for the following questions:
(1) Have stochastic games under the average reward criterion a value vector?
(2) Are there optimal (nonstationary) policies?
(3) For which subclasses do exist stationary optimal policies?

The answer to the first question remained open for over twenty years and was answered in the affirmative by Mertens and Neyman ([198]). This is a deep result, based on ingenious analysis in a series of three papers by Bewley and Kohlberg ([25],[26],[27]), who expressed the value vector of the discounted stochastic game in a Puiseux series, the so-called *limit discount equation.* We will not present the proof of the existence of the value vector in these lecture notes.

The Big Match shows that in general there are no optimal policies. The existence of ε-optimal policies follows from the existence of the value vector. Let ϕ be the value vector. Then, for any $\varepsilon > 0$ and any state i, we obtain from $sup_{R_1}\, inf_{R_2}\, \phi_i(R_1, R_2) = \phi_i$ that there exists a policy R_1^ε such that $inf_{R_2}\, \phi_i(R_1^\varepsilon, R_2) \geq \phi_i - \varepsilon$, implying $\phi_i(R_1^\varepsilon, R_2) \geq \phi_i - \varepsilon$ for all policies R_2 for player 2. Therefore, policy R_1^ε is a ε-optimal policy for player 1. Similarly it can be shown that player 2 has an ε-optimal policy for any $\varepsilon > 0$.

In view of the fact that in general undiscounted games need not possess optimal stationary policies, the algorithmic development for computing such policies centered around 'natural' classes that possess optimal stationary policies and on supplying algorithms for their computation. These classes of games can be roughly divided into two groups:
(1) Those that make assumptions on the ergodic properties of the underlying Markov chains.
(2) Those that make assumptions on the structure of the game data (transitions and/or rewards).

In the sequel we will encounter several special stochastic games which have stationary or even deterministic optimal policies.

10.4.3 Mathematical programming

Inspired by the concepts of super- and subharmonicity for both MDPs (cf. Theorem 5.17) and discounted stochastic games (cf. Theorem 10.9) we define for undiscounted stochastic games super- and subharmonicity as follows:

A vector $v \in \mathbb{R}^N$ is *superharmonic* if there exists a vector $t \in \mathbb{R}^N$ and a policy $\rho^\infty \in \Gamma$ such that the triple (v, t, ρ) satisfies the following system of inequalities

$$\left\{ \begin{array}{lcll} v_i & \geq & \sum_j p_{ij}(a,\rho)v_j & \text{for every } (i,a) \in S \times A; \\ v_i + t_i & \geq & r_i(a,\rho) + \sum_j p_{ij}(a,\rho)t_j & \text{for every } (i,a) \in S \times A. \end{array} \right. \tag{10.61}$$

A vector $v \in \mathbb{R}^N$ is *subharmonic* if there exists a vector $u \in \mathbb{R}^N$ and a policy $\pi^\infty \in \Pi$ such that the triple (v, u, π) satisfies the following system of inequalities

$$\left\{ \begin{array}{lcll} v_i & \leq & \sum_j p_{ij}(\pi,b)v_j & \text{for every } (i,b) \in S \times B; \\ v_i + u_i & \leq & r_i(\pi,b) + \sum_j p_{ij}(\pi,b)u_j & \text{for every } (i,b) \in S \times B. \end{array} \right. \tag{10.62}$$

Lemma 10.19

If (v, t, ρ) and (v, u, π) satisfy (10.61) and (10.62), then $v = P(\pi, \rho)v$ and $v+t = r(\pi, \rho)+P(\pi, \rho)t$.

Proof

The first part of relation (10.61) implies $v \geq r(\pi, \rho) + P(\pi, \rho)v$; similarly, the first of relation (10.62) implies $v \leq P(\pi, \rho)v$. Hence, $v = P(\pi, \rho)v$.
The second part of relation (10.61) implies $v + t \geq P(\pi, \rho)v$; similarly, the second part of relation (10.62) implies $v + t \leq P(\pi, \rho)v$; Hence, $v + t = P(\pi, \rho)v$. □

Theorem 10.30

An undiscounted stochastic game has stationary optimal policies $(\pi^*)^\infty$ and $(\rho^*)^\infty$ for player 1 and 2, respectively, if and only if (v, t, ρ^*) and (v, u, π^*) are feasible solutions of (10.61) and (10.62), respectively.

Proof

Assume that (v, t, ρ^*) and (v, u, π^*) are feasible solutions of (10.61) and (10.62). Then, for any $\pi^\infty \in \Pi$, $v \geq P(\pi, \rho^*)v$ and $v+t \geq r(\pi, \rho^*)+P(\pi, \rho^*)t$. The first inequality yields $v \geq P^*(\pi, \rho^*)v$, so we have $v \geq P^*(\pi, \rho^*)v \geq P^*(\pi, \rho^*)\{r(\pi, \rho^*) + P(\pi, \rho^*)t\} = P^*(\pi, \rho^*)r(\pi, \rho^*) = \phi\big(\pi^\infty, (\rho^*)^\infty\big)$. Hence,

$$v \geq \phi\big(\pi^\infty, (\rho^*)^\infty\big) \text{ for all } \pi^\infty \in \Pi. \tag{10.63}$$

Similarly, we derive

$$v \leq \phi\big((\pi^*)^\infty, \rho^\infty\big) \text{ for all } \rho^\infty \in \Gamma. \tag{10.64}$$

Therefore, $\phi\big(\pi^\infty, (\rho^*)^\infty\big) \leq v \leq \phi\big((\pi^*)^\infty, \rho^\infty\big)$ for all $\pi^\infty \in \Pi$ and $\rho^\infty \in \Gamma$, implying that the stochastic game has value vector $\phi = v$ and $(\pi^*)^\infty$ and $(\rho^*)^\infty$ are optimal stationary policies for player 1 and 2, respectively.
Now assume that $(\pi^*)^\infty$ and $(\rho^*)^\infty$ are stationary optimal policies for player 1 and 2, respectively. Then, $\phi\big((\pi^*)^\infty, R_2\big) \geq \inf_{R_2} \sup_{R_1} \phi(R_1, R_2) \geq \sup_{R_1} \inf_{R_2} \phi(R_1, R_2) \geq \phi\big(R_1, (\rho^*)^\infty\big)$ for all R_1 and R_2, implying $\phi\big((\pi^*)^\infty, R_2\big) \geq \phi = \phi\big((\pi^*)^\infty, (\rho^*)^\infty\big) \geq \phi\big(R_1, (\rho^*)^\infty\big)$ for all R_1 and R_2.
Hence, $(\pi^*)^\infty$ is an optimal policy in the MDP induced by the stationary policy $(\rho^*)^\infty$. Consequently ϕ is the smallest superharmonic vector in the sense of an undiscounted MDP problem, i.e. $(v = \phi, t, \rho = \rho^*)$ is a feasible solution of (10.61) for some t.
Similarly, it follows that $(\rho^*)^\infty$ is an optimal policy in the MDP induced by the stationary policy $(\pi^*)^\infty$ with respect to minimizing the average rewards. Therefore, ϕ is the largest subharmonic vector in the sense of an undiscounted MDP problem, i.e. $(v = \phi, u, \pi = \pi^*)$ is a feasible solution of (10.62) for some u. □

The systems (10.61) and (10.62) contain a mixture of linear and nonlinear terms. A method to solve these systems is to transform the systems to a nonlinear program. In the next corollary we have exhibit this idea. The nonlinear parts are moved to the objective function and the constraints are linear. We add some variables: $w_i(a), x_i(a)$ in (10.61) and $y_i(b), z_i(b)$ in (10.62) and obtain the following result.

Corollary 10.5

An undiscounted stochastic game has a value vector and optimal stationary policies if and only if the following nonlinear program has a global minimum value zero.

$$min \left\{ \sum_{(i,a)} \left\{ w_i(a) - \sum_j \sum_b p_{ij}(a,b)\rho_{ib}v_j \right\}^2 + \sum_{(i,a)} \left\{ x_i(a) - \sum_j \sum_b p_{ij}(a,b)\rho_{ib}t_j \right\}^2 + \sum_{(i,b)} \left\{ y_i(b) - \sum_i \sum_a p_{ij}(a,b)\pi_{ia}v_j \right\}^2 + \sum_{(i,b)} \left\{ z_i(b) - \sum_j \sum_b p_{ij}(a,b)\pi_{ia}u_j \right\}^2 \right\}$$

subject to

(1) $v_i - w_i(a) \geq 0,\ (i,a) \in S \times A;$

(2) $v_i + t_i - x_i(a) - \sum_b r_i(a,b)\rho_{ib} \geq 0,\ (i,a) \in S \times A;$

(3) $- v_i + y_i(b) \geq 0,\ (i,b) \in S \times B;$

(4) $- v_i - t_i + z_i(b) + \sum_a r_i(a,b)\pi_{ia} \geq 0,\ (i,b) \in S \times B;$

(5) $\pi_{ia} \geq 0,\ (i,a) \in S \times A;\ \sum_a \pi_{ia} = 1,\ i \in S;$

(6) $\rho_{ib} \geq 0,\ (i,b) \in S \times B;\ \sum_b \rho_{ib} = 1,\ i \in S.$

(7) $w_i(a),\ x_i(a) \geq 0,\ (i,a) \in S \times A;$

(8) $y_i(b),\ z_i(b) \geq 0,\ (i,b) \in S \times B.$

We can also formulate another, strongly related, nonlinear program in which the objective function is linear and the constraints are (partly) nonlinear. In this formulation we use different vectors for the superharmonicity (v^1, t^1) and the subharmonicity (v^2, t^2). It turns out that there are optimal stationary policies if and only if the program is feasible with optimum objective value 0.

Theorem 10.31

An undiscounted stochastic game has stationary optimal policies $(\pi^*)^\infty$ and $(\rho^*)^\infty$ for player 1 and 2, respectively, if and only if $(v^1, v^2, t^1, t^2, \pi = \pi^*, \rho = \rho^*)$ is an optimal solution of the nonlinear program

$$min \left\{ \sum_i (v_i^1 - v_i^2) \right\}$$

subject to

(1) $v_i^1 \geq \sum_j \sum_b p_{ij}(a,b)\rho_{ib}v_j^1,\ (i,a) \in S \times A;$

(2) $v_i^1 + t_i^1 \geq \sum_b r_i(a,b)\rho_{ib} + \sum_j \sum_b p_{ij}(a,b)\rho_{ib}t_j^1,\ (i,a) \in S \times A;$

(3) $- v_j^2 \geq - \sum_i \sum_a p_{ij}(a,b)\pi_{ia}v_j^2,\ (i,b) \in S \times B;$

(4) $- v_j^2 - t_j^2 \geq - \sum_a r_i(a,b)\pi_{ia} - \sum_i \sum_a p_{ij}(a,b)\pi_{ia}t_j^2,\ (i,b) \in S \times B;$

(5) $\pi_{ia} \geq 0,\ (i,a) \in S \times A;\ \sum_a \pi_{ia} = 1,\ i \in S;$

(6) $\rho_{ib} \geq 0,\ (i,b) \in S \times B;\ \sum_b \rho_{ib} = 1,\ i \in S.$

with optimum value 0.

Proof

Assume that $(v^1, v^2, t^1, t^2, \pi = \pi^*, \rho = \rho^*)$ is an optimal solution of the nonlinear program with optimum value 0. Then, (v^1, t^1, ρ^*) and (v^2, t^2, π^*) are feasible solutions of (10.61) and (10.62), respectively. Then, for any $\pi^\infty \in \Pi$, $v^1 \geq P(\pi, \rho^*)v^1$ and $v^1 + t^1 \geq r(\pi, \rho^*) + P(\pi, \rho^*)t^1$, which implies $v^1 \geq \phi\big(\pi^\infty, (\rho^*)^\infty\big)$ for all $\pi^\infty \in \Pi$. Similarly, we derive $v^2 \leq \phi\big((\pi^*)^\infty, \rho^\infty\big)$ for all $\rho^\infty \in \Gamma$. So, $v^1 \geq \phi\big((\pi^*)^\infty, (\rho^*)^\infty\big) \geq v^2$, i.e. $v^1 - v^2 \geq 0$. Since $\sum_i (v_i^1 - v_i^2) = 0$, we obtain $v^1 = v^2$. Furthermore, $\phi\big(\pi^\infty, (\rho^*)^\infty\big) \leq v^1 = v^2 \leq \phi\big((\pi^*)^\infty, \rho^\infty\big)$ for all $\pi^\infty \in \Pi$ and $\rho^\infty \in \Gamma$, implying that the stochastic game has value vector $\phi = v^1 = v^2$ and $(\pi^*)^\infty$ and $(\rho^*)^\infty$ are optimal stationary policies for player 1 and 2, respectively.

Now assume that $(\pi^*)^\infty$ and $(\rho^*)^\infty$ are optimal stationary policies for player 1 and 2. Then, $\phi\big((\pi^*)^\infty, R_2\big) \geq \inf_{R_2} \sup_{R_1} \phi(R_1, R_2) \geq \sup_{R_1} \inf_{R_2} \phi(R_1, R_2) \geq \phi\big(R_1, (\rho^*)^\infty\big)$ for all R_1 and R_2, implying $\phi\big((\pi^*)^\infty, R_2\big) \geq \phi = \phi\big((\pi^*)^\infty, (\rho^*)^\infty\big) \geq \phi\big(R_1, (\rho^*)^\infty\big)$ for all R_1 and R_2. Hence, $(\pi^*)^\infty$ is an optimal policy in the MDP induced by the stationary policy $(\rho^*)^\infty$. Consequently, ϕ is the smallest superharmonic vector in the sense of an undiscounted MDP problem. Therefore, $(v^1 = \phi, t^1, \rho = \rho^*)$ is a feasible solution of (10.61) for some t^1.

Similarly, it follows that $(\rho^*)^\infty$ is an optimal policy in the MDP induced by the stationary policy $(\pi^*)^\infty$ with respect to minimizing the average rewards. Therefore, ϕ is the largest subharmonic vector in the sense of an undiscounted MDP problem, i.e. $(v^2 = \phi, t^2, \pi = \pi^*)$ is a feasible solution of (10.62) for some t^2. Hence, $(v^1 = \phi, v^2 = \phi, t^1, t^2, \pi = \pi^*, \rho = \rho^*)$ is an optimal solution of the nonlinear program with optimum value 0. □

In the sequel of this section we shall show that the conditions of the characterization of stochastic games with optimal stationary optimal strategies as given in Theorem 10.3110.31 can be relaxed for games with a certain property which is called *uniform discount optimality.* From this relaxed set of conditions, one-step algorithms are developed for *ARAT* games and for switching-controller stochastic games. Each algorithm requires the solution of a single bilinear program.

A stochastic game is said to possess *uniformly discount optimal stationary policies* if a pair of stationary policies, optimal in the undiscounted game, is also optimal in corresponding discounted game for all discount factors a close enough to 1.

A pair of optimal stationary policies $(\pi^*)^\infty$ and $(\rho^*)^\infty$ for player 1 and 2, respectively, is *asymptotically stable* if there exists an $\alpha_0 \in (0, 1)$ and stationary policies $(\pi^\alpha)^\infty$ and $(\rho^\alpha)^\infty$ for player 1 and 2, respectively, such that for each $\alpha \in (\alpha_0, 1)$:

(1) $(\pi^\alpha)^\infty$ and $(\rho^\alpha)^\infty$ is an optimal pair for the α-discounted game;

(2) $\lim_{\alpha \uparrow 1} \pi^\alpha = \pi^*$ and $\lim_{\alpha \uparrow 1} \rho^\alpha = \rho^*$;

(3) $r(\pi^\alpha, \rho^\alpha) = r(\pi^*, \rho^*)$; $P(\pi^\alpha, \rho) = P(\pi^*, \rho)$ for all stationary policies ρ^∞ for player 2;
$P(\pi, \rho^\alpha) = P(\pi, \rho^*)$ for all stationary policies π^∞ for player 1.

Notice that a pair $\big((\pi^*)^\infty, (\rho^*)^\infty\big)$ uniformly discount optimal stationary policies is asymptotically stable: set $\pi^\alpha := \pi^*$ and $\rho^\alpha := \rho^*$ for all $\alpha \in (\alpha_0, 1)$. The problem of determining whether an undiscounted stochastic game has optimal stationary policies is equivalent to the problem of solving a bilinear feasibility problem (see Theorem 10.30). The next theorem shows that this bilinear system can be simplified if the stochastic game possesses asymptotically stable optimal stationary policies.

Theorem 10.32

If a stochastic game possesses asymptotically stable optimal policies for player 1 and 2, respectively, then the bilinear system$(\pi^*)^\infty$ *and* $(\rho^*)^\infty$

$$\begin{cases} v_i & \geq & \sum_j p_{ij}(a,\rho)v_j & \textit{for every } (i,a) \in S \times A \\ v_i + t_i & \geq & r_i(a,\rho) + \sum_j p_{ij}(a,\rho)t_j & \textit{for every } (i,a) \in S \times A \\ v_i & \leq & \sum_j p_{ij}(\pi,b)v_j & \textit{for every } (i,b) \in S \times B \\ v_i + t_i & \leq & r_i(\pi,b) + \sum_j p_{ij}(\pi,b)t_j & \textit{for every } (i,b) \in S \times B \end{cases} \tag{10.65}$$

has a feasible solution $(v,t,\rho = \rho^*, \pi = \pi^*)$. .

Proof

Let $\big((\pi^*)^\infty, (\rho^*)^\infty\big)$ be a pair of asymptotically stable optimal policies with corresponding $(\pi^\alpha, \rho^\alpha)$. Set $t^\alpha := \sum_{t=0}^\infty \alpha^t \cdot \{P^t(\pi^\alpha,\rho^\alpha) - P^*(\pi^\alpha,\rho^\alpha)\}r(\pi^\alpha,\rho^\alpha)$ for all $\alpha \in (\alpha_0, 1)$, $v := \phi\big((\pi^*)^\infty, (\rho^*)^\infty\big)$ and $t^1 := D(\pi^*,\rho^*)r(\pi^*,\rho^*)$. Notice that

$$t^\alpha = v^\alpha\big((\pi^\alpha,\rho^\alpha)\big) - \frac{1}{1-\alpha} \cdot \phi\big((\pi^\alpha,\rho^\alpha)\big). \tag{10.66}$$

Furthermore, by Theorem 5.7 part (2), $t^1 = \lim_{\alpha \uparrow 1} t^\alpha$. Define the following numbers:

$C(i,a) := v_i - \sum_j p_{ij}(a,\rho^*)v_j,\ (i,a) \in S \times A;$

$D(i,a) := v_i + t^1 - r_i(a,\rho^*) - \sum_j p_{ij}(a,\rho^*)t^1_j,\ (i,a) \in S \times A;$

$G(i,b) := v_i - \sum_j p_{ij}(\pi^*,b)v_j,\ (i,b) \in S \times B;$

$H(i,b) := v_i + t^1 - r_i(\pi^*,b) - \sum_j p_{ij}(\pi^*,b)t^1_j,\ (i,b) \in S \times B.$

We have to show: $C(i,a) \geq 0,\ (i,a) \in S \times A$, $D(i,a) \geq 0,\ (i,a) \in S \times A$, $G(i,b) \leq 0,\ (i,b) \in S \times B$ and $H(i,b) \leq 0,\ (i,b) \in S \times B$.

Since $(\pi^*)^\infty$ and $(\rho^*)^\infty$ are stationary optimal policies for player 1 and 2, respectively, it follows from the proof of Theorem 10.31 that $(\pi^*)^\infty$ is an optimal policy in the MDP induced by $(\rho^*)^\infty$, and $(\rho^*)^\infty$ is an optimal policy in the MDP induced by $(\pi^*)^\infty$. Therefore, $v := \phi\big((\pi^*)^\infty, (\rho^*)^\infty\big)$ is simultaneously the value vector of the MDPs induced by ρ^* and π^*, respectively. From Chapter 5 it follows that $C(i,a) \geq 0$ for all $(i,a) \in S \times A$ and, similarly, $G(i,b) \leq 0$ for all $(i,b) \in S \times B$, and furthermore,

$$min_{a \in A(i)}\, C(i,a) = 0\ ,\ i \in S \text{ and } max_{b \in B(i)}\, G(i,b) = 0\ ,\ i \in S. \tag{10.67}$$

Let $\overline{A}(i) := \{a \in A(i) \mid C(i,a) = 0\}$ and $\overline{B}(i) := \{b \in B(i) \mid G(i,b) = 0\}$. We shall show that $D(i,a) \geq= 0,\ (i,a) \in S \times A$. Then, similarly, it can be shown that $H(i,b) = 0,\ (i,b) \in S \times B$. Since $\big((\pi^\alpha)^\infty, (\rho^\alpha)^\infty\big)$ is an optimal pair for the α-discounted game, $v^\alpha\big((\pi^\alpha)^\infty, (\rho^\alpha)^\infty\big)$ is the value vector of the MDP induced by ρ^α. Hence, we can write

$$v^\alpha_i\big((\pi^\alpha)^\infty, (\rho^\alpha)^\infty\big) \geq r_i(a,\rho^\alpha) + \alpha \sum_j p_{ij}(a,\rho^\alpha)v^\alpha_j\big((\pi^\alpha)^\infty, (\rho^\alpha)^\infty\big),\ (i,a) \in S \times A,\ \alpha \in (\alpha_0, 1). \tag{10.68}$$

Since for all $\alpha > \alpha_0$, $r(\pi^\alpha,\rho^\alpha) = r(\pi^*,\rho^*)$ and also $P(\pi^\alpha,\rho^\alpha) = P(\pi^*,\rho^*)$, we have for all $\alpha > \alpha_0$, $P^*(\pi^\alpha,\rho^\alpha) = P^*(\pi^*,\rho^*)$ and $\phi\big((\pi^\alpha)^\infty,(\rho^\alpha)^\infty\big) = \phi\big((\pi^*)^\infty,(\rho^*)^\infty\big)$ for all $\alpha > \alpha_0$. Hence, by (10.66) and (10.68), we obtain for all $(i,a) \in S \times A$ and all $\alpha \in (\alpha_0,1)$,

$$\tfrac{1}{1-\alpha}\cdot\phi_i\big((\pi^*)^\infty,(\rho^*)^\infty\big) + t_i^\alpha \ge r_i(a,\rho^\alpha) + \alpha\textstyle\sum_j p_{ij}(a,\rho^\alpha)\cdot\{\tfrac{1}{1-\alpha}\cdot\phi_j\big((\pi^*)^\infty,(\rho^*)^\infty\big) + t_j^\alpha\}.$$

Since $v = \phi\big((\pi^*)^\infty,(\rho^*)^\infty\big)$ and $\frac{1}{1-\alpha} = 1 + \frac{\alpha}{1-\alpha}$, we obtain

$$v_i + t_i^\alpha \ge r_i(a,\rho^\alpha) + \alpha\textstyle\sum_j p_{ij}(a,\rho^\alpha)t_j^\alpha + s_i^\alpha(a),\ (i,a) \in S \times A,\ \alpha \in (\alpha_0,1),$$

where $s_i^\alpha(a) := \frac{\alpha}{1-\alpha}\{\sum_j p_{ij}(a,\rho^\alpha)v_j - v_i\} = 0$ for all $(i,a) \in S \times \overline{A}$ and all $\alpha \in (\alpha_0,1)$.

Therefore, letting α increase to 1 and since $t^1 = \lim_{\alpha\uparrow 1} t^\alpha$, we have

$$v_i + t_i^1 \ge r_i(a,\rho^*) + \textstyle\sum_j p_{ij}(a,\rho^*)t_j^1,\ (i,a) \in S \times \overline{A}, \text{ i.e. } D(i,a) \ge 0,\ (i,a) \in S \times \overline{A}.$$

A vector t can be defined such that $(v,t,\rho=\rho^*,\pi=\pi^*)$ is a feasible solution of the second and fourth sets of inequalities of (10.65). We present the validation of the second set. The validation of the fourth set can be done similarly.

To this end, let $A^*(i) := \{a \in A(i) \mid D(i,a) < 0\}$ and $B^*(i) := \{b \in B(i) \mid H(i,b) > 0\}$. We have seen that $A^*(i) \cap \overline{A}(i) = \emptyset$ and $B^*(i) \cap \overline{B}(i) = \emptyset$. Define M by

$$M := min\Big\{min\{\tfrac{D(i,a)}{C(i,a)} \mid (i,a) \in S \times A^*\},\ min\{\tfrac{H(i,b)}{G(i,b)} \mid (i,b) \in S \times B^*\}\Big\},$$

where a minimum over the empty set will be taken to be zero. Note that $M \le 0$.

Define t by $t := t^1 - M \cdot v$. The proof that $(v,t,\rho=\rho^*,\pi=\pi^*)$ is a feasible solution of the second set of inequalities of (10.65) is strongly related to the proof of Theorem 5.17. We distinguish between the cases (i) $a \in \overline{A}(i)$, (ii) $a \in A^*(i)$ and (iii) $a \notin \overline{A}(i) \cup A^*(i)$.

Case (i): $a \in \overline{A}(i)$, i.e. $v_i = \sum_j p_{ij}(a,\rho^*)v_j$.

$$v_i + t_i = v_i + t_i^1 - M\cdot v_i \ge r_i(a,\rho^*) + \textstyle\sum_j p_{ij}(a,\rho^*)(t_j^1 - M\cdot v_j) = r_i(a,\rho^*) + \textstyle\sum_j p_{ij}(a,\rho^*)t_j.$$

Case (ii): $a \in A^*(i)$, i.e. $v_i + t_i^1 < r_i(a,\rho^*) + \sum_j p_{ij}(a,\rho^*)t_j^1$ and $v_i > \sum_j p_{ij}(a,\rho^*)v_j$.

$$\begin{aligned}
v_i + t_i &= v_i + t_i^1 - M\cdot v_i = v_i + t_i^1 - M\cdot\{C(i,a) + \textstyle\sum_j p_{ij}(a,\rho^*)v_j\}\\
&= D(i,a) + r_i(a,\rho^*) + \textstyle\sum_j p_{ij}(a,\rho^*)t_j^1 - M\cdot C(i,a) - M\cdot\textstyle\sum_j p_{ij}(a,\rho^*)v_j\\
&= r_i(a,\rho^*) + \textstyle\sum_j p_{ij}(a,\rho^*)(t_j^1 - M\cdot v_j) + D(i,a) - M\cdot C(i,a)\\
&\ge r_i(a,\rho^*) + \textstyle\sum_j p_{ij}(a,\rho^*)t_j.
\end{aligned}$$

Case (iii): $a \notin \overline{A}(i) \cup A^*(i)$, i.e. $v_i + t_i^1 \ge r_i(a,\rho^*) + \sum_j p_{ij}(a,\rho^*)t_j^1$ and $v_i > \sum_j p_{ij}(a,\rho^*)v_j$.

$$\begin{aligned}
v_i + t_i &= v_i + t_i^1 - M\cdot v_i = v_i + t_i^1 - M\cdot\{C(i,a) + \textstyle\sum_j p_{ij}(a,\rho^*)v_j\}\\
&= D(i,a) + r_i(a,\rho^*) + \textstyle\sum_j p_{ij}(a,\rho^*)t_j^1 - M\cdot C(i,a) - M\cdot\textstyle\sum_j p_{ij}(a,\rho^*)v_j\\
&\ge r_i(a,\rho^*) + \textstyle\sum_j p_{ij}(a,\rho^*)(t_j^1 - M\cdot v_j) = r_i(a,\rho^*) + \textstyle\sum_j p_{ij}(a,\rho^*)t_j.
\end{aligned}$$

ARAT stochastic games

An additive reward and additive transition (*ARAT*) stochastic game is defined by the property that the rewards as well as the transitions can be written as the sum of a term determined by player 1 and a term determined by player 2: $r_i(a,b) = r_i^1(a) + r_i^2(b),\ i \in S,\ a \in A(i),\ b \in B(i)$ and $p_{ij}(a,b) = p_{ij}^1(a) + p_{ij}^2(b),\ i,j \in S,\ a \in A(i),\ b \in B(i)$.

Theorem 10.33

(1) Both players possess uniform discount optimal deterministic policies.

(2) Uniform discount optimal deterministic policies are optimal for the average reward criterion as well.

(3) The ordered field property holds for the the average reward criterion.

Proof

(1) From Theorem 10.17, part (1), we know that players have discounted optimal deterministic policies for any discount factor $\alpha \in (0,1)$. Because there is only a finite number of deterministic policies it can be shown, similar as in the proof of Theorem 5.9, that this give rise to the existence of uniform discount optimal deterministic policies for both players.

(2) Let f_*^∞ and g_*^∞ be uniform discount optimal deterministic policies, i.e. for all stationary policies π^∞ and ρ^∞ we have, $v^\alpha(\pi^\infty, g_*^\infty) \leq v^\alpha(f_*^\infty, g_*^\infty) \leq v^\alpha(f_*^\infty, g_*^\infty)$ for all $\alpha \in (\alpha_0, 1)$ for some $\alpha_0 \in (0,1)$. By taking $\lim_{\alpha \uparrow 1}$, we obtain for all stationary policies π^∞ and ρ^∞, $\phi(\pi^\infty, g_*^\infty) \leq \phi(f_*^\infty, g_*^\infty) \leq \phi(f_*^\infty, g_*^\infty)$. This implies that f_*^∞ and g_*^∞ are optimal deterministic policies for the average reward criterion.

(3) The value vector ϕ satisfies $\phi = \phi(f_*^\infty, g_*^\infty) = P^*(f_*, g_*)r(f_*, g_*)$, where f_*^∞ and g_*^∞ are optimal deterministic policies for the average reward criterion. Since $P^*(f_*, g_*)r(f_*, g_*)$ can be computed by solving some systems of linear equations (see Algorithm 5.5), the ordered field property holds for the the average reward criterion. □

Remark

Since *ARAT* games possess uniform discount optimal stationary policies such games also possess asymptotically stable stationary optimal policies.

Define the bilinear function $\psi(g,h,u,w,\pi,\rho)$ by

$$\psi(g,h,u,w,\pi,\rho) := \sum_i (u_i - g_i) - \sum_i \{\sum_a r_i^1(a)\pi_{ia} + \sum_j \sum_a (h_j + w_j) p_{ij}^1(a)\pi_{ia} - w_i\}$$
$$+ \sum_i \{\sum_b r_i^2(b)\rho_{ib} + \sum_j \sum_b (h_j + w_j) p_{ij}^2(b)\rho_{ib} - h_i\}.$$

Furthermore, we define the following bilinear program, which is called *BLP*:

$min\, \psi(g,h,u,w,\pi,\rho)$ subject to the following constraints

(1) $u_i - \sum_j p_{ij}^1(a)(g_j + u_j) \geq 0,\ (i,a) \in S \times A.$

(2) $w_i - \sum_j p_{ij}^1(a)(h_j + w_j - g_j - u_j) \geq r_i^1(a),\ (i,a) \in S \times A.$

(3) $g_i - \sum_j p_{ij}^2(b)(g_j + u_j) \leq 0,\ (i,b) \in S \times B.$

(4) $h_i - \sum_j p_{ij}^2(b)(h_j + w_j - g_j - u_j) \leq r_i^2(b),\ (i,b) \in S \times B.$

(5) $\sum_a \pi_{ia} = 1,\ i \in S.$

(6) $\sum_b \rho_{ib} = 1,\ i \in S.$

(7) $\pi_{ia} \geq 0,\ (i,a) \in S \times A.$

(8) $\rho_{ib} \geq 0,\ (i,b) \in S \times B.$

Theorem 10.34

(1) *From any pair of asymptotically stable optimal stationary policies an optimal solution of the BLP can be derived, which has optimum value zero.*

(2) *The value and optimal stationary policies of the ARAT game can be derived from any optimal solution of BLP .*

Proof

(1) We have already observed that $ARAT$ stochastic games possess asymptotically stable optimal stationary policies, say $(\pi^*)^\infty$ and $(\rho^*)^\infty$. Therefore, by Theorem 10.32, the bilinear system (10.65), which we denote by BLS, has a feasible solution $(v^*, t^*, \rho^*, \pi^*)$. Hence, we obtain

$$\begin{array}{llll}
v_i^* - \sum_j p_{ij}^1(a)v_j^* - \sum_j p_{ij}^2(\rho^*)v_j^* & \geq & 0 & \text{for every } (i,a) \in S \times A \\
v_i^* + t_i^* - \sum_j p_{ij}^1(a)t_j^* - \sum_j p_{ij}^2(\rho^*)t_j^* & \geq & r_i^1(a) + r_i^2(\rho^*) & \text{for every } (i,a) \in S \times A \\
v_i^* - \sum_j p_{ij}^1(\pi^*)v_j^* - \sum_j p_{ij}^2(b)v_j^* & \leq & 0 & \text{for every } (i,b) \in S \times B \\
v_i^* + t_i^* - \sum_j p_{ij}^1(\pi^*)t_j^* - \sum_j p_{ij}^2(b)t_j^* & \leq & r_i^1(\pi^*) + r_i^2(b) & \text{for every } (i,b) \in S \times B
\end{array}$$

Then, we obtain from Lemma 10.19

$v^* = P^1(\pi^*)v^* + P^2(\rho^*)v^*$ and $v^* + t^* = r^1(\pi^*) + r^2(\rho^*) + P^1(\pi^*)t^* + P^2(\rho^*)t^*$.

Define g^*, h^*, u^* and w^* by:

$g^* := P^2(\rho^*)v^*,\ h^* := r^2(\rho^*) + P^2(\rho^*)t^*,\ u^* := P^1(\pi^*)v^*$ and $w^* := r^1(\pi^*) + P^1(\pi^*)t^*$.

Then, we have $v^* = u^* + g^*$ and $v^* + t^* = w^* + h^*$. Furthermore, by the above inequalities,

$$\begin{array}{lll}
u_i^* - \sum_j p_{ij}^1(a)(g_j^* + u_j^*) & = & u_i^* + g_i^* - \sum_j p_{ij}^1(a)v_j^* - g_i^* \\
& = & v_i^* - \sum_j p_{ij}^1(a)v_j^* - \sum_j p_{ij}^2(\rho^*)v_j^* \\
& \geq & 0,\ (i,a) \in S \times A.
\end{array}$$

$$\begin{array}{lll}
w_i^* - \sum_j p_{ij}^1(a)(h_j^* + w_j^* - g_j^* - u_j^*) & = & w_i^* + h_i^* - \sum_j p_{ij}^1(a)t_j^* - h_i^* \\
& = & v_i^* + t_i^* - \sum_j p_{ij}^1(a)t_j^* - r_i^2(\rho^*) - \sum_j p_{ij}^2(\rho^*)t_j^* \\
& \geq & r_i^1(a),\ (i,a) \in S \times A.
\end{array}$$

$$\begin{array}{lll}
g_i^* - \sum_j p_{ij}^2(b)(g_j^* + u_j^*) & = & u_i^* + g_i^* - \sum_j p_{ij}^2(b)v_j^* - u_i^* \\
& = & v_i^* - \sum_j p_{ij}^2(b)v_j^* - \sum_j p_{ij}^1(\pi^*)v_j^* \\
& \leq & 0,\ (i,b) \in S \times B.
\end{array}$$

$$\begin{array}{lll}
h_i^* - \sum_j p_{ij}^2(b)(h_j^* + w_j^* - g_j^* - u_j^*) & = & w_i^* + h_i^* - \sum_j p_{ij}^2(b)t_j^* - w_i^* \\
& = & v_i^* + t_i^* - \sum_j p_{ij}^2(b)t_j^* - r_i^1(\pi^*) - \sum_j p_{ij}^1(\pi^*)t_j^* \\
& \leq & r_i^2(b),\ (i,b) \in S \times B.
\end{array}$$

Hence, $z^* := (g^*, h^*, u^*, w^*, \pi^*, \rho^*)$ is a feasible solution of BLP. For the value $\psi(z^*)$ we obtain

$$\begin{array}{lll}
\psi(z^*) & = & \sum_i (u_i^* - g_i^*) - \sum_i \{\sum_a r_i^1(a)\pi_{ia}^* + \sum_j \sum_a (h_j^* + w_j^*)p_{ij}^1(a)\pi_{ia}^* - w_i^*\} \\
& & \qquad + \sum_i \{\sum_b r_i^2(b)\rho_{ib}^* + \sum_j \sum_b (h_j^* + w_j^*)p_{ij}^2(b)\rho_{ib}^* - h_i^*\} \\
& = & \sum_i (u_i^* - g_i^*) - \sum_i \{r_i^1(\pi^*) + \sum_j (h_j^* + w_j^*)p_{ij}^1(\pi^*) - w_i^*\} \\
& & \qquad + \sum_i \{r_i^2(\rho^*) + \sum_j (h_j^* + w_j^*)p_{ij}^2(\rho^*) - h_i^*\}
\end{array}$$

$$\begin{aligned}
&= \textstyle\sum_i \{u_i^* - r_i^1(\pi^*) - \sum_j (h_j^* + w_j^*)p_{ij}^1(\pi^*) + w_i^*\} \\
&\qquad\qquad\qquad - \textstyle\sum_i \{g_i^* - r_i^2(\rho^*) - \sum_j (h_j^* + w_j^*)p_{ij}^2(\rho^*) + h_i^*\} \\
&= \textstyle\sum_i \{u_i^* - r_i^1(\pi^*) - \sum_j (v_j^* + t_j^*)p_{ij}^1(\pi^*) + w_i^*\} \\
&\qquad\qquad\qquad - \textstyle\sum_i \{g_i^* - r_i^2(\rho^*) - \sum_j (v_j^* + t_j^*)p_{ij}^2(\rho^*) + h_i^*\} \\
&= \textstyle\sum_i \{u_i^* - \sum_j p_{ij}^1(\pi^*)v_j^* - r_i^1(\pi^*) - \sum_j p_{ij}^1(\pi^*)t_j^* + w_i^*\} \\
&\qquad - \textstyle\sum_i \{g_i^* - r_i^2(\rho^*) - \sum_j p_{ij}^2(\rho^*)v_j^* - r_i^2(\rho^*) - \sum_j p_{ij}^2(\rho^*)t_j^* + h_i^*\} \\
&= \textstyle\sum_i \{u_i^* - \sum_j p_{ij}^1(\pi^*)v_j^*\} - \sum_i \{g_i^* - r_i^2(\rho^*) - \sum_j p_{ij}^2(\rho^*)v_j^*\} \\
&= 0.
\end{aligned}$$

Now, we shall show that the objective function is at least zero. Let $z = (g, h, u, w, \pi, \rho)$ be a feasible solution of BLP. Adding (1) and (2) gives $u_i + w_i - \sum_j p_{ij}^1(a)(h_j + w_j) \geq r_i^1(a)$ for every $(i, a) \in S \times A$. This implies $\sum_i u_i + \sum_i w_i - \sum_i r_i^1(\pi) - \sum_i p_{ij}^1(\pi)(h_j + w_j) \geq 0$. Similarly, using (3) and (4), we obtain $\sum_i g_i + \sum_i h_i - \sum_i r_i^2(\rho) - \sum_i p_{ij}^2(\rho)(h_j + w_j) \leq 0$. Therefore, $\psi(g, h, u, w, \pi, \rho) \geq 0$.

(2) Let (g, h, u, w, π, ρ) be a feasible solution of BLP. Define $v_i := g_i + u_i,\ t_i := h_i + w_i - g_i - u_i$ for every $i \in S$. Similarly as for $(g^*, h^*, u^*, w^*, \pi^*, \rho^*)$ in part (1), we can derive

$$\begin{aligned}
v_i - \textstyle\sum_j p_{ij}(a, \rho)v_j &= v_i - \textstyle\sum_j p_{ij}^1(a)v_j - \sum_j p_{ij}^2(\rho)v_j \\
&= u_i - \textstyle\sum_j p_{ij}^1(a)(g_j + u_j) \geq 0,\ (i, a) \in S \times A. \\
v_i + t_i - \textstyle\sum_j p_{ij}(a, \rho)t_j &= v_i + t_i - \textstyle\sum_j p_{ij}^1(a)t_j - \sum_j p_{ij}^2(\rho)t_j \\
&= w_i + - \textstyle\sum_j p_{ij}^1(a)(h_j + w_j - g_j - u_j) + r_i^2(\rho) \\
&\geq r_i^1(a) + r_i^2(\rho) = r_i(a, \rho),\ (i, a) \in S \times A. \\
v_i - \textstyle\sum_j p_{ij}(\pi, b)v_j &= v_i - \textstyle\sum_j p_{ij}^1(\pi)v_j - \sum_j p_{ij}^2(b)v_j \\
&= g_i - \textstyle\sum_j p_{ij}^2(b)(g_j + u_j) \leq 0,\ (i, b) \in S \times B. \\
v_i + t_i - \textstyle\sum_j p_{ij}(\pi, b)t_j &= v_i + t_i - \textstyle\sum_j p_{ij}^1(\pi)t_j - \sum_j p_{ij}^2(b)t_j \\
&= h_i - \textstyle\sum_j p_{ij}^2(b)(h_j + w_j - g_j - u_j) + r_i^1(\pi) \\
&\leq r_i^1(\pi) + r_i^2(b) = r_i(\pi, b),\ (i, b) \in S \times B.
\end{aligned}$$

Hence, (u, t, π, ρ) are feasible solutions of (10.61) and (10.62), respectively. By Theorem 10.30, π^∞ and ρ^∞ are stationary optimal policies player 1 and 2, respectively. □

In Theorem 10.33 the existence of deterministic optimal policies is shown. The bilinear program BLP provides, by Theorem 10.34, optimal stationary policies. The next theorem shows that optimal deterministic policies can be derived from any optimal solution of BLP by playing with probability 1 any action which has a positive probability in the optimal solution of the BLP.

Theorem 10.35

(1) Let π^* be part of the optimal solution of the bilinear program BLP. Then, any deterministic policy f_*^∞ with $f_*(i)$ such that $\pi_{if_*(i)}^* > 0$ for all $i \in S$ is optimal for player 1.

(2) Let ρ^* be part of the optimal solution of the bilinear program BLP. Then, any deterministic policy g_*^∞ with $g_*(i)$ such that $\rho_{ig_*(i)}^* > 0$ for all $i \in S$ is optimal for player 2.

Proof

(1) Let $z^* = (g^*, h^*, u^*, w^*, \pi^*, \rho^*)$ be an optimal solution of BLP. Then, we can write, using $v^* := g^* + u^*$ and $t^* := h^* + w^* - g^* - u^* = h^* + w^* - v^*$,

$$\begin{array}{rcl} \psi(z^*) & = & \sum_i (u_i^* - g_i^*) - \sum_i \{\sum_a r_i^1(a)\pi_{ia}^* + \sum_j \sum_a (h_j^* + w_j^*)p_{ij}^1(a)\pi_{ia}^* - w_i^*\} \\ & & \hfill + \sum_i \{\sum_b r_i^2(b)\rho_{ib}^* + \sum_j \sum_b (h_j^* + w_j^*)p_{ij}^2(b)\rho_{ib}^* - h_i^*\} \\ & = & \sum_i (u_i^* - g_i^*) - \sum_i \{r_i^1(\pi^*) + \sum_j (h_j^* + w_j^*)p_{ij}^1(\pi^*) - w_i^*\} \\ & & \hfill + \sum_i \{r_i^2(\rho^*) + \sum_j (h_j^* + w_j^*)p_{ij}^2(\rho^*) - h_i^*\} \\ & = & \sum_i \{u_i^* - r_i^1(\pi^*) - \sum_j p_{ij}^1(\pi^*)(v_j^* + t_j^*) + w_i^*\} \\ & & \hfill + \sum_i \{-g_i^* + r_i^2(\rho^*) + \sum_j p_{ij}^2(\rho^*)(v_j^* + t_j^*) - h_i^*\} \\ & = & \sum_i \{u_i^* - \sum_j p_{ij}^1(\pi^*)v_j^*\} + \sum_i \{w_i^* - \sum_j p_{ij}^1(\pi^*)t_j^* - r_i^1(\pi^*)\} \\ & & \hfill - \sum_i \{g_i^* - \sum_j p_{ij}^2(\rho^*)v_j^*\} - \sum_i \{h_i^* - \sum_j p_{ij}^2(\rho^*)t_j^*) - r_i^2(\rho^*)\}. \end{array}$$

Since all terms of (1) and (2) in BLP are nonnegative, and all terms of (3) and (4) are nonpositive, $\psi(z^*) = 0$ implies that the inequalities of (1), (2), (3) and (4) of BLP corresponding with $\pi_{ia}^* > 0$ and $\rho_{ib}^* > 0$ are equalities. Note that the constraints (1), (2), (3) and (4) do not depend on the variables π_{ia} and ρ_{ib}. Let $z_1^* := (g^*, h^*, u^*, w^*, f_*, \rho^*)$, where f_* is such that $\pi_{if_*(i)}^* > 0$ for all $i \in S$. Then, z_1^* is also feasible and is also an optimal solution of BLP. Hence, f_*^∞ is optimal for player 1.

(2) The proof is similar to part (1) of this theorem. □

Remark 1

The bilinear program BLP is of the general form $min\{a^Tx + x^TBy + c^Ty \mid Dx \geq f;\ Gy \geq h\}$, where $x \in R^n$ and $y \in R^m$ are the variables and a, c, f, h are appropriate sized vectors, and B, D, G are appropriate sized matrices. If a solution exists, which always does for our BLP, then there must be a solution (x^*, y^*) such that x^* is an extreme point of the polyhedron $X := \{x \mid Dx \geq f\}$ and y^* is an extreme point of the polyhedron $Y := \{y \mid Gx \geq h\}$. Sherali and Shetty ([270]) and Gallo and Ulkucu ([101]) developed finite algorithms for solving such bilinear programs.

Remark 2

An analogous treatment of the $ARAT$ model with the discounted reward criterion is also possible.

10.4.4 Perfect information and irreducible games

Perfect information

We have seen in Corollary 10.2 that a discounted stochastic game with perfect information has optimal deterministic policies. For undiscounted stochastic games we have the same result, but the proof is more complicated.

Theorem 10.36

In an undiscounted stochastic game with perfect information, both players possess optimal deterministic policies.

Proof

For any $\alpha \in (0,1)$ there exists deterministic stationary policies f_α^∞ and g_α^∞ such that

$$v^\alpha(f^\infty, g_\alpha^\infty) \leq v^\alpha(f_\alpha^\infty, g_\alpha^\infty) \leq v^\alpha(f_\alpha^\infty, g^\infty) \text{ for all } f^\infty \in F \text{ and } g^\infty \in G,$$

where F and G are the sets of deterministic stationary policies for player 1 and 2, respectively. Since $F \times G$ is a finite set, we can therefore find a pair $f_*^\infty \in F$ and $g_*^\infty \in G$ and a sequence $\{\alpha_n\}_{n=1}^\infty$ such that

$$v_n^\alpha(f^\infty, g_*^\infty) \leq v_n^\alpha(f_*^\infty, g_*^\infty) \leq v_n^\alpha(f_*^\infty, g^\infty) \text{ for all } f^\infty \in F,\ g^\infty \in G \text{ and } n = 1, 2, \ldots,$$

For any $f^\infty \in F,\ g^\infty \in G$, the vector $v^\alpha(f^\infty, g^\infty)$ is the unique solution of the linear system $x = r(f,g) + \alpha P(f,g)x$. Since this linear system can be solved by Cramer's rule, the numbers $v_i^\alpha(f^\infty, g^\infty),\ i \in S$, are rational functions in α. Therefore, also - for all $i \in S$ - the functions $h_i^1(\alpha) = v_i^\alpha(f^\infty, g_*^\infty) - v_i^\alpha(f_*^\infty, g_*^\infty)$ and $h_i^2(\alpha) = v_i^\alpha(f_*^\infty, g^\infty) - v_i^\alpha(f_*^\infty, g_*^\infty)$ are rational in α for all $f^\infty \in F,\ g^\infty \in G$ and $i \in S$. Hence, for $k = 1,2$ and all $i \in S$, either $h_i^k(\alpha) \equiv 0$ or $h_i^k(\alpha)$ has a finite number of zero's in $(0,1)$. In the last case let $\alpha_* \in (0,1)$ be the largest zero of the finite number of functions $h_i^k(\alpha),\ k = 1,2,\ i \in S$. With this α_*, we have for all $\alpha \geq \alpha_*$:

$$v^\alpha(f^\infty, g_*^\infty) \leq v^\alpha(f_*^\infty, g_*^\infty) \leq v^\alpha(f_*^\infty, g^\infty) \text{ for all } f^\infty \in F,\ g^\infty \in G \text{ and all } \alpha \geq \alpha_*,$$

and consequently

$$(1-\alpha)v^\alpha(f^\infty, g_*^\infty) \leq (1-\alpha)v^\alpha(f_*^\infty, g_*^\infty) \leq (1-\alpha)v^\alpha(f_*^\infty, g^\infty) \text{ for all } f^\infty \in F,\ g^\infty \in G,\ \alpha \geq \alpha_*.$$

Then, using the Laurent series expansion, which implies $\lim_{\alpha\to\infty}(1-\alpha)v^\alpha(f^\infty, g^\infty) = \phi(f^\infty, g^\infty)$ for all $f^\infty \in F,\ g^\infty \in G$, we obtain

$$\phi(f^\infty, g_*^\infty) \leq \phi(f_*^\infty, g_*^\infty) \leq \phi(f_*^\infty, g^\infty) \text{ for all } f^\infty \in F \text{ and } g^\infty \in G,$$

also implying (by MDP) $\phi(R_1, g_*^\infty) \leq \phi(f_*^\infty, g_*^\infty) \leq \phi(f_*^\infty, R_2)$ for all policies R_1 and R_2, i.e. f_*^∞ and $g^\infty \in G$ are optimal deterministic policies. □

Remark

Like discounted stochastic games, it is for undiscounted stochastic games with perfect information also an open problem to find an efficient finite algorithm.

Irreducible games

The class of *irreducible stochastic games* are characterized by the property that for each pair of stationary policies, say $(\pi^\infty, \rho^\infty)$, the Markov chain $P(\pi,\rho)$ is an irreducible Markov chain. We first show a lemma on the relation between the linear program (6.3) and the optimality equation (6.1).

Lemma 10.20

Any optimal solution (x^*, y^*) of the linear program (6.3) is a solution of the optimality equation (6.1), i.e. $x^* + y_i^* = max_{a\in A(i)}\{r_i(a) + \sum_j p_{ij}(a)y_j^*\},\ i \in S.$

Proof

From the constraints of the linear program (6.3) it follows that

$$x^* + y_i^* \geq max_{a \in A(i)} \big\{ r_i(a) + \textstyle\sum_j \, p_{ij}(a) y_j^* \big\}, \; i \in S.$$

From the theory of irreducible MDPs we know that any feasible solution of the set

$$\begin{cases} \sum_{i,a} \{\delta_{ij} - p_{ij}(a,\rho)\} x_i(a) = 0, j \in S \\ \sum_{i,a} \, x_i(a) = 1; \; x_i(a) \geq 0, \; i \in S, \; a \in A(i) \end{cases}$$

satisfies $\sum_a \, x_j(a) > 0$ for all $j \in S$. The complementary slackness property of linear programming implies that any $(i,a) \in S \times A$ with $x_i^*(a) > 0$, satisfies $x^* + \sum_j \{\delta_{ij} - p_{ij}(a)\} y_j^* = r_i(a)$. Hence,

$$x^* + y_i^* = max_{a \in A(i)} \big\{ r_i(a) + \textstyle\sum_j \, p_{ij}(a) y_j^* \big\}, \; i \in S. \qquad \square$$

Suppose that player 2 plays a fixed stationary policy ρ^∞. Then, the game becomes an MDP for player 1 and let $\phi(\rho) = max_{R_1} \, \phi(R_1, \rho^\infty)$. Because of the property of irreducibility, $\phi(\rho)$ has identical components. So, we may view $\phi(\rho)$ as a real function of ρ. We will first show that $\phi(\rho)$ is a continuous function of ρ. Therefore, we consider the following set of linear (in)equalities, which are a combination of the linear programs (6.3) and (6.4).

$$\begin{cases} z + \sum_j \{\delta_{ij} - p_{ij}(a,\rho)\} y_j & \geq & r_i(a,\rho), & i \in S, \; a \in A(i) \\ \sum_{i,a} \{\delta_{ij} - p_{ij}(a,\rho)\} x_i(a) & = & 0, & j \in S \\ \sum_{i,a} \, x_i(a) & = & 1 & \\ \sum_{i,a} \, r_i(a,\rho) x_i(a) - z & \geq & 0 & \\ x_i(a) & \geq & 0, & i \in S, \; a \in A(i) \\ y_1 & = & 0 & \end{cases} \tag{10.69}$$

Since, without $y_1 = 0$, for any solution (z,y,x) also $(z, y + c \cdot e, x)$ is a solution, this additional constraint may be imposed. From the theory of linear programming we know that for any pair of feasible solutions (z,y) and x of (6.3) and (6.4), respectively, the value of the objective function z is at least the value of the objective function $\sum_{i,a} \, r_i(a) x_i(a)$. Hence, the inequality $\sum_{i,a} \, r_i(a,\rho) x_i(a) - z \geq 0$ implies that only optimal solutions are feasible for (10.69) and that $z = \phi(\rho)$.

Consider a sequence of $\rho^n, \; n = 1, 2, \ldots$, and the corresponding feasible solutions (z^n, y^n, x^n) of (10.69). Then, these elements are bounded, namely:

(1) $\rho_{ib}^n \geq 0$ for all $(i,b) \in S \times B$ and $\sum_b \, \rho_{ib}^n = 1$ for all $i \in S$: the set $\{\rho^n\}_{n=1}^\infty$ is bounded.

(2) $z^n = \phi(\rho^n)$, which is bounded because $\phi(\rho^n) \leq max_{(i,a,b)} \, |r_i(a,b)|$: the set $\{z^n\}_{n=1}^\infty$ is bounded.

(3) From Lemma 10.20 we obtain $z^n + y_i^n = max_{a \in A(i)} \, \big\{ r_i(a\rho^n) + \sum_j \, p_{ij}(a,\rho^n) y_j^n \big\}, \; i \in S$. Then, with $y_1^n = 0$, Theorem 6.1 yields that $y^n = u^0\big(f_0(\rho^n)\big) - u_1^0\big(f_0(\rho^n)\big) \cdot e$, where $\big(f_0(\rho^n)\big)$ is a Blackwell optimal deterministic stationary policy in the MDP induced by ρ^n. Since there are only a finite number of deterministic stationary policies there are only a finite number of different y^n: the set $\{y^n\}_{n=1}^\infty$ is bounded.

(4) $x_i(a) \geq 0$ for all $(i,a) \in S \times A$ and $\sum_{i,a} x_i(a) = 1$: the set $\{x^n\}_{n=1}^\infty$ is bounded.

Consider a limit point (ρ^*, z^*, y^*, x^*) of the sequence $\{(\rho^n, z^n, y^n, x^n)\}_{n=1}^\infty$. For convenience, let $(\rho^*, z^*, y^*, x^*) = \lim_{n\to\infty}(\rho^n, z^n, y^n, x^n)$. For $z^* = lim_{n\to\infty} z^n = lim_{n\to\infty}\phi(\rho^n)$ we have to show $z^* = \phi(\rho^*)$, i.e. $z^* \geq \phi\big(\pi^\infty, (\rho^*)^\infty\big)$ for all $\pi^\infty \in \Pi$ and $z^* = \phi\big((\pi^*)^\infty, (\rho^*)^\infty\big)$ for some $(\pi^*)^\infty \in \Pi$. From the first set of the constraints of (10.69), we obtain $z^* \cdot e + \{I - P(\pi, \rho^*\}y^* \geq r(\pi, \rho^*)$, implying, by multiplication with $P^*(\pi, \rho^*)$, that $z^* \geq \phi\big(\pi^\infty, (\rho^*)^\infty\big)$ for all $\pi^\infty \in \Pi$. Let $(\pi^n)^\infty$ be the stationary policy that corresponds to x^n (see Theorem 6.5). The linear function $\sum_{i,a} r_i(a, \rho^n)x_i^n(a) = \phi\big((\pi^n)^\infty, (\rho^n)^\infty\big) \;\to\; \sum_{i,a} r_i(a, \rho^*)x_i^*(a) = \phi\big((\pi^*)^\infty, (\rho^*)^\infty\big)$, where $(\pi^*)^\infty$ is the stationary policy that corresponds to x^*. From the fourth constraint of (10.69) it follows that $\phi\big((\pi^*)^\infty, (\rho^*)^\infty\big) \geq z^*$. Hence, we have shown that $\phi\big((\pi^*)^\infty, (\rho^*)^\infty\big) \geq z^* \geq \phi\big(\pi^\infty, (\rho^*)^\infty\big)$ for all $\pi^\infty \in \Pi$, i.e. $z^* = \phi(\rho^*)$, completing the proof that $\phi(\rho)$ is a continuous function of ρ.

The function $\phi(\rho)$ is continuous on the compact set Γ of all stationary policies. Therefore, there exists a stationary policy, say $(\rho^*)^\infty \in \Gamma$ such that $\phi(\rho^*) = min_{\rho^\infty\in\Gamma}\, max_{R_1}\, \phi(R_1, \rho^\infty)$. We will show that $(\rho^*)^\infty$ is an optimal policy for player 2. Therefore, we consider the associate MDP with rewards $r_i(a, \rho^*),\ (i, a) \in S \times A$ and transition probabilities $p_{ij}(a, \rho^*),\ j \in S,\ (i, a) \in S \times A$. For this model, let (x^*, y^*) be an optimal solution of the linear program (6.3).

Let $M_x[i]$ be a payoff matrix with $m = \#A(i)$ rows and $n = \#B(i)$ columns and with payoff $r_i(a, b) + \sum_j p_{ij}(a, b)x_j$, if player 1 chooses row a and player 2 column b.

Theorem 10.37

Let (x^*, y^*) be an optimal solution of the linear program (6.3) associated with policy $(\rho^*)^\infty$ for player 2. Then, $x^* + y_i^* = val\big(M_{y^*}[i]\big)$ for all $i \in S$.

Proof

We have to show that $max_a\, min_b\, \{r_i(a, b) + \sum_j p_{ij}(a, b)y_j^*\} = x + y_i^*,\ i \in S$. From Lemma 10.20 it follows that

$$x^* + y_i^* = max_{a\in A(i)}\, \{r_i(a, \rho^*) + \sum_j p_{ij}(a, \rho^*)y_j^*\},\ i \in S, \tag{10.70}$$

implying $max_a\, min_b\, \{r_i(a, b) + \sum_j p_{ij}(a, b)y_j^*\} \leq max_a\, \{r_i(a, \rho^*) + \sum_j p_{ij}(a, \rho^*)y_j^*\} = x + y_i^*$ for all $i \in S$.

Finally, we have to show that $max_a\, min_b\, \{r_i(a, b) + \sum_j p_{ij}(a, b)y_j^*\} \geq x + y_i^*,\ i \in S$. Suppose the contrary, i.e. there is a state $k \in S$ and a mixed strategy $\{\rho_{kb},\ b \in B(k)\}$ such that

$$x^* + y_k^* > max_{a\in A(k)}\, \{r_k(a, \rho) + \sum_j p_{kj}(a, \rho)y_j^*\}. \tag{10.71}$$

Consider the policy $\overline{\rho}^\infty$ defined by $\overline{\rho}_{ib} = \begin{cases} \rho_{ib}^* & \text{if } i \neq k,\ b \in B(i); \\ \rho_{kb} & \text{if } i = k,\ b \in B(k). \end{cases}$

Then, $\phi(\overline{\rho})$ is the optimum value of the linear program associated with policy $\overline{\rho}^\infty$. Since (x^*, y^*) is also feasible for this linear program (because of (10.70) and (10.71)) and Lemma 10.20 is not satisfied (because of (10.38)), we obtain $\phi(\overline{\rho}) < \phi(\rho^*)$. However, this contradicts the property of ρ^*, namely that $\phi(\rho^*) = min_{\rho^\infty\in\Gamma}\, \phi(\rho)$. □

Theorem 10.38
If $x + y_i = val(M_y[i]),\ i \in S$ and $x^* + y^*_i = val(M_{y^*}[i]),\ i \in S$, then $x = x^*$ and $y = y^* + c \cdot e$ for some scalar c, i.e. x is unique and y is unique up to an additional constant.

Proof
Let $\{\pi^*_{ia},\ a \in A(i)\}$ be an optimal mixed strategy for player 1 in the matrix game $M^*_y[i]$, and let $\{\rho_{ib},\ b \in B(i)\}$ be an optimal mixed strategy for player 2 in the matrix game $M_y[i]$. Therefore,

$$val(M_y[i]) = x + y_i \geq r_i(\pi^*, \rho) + \textstyle\sum_j p_{ij}(\pi^*, \rho)y_j.$$

and

$$val(M_{y^*}[i]) = x^* + y^*_i \leq r_i(\pi^*, \rho) + \textstyle\sum_j p_{ij}(\pi^*, \rho)y^*_j.$$

Subtracting the second inequality from the first one obtains

$$(x - x^*) \cdot e + (y - y^*) \geq P(\pi^*, \rho)(y - y^*).$$

Multiplying this equation by $P^*(\pi^*, \rho)$ yields $x \geq x^*$. Interchanging the roles of the solutions (x, y) and (x^*, y^*), we may establish similarly that $x^* \geq x$. Therefore, $x = x^*$, and, setting $x - x^* = 0$ and $z = y - y^*$, we have $z - P(\pi^*, \rho)z \geq 0$. Since $P^*(\pi^*, \rho)\{z - P(\pi^*, \rho)z\} = 0$ and $P^*(\pi^*, \rho)$ is a matrix with strictly positive elements, we obtain $z = P(\pi^*, \rho)$, implying $z = P^*(\pi^*, \rho)z$. Because the matrix $P^*(\pi^*, \rho)$ has identical rows, all components of $z = y - y^*$ are equal. Hence, $y = y^* + c \cdot e$ for some scalar c. □

Corollary 10.6 *The equation $x + y_i = val(M_y[i]),\ i \in S$, has a solution (x^*, y^*) in which x^* is unique. Furthermore, x^* is the value of the stochastic game and optimal strategies in the matrix games $M_{y^*}[i],\ i \in S$, are optimal stationary policies for the stochastic game.*

Proof
From the Theorems 10.37 and 10.38 it follows that the equation $x + y_i = val(M_y[i]),\ i \in S$, has a solution (x^*, y^*) in which x^* is unique. Let $\{\pi^*_{ia},\ a \in A(i)\}$ and $\{\rho^*_{ia},\ b \in B(i)\}$ be optimal mixed strategies for player 1 and 2, respectively, in the matrix game $M^*_y[i],\ i \in S$. Then,

$$r(\pi, \rho^*) + P(\pi, \rho^*)y^* \leq x^* \cdot e + y^* \leq r(\pi^*, \rho) + P(\pi^*, \rho)y^* \text{ for all } \pi^\infty \in F \text{ and } \rho^\infty \in \Gamma.$$

Hence, by multiplying the first inequality by $P^*(\pi, \rho^*)$ we obtain $\phi\big(\pi^\infty, (\rho^*)^\infty\big) \leq x^*$. Similarly, by multiplying the second inequality by $P^*(\pi^*, \rho)$ we obtain $x^* \leq \phi\big((\pi^*)^\infty, \rho^\infty\big)$. Therefore, $(\pi^*)^\infty$ and $(\rho^*)^\infty$ are optimal stationary policies and x^* is the value. □

Algorithm 10.15 *Value iteration for undiscounted games (irreducible case)*
Input: Instance of a two-person irreducible stochastic game with perfect information.
Output: The value and a pair $(\pi^*)^\infty$ and $(\rho^*)^\infty$ of stationary optimal policies.

1. $t := 0$; select any stationary policy $(\rho^t)^\infty$ for player 2.

2. Solve the MDP induced by policy $(\rho^t)^\infty$, i.e. compute (x^t, y^t) such that $y^t_1 = 0$ and

   $x^t + y^t_i = max_a \{r_i(a, \rho^t) + \sum_j p_{ij}(a, \rho^t)y^t_j\},\ i \in S.$

```
3. for all $I \in S$ do
      determine optimal stationary strategies $\pi_{ia}^{t+1},\ a \in A(i)$, for player 1, and $\rho_{ib}^{t+1}$,
      $b \in B(i)$, for player 2, in the matrix game $M_{y^t}[i]$, where $M_{y^t}[i]$ is the matrix with
      entries $r_i(a,b) + \sum_j p_{ij}(a,b)y_j^t,\ a \in A(i),\ b \in B(i)$.
4. if $val\big(M_{y^t}[i]\big) = x^t + y_i^t,\ i \in S$ then
      begin $x^* := x^t$ is the value; $(\pi^*)^\infty := (\pi^{t+1})^\infty$ and $(\rho^*)^\infty := (\rho^{t+1})^\infty$ are optimal
            stationary policies for player 1 and 2, respectively (STOP)
      end
   else begin $t := t+1$; return to step 2 end
```

Remark

Step 2 of this algorithm can be solved by the linear program (6.3). We will show that the sequence $\{x^t,\ t = 0,1,\dots\}$ converges to the value of the stochastic game.

Theorem 10.39

(1) *The sequences $\{x^t,\ t = 0,1,\dots\}$ and $\{y^t,\ t = 0,1,\dots\}$ are convergent.*

(2) *Let $x^* = lim_{t\to\infty} x^t$ and $y^* = lim_{t\to\infty} y^t$. Then $x^* + y_i^* = val(M_{y^*}[i]),\ i \in S$.*

(3) *If $x^t + y_i^t = val(M_{y^t}[i]),\ i \in S$, then x^t is the value and $(\pi^{t+1})^\infty$ and $(\rho^{t+1})^\infty$ are optimal stationary policies for player 1 and 2, respectively.*

Proof

Since π^{t+1} and ρ^{t+1} are optimal strategies in the matrix games $M_{y^t}[i],\ i \in S$, we have

$$val(M_{y^t}) \geq r(\pi, \rho^{t+1}) + P(\pi, \rho^{t+1})y^t \text{ for all } \pi^\infty \in \Pi,$$

and

$$val(M_{y^t}) \leq r(\pi^{t+1}, \rho) + P(\pi^{t+1}, \rho)y^t \text{ for all } \rho^\infty \in \Gamma,$$

implying $val(M_{y^t}) \leq r(\pi^{t+1}, \rho^t) + P(\pi^{t+1}, \rho^t)y^t$. By step 2 of the algorithm we have

$$y^t = r(\pi^{t+1}, \rho^t) + P(\pi^{t+1}, \rho^t)y^t - x^t \cdot e \geq val(M_{y^t}) - x^t \cdot e.$$

Therefore, $val(M_{y^t}) \geq r(\pi, \rho^{t+1}) + P(\pi, \rho^{t+1})\{val(M_{y^t}) - x^t \cdot e\}$ for all $\pi^\infty \in \Pi$. Multiplication with $P^*(\pi, \rho^{t+1})$ gives

$$x^t \geq \phi\big(\pi^\infty, (\rho^{t+1})^\infty\big) \text{ for all } \pi^\infty \in \Pi, \text{ i.e. } x^t \geq max_{\pi^\infty \in \Pi}\, \phi\big(\pi^\infty, (\rho^{t+1})^\infty\big) = x^{t+1}.$$

Hence, the sequence $\{x^t,\ t = 0,1,\dots\}$ is nonincreasing and bounded below by $-max|r_i(a,b)|$: $\{x^t,\ t = 0,1,\dots\}$ is convergent. In the proof that $\phi(\rho)$, defined as $\phi(\rho) = max_{R_1}\, \phi(R_1, \rho^\infty)$, is a continuous function of ρ, we have seen that $\{y^t\}_{t=1}^\infty$ is a bounded sequence. Therefore, we may choose a convergent subsequence of vectors $\{(x^t, y^t)\}_{t=1}^\infty$, and let us denote the vector to which they converge by (x^+, y^+). Let the corresponding stationary policies ρ^t for player 2 converge to ρ^+. Since ρ^{t+1} is an optimal solution for player 2 in the matrix games $M_{y^t}[i],\ i \in S$, it follows by continuity that ρ^+ is an optimal policy in the matrix games $M_{y^+}[i],\ i \in S$.

Since $x^t + y_i^t \geq val\big(M_{y^t}\big)[i]$, it follows - also by continuity - that $x^+ + y_i^+ \geq val\big(M_{y^+}\big)[i]$. If, for some k, $x^+ + y_k^+ > val\big(M_{y^+}\big)[k]$, then this implies $x^+ + y_k^+ > max_a\,\{r_i(a,\rho^+) + \sum_j p_{kj}(a,\rho^+)y_j^+\}$. Similarly as in the proof of Theorem 10.37 we obtain $\phi(\rho^+) < x^+$. But $x^+ \leq x^t = \phi(\rho^t)$ and the continuity of $\phi(\rho)$ imply that $x^+ \leq \phi(\rho^+)$, which yields a contradiction. This contradiction establishes that $x^+ + y_i^+ = val\big(M_{y^+}\big)[i],\ i \in S$. Because the solution of this functional equation is unique, every convergent subsequence has the same limit, and it follows that the sequence produced in the algorithm converges to this functional equation. Part (3) follows directly from Corollary 10.6. □

10.4.5 Finite methods

When the value and the optimal policies lie in the same ordered field as the data one can hope to arrive at a solution by a finite number of operations. If the ordered field property is not valid then one can only try iterative procedures for solving these stochastic games. As is the case for discounted stochastic games, also in undiscounted stochastic games the ordered field property does not hold, in general. This is illustrated by the following example.

Example 10.7 (continued)

In Example 10.7 we have derived that $v_1^\alpha = \frac{-(1+\alpha)+\sqrt{(1+\alpha)}}{1-\alpha}$. It can be shown[2] that ϕ, the value vector of the undiscounted game, satisfies $\phi = lim_{\alpha\uparrow 1}\,(1-\alpha)v^\alpha$, where v^α is the value vector of the α-discounted stochastic game. Hence, $\phi_1 = lim_{\alpha\uparrow 1}\,\{-(1+\alpha)+\sqrt{1+\alpha}\} = -2+\sqrt{2}$, which lies not in the ordered field of the rational numbers.

We consider the following special games, which have the ordered field property, as we will show:
(1) The single-controller stochastic game.
(2) The switching-controller stochastic game.
(3) The separabel reward - state independent transitions (SER-SIT) stochastic game.
(4) The additive reward - additive transitions (ARAT) stochastic game.

Single-controller stochastic game: the multichain case

In the single-controller stochastic game, where player 1 is the 'single-controller', the transition probabilities $p_{ij}(a,b)$ are independent of b denoted by $p_{ij}(a)$. Under this assumption the concept of superharmonicity for a vector $v \in \mathbb{R}^N$ means that there exists a vector $t \in \mathbb{R}^N$ and a policy $\rho^\infty \in \Gamma$ such that the triple (v,t,ρ) satisfies

$$\begin{cases} v_i \geq \sum_j p_{ij}(a)v_j & \text{for every } (i,a) \in S \times A; \\ v_i + t_i \geq r_i(a,\rho) + \sum_j p_{ij}(a)t_j & \text{for every } (i,a) \in S \times A. \end{cases} \tag{10.72}$$

[2] see Corollary 5.2.7 in [99]

Therefore, the problem to find the smallest superharmonic vector is the following linear program

$$min \left\{ \sum_i v_i \;\middle|\; \begin{array}{rcrcrl} \sum_j \{\delta_{ij} - p_{ij}(a)\}v_j & & & & & \geq 0,\ a \in A(i),\ i \in S \\ v_i & + & \sum_j \{\delta_{ij} - p_{ij}(a)\}t_j & - & \sum_b r_i(a,b)\rho_{ib} & \geq 0,\ a \in A(i),\ i \in S \\ & & & & \sum_b \rho_{ib} & = 1,\ i \in S \\ & & & & \rho_{ib} & \geq 0,\ b \in B(i),\ i \in S \end{array} \right\}. \tag{10.73}$$

The dual program is

$$max \left\{ \sum_i z_i \;\middle|\; \begin{array}{rcrcl} \sum_{(i,a)} \{\delta_{ij} - p_{ij}(a)\}x_i(a) & & & = & 0,\ j \in S \\ \sum_a x_j(a) & + & \sum_{(i,a)} \{\delta_{ij} - p_{ij}(a)\}y_i(a) & = & 1,\ j \in S \\ -\sum_a r_i(a,b)x_i(a) & + & z_i & \leq & 0,\ (i,b) \in S \times B \\ & & x_i(a), y_i(a) & \geq & 0,\ (i,a) \in S \times A \end{array} \right\}. \tag{10.74}$$

Lemma 10.21

The linear programs (10.73) and (10.74) have finite optimal solutions.

Proof

Take an arbitrary stationary policy ρ^∞ for player 2, and let $t = 0$ and $v_i = max_{i,a,b}\, r_i(a,b),\ i \in S$. Then, (v, t, ρ) is a feasible solution of (10.73). For the existence of finite optimal solutions it is sufficient to show, by the duality theorem of linear programming, that the optimum of (10.73) is bounded below. Let (v, t, ρ) be any feasible solution of (10.73). Then, for any $\pi^\infty \in F$, we have from the first equations of (10.73) $v \geq P(\pi)v$, implying $v \geq P^*(\pi)v$. From the second set of equation, we obtain $v + \{I - P(\pi)\}t \geq r(\pi, \rho)$. Therefore, we can write,

$$v \geq P^*(\pi)v \geq P^*(\pi)\big\{\{r(\pi,\rho) - \{I - P(\pi)\}t\big\} = P^*(\pi)\{r(\pi,\rho) = \phi(\pi^\infty, \rho^\infty).$$

Now we have $v_i \geq \phi_i(\pi^\infty, \rho^\infty) \geq min_{i,a,b}\, r_i(a,b),\ i \in S$, which shows that the optimum of (10.73) is bounded below. □

The following theorem shows that the value vector and optimal stationary policies for both players can be obtained from the optimal solutions of the dual pair of linear programs.

Theorem 10.40

Let (v^*, t^*, ρ^*) and (x^*, y^*, z^*) be optimal solutions of the linear programs (10.73) and (10.74). Define the policy $(\pi^*)^\infty$ by $\pi^*_{ia} := \begin{cases} \frac{x_i^*(a)}{\sum_a x_i^*(a)},\ i \in S_{x^*},\ a \in A(i),\ \textit{where } S_{x^*} := \{i \mid \sum_a x_i^*(a) > 0\}; \\ \frac{y_i^*(a)}{\sum_a y_i^*(a)},\ i \notin S_{x^*},\ a \in A(i). \end{cases}$

Then, v^* is the value vector and $(\pi^*)^\infty$ and $(\rho^*)^\infty$ are optimal stationary policies for player 1 and 2, respectively.

Proof

The constraints of program (10.74) imply $\sum_a x_j^*(a)+\sum_a y_j^*(a) = 1+\sum_{(i,a)} p_{ij}(a)y_i^*(a) > 0,\ j \in S$. Hence, the policy $(\pi^*)^\infty$ is well-defined. From the constraints of program (10.73) we obtain

$$v^* \geq P(\pi)v^* \text{ and } v^* \geq r(\pi,\rho^*) - \{I - P(\pi)\}t^* \text{ for all } \pi^\infty \in \Pi.$$

Therefore, we have

$$v^* \geq P^*(\pi)v^* \geq P^*(\pi)\{r(\pi,\rho^*) - \{I-P(\pi)\}t^*\} = \phi\big(\pi^\infty,(\rho^*)^\infty\big) \text{ for all } \pi^\infty \in \Pi. \tag{10.75}$$

Since $\pi_{ia}^* > 0$ if and only if $\begin{cases} x_i^*(a) > 0 \text{ for } i \in S_{x^*} \\ y_i^*(a) > 0 \text{ for } i \notin S_{x^*} \end{cases}$ it follows from the complementary slackness property of linear programming that

$$\begin{cases} \sum_a \pi_i^*(a) \cdot \{v_i^* + \sum_j \{\delta_{ij} - p_{ij}(a)\}t_j^* - \sum_b r_i(a,b)\rho_{ib}^*\} & = & 0,\ i \in S_{x^*} \\ \sum_a \pi_i^*(a) \cdot \sum_j \{\delta_{ij} - p_{ij}(a)\}v_j^* & = & 0,\ i \notin S_{x^*} \end{cases}$$

Suppose that $\pi_k^*(a_k) \cdot \sum_j \{\delta_{kj} - p_{kj}(a_k)\}v_j^* \neq 0$ for some $k \in S_{x^*},\ a_k \in A(k)$. Then, the definition of π^* and the constraints of (10.73) imply that $x_k^*(a_k) \cdot \sum_j \{\delta_{kj} - p_{kj}(a_k)\}v_j^* > 0$. Hence, we get $\sum_{(i,a)} x_i^*(a) \cdot \sum_j \{\delta_{ij} - p_{ij}(a)\}v_j^* > 0$, which is contradictory to

$$\sum_{(i,a)} x_i^*(a) \cdot \sum_j \{\delta_{ij} - p_{ij}(a)\}v_j^* = \sum_j \big\{ \sum_{(i,a)} \{\delta_{ij} - p_{ij}(a)\}x_i^*(a)\big\}v_j^* = 0.$$

Therefore, we obtain

$$\begin{cases} \sum_a \pi_i^*(a) \cdot \{v_i^* + \sum_j \{\delta_{ij} - p_{ij}(a)\}t_j^* - \sum_b r_i(a,b)\rho_{ib}^*\} & = & 0,\ i \in S_{x^*}; \\ \sum_a \pi_i^*(a) \cdot \sum_j \{\delta_{ij} - p_{ij}(a)\}v_j^* & = & 0,\ i \in S. \end{cases}$$

Hence,

$$\begin{cases} v_i^* & + & \big\{\{I - P(\pi^*)\}t^*\big\}_i & = & r_i(\pi^*,\rho^*), & i \in S_{x^*}; \\ \big\{\{I - P(\pi^*)\}v^*\big\}_i & & & = & 0, & i \in S. \end{cases}$$

The second equation implies $v^* = P^*(\pi^*)v^*$. Since S_{x^*} is the set of recurrent states in the Markov chain induced by $P(\pi^*)$ (see the proof of Theorem 5.20), we obtain

$$v^* = P^*(\pi^*)v^* = P^*(\pi^*)\big\{r(\pi^*,\rho^*) - \{I - P(\pi^*)\}t^*\big\} = P^*(\pi^*)r(\pi^*,\rho^*) = \phi\big((\pi^*)^\infty,(\rho^*)^\infty\big),$$

implying, using (10.75),

$$\phi\big(\pi^\infty,(\rho^*)^\infty\big) \leq v^* = \phi\big((\pi^*)^\infty,(\rho^*)^\infty\big) \text{ for all } \pi^\infty \in \Pi. \tag{10.76}$$

Let $x_i^* := \sum_a x_i^*(a),\ i \in S$. Suppose that $S_1, S_2, \dots, S_m$ are the ergodic sets and let T be the set of transient states in the Markov chain induced by $P(\pi^*)$. Let $n_k = |S_k|,\ k = 1,2,\dots,m$. Then, we shall show that $x^* = \{P^*(\pi^*)\}^T\gamma$, where γ is a strictly positive vector with elements

$\gamma_l := \begin{cases} \frac{1}{n} & l \in T; \\ \frac{1}{n_k} \cdot \sum_{j \in S_k} \{x_j^* - \frac{1}{n}\sum_{i \in T} p_{ij}^*(\pi^*)\} & l \in S_k,\ k = 1,2,\dots,m. \end{cases}$, where n is sufficienlty large such that $\gamma_l > 0$, i.e. $n > max_{j \in S_{x^*}} \big\{\frac{1}{x_j^*} \cdot \sum_{i \in T} p_{ij}^*(\pi^*)\big\}$. Now, we have

$$\begin{aligned} \textstyle\sum_l \gamma_l \cdot \sum_{j\in S_k} p^*_{lj}(\pi^*) &= \textstyle\sum_{l\in T} \gamma_l \cdot \sum_{j\in S_k} p^*_{lj}(\pi^*) + \sum_{l\in S_k} \gamma_l \cdot \sum_{j\in S_k} p^*_{lj}(\pi^*) \\ &= \textstyle\frac{1}{n}\sum_{l\in T}\sum_{j\in S_k} p^*_{lj}(\pi^*) + \sum_{l\in S_k}\gamma_l \\ &= \textstyle\frac{1}{n}\sum_{l\in T}\sum_{j\in S_k} p^*_{lj}(\pi^*) + \sum_{l\in S_k}\{\frac{1}{n_k}\cdot\sum_{j\in S_k}\{x^*_j - \frac{1}{n}\sum_{i\in T}p^*_{ij}(\pi^*)\}\} \\ &= \textstyle\frac{1}{n}\sum_{l\in T}\sum_{j\in S_k}p^*_{lj}(\pi^*) + \sum_{j\in S_k}\{x^*_j - \frac{1}{n}\sum_{i\in T}p^*_{ij}(\pi^*)\}\} \\ &= \textstyle\sum_{j\in S_k} x^*_j,\ k=1,2,\dots,m. \end{aligned}$$

From program (10.74) and the definition of π^* it follows that $x^* = \{P(\pi^*)\}^T x^*$ and, consequently, $x^* = \{P^*(\pi^*)\}^T x^*$. Since $S\backslash S_{x^*}$ is the set of transient states T in the Markov chain induced by $P(\pi^*)$ (see the proof of Theorem 5.20), we have $p^*_{li} = 0,\ l \in S$. Therefore, we obtain

$$0 = x^*_i = \sum_l p^*_{li}(\pi^*)\gamma_l = \big\{\{P^*(\pi^*)\}^T\gamma\big\}_i,\ i \notin S_{x^*}. \tag{10.77}$$

For $i \in S_k$, it follows that

$$\begin{aligned} x^*_i &= \textstyle\sum_j p^*_{ji}(\pi^*)x^*_j = \sum_{j\in S_k} p^*_{ji}(\pi^*)x^*_j + \sum_{j\in T} p^*_{ji}(\pi^*)x^*_j \\ &= \textstyle p^*_{ii}(\pi^*)\cdot\sum_{j\in S_k} x^*_j + \sum_{j\in S\backslash S_{x^*}} p^*_{ji}(\pi^*)x^*_j \ = \ p^*_{ii}(\pi^*)\cdot\big\{\sum_l \gamma_l\cdot\sum_{j\in S_k} p^*_{lj}(\pi^*)\big\} + 0 \\ &= \textstyle\sum_l \gamma_l\cdot\sum_{j\in S_k} p^*_{lj}(\pi^*)p^*_{ji}(\pi^*) \ = \ \sum_l \gamma_l\cdot p^*_{li}(\pi^*), \end{aligned}$$

implying

$$x^*_i = \big\{\{P^*(\pi^*)\}^T\gamma\big\}_i,\ i\in S_k,\ k=1,2,\dots,m. \tag{10.78}$$

Combining (10.77) and (10.78) yields $x^* = \{P^*(\pi^*)\}^T\gamma$. Using again the complementary slackness property of linear programming yields

$$\textstyle\sum_i\sum_b \rho_{ib}\cdot\{z^*_i - \sum_a r_i(a,b)x^*_i(a)\} = 0.$$

Therefore,

$$\begin{aligned} \textstyle\sum_i z^*_i &= \textstyle\sum_i\sum_b\sum_a r_i(a,b)\rho^*_{ib}x^*_i(a) \ = \ \sum_i\big\{\sum_b\sum_a r_i(a,b)\rho^*_{ib}\pi^*_{ia}\cdot x^*_i\big\} \\ &= \textstyle\sum_i\big\{r_i(\pi^*,\rho^*)\cdot x^*_i\big\} \ = \ \sum_i\big\{r_i(\pi^*,\rho^*)\cdot\sum_l\gamma_l\cdot p^*_{li}(\pi^*)\big\} \\ &= \textstyle\sum_l\gamma_l\cdot\big\{\sum_i p^*_{li}(\pi^*)r_i(\pi^*,\rho^*)\big\}, \end{aligned}$$

implying

$$\sum_i z^*_i = \gamma^T\phi\big((\pi^*)^\infty,(\rho^*)^\infty\big). \tag{10.79}$$

For any stationary policy $\rho^\infty\in\Gamma$, we have in view of the constraints of linear program (10.74)

$$\sum_i z^*_i = \sum_i\sum_b \rho_{ib}z^*_i \le \sum_i\sum_b\sum_a r_i(a,b)\rho_{ib}\pi^*_{ia}\cdot x^*_i = \gamma^T\phi\big((\pi^*)^\infty,\rho^\infty\big). \tag{10.80}$$

Since γ is strictly positive, (10.79) and (10.80) yields

$$\phi\big((\pi^*)^\infty,(\rho^*)^\infty\big) \le \phi\big((\pi^*)^\infty,\rho^\infty\big) \text{ for every } \rho^\infty\in\Gamma. \tag{10.81}$$

From (10.76) and (10.81) we obtain

$$\phi\big(\pi^\infty,(\rho^*)^\infty\big) \le v^* = \phi\big((\pi^*)^\infty,(\rho^*)^\infty\big) \le \phi\big((\pi^*)^\infty,\rho^\infty\big) \text{ for all } \pi^\infty\in\Pi \text{ and } \rho^\infty\in\Gamma, \tag{10.82}$$

showing that v^* is the value vector and $(\pi^*)^\infty$ and $(\rho^*)^\infty$ are optimal stationary policies for player 1 and 2, respectively.

□

Algorithm 10.16 *Single-controller game with no discounting*
Input: Instance of a two-person single-controller stochastic game.
Output: The value $v*$ and a pair $(\pi^*)^\infty$ and $(\rho^*)^\infty$ of stationary optimal policies.

1. Compute optimal solutions (v^*, t^*, ρ^*) and (x^*, y^*, z^*) of the linear programs (10.73) and (10.74), respectively.
2. Define the stationary policy $(\pi^*)^\infty$ by $\pi^*_{ia} := \begin{cases} \frac{x^*_i(a)}{\sum_a x^*_i(a)}, \ i \in S_{x^*}, \ a \in A(i) \\ \frac{y^*_i(a)}{\sum_a y^*_i(a)}, \ i \notin S_{x^*}, \ a \in A(i) \end{cases}$,
   where $S_{x^*} := \{i \mid \sum_a x^*_i(a) > 0\}$.
3. v^* is the value vector and $(\pi^*)^\infty$ and $(\rho^*)^\infty$ optimal stationary policies for player 1 and 2.

Example 10.5 (continued)
For this example the linear programs (10.73) and (10.74) become

minimize $v_1 + v_2$
subject to

$$\begin{array}{rrcrcrcrcrcrcrcrcrcrcl}
 & v_1 & - & v_2 & & & & & & & & & & & & & & & & \geq & 0 \\
- & v_1 & + & v_2 & & & & & & & & & & & & & & & & \geq & 0 \\
- & v_1 & + & v_2 & & & & & & & & & & & & & & & & \geq & 0 \\
 & v_1 & & & & & & & - & 5\rho_{11} & - & \rho_{12} & - & 6\rho_{13} & & & & & & \geq & 0 \\
 & v_1 & & & + & t_1 & - & t_2 & - & 4\rho_{11} & - & 6\rho_{12} & - & 2\rho_{13} & & & & & & \geq & 0 \\
 & & & v_2 & - & t_1 & + & t_2 & & & & & & & - & 6\rho_{21} & & & & \geq & 0 \\
 & & & v_2 & & & & & & & & & - & 3\rho_{21} & - & 4\rho_{22} & & & & \geq & 0 \\
 & & & v_2 & - & t_1 & + & t_2 & & & & & & & & & - & 6\rho_{22} & & \geq & 0
\end{array}$$

$$\rho_{11} + \rho_{12} + \rho_{13} = 1; \ \rho_{21} + \rho_{22} = 1; \ \rho_{11}, \rho_{12}, \rho_{13}, \rho_{21}, \rho_{22} \ \geq \ 0$$

and

maximize $z_1 + z_2$
subject to

$$\begin{array}{rrcrcrcrcrcrcrcrcrcl}
 & & & x_{12} & - & x_{21} & & & - & x_{23} & & & & & & & & & = & 1 \\
 & & - & x_{12} & + & x_{21} & & & + & x_{23} & & & & & & & & & = & 1 \\
 & x_{11} & + & x_{12} & & & & & & & + & y_{12} & - & y_{21} & - & y_{23} & & & = & 1 \\
 & & & & & x_{21} & + & x_{22} & & & - & y_{12} & + & y_{21} & + & y_{23} & & & = & 1 \\
- & 5x_{11} & - & 4x_{12} & & & & & & & & & & & & & + & z_1 & \leq & 0 \\
- & x_{11} & - & 6x_{12} & & & & & & & & & & & & & + & z_1 & \leq & 0 \\
- & 6x_{11} & - & 2x_{12} & & & & & & & & & & & & & + & z_1 & \leq & 0 \\
 & & & & - & 6x_{21} & - & 3x_{22} & & & & & & & & & + & z_2 & \leq & 0 \\
 & & & & & & - & 4x_{22} & - & 6x_{23} & & & & & & & + & z_2 & \leq & 0
\end{array}$$

$$x_{11}, x_{12}, x_{21}, x_{22}, x_{23}, y_{11}, y_{12}, y_{21}, y_{22}, y_{23} \ \geq \ 0$$

The optimal solutions are:
$v_1^* = 3.5,\ v_2^* = 3.5;\ t_1^* = 0.5;\ t_2 = 0;\ \rho_{11}^* = 0,\ \rho_{12}^* = 0.5,\ \rho_{13}^* = 0.5,\ \rho_{21}^* = 0.5,\ \rho_{22}^* = 0.5$ and
$z_1^* = 1.546,\ z_2^* = 5.454;\ x_{11}^* = 0.182,\ x_{12}^* = 0.227,\ x_{21}^* = 0.227,\ x_{22}^* = 1.364,\ x_{23}^* = 0;$
$y_{12}^* = 0.591,\ y_{21}^* = 0,\ y_{23}^* = 0;$
The optimal policy for player 1 is: $\pi_{11}^* = 0.444,\ \pi_{12}^* = 0.556,\ \pi_{21}^* = 0.143,\ \pi_{22}^* = 0.857,\ \pi_{23}^* = 0.$

Remark

Consider a *two-person zero-sum undiscounted semi-Markov game* in which player 1 controls the transitions. This model can be described as follows: state space S; action sets $A(i)$ and $B(i)$, $i \in S$, for player 1 and 2; transition probabilities $p_{ij}(a),\ (i,a) \in S \times A,\ j \in S$, which depend only on the actions chosen by player 1; immediate rewards $r_i(a,b),\ (i,a,b) \in S \times A \times B$; reward rates $s_i(a,b),\ (i,a,b) \in S \times A \times B$; sojourn time distributions $F_{ij}(a,t),\ (i,a) \in S \times A,\ j \in S$, which depend only on the actions chosen by player 1.

Let $\tau_i(a) := \sum_j p_{ij}(a) \cdot \int_0^\infty t\, dF_{ij}(a,t)$ and $r_i^*(a) := r_i(a) + \tau_i(a) \cdot s_i(a),\ (i,a) \in S \times A$. From these quantities we compute the transition numbers $\overline{p}_{ij}(a) := \delta_{ij} - \{\delta_{ij} - p_{ij}(a)\} \cdot \frac{\tau}{\tau_i(a)}$ for all $i,j \in S,\ a \in A(i)$, where τ is defined by (9.134), and the rewards $\overline{r}_i(a,b) := r_i^*(a,b) \cdot \frac{1}{\tau_i(a)}$ for all $(i,a,b) \in S \times A \times B,\ j \in S$.

Analogously to the analysis in Section 9.7.5 it can straightforward be shown that this undiscounted semi-Markov game is equivalent to the undiscounted Markov game $(S, A, B, \overline{p}, \overline{r})$. Therefore, the results of a single-controller undiscounted Markov game are also applicable to an undiscounted single-controller semi-Markov game.

Additional constraints

For the additional constraints we assume that, besides the immediate rewards, there are for $k = 1, 2, \dots, m$ also certain immediate costs which only depend on the state and the action chosen by player 1. These costs are denoted by $c_i^k(a),\ (i,a) \in S \times A,\ k = 1, 2, \dots, m$. For any pair of policies R_1 and R_2 for player 1 and 2, respectively, let the average reward and the average kth cost function with respect to an initial distribution β with $\beta_j \geq 0,\ j \in S$, be defined by

$\phi(\beta, R_1, R_2) := \liminf_{T\to\infty} \frac{1}{T} \sum_{t=1}^T \sum_j \beta_j \cdot \sum_{(i,a)} \mathbb{P}_{R_1,R_2}\{X_t = i,\ Y_t = a,\ Z_t = b \mid X_1 = j\} \cdot r_i(a,b)$

and

$c_k(\beta, R_1) := \liminf_{T\to\infty} \frac{1}{T} \sum_{t=1}^T \sum_j \beta_j \cdot \sum_{(i,a)} \mathbb{P}_{R_1,R_2}\{X_t = i,\ Y_t = a,\ Z_t = b \mid X_1 = j\} \cdot c_i^k(a).$

The constraints are: $c_k(\beta, R_1) \leq b_k,\ k = 1, 2, \dots, m$ for some real numbers $b_1, b_2, \dots, b_m$. Let $C_0^1 := \{R_1 \in C^1 \mid c_k(\beta, R_1) \leq b_k,\ k = 1, 2, \dots, m\}$, the set of feasible solutions for player 1.

As we have seen in Section 9.2.6, a constrained MDP has always an optimal Markov policy $R_1 = (\pi^1, \pi^2, \dots)$, but it does not have in general an optimal stationary policy. Player 2 does not influence the process, but only the payoffs. If player 1 chooses at time t the decision rule π^t, then an optimal action for player 2 in state i will be action b_*, where the action b_* is such that $r_i(\pi^t, b_*) = min_{b \in B(i)} r_i(\pi^t, b)$. Since this rule for player 2 is time dependent, an optimal policy for player 2 is also not stationary, in general. However, linear programming formulations deal with stationary policies for player 2. Therefore, we restrict the set of policies for player 2 to the set of stationary policies. When it turns out that player 1 has a stationary optimal policy, then an optimal policy for player 2 in the set $C^2(S)$ is also optimal in the set C^2 of all policies for player 2. We shall present some conditions under which player 1 has a stationary optimal policy.

A policy R_1^* is *optimal for player 1* in the constrained Markov game if $R_1^* \in C_0^1$ and

$$\inf_{\rho^\infty \in C^2(S)} \phi(\beta, R_1^*, \rho^\infty) = \sup_{R_1 \in C_0^1} \inf_{\rho^\infty \in C^2(S)} \phi(\beta, R_1, \rho^\infty). \tag{10.83}$$

A policy R_2^* is *optimal for player 2* in the constrained Markov game if $R_2^* \in C^2(S)$, say $R_2^* = (\rho^*)^\infty$ and

$$\sup_{R_1 \in C_0^1} \phi(\beta, R_1, (\rho^*)^\infty) = \inf_{\rho^\infty \in C^2(S)} \sup_{R_1 \in C_0^1} \phi(\beta, R_1, \rho^\infty). \tag{10.84}$$

The constrained Markov game has a *value* if

$$\sup_{R_1 \in C_0^1} \inf_{\rho^\infty \in C^2(S)} \phi(\beta, R_1, \rho^\infty) = \inf_{\rho^\infty \in C^2(S)} \sup_{R_1 \in C_0^1} \phi(\beta, R_1, \rho^\infty). \tag{10.85}$$

In order to find the value of the constrained Markov game and optimal policies for both players we consider the following dual pair of linear programs

$$max \left\{ \sum_i z_i \;\middle|\; \begin{array}{rcrcl}
\sum_{(i,a)}\{\delta_{ij} - p_{ij}(a)\}x_i(a) & & & = & 0,\ j \in S \\
\sum_a x_j(a) & + & \sum_{(i,a)}\{\delta_{ij} - p_{ij}(a)\}y_i(a) & = & \beta_j,\ j \in S \\
-\sum_a r_i(a,b)x_i(a) & + & z_i & \le & 0,\ (i,b) \in S \times B \\
\sum_{(i,a)} c_i^k(a)x_i(a) & & & \le & b_k,\ k = 1, 2, \ldots, m \\
 & & x_i(a), y_i(a) & \ge & 0,\ (i,a) \in S \times A
\end{array} \right\} \tag{10.86}$$

and

$$min \left\{ \begin{array}{l} \sum_i \beta_j v_i + \\ \sum_k b_k w_k \end{array} \;\middle|\; \begin{array}{rl}
\sum_j\{\delta_{ij} - p_{ij}(a)\}v_j & \ge 0,\ a \in A(i),\ i \in S \\
v_i + \sum_j\{\delta_{ij} - p_{ij}(a)\}t_j - \sum_b r_i(a,b)\rho_{ib} + \sum_k c_i^k(a)w_k & \ge 0,\ a \in A(i),\ i \in S \\
\sum_b \rho_{ib} & = 1,\ i \in S \\
\rho_{ib} & \ge 0,\ b \in B(i),\ i \in S \\
w_k & \ge 0,\ k = 1, 2, \ldots, m
\end{array} \right\}. \tag{10.87}$$

For any $R_1 \in C^1$, the average state-action frequencies and the vector sets $L, L(M), L(C), L(S), L(D)$ are defined as in Section 9.2.6. From Theorem 9.21, we have $L = L(M) = L(C) = \overline{LS)} = \overline{L(D)}$, where $\overline{LS)}$ and $\overline{L(D)}$ are the closed convex hull of the sets $L(S)$ and $L(D)$, respectively. Also the polyhedron Q is defined in Section 9.2.6, namely

$$Q := \left\{ x \;\middle|\; \begin{array}{rcrcl}
\sum_{(i,a)}\{\delta_{ij} - p_{ij}(a)\}x_{ia} & & & = & 0,\ j \in S \\
\sum_a x_{ja} & + & \sum_{(i,a)}\{\delta_{ij} - p_{ij}(a)\}y_{ia} & = & \beta_j,\ j \in S \\
 & & x_{ia},\ y_{ia} & \ge & 0,\ (i,a) \in S \times A
\end{array} \right\}. \tag{10.88}$$

Let (x, y, z) be a feasible solution of the linear programs (10.87). Then, one can construct, by the steps 3, 4, 5 and 6 of Algorithm 9.5, a policy $R_1 \in L(M) \cap L(C)$. The next lemma shows that $|X(R_1)| = 1$ and $x(R_1) = x$.

Lemma 10.22

Let (x, y, z) be a feasible solution of the linear programs (10.87) and let the policy R_1 be constructed by the steps 3, 4, 5 and 6 of Algorithm 9.5. Then, $|X(R_1)| = 1$ and $x(R_1) = x$.

Proof

Since $x^l := x(f_l^\infty) = \beta^T P^*(f_l)$ for all deterministic policies f_l^∞, we have

$$x_{ia} = \textstyle\sum_l p_l \cdot x^l_{ia} = \lim_{T\to\infty} \sum_{t=1}^T \sum_j \beta_j \cdot \sum_l p_l \cdot \mathbb{P}_{f_l^\infty}\{X_t = i,\ Y_T = a \mid X_1 = j\},\ (i,a) \in S \times A,$$

where the numbers p_l are determined in step 5 of Algorithm 9.5. From Theorem 1.1 it follows that the policy R_1, constructed in Algorithm 9.5, satisfies $R_1 \in C(M)$ and for all $(i,a) \in S \times A$,

$$\textstyle\sum_j \beta_j \cdot \mathbb{P}_{R_l}\{X_t = i,\ Y_T = a \mid X_1 = j\} = \sum_j \beta_j \cdot \sum_l p_l \cdot \mathbb{P}_{f_l^\infty}\{X_t = i,\ Y_T = a \mid X_1 = j\}.$$

Hence, we obtain

$$\begin{array}{rcl} x_{ia} & = & \lim_{T\to\infty} \sum_{t=1}^T \sum_j \beta_j \cdot \sum_j \mathbb{P}_{R_l}\{X_t = i,\ Y_T = a \mid X_1 = j\} \\ & = & \lim_{T\to\infty} x^T_{ia}(R_1) \ = \ x_{ia}(R_1),\ (i,a) \in S \times A. \end{array}$$

Therefore, we have shown that $x(R_1) \in L(M) \cap L(C)$ and $x(R_1) = x$. □

Theorem 10.41

(1) If (10.86) is infeasible, then $C_0^1 = \emptyset$.

(2) If (x^*, y^*, z^*) and (v^*, t^*, ρ^*, w^*) are optimal solutions of the linear programs (10.86) and (10.87), respectively, then $\sum_i z_i^*$ is the value of the constrained game and R_1^* and $(\rho^*)^\infty$, where R_1^* is such that $x(R_1^*) = x^*$, are optimal policies for player 1 and 2, respectively.

Proof

(1) Suppose that $C_0^1 \neq \emptyset$. Let $R_1 \in C_0^1$ and let $x(R_1)$ be such that $x(R_1) \in X(R_1)$. Then, $x(R_1) = x$ for some $x \in Q$. We also have $b_k \geq c_k(R_1) \geq \sum_{(i,a)} x_{ia}(R_1) c_i^k(a) = \sum_{(i,a)} x_{ia} c_i^k(a)$ for $k = 1, 2, \dots, m$. Hence, (10.86) is feasible, which yields the desired contradiction.

(2) From the complementary slackness property of linear programming, we obtain

$$\sum_{i,b} \{z_i^* - \sum_a r_i(a,b) x_i^*(a)\} \rho^*_{ib} = 0, \text{ i.e } \sum_i z_i^* = \sum_{i,a} r_i(a, \rho^*) x_i^*(a). \tag{10.89}$$

Take any policy $R_1 \in C_0^1$. Since $L = Q$ and $b_k \geq c_k(R_1) \geq \sum_{(i,a)} x_{ia}(R_1) c_i^k(a)$ for $1 \leq k \leq m$ and for any $x(R_1) \in X(R_1)$, there exists vectors y and z such that $\big(x(R_1), y, z\big)$ is a feasible solution of (10.86). Therefore, we may write

$$\begin{array}{rcl} \phi\big(\beta, R_1, (\rho^*)^\infty\big) & = & \liminf_{T\to\infty} \sum_{i,a} x^T_{ia}(R_1) r_i(a, \rho^*) \\ & \leq & \sum_{i,a} x_{ia}(R_1) r_i(a, \rho^*) \\ & = & \sum_{i,a} \{\sum_b r_i(a,b) \rho^*_{ib}\}\, x_{ia}(R_1) \\ & \leq & \sum_{i,a} \big\{v_i^* + \sum_j \{\delta_{ij} - p_{ij}(a)\} t_j^* + \sum_k c_i^k(a) w_k^*\big\}\, x_{ia}(R_1) \end{array}$$

$$\begin{array}{rl} = & \sum_i v_i^* \cdot \sum_a x_{ia}(R_1) + \sum_j \big\{ \sum_{i,a} \{\delta_{ij} - p_{ij}(a)\} x_{ia}(R_1)\big\} t_j^* + \\ & \hfill \sum_k \{\sum_{i,a} c_i^k(a) x_{ia}(R_1)\} w_k^* \\ = & \sum_j v_j^* \cdot \sum_a x_{ja}(R_1) + \sum_k \{\sum_{i,a} c_i^k(a) x_{ia}(R_1)\} w_k^* \\ = & \sum_j v_j^* \cdot \big\{\beta_j - \sum_{i,a} \{\delta_{ij} - p_{ij}(a)\} y_i^*(a)\big\} + \sum_k \{\sum_{i,a} c_i^k(a) x_{ia}(R_1)\} w_k^* \\ = & \sum_j \beta_j v_j^* - \sum_{i,a} \big\{ \sum_j \{\delta_{ij} - p_{ij}(a)\} v_j^* \big\} y_i^*(a) + \sum_k b_k w_k^* \\ \leq & \sum_j \beta_j v_j^* + \sum_k b_k w_k^* = \text{optimum } (10.87) = \text{optimum } (10.86) = \sum_i z_i^*. \end{array}$$

We also have, using the properties that $x(R_1^*) \in L(C)$ and $x(R_1^*) = x^*$ (see Lemma 10.22), and equation (10.89),

$$\begin{array}{rcl} \phi\big(\beta, R_1^*, (\rho^*)^\infty\big) & = & \liminf_{T\to\infty} \sum_{i,a} x_{ia}^T(R_1^*) r_i(a, \rho^*) \\ & = & \sum_{i,a} x_{ia}(R_1^*) r_i(a, \rho^*) = \sum_{i,a} x_{ia}^* r_i(a, \rho^*) = \sum_i z_i^*. \end{array}$$

Let $\rho^\infty \in C^2(S)$ be arbitrarily chosen. Then, we obtain

$$\begin{array}{rcl} \phi(\beta, R_1^*, \rho^\infty) & = & \liminf_{T\to\infty} \sum_{i,a} x_{ia}^T(R_1^*) r_i(a, \rho) \\ & = & \sum_{i,a} x_{ia}(R_1^*) r_i(a, \rho) = \sum_{i,a} x_{ia}^* \sum_b r_i(a, b) \rho_{ib} \\ & \geq & \sum_i z_i^* \sum_b \rho_{ib} = \sum_i z_i^*. \end{array}$$

Hence, we have shown

$$\phi\big(\beta, R_1, (\rho^*)^\infty\big) \leq \textstyle\sum_i z_i^* \leq \phi(\beta, R_1^*, \rho^\infty \text{ for every } R_1 \in C_0^1 \text{ and every } \rho^\infty \in C^2(S).$$

So, we have shown that $\sum_i z_i^*$ is the value of the constrained game and that R_1^* and $(\rho^*)^\infty$ are optimal policies for player 1 and 2, respectively. □

In general, there does not exist a stationary optimal policy for player 1. However, if x^* satisfies $x^* = x(\pi^*)$, where the decision rule π^* is defined by

$$\pi_{ia}^* := \begin{cases} \frac{x_i^*(a)}{x_i^*} & i \in S_{x^*} \\ \frac{y_i^*(a)}{y_i^*} & i \in S_{y^*} \\ \text{arbitrary} & \text{if } i \notin S_{x^*} \cup S_{y^*}, \end{cases} \tag{10.90}$$

where $x_i^* := \sum_a x_i^*(a)$, $y_i^* := \sum_a y_i^*(a)$, $S_{x^*} := \{i \mid x_i^* > 0\}$ and $S_{y^*} := \{i \mid x_i^* = 0,\ y_i^* > 0\}$. Notice that, since $\beta_j = 0$ is allowed for one or more $j \in S$, it is possible that $S_{x^*} \cup S_{y^*} \neq S$. Similarly as in Lemma 9.16, it can be shown that if $x_i^*(a) = \pi_i^* \cdot \{\beta^T P^*(\pi^{)}\}_i$, $(i, a) \in S \times A$, where π^* is defined by (10.90), $(\pi^*)^\infty$ is an optimal policy for player 1. In Lemma 9.17 we have shown that if $\frac{x_i^*(a)}{x_i^*} = \frac{y_i^*(a)}{y_i^*}$ for all $a \in A(i)$ and all i for which $x_i^* > 0$ and $y_i^* > 0$, then x^* satisfies the condition that $x_i^*(a) = \pi_i^* \cdot \{\beta^T P^*(\pi^{)}\}_i$ for all $(i, a) \in S \times A$.

Algorithm 10.17 *Single-controller constrained game with no discounting (multichain case)*

Input: Instance of a two-person single-controller constrained stochastic game.

Output: The value and a pair R_1^* and $(\rho^*)^\infty$ of optimal policies (if the constrained game is feasible).

1. Solve the dual pair of linear programs (10.86) and (10.87).
2. **if** (10.86) is infeasible **then**

   the constrained Markov game does not have a feasible solution (STOP).
3. Let (x^*, y^*, z^*) and (v^*, t^*, ρ^*, w^*) be optimal solutions of program (10.86) and (10.87), respectively. Determine, by the steps 3, 4, 5 and 6 of Algorithm 9.5 a policy R_1^* such that $R_1^* \in L(M) \cap L(C)$ and $x(R_1^*) = x^*$. Then, $\sum_i z_i^*$ is the value of the constrained game and R_1^* and $(\rho^*)^\infty$ are optimal policies for player 1 and 2, respectively (STOP).

Single-controller stochastic game: the unichain case

In the unichain case the stationary matrix $P^*(\pi)$ has identical rows. Hence, for all $\pi^\infty \in C^1(S)$ and all $\rho^\infty \in C^2(S)$, the average reward vector $\phi(\pi^\infty, \rho^\infty)$ has identical components. Therefore, we consider $\phi(\pi^\infty, \rho^\infty)$ as a scalar. Furthermore, we denote the identical rows of $P^*(\pi)$ as $p^*(\pi)$. Instead of the linear programs (10.73) and (10.74) we consider the following dual pair of linear programs:

$$min \left\{ v \;\middle|\; \begin{array}{ccccll} v & + & \sum_j \{\delta_{ij} - p_{ij}(a)\}t_j & - & \sum_b r_i(a,b)\rho_{ib} & \geq 0, \; a \in A(i), \; i \in S \\ & & & & \sum_b \rho_{ib} & = 1, \; i \in S \\ & & & & \rho_{ib} & \geq 0, \; b \in B(i), \; i \in S \end{array} \right\} \tag{10.91}$$

and

$$max \left\{ \sum_i z_i \;\middle|\; \begin{array}{rcrcl} \sum_{(i,a)} \{\delta_{ij} - p_{ij}(a)\} x_i(a) & & & = & 0, \; j \in S \\ \sum_a x_j(a) & & & = & 1, \; j \in S \\ -\sum_a r_i(a,b) x_i(a) & + & z_i & \leq & 0, \; (i,b) \in S \times B \\ & & x_i(a) & \geq & 0, \; (i,a) \in S \times A \end{array} \right\}. \tag{10.92}$$

Theorem 10.42
Let (v^*, t^*, ρ^*) and (x^*, z^*) be optimal solutions of the linear programs (10.91) and (10.92). Define the policy $(\pi^*)^\infty$ by $\pi_{ia}^* := \frac{x_i^*(a)}{\sum_a x_i^*(a)}$, $i \in S_{x^*}$, $a \in A(i)$ and for $i \notin S_{x^*}$ take for π_{ia}^*, $a \in A(i)$, an arbitrary probability vector. Then, v^* is the value and $(\pi^*)^\infty$ and $(\rho^*)^\infty$ are optimal stationary policies for player 1 and 2, respectively.

Proof
Note that it is sufficient to show that

$$\phi\big(\pi^\infty, (\rho^*)^\infty\big) \leq v^* \leq \phi\big((\pi^*)^\infty, \rho^\infty\big) \text{ for all } \pi^\infty \in C^1(S) \text{ and all } \rho^\infty \in C^2(S).$$

From the constraints of program (10.91), we obtain $v^* + \{I - P(\pi)\}t^* \geq r(\pi, \rho^*)$ for all $\pi^\infty \in C^1(S)$. By multiplying this inequality by $P^*(\pi)$, we get $v^* \geq p^*(\pi) r(\pi, \rho^*) = \phi\big(\pi^\infty, (\rho^*)^\infty\big)$ for all $\pi^\infty \in C^1(S)$. From the constraints of program (10.92) it follows that for all $\rho^\infty \in C^2(S)$,

$$v^* = \textstyle\sum_i z_i^* \leq \sum_{i,a} x_i^*(a) r_i(a,\rho) = \sum_{i,a} \pi_{ia}^* \cdot x_i^* \cdot r_i(a,\rho) = \sum_i x_i^* \cdot r_i(\pi^*, \rho).$$

Furthermore, we have for the N-vector x^* with components $x_i^*,\ i \in S,\ (x^*)^T = (x^*)^T P(\pi^*)$ and $(x^*)^T e = 1$, implying $(x^*)^T = (x^*)^T P^*(\pi^*)$ and $(x^*)^T e = 1$. Hence, $x^* = p^*(\pi^*)$. Consequently, $\sum_i x^* \cdot r_i(\pi^*, \rho) = \phi\big((\pi^*)^\infty, \rho^\infty\big)$ for all $\rho \in C^2(S)$. Therefore, we have shown $v^* \le \phi\big((\pi^*)^\infty, \rho^\infty\big)$ for all $\rho^\infty \in C^2(S)$, which completes the proof of the theorem. □

Algorithm 10.18 *Single-controller game with no discounting (unichain case)*
Input: Instance of a two-person single-controller unichain stochastic game.
Output: The value v^* and a pair $(\pi^*)^\infty$ and $(\rho^*)^\infty$ of optimal stationary policies.

1. Compute optimal solutions (v^*, t^*, ρ^*) and (x^*, z^*) of the linear programs (10.91) and (10.92), respectively.

2. Define the stationary policy $(\pi^*)^\infty$ by $\pi^*_{ia} := \begin{cases} \frac{x_i^*(a)}{\sum_a x_i^*(a)},\ i \in S_{x^*},\ a \in A(i) \\ \frac{1}{|A(i)|},\ i \notin S_{x^*},\ a \in A(i) \end{cases}$,

   where $S_{x^*} := \{i \mid \sum_a x_i^*(a) > 0\}$.

3. v^* is the value and $(\pi^*)^\infty$ and $(\rho^*)^\infty$ optimal stationary policies for player 1 and 2, respectively.

Additional constraints

In this subsection we consider the constrained Markov game with the reward function $\phi(\beta, R_1, R_2)$ and the costs functions $c_k(\beta, R_1)$, for which we require that $c_k(\beta, R_1) \le b_k,\ k = 1, 2, \dots, m$, for given numbers $b_k,\ k = 1, 2, \dots, m$. The feasible set of policies for player 1 is the set C_0^1, defined by $C_0^1 := \{R1 \in C^1 \mid c_k(\beta, R_1) \le b_k,\ k = 1, 2, \dots, m\}$. In the multichain case we restricted the set of policies for player 2 to the set $C^2(S)$ of stationary policies. In the unichain case there is no need to this restriction. So, player 2 can choose any policy $R_2 \in C^2$. We consider for the constrained Markov game in the unichain case the following dual pair of linear programs

$$max \left\{ \sum_i z_i \;\middle|\; \begin{array}{rcrcl} \sum_{(i,a)} \{\delta_{ij} - p_{ij}(a)\} x_i(a) & & & = & 0,\ j \in S \\ \sum_{(i,a)} x_i(a)(a) & & & = & 1 \\ -\sum_a r_i(a,b) x_i(a) & + & z_i & \le & 0,\ (i,b) \in S \times B \\ \sum_{(i,a)} c_i^k(a) x_i(a) & & & \le & b_k,\ k = 1, 2, \dots, m \\ & & x_i(a) & \ge & 0,\ (i,a) \in S \times A \end{array} \right\} \tag{10.93}$$

and

$$min \left\{ v_i + \sum_k b_k w_k \;\middle|\; \begin{array}{rl} v + \sum_j \{\delta_{ij} - p_{ij}(a)\} t_j - \sum_b r_i(a,b) \rho_{ib} + \sum_k c_i^k(a) w_k & \ge 0,\ a \in A(i),\ i \in S \\ \sum_b \rho_{ib} & = 1,\ i \in S \\ \rho_{ib} & \ge 0,\ b \in B(i),\ i \in S \\ w_k & \ge 0,\ k = 1, 2, \dots, m \end{array} \right\}. \tag{10.94}$$

Theorem 10.43

(1) If (10.93) is infeasible, then $C_0^1 = \emptyset$*.*

(2) If (x^*, z^*) *and* (v^*, t^*, ρ^*, w^*) *are optimal solutions of the linear programs (10.93) and (10.94), respectively, then* $\sum_i z_i^*$ *is the value of the constrained game and* $(\pi^*)^\infty$ *and* $(\rho^*)^\infty$*, where* $\pi^*_{ia} := \frac{x_i^*(a)}{\sum_a x_i^*(a)}$*,* $i \in S_{x^*}$*,* $a \in A(i)$ *and for* $i \notin S_{x^*}$ *take for* π^*_{ia}*,* $a \in A(i)$*, an arbitrary probability vector, are optimal policies for player 1 and 2, respectively.*

Proof

(1) Suppose that $C_0^1 \neq \emptyset$ and let $R_1 \in C_0^1$. In the unichain case we have, by Theorem 9.24, $L = Q_0$. Hence, $x(R_1) = x$ for some $x \in Q_0$. Furthermore, we can write

$$b_k \geq c_k(R_1) \geq \textstyle\sum_{i,a} x_{ia}(R_1) c_i^k(a) = \sum_{i,a} x_{ia} c_i^k(a) \text{ for } k = 1, 2, \dots, m$$

Therefore, (10.93) is feasible, which yields the desired contradiction.

(2) From the complementary slackness property of linear programming, we obtain

$$b_k \geq c_k(R_1) \geq \textstyle\sum_{i,a} x_{ia}(R_1) c_i^k(a) = \sum_{i,a} x_{ia} c_i^k(a) \text{ for } k = 1, 2, \dots, m$$

$$\textstyle\sum_{i,b} \{z_i^* - \sum_a r_i(a,b) x_i^*(a)\} \rho^*_{ib} = 0, \text{ i.e. } \sum_i z_i^* = \sum_{i,a} r_i(a, \rho^*) x_i^*(a).$$

Take any policy $R_1 \in C_0^1$. Since $L = Q_0$ and $b_k \geq c_k(R_1) \geq \sum_{(i,a)} x_{ia}(R_1) c_i^k(a)$ for $1 \leq k \leq m$ and for any $x(R_1) \in X(R_1)$, there exists a vector z such that $\big(x(R_1), z\big)$ is a feasible solution of (10.93). Therefore, we may write

$$\begin{aligned}
\phi\big(\beta, R_1, (\rho^*)^\infty\big) &= \liminf_{T\to\infty} \textstyle\sum_{i,a} x_{ia}^T(R_1) r_i(a, \rho^*) \\
&\leq \textstyle\sum_{i,a} x_{ia}(R_1) r_i(a, \rho^*) \\
&= \textstyle\sum_{i,a} \{\sum_b r_i(a,b)\rho^*_{ib}\}\, x_{ia}(R_1) \\
&\leq \textstyle\sum_{i,a} \big\{v_i^* + \sum_j \{\delta_{ij} - p_{ij}(a)\} t_j^* + \sum_k c_i^k(a) w_k^*\big\}\, x_{ia}(R_1) \\
&= v^* \cdot \textstyle\sum_{i,a} x_{ia}(R_1) + \sum_j \big\{\sum_{i,a} \{\delta_{ij} - p_{ij}(a)\} x_{ia}(R_1)\big\} t_j^* + \\
&\qquad\qquad \textstyle\sum_k \{\sum_{i,a} c_i^k(a) x_{ia}(R_1)\} w_k^* \\
&= v^* \cdot \textstyle\sum_a x_{ja}(R_1) + \sum_k \{\sum_{i,a} c_i^k(a) x_{ia}(R_1)\} w_k^* \\
&= v^* + \textstyle\sum_k \{\sum_{i,a} c_i^k(a) x_{ia}(R_1)\} w_k^* \\
&= \textstyle\sum_j \beta_j v_j^* - \sum_{i,a} \big\{\sum_j \{\delta_{ij} - p_{ij}(a)\} v_j^*\big\} y_i^*(a) + \sum_k b_k w_k^* \\
&\leq v^* + \textstyle\sum_k b_k w_k^* = \text{optimum } (10.94) = \text{optimum } (10.93) = \sum_i z_i^*.
\end{aligned}$$

Let $\rho^\infty \in C^2(S)$ be arbitrarily chosen. Then, we obtain

$$\begin{aligned}
\phi(\beta, (\pi^*)^\infty, \rho^\infty) &= \beta^T P^*(\pi^*) r(\pi^*, \rho) = p^*(\pi^*) r(\pi^*, \rho) \\
&= \textstyle\sum_{i,a} x^*_{ia} \sum_b r_i(a,b) \rho_{ib} \geq \sum_i z_i^* \sum_b \rho_{ib} = \sum_i z_i^*.
\end{aligned}$$

Hence, we have shown

$$\phi\big(\beta, R_1, (\rho^*)^\infty\big) \leq \textstyle\sum_i z_i^* \leq \phi\big(\beta, (\pi^*)^\infty, \rho^\infty\big) \text{ for every } R_1 \in C_0^1 \text{ and every } \rho^\infty \in C^2(S).$$

If player 1 uses the stationary policy $(\pi^*)^\infty$ the Markov game becomes an MDP. Therefore, $\inf_{\rho^\infty \in C^2(S)} \phi\big(\beta, (\pi^*)^\infty, \rho^\infty\big) = \inf_{R_2 \in C^2(S)} \phi\big(\beta, (\pi^*)^\infty, R_2\big)$. Consequently,

$$\phi\big(\beta, R_1, (\rho^*)^\infty\big) \leq \textstyle\sum_i z_i^* \leq \phi\big(\beta, (\pi^*)^\infty, R_2\big) \text{ for every } R_1 \in C_0^1 \text{ and every } R_2 \in C^2,$$

implying that $\sum_i z_i^*$ is the value of the constrained game and $(\pi^*)^\infty$ and $(\rho^*)^\infty$ are optimal policies for player 1 and 2, respectively. □

Algorithm 10.19 *Single-controller constrained game with no discounting (unichain case)*
Input: Instance of a two-person single-controller constrained unichained stochastic game.
Output: The value and a pair $(\pi^*)^\infty$ and $(\rho^*)^\infty$ of optimal policies (if the constrained game is feasible).

1. Solve the dual pair of linear programs (10.93) and (10.94).
2. **if** (10.93) is infeasible **then**

   the constrained Markov game does not have a feasible solution (STOP).
3. Let (x^*, z^*) and (v^*, t^*, ρ^*, w^*) be optimal solutions of program (10.93) and (10.94), respectively.
4. Define the stationary policy $(\pi^*)^\infty$ by $\pi^*_{ia} := \begin{cases} \frac{x^*_i(a)}{\sum_a x^*_i(a)}, \ i \in S_{x^*}, \ a \in A(i) \\ \frac{1}{|A(i)|}, \ i \notin S_{x^*}, \ a \in A(i) \end{cases}$, where $S_{x^*} := \{i \mid \sum_a x^*_i(a) > 0\}$.
5. v^* is the value and $(\pi^*)^\infty$ and $(\rho^*)^\infty$ optimal stationary policies for player 1 and 2, respectively (STOP).

Switching-controller stochastic game

The model and the notation is the same as in the discounted case. If player 1 uses a stationary policy π^∞, the stochastic game reduces to an MDP; similarly, if player 2 uses a stationary policy ρ^∞, the the stochastic game becomes an MDP. Therefore, the next Lemma holds.

Lemma 10.23
(1) $\inf_{R_2} \phi(\pi^\infty, R_2) = \min_{\rho^\infty \in C^2(S)} \phi(\pi^\infty, \rho^\infty) = \min_{g^\infty \in C^2(D)} \phi(\pi^\infty, g^\infty)$.
(2) $\sup_{R_1} \phi(R_1, \rho^\infty) = \max_{\pi^\infty \in C^1(S)} \phi(\pi^\infty, \rho^\infty) = \max_{f^\infty \in C^1(D)} \phi(f^\infty, \rho^\infty)$.

The following lemma presents a result for the single-controller stochastic game, i.e. $S = S_1$.

Lemma 10.24
Assume that $S = S_1$. Let (v^*, t^*, ρ^*) be an optimal solution of the linear program (10.73). Let R denote the set of states i for which player 1 has an optimal stationary policy π^∞ such that state i is recurrent in the Markov chain $P(\pi)$. Let $A^*(i) := \{a \in A(i) \mid v^*_i = \sum_j p_{ij}(a)v^*_j\}$, $i \in S$. Then,

(1) $v^*_i = max_{a \in A(i)} \sum_j p_{ij}(a) v^*_j$ *for all* $i \in S$.

(2) *If* t^* *satisfies* $v^*_i + t^*_i \geq val_{A^*(i) \times B(i)} \{r_i(a,b) + \sum_j p_{ij}(a)t^*_j\}$ *for all* $i \in S$, *then* $v^*_i + t^*_i = val_{A^*(i) \times B(i)} \{r_i(a,b) + \sum_j p_{ij}(a)t^*_j\}$ *for all* $i \in R$.

Proof

(1) From the constraints of (10.73) we obtain $v_i^* \geq max_{a \in A(i)} \sum_j p_{ij}(a)v_j^*,\ i \in S$. In Theorem 10.40 it is shown that $v_i^* = \sum_j p_{ij}(\pi^*)v_j^*,\ i \in S$ for an optimal policy $(\pi^*)^\infty$ for player 1. Hence, we have shown the equalities $v_i^* = max_{a \in A(i)} \sum_j p_{ij}(a)v_j^*,\ i \in S$.

(2) By part (1), $A^*(i) \neq \emptyset,\ i \in S$. Suppose that $v_k^* + t_k^* > val_{A^*(k) \times B(k)} \{r_k(a,b) + \sum_j p_{kj}(a)t_j^*\}$ for some $ki \in R$. Let π^∞ be an optimal stationary policy for player 1 such that k is recurrent with respect to $P(\pi)$, and let $R(\pi)$ be the ergodic set to which k belongs. We denote the parts of the matrix $P(\pi)$ that belong to $R(\pi)$ by $\hat{P}(\pi)$; similarly, for other matrices and vectors. Since $P^*(\pi) \geq 0,\ v^* \geq P(\pi)v^*,\ P^*(\pi)v^* = P^*(\pi)P(\pi)v^*$ and $p_{ii}(\pi) > 0$ for all $i \in R(\pi)$, we have $\hat{v}^* = \hat{P}(\pi)\hat{v}^*$. Hence, by the constraints of (10.73), we have: if $i \in R(\pi)$ and $\pi_{ia} > 0$, then $a \in A^*(i)$. Consequently, in the states of $R(\pi)$ is π a feasible strategy for the matrix game with elements $r_i(a,b) + \sum_j p_{ij}(a)t_j^*$ with $(a,b) \in A^*(i) \times B(i)$. Let ρ be an optimal strategy for player 2 in this matrix games. Then,

$$v_i^* + t_i^* \geq val_{A^*(i) \times B(i)} \{r_i(a,b) + \textstyle\sum_j p_{ij}(a)t_j^*\} \geq \{r(\pi,\rho) + P(\pi)t^*\}_i$$

and

$$v_k^* + t_k^* > val_{A^*(k) \times B(k)} \{r_k(a,b) + \textstyle\sum_j p_{kj}(a)t_j^*\} \geq \{r(\pi,\rho) + P(\pi)t^*\}_k.$$

Hence, in vector notation, $\hat{v}^* > \hat{r}(\pi,\rho) + \hat{P}(\pi)\hat{t}^* - \hat{t}^*$. Since the elements of $\hat{P}(\pi)^*$ are strictly positive, we obtain

$$\hat{v}^* = \hat{P}(\pi)^*\hat{v}^* > \hat{P}(\pi)^*\hat{r}(\pi,\rho) + \hat{P}(\pi)^*\hat{P}(\pi)\hat{t}^* - \hat{P}(\pi)^*\hat{t}^* = \hat{P}(\pi)^*\hat{r}(\pi,\rho) = \hat{\phi}(\pi^\infty,\rho^\infty).$$

Therefore, π^∞ cannot be an optimal policy for player 1, which gives a contradiction. □

Next, we consider the stochastic game with rewards $r_i^*(a,b)$, where $r_i^*(a,b) := r_i(a,b) - v_i^*$ for all i,a,b, and with the total rewards as optimality criterion. The dual pair of linear programs, i.e. (10.31) and (10.32) with $\alpha = 1$, becomes

$$min \left\{ \sum_i u_i \;\middle|\; \begin{array}{rcrl} \sum_j \{\delta_{ij} - p_{ij}(a)\}u_j & - & \sum_b r_i^*(a,b)\rho_{ib} & \geq 0,\ a \in A(i),\ i \in S \\ & & \sum_b \rho_{ib} & = 1,\ i \in S \\ & & \rho_{ib} & \geq 0,\ b \in B(i),\ i \in S \end{array} \right\} \tag{10.95}$$

and

$$max \left\{ \sum_i w_i \;\middle|\; \begin{array}{rcrcl} \sum_{(i,a)} \{\delta_{ij} - p_{ij}(a)\}y_i(a) & & & = & 1,\ j \in S \\ -\sum_a r_i^*(a,b)y_i(a) & + & w_i & \leq & 0,\ (i,b) \in S \times B \\ & & y_i(a) & \geq & 0,\ (i,a) \in S \times A \end{array} \right\}. \tag{10.96}$$

A single-controller game is *semi-transient* if:

(1) $\sum_j p_{ij}(a) \leq 1$ for all $(i,a) \in S \times A$;

(2) There exists a stationary policy π^∞ for player 1 such that the Markov chain $P(\pi)$ is transient;

(3) The value vector of the game with the average reward criterion is the zero-vector.

Lemma 10.25

Assume that the single-controller stochastic game with payoffs $r_i^*(a,b)$ is semi-transient. Then,

(1) *The linear programs (10.95) and (10.96) have finite optimal solutions.*

(2) *If (u^*, ρ^∞) is an optimal solution of program (eq-10.95), then*
$u_i^* = val_{A(i)\times B(i)}\,\{r_i^*(a,b) + \sum_j p_{ij}(a)u_j^*\}$ *for all* $i \in S$.

Proof

(1) From the theory of linear programming it follows that it is sufficient to show that (10.95) and (10.96) have feasible solutions. Since the game is semi-transient, there are stationary optimal policies $(\pi^*)^\infty$ and $(\rho^*)^\infty$ with $\phi\big((\pi^*)^\infty,(\rho^*)^\infty\big) = 0$. Hence, we have

$$\phi\big(\pi^\infty,(\rho^*)^\infty\big) \le \phi\big((\pi^*)^\infty,(\rho^*)^\infty\big) = 0 \le \phi\big((\pi^*)^\infty,\rho^\infty\big) \text{ for all } \pi^\infty \text{ and } \rho^\infty.$$

Given the stationary policy $(\rho^*)^\infty$ for player 2, the game becomes an MDP for player 1 with value vector 0. The linear program to compute the value vector of this MDP is

$$min\left\{\sum_i g_i \;\middle|\; \begin{array}{rrcrl} & \sum_j\{\delta_{ij}-p_{ij}(a)\}g_j & \ge & 0, & (i,a)\in S\times A \\ g_i & \sum_j\{\delta_{ij}-p_{ij}(a)\}h_j & \ge & r_i^*,\rho^*), & (i,a)\in S\times A \end{array}\right\} \tag{10.97}$$

and has optimal solution $(g^* = 0, h^*)$. Hence, (u^*,ρ^*) with $u^* := h^*$ is feasible for (10.95). Let π^∞ be a stationary policy such that the Markov chain $P(\pi)$ is transient (by the assumption of the lemma π^∞ exists). Since π^∞ is transient, $y_i(a) := \{e^T\big(I-P(\pi)\big)^{-1}\}_i\cdot\pi_{ia},\ (i,a)\in S\times A$ is well-defined and $y_i(a)\ge 0,\ (i,a)\in S\times A$. Let $q^T := e^T\{I-P(\pi)\}^{-1}$. Then, we can write

$$\begin{aligned}\textstyle\sum_{(i,a)}\{\delta_{ij}-p_{ij}(a)\}y_i(a) &= q_j - \textstyle\sum_i p_{ij}(\pi)q_i \\ &= \{q^T - q^TP(\pi)\}_j = \{q^T\big(I-P(\pi)\big)\}_j \\ &= \{e^T\big(I-P(\pi)\big)^{-1}\big(I-P(\pi)\big)\}_j = 1,\ j\in S.\end{aligned}$$

Take $w_i := min_{(i,b)\in S\times B}\sum_a r_i^*(a,b)y_i(a)$. Then, (y,w) is feasible for (10.96).

(2) Let (u^*,ρ^*) and (y^*,w^*) be optimal solutions for (10.95) and (10.96), respectively. From the constraints of (10.95) we obtain $u^* \ge r^*(\pi,\rho^*) + P(\pi)u^*$ for all policies π^∞ of player 1.
By the complementary slackness property of linear programming we obtain

$$y_i^*(a)\cdot\Big\{\sum_j\{\delta_{ij}-p_{ij}(a)\}u_j^* - \sum_b r_i^*(a,\rho^*)\Big\} = 0 \text{ for all } (i,a)\in S\times A. \tag{10.98}$$

Take $\pi_{ia}^* := \frac{y_i^*(a)}{\sum_a y_i^*(a)},\ (i,a)\in S\times A$. Since $\sum_a y_i^*(a) = 1 + \sum_{(i,a)} y_i^*(a) \ge 1,\ j\in S$, the policy $(\pi^*)^\infty$ is well-defined. Since $\pi_{ia}^* > 0$ if and only if $y_i^*(a) > 0$, (10.98) implies

$$\pi_{ia}^*\cdot\big\{\textstyle\sum_j\{\delta_{ij}-p_{ij}(a)\}u_j^* - \sum_b r_i^*(a,\rho^*)\big\} = 0 \text{ for all } (i,a)\in S\times A.$$

Hence, $\sum_a \pi_{ia}^*\cdot\big\{\sum_j\{\delta_{ij}-p_{ij}(a)\}u_j^* - \sum_b r_i^*(a,\rho^*)\big\} = 0$ for all $i\in S$ and consequently,

$$u^* = r^*(\pi^*,\rho^*) + P(\pi^*)u^*. \tag{10.99}$$

From the constraints of (10.96) it also follows that $(y^*)^T = e^T + (y^*)^TP(\pi^*)$, where y^* is a vector with components $y_i^* := \sum_a y_i^*(a),\ i\in S$. Therefore, we have

$$(y^*)^T = \textstyle\sum_{t=1}^n e^TP^{t-1}(\pi^*) + (y^*)^TP^n(\pi^*) \ge \sum_{t=1}^n e^TP^{t-1}(\pi^*) \text{ for all } n\in\mathbb{N}.$$

Hence, $\sum_{t=1}^{\infty} e^TP^{t-1}(\pi^*) \le (y^*)^T$, implying $P(\pi^*)$ is a transient Markov chain and $I - P(\pi^*)$ is nonsingular. Therefore, (10.99) implies $u^* = \{I - P(\pi^*)\}^{-1}r^*(\pi^*, \rho^*) = v\big((\pi^*)^\infty, (\rho^*)^\infty\big)$, the total rewards, and $(y^*)^T = e^T\{I - P(\pi^*)\}^{-1}$. Since the optimum values of (10.95) and (10.96) are equal, we also can write

$$e^Tv\big((\pi^*)^\infty, (\rho^*)^\infty\big) = e^Tu^* = e^Tw^* \le \textstyle\sum_{(i,a)} r_i^*(a, \rho)y_i^*(a) \text{ for all policies } \rho^\infty \text{ of player 2.}$$

Hence, for all policies ρ^∞ of player 2,

$$\begin{aligned} e^Tv\big((\pi^*)^\infty, (\rho^*)^\infty\big) &\le \textstyle\sum_{(i,a)} r_i^*(a, \rho)y_i^*(a) = \sum_i r_i^*(\pi^*, \rho)y_i^* \\ &= e^T\{I - P(\pi^*)\}^{-1}r^*(\pi^*, \rho) = e^Tv\big((\pi^*)^\infty, \rho^\infty\big). \end{aligned}$$

Therefore, $(\rho^*)^\infty$ is an optimal policy in the MDP, with player 2 as decision maker, which is induced by the transient policy $(\pi^*)^\infty$ for player 1 and in which the total rewards are involved, i.e. $v\big((\pi^*)^\infty, (\rho^*)^\infty\big) \le v\big((\pi^*)^\infty, \rho^\infty\big)$ for every ρ^∞. Since $u^* = v\big((\pi^*)^\infty, (\rho^*)^\infty\big)$, we have

$$u^* \le v\big((\pi^*)^\infty, \rho^\infty\big) = r^*(\pi^*, \rho^*) + P(\pi^*)v\big((\pi^*)^\infty, \rho^\infty\big)u^* \text{ for every } \rho^\infty.$$

Hence, we have shown $r^*(\pi, \rho^*) + P(\pi)u^* \le u^* \le r^*(\pi^*, \rho) + P(\pi^*)u^*$ for every π^∞ and ρ^∞, i.e. $u_i^* = val_{A(i)\times B(i)} \{r_i^*(a, b) + \sum_j p_{ij}(a)u_j^*\}$ for all $i \in S$. □

Note

The proof of the above lemma is related to the problem to determine an optimal transient policy in an MDP with the total reward criterion, which is considered in Section 3.3 of [148].

We shall state a finite algorithm for the switching-controller stochastic game. If we fix for player 2 a stationary policy $(\rho^2)^\infty$ on the states S_2, the game becomes a single-controller stochastic game in which player 1 controls the transitions on S_1 and the transitions on S_2 follow the Markov chain induced by the policy $(\rho^2)^\infty$. If we denote this game by $(\overline{S}, \overline{A}, \overline{B}, \overline{p}, \overline{r})$, then:

$$\overline{S} := S;\ \overline{A}(i) := A(i),\ i \in \overline{S};\ \overline{B}(i) := \begin{cases} B(i), & i \in S_1 \\ \{1\}, & i \in S_2 \end{cases} :\ \overline{p}_{ij}(a) := \begin{cases} p_{ij}(a), & i \in S_1,\ j \in \overline{S},\ a \in \overline{A}(i) \\ p_{ij}(\rho^2), & i \in S_2,\ j \in \overline{S},\ a \in \overline{A}(i) \end{cases};$$

$$\overline{r}_i(a, b) := \begin{cases} r_i(a, b), & i \in S_1,\ a \in \overline{A}(i),\ b \in \overline{B}(i) \\ r_i(a, \rho^2), & i \in S_2,\ a \in \overline{A}(i),\ b \in \overline{B}(i) \end{cases}.$$

Denote the primal linear program for this single-controller stochastic game by $LP_1(\rho^2)$.

Fix a subset $S_0 \subseteq S$, vectors g and h, a stationary policy $(\rho^2)^\infty$ for player 2 on S_2 and for each $i \in S_0$ a nonempty action set $\hat{A}(i) \in \overline{A}(i)$. For these fixed choices, we define a single-controller stochastic game $(\hat{S}, \hat{A}, \hat{B}, \hat{p}, \hat{r})$ by:

$\hat{S} = \hat{S}_1 \cup \hat{S}_2$, where $\hat{S} := S_0,\ \hat{S}_1 := S_0 \cap S_1,\ \hat{S}_2 := S_0 \cap S_2;\ \hat{A}(i) := \overline{A}(i),\ i \in \hat{S}$.

$$\hat{B}(i) := \begin{cases} B(i), & i \in \hat{S}_1 \\ \{1\}, & i \in \hat{S}_2 \end{cases} :\ \hat{p}_{ij}(a) := \begin{cases} p_{ij}(a), & i \in \hat{S}_1,\ j \in \hat{S},\ a \in \hat{A}(i) \\ p_{ij}(\rho^2), & i \in \hat{S}_2,\ j \in \hat{S},\ a \in \hat{A}(i) \end{cases};$$

$$\hat{r}_i(a, b) := \begin{cases} r_i(a, b) - g_i + \sum_{j \in S\backslash S_0} p_{ij}(a)h_j, & i \in \hat{S}_1,\ a \in \hat{A}(i),\ b \in \hat{B}(i) \\ r_i(a, \rho^2) - g_i + \sum_{j \in S\backslash S_0} p_{ij}(\rho^2)h_j, & i \in \hat{S}_2,\ a \in \hat{A}(i),\ b \in \hat{B}(i) \end{cases}.$$

Denote the primal linear program for this single-controller stochastic game by $LP_2(S_0, g, h, \rho^2, \hat{A})$.

Algorithm 10.20 *Switching-controller game with no discounting*

Input: Instance of a two-person switching-controller constrained stochastic game.

Output: The value vector ϕ^* and a pair $(\pi^*)^\infty$ and $(\rho^*)^\infty$ of optimal policies.

```
1. (a) Set $n := 0;\ M := max_{i,a,b}\,|r_i(a,b)|;\ g(n) := M \cdot e;\ h(n) := 0 \cdot e;\ S(n) := \emptyset$.
   (b) for all $i \in S_2$ do
           determine an extreme optimal strategy $\rho_i^2(n)$ for player 2 in the matrix game
           $M_i^2(n)$ with elements $r_i(a,b),\ (a,b) \in A(i) \times B(i)$.

2. (a) for all $i \in S_1$ do $A_i(n+1) := \{a \in A(i) \mid g_i(n) = \sum_j p_{ij}(a) g_j(n)\}$.
   (b) for all $i \in S_2$ do $B_i(n+1) := \{b \in B(i) \mid \sum_j p_{ij}(b) g_j(n) = min_{\overline{b}} \sum_j p_{ij}(\overline{b}) g_j(n)$.

3. for all $i \in S_2$ do
   begin
   determine an extreme optimal strategy $\rho_i^2(n+1)$ for player 2 in the matrix game $M_i^2(n+1)$
   with elements $r_i(a,b) + \sum_j p_{ij}(b) h_j(n),\ (a,b) \in A(i) \times B_i(n+1)$;
   if $car\big(\rho_i^2(n+1)\big) \subseteq B_i(n+1)$ and $g_i(n) + h_i(n) = val\big(M_i^2(n+1)\big)$ then $\rho_i^2(n+1) := \rho^2(n)$
   end

4. (a) determine an optimal solution $(v,t)$ of the linear program $LP_1\big(\rho^2(n+1)\big)$.
   (b) $g(n+1) := v;\ v(n+1) := t$.
   (c) if $g(n+1) \neq g(n)$ then
           begin $h(n+1) := v(n+1);\ S(n+1) := \emptyset;\ n := n+1$; return to step 2 end
       else go to step 5.

5. (a) for all $i \in S_1$ do determine $val\big(M_i^1(n+1)\big)$, where $M_i^1(n+1)$ is the matrix game
       with elements $r_i(a,b) + \sum_j p_{ij}(a) h_j(n),\ (a,b) \in A_i(n+1) \times B(i)$.
   (b) $G_1(n+1) := \{i \in S_1 \mid g_i(n) + h_i(n) > val\big(M_i^1(n)\big)\}$.
   (c) $G_2(n+1) := \{i \in S_2 \mid g_i(n) + h_i(n) > val\big(M_i^2(n)\big)\}$.
   (d) $G(n+1) := G_1(n+1) \cup G_2(n+1)$.
   (e) if $G(n+1) = \emptyset$ then go to step 7
       else $S(n+1) := S(n) \cup G(n+1)$.

6. (a) $A_i(n+1) := A(i),\ i \in S_1 \cap S(n+1)$.
   (b) determine an optimal solution $t$ of $LP_2\big(S(n+1), g(n+1), h(n), \rho^2(n+1), A(n+1)\big)$.
   (c) $u(n+1) := t;\ h_i(n+1) := \begin{cases} h_i(n) & \text{if } i \notin S(n+1) \\ h_i(n+1) & \text{if } i \in S(n+1) \end{cases};\ n := n+1$.
   (d) return to step 2.
```

7. (a) **for all** $i \in S_1$ **do**

   determine $\pi^*_{ia},\ a \in A_i(n+1)$ and $\rho^*_{ib},\ b \in B(i)$, as optimal strategies for player 1 and 2, respectively, in the matrix game $M^1_i(n+1)$ on $A_i(n+1) \times B(i)$, where $A_i(n+1) := \{a \in A(i) \mid g_i(n) = \sum_j p_{ij}(a)g_j(n)\}$.

   (b) **for all** $i \in S_2$ **do**

   determine $\pi^*_{ia},\ a \in A(i)$ and $\rho^*_{ib},\ b \in B_i(n+1)$, as optimal strategies for player 1 and 2, respectively, in the matrix game $M^2_i(n+1)$ on $A(i) \times B_i(n+1)$, where $B_i(n+1) := \{b \in B(i) \mid \sum_j p_{ij}(b)g_j(n) = min_{\overline{b}} \sum_j p_{ij}(\overline{b})g_j(n)$.

8. $\phi^* := g(n)$ is the value vector of the game and $(\pi^*)^\infty$ and $(\rho^*)^\infty$ are optimal policies of player 1 and 2, respectively (STOP).

Example 10.6 (continued)

Start

1. (a) $n := 0;\ M := 7;\ g(0) := (7,7);\ h(0) := (0,0);\ S(0) := \emptyset$.

   (b) $i = 2$: $M^2_2(0) = \begin{pmatrix}4 & 6\\ 7 & 5\end{pmatrix}$ with $val\big(M^2_2(0)\big) = 5.5$ and $\rho^2_2(0) = (0.25, 0.75)$.

Iteration 1

2. (a) $i = 1$: $A_1(1) := \{1,2\}$.

   (b) $i = 2$: $B_2(1) := \{1,2\}$.

3. $i = 2$: $M^2_2(1) = \begin{pmatrix}4 & 6\\ 7 & 5\end{pmatrix}$ with $val\big(M^2_2(1)\big) = 5.5$ and $\rho^2_2(1) = (0.25, 0.75)$.

   $car\big(\rho^2_2(1)\big) \subseteq B_2(1)$ and $g_2(0) + h_2(0) = 7 \neq 5.5 = val\big(M^2_2(1)\big)$.

4. (a) The linear program $LP_1\big(\rho^2_2(1)\big)$ becomes:

$$min\left\{ v_1 + v_2 \;\middle|\; \begin{array}{rcrcrcrcrcrcl} v_1 & - & v_2 & & & & & & & & & \geq & 0\\ -\frac{1}{4}v_1 & + & \frac{1}{4}v_2 & & & & & & & & & \geq & 0\\ v_1 & & & + & t_1 & - & t_2 & - & 3\rho_{11} & - & \rho_{12} & \geq & 0\\ v_1 & & & & & & & - & \rho_{11} & - & 4\rho_{12} & \geq & 0\\ & & v_2 & - & \frac{1}{4}t_1 & + & \frac{1}{4}t_2 & & & & & \geq & \frac{25}{4}\\ & & & & & & & & \rho_{11} & + & \rho_{12} & = & 1\\ & & & & & & & & & & \rho_{11},\ \rho_{12} & \geq & 0 \end{array} \right\}$$

   with optimal solution $v_1 = v_2 = \frac{26}{5};\ t_1 = 0,\ t_2 = \frac{21}{5};\ \rho_{11} = 0,\ \rho_{12} = 1$.

   (b) $g(1) := (\frac{26}{5}, \frac{26}{5});\ v(1) := (0, \frac{21}{5})$.

   (c) $g(1) \neq g(0)$: $h(1) := (0, \frac{21}{5});\ S(1) := \emptyset;\ n := 1$.

Iteration 2

2. (a) $i = 1$: $A_1(2) := \{1,2\}$.

   (b) $i = 2$: $B_2(2) := \{1,2\}$.

3. $i = 2$: $M^2_2(2) = \begin{pmatrix}4 & 51/5\\ 7 & 46/5\end{pmatrix}$ with $val\big(M^2_2(2)\big) = 7$ and $\rho^2_2(2) = (1,0)$.

   $car\big(\rho^2_2(2)\big) = \{1\} \subseteq B_2(2)$ and $g_2(1) + h_2(1) = 9.4 \neq 7 = val\big(M^2_2(2)\big)$.

4. (a) The linear program $LP_1\big(\rho_2^2(2)\big)$ becomes:

$$min\left\{ v_1+v_2 \;\middle|\; \begin{array}{rcrcrcrcrcrcl} v_1 & - & v_2 & & & & & & & & & \geq & 0 \\ -v_1 & + & v_2 & & & & & & & & & \geq & 0 \\ v_1 & & & + & t_1 & - & t_2 & - & 3\rho_{11} & - & \rho_{12} & \geq & 0 \\ v_1 & & & & & & & - & \rho_{11} & - & 4\rho_{12} & \geq & 0 \\ & & v_2 & - & t_1 & + & t_2 & & & & & \geq & 6 \\ & & & & & & & & \rho_{11} & + & \rho_{12} & = & 1 \\ & & & & & & & & & & \rho_{11},\ \rho_{12} & \geq & 0 \end{array} \right\}$$

with optimal solution $v_1 = v_2 = \frac{29}{8};\ t_1 = 0,\ t_2 = \frac{19}{8};\ \rho_{11} = \frac{1}{8},\ \rho_{12} = \frac{7}{8}$.

(b) $g(2) := (\frac{29}{8}, \frac{29}{8});\ v(2) := (0, \frac{19}{8})$.

(c) $g(2) \neq g(1)$: $h(2) := (0, \frac{19}{8});\ S(2) := \emptyset;\ n := 2$.

Iteration 3

2. (a) $i = 1$: $A_1(3) := \{1, 2\}$.

(b) $i = 2$: $B_2(3) := \{1, 2\}$.

3. $i = 2$: $M_2^2(3) = \left(\begin{smallmatrix}4 & 67/8 \\ 7 & 59/8\end{smallmatrix}\right)$ with $val\big(M_2^2(3)\big) = 7$ and $\rho_2^2(3) = (1, 0)$.

$car\big(\rho_2^2(3)\big) = \{1\} \subseteq B_2(3)$ and $g_2(2) + h_2(2) = 6 \neq 7 = val\big(M_2^2(3)\big)$.

4. (a) The linear program $LP_1\big(\rho_2^2(3)\big)$ becomes:

$$min\left\{ v_1+v_2 \;\middle|\; \begin{array}{rcrcrcrcrcrcl} v_1 & - & v_2 & & & & & & & & & \geq & 0 \\ -v_1 & + & v_2 & & & & & & & & & \geq & 0 \\ v_1 & & & + & t_1 & - & t_2 & - & 3\rho_{11} & - & \rho_{12} & \geq & 0 \\ v_1 & & & & & & & - & \rho_{11} & - & 4\rho_{12} & \geq & 0 \\ & & v_2 & - & t_1 & + & t_2 & & & & & \geq & 6 \\ & & & & & & & & \rho_{11} & + & \rho_{12} & = & 1 \\ & & & & & & & & & & \rho_{11},\ \rho_{12} & \geq & 0 \end{array} \right\}$$

with optimal solution $v_1 = v_2 = \frac{29}{8};\ t_1 = 0,\ t_2 = \frac{19}{8};\ \rho_{11} = \frac{1}{8},\ \rho_{12} = \frac{7}{8}$.

(b) $g(3) := (\frac{29}{8}, \frac{29}{8});\ v(3) := (0, \frac{19}{8})$.

(c) $g(3) = g(2)$.

5. (a) $i = 1$: $M_1^1(3) = \left(\begin{smallmatrix}43/8 & 27/8 \\ 1 & 4\end{smallmatrix}\right)$ with $val\big(M_1^1(3)\big) = 29/8$.

(b) $g_1(2) + h_1(2) = 29/8 = val\big(M_1^1(3)\big) \ \rightarrow\ G_1(3) = \emptyset$.

(c) $g_2(2) + h_2(2) = 6 < 7 = val\big(M_2^2(3)\big) \ \rightarrow\ G_(3) = \emptyset$.

(d) $G_3 = \emptyset$.

7. (a) $i = 1$: $M_1^1(3) = \left(\begin{smallmatrix}43/8 & 27/8 \\ 1 & 4\end{smallmatrix}\right)$ with $val\big(M_1^1(3)\big) = 29/8$ and optimal strategies $\pi_{11}^* = 0.6$, $\pi_{12}^* = 0.4;\ \rho_{11}^* = 0.125,\ \rho_{12}^* = 0.875$.

(b) $i = 2$: $M_2^2(3) = \left(\begin{smallmatrix}4 & 67/8 \\ 7 & 59/8\end{smallmatrix}\right)$ with $val\big(M_2^2(3)\big) = 7$ and optimal strategies $\pi_{21}^* = 0,\ \pi_{22}^* = 1;$ $\rho_{21}^* = 1,\ \rho_{22}^* = 0$.

8. $\phi^* = (29/8, 29/8)$ is the value vector and $(\pi^*)^\infty$ and $(\rho^*)^\infty$ are optimal policies of player 1 and 2, respectively.

In proving that Algorithm 10.20 terminates with optimal policies, we shall show that in each iteration the following 8 properties are valid, where we take $g(-1) := (M+1) \cdot e$.

(1) $g_i(n) \geq \sum_j p_{ij}(a)g_j(n)$ for all $(i,a) \in S_1 \times A$.

(2) $g_i(n) \geq \sum_j p_{ij}\big(\rho^2(n)\big)g_j(n)$ for all $i \in S_2$.

(3) $g_i(n) + h_i(n) \geq val\big(M_i^1(n+1)\big)$ for all $i \in S_1$.

(4) $g_i(n) + h_i(n) \geq r_i\big(a, \rho^2(n)\big) + \sum_j p_{ij}\big(\rho^2(n)\big)h_j(n)$ for all $(i,a) \in S_2 \times A$.

(5) $g(n) \leq g(n-1)$.

(6) If $g(n) = g(n-1)$, then $R\big(\rho^2(n)\big) \subseteq R\big(\rho^2(n-1)\big)$ and $\rho_i^2(n) = \rho_i^2(n-1)$ for all $i \in R\big(\rho^2(n)\big) \cap S_2$, where $R(\rho)$ is the set of states k, in the single-controller game induced by ρ, for which player 1 has an optimal policy π^∞ such that state k is recurrent in the Markov chain $P(\pi)$.

(7) $S(n) \cap R\big(\rho^2(n)\big) = \emptyset$.

(8) If $g(n) = g(n-1)$ and $G(n) \neq \emptyset$, then $h(n) < h(n-1)$.

It is easy to verify that the 8 properties hold for $n = 0$. For the proof of the induction step we need several lemmas.

Lemma 10.26

Suppose that $g_i(n) = min_{\overline{b}} \sum_j p_{ij}(\overline{b})g_j(n)$ for all $i \in S_2$. Then,

(a) $car\big(\rho^2(n)\big) \subseteq B_i(n+1)$.

(b) *If property (4) holds, then $g_i(n) + h_i(n) \geq r_i\big(a, \rho^2(n+1)\big) + \sum_j p_{ij}\big(\rho^2(n+1)\big)h_j(n)$ for all $(i,a) \in S_2 \times A$.*

Proof

(a) By step 4 of Algorithm 10.20, $\big(g(n), v(n)\big)$ is an optimal solution of program $LP_1\big(\rho^2(n)\big)$. Hence, $g_i(n) \geq \sum_j p_{ij}\big(\rho^2(n)\big)g_j(n)$ for all $i \in S_2$. By the assumption of the lemma, we have $g_i(n) \leq \sum_j p_{ij}(b)g_j(n)$ for all $(i,b) \in S_2 \times B$. Therefore, for all $b \in car\big(\rho^2(n)\big)$, we obtain $min_{\overline{b}} \sum_j p_{ij}(\overline{b})g_j(n) = g_i(n) = \sum_j p_{ij}(b)g_j(n)$, i.e. $b \in B_i(n+1)$.

(b) Take any $i \in S_2$. Then, by property (4), $g_i(n)+h_i(n) \geq max_a \{r_i\big(a, \rho^2(n)\big)+\sum_j p_{ij}\big(\rho^2(n)\big)h_j(n)\}$. Since $car\big(\rho^2(n)\big) \subseteq B_i(n+1)$, the strategy $\rho_i^2(n)$ is feasible for player 2 in the matrix game $M_i^2(n+1)$ on $A(i) \times B_i(n+1)$. Therefore,

$$g_i(n) + h_i(n) \geq max_a \{r_i\big(a, \rho^2(n)\big) + \textstyle\sum_j p_{ij}\big(\rho^2(n)\big)h_j(n)\} \geq val\big(M_i^2(n+1)\big).$$

Because the strategy $\rho_i^2(n+1)$ is optimal for player 2 in the matrix game $M_i^2(n+1)$, we have

$$val\big(M_i^2(n+1)\big) = max_a \{r_i\big(a, \rho^2(n+1)\big) + \textstyle\sum_j p_{ij}\big(\rho^2(n+1)\big)h_j(n)\}.$$

Hence, we have $g_i(n) + h_i(n) \geq r_i\big(a, \rho^2(n+1)\big) + \sum_j p_{ij}\big(\rho^2(n+1)\big)h_j(n)$ for all $a \in A(i)$. □

Lemma 10.27

(a) $g_i(n+1) \geq \sum_j p_{ij}(a)g_j(n+1)$ *for all* $(i,a) \in S_1 \times A$.

(b) $g_i(n+1) \geq \sum_j p_{ij}\big(\rho^2(n+1)\big)g_j(n+1)$ *for all* $i \in S_2$.

Proof

The proof of this Lemma an immediate consequence of the fact that $\big(g(n+1), v(n+1)\big)$ is an optimal solution of program $LP_1\big(\rho^2(n+1)\big)$. □

Lemma 10.28

Assume that the properties (1), (2), (3) and (4) hold. Then, $g(n+1) \leq g(n)$.

Proof

Take an arbitrary deterministic policy f^∞ for player 1 and choose the stationary policy ρ^∞ for player 2 as follows:

If $i \in S_1$: take an optimal strategy of player 2 in the matrix game $M_i^1(n+1)$ on $A_i(n+1) \times B(i)$.

If $i \in S2$: take $\rho_i(n+1)$, the extreme optimal strategy of player 2 in the matrix game $M_i^2(n+1)$ on $A(i) \times B_i(n+1)$.

From the properties (1) and (2) we obtain:

If $i \in S_1$: $g_i(n) \geq \sum_j p_{ij}(f)g_j(n)$.

If $i \in S_2$: Since $\rho_i^2(n+1)$ is defined on $B_i(n+1)$, we have $\sum_j p_{ij}(b)g_j(n) = min_{\overline{b}} \sum_j p_{ij}(\overline{b})g_j(n)$, if $\rho_{ib}^2(n+1) > 0$. Hence, $\sum_j p_{ij}\big(\rho^2(n+1)\big)g_j(n) = min_{\overline{b}} \sum_j p_{ij}(\overline{b})g_j(n)$. Consequently, $g_i(n) \geq \sum_j p_{ij}\big(\rho^2(n)\big)g_j(n) \geq min_{\overline{b}} \sum_j p_{ij}(\overline{b})g_j(n) = \sum_j p_{ij}\big(\rho^2(n+1)\big)g_j(n)$.

By the definition of ρ^∞, we have $g(n) \geq P(f,\rho)g(n)$, implying $g_i(n) = \{P(f,\rho)g(n)\}_i,\ i \in R(f,\rho)$ and $g(n) \geq P^*(f,\rho)g(n)$.

If $i \in R(f,\rho) \cap S_1$: $g_i(n) = \sum_j p_{ij}(f)g_j(n)$, i.e. $f(i) \in A_i(n+1)$. By property (3), we have $g_i(n) + h_i(n) \geq val\big(M_i^1(n+1)\big)$. Since ρ is optimal for player in the matrix game $M_i^1(n+1)$, we have $val\big(M_i^1(n+1)\big) \geq r_i(f,\rho) + \{P(f,\rho)\}_i$.

If $i \in R(f,\rho) \cap S_2$: $g_i(n) = \sum_j p_{ij}\big(\rho^2(n+1)\big)g_j(n) = min_{\overline{b}} \sum_j p_{ij}(\overline{b})g_j(n)$. By Lemma 10.26, we obtain $g_i(n) + h_i(n) \geq r_i(f,\rho) + \{P(f,\rho)h(n)\}_i$.

Hence, if $i \in R(f,\rho)$ we have $g_i(n) + h_i(n) \geq r_i(f,\rho) + \{P(f,\rho)h(n)\}_i$, implying the inequality $g(n) \geq P^*(f,\rho)r(f,\rho) = \phi(f^\infty,\rho^\infty)$. Since f^∞ is arbitrarily chosen, we obtain

$$\begin{aligned} g(n) &\geq max_f\, phi(f^\infty,\rho^\infty) \geq max_\pi\, \phi(\pi^\infty,\rho^\infty) \\ &\geq max_\pi\, min_\rho \,\{\phi(\pi^\infty,\rho^\infty) \mid \rho_i = \rho_i^2(n+1) \text{ for all } i \in S_2\} = g(n+1). \end{aligned}$$ □

Lemma 10.29

Assume that the properties (2), (3) and (4) hold. Furthermore, assume that $g(n+1) = g(n)$. *Then,* $R\big(\rho^2(n+1)\big) \subseteq R\big(\rho^2(n)\big)$ *and* $\rho_i^2(n+1) = \rho_i^2(n)$ *for all* $i \in R\big(\rho^2(n)\big) \cap S_2$.

Proof

Fix for player 2 the strategy $\rho_i^2(n+1)$ in the states $i \in S_2$. Because $g(n+1)$ is the v^*-part of the optimal solution of the linear program $LP_1\big(\rho^2(n+1)\big)$, we obtain by Lemma 10.24 (1), $g_i(n+1) = max_{a \in A(i)} \sum_j p_{ij}(a)g_j(n+1),\ i \in S$. In the states $i \in S_2$ player 1 has no influence on the transactions, namely: $p_{ij}(a) = p_{ij}\big(\rho^2(n+1)\big)$ for all $i \in S_2,\ j \in S$ and $a \in A(i)$. Therefore, we obtain for all $i \in S_2$: $g_i(n+1) = \sum_j p_{ij}\big(\rho^2(n+1)\big)g_j(n+1)$.

Since $car\big(\rho^2(n+1)\big) \subseteq B_i(n+1)$, we have $g_i(n+1) = min_{\overline{b}} \sum_j p_{ij}(\overline{b})g_j(n+1)$ for all $i \in S_2$. Because $g(n+1) = g(n)$, we have $g_i(n) = min_{\overline{b}} \sum_j p_{ij}(\overline{b})g_j(n+1)$ for all $i \in S_2$. Hence, by Lemma 10.26, $g_i(n) + h_i(n) \geq max_a \{r_i\big(a, \rho^2(n+1)\big) + \sum_j p_{ij}\big(\rho^2(n+1)\big)h_j(n)\}$ for all $i \in S_2$. Because $\rho^2_i(n+1)$ is an optimal strategy in the matrix game $M^2_i(n+1),\ i \in S_2$, we have

$$g_i(n) + h_i(n) \geq max_a \{r_i\big(a, \rho^2(n+1)\big) + \textstyle\sum_j p_{ij}\big(\rho^2(n+1)\big)h_j(n)\} = val\big(M^2_i(n+1)\big),\ i \in S_2.$$

Since $g(n+1) = g(n)$ equals the v^*-part of the optimal solution of $LP_1\big(\rho^2(n+1)\big)$, Lemma 10.24 (2) can be applied with $v^* := g(n),\ t^* := h(n)$ and $A^*(i) := A_i(n+1)$.
Hence, $g_i(n) + h_i(n) = val\big(M^2_i(n+1)\big)$ for all $i \in R\big(\rho^2(n)\big) \cap S_2$. So, by step 3 of Algorithm 10.20, $\rho^2_i(n+1) = \rho^2(n)$ for all $i \in R\big(\rho^2(n)\big) \cap S_2$.
Fix $k \in R\big(\rho^2(n+1)\big)$, i.e. there exists an optimal policy π^∞ for player 1 in the single-controller stochastic game induced by the policy $\rho^2_i(n+1)$ on the states $i \in S_2$ such that k is recurrent under $P(\pi)$. Because $\rho^2_i(n+1) = \rho^2_i(n)$ for all $i \in R\big(\rho^2(n)\big) \cap S_2$ and $g(n+1) = g(n)$, the policy π^∞ is also optimal in the single-controller stochastic game induced by the policy $\rho^2_i(n)$ on the states of S_2. Clearly, state k is recurrent, which shows that $R\big(\rho^2(n+1)\big) \subseteq R\big(\rho^2(n)\big)$. □

Lemma 10.30
Assume that the properties (2), (3), (4) and (7) hold. Then, $S(n+1) \cap R\big(\rho^2(n+1)\big) = \emptyset$.

Proof
If $g(n+1) \neq g(n)$, then by step 4(b) of Algorithm 10.20 we obtain $S(n+1) = \emptyset$, implying $S(n+1) \cap R\big(\rho^2(n+1)\big) = \emptyset$. Hence, suppose for the remaining part of the proof $g(n+1) = g(n)$. By Lemma 10.29, $R\big(\rho^2(n+1)\big) \subseteq R\big(\rho^2(n)\big)$ and, by property (7), $S(n) \cap R\big(\rho^2(n+1)\big) = \emptyset$. Because $S(n+1) = S(n) \cup G(n+1)$, it is sufficient to show that $G(n+1) \cap R\big(\rho^2(n+1)\big) = \emptyset$. From the proof of Lemma 10.29, we obtain $g_i(n) + h_i(n) = val\big(M^2_i(n+1)\big)$ for all $i \in R\big(\rho^2(n+1)\big) \cap S_2$. Lemma 10.24 and property (3) imply $g_i(n) + h_i(n) = val\big(M^1_i(n+1)\big)$ for all $i \in R\big(\rho^2(n+1)\big) \cap S_1$. Therefore, it follows from the definition of $G(n+1)$ in step 5 (d) of Algorithm 10.20 that $G(n+1) \cap R\big(\rho^2(n+1)\big) = \emptyset$. □

Lemma 10.31
Assume that the properties (2), (3), (4) and (7) hold. Furthermore, suppose that $g(n+1) = g(n)$ *and* $G(n+1) \neq \emptyset$. *Then,* $h(n+1) < h(n)$.

Outline of the proof (for details see [316])
Since $G(n+1) \neq \emptyset$ and $S(n+1) = S(n) \cup G(n+1)$ (see step 5 (e) of Algorithm 10.20), we also have $S(n+1) \neq \emptyset$. From the proof of Lemma 10.29 we obtain

$$g_i(n) + h_i(n) \geq val(M^2_i(n+1)),\ i \in S_2. \tag{10.100}$$

By property (3), we also have

$$g_i(n) + h_i(n) \geq val(M^1_i(n+1)),\ i \in S_1. \tag{10.101}$$

Because $G(n+1) \neq \emptyset$, the strict inequality holds in (10.100) or (10.101) in at least one state. Consider the stochastic game for program $LP_2\big(S(n{+}1), g(n{+}1), h(n), \rho^2(n{+}1), A(t{+}1)\big)$ in step 6 (b) of Algorithm 10.20. It can be shown that this is a semi-transient single-controller stochastic game. By step 6 (c) of Algorithm 10.20, we have $h_i(n+1) := \begin{cases} h_i(n) & \text{if } i \notin S(n+1); \\ h_i(n+1) & \text{if } i \in S(n+1). \end{cases}$ Therefore, we have to show that $u_i(n+1) < h_i(n)$ for at least one component $i \in S(n+1)$. This can be done using that the inequality (10.100) and (10.101)(10.102) is strict in at least one component. □

Lemma 10.32

Assume that the properties (2), (3), (4) and (7) hold. Then,
(a) $g_i(n+1) + h_i(n+1) \geq val\big(M_i^1(n+2)\big)$ *for all* $i \in S_1$.
(b) $g_i(n+1) + h_i(n+1) \geq r_i\big(a, \rho^2(n+1)\big) + \sum_j p_{ij}\big(\rho^2(n+1)\big)h_j(n+1)$ *for all* $(i,a) \in S_2 \times A$.

Proof

If $g(n+1) \neq g(n)$, then $h(n+1) = v(n+1)$ and the result follows from $LP_1\big(\rho^2(n+1)\big)$. Suppose that $g(n+1) = g(n)$. In the proof of Lemma 10.31 is shown that $h(n+1) \leq h(n)$. Furthermore, from the proof of Lemma 10.29, we obtain

$$\begin{aligned} g_i(n) + h_i(n) &\geq r_i\big(a, \rho^2(n+1)\big) + \textstyle\sum_j p_{ij}\big(\rho^2(n+1)\big)h_j(n) \text{ for all } (i,a) \in S_2 \times A. \\ &\geq r_i\big(a, \rho^2(n+1)\big) + \textstyle\sum_j p_{ij}\big(\rho^2(n+1)\big)h_j(n+1) \text{ for all } (i,a) \in S_2 \times A. \end{aligned}$$

If $i \notin S(n+1)$: $h_i(n+1) = h_i(n)$ by step 6 (c) of Algorithm 10.20. So, we obtain

$$\begin{aligned} g_i(n+1) + h_i(n+1) &= g_i(n) + h_i(n) \\ &\geq r_i\big(a, \rho^2(n+1)\big) + \textstyle\sum_j p_{ij}\big(\rho^2(n+1)\big)h_j(n+1) \end{aligned}$$

for all $(i,a) \in S_2 \times A$, which proves part (b). Since $h(n+1) \leq h(n)$ and by (3), $val\big(M_i^1(n+2)\big) \leq val\big(M_i^1(n)\big) \leq g_i(n) + h_i(n) = g_i(n+1) + h_i(n+1),\ i \in S_1$, which proves part (a).

If $i \in S(n{+}1)$: $h_i(n{+}1) = u_i(n{+}1)$ by step 6 (c) of Algorithm 10.20. Since $h_i(n{+}1) = u_i(n{+}1)$, $i \in S(n+1)$, is a solution of $LP_2\big(S(n+1), g(n+1), h(n), \rho^2(n+1), A(n+1)\big)$ and since $h_i(n) = h_i(n+1), i \notin S(n+1)$, we obtain for all $(i,a) \in S_2 \times A$: $g_i(n+1) + h_i(n+1) \geq r_i\big(a, \rho^2(n+1)\big) + \sum_j p_{ij}\big(\rho^2(n+1)\big)h_j(n+1)$, which proves part (b).

For $i \in S_1,\ LP_2\big(S(n+1), g(n+1), h(n), \rho^2(n+1), A(n+1)\big)$ implies $g_i(n+1) + h_i(n+1) \geq r_i\big(a, \rho^2(n+1)\big) + \sum_j p_{ij}(a)h_j(n+1),\ a \in A_i(n+1)$. Therefore, we obtain for all $i \in S_1$:

$$\begin{aligned} val\big(M_i^1(n+2)\big) &= max_\pi\, min_\rho \{r_i(\pi,\rho) + \textstyle\sum_j p_{ij}(\pi)h_j(n+1)\} \\ &\leq max_\pi\{r_i\big(\pi, \rho^2(n+1)\big) + \textstyle\sum_j p_{ij}(\pi)h_j(n+1)\} \\ &\leq max_a\{r_i\big(a, \rho^2(n+1)\big) + \textstyle\sum_j p_{ij}(a)h_j(n+1)\} \\ &\leq g_i(n+1) + h_i(n+1), \end{aligned}$$

which proves part (a). □

Theorem 10.44
The properties (1), (2), until (8) hold for all n.

Proof
We apply induction on n. It is easy to verify the 8 properties for $n = 0$. Assume that the properties hold for some n. Then, we have to prove the properties for $n+1$.
Properties (1) and (2): see Lemma 10.27.
Properties (3) and (4): see Lemma 10.32.
Property (5): see Lemma 10.28.
Property (6): see Lemma 10.29 (use also $R\big(\rho^2(n+1)\big) \subseteq R\big(\rho^2(n)\big)$).
Property (7): see Lemma 10.30.
Property (8): see Lemma 10.31. □

Theorem 10.45
Algorithm 10.20 terminates after a finite number of iterations.

Proof
Parthasarathy and Raghavan have shown (see Lemma 4.1 in [212]) that an extreme optimal action for player 2 in the matrix game with elements $r_i(a,b)+\sum_j p_{ij}(b)h_j(n)$ on $A(i)\times B_i(n+1)$ is also an extreme optimal action for player 2 in some subgame with elements $r_i(a,b)$ on $A(i)\times \overline{B}_i(n+1)$ with $\overline{B}_i(n+1) \subseteq B_i(n+1)$. From a theorem by Shapley and Snow ([268]), and also from the linear programming approach of matrix games, we know that optimal strategies can be found in a submatrix game, where this submatrix is square and nonsingular. Since a matrix game has only a finite number of submatrices and since the set S_2 is finite, there is a finite set from which $\rho^2(n+1)$ can be chosen in step 3 of Algorithm 10.20 and this set is independent of n.
By the properties (5) and (8) we can see that for each n exactly one of the following events occurs:
(a) $g(n) = g(n-1)$ and $G(n) = \emptyset$.
(b) $g(n) < g(n-1)$.
(c) $g(n) = g(n-1),\ G(n) \neq \emptyset$ and $\rho^2(n) = \rho^2(n-1)$.
(d) $g(n) = g(n-1),\ G(n) \neq \emptyset$ and $\rho^2(n) \neq \rho^2(n-1)$.
We consider these four cases separately.

Case (a):
In this case the algorithm terminates, via step 5 (e) and step 7, in step 8.

Case (b):
$\rho^2(k) \neq \rho^2(l)$ for all $k \geq n$ and $l \leq n-1$, namely:

> Assume $\rho^2(k) = \rho^2(l)$ for some $k \geq n$ and $l \leq n-1$. Since in step 4 (a) of the algorithm $LP_1(\rho)$ only depends on ρ and the optimal solution g is unique, we have $g(k) = g(l)$ for some $k \geq n$ and $l \leq n-1$. This contradicts (5) and (a).

Since there is a finite set from which $\rho^2(n+1)$ can be chosen in step 3 of the algorithm, this case cannot occur infinitely often.

Case (c):

$S(n-1) \subset S(n)$ and $S(n-1) \neq S(n)$, namely:

Since $S(n) = S(n-1) \cup G(n)$ and $G(n) \neq \emptyset$, it is sufficient to show $S(n-1) \cap G(n) = \emptyset$.
Take any $i \in S(n-1)$. Because $u(n-1)$ is an optimal solution of the linear program $LP_2\big(S(n-1), g(n-1), h(n-2), \rho^2(n-1), A(n-1)\big)$, we obtain by Lemma 10.25:
If $i \in S(n-1) \cap S1$:
$u_i(n-1) = val\{r_i(a,b) - g_i(n-1) + \sum_{j \notin S(n-1)} p_{ij}(a)h_j(n-2) + \sum_{j \in S(n-1)} p_{ij}(a)u_j(n-1)\}$.
Because, see step 6 (c) of the algorithm, $h_j(n-2) = h_j(n-1),\ i \notin S(n-1)$ and furthermore, $u_j(n-1) = h_j(n-1),\ i \in S(n-1)$, we obtain
$g_i(n-1) + h_i(n-1) = val\{r_i(a,b) + \sum_j p_{ij}(a)h_j(n-1)\} = val\big(M_i^1(n-1)\big),\ i \in S(n-1) \cap S_1$, i.e. $i \notin G_1(n)$.
If $i \in S(n-1) \cap S_2$:
Similarly as above and because player 2 uses strategy $\rho_i^2(n-1) = \rho_i^2(n)$ in the states $i \in S_2$, we obtain $g_i(n-1) + h_i(n-1) = val\{r_i\big(a, \rho^2(n)\big) + \sum_j p_{ij}\big(\rho^2(n)\big)h_j(n-1)\}$. Since $\rho^2(n)$ is an extreme optimal strategy for player 2 in the matrix game $M_i^2(n-1)$, we have $g_i(n-1) + h_i(n-1) = val\big(M_i^2(n-1)\big)$, i.e. $i \notin G_2(n)$.
Hence, we have shown that $S(n-1) \cap G(n) = \emptyset$.

Case (d):

This case cannot be repeat itself infinitely often, namely:

Assume that from stage n this case repeats itself infinitely often. Then, $g(m) = g(n)$ for all $m \geq n$ and $G(m) \neq \emptyset$ for $m \geq n$. Since $S(m+1) = S(m) \cup G(m+1)$ and $|S(m+1| \leq N$ for all m, we may assume without loss of generality that $S(n) = S(n+1) = S(n+2) = \cdots$.
Consider the programs $LP_2\big(S(n+k), g(n+k), h(n+k-1), \rho^2(n+k), A(n+k)\big)$ for $k = 0, 1, \ldots$.
These programs only depend on $\rho^2(n+k)$ as the other parameters do not change ($h(n+k-1)$ is used for the states $S \backslash S(n+k)$ and the values $h_i(n+k-1)$ do not change in these states by step 6 (c) of the algorithm). Lemma 10.31 implies $h(n) > h(n+1) > h(n+2) > \cdots$.
Suppose that $\rho^2(n+l) = \rho^2(n+k)$ for some $l > k$. Then, the solution the the program for $n+l$ and $n+k$ are equal, and consequently (see step 7 (c) of the algorithm) $h(n+l) = h(n+k)$.
But this yields a contradiction.

Now we can finish the proof as follows. Assume that the algorithm does not terminate. Since in case (a) the algorithm terminates and case (b) cannot occur infinitely often, we only have to consider the situation in which only the cases (c) and (d) occur from a certain stage n. In this situation $g(m) = g(n)$ for all $m \geq n$ and $S(n) \subseteq S(n+1) \subseteq S(n+2) \subseteq \cdots$. From the result of case (c), i.e. $S(n-1) \subset S(n)$ and $S(n-1) \neq S(n)$, it follows that case (c) cannot occur infinitely often. Hence, we are always in case (d) from a certain stage. But this contradicts the result that case (d) cannot be repeat itself infinitely. □

Theorem 10.46

Algorithm 10.20 terminates with optimal policies for both players and with the value vector.

Proof

By Theorem 10.45 the algorithm terminates, say in stage $n+1$, so we have $g(n+1) = g(n)$. From the proof of Lemma 10.29 we obtain

$$g_i(n) = min_{\overline{b}} \sum_j p_{ij}(\overline{b})g_j(n),\ i \in S_2. \tag{10.102}$$

Since $g(n)$ is the optimal solution of $LP_1\big(\rho^2(n)\big)$, we observe from Lemma 10.24, part (1), that

$$g_i(n) = max_a \sum_j p_{ij}(a)g_j(n),\ i \in S_1. \tag{10.103}$$

From the definitions of π^* and ρ^* in step 7 of the algorithm we know

$$car(\pi^*) \subseteq A_i(n+1),\ i \in S_1 \text{ and } car(\rho^*) \subseteq B_i(n+1),\ i \in S_2. \tag{10.104}$$

Because $G_1(n+1) = \emptyset$, property (3) implies

$$g_i(n) + h_i(n) = val\big(M_i^1(n+1)\big),\ i \in S_1. \tag{10.105}$$

From the proof of Lemma 10.26 we obtain

$g_i(n) + h_i(n) \geq max_a \{r_i\big(a, \rho^2(n+1)\big) + \sum_j p_{ij}\big(\rho^2(n+1)\big)h_j(n)\} = val\big(M_i^2(n+1)\big),\ i \in S_2.$

As $G_2(n+1) = \emptyset$, we have

$$g_i(n) + h_i(n) = val\big(M_i^2(n+1)\big),\ i \in S_2. \tag{10.106}$$

Let f^∞ be an arbitrary deterministic policy for player 1. For $i \in S_1$, relation (10.103) implies $g_i(n) \geq \sum_j p_{ij}(f)g_j(n)$. For $i \in S_2$, since $car(\rho^*) \subseteq B_i(n+1)$ and by relation (10.102), we have $\sum_j p_{ij}(\rho^*)g_j(n) = min_{\overline{b}} \sum_j p_{ij}(\overline{b})g_j(n) = g_i(n)$. Hence, $g(n) \geq P(f, \rho^*)g(n)$, which yields

$$g(n) \geq P^*(f, \rho^*)g(n) \tag{10.107}$$

For $i \in S_1$, by relation (10.105) and because ρ^* is optimal for $M_i^1(n+1)$, we can write

$$\begin{aligned} g_i(n) + h_i(n) &= val\big(M_i^1(n+1)\big) = max_\pi\, min_\rho \{r_i(\pi, \rho) + \textstyle\sum_j p_{ij}(\pi)h_j(n)\} \\ &= max_\pi \{r_i(\pi, \rho^*) + \textstyle\sum_j p_{ij}(\pi)h_j(n)\} \\ &\geq r_i(f, \rho^*) + \textstyle\sum_j p_{ij}(f)h_j(n). \end{aligned}$$

For $i \in S_2$, by relation (10.106) and because ρ^* is optimal for $M_i^2(n+1)$, we can write

$$\begin{aligned} g_i(n) + h_i(n) &= val\big(M_i^2(n+1)\big) = max_\pi\, min_\rho \{r_i(\pi, \rho) + \textstyle\sum_j p_{ij}(\rho)h_j(n)\} \\ &= max_\pi \{r_i(\pi, \rho^*) + \textstyle\sum_j p_{ij}(\rho^*)h_j(n)\} \\ &\geq r_i(f, \rho^*) + \textstyle\sum_j p_{ij}(\rho^*)h_j(n). \end{aligned}$$

Hence,

$$g(n) + h(n) \geq r(f, \rho^*) + P^*(f, \rho^*)g(n). \tag{10.108}$$

Combining (10.107) and (10.108) yields

$$g(n) \geq P^*(f,\rho^*)g(n) \geq P^*(f,\rho^*)\{r(f,\rho^*) + P(f,\rho^*)h(n) - h(n)\} = \phi\big(f^\infty,(\rho^*)^\infty\big). \quad (10.109)$$

Let g^∞ be an arbitrary deterministic policy for player 2. For $\in S_2$, relation (10.102) implies $g_i(n) \leq \sum_j p_{ij}(g)g_j(n)$. For $i \in S_1$, since $car(\pi^*) \subseteq A_i(n+1)$, we have $\sum_j p_{ij}(\pi^*)g_j(n) = g_i(n)$. Hence, $g(n) \leq P(\pi^*,g)g(n)$, which yields

$$g(n) \leq P^*(\pi^*,g)g(n) \quad (10.110)$$

For $i \in S_1$, by relation (10.105) and because π^* is optimal for $M_i^1(n+1)$, we can write

$$\begin{array}{rcl} g_i(n)+h_i(n) & = & val\big(M_i^1(n+1)\big) = max_\pi\, min_\rho\, \{r_i(\pi,\rho) + \sum_j p_{ij}(\pi)h_j(n)\} \\ & = & min\rho\, \{r_i(\pi^*,\rho) + \sum_j p_{ij}(\pi^*)h_j(n)\} \\ & \leq & r_i(\pi^*,g) + \sum_j p_{ij}(\pi^*)h_j(n). \end{array}$$

For $i \in S_2$, by relation (10.106) and because π^* is optimal for $M_i^2(n+1)$, we can write

$$\begin{array}{rcl} g_i(n)+h_i(n) & = & val\big(M_i^2(n+1)\big) = max_\pi\, min_\rho\, \{r_i(\pi,\rho) + \sum_j p_{ij}(\rho)h_j(n)\} \\ & = & min\rho\, \{r_i(\pi^*,\rho) + \sum_j p_{ij}(\rho)h_j(n)\} \\ & \leq & r_i(\pi^*,g) + \sum_j p_{ij}(g)h_j(n). \end{array}$$

Hence,

$$g(n)+h(n) \leq r(\pi^*,g) + P^*(\pi^*,g)g(n). \quad (10.111)$$

Combining (10.110) and (10.111) yields

$$g(n) \leq P^*(\pi^*,g)g(n) \leq P^*(\pi^*,g)\{r(\pi^*,g) + P(\pi^*,g)h(n) - h(n)\} = \phi\big((\pi^*)^\infty,g^\infty\big). \quad (10.112)$$

The relations (10.108) and (10.111) imply $\phi\big(f^\infty,(\rho^*)^\infty\big) \leq g(n) \leq \phi\big((\pi^*)^\infty,g^\infty\big)$ for all deterministic policies f^∞ and g^∞ for player 1 and 2, respectively. Hence, $(\pi^*)^\infty$ and $(\rho^*)^\infty$ are optimal stationary policies and $g(n)$ is the value vector. □

Remark

Algorithm 10.20 provides a constructive proof of the existence of the value and of optimal stationary policies for both players. Furthermore, the algorithm proves the ordered field property for the switching-controller stochastic game.

Switching-controller stochastic game and bilinear programming

Filar ([89]) has established that switching-controller stochastic games possess asymptotic stable optimal stationary policies. In Theorem 10.32 we have shown that in a stochastic game which possesses asymptotic stable optimal stationary policies, say π^∞ and ρ^∞, the bilinear system (10.65) has a feasible solution (v,t,ρ,π). Define the vectors g and h by

$$g_i := \begin{cases} \sum_j \sum_a p_{ij}(a)\pi_{ia}v_j, & i \in S_1 \\ \sum_j \sum_b p_{ij}(b)\rho_{ib}v_j, & i \in S_2 \end{cases} \text{ and } h_i := \begin{cases} \sum_j \sum_a p_{ij}(a)\pi_{ia}t_j, & i \in S_1 \\ \sum_j \sum_b p_{ij}(b)\rho_{ib}t_j, & i \in S_2 \end{cases}$$

Then, the constraints of (10.65) become the following 8 sets of bilinear inequalities:

$$\begin{array}{lllll}
(1) & v_i - \sum_j p_{ij}(a)v_j & \geq & 0, & (i,a) \in S_1 \times A \\
(2) & v_i - g_i & \geq & 0, & i \in S_2 \\
(3) & v_i + t_i - \sum_j p_{ij}(a)t_j - \sum_b r_i(a,b)\rho_{ib} & \geq & 0, & (i,a) \in S_1 \times A \\
(4) & v_i + t_i - h_i - \sum_b r_i(a,b)\rho_{ib} & \geq & 0, & (i,a) \in S_1 \times A \\
(5) & v_i - g_i & \leq & 0, & i \in S_1 \\
(6) & v_i - \sum_j p_{ij}(b)v_j & \leq & 0, & (i,b) \in S_2 \times A \\
(7) & v_i + t_i - h_i - \sum_a r_i(a,b)\pi_{ia} & \leq & 0, & (i,b) \in S_1 \times B \\
(8) & v_i + t_i - \sum_j p_{ij}(b)t_j - \sum_a r_i(a,b)\pi_{ia} & \leq & 0, & (i,b) \in S_2 \times B
\end{array}$$

Consider the bilinear program $BLP2$ with objective function

$$\phi(g,h,v,t,\pi,\rho) := \sum_{i\in S_1} \{g_i - \sum_j \sum_a p_{ij}(a)\pi_{ia}v_j\} + \sum_{i\in S_1} \{h_i - \sum_j \sum_a p_{ij}(a)\pi_{ia}t_j\}$$
$$\sum_{i\in S_2} \{g_i - \sum_j \sum_b p_{ij}(b)\rho_{ib}v_j\} - \sum_{i\in S_2} \{h_i - \sum_j \sum_b p_{ij}(b)\rho_{ib}t_j\}$$

and as constraints the above eight sets of inequalities and with the addition constraints $\sum_a \pi_{ia} = 1,\ i \in S,\ \pi_{ia} \geq 0,\ (i,a) \in S \times A;\ \sum_b \rho_{ib} = 1,\ i \in S,\ \rho_{ib} \geq 0,\ (i,b) \in S \times B.$

Theorem 10.47

(1) An optimal solution of $BLP2$ has value 0 and can be derived from any pair of asymptotic stable optimal stationary policies $(\pi^*)^\infty$ and $(\rho^*)^\infty$.

(2) The value vector and optimal stationary policies $(\pi^*)^\infty$ and $(\rho^*)^\infty$ can be obtained from any optimal solution $(g^*,h^*,v^*,t^*,\pi^*,\rho^*)$ of $BLP2$.

Proof

(1) Let (v^*,t^*,π^*,ρ^*) be a feasible solution of the bilinear system (10.65). Such solution exists by Theorem 10.32. Define g^* and h^* by

$$g_i^* := \begin{cases} \sum_j \sum_a p_{ij}(a)\pi_{ia}^* v_j^*, & i \in S_1 \\ \sum_j \sum_b p_{ij}(b)\rho_{ib}^* v_j^*, & i \in S_2 \end{cases} \text{ and } h_i^* := \begin{cases} \sum_j \sum_a p_{ij}(a)\pi_{ia}^* t_j^*, & i \in S_1 \\ \sum_j \sum_b p_{ij}(b)\rho_{ib}^* t_j^*, & i \in S_2 \end{cases}$$

Then, it is obvious that $(g^*,h^*,v^*,t^*,\pi^*,\rho^*)$ is a feasible solution of $BLP2$ with value zero of the objective function. Hence, it is sufficient to show that the objective function is at least 0 for any feasible solution (g,h,v,t,π,ρ). Let (g,h,v,t,π,ρ) be a feasible solution. From (1) and (5), we obtain $g_i \geq v_i \geq \sum_j p_{ij}(\pi)v_j,\ i \in S_1$. Hence, $\sum_{i\in S_1} \{g_i - \sum_j \sum_a p_{ij}(a)\pi_{ia}v_j\} \geq 0$. From (3) and (7), we obtain $v_i + t_i - \sum_j p_{ij}(\pi)t_j - r_i(\pi,\rho) \geq 0,\ i \in S_1$ and $v_i + t_i - r_i(\pi,\rho) \leq h_i$, $i \in S_1$. Therefore, we have $\sum_{i\in S_1} \{h_i - \sum_j \sum_a p_{ij}(a)\pi_{ia}t_j\} \geq 0$. Similarly, it can be shown that the third and fourth term of the objective function are nonpositive. Therefore $\phi(g,h,v,t,\pi,\rho) \geq 0$ for any feasible solution (g,h,v,t,π,ρ) of the bilinear program.

(2) Let $(g^*,h^*,v^*,t^*,\pi^*,\rho^*)$ be an optimal solution of $BLP2$. Then, it follows from part (1) that $g_i^* = \sum_j \sum_a p_{ij}(a)\pi_{ia}^* v_j^*,\ i \in S_1$, and $h_i^* = \sum_j \sum_a p_{ij}(a)\pi_{ia}^* t_j^*,\ i \in S_1$. Similarly, we obtain $g_i^* = \sum_j \sum_b p_{ij}(b)\rho_{ib}^* v_j^*,\ i \in S_2$, and $h_i^* = \sum_j \sum_b p_{ij}(b)\rho_{ib}^* t_j^*,\ i \in S_2$. Hence, (v^*,t^*,π^*,ρ^*) is a feasible solution of the bilinear system (10.65). Then, by Theorem 10.30, $(\pi^*)^\infty$ and $(\rho^*)^\infty$ are optimal stationary policies and v^* is the value vector. □

Remark 1

In $BLP2$ the variables g, h, v and t appear in constraints together with the variables π and ρ. Bilinear programs of this form may have solutions not at a vertex. The method proposed by Faiz and Falk ([79]) gives a finite method for an ε-optimal solution.

Remark 2

An analogous result holds for the switching-controller stochastic game with the discounted reward criterion.

SER-SIT games

In this subsection we consider a stochastic game with *separable* rewards SER) and *state independent* transitions (SIT), i.e. $r_i(a,b) = s_i + t(a,b)$ and $p_{ij}(a,b) = p_j(a,b),\ j \in S$, for all i, a, b, under the average reward criterion. Let $|A(i)| = m$ and $|B(i)| = n$ for all $i \in S$ (notice that the SIT-property makes only sense when in all states the number of actions for player 1 (player 2) is the same). Consider the matrix games with $m \times n$ matrix $M = (m_{ab})$, where $m_{ab} := t(a,b) + \sum_j p_j(a,b)s_j,\ 1 \leq a \leq m,\ 1 \leq b \leq n.$

Theorem 10.48

Let $\pi^* = (\pi_1, \pi_2, \ldots, \pi_m)$ and $\rho^* = (\rho_1, \rho_2, \ldots, \rho_n)$ be optimal mixed strategies of the matrix game with matrix M. Then, $\phi = val(M) \cdot e$ is the value vector and $(\pi^*)^\infty$ and $(\rho^*)^\infty$ are optimal policies for player 1 and player 2, respectively.

Proof

Since $val(M) \leq t(\pi^*, \rho) + \sum_j p_j(\pi^*, \rho)s_j$ for all ρ, we also have, in vector notation, where the matrix $P(\pi^*, \rho)$ has identical rows, $s + val(M) \cdot e \leq s + t(\pi^*, \rho) \cdot e + P(\pi^*, \rho)s$ for all ρ. By applying $P^*(\pi^*, \rho)$ on both sides, we obtain $val(M) \cdot e \leq P^*(\pi^*, \rho)\{s + t(\pi^*, \rho) \cdot e\} = \phi\big((\pi^*)^\infty, \rho^\infty\big)$ for all $\rho^\infty \in \Gamma$. Similarly, one can prove $val(M) \cdot e \geq P^*(\pi, \rho^*)\{s + t(\pi, \rho^*) \cdot e\} = \phi\big(\pi^\infty, (\rho^*)^\infty\big)$ for all $\pi^\infty \in \Pi$. Hence, $\phi = val(M) \cdot e$ is the value vector and $(\pi^*)^\infty$ and $(\rho^*)^\infty$ are optimal policies for player 1 and player 2, respectively. □

Algorithm 10.21 *SER-SIT game with no discounting*
Input: Instance of a two-person SER-SIT stochastic game
Output: The value vector ϕ and a pair $\big((\pi^*)^\infty, (\rho^*)^\infty\big)$ of optimal stationary policies.

1. Compute the matrix M with entries $m_{ab} := t(a,b) + \sum_j p_j(a,b)s_j,\ a \in A(i),\ b \in B(i)$.

2. Determine the value ϕ and optimal mixed strategies π^* and ρ^* of the matrix game with matrix M.

3. $\phi \cdot e$ is the value vector; $(\pi^*)^\infty$ and $(\rho^*)^\infty$ are optimal stationary policies for player 1 and player 2, respectively (STOP).

Remark

Since ϕ and the optimal mixed strategies π^* and ρ^* can be computed by linear programming, *SER-SIT* games possess the ordered field property.

ARAT games

An additive reward and additive transition (*ARAT*) stochastic game is defined by the property that the rewards as well as the transitions can be written as the sum of a term determined by player 1 and a term determined by player 2: $r_i(a,b) = r_i^1(a) + r_i^2(b),\ i \in S,\ a \in A(i),\ b \in B(i)$ and $p_{ij}(a,b) = p_{ij}^1(a) + p_{ij}^2(b),\ i,j \in S,\ a \in A(i),\ b \in B(i)$. We will argue the result that both players have optimal deterministic and stationary policies and that the ordered field property holds. For the details we refer to [233] and [99].

We have seen in Theorem 10.17 that, in the case of discounted rewards, both players have optimal deterministic and stationary policies and that the ordered field property holds. As usual, since there are only a finite number of deterministic and stationary policies, taking a sequence of discount factors tending to 1, some optimal deterministic and stationary pair of policies appears infinitely often, giving rise to a uniform discount optimal policy. But then, such pair is also average reward optimal.

A finite algorithm to compute the value vector and optimal deterministic and stationary policies resembles the algorithm of Vrieze, Raghavan, Tijs and Filar ([316]). There are some simplifications: no partition of the state space is needed, so $S_1 = S$ and $S_2 = \emptyset$. Furthermore, the policies can be taken deterministic and stationary.

10.5 Two-person general-sum stochastic game

10.5.1 Introduction

In a *two-person general-sum stochastic game*, if the players 1 and 2 choose in state i independently the actions a and b, they receive $r_i^1(a,b)$ and $r_i^2(a,b)$, respectively. In this game the two players try to maximize their own payoff. The zero-sum game is the special case in which $r_i^2(a,b) = -r_i^1(a,b)$ for all i, a and b. In a general-sum stochastic game the usual concepts for the value and optimal policies make no sense. It looks reasonable to assume that the solution of the nonzero-sum game is such that, given the policy of one player, the policy of the other player is such that it maximizes his payoff. This viewpoint leads to the concept of *equilibrium policies.*

A pair (R_1^*, R_2^*) is a pair of equilibrium policies if R_1^* is the best answer against R_2^*, and R_2^* is the best answer against R_1^*. Hence, in an equilibrium neither of the players has an incentive for a unilateral deviation from such an equilibrium policy. The formal definitions are as follows.

For the policies R_1 and R_2 for player 1 and 2, respectively, the *total discounted rewards* $v^{1,\alpha}(R_1,R_2)$ and $v^{2,\alpha}(R_1,R_2)$, and the*average rewards* $\phi^1(R_1,R_2)$ and $\phi^2(R1,R2)$ are defined by

$$v_i^{1,\alpha}(R_1,R_2) := \sum_{t=1}^{\infty} \alpha^{t-1} \sum_{j,a,b} \mathbb{P}_{i,R_1,R_2}\{X_t = j,\ Y_t = a,\ Z_t = b\} \cdot r_j^1(a,b),\ i \in S; \qquad (10.113)$$

$$v_i^{2,\alpha}(R_1,R_2) := \sum_{t=1}^{\infty} \alpha^{t-1} \sum_{j,a,b} \mathbb{P}_{i,R_1,R_2}\{X_t=j,\ Y_t=a,\ Z_t=b\}\cdot r_j^2(a,b),\ i\in S; \tag{10.114}$$

$$\phi_i^1(R_1,R_2) := \liminf_{T\to\infty} \frac{1}{T} \sum_{t=1}^{T} \sum_{j,a,b} \mathbb{P}_{i,R_1,R_2}\{X_t=j,\ Y_t=a,\ Z_t=b\}\cdot r_j^1(a,b),\ i\in S; \tag{10.115}$$

$$\phi_i^2(R_1,R_2) := \liminf_{T\to\infty} \frac{1}{T} \sum_{t=1}^{T} \sum_{j,a,b} \mathbb{P}_{i,R_1,R_2}\{X_t=j,\ Y_t=a,\ Z_t=b\}\cdot r_j^2(a,b),\ i\in S. \tag{10.116}$$

A pair (R_1^*,R_2^*) is a pair of equilibrium policies for discounted rewards if

$$\begin{aligned} v_i^{1,\alpha}(R_1^*,R_2^*) &\geq v_i^{1,\alpha}(R_1,R_2^*) \text{ for all } R_1 \text{ and all } i\in S & (10.117)\\ v_i^{2,\alpha}(R_1^*,R_2^*) &\geq v_i^{2,\alpha}(R_1^*,R_2) \text{ for all } R_2 \text{ and all } i\in S & (10.118)\end{aligned}$$

A pair (R_1^*,R_2^*) is a pair of equilibrium policies for undiscounted rewards if

$$\begin{aligned} \phi_i^{1,\alpha}(R_1^*,R_2^*) &\geq \phi_i^{1,\alpha}(R_1,R_2^*) \text{ for all } R_1 \text{ and all } i\in S & (10.119)\\ \phi_i^{2,\alpha}(R_1^*,R_2^*) &\geq \phi_i^{2,\alpha}(R_1^*,R_2) \text{ for all } R_2 \text{ and all } i\in S & (10.120)\end{aligned}$$

10.5.2 Discounted rewards

Before showing the existence of an equilibrium in nonzero-sum discounted stochastic games, we shall give two lemmata. In the sequel $v^{1,\alpha}$ will denote the value of the zero-sum stochastic game based on the payoffs of player 1. Furthermore, $v^{2,\alpha}$ will denote the value of the zero-sum stochastic game based on the payoffs of player 2, where player 2 is the maximizing player and player 1 the minimizing player.

Lemma 10.33

Let (R_1^*,R_2^*) be a pair of equilibrium policies of a discounted general-sum stochastic game. Then, $v^{1,\alpha}(R_1^*,R_2^*)\geq v^{1,\alpha}$ and $v^{2,\alpha}(R_1^*,R_2^*)\geq v^{2,\alpha}$.

Proof

$v^{1,\alpha}(R_1^*,R_2^*)\geq v^{1,\alpha}(R_1,R_2^*)$ for all policies R_1. Therefore, $max_{R_1}\, v^{1,\alpha}(R_1,R_2^*) = v^{1,\alpha}(R_1^*,R_2^*)$, implying $v^{1,\alpha} = min_{R_2}\, max_{R_1}\, v^{1,\alpha}(R_1,R_2) \leq max_{R_1}\, v^{1,\alpha}(R_1,R_2^*) = v^{1,\alpha}(R_1^*,R_2^*)$. Similarly, it can be shown that $v^{2,\alpha}(R_1^*,R_2^*)\geq v^{2,\alpha}$. □

Lemma 10.34

For a discounted zero-sum stochastic game, the following statements are equivalent:
(1) (R_1^*,R_2^*) is a pair of equilibrium policies.
(2) R_1^* is optimal for player 1, R_2^* is optimal for player 2, and $v^{1,\alpha}(R_1^*,R_2^*)=v^{1,\alpha}$.

Proof

Assume that (R_1^*, R_2^*) is a pair of equilibrium policies. Then, for all policies R_1 and R_2 we have $v^{1,\alpha}(R_1^*, R_2) = -v^{2,\alpha}(R_1^*, R_2) \geq -v^{2,\alpha}(R_1^*, R_2^*) = v^{1,\alpha}(R_1^*, R_2^*) \geq v^{1,\alpha}(R_1, R_2^*)$. Hence, by Theorem 10.4, R_1^* is optimal for player 1, R_2^* is optimal for player 2, and $v^{1,\alpha}(R_1^*, R_2^*) = v^{1,\alpha}$. Conversely, assume that R_1^* and R_2^* are optimal for player 1 and player 2, and $v^{1,\alpha}(R_1^*, R_2^*) = v^{1,\alpha}$. Then, for all policies R_1 and R_2 for player 1 and 2, respectively, we have

$$v^{1,\alpha}(R_1^*, R_2) \geq \inf_{R_2} \sup_{R_1} v^{1,\alpha}(R_1, R_2) \geq \sup_{R_1} \inf_{R_2} v^{1,\alpha}(R_1, R_2) \geq v^{1,\alpha}(R_1, R_2^*).$$

Therefore, $v^{1,\alpha}(R_1^*, R_2^*) \geq v^{1,\alpha}(R_1, R_2^*)$ for all policies R_1, and also $v^{1,\alpha}(R_1^*, R_2^*) \leq v^{1,\alpha}(R_1^*, R_2)$ for all policies R_2. Because $v^{1,\alpha}(R_1, R_2) = -v^{2,\alpha}(R_1, R_2)$, also we have $v^{2,\alpha}(R_1^*, R_2^*) \geq v^{2,\alpha}(R_1^*, R_2)$ for all policies R_2. Hence, (R_1^*, R_2^*) is a pair of equilibrium policies. □

Theorem 10.49

Every discounted general-sum stochastic game possesses at least one pair of equilibrium policies in stationary policies.

Proof

The proof, based on fixed point arguments, is divided into two parts. In part 1 a multi-valued mapping T is defined and is shown to possess at least one fixed point. In part 2 it is shown that every fixed point of T coincides with an equilibrium point in stationary policies.

Part 1

Let Π and Σ denote the set of stationary decision rules for for player 1 and 2, respectively. Obviously, Π and Σ are convex and compact. Define the multi-valued mapping $T : \Pi \times \Sigma \to \Pi \times \Sigma$ as follows:

$$T(\overline{\pi}, \overline{\sigma}) := \left\{ (\hat{\pi}, \hat{\sigma}) \in \Pi \times \Sigma \;\middle|\; \begin{array}{lcl} v^{1,\alpha}(\hat{\pi}^\infty, \overline{\sigma}^\infty) & \geq & v^{1,\alpha}(\pi^\infty, \overline{\sigma}^\infty) \text{ for all } \pi \in \Pi \\ v^{2,\alpha}(\overline{\pi}^\infty, \hat{\sigma}^\infty) & \geq & v^{2,\alpha}(\overline{\pi}^\infty, \sigma^\infty) \text{ for all } \sigma \in \Sigma \end{array} \right\}.$$

Given a fixed $\sigma \in \Sigma$ for player 2, the process becomes an MDP, denoted by MDP(σ), for player 1. For MDPs it is well known that the set of optimal stationary policies is the convex hull of the finite set of deterministic stationary policies. So, $\Pi \times \Sigma$ is compact and convex. For the existence of a fixed point, we apply Kakutani's Theorem (see [145]).[3] Therefore, it is sufficient to show that if $(\overline{\pi}_n, \overline{\sigma}_n) \to (\overline{\pi}, \overline{\sigma})$ and if $(\hat{\pi}_n, \hat{\sigma}_n) \to (\hat{\pi}, \hat{\sigma})$ are such that $(\hat{\pi}_n, \hat{\sigma}_n) \in T(\overline{\pi}_n, \overline{\sigma}_n)$ for all n, then it holds that $(\hat{\pi}, \hat{\sigma}) \in T(\overline{\pi}, \overline{\sigma})$. Let $(\overline{\pi}_n, \overline{\sigma}_n) \to (\overline{\pi}, \overline{\sigma})$, $(\hat{\pi}_n, \hat{\sigma}_n) \to (\hat{\pi}, \hat{\sigma})$ and $(\hat{\pi}_n, \hat{\sigma}_n) \in T(\overline{\pi}_n, \overline{\sigma}_n)$ for all n. Then, $v^{1,\alpha}(\hat{\pi}_n^\infty, \overline{\sigma}_n^\infty) \geq v^{1,\alpha}(\pi^\infty, \overline{\sigma}_n^\infty)$ for all $\pi \in \Pi$ and all n. Hence,

$$\{I - \alpha P(\hat{\pi}_n, \overline{\sigma}_n)\}^{-1} r^1(\hat{\pi}_n, \overline{\sigma}_n) \geq \{I - \alpha P(\hat{\pi}, \overline{\sigma}_n)\}^{-1} r^1(\hat{\pi}, \overline{\sigma}_n) \text{ for all } \pi \in \Pi \text{ and all } n.$$

By the continuity of the matrices $\{I - \alpha P(\pi, \sigma)\}^{-1}$,[4] we obtain

[3] **Theorem (Kakutani, 1941)**:
Let X be a nonempty compact convex subset of $\mathbb{R}^n$ and let $F : X \to X$ be a multi-valued mapping for which: (i) for all $x \in X$ the set $F(x)$ is nonempty and convex; (ii) the graph of F is closed i.e. for all sequences $\{x_n\}$ and $\{y_n\}$ such that $y_n \in F(x_n)$ for all n, $x_n \to x$, and $y_n \to y$, we have $y \in F(x)$. Then, F has a fixed point.

[4] Each column of the inverse of a nonsingular matrix A can be computed by solving a system of linear equations with Cramers rule; $Ax = e^j$ gives the jth column of the inverse, where e^j is the jth unit vector. Cramers rule says that the elements of the jth column of the inverse are a quotient of polynomials with non-zero denominator $det(A)$ and as numerator the determinant of the matrix A, but in column j the vector e^j. Hence, the inverse matrix is continuous in the original data.

$$v^{1,\alpha}(\hat{\pi}^\infty,\overline{\sigma}^\infty) = \{I-\alpha P(\hat{\pi},\overline{\sigma})\}^{-1}r^1(\hat{\pi},\overline{\sigma}) \geq \{I-\alpha P(\hat{\pi},\overline{\sigma})\}^{-1}r^1(\hat{\pi},\overline{\sigma}) = v^{1,\alpha}(\pi^\infty,\overline{\sigma}^\infty)$$

for all $\pi \in \Pi$. Similarly, it can be shown that $v^{2,\alpha}(\overline{\pi}^\infty,\hat{\sigma}^\infty) \geq v^{2,\alpha}(\overline{\pi}^\infty,\sigma^\infty)$ for all $\sigma \in \Sigma$.

Part 2

We shall show that every fixed point of T coincides with an equilibrium point in stationary policies. Let (π_*,σ_*) be a fixed point of T. Then, $v^{1,\alpha}(\pi_*^\infty,\sigma_*^\infty) \geq v^{1,\alpha}(\pi^\infty,\sigma_*^\infty)$ for all $\pi \in \Pi$ and $v^{2,\alpha}(\pi_*^\infty,\sigma_*^\infty) \geq v^{2,\alpha}(\pi_*^\infty,\sigma^\infty)$ for all $\sigma \in \Sigma$. Trivially, the reverse statement also holds. □

Theorem 10.50

The following assertions are equivalent:

(1) $(\overline{\pi}^\infty,\overline{\rho}^\infty)$ *is a pair of stationary equilibrium policies;*

(2) *For each* $i \in S$, *the pair* $\big(\overline{\pi}(i),\overline{\rho}(i)\big)$, *with components* $\overline{\pi}_{ia},\ a \in A(i)$ *and* $\overline{\rho}_{ib},\ b \in B(i)$, *is an equilibrium point in the static bimatrix game* $\big(M^1[i],M^2[i]\big)$, *where for* $k=1,2$ *and* $a \in A(i),\ b \in B(i)$, $m^k_{ab}[i] := r^k_i(a,b) + \alpha\sum_{j=1}^N p_{ij}(a,b)v^{k,\alpha}_j(\overline{\pi}^\infty,\overline{\rho}^\infty)$, *where* $m^k_{ab}[i]$ *is the* (a,b)*th entry of the matrix* $M^k[i]$.

Proof

If (1) is true, i.e. $v^{1,\alpha}(\overline{\pi}^\infty,\overline{\rho}^\infty) \geq v^{1,\alpha}(R_1,\overline{\rho}^\infty)$ for all policies R_1 for player 1. Therefore, $\overline{\pi}^\infty$ is optimal for MDP$(\overline{\rho})$. Let $Car\big(\overline{\pi}(i)\big) := \{a \in A(i) \mid \overline{\pi}_{ia} > 0\}$. We know from the theory of MDPs that for all $\overline{a} \in Car\big(\overline{\pi}(i)\big)$, we have

$$\begin{array}{rcl} v^{1,\alpha}_i(\overline{\pi}^\infty,\overline{\rho}^\infty) & = & r^1_i(\overline{a},\overline{\rho}) + \alpha\sum_{j=1}^N p_{ij}(\overline{a},\overline{\rho})v^{1,\alpha}_j(\overline{\pi}^\infty,\overline{\rho}^\infty) \\ & \geq & r^1_i(a,\overline{\rho}) + \alpha\sum_{j=1}^N p_{ij}(a,\overline{\rho})v^{1,\alpha}_j(\overline{\pi}^\infty,\overline{\rho}^\infty) \text{ for all } a \in A(i) \end{array}$$

So, $\overline{\pi}(i)$ is the best answer to $\overline{\rho}$ in the matrix game $M^1[i]$. Similarly, $\overline{\rho}(i)$ is the best answer to $\overline{\pi}$ in the matrix game $M^2[i]$. Hence, the pair $\big(\overline{\pi}(i),\overline{\rho}(i)\big)$ is an equilibrium point in the static bimatrix game $\big(M^1[i],M^2[i]\big)$.

Conversely, assume that (2) is true. Then, by the definition of equilibrium point, we obtain

$$v^{1,\alpha}_i(\overline{\pi}^\infty,\overline{\rho}^\infty) \geq r^1_i(\pi,\overline{\rho}) + \alpha\sum_{j=1}^N p_{ij}(\pi,\overline{\rho})v^{1,\alpha}_j(\overline{\pi}^\infty,\overline{\rho}^\infty),\ i \in S.$$

Hence, $v^{1,\alpha}(\overline{\pi}^\infty,\overline{\rho}^\infty) \geq v^{1,\alpha}(\pi^\infty,\overline{\rho}^\infty)$ for all stationary policies π^∞ for player 1. From the theory of MDPs we know that then also $v^{1,\alpha}(\overline{\pi}^\infty,\overline{\rho}^\infty) \geq v^{1,\alpha}(R_1,\overline{\rho}^\infty)$ for all policies R_1 for player 1. By similar arguments it follows that $v^{2,\alpha}(\overline{\pi}^\infty,\overline{\rho}^\infty) \geq v^{2,\alpha}(\overline{\pi}^\infty,R_2)$ for all policies R_2 for player 2. Therefore, $(\overline{\pi}^\infty,\overline{\rho}^\infty)$ is a pair of stationary equilibrium policies. □

The next corollary follows straightforwardly from the proof of Theorem 10.50.

Corollary 10.7

The pair $(\overline{\pi}^\infty,\overline{\rho}^\infty)$ *is a pair of stationary equilibrium policies if and only if for every pair* $(\overline{f}^\infty,\overline{g}^\infty)$ *of deterministic equilibrium policies with* $\overline{f}(i) \in Car\big(\overline{\pi}(i)\big)$ *and* $\overline{g}(i) \in Car\big(\overline{\rho}(i)\big)$ *for each* $i \in S$ *it holds that:*

(1) $v^{1,\alpha}(\overline{f}^\infty,\overline{\rho}^\infty) \geq v^{1,\alpha}(R_1,\overline{\rho}^\infty)$ *for all policies* R_1 *for player 1;*

(2) $v^{2,\alpha}(\overline{\pi}^\infty,\overline{g}^\infty) \geq v^{2,\alpha}(\overline{\pi}^\infty,R_2)$ *for all policies* R_2 *for player 2.*

10.5.3 Single-controller stochastic games

As we have seen in Section 10.1.4, there is an interesting connection between quadratic programming and bimatrix games. In particular, quadratic program (10.12) has a global maximum of zero and the optimal solution gives equilibrium points of the bimatrix game in question. In this section we generalize the above result to the class of two-person, general-sum, single-controller stochastic games. The results apply to models with both discounted and average reward criteria.

In the single-controller stochastic game is player 1 the single-controller. This means that the transition probabilities $p_{ij}(a,b)$ are independent of b. Therefore, we denote these probabilities as $p_{ij}(a)$. Then, a stationary policy π^∞ for player 1 defines a Markov chain $P(\pi)$ with elements $p_{ij}(\pi) := \sum_a p_{ij}(a)\pi_{ia},\ i.j \in S$. Furthermore, when player 2 has stationary policy σ^∞, the discounted and average rewards for player k are $v^{k,a}(\pi^\infty,\sigma^\infty) = \{I - \alpha P(\pi)\}^{-1}r^k(\pi,\sigma)$ and $\phi^k(\pi^\infty,\sigma^\infty) = P^*(\pi)r^k(\pi,\sigma)$, respectively, for $k = 1,2$.

Discounted rewards

It is well known that an equilibrium point in stationary policies is also an equilibrium point in the space of all policies. With the original game Γ with payoffs $v^{1,a}(\pi^\infty,\sigma^\infty)$ and $v^{2,a}(\pi^\infty,\sigma^\infty)$, we can associate the game $\overline{\Gamma}$ with payoffs $\overline{v}^{1,a}(\pi^\infty,\sigma^\infty)$ and $\overline{v}^{2,a}(\pi^\infty,\sigma^\infty)$, defined as follows: $\overline{v}^{1,a}(\pi^\infty,\sigma^\infty) := v^{1,a}(\pi^\infty,\sigma^\infty)$ and $\overline{v}^{2,a}(\pi^\infty,\sigma^\infty) := r^2(\pi,\sigma)$.

Lemma 10.35

The games Γ and $\overline{\Gamma}$ have the same set of equilibrium points.

Proof

Let $(\overline{\pi}^\infty,\overline{\sigma}^\infty)$ be an equilibrium point of $\overline{\Gamma}$, i.e. $v^{1,\alpha}(\overline{\pi}^\infty,\overline{\sigma}^\infty) \geq v^{1,\alpha}(\pi^\infty,\overline{\sigma}^\infty)$ for all $\pi \in \Pi$ and $r^2(\overline{\pi},\overline{\sigma}) \geq r^2(\overline{\pi},\sigma)$ for all $\sigma \in \Sigma$. Hence, for all $\sigma \in \Sigma$, we can write

$$v^{2,\alpha}(\overline{\pi}^\infty,\overline{\sigma}^\infty) = \{I - \alpha P(\overline{\pi})\}^{-1}r^2(\overline{\pi},\overline{\sigma}) \geq \{I - \alpha P(\overline{\pi})\}^{-1}r^2(\overline{\pi},\sigma) = v^{2,\alpha}(\overline{\pi}^\infty,\sigma^\infty).$$

Therefore, $(\overline{\pi}^\infty,\overline{\sigma}^\infty)$ be an equilibrium point of Γ.

Conversely, let $(\overline{\pi}^\infty,\overline{\sigma}^\infty)$ be an equilibrium point of Γ, i.e. $v^{1,\alpha}(\overline{\pi}^\infty,\overline{\sigma}^\infty) \geq v^{1,\alpha}(\pi^\infty,\overline{\sigma}^\infty)$ for all $\pi \in \Pi$ and $v^{2,\alpha}(\overline{\pi}^\infty,\overline{\sigma}^\infty) \geq v^{2,\alpha}(\overline{\pi}^\infty,\sigma^\infty)$ for all $\sigma \in \Sigma$. Since $\overline{\sigma}^\infty$ is an optimal policy for MDP$(\overline{\pi})$ with rewards $r^2_i(\overline{\pi},b)$ for all $(i,b) \in S \times B$, it follows from the method of policy iteration that $v^{2,\alpha}(\overline{\pi}^\infty,\overline{\sigma}^\infty) = max_\sigma\, \{r^2_i(\overline{\pi},\sigma) + \alpha \sum_j p_{ij}(\overline{\pi})v^{2,\alpha}_j(\overline{\pi}^\infty,\overline{\sigma}^\infty)\}$. Hence, we obtain for all $\sigma \in \Sigma$

$$r^2(\overline{\pi},\overline{\sigma}) + \alpha P(\overline{\pi})v^{2,\alpha}(\overline{\pi}^\infty,\overline{\sigma}^\infty)v^{2,\alpha}(\overline{\pi}^\infty,\sigma).$$

Therefore, $r^2(\overline{\pi},\overline{\sigma}) \geq r^2(\overline{\pi},\sigma)$ for all $\sigma \in \Sigma$, and consequently, $(\overline{\pi}^\infty,\overline{\sigma}^\infty)$ be an equilibrium point of $\overline{\Gamma}$. □

Lemma 10.36

Let $E(\overline{\pi}) := \{\overline{\sigma} \in \Sigma \mid (\overline{\pi}^\infty,\overline{\sigma}^\infty)$ is an equilibrium point of $\Gamma\}$. Then, $E(\overline{\pi})$ is convex.

Proof

Suppose that $\overline{\sigma}_1, \overline{\sigma}_2 \in E(\overline{\pi})$, and let $0 \le \lambda \le 1$. Then, for $\overline{\sigma} := \lambda\overline{\sigma}_1 + (1-\lambda)\overline{\sigma}_2$, we have $\overline{\sigma} \in \Sigma$ and for $k = 1, 2$, we obtain

$$\begin{array}{rcl} v^{k,\alpha}(\overline{\pi}^\infty, \overline{\sigma}^\infty) & = & \{I - \alpha P(\overline{\pi}\}^{-1} r^k(\overline{\pi}, \overline{\sigma}) \\ & = & \lambda\{I - \alpha P(\overline{\pi}\}^{-1} r^k(\overline{\pi}, \overline{\sigma}_1) + (1-\lambda)\{I - \alpha P(\overline{\pi}\}^{-1} r^k(\overline{\pi}, \overline{\sigma}_2) \\ & = & \lambda v^{k,\alpha}(\overline{\pi}^\infty, \overline{\sigma}_1^\infty) + (1-\lambda) v^{k,\alpha}(\overline{\pi}^\infty, \overline{\sigma}_2^\infty). \end{array}$$

Since $v^{1,\alpha}(\overline{\pi}^\infty, \overline{\sigma}^\infty)$ and $v^{2,\alpha}(\overline{\pi}^\infty, \overline{\sigma}^\infty)$ are linear in $\overline{\sigma}$, the lemma follows. □

Theorem 10.51

Let $(\overline{\pi}, \overline{\sigma})$ be an equilibrium point, and let $v^1 := v^{1,\alpha}(\overline{\pi}^\infty, \overline{\sigma}^\infty)$. Furthermore, let $\overline{\sigma}$ an extreme point of $E(\overline{\pi})$. Then, $(v^1, \overline{\sigma})$is an extreme solution of the following linear system:

(1) $\sum_j \{\delta_{ij} - \alpha p_{ij}(a)\} v_j \ge \sum_b r_i^1(a,b)\sigma_{ib},\ (i,a) \in S \times A.$

(2) $\sum_b r_i^2(\overline{\pi}, b) \ge r_i^2(\overline{\pi}, b),\ (i,b) \in S \times B.$

(3) $\sum_b \sigma_{ib} = 1,\ i \in S.$

(4) $\sigma_{ib} \ge 0,\ (i,b) \in S \times B.$

Proof

We first show that $(v^1, \overline{\sigma})$ is a feasible solution of the linear system. Since v^1 is the value of MDP(σ) with rewards $r_i^1(a, \overline{\sigma}),\ (i,a) \in S \times A$, it follows from the linear programming method for MDPs that (1) is satisfied. Obviously, $\overline{\sigma}$ satisfies (3) and (4). Therefore, we have to prove the inequalities of (2), i.e. $r_i^2(\overline{\pi}, \overline{\sigma}) \ge r_i^2(\overline{\pi}, b)$ for all $(i,b) \in S \times B$. Since $(\overline{\pi}^\infty, \overline{\sigma}^\infty)$ is an equilibrium point of Γ, we have $r^2(\overline{\pi}, \overline{\sigma}) \ge r^2(\overline{\pi}, \sigma)$ for all $\sigma \in \Sigma$. Take any $(i_*, b_*) \in S \times B$ and let σ such that $\sigma_{ib} = 1$ for $(i,b) = (i_*, b_*)$ and $\sigma_{ib} = 0$ for $(i,b) \ne (i_*, b_*)$. Then, $r_{i_*}^2(\overline{\pi}, \overline{\sigma}) \ge r_{i_*}^2(\overline{\pi}, b_*)$; so, also (2) is satisfied.

Suppose that $(v^1, \overline{\sigma})$ is not an extreme solution. Then, $(v^1, \overline{\sigma}) = \frac{1}{2}(w^1, \overline{\sigma}_1) + \frac{1}{2}(w^2, \overline{\sigma}_2)$, where $(w^1, \overline{\sigma}_1)$ and $(w^2, \overline{\sigma}_2)$ are also feasible solutions of the linear system. From (1) it follows that $\{I - \alpha P(\pi)\} w^k \ge r^1(\pi, \overline{\sigma}_k)$ for all $\pi \in \Pi$ and for $k = 1, 2$. Hence,

$$w^k \ge \{I - \alpha P(\pi)\}^{-1} r^1(\pi, \overline{\sigma}_k) = v^{1,\alpha}(\pi^\infty, \overline{\sigma}_k^\infty) \text{ for all } \pi \in \Pi \text{ and for } k = 1, 2.$$

Now, we can write

$$\begin{array}{rcl} v^1 & = & \frac{1}{2}(w^1 + w^2) \\ & \ge & \frac{1}{2} \cdot max_\pi\, v^{1,\alpha}(\pi^\infty, \overline{\sigma}_1^\infty) + \frac{1}{2} \cdot max_\pi\, v^{1,\alpha}(\pi^\infty, \overline{\sigma}_2^\infty) \\ & \ge & \frac{1}{2} \cdot max_\pi \{v^{1,\alpha}(\pi^\infty, \overline{\sigma}_1^\infty) + v^{1,\alpha}(\pi^\infty, \overline{\sigma}_2^\infty)\} \\ & = & \frac{1}{2} \cdot max_\pi \big\{\{I - \alpha P(\pi)\}^{-1} r^1(\pi, \overline{\sigma}_1) + \{I - \alpha P(\pi)\}^{-1} r^1(\pi, \overline{\sigma}_2)\big\} \\ & = & max_\pi \{I - \alpha P(\pi)\}^{-1} r^1(\pi, \overline{\sigma}) \ = \ max_\pi\, v^{1,\alpha}(\pi^\infty, \overline{\sigma}^\infty) \ = \ v^1. \end{array}$$

Hence, $v^1 = w^1 = w^2 = max_\pi\, v^{1,\alpha}(\pi^\infty, \overline{\sigma}_1^\infty) = max_\pi\, v^{1,\alpha}(\pi^\infty, \overline{\sigma}_2^\infty)$. From the linear constraints (3) of the linear system it follows that $r^2(\overline{\pi}, \overline{\sigma}_1) = max_\sigma\, r^2(\overline{\pi}, \sigma)$. Therefore, $(\overline{\pi}, \overline{\sigma}_1)$ is an equilibrium point, i.e. $\overline{\sigma}_1 \in E(\overline{\pi})$. Similarly can be shown that $\overline{\sigma}_2 \in E(\overline{\pi})$. Since $\overline{\sigma}_1, \overline{\sigma}_2 \in E(\overline{\pi})$ and $\overline{\sigma} = \frac{1}{2}\overline{\sigma}_1 + \frac{1}{2}\overline{\sigma}_2$, $\overline{\sigma}$ is not an extreme point of $E(\overline{\pi})$, which gives a contradiction. □

Any extreme solution of a system of linear (in)equalities is the solution of a nonsingular square system of equated constraints (for a proof of this property see Theorem 9 in [108]). Therefore, any extreme solution lies in the same ordered field as that of the entries. Thus, by Theorem 10.51, we have the following result.

Theorem 10.52
For any general-sum, single-controller stochastic game with discounted rewards, there exists a pair of equilibrium stationary policies with entries lying in the same ordered field as that of the entries of the stochastic game.

This result indicates that finite algorithms for computing an equilibrium point may exist. For zero-sum games, we have seen in previous sections of this chapter that the single-controller and switching-controller cases possess the ordered field property and that finite algorithms exist for the computation of optimal policies.

It is clear that with $\overline{\sigma} \in \Sigma$ held fixed, the problem of finding $v^{1,\alpha}(\overline{\sigma}^{\infty}) = max_{R_1} v^{1,\alpha}(R_1, \overline{\sigma}^{\infty})$ is exactly the discounted reward MDP which can be solved with the help of the following pair of primal and dual linear programs:

$$min \left\{ \sum_j \beta_j v_j \;\middle|\; \sum_j \{\delta_{ij} - \alpha p_{ij}(a)\} v_j \geq r^1(a, \overline{\sigma}),\ (i,a) \in S \times A \right\}, \tag{10.121}$$

where $\beta_j > 0,\ j \in S$, is arbitrarily chosen, and

$$max \left\{ \sum_{(i,a)} r_i^1(a, \overline{\sigma}) x_i(a) \;\middle|\; \begin{array}{rcl} \sum_{(i,a)} \{\delta_{ij} - \alpha p_{ij}(a)\} x_i(a) & = & \beta_j,\ j \in S \\ x_i(a) & \geq & 0,\ (i,a) \in S \times A \end{array} \right\}. \tag{10.122}$$

In section 3.5 we have showed that if $\overline{v}$ and $\overline{x}$ are optimal solutions of the problems (10.121) and (10.122), respectively, then $\overline{v} = v^{1,\alpha}(\overline{\pi}^{\infty}, \overline{\sigma}^{\infty})$, where $\overline{\pi} \in \Pi$ being appropriately constructed by

$$\overline{\pi}_{ia} := \frac{\overline{x}_i(a)}{\sum_a \overline{x}_i(a)},\ (i,a) \in S \times A. \tag{10.123}$$

We now introduce the quadratic program:

$max \left\{ \sum_{(i,a,b)} \{r_i^1(a,b) + r_i^2(a,b)\} \cdot \sigma_{ib} \cdot x_i(a) - \sum_i \beta_i v_i + \sum_i z_i \right\}$

subject to

(1) $\sum_j \{\delta_{ij} - \alpha p_{ij}(a)\} \cdot v_j \geq \sum_b r^1(a,b) \cdot \sigma_{ib},\ (i,a) \in S \times A.$

(2) $\sum_{(i,a)} \{\delta_{ij} - \alpha p_{ij}(a)\} \cdot x_i(a) = \beta_j,\ j \in S.$

(3) $\sum_a r_i^2(a,b) x_i(a) + z_i \leq 0,\ (i,b) \in S \times B.$

(4) $\sum_b \sigma_{ib} = 1,\ i \in S.$

(5) $x_i(a) \geq 0,\ (i,a) \in S \times A.$

(6) $\sigma_{ib} \geq 0,\ (i,b) \in S \times B.$

This quadratic program may be considered as a generalization of the dual pair of linear programs (10.31) and (10.32 in which the objective function and the right-hand-side of the jth equality constraint are replaced by $\sum_i \beta_i v_i$, (instead of $\sum_i v_i$) and β_j instead of 1; furthermore, we use σ instead of ρ and r^1 instead of r.

We know from the theory of linear programs that if these programs have feasible solutions $(\overline{v}, \overline{\sigma})$ and $(\overline{x}, \overline{z})$ satisfying $\sum_i \beta_i \overline{v}_i = \sum_i \overline{z}_i$, then these solutions are both optimal. Note that the feasibility of (10.31) is equivalent to the conditions (1), (4) and (6); the feasibility of (10.32) corresponds to (2), (3) with $r_i^2(a,b) = -r_i^1(a,b)$ for all (i,a,b), and (5); furthermore, with $r_i^2(a,b) = -r_i^1(a,b)$ for all (i,a,b), the condition $\sum_i \beta_i \overline{v}_i = \sum_i \overline{z}_i$ is the same as value 0 for the objective function of the quadratic program.

Let $(\overline{\pi}^\infty, \overline{\sigma}^\infty)$ be an equilibrium point of $\overline{\Gamma}$. Then, $v^{1,\alpha}(\overline{\pi}^\infty, \overline{\sigma}^\infty) \geq v^{1,\alpha}(\pi^\infty, \overline{\sigma}^\infty)$ for all $\pi \in \Pi$, i.e. $\overline{\pi}^\infty$ an optimal policy in MDP$(\overline{\sigma})$. Furthermore, we know from Section 3.5 that $\overline{x}$, defined by $\overline{x}_i(a) := \left\{\beta^T\{I - \alpha P(\overline{\pi}\}^{-1}\right\}_i \cdot \overline{\pi}_{ia}$ for all $(i,a) \in S \times A$, is an optimal solution of (10.122). Define $\overline{z}$ by $\overline{z}_i := -\sum_{(a,b)} r_i^2(a,b) \cdot \overline{\sigma}_{ib} \cdot \overline{x}_i(a) = -\sum_a r_i^2(a, \overline{\sigma}) \cdot \overline{x}_i(a),\ i \in S$. Let $\overline{v}$ be an optimal solution of (10.121).

Theorem 10.53

(1) *Let $(\overline{\sigma}, \overline{v}, \overline{x}, \overline{z})$ be defined as described above. Then, $(\overline{\sigma}, \overline{v}, \overline{x}, \overline{z})$ is an optimal solution of the quadratic program with value 0.*

(2) *Let $(\sigma^*, v^*, x^*, z^*)$ be an optimal solution of the quadratic program. Then, $\big((\pi^*)^\infty, (\sigma^*)^\infty\big)$, where π^* is defined by $\pi^*_{ia} := \frac{x_i^*(a)}{\sum_a x_i^*(a)}$, $(i,a) \in S \times A$, is an equilibrium point of $\overline{\Gamma}$.*

Proof

(1) We first show that $(\overline{\sigma}, \overline{v}, \overline{x}, \overline{z})$ is a feasible solution of the quadratic program. Therefore, we have to show (the other constraints are trivially satisfied) that $\sum_a r_i^2(a,b) \cdot \overline{x}_i(a) + \overline{z}_i \leq 0$ for all $(i,b) \in S \times B$, i.e. $\sum_a r_i^2(a, \overline{\sigma}) \cdot \overline{x}_i(a) \geq \sum_a r_i^2(a,b) \cdot \overline{x}_i(a)$ for all $(i,b) \in S \times B$.

Let $\overline{x}_i := \sum_a \overline{x}_i(a),\ i \in S$. Then, $\overline{x}_i(a) = \overline{x}_i \cdot \overline{\pi}_{ia},\ (i,a) \in S \times A$ and $\overline{x}_i > 0,\ i \in S$.

Select any pair $(i,b) \in S \times B$. We have to show $\sum_a r_i^2(a, \overline{\sigma}) \cdot \overline{x}_i \cdot \overline{\pi}_{ia} \geq \sum_a r_i^2(a,b) \cdot \overline{x}_i \cdot \overline{\pi}_{ia}$, i.e. $r_i^2(\overline{\pi}, \overline{\sigma}) \cdot \overline{x}_i \geq r_i^2(\overline{\pi}, b) \cdot \overline{x}_i$. Since $\overline{x}_i > 0$, we have to show $r_i^2(\overline{\pi}, \overline{\sigma}) \geq r_i^2(\overline{\pi}, b)$, which follows directly from the property that $(\overline{\pi}^\infty, \overline{\sigma}^\infty)$ is an equilibrium point of $\overline{\Gamma}$.

We have $\sum_j \beta_j \overline{v}_j = \sum_{(i,a)} r_i^1(a, \overline{\sigma}) \cdot \overline{x}_i(a) = \sum_{(i,a,b)} r_i^1(a,b) \cdot \overline{\sigma}_{ib} \cdot \overline{x}_i(a)$, because the optima of (10.121) and (10.122) are equal. Furthermore, $\sum_i \overline{z}_i = \sum_{(i,a,b)} r_i^2(a,b) \cdot \overline{\sigma}_{ib} \cdot \overline{x}_i(a)$. Hence, the value of the objective function for the feasible solution $(\overline{\sigma}, \overline{v}, \overline{x}, \overline{z})$ equals 0.

Let (σ, v, x, z) be any feasible solution of the quadratic program. Since (σ, v) and x are feasible for (10.121) and (10.122) respectively, we have

$$\textstyle\sum_j \beta_j\ v_j \geq \sum_{(i,a)} r_i^1(a, \sigma) \cdot x_i(a) = \sum_{(i,a,b)} r_i^1(a,b) \cdot \sigma_{ib} \cdot x_i(a).$$

By the summation of (3) over all $(i,b) \in S \times B$, we obtain

$$\textstyle\sum_{(i,a,b)} r_i^2(a,b) \cdot \sigma_{ib} \cdot x_i(a) + \sum_i z_i = \sum_{(i,a)} r_i^2(a, \sigma) \cdot x_i(a) + \sum_i z_i \leq 0.$$

Hence,

$$\textstyle\sum_{(i,a,b)} \{r_i^1(a,b) + r_i^2(a,b)\} \cdot \sigma_{ib} \cdot x_i(a) - \sum_j \beta_j\ v_j + \sum_i z_i \leq 0.$$

Therefore, $(\overline{\sigma}, \overline{v}, \overline{x}, \overline{z})$ is an optimal solution of the quadratic program with value 0.

(2) Let $(\sigma^*, v^*, x^*, z^*)$ be an optimal solution of the quadratic program. Then,

$$\begin{array}{rcl} 0 & = & \sum_{(i,a,b)} \{r_i^1(a,b) + r_i^2(a,b)\} \cdot \sigma_{ib}^* \cdot x_i^*(a) - \sum_j \beta_j v_j^* + \sum_i z_i^* \\ & = & \{\sum_{(i,a)} r_i^1(a,\sigma^*) \cdot x_i^*(a) - \sum_j \beta_j v_j^*\} + \sum_i \{z_i^* + \sum_a r_i^2(a,\sigma^*) \cdot x_i^*(a)\}. \end{array}$$

Since v^* and x^* are feasible solutions of the modifications of (10.121) and (10.122), where $\overline{\sigma}$ is replaced by σ^*, we have $\{\sum_{(i,a)} r_i^1(a,\sigma^*) \cdot x_i^*(a) - \sum_j \beta_j v_j^*\} \leq 0$. Furthermore, from (3) it follows that also $z_i^* + \sum_a r_i^2(a,\sigma^*) \cdot x_i^*(a) \leq 0$ for all $i \in S$. Hence,

$$\sum_{(i,a)} r_i^1(a,\sigma^*) \cdot x_i^*(a) = \sum_j \beta_j v_j^* \text{ and } z_i^* = -\sum_a r_i^2(a,\sigma^*) \cdot x_i^*(a) \text{ for all } i \in S. \tag{10.124}$$

Because $\sum_{(i,a)} r_i^1(a,\sigma^*) \cdot x_i^*(a) = \sum_j \beta_j v_j^*$, the feasible solutions v^* and x^* are optimal for the modifications of (10.121) and (10.122) with σ^* instead of $\overline{\sigma}$. Hence, $(\pi^*)^\infty$, defined by $\pi_{ia}^* := \frac{x_i^*(a)}{\sum_a x_i^*(a)}$ for all $(i,a) \in S \times A$ is an optimal policy for MDP(σ^*) and consequently,

$$v^{1,\alpha}\big((\pi^*)^\infty, (\sigma^*)^\infty\big) \geq v^{1,\alpha}\big(\pi^\infty, (\sigma^*)^\infty\big) \text{ for all } \pi \in \Pi.$$

From (3), (10.124) and the definition of π^* it follows that

$\sum_a r_i^2(a,b) \cdot \pi_{ia}^* \cdot x_i^* - \sum_a r_i^2(a,\sigma^*) \cdot \pi_{ia}^* \cdot x_i^* \leq 0$ for all $(i,b) \in S \times B$, where $x_i^* := \sum_a x_i^*(a)$, $i \in S$. Because $x_i^* > 0$ for all $i \in S$, we obtain $r_i^2(\pi^*, b) - r_i^2(\pi^*, \sigma^*) \leq 0$ for all $(i,b) \in S \times B$. Hence,

$$v^{2,\alpha}\big((\pi^*)^\infty, (\sigma^*)^\infty\big) = r_i^2(\pi^*, \sigma^*) \geq r_i^2(\pi^*, \sigma) = v^{2,\alpha}\big((\pi^*)^\infty, \sigma^\infty\big) \text{ for all } \sigma \in \Sigma.$$

Therefore, we have shown that $\big((\pi^*)^\infty, (\sigma^*)^\infty\big)$ is an equilibrium point of $\overline{\Gamma}$. □

Average rewards

Most of the results for discounted rewards can be transformed to average rewards. Let Γ be the original game with payoffs $\phi^1(\pi^\infty, \sigma^\infty)$ and $\phi^2(\pi^\infty, \sigma^\infty)$ for player 1 and player 2, respectively. Define the associated game $\overline{\Gamma}$ with payoffs $\overline{\phi}^1(\pi^\infty, \sigma^\infty)$ and $\overline{\phi}^2(\pi^\infty, \sigma^\infty)$ for player 1 and player 2, respectively, by $\overline{\phi}^1(\pi^\infty, \sigma^\infty) := \phi^1(\pi^\infty, \sigma^\infty)$ and $\overline{\phi}^2(\pi^\infty, \sigma^\infty) := r^2(\pi, \sigma)$.

Lemma 10.37

If $(\overline{\pi}^\infty, \overline{\sigma}^\infty)$ is an equilibrium point of Γ, then $(\overline{\pi}^\infty, \overline{\sigma}^\infty)$ is also an equilibrium point of Γ.

Proof

Let $(\overline{\pi}^\infty, \overline{\sigma}^\infty)$ is an equilibrium point of Γ, i.e. $\phi^1(\overline{\pi}^\infty, \overline{\sigma}^\infty) \geq \phi^1(\pi^\infty, \overline{\sigma}^\infty)$ for all $\pi \in \Pi$ and $r^2(\overline{\pi}^\infty, \overline{\sigma}^\infty) \geq r^2(\overline{\pi}, \sigma)$ for all $\sigma \in \Sigma$. Hence, for all $\sigma \in \Sigma$, we can write

$$\phi^2(\overline{\pi}^\infty, \overline{\sigma}^\infty) = P^*(\overline{\pi}) r^1(\overline{\pi}, \overline{\sigma}) \geq P^*(\overline{\pi}) r^2(\overline{\pi}, \sigma) = \phi^2(\overline{\pi}^\infty, \sigma^\infty).$$

Therefore, $(\overline{\pi}^\infty, \overline{\sigma}^\infty)$ is also an equilibrium point of Γ. □

Theorem 10.54

Let π^∞ and σ^∞ be stationary policies for player 1 and 2, respectively. Then, for $k = 1$ and 2, $\phi^k(\pi^\infty, \sigma^\infty) = \lim_{\alpha \uparrow 1} (1-\alpha) v^{k,\alpha}(\pi^\infty, \sigma^\infty)$.

Proof
$\phi^k(\pi^\infty,\sigma^\infty) = P^*(\pi)r^k(\pi,\sigma)$. Since $P^*(\pi)$ is the Cesaro limit of $P^t(\pi)$, it is also the Abel limit, which implies $P^*(\pi)r^k(\pi,\sigma) = \lim_{\alpha\uparrow 1}(1-\alpha)\sum_{t=0}^\infty\{\alpha P(\pi)\}^t r^k(\pi,\sigma) = \lim_{\alpha\uparrow 1}(1-\alpha)v^{k,\alpha}(\pi^\infty,\sigma^\infty)$ for $k=1$ and $k=2$. □

Let $\big(\overline{\pi}^\infty(\alpha),\overline{\sigma}^\infty(\alpha)\big)$ be an equilibrium point for discount factor α and define the vector $v^1(\alpha)$ by $v^1(\alpha) := v^{1,\alpha}\big(\overline{\pi}^\infty(\alpha),\overline{\sigma}^\infty(\alpha)\big)$. Using Theorem 10.51, Theorem 9 in [108] and the property that $\sigma(\alpha)$ and $v^1(\alpha)$ are rational functions in α, the following result can be shown (for details see [212]).

Theorem 10.55
For any general-sum, single-controller stochastic game with average rewards, there exists a pair of equilibrium stationary policies with entries lying in the same ordered field as that of the entries of the stochastic game.

It is well known that an equilibrium point in stationary policies is also an equilibrium point in the space of all policies. Therefore, we may restrict the set of policies to the set of stationary policies.

It is clear that with $\overline{\sigma}\in\Sigma$ held fixed, the problem of finding $\phi^1(\overline{\sigma}) := max_{R_1}\,\phi^1(R_1,\overline{\sigma}^\infty)$ is exactly the average reward MDP which can be solved with the help of the following pair of primal and dual linear programs:

$$min\left\{\sum_j \beta_j v_j \;\middle|\; \begin{array}{rcll} \sum_j\{\delta_{ij}-p_{ij}(a)\}v_j & \geq & 0, & (i,a)\in S\times A \\ v_i+\sum_j\{\delta_{ij}-p_{ij}(a)\}t_j & \geq & r_i^1(a,\overline{\sigma}) & (i,a)\in S\times A \end{array}\right\}, \tag{10.125}$$

where $\beta_j>0,\ j\in S$, is arbitrarily chosen and

$$max\left\{\sum_{(i,a)} r_i^1(a,\overline{\sigma})x_i(a) \;\middle|\; \begin{array}{rcl} \sum_{(i,a)}\{\delta_{ij}-p_{ij}(a)\}x_i(a) & = & 0,\ j\in S \\ \sum_a x_j(a)+\ \sum_{(i,a)}\{\delta_{ij}-p_{ij}(a)\}y_i(a) & = & \beta_j,\ j\in S \\ x_i(a),y_i(a) & \geq & 0,\ (i,a)\in S\times A \end{array}\right\} \tag{10.126}$$

In section 5.8 we have showed the following. If $(\overline{v},\overline{t})$ and $(\overline{x},\overline{y})$ are optimal solutions of the problems (10.125) and (10.126) respectively, then $\overline{v}=\phi^1(\overline{\sigma})=\phi^1(\overline{\pi}^\infty,\overline{\sigma}^\infty)$, where $\overline{\pi}$ being appropriately constructed from $(\overline{x},\overline{y})$ by

$$\overline{\pi}_{ia} := \begin{cases} \frac{\overline{x}_i(a)}{\sum_a\overline{x}_i(a)}, & a\in A(i),\ i\in S_{\overline{x}}; \\ \frac{\overline{y}_i(a)}{\sum_a\overline{y}_i(a)}, & a\in A(i),\ i\notin S_{\overline{x}}, \end{cases} \tag{10.127}$$

where $S_{\overline{x}} := \{i \mid \sum_a \overline{x}_i(a)>0\}$. We now introduce the quadratic program

$max\,\{\sum_{(i,a,b)}\{r_i^1(a,b)+r_i^2(a,b)\}\cdot\sigma_{ib}\cdot x_i(a)-\sum_i\beta_i v_i+\sum_i z_i\}$

subject to

(1) $\sum_j \{\delta_{ij} - p_{ij}(a)\} \cdot v_j \geq 0,\ (i,a) \in S \times A.$

(2) $v_i + \sum_j \{\delta_{ij} - p_{ij}(a)\} \cdot t_j \geq \sum_b r^1(a,b) \cdot \sigma_{ib},\ (i,a) \in S \times A.$

(3) $\sum_{(i,a)} \{\delta_{ij} - p_{ij}(a)\} \cdot x_i(a) = 0,\ j \in S.$

(4) $\sum_a x_j(a) + \sum_{(i,a)} \{\delta_{ij} - p_{ij}(a)\} \cdot y_i(a) = \beta_j,\ j \in S.$

(5) $\sum_a r^2_i(a,b) \cdot x_i(a) + z_i \leq 0,\ (i,b) \in S \times B.$

(6) $\sum_b \sigma_{ib} = 1,\ i \in S.$

(7) $x_i(a), y_i(a) \geq 0,\ (i,a) \in S \times A.$

(8) $\sigma_{ib} \geq 0,\ (i,b) \in S \times B.$

This quadratic program may be considered as a generalization of the dual pair of linear programs (10.73) and (10.74 in which the objective function and the right-hand-side of the jth equality in the second set of constraints are replaced by $\sum_i \beta_i v_i$, (instead of $\sum_i v_i$) and β_j instead of 1; furthermore, we use σ instead of ρ and r^1 instead of r.

We know from the theory of linear programs that if these programs have feasible solutions $(\overline{v}, \overline{t}, \overline{\sigma})$ and $(\overline{x}, \overline{y}, \overline{z})$ satisfying $\sum_i \beta_i \overline{v}_i = \sum_i \overline{z}_i$, then these solutions are both optimal. Note that the feasibility of (10.73) is equivalent to the conditions (1),(2), (6) and (8); the feasibility of (10.74) corresponds to (3), (4) with $r^2_i(a,b) = -r^1_i(a,b)$ for all (i,a,b), and (7); furthermore, with $r^2_i(a,b) = -r^1_i(a,b)$ for all (i,a,b), the condition $\sum_i \beta_i \overline{v}_i = \sum_i \overline{z}_i$ is the same as value 0 for the objective function of the quadratic program.

Let $(\overline{\pi}^\infty, \overline{\sigma}^\infty)$ be an equilibrium point of $\overline{\Gamma}$. Then, $\phi^1(\overline{\pi}^\infty, \overline{\sigma}^\infty) \geq \phi^1(\pi^\infty, \overline{\sigma}^\infty)$ for all $\pi \in \Pi$, i.e. $\overline{\pi}^\infty$ an optimal policy in MDP$(\overline{\sigma})$. Furthermore, we know from Section 5.8 that $(\overline{x}, \overline{y})$ is an optimal solution of (10.126), where $(\overline{x}, \overline{y})$ is defined by $\overline{x}_i(a) := \{\beta^T P^*(\overline{\pi})\}_i \cdot \overline{\pi}_{ia}$ for all $(i,a) \in S \times A$ and $\overline{y}_i(a) := \{\beta^T D(\overline{\pi}) + \gamma^T P^*(\overline{\pi})\}_i \cdot \overline{\pi}_{ia}$ for all $(i,a) \in S \times A$ and the vector γ is defined by

$$\gamma_i := \begin{cases} 0 & i \in T \\ max_{l \in S_j} \left\{ -\frac{\sum_{k \in S} \beta_k d_{kl}(\overline{\pi})}{\sum_{k \in S_j} p^*_{kl}(\overline{\pi})} \right\} & i \in S_j,\ 1 \leq j \leq m \end{cases}$$ with T the set of transient states and $S_j,\ 1 \leq j \leq m$, the sets of recurrent classes in the Markov chain $P(\overline{\pi})$. Let $(\overline{v}, \overline{t})$ be an optimal solution of (10.125) and define $\overline{z}$ by $\overline{z}_i := -\sum_{(a,b)} r^2_i(a,b) \cdot \overline{\sigma}_{ib} \cdot \overline{x}_i(a) = -\sum_a r^2_i(a, \overline{\sigma}) \cdot \overline{x}_i(a),\ i \in S.$

Theorem 10.56

(1) Let $(\overline{\sigma}, \overline{v}, \overline{t}, \overline{x}, \overline{y}, \overline{z})$ be defined as described above. Then, $(\overline{\sigma}, \overline{v}, \overline{t}, \overline{x}, \overline{y}, \overline{z})$ is an optimal solution of the quadratic program with value 0.

(2) Let $(\sigma^*, v^*, t^*, x^*, y^*, z^*)$ be an optimal solution of the quadratic program. Then, $\big((\pi^*)^\infty, (\sigma^*)^\infty\big)$, where π^* is defined by $\pi^*_{ia} := \begin{cases} \frac{x^*_i(a)}{\sum_a x^*_i(a)}, & a \in A(i),\ i \in S_{x^*} \\ \frac{y^*_i(a)}{\sum_a y^*_i(a)}, & a \in A(i),\ i \notin S_{x^*} \end{cases}$ with $S_{x^*} := \{i \mid \sum_a x^*_i(a) > 0\}$, is an equilibrium point of $\overline{\Gamma}$.

Proof

(1) We first show that $(\overline{\sigma}, \overline{v}, \overline{t}, \overline{x}, \overline{y}, \overline{z})$ is a feasible solution of the quadratic program. Therefore, we have to show (the other constraints are trivially satisfied) that $\sum_a r^2_i(a,b) \cdot \overline{x}_i(a) + \overline{z}_i \leq 0,$

for all $(i,b) \in S \times B$, i.e. $\sum_a r_i^2(a,\overline{\sigma}) \cdot \overline{x}_i(a) \geq \sum_a r_i^2(a,b) \cdot \overline{x}_i(a)$ for all $(i,b) \in S \times B$.
Select any pair $(i,b) \in S \times B$.
If $i \notin S_{\overline{x}}$: $\overline{x}_i(a) = 0,\ a \in A(i)$ and $\sum_a r_i^2(a,\overline{\sigma}) \cdot \overline{x}_i(a) = \sum_a r_i^2(a,b) \cdot \overline{x}_i(a) = 0$.
If $i \in S_{\overline{x}}$, then $\overline{x}_i(a) = \overline{x}_i \cdot \overline{\pi}_{ia}$ for all $a \in A(i)$, where $\overline{x}_i := \sum_a \overline{x}_i(a) > 0$. We have to show $\sum_a r_i^2(a,\overline{\sigma}) \cdot \overline{x}_i \cdot \overline{\pi}_{ia} \geq \sum_a r_i^2(a,b) \cdot \overline{x}_i \cdot \overline{\pi}_{ia}$, i.e. $r_i^2(\overline{\pi},\overline{\sigma}) \geq r_i^2(\overline{\pi},b)$, which follows directly from the property that $(\overline{\pi}^\infty,\overline{\sigma}^\infty)$ is an equilibrium point of $\overline{\Gamma}$.
We have $\sum_j \beta_j \overline{v}_j = \sum_{(i,a)} r_i^1(a,\overline{\sigma}) \cdot \overline{x}_i(a) = \sum_{(i,a,b)} r_i^1(a,b) \cdot \overline{\sigma}_{ib} \cdot \overline{x}_i(a)$, because the optima of (10.125) and (10.126) are equal. Furthermore, $\sum_i \overline{z}_i = \sum_{(i,a,b)} r_i^2(a,b) \cdot \overline{\sigma}_{ib} \cdot \overline{x}_i(a)$. Hence, the value of the objective function for the feasible solution $(\overline{\sigma},\overline{v},\overline{t},\overline{x},\overline{y},\overline{z})$ equals 0.
Let (σ,v,t,x,y,z) be any feasible solution of the quadratic program. Since (σ,v,t) and (x,y) are feasible for (10.125) and (10.126) respectively, we have

$$\textstyle\sum_j \beta_j\ v_j \geq \sum_{(i,a)} r_i^1(a,\sigma) \cdot x_i(a) = \sum_{(i,a,b)} r_i^1(a,b) \cdot \sigma_{ib} \cdot x_i(a).$$

By the summation of (5) over all $(i,b) \in S \times B$, we obtain

$$\textstyle\sum_{(i,a,b)} r_i^2(a,b) \cdot \sigma_{ib} \cdot x_i(a) + \sum_i z_i = \sum_{(i,a)} r_i^2(a,\sigma) \cdot x_i(a) + \sum_i z_i \leq 0.$$

Hence,

$$\textstyle\sum_{(i,a,b)} \{r_i^1(a,b) + r_i^2(a,b)\} \cdot \sigma_{ib} \cdot x_i(a) - \sum_j \beta_j\ v_j + \sum_i z_i \leq 0.$$

Therefore, $(\overline{\sigma},\overline{v},\overline{t},\overline{x},\overline{y},\overline{z})$ is an optimal solution of the quadratic program with value 0.

(2) Let $(\sigma^*,v^*,t^*,x^*,y^*,z^*)$ be an optimal solution of the quadratic program. Then,

$$\begin{array}{rcl} 0 & = & \sum_{(i,a,b)} \{r_i^1(a,b) + r_i^2(a,b)\} \cdot \sigma_{ib}^* \cdot x_i^*(a) - \sum_j \beta_j\, v_j^* + \sum_i z_i^* \\ & = & \{\sum_{(i,a)} r_i^1(a,\sigma^*) \cdot x_i^*(a) - \sum_j \beta_j\, v_j^*\} + \sum_i \{z_i^* + \sum_a r_i^2(a,\sigma^*) \cdot x_i^*(a)\}. \end{array}$$

Since (v^*,t^*) and (x^*,y^*) are feasible solutions of the modifications of (10.125) and (10.125), where $\overline{\sigma}$ is replaced by σ^*, we have $\{\sum_{(i,a)} r_i^1(a,\sigma^*) \cdot x_i^*(a) - \sum_j \beta_j\, v_j^*\} \leq 0$.
Furthermore, from (5) it follows that also $z_i^* + \sum_a r_i^2(a,\sigma^*) \cdot x_i^*(a)\} \leq 0$ for all $i \in S$. Hence,

$$\sum_{(i,a)} r_i^1(a,\sigma^*) \cdot x_i^*(a) = \sum_j \beta_j\, v_j^* \text{ and } z_i^* = -\sum_a r_i^2(a,\sigma^*) \cdot x_i^*(a) \text{ for all } i \in S. \tag{10.128}$$

Because $\sum_{(i,a)} r_i^1(a,\sigma^*) \cdot x_i^*(a) = \sum_j \beta_j\, v_j^*$, the feasible solutions (v^*,t^*) and (x^*,y^*) are optimal for the modifications of (10.125) and (10.126) with σ^* instead of $\overline{\sigma}$. Hence, $(\pi^*)^\infty$ is an optimal policy for MDP(σ^*) and consequently,

$$\phi^1\big((\pi^*)^\infty,(\sigma^*)^\infty\big) \geq \phi^1\big(\pi^\infty,(\sigma^*)^\infty\big) \text{ for all } \pi \in \Pi.$$

From the proof of Theorem 5.20 it follows that S_{x^*} is the set of recurrent states in the Markov chain induced by the stationary policy $(\pi^*)^\infty$. Recall that $\pi_{ia}^* := \frac{x_i^*(a)}{\sum_a x_i^*(a)},\ a \in A(i),\ i \in S_{x^*}$, implying $\pi_{ia}^* \cdot x_i^* = x_i^*(a)$ for all $(i,a) \in S \times A$, where $x_i^* := \sum_a x_i^*(a),\ i \in S$. From (5) it follows that $\sum_a r_i^2(a,b) \cdot \pi_{ia}^* \cdot x_i^* - \sum_a r_i^2(a,\sigma^*) \cdot \pi_{ia}^* \cdot x_i^* \leq 0$ for all $(i,b) \in S \times B$. Hence, $r_i^2(\pi,\sigma) \cdot x_i^* \leq r_i^2(\pi^*,\sigma) \cdot x_i^*$ for all $i \in S$ and $\sigma \in \Sigma$. Since $x_i^* > 0,\ i \in S_{x^*}$, we have $r_i^2(\pi,\sigma) \leq r_i^2(\pi^*,\sigma)$ for all $i \in S_{x^*}$ and $\sigma \in \Sigma$. Since $p_{ij}^* = 0,\ i \in S,\ j \notin S_{x^*}$, we can write for all $i \in S$

$$\begin{array}{rcl} \phi_i^1\big((\pi^*)^\infty,\sigma^\infty\big) & = & \{P^*(\pi^*)r^2(\pi^*,\sigma)\}_i \ = \ \sum_{j \in S_{x^*}} p_{ij}^* r_j^2(\pi^*,\sigma) \\ & \leq & \sum_{j \in S_{x^*}} p_{ij}^* r_j^2(\pi^*,\sigma^*) = \phi_i^1\big((\pi^*)^\infty,(\sigma^*)^\infty\big). \end{array}$$

We have shown that for all $\pi \in \Pi$ and all $\sigma \in \Sigma$, $\phi^1\big((\pi^*)^\infty, (\sigma^*)^\infty\big) \geq \phi^1\big(\pi^\infty, (\sigma^*)^\infty\big)$ and $\phi^2\big((\pi^*)^\infty, (\sigma^*)^\infty\big) \geq \phi^2\big(\pi^\infty, (\sigma^*)^\infty\big)$, i.e. $\big((\pi^*)^\infty, (\sigma^*)^\infty\big)$ is an equilibrium point of Γ. Therefore, by Lemma 10.37, $\big((\pi^*)^\infty, (\sigma^*)^\infty\big)$ is an equilibrium point of $\overline{\Gamma}$. □

10.6 Bibliographic notes

The name *stochastic game* stems from the seminal paper by Shapley ([267]). Some authors use the name *Markov game*, which expresses the relation with Markov decision processes. For books and surveys on stochastic games we refer to [213], [315], [33], [231] and [99]. The book of Von Neumann and Morgenstern ([313]) generally is seen as the starting point of game theory. A standard book, including much material on matrix games, is Owen ([209]). The relation between bimatrix games and quadratic programming is due to Mangasarian and Stone ([191]).

The fixed point result for discounted games, i.e. the value vector v^α is the unique solution of $x = Tx$ is due to Shapley ([267]). The subsection on discounted games with perfect information and Algorithm 10.1 was based on Raghavan and Syed ([232]). The matarial on Blackwell optimality with Algorithm 10.2 is due to Avrachenkov, Cottatellucci and Maggi ([8]). The mathematical programming formulations of section 10.2.2 were presented by Rothblum ([245]), and Hordijk and Kallenberg ([127]). The method of value iteration (Algorithm 10.3) for discounted games is also due to Shapley ([267]).

The iterative algorithms 10.4, 10.5 and 10.6 were proposed by Hoffman and Karp ([119]), Pollatschek and Avi-Itzhak ([218]), and Van der Wal ([294]), respectively. Example 10.3 that shows that Algorithm 10.5 does not converge in general is also due to Van der Wal ([294]). For a survey on (modified) value iteration methods we refer to Van der Wal and Wessels ([299]).

The notion that the ordered field property holds for discounted stochastic games in which one player controls the transitions, which yields a finite algorithm for such games, is due to Parthasarathy and Raghavan ([212]), a paper that also contains Example 10.4; see also Hordijk and Kallenberg ([127]). The switching-controller stochastic game first was studied by Filar ([89]). Algorithm 10.8 is due to Vrieze ([315]); see also Vrieze, Tijs, Raghavan and Filar ([316]). The *SER-SIT* game was introduced by Sobel ([277]) and later studied by Parthasarathy, Tijs and Vrieze ([214]). Raghavan, Tijs and Vrieze ([233]) have solved the *ARAT* stochastic game.

The material in the section on Markov games with the total reward criterion, including unconstrained and constrained single-controller games, is due to Hordijk and Kallenberg ([127]).

The average reward stochastic games were introduced in 1957 by Gillette ([103]), who studied two special classes: games with perfect information and irreducible games. Gillette's proof that the above classes possess stationary optimal policies were later completed by Liggett and Lippman ([179]). Gillette's paper contains also the example of the Big Match, showing that undiscounted games were inherently more complex than discounted games. The complete analysis of the Big Match was made by Blackwell and Ferguson ([31]).

For the results that an undiscounted stochastic game possesses optimal stationary policies if and only if a global minimum with objective value zero can be found to an appropriate nonlinear program we refer to Filar and Schultz ([96]) and to Filar, Schultz, Thuijsman and Vrieze ([94]). The proof that in a game with perfect information both players possess optimal deterministic policies is due to Federgruen ([80]). The value iteration method, described in Algorithm 10.15, can be found in the paper by Hoffman and Karp ([119]). Van der Wal ([295]) developed a value iteration algorithm for the unichain case.

Stern, in his PhD thesis ([282]) proved that in the undiscounted single-controller stochastic game both players possess optimal stationary policies. Hordijk and Kallenberg ([128]) and independently Vrieze ([314]) discovered the linear programming solution this class of games. In Hordijk and Kallenberg's paper [128] also the solution of Makov games with additional constraints can be found. Filar ([91]), Filar and Raghavan ([95]) and Bayal-Gürsoy ([14]) made also contributions to the undiscounted single-controller stochastic game.

The existence of optimal stationary policies for the switching-controller undiscounted stochastic game is due to Filar ([89]). Vrieze, Raghavan, Tijs and Filar ([316]) have developed Algorithm 10.20. The solution of the undiscounted *SER-SIT* game was presented in Parthasarathy, Tijs and Vrieze ([214]). It is not known whether stochastic games with additive transitions have stationary optimal policies. When also the rewards are additive (*ARAT* games), then Raghavan, Tijs and Vrieze ([233]) have shown that the undiscounted *ARAT* stochastic game possesses the ordered field property and that both players have deterministic and stationary optimal policies.

Sobel([276]) was the first who studied two-person, general-sum, stochastic games. He established the existence of stationary equilibria in the discounted case. Parthasarathy and Raghavan ([212]) have shown that two-person, general-sum, single-controller stochastic games possess, both for discounted and average rewards, stationary equilibria points and the ordered field property. The algorithmic approach with quadratic programs for the computation of stationary equilibria as described in Theorem 10.53 and Theorem 10.56 is due to Filar ([92]). In [90] Filar has shown that the set of stationary equilibrium points is the union of a finite number of sets such that every element of these sets can be constructed from a finite number of extreme equilibrium strategies for player 1 and from a finite number of pseudo-extreme equilibrium strategies for player 2. These extreme and pseudo-extreme equilibrium strategies can themselves constructed by finite (but inefficient) algorithms. This result holds for two-person, general-sum, single-controller stochastic games both with the discounted as the average reward criterion.

Filar and Vrieze ([98]) have considered stochastic games in which the players aggregate their sequences of expected rewards according to weighted criteria. These are either a convex combination of two discounted objectives or one discounted and one limiting average reward objective. In both cases they have established the existence of the value vector of these games. For the convex combination of two discounted objectives they have shown that both players possess optimal Markov policies and ε-optimal policies that are ultimately stationary. For the discounted/average reward objective no optimal or ε-optimal Markov policies needs to exist, but both players have ε-optimal policies that are ultimately ε-optimal in the average reward game.

10.7 Exercises

Exercise 10.1

Consider the following discounted stochastic game:

$S = \{1, 2\};\ A(1) = \{1, 2, 3\},\ A(2) = \{1, 2\};\ B(1) = \{1, 2\},\ B(2) = \{1, 2, 3\};\ \alpha = \frac{1}{2}.$

$r_1(1, 1) = 1;\ r_1(1, 2) = 2;\ r_1(2, 1) = 5;\ r_1(2, 2) = 0;\ r_1(3, 1) = 0;\ r_1(3, 2) = 4;$

$r_2(1, 1) = 0;\ r_2(1, 2) = 3;\ r_2(1, 3) = 6;\ r_2(2, 1) = 6;\ r_2(2, 2) = 2;\ r_2(2, 3) = 0.$

$p_{11}(1, 1) = 1,\ p_{12}(1, 1) = 0;\ p_{11}(1, 2) = 0,\ p_{12}(1, 2) = 1;\ p_{11}(2, 1) = 1,\ p_{12}(2, 1) = 0;$

$p_{11}(2, 2) = 0,\ p_{12}(2, 2) = 1;\ p_{11}(3, 1) = 1,\ p_{12}(3, 1) = 0;\ p_{11}(3, 2) = 0,\ p_{12}(3, 2) = 1;$

$p_{21}(1, 1) = 1,\ p_{22}(1, 1) = 0;\ p_{21}(1, 2) = 0,\ p_{22}(1, 2) = 1;\ p_{21}(1, 3) = 1,\ p_{22}(1, 3) = 0;$

$p_{21}(2, 1) = 1,\ p_{22}(2, 1) = 0;\ p_{21}(2, 2) = 0,\ p_{22}(2, 2) = 1;\ p_{21}(2, 3) = 1,\ p_{22}(2, 3) = 0.$

Apply Algorithm 10.1 to compute x^2, starting with $x^0 = (0, 0)$.

Exercise 10.2

Execute one iteration of Algorithm 10.4 on the model of Exercise 10.1.
Start with $\rho^*_{11} = \rho^*_{12} = \frac{1}{2};\ \rho^*_{21} = \rho^*_{22} = \rho^*_{23} = \frac{1}{3}$.

Exercise 10.3

Execute one iteration of Algorithm 10.5 on the model of Exercise 10.1. Start with $x = (0, 0)$.

Exercise 10.4

Execute one iteration of Algorithm 10.4 on the model of Exercise 10.1. Start with $x = (6, 6)$ and take $k = 2$.

Exercise 10.5

Consider the single-controller stochastic game in which player 2 controls the transitions.

a. Formulate the dual pair of linear programs for this stochastic game analogous to the programs (10.31) and (10.32).

b. Give the analogon of Theorem 10.15.

Exercise 10.6

The stochastic game of Exercise 10.1 is a single-controller stochastic game in which player 2 controls the transitions. Determine the value vector and optimal policies for the two players by linear programming as indicated in Exercise 10.5.

Exercise 10.7

Apply Algorithm 10.8 to the following switching control stochastic game.

$S = \{1,2\};\ S_1 = \{1\},\ S_2 = \{2\};\ A(1) = B(1) = A(2) = B(2) = \{1,2\};\ \alpha = \frac{1}{2}$.

$r_1(1,1) = 4,\ r_1(1,2) = 0,\ r_1(2,1) = 0,\ r_1(2,2) = 6;$

$r_2(1,1) = 3,\ r_2(1,2) = 5,\ r_2(2,1) = 6,\ r_2(2,2) = 4.$

$p_{11}(1) = 1,\ p_{12}(1) = 0;\ p_{11}(2) = 0,\ p_{12}(2) = 1;\ p_{21}(1) = 1,\ p_{22}(1) = 0;\ p_{21}(2) = 0,\ p_{22}(2) = 1.$

Start with $\rho_{21}^0 = 1,\ \rho_{22}^0 = 0$.

Exercise 10.8

Apply Algorithm 10.9 to the following $SER - SIT$ stochastic game.

$S = \{1,2\};\ A(1) = B(1) = A(2) = B(2) = \{1,2\};\ \alpha = \frac{1}{2}$.

$s_1 = 0,\ s_2 = 1;\ t(1,1) = 0,\ t(1,2) = 2,\ t(2,1) = 1,\ t(2,2) = 3.$

$p_1(1,1) = \frac{1}{2},\ p_2(1,1) = \frac{1}{2};\ p_1(1,2) = 1,\ p_2(1,2) = 0;$

$p_1(2,1) = 0,\ p_2(2,1) = 1;\ p_1(2,2) = \frac{1}{2},\ p_2(2,2) = \frac{1}{2}.$

Exercise 10.9

Consider the following model which has the SER property but not the SIT property.

$S = \{1,2,3\};\ A(i) = B(i) = \{1,2\}$ for $i = 1,2,3$. $s_1 = 1,\ s_2 = 1,\ s_3 = 2;\ t(a,b) = 0$ for all (a,b).

$p_{11}(1,1) = 1,\ p_{12}(1,1) = 0,\ p_{13}(1,1) = 0;\ p_{11}(1,2) = 0,\ p_{12}(1,2) = 1,\ p_{13}(1,2) = 0;$

$p_{11}(2,1) = 0,\ p_{12}(2,1) = 1,\ p_{13}(2,1) = 0;\ p_{11}(2,2) = 1,\ p_{12}(2,2) = 0,\ p_{13}(2,2) = 0;$

$p_{21}(1,1) = 0,\ p_{22}(1,1) = 1,\ p_{23}(1,1) = 0;\ p_{21}(1,2) = 0,\ p_{22}(1,2) = 0,\ p_{23}(1,2) = 1;$

$p_{21}(2,1) = 1,\ p_{22}(2,1) = 0,\ p_{23}(2,1) = 0;\ p_{21}(2,2) = 0,\ p_{22}(2,2) = 1,\ p_{23}(2,2) = 0;$

$p_{31}(1,1) = 0,\ p_{32}(1,1) = 1,\ p_{33}(1,1) = 0;\ p_{31}(1,2) = 0,\ p_{32}(1,2) = 1,\ p_{33}(1,2) = 0;$

$p_{31}(2,1) = 0,\ p_{32}(2,1) = 1,\ p_{33}(2,1) = 0;\ p_{31}(2,2) = 0,\ p_{32}(2,2) = 1,\ p_{33}(2,2) = 0.$

a. Show that $v_1^\alpha = 1 + \frac{1}{2}\alpha(v_1^\alpha + v_2^\alpha)$ and $v_3^\alpha = 2 + \alpha v_2^\alpha$.

b. Show that $v_2^\alpha = \frac{(6+\alpha)-\sqrt{(\alpha^2-20\alpha+36)}}{8(1-\alpha)}$.

c. Show that this game does not possess the ordered field property.

Exercise 10.10

Show that, without using Theorem 10.30, the Big Match does not satisfy both (10.61) and (10.62).

Exercise 10.11

Execute Algorithm 10.16 to compute the value vector and optimal stationary policies for both players for the following undiscounted stochastic game:

$S = \{1,2,3\};\ A(1) = B(1) = \{1,2\};\ A(2) = \{1\},\ B(2) = \{1,2\};\ A(3) = B(3) = \{1\}.$

$p_{11}(1) = 1,\ p_{12}(1) = 0,\ p_{13}(1) = 0;\ p_{11}(2) = 0,\ p_{12}(2) = \frac{1}{2},\ p_{13}(2) = \frac{1}{2};$

$p_{21}(1) = 0,\ p_{22}(1) = 1,\ p_{23}(1) = 0;\ p_{31}(1) = 0,\ p_{32}(2) = 0,\ p_{33}(1) = 1.$

$r_1(1,1) = 1,\ r_1(1,2) = 0,\ r_1(2,1) = 0,\ r_1(2,2) = 1;\ r_2(1,1) = 4,\ r_2(1,2) = 2,\ r_3(1,1) = -1.$

Bibliography

[1] Aho, A.J., J.E. Hopcroft and J.D. Ullman: *The design and analysis of computer algorithms*, Addison-Wesley, Massachusetts, 1974.

[2] Altman, E.: *Constrained Markov decision processes*, Chapman & Hall/CRC, 1999.

[3] Altman, E., A. Hordijk and L.C.M. Kallenberg: *On the value function in constrained control of Markov chains*, Mathematical Methods of Operations Research 44 (1996) 389–399.

[4] Altman, E. and A. Shwartz: *Sensitivity of constrained Markov decision processes*, Annals of Operations Research 33 (1991) 1–22.

[5] Altman, E. and F. Spieksma: *The linear program approach in multi-chain Markov decision processes revisited*, ZOR - Mathematical Methods of Operations Research 42 (1995) 169–188.

[6] Applegate, S.D., W. Cook and M. Mevenkamp, *SQopt reference manual*, 2003.

[7] Avrachenkov, K.E. and E. Altman: *Sensitive discount optimality via nested linear programs for ergodic Markov decision processes*, IDC'99 Proceedings (1999) 53–58.

[8] Avrachenkov, K., L. Cottatellucci and L. Maggi: *Algorithms for uniform optimal strategies in two-player zero-sum stochastic games with perfect information.* OPerations Research Letters 40 (2012) 50–60.

[9] Baras, J.S., D.J. Ma and A.M. Makowsky: *K competing queues with linear costs and geometric service requirements: the μc-rule is always optimal*, Systems Control Letters 6 (1985) 173–180.

[10] Bartmann, D.: *A method of bisection for discounted Markov decision problems*, Zeitschrift für Operations Research 23 (1979) 275–287.

[11] Bather, J.: *Optimal decision procedures for finite Markov chains. Part I: Examples*, Advances in Applied Probability 5 (1973) 328–339.

[12] Bather, J.: *Optimal decision procedures for finite Markov chains. Part II: Communicating systems*, Advances in Applied Probability 5 (1973) 521–540.

[13] Bauer, H.: *Probability theory and elements of measure theory*, Second English Edition, Academic Press, London, 1981.

[14] Bayal-Gürsoy, M.: *Two-person zero-sum stochastic games*, annals of Operations Research 28 (1991) 135–152.

[15] Bayal-Gürsoy, M. and K.W. Ross: *Variability-sensitive Markov decision processes*, Mathematics of Operations Research 17 (1992) 558–571.

[16] Beckmann, M.: *An inventory model for arbitrary interval and quantity distributions of demands*, Management Science 8 (1961) 35–57.

[17] Bellman, R.: *Dynamic programming*, Princeton University Press, Princeton, 1957.

[18] Bellman, R., I. Glicksberg and O. Gross: *On the optimal inventory equation*, Management Science 2 (1955) 83–104.

[19] Bello, D. and G. Riano: *Linear programming solvers for Markov decision processes*, in: M. DeVore (ed.), *Proceedings of the 2006 IEEE System and Information Engineering Design Symposium* (2006) 93-98.

[20] Ben-Israel, A. and S.D. Flåm: *A bisection/successive approximation method for computing Gittins indices*, Zeitschrift für Operations Research 34 (1990) 411–422.

[21] Bertsekas, D.P. and S.E. Shreve: *Stochastic optimal control: the discrete time case*, Academic Press, New York, 1978.

[22] Bertsekas, D.P.: *Dynamic programming: deterministic and stochastic models*, Prentice-Hall, 1987.

[23] Bertsimas, D. and J. Nino-Mora: *Conservation laws, extended polymatroids and multi-armed bandit problems: a unified approach to indexable systems*, Mathematics of Operations Research 21 (1996) 257–306.

[24] Beutler, F.J. and K.W. Ross: *Optimal policies for controlled Markov chains with a constraint*, Journal of Mathematical Analysis and Applications 112 (1985) 236–252.

[25] Bewley, T. and E. Kohlberg: *The asymptotic theory of stochastic games*, Mathematics of Operations Research 1 (1976) 197–208.

[26] Bewley, T. and E. Kohlberg: *The asymptotic solution of a recursive equation arising in stochastic games*, Mathematics of Operations Research 1 (1976) 321–336.

[27] Bewley, T. and E. Kohlberg: *On stochastic games with stationary optimal solutions*, Mathematics of Operations Research 3 (1978) 104–127.

[28] Bierth, K.-J.: *An expected average reward criterion*, Stochastic Processes and Applications 26 (1987) 133–140.

[29] Blackwell, D.: *Discrete dynamic programming*, Annals of Mathematical Statistics 33 (1962) 719–726.

[30] Blackwell, D.: *Positive dynamic programming*, Proceedings Fifth Berkeley Symposium Mathematical Statistics and Probability, Volume 1 (1967) 415–418.

[31] Blackwell, D. and T. Ferguson: *The Big Match*, Annals of Mathematical Statistics 39 (1968) 159–163.

[32] Breiman, L.: *Stopping-rule problems*, in: E.F. Beckenbach (ed.), *Applied Combinatorial Mathematics*, Wiley, New York, 1964, 284–319.

[33] Breton, M., J.A. Filar, A. Haurie and T.A. Shultz: *On the computation of equilibria in discounted stochastic games*, in: T.Basar (ed.), *Dynamic games and applications in economics*, Lecture Notes in Economics and Mathematical Systems no. 265 (1985), Springer-Verlag.

[34] Brown, B.W.: *On the iterative method of dynamic programming on a finite space discrete Markov process*, Annals of Mathematical Statistics 36 (1965) 1279–1285.

[35] Bruno, J., P. Downey and G. Frederickson: *Sequencing tasks with exponential service times to minimize the expected flowtime or makespan* Journal of the ACM 28 (1981) 100–113.

[36] Buyukkoc, C., P. Varaiya and J. Walrand: *The μc-rule revisited*, Advances in Applied Probability 17 (1985) 237–238.

[37] Cesaro, E.: *Sur la multiplication des séries*, Bulletin des Sciences Mathématiques 14 (1890) 114–120.

[38] Chen, Y.-R. and M.N. Katehakis: *Linear programming for finite state bandit problems*, Mathematics of Operations Research 11 (1986) 180–183.

[39] Cheng, M.C.: *New criteria for the simplex method*, Mathematical Programming 19 (1980) 230–236.

[40] Chung, K.L.: *Markov chains with stationary transition probabilities*, Springer, 1960.

[41] Chung, K.-J.: *A note on maximal mean/standard deviation ratio in an undiscounted MDP*, OR Letters 8 (1989) 201–204.

[42] Chung, K.-J.: *Remarks on maximal mean/standard deviation ratio in undiscounted MDPs*, Optimization 26 (1992) 385–392.

[43] Chung, K.-J.: *Mean-variance tradeoffs in an undiscounted MDP: the unichain case*, Operations Research 42 (1994) 184–188.

[44] Cohen, E. and N. Megiddo: *Improved algorithms for linear inequalities with two variables per inequality*, SIAM Journal of Computing 23 (1994) 131–1347.

[45] Collins, E.J.: *Finite-horizon variance penalized Markov decision processes*, OR Spektrum 19 (1997) 35–39.

[46] Cook, S.A.: *The complexity of theorem proving procedures*, Proceedings of the 3rd ACM Symposium on the theory of computing. ACM (1971) 151-158.

[47] Cox, D.R. and W.L. Smith: *Queues*, Methuen, London, 1961.

[48] Dantzig, G.B.: *Linear programming and extensions*, Princeton University Presss, 1963.

[49] Dash Optimization, *Xpress-MP essentials*, second edition, Dash Optimization Inc., 2002.

[50] De Cani, J.S.: *A dynamic programming algorithm for embedded Markov chains when the planning horizon is at infinity*, Management Science 10 (1964) 716–733.

[51] De Ghellinck, G.T.: *Les problèmes de décisions séquentielles*, Cahiers du Centre de Recherche Opérationelle 2 (1960) 161–179.

[52] De Ghellinck, G.T. and G.D. Eppen: *Linear programming solutions for separable Markovian decision problems*, Management Science 13 (1967) 371–394.

[53] Dekker, R.: *Denumerable Markov decision chains: Optimal policies for small interest rate*, Ph.D. Dissertation, Leiden University, 1985.

[54] Dekker, R. and A. Hordijk: *Average, sensitive and Blackwell optimality in denumerable Markov decision chains with unbounded rewards*, Mathematics of Operations Research 13 (1988) 395–421.

[55] Dembo, R.S. and M. Haviv: *Truncated policy iteration methods*, OR Letters 3 (1984) 243–246.

[56] Denardo, E.V.: *Contraction mappings in the theory underlying dynamic programming*, SIAM Review 9 (1967) 165–177.

[57] Denardo, E.V.: *Separable Markov decision problem*, Management Science 14 (1968) 451–462.

[58] Denardo, E.V.: *On linear programming in a Markov decision problem*, Management Science 16 (1970) 281–288.

[59] Denardo, E.V.: *Computing a bias-optimal policy in a discrete-time Markov decision problem*, Operations Research 18 (1970) 279–289.

[60] Denardo, E.V.: *Markov renewal programs with small interest rates*, Annals of Mathematical Statistics 42 (1971) 279–289.

[61] Denardo, E.V.: *A Markov decision problem*, in: T.C.Hu and S.M.Robinson (eds.) *Mathematical Programming*, Academic Press (1973) 33–68.

[62] Denardo, E.V.: *Stopping and regeneration*, Draft of Chapter 7, problem 4 (1975).

[63] Denardo, E.V.: *Dynamic programming: Models and Applications*, Prentice-Hall, 1982.

[64] Denardo, E.V. and B.L. Fox: *Multichain Markov renewal programs*, SIAM Journal on Applied Mathematics 16 (1968) 468–487.

[65] Denardo, E.V. and B.L. Miller: *An optimality condition for discrete dynamic programming with no discounting*, Annals of Mathematical Statistics 39 (1968) 1220–1227.

[66] Denardo, E.V. and U.G. Rothum: *Overtaking optimality for Markov decision chains*, Mathematics of Operations Research 4 (1979) 144–152.

[67] D'Epenoux, F.: *Sur un problème de production et de stockage dans l'aléatoire*, Revue Française de Recherche Opérationelle 14 (1960) 3–16 .

[68] Derman, C.: *On optimal replacement rules when changes of state are Markovian*, in: R.Bellman (ed.) *Mathematical Optimization Techniques*, University of California Press, Berkeley (1963) 201–210.

[69] Derman, C.: *Finite state Markovian decision processes*, Academic Press, New York, 1970.

[70] Derman, C. and M. Klein: *Some remarks on finite horizon Markovian decision models*, Operations Research 13 (1965) 272–278.

[71] Derman, C. and R. Strauch: *A note on memoryless rules for controlling sequential control problems*, Annals of Mathematical Statistics 37 (1966) 276–278.

[72] Derman, C., G.J. Lieberman and S.M. Ross: *A sequential stochastic assignment model*, Management Science 18 (1972) 349–355.

[73] Derman, C. and A.F. Veinott Jr.: *Constrained Markov decision chains*, Management Science 19 (1972) 389–390.

[74] Doob, J.L.: Stochastic processes, Wiley, 1953.

[75] Dubins, L.E. and L.J. Savage: *How to gamble if you must: inequalities for stochastic processes*, McGraw-Hill, New York, 1965.

[76] Durinovic, S., H.M. Lee, M.N. Katehakis and J.A. Filar: *Multiobjective Markov decision processes with average reward criterion* Large Scale Systems 10 (1986) 215–226.

[77] Eaves, B.C. and A.F. Veinott, Jr.: *Maximum-stopping-value policies in finite Markov population decision chains*, Report, Stanford University, 2007.

[78] Ephremides, A., P. Varaiya and J. Walrand: *A simple dynamic routing problem*, IEEE Transactions on Automatic Control AC-25 (1980) 690–693.

[79] Faiz, A. and J. Falk: *Jointly constrained biconvex programming*, Mathematics of Operations Research 8 (1983) 273–286.

[80] Federgruen, A.: *Markovian control problems: functional equations and algorithms*, Mathematical Centre Tracts no.97, Amsterdam, 1984.

[81] Federgruen, A., P.J. Schweitzer and H.C. Tijms: *Contraction mappings underlying undiscounted Markov decision problems*, Journal of Mathematical Analyss and Applications 65 (1978) 711–730.

[82] Federgruen, A. and D. Spreen: *A new specification of the multichain policy iteration algorithm in undiscounted Markov renewal programs*, Management Science 26 (1980) 1211–1217.

[83] Feinberg, E.A.: *Constrained semi-Markov decision processes with average rewards*, Mathematical Methods op Operations Research 39 (1994) 257–288.

[84] Feinberg, E.A. and A. Shwartz: *Markov decision problems with weighted discounted criteria*, Mathematics of Operations Research 19 (1994) 152–168.

[85] Feinberg, E.A. and A. Shwartz: *Constrained Markov decision models with weighted discounted rewards*, Mathematics of Operations Research 20 (1995) 302–320.

[86] Feinberg, E.A. and A. Shwartz: *Constrained dynamic programming with two discount factors: applications and an algorithm*, IEEE Transactions on Automatic Control 44 (1999) 628–631.

[87] Feinberg, E.A. and F. Yang: *On Polynomial Classification Problems for Markov Decision Processes* Proceedings of the 2008 NSF Engineering Research and Innovation Conference, Knoxville, TN.

[88] Feller, W.: *An introduction to probability theory and its aplications*, Volume I, third edition, Wiley, 1970.

[89] Filar, J.A.: *Ordered field property for stochastic games when the player who controls transitions changes from state to state*, Journal on Optimization Theory and Applications 34 (1981) 503–513.

[90] Filar, J.A.: *On stationary equilibria of a single-controller stochastic game*, Mathematical Programming 30 (1984) 313–325.

[91] Filar, J.A.: *The completely mixed single-controller stochastic game*, Proceedings of the American Mathematical Society 95 (1985) 585–594.

[92] Filar, J.A.: *Quadratic programming and the single-controller stochastic game*, Journal on Mathematical Analysis and Applications 113 (1986) 136–147.

[93] Filar, J.A., L.C.M. Kallenberg and H.M. Lee: *Variance-penalized Markov decision processes*, Mathematics of Operations Research 14 (1989) 147–161.

[94] Filar, J.A., Schultz, F. Thuijsman and O.J. Vrieze: *Nonlinear programming and stationary equilibria in stochastic games*, Mathematical Programming 50 (1991) 227–237.

[95] Filar, J.A. and T.E.S. Raghavan: *A matrix game solution of the single-controller stochastic game*, Mathematics of Operations Research 9 (1984) 356–362.

[96] Filar, J.A. and T. Schultz: *Nonlinear programming and stationary strategies in stochastic games*, Mathematical Programming 35 (1988) 243–247.

[97] Filar, J.A. and T. Schultz: *Communicating MDPs: Equivalence and LP properties*, Operations Research Letters 7 (1988) 303–307.

[98] Filar, J.A. and O.J. Vrieze: *Weighted reward criteria in competitive Markov decision processes*, Methods of Operations Research 36 (1992) 343–358.

[99] Filar, J.A. and O.J. Vrieze: *Competitive Markov decision processes*, Springer-Verlag, 1997.

[100] Gal, S.: *A $\mathcal{O}(N^3)$ algorithm for optimal replacement problems*, SIAM Journal of Control and Optimization 22 (1984) 902–910.

[101] Gallo, G. and A. Alkucu: *Bilinear programming: an exact algorithm*, Mathematical Programming 12 (1977) 173–194.

[102] Garey, M.R. and D.S. Johnson: *Computers and intractability - A guide to the theory of NP-completeness*, Freeman, San Francisco, California, 1979.

[103] Gillette, D: *Stochastic games with zero stop probabilities* in: Dresher, M., A.W. Tucker and P. Wolfe (eds.), *Contributions to the theory of games*, vol. III, Princeton University Press, Annals of Mathematics Studies 39 (1957) 179–187.

[104] Gittins, J.C.: *Bandit processes and dynamic allocation indices*, Journal of the Royal Statistic Society Series B 14 (1979) 148–177.

[105] Gittins, J.C. and D.M. Jones: *A dynamic allocation index for the sequential design of experiments*, in: J. Gani (ed.) *Progress in Statistics* North Holland, Amsterdam (1974) 241–266.

[106] Glazebrook, K.D.: *Scheduling tasks with exponential service times on parallel processors* Journal of Applied Probability 16 (1979) 685–689.

[107] Glazebrook, K.D. and R.W. Owen: *New results for generalized bandit problems* International Journal of System Science22 (1991) 479–494.

[108] Goldman, A.J. and A.W. Tucker: *Theory of linear programming*: in: Linear inequalities and related systems, H.W. Kuhn and A.W. Tucker (eds.), Annals of Mathematical Studies (1956) 53–97.

[109] Grinold, R.C.: *Elimination of suboptimal actions in Markov decision problems*, Operations Research $\underline{21}$ (1973) 848–851.

[110] Hartley, R., A.C. Lavercombe and L.C. Thomas: *Computational comparison of policy iteration algorithms for discounted Markov decision processes*, Computers & Operations Research $\underline{13}$ (1986) 411-420.

[111] Hastings, N.A.J.: *Some notes on dynamic programming and replacement*, Operational Research Quarterly $\underline{19}$ (1968) 453–464.

[112] Hastings, N.A.J.: *Optimization of discounted Markov decision problems*, Operations Research Quarterly $\underline{20}$ (1969) 499–500.

[113] Hastings, N.A.J.: *A test for nonoptimal actions in undiscounted finite Markov decision chains*, Management Science $\underline{23}$ (1976) 87–92.

[114] Haviv, M. and M.L. Puterman: *An improved algorithm for solving communicating average reward Markov decision processes*, Annals of Operations Research $\underline{28}$ (1991) 229–242.

[115] Hertzberg, M. and U. Yechiali: *Criteria for selecting the relaxation factor of the value iteration algorithm for undiscounted Markov and semi-Markov decision processes*, Operations Research Letters $\underline{10}$ (1991) 193–202.

[116] Hertzberg, M. and U. Yechiali: *Accelerated procedures of the value iteration algorithm for discounted Markov decision processes, based on a one-step look-ahead analysis*, Operations Research $\underline{42}$ (1994) 940–946.

[117] Heyman, D.P. and M.J. Sobel: *Stochastic models in Operations Research, Volume II: Stochastic optimization*, MacGraw-Hill, 1984.

[118] Hochbaum, D. and J. Naor: *Simple and fast algorithms for linear and integer programs with two variables per inequality*, SIAM Journal of Computing $\underline{23}$ (1994) 1179–1192.

[119] Hoffman, A.J. and R.M. Karp: *On non-terminating stochastic games*, Management Science $\underline{12}$ (1966) 359–370.

[120] Hordijk, A.: *A sufficient condition for the existence of an optimal policy with respect to the average cost criterion in Markovian decision processes*, Transactions of the Sixth Conference on Information Theory, Statistical Decision Functions, Random Processes (1971) 263–274.

[121] Hordijk, A.: *Dynamic programming and Markov potential theory*, Mathematical Centre, Amsterdam, 1974.

[122] Hordijk, A.: *Convergent dynamic programming*, Report BW 47/75, Mathematical Centre, Amsterdam, 1975.

[123] Hordijk, A.: *Stochastic dynaming programming*, Course notes, University of Leiden (in Dutch), 1976.

[124] Hordijk, A.: *From linear to dynamic programming via shortest paths*, Mathematical Centre Tract no. 100, Amsterdam, 1978.

[125] Hordijk, A., R. Dekker and L.C.M. Kallenberg: *Sensitivity analysis in discounted Markov decision problems*, OR Spektrum 7 (1985) 143–151.

[126] Hordijk, A. and L.C.M. Kallenberg: *Linear programming and Markov decision chains*, Management Science 25 (1979) 352–362.

[127] Hordijk, A. and L.C.M.Kallenberg: *Linear programming and Markov games I*, in: O. Moeschlin and D. Pallaschke (eds.), *Game theory and mathematical economics*, North Holland (1981) 291–305.

[128] Hordijk, A. and L.C.M.Kallenberg: *Linear programming and Markov games II*, in: O. Moeschlin and D. Pallaschke (eds.), *Game theory and mathematical economics*, North Holland (1981) 307–320.

[129] Hordijk, A. and L.C.M. Kallenberg: *Transient policies in discrete dynamic programming: linear programming including suboptimality and additional constraints*, Mathematical Programming 30 (1984) 46–70.

[130] Hordijk, A. and L.C.M. Kallenberg: *Constrained undiscounted stochastic dynamic programming*, Mathematics of Operations Research 9 (1984) 276–289.

[131] Hordijk, A. and G.M. Koole: *On the optimality of LEFT and μc rules for parallel processors and dependent arrival processes*, Advances in Applied Probability 25 (1993) 979–996.

[132] Hordijk, A. and H.C. Tijms: *Colloquium Markov programming*, Mathematical Centre Report BC 1/70, Mathematical Centre, Amsterdam (in Dutch).

[133] Hordijk, A. and N.M. van Dijk: *Time-discretization for controlled Markov processes. I. General approximation results*, Kybernetika 32 (1996) 1–16.

[134] Howard, R.A.: *Dynamic programming and Markov processes*, MIT Press, Cambridge, 1960.

[135] Howard, R.A.: *Semi-Markovian decision processes*, Proceedings International Statistical Institute, Ottawa, Canada, 1963.

[136] Hu, G. and C. Wu: *Relative value iteration algorithm based on contraction span seminorm*, OR Transactions 3 (1999) 1–9.

[137] Huang, Y and L.C.M. Kallenberg: *On finding optimal policies for Markov decision chains: A unifying framework for mean-variance tradeoffs*, Mathematics of Operations Research 19 (1994) 434–448.

[138] Iglehart, D.: *Optimality of (s, S)-policies in the infinite horizon dynamic inventory problem*, Management Science 9 (1963) 259–267.

[139] Iglehart, D.: *Dynamic programming and stationary analysis of inventory problems*, Chapter 1 in: H. Scarf, D. Gilford and M. Shelly (eds.), *Multistage inventory models and techniques*, Stanford University Press, Stanford, 1963.

[140] Iserman, M.: *Proper efficiency and the linear vector maximization problem*, Operations Research 22 (1974) 189–191.

[141] Jeroslow, R.G.: *An algorithm for discrete dynamic programming with interest rates near zero*, Management Science Research Report no. 300, Carnegie-Mellon University, Pittsburg, 1972.

[142] Jewell, W.S.: *Markov renewal programming. I: Formulation, finite return models*, Operations Research 11 (1963) 938–948.

[143] Jewell, W.S.: *Markov renewal programming. II: Infinite return models, example*, Operations Research 11 (1963) 949–971.

[144] Johnson, S.M.: *Optimal two- and three-stages production schedules with setup times included*, Naval Research Logistics Quarterly 1 (1954) 61–68.

[145] Kakutani, S: *A generalization of Brouwer's fixed point theorem*, Duke Mathematical Journal 8 (1941) 457–459.

[146] Kallenberg, L.C.M.: *Finite horizon dynamic programming and linear programming*, Methods of Operations Research 43 (1981) 105–112.

[147] Kallenberg, L.C.M.: *Unconstrained and constrained dynamic programming over a finite horizon*, Report, University of Leiden, 1981.

[148] Kallenberg, L.C.M.: *Linear programming and finite Markovian control problems*, Mathematical Centre Tract no.148, Amsterdam, 1983.

[149] Kallenberg, L.C.M.: *A note on M.N.Katehakis and Y.-R.Chen's computation of the Gittins index*, Mathematics of Operations Research 11 (1986) 184–186.

[150] Kallenberg, L.C.M.: *Separable Markov decision problems*, OR Spektrum 14 (1992) 43–52.

[151] Kallenberg, L.C.M.: *Survey of linear programming for standard and nonstandard Markovian control problem. Part I: Theory*, ZOR - Mathematical Methods of Operations Research 40 (1994) 1–42.

[152] Kallenberg, L.C.M.: *Survey of linear programming for standard and nonstandard Markovian control problem. Part II: Applications*, ZOR - Mathematical Methods of Operations Research 40 (1994) 127–143.

[153] Kallenberg, L.C.M.: *Classification problems in MDPs*, in: Z. How, J.A. Filar and A. Chen (ed.) *Markov processes and controlled Markov chains*, Kluwer Boston (2002) 151–165.

[154] Kao, E.P.C.: *Optimal replacement rules when changes of state are semi-Markovian*, Operations Research 21 (1973) 1231–1249.

[155] Karlin, S.: *Mathematical methods and theory in games, programming and economics*, Volume I, Addison-Wesley, 1959.

[156] Karlin, S.: *Dynamic inventory policy with varying stochastic demands*, Management Science 6 (1960) 231–258.

[157] Karmarkar, L.G.: *A new polynomial-time algorithm for linear programming*, Combinatorica 4 (1984) 373-395.

[158] Karp, R.: *Reducibility among combinatorial problems*, in: R.E. Miller and J.W. Thatcher (eds.): *Complexity of computer computations*, Plenum Press (1972) 85103.

[159] Katehakis, M.N. and C. Derman: *Optimal repair allocation in a series system*, Mathematics of Operations Research 9 (1984) 615–623.

[160] Katehakis, M.N. and C. Derman: *On the maintenance of systems composed of highly reliable components*, Management Science 35 (1989) 551–560.

[161] Katehakis M. N. and U. Rothblum: *Finite state multi-armed bandit sensitive-discount, average-reward and average-overtaking optimality*, Annals of Applied Probability 6 (1996) 1024–1034.

[162] Katehakis, M.N. and A.F. Veinott Jr.: *The multi-armed bandit problem: decomposition and computation*, Mathematics of Operations Research 12 (1987) 262–268.

[163] Kato, T.: *Perturbation theory for linear operators*, Springer, 1966.

[164] Kayne, R. and R. Wilson: *Linear algebra* Oxford University Press, 1998.

[165] Kawai, H.: *A variance minimization problem for a Markov decision process*, European Journal of Operations Research 31 (1987) 140–145.

[166] Kawai, H. and N. Katoh: *Variance constrained Markov decision process*, Journal of the Operations Research Society of Japan 30 (1987) 88–100.

[167] Kemeny, J. and L. Snell: *Finite Markov chains*, Van Nostrand, 1960.

[168] Khachiyan, L.G.: *A polynomial algorithm in linear programming* Soviet Mathematics Doklady 20 (1979) 191–194.

[169] Klee, V. and G.J. Minty: *How good is the simplex method?* in: Shisha (ed) *Inequalities III*, Proceedings of the Third Symposium on Inequalities, held at the University of California, Los Angelos, California, Academic Press (1972) 159–175.

[170] Kolesar, P.: *Minimum cost replacement under Markovian deterioration*, Management Science 12 (1966) 694–706.

[171] Koole, G.M.: *Stochastic scheduling and dynamic programming*, CWI Tract 113, CWI, Amsterdam, 1995.

[172] Krass, D., J.A. Filar and S.S. Sinha: *A weighted Markov decision process*, Operations Research 40 (1992) 1180-1187.

[173] Kushner, H.J. and A.J. Kleinman: *Mathematical programming and the control of Markov chains*, IEEE Transactions on Automatic Control AC-13 (1968) 801–820.

[174] Kushner, H.J. and A.J. Kleinman: *Accelerated procedures for the solution of discrete Markov control problems*, IEEE Transactions on Automatic Control AC-16 (1971) 147–152.

[175] Ladner, R.E.: *The circuit value problem is log space complete for $\mathcal{P}$*, ACM SIGACT News 7 (1975) 18-20.

[176] Lasserre, J.B.: *Updating formula for Markov chains and applications*, LAAS Technical Report, 1991.

[177] Lasserre, J.B.: *Detecting optimal and non-optimal actions in average-cost Markov decision processes* Journal of Applied Probability 31 (1994) 979–990.

[178] Lasserre, J.B.: *A new policy iteration scheme for Markov decision processes using Schweitzer's formula*, Journal of Applied Probability 31 (1994) 268–273.

[179] Liggett, T.M. and S.A. Lippman: *Stochastic games with perfect information and time averqage payoff*, SIAM Review 11 (1969) 604–607.

[180] Lin, W. and P.R. Kumar: *Optimal control of a queueing system with two heterogeneous servers*, IEEE Tansactions on Automatic Control AC-29 (1984) 696–705.

[181] Lippman, S.A.: *Criterion equivalence in discrete dynamic programming*, Operations Research 17 (1968) 920–923.

[182] Lippman, S.A.: *Applying a new device in the optimization of exponential queueing systems*, Operations Research 23 (1975) 687–710.

[183] Littman, M.L., T.L. Dean and L.P. Kaelbling: *On the complexity of solving Markov decision problems* Proceedings of the Eleventh Annual Conference on Uncertainty in Artificial Intelligence (1995) 394-402.

[184] Liu, J.Y. and K. Liu: *An algorithm on the Gittins index*, Systems Science and Mathematical Science 7 (1994) 106–114.

[185] Loève, M.: *Probability theory*, Van Nostrand, 1955.

[186] Lozovanu, D. and C. Petric: *Algorithms for finding the minimum cycle mean in the weighted directed graph*, Computer Science Journal of Moldava 6 (1998) 27–34.

[187] Luenberger, D.G.: *Optimization by vector space methods*, Wiley, 1968.

[188] MacQueen, J.: *A modified programming method for Markovian decision problems*, Journal of Mathematical Analysis and Applications 14 (1966) 38–43.

[189] MacQueen, J.: *A test for suboptimal actions in Markov decision problems*, Operations Research 15 (1967) 559–561.

[190] Madani, O., M. Thorup and U. Zwick: *Discounted deterministic Markov decision processes and discounted all-pairs shortest paths*, ACM Transactions on Algorithms 6 (2010) 33: 1–15.

[191] Mangasarian, O.L. and H. Stone: *Two-person nonzero-sum games and quadratic programming*, Journal of Mathematical Analysis and Applications 9 (1964) 348–355.

[192] Manne, A.S.: *Programming of economic lot sizes*, Management Science 4 (1958) 115–135.

[193] Manne, A.S.: *Linear programming and sequential decisions*, Management Science 6 (1960) 259–267.

[194] Manne, A.S. and A.F. Veinott, Jr.: Chapter 11 in A.S. Manne (ed.), *Investments for capacity expansion: size, location and time-phasing*, MIT Press, 1967.

[195] Mansour, Y. and S. Singh: *On the complexity of policy iteration*, Proceedings of the 15th International Conference on Uncertainty in Artificial Intelligence (1999) 401–408.

[196] McCuaig, W.: *Intercyclic digraphs*, in: N. Robertson and P. Seymour (eds.) *Graph structure theory*, Contemporary Mathematics 147, American Mathematical Society (1993) 203–245.

[197] Melekopoglou, M. and A. Condon: *On the complexity of the policy improvement algorithm for Markov decision processes*, INFORMS Journal on Computing 6 (1994) 188–192.

[198] Mertens, J.F. and A. Neyman: *Stochastis games*, International Journal of Game Theory 10 (1981) 53–56.

[199] Miller, B.L. and A.F. Veinott Jr.: *Discrete dynamic programming with a small interest rate*, Annals of Mathematical Statistics 40 (1969) 366–370.

[200] Mine, H. and S. Osaki: *Markovian decision processes*, Elsevier, New York, 1970.

[201] Morton, T.E.: *On the asymptotic convergence rate of cost differences for Markovian decision processes*, Operations Research 19 (1971) 244–248.

[202] Nash, J.F.: *Non-cooperative games*, Annals of Mathematics 54 (1951) 286–295.

[203] Nazareth, J.L. and R.B. Kulkarni: *Linear programming formulations of Markov decision processes* Operations Research Letters 5 (1986) 13–16.

[204] Ng, M.K.: *A note on policy algorithms for discounted Markov decision problems*, Operation Research Letters 25 (1999) 195–197.

[205] Nino-Mora, J.: *A* $(2/3)n^3$ *fast-pivoting algorithm for the Gittins index and optimal stopping of a Markov chain* INFORMS Journal of Computing 19 (2007) 596–606.

[206] Norman, J.M.: *Dynamic programming in tennis - when to use a fast serve*, Journal of the Operational Research Society 36 (1987) 75–77.

[207] Odoni, A.R.: *On finding the maximal gain for Markov decision processes*, Operations Research 17 (1969) 857–860.

[208] O'Sullivan, M.J.: *New methods for dynamic programming over an infinite horizon*, PhD Thesis, Department of Management Science and Engineering, Stanford University, 2003.

[209] Owen, G.: *Game theory*, Academic Press, 1982.

[210] Papadimitriou, C.H.: *Computational complexity*, Addison-Wesley, Reading, Massachusetts, 1994.

[211] Papadimitriou, C.H. and J.N. Tsitsiklis: *The complexity of Markov decision processes*, Mathematics of Operations Research 12 (1987) 441–450.

[212] Parthasarathy, T. and T.E.S. Raghavan: *An orderfield property for stochastic games when one player controls the transitions*, Journal of Optimization Theory and Applications 33 (1981) 375–392.

[213] Parthasarathy, T. and T.E.S. Raghavan: *Some topics in two-person games*, Elsevier, 1971.

[214] Parthasarathy, T., S.H. Tijs and O.J. Vrieze: *Stochastic games with state independent transitions and separable rewards*, in: Hammer, G. and D. Pallschke (eds.): *Selected topics in Operations Research and Mathematical Economics*, Springer (1984) 262–271.

[215] Pinedo, M. and L. Schrage: *Stochastic shop scheduling: a survey*, in: Dempster, M.A.H., J.K. Lenstra and A.H.G. Rinnooy Kan (eds.), *Deterministic and stochastic scheduling*, Reidel, Dordrecht, Holland (1982) 181–196.

[216] Pinedo, M. and G. Weiss: *Scheduling of stochastic tasks on two parallel processors* Naval Research Logistics Quarterly 26 (1979) 527–535.

[217] Platzman, L.K.: *Improved conditions for convergence in undiscounted Markov renewal programming*, Operations Research 25 (1977) 529–533.

[218] Pollatschek, M. and Avi-Itzhak: *Algorithms for stochastic games with geometric interpretation*, Manangement Science 15 (1969) 399–415.

[219] Popyack, J.L., R.L. Brown and C.C. White III: *Discrete versions of an algorithm due to Varaya*, IEEE Transactions on Automatic Control 24 (1979) 503–504.

[220] Porteus, E.L.: *Some bounds for discounted sequential decision processes*, Management Science 18 (1971) 7–11.

[221] Porteus, E.L.: *On the optimality of generalized (s, S) policies*, Management Science 17 (1971) 411–426.

[222] Porteus, E.L.: *Bounds and transformations for discounted finite Markov decision chains*, Operations Research 23 (1975) 761–784.

[223] Porteus, E.L.: *Improved iterative computation of the expected discounted return in Markov and semi-Markov chains*, Zeitschrift für Operations Research 24 (1980) 155-170.

[224] Porteus, E.L. and J.C. Totten: *Accelerated computation of the expected discounted return in a Markov chain*, Operations Research 26 (1978) 350-358.

[225] Powell, R.E. and S.M. Shah: *Summability theory and applications*, Van Nostrand Reinhold, London (1972).

[226] Prussing, J.E.: *How to serve in tennis*, The Mathematical Gazette 61 (1977) 294–296 .

[227] Puterman, M.L.: *Markov decision processes*, Wiley, New York, 1994.

[228] Puterman, M.L. and S.L. Brumelle: *On the convergence of policy iteration in stationary dynamic programming*, Mathematics of Operations Research 4 (1979) 60–69.

[229] Puterman, M.L. and M.C. Shin: *Modified policy iteration algorithms for discounted Markov decision chains*, Management Science 24 (1978) 1127–1137.

[230] Puterman, M.L. and M.C. Shin: *Action elimination procedures for modified policy iteration algorithms*, Operations Research 30 (1982) 301–318.

[231] Raghavan, T.E.S. and J.A. Filar: *Algorithms for stochastic games - a survey*, Zeitschrift für Operations Research 35 (1991) 437–472.

[232] Raghavan, T.E.S. and Z. Syed: *A policy-improvement type algorithm for solving zero-sum two-person stochastic games with perfect information*, Mathematical Programming 95 (2003) 513–532.

[233] Raghavan, T.E.S., S.H. Tijs and O.J. Vrieze: *On stochastic games with additive reward and trasition structure*, Journal of Optimization Theory and Applications 47 (1985) 451–464.

[234] Reetz, D.: *Solution of a Markovian decision problem by successive overrelaxation*, Zeitschrift für Operations Research 17 (1973) 29–32.

[235] Righter, R.: *Scheduling*, in: Shaked, M. and J.G. Shanthikumar (eds.), *Stochastic orders and their applications*, Academic Press, 1994, 381–432.

[236] Ross, S.M.: *Applied probability models with optimization applications*, Holden-Day, San Francisco, 1970.

[237] Ross, S.M.: *Average cost semi-Markov decision processes*, Journal of Applied Probability 7 (1970) 649–656.

[238] Ross, S.M.: *Dynamic programming and gambling models*, Advances in Applied Probability 6 (1974) 593–606.

[239] Ross, S.M.: *Introduction to stochastic dynamic programming*, Academic Press, New York, 1983.

[240] Ross, K.W.: *Randomized and past-dependent policies for Markov decision processes with multiple constraints*, Operations Research 37 (1989) 474–477.

[241] Ross, K.W. and R. Varadarajan: *Markov decision processes with sample path constraints: the communicating case*, Operations Research 37 (1989) 780–790.

[242] Ross, K.W. and R. Varadarajan: *Multichain Markov decision processes with a sample path constraint: a decomposition approach*, Mathematics of Operations Research 16 (1991) 195–207.

[243] Rothblum, U.G.: *Normalized Markov decision chains. I: Sensitive discount optimality*, Operations Research 23 (1975) 785-795.

[244] Rothblum, U.G.: *Normalized Markov decision chains. II: Optimality of nonstationary policies*, SIAM Journal of Control and Optimization 15 (1977) 221232.

[245] Rothblum, U.G.: *Solving stopping stochastic games by maximizing a linear function subject to quadratic constraints*, O. Moeschlin and D. Pallaschke (eds.), *Game theory and mathematical economics*, North Holland (1978) 103–105.

[246] Rothblum, U.G.: *Multiplicative Markov decision chains*, Mathematics of Operations Research 9 (1984) 6–24.

[247] Rothblum, U.G.: *Nonnegative matrices and stochastic matrices*, in L. Hogben, editor, *Handbook of Linear Algebra*, CRC Press, 2006.

[248] Rothblum, U.G. and A.F. Veinott Jr.: *Cumulative average optimality for normalized Markov decision chains*, Working Paper, Dept. of Operations Research, Stanford University, 1975.

[249] Rudin, W.: *Principles of mathematical analysis*, McGraw Hill, New York, 1976.

[250] Savitch, W.J. : Relationships between nondeterministic and deterministic tape complexities, Journal of Computational System Sciences 4 (1970) 177–192.

[251] Scarf, H.: *The optimality of* (s, S)*-policies in the dynamic inventory problem*, Chapter 13 in: K.J. Arrow, S. Karlin and P. Suppes (eds.) *Mathematical methods in the social sciences*, Stanford University Press, Stanford, 1960.

[252] Scarf, H.: *A survey of analytic techniques in inventory theory*, Chapter 7 in: H. Scarf, D. Gilford and M. Shelly (eds.), *Multistage inventory models and techniques*, Stanford University Press, Stanford, 1963.

[253] Schrijver, A.: *Combinatorial optimization: Polyhedra and efficiency*, Springer-Verlag, Berlin, 2003.

[254] Schweitzer, P.J.: *Perturbation theory and Markov decision chains*, PhD dissertation, Massachusetts Institute of Technology, 1965.

[255] Schweitzer, P.J.: *Multiple policy improvements in undiscounted Markov renewal programming*, Operations Research 19 (1971) 784–793.

[256] Schweitzer, P.J.: *Iterative solution of the functional equations of undiscounted Markov renewal programming*, Journal of Mathematical Analysis and Applications 34 (1971) 495–501.

[257] Schweitzer, P.J. and A. Federgruen: *The asymptotic behavior of undiscounted value iteration in a Markov decision problem*, Mathematics of Operations Research 2 (1977) 360–381.

[258] Schweitzer, P.J. and A. Federgruen: *The functional equation of undiscounted Markov renewal programming*, Mathematics of Operations Research 3 (1978) 308–321.

[259] Schweitzer, P.J. and A. Federgruen: *Foolproof convergence in multichain policy iteration*, Journal of Mathematical Analysis and Applications 64 (1978) 360–368.

[260] Schweitzer, P.J. and A. Federgruen: *Geometric convergence of value-iteration in multichain Markovian renewal programming*, Advances in Applied Probability 11 (1979) 188–217.

[261] Senata, E.: *Nonnegative matrices and Markov chains*, Springer-Verlag, 1981.

[262] Sennott, L.I.: *Stochastic dynamic programming and the control of queueing systems*, Wiley, 1999.

[263] Serfozo, R.: *Monotone optimal policies for Markov decision processes*, Mathematical Programming Study 6 (1976) 202–215.

[264] Serfozo, R.: *An equivalence between continuous and discrete time Markov decision processes*, Operations Research 27 (1979) 616–620.

[265] Serin, Y.: *Structured policies for Markov decision processes with linear constraints*, Working paper, Middle East Technical University, Ankara, Turkey (2000)).

[266] Shapiro, J.F.: *Brouwer's fixed-point theorem and finite state space Markovian decision theory*, Journal of Mathematical Analysis and Applications 49 (1975) 710–712.

[267] Shapley, L.S.: *Stochastic games*, Proceedings of the National Academy of Sciences 39 (1953) 1095–1100.

[268] Shapley, L.S. and R.N. Snow: *Basic solutions of discrete games*, in: H.W. Kuhn & A.W. Tucker (eds.), *Contributions to the theory of games*, Vol. I, Annals of Mathematical Studies no. 24, pp. 27–35, Princeton University Press (1950).

[269] Sherif, Y.S. and M.L. Smith: *Optimal maintenance policies for systems subject to failure - A review*, Naval Research Logistics Quarterly 28 (1981) 47–74.

[270] Sherali, H.N. and C.M. Shetty: *A finitely convergent algorithm for bilinear programming problems*, Mathematical Programming 19 (1980) 14–31.

[271] Shiloach, Y. and U. Vishkin: *On* $\mathcal{O}(\updownarrow \wr \} \backslash)$ *parallel connectivity algorithm*, Journal of Algoriithms 3 (1982) 14–31.

[272] Sinha, S.: *An extension theorem for the class of stochastic games having ordered field property*, Opsearch 23 (1986) 197–205.

[273] Sladky, K.: *On the set of optimal controls for Markov chains with rewards*, Kybernetica 10 (1974) 350–367.

[274] Smallwood, R.D.: *Optimum policy regions for Markov processes with discounting*, Operations Research 14 (1966) 658–669.

[275] Smith, D.R.: *Optimal repair of a series system*, Operations Research 26 (1978) 653–662.

[276] Sobel, M.J.: *Noncooperative stochastic games*, Annals of Mathematical Statistics 42 (1971) 1930–1935.

[277] Sobel, M.J.: *Myopic solutions of Markov decision processes and stochastic games*, Operations Research 29 (1981) 995–1009.

[278] Sobel, M.J.: *Maximal mean/standard deviation ratio in an undiscounted MDP*, OR Letters 4 (1985) 157–159.

[279] Sobel, M.J.: *Mean-variance tradeoffs in undiscounted MDP*, Operations Research 42 (1994) 175–183.

[280] Spreen, D.: *A further anticycling rule in multichain policy iteration for undiscounted Markov renewal programs*, Zeitschrift für Operations Research 25 (1981) 225–233.

[281] Stein, J.: *On efficiency of linear programming applied to discounted Markovian decision problems*, OR Spektrum 10 (1988) 153-160.

[282] Stern, M.: *On stochastic games with limiting average payoff*, PhD thesis, University of Illinois at Chicago, Chicago, 1975.

[283] Stoer, J. and R. Bulirsch: *Introduction to numerical analysis*, Springer,1980.

[284] Stoer, J. and C. Witzgall: *Convexity and optimization in finite dimensions*, Springer,1970.

[285] Strauch, R.: *Negative dynamic programming*, Annals Mathematical Statistics 37 (1966) 871–889.

[286] Strauch, R. and A.F. Veinott Jr.: *A property of sequential control processes*, Report, Rand McNally, Chicago, 1966.

[287] Tarjan, R.E.: *Depth-first search and linear graph algorithms*, SIAM Journal of Computing 1 (1972) 146–160.

[288] Tijms, H.C.: *Stochastic models: An algorithmic approach*, Wiley Series in Probability and Mathematical Statistics, Wiley, 1994.

[289] Topkis, D.: *Minimizing a submodular function on a lattice*, Operations Research 26 (1978) 305–321.

[290] Tseng, p.: *Solving H-horizon, stationary Markov decision problems in time proportional to $\log(H)$*, Operations Letters 9 (1990) 287–297.

[291] Tsitsiklis, J.N.: *A lemma on the multi-armed bandit problem*, IEEE Transactions on Automatic Control 31 (1986) 576–577.

[292] Tsitsiklis, J.N.: *A short proof of the Gittins index theorem*, Annals of Applied Probability 4 (1994) 194–199.

[293] Tsitsiklis, J.N.: *NP-hardness of checking the unichain condition in average cost MDPs*, Operations Research Letters 35 (2007) 319–323.

[294] Van der Wal, J.: *Discounted Markov games: successive approximations and stopping times*, International Journal of Game Theory 6 (1977) 11–22.

[295] Van der Wal, J.: *Successive approximations for average reward Markov games*, International Journal of Game Theory 9 (1980) 13–24.

[296] Van der Wal, J.: *The method of value oriented successive approximation for the average reward Markov decision processes*, OR Spektrum 1 (1980) 233–242.

[297] Van der Wal, J.: *Stochastic dynamic programming*, Mathematical Centre, Amsterdam, 1981.

[298] Van der Wal, J. and J.A.E.E. Van Nunen: *A note on the convergence of value oriented successive approximations method*, Report, Eindhoven University of Technology, 1977.

[299] Van der Wal, J. and J. Wessels: *Successive approximations for Markov games*, in: H. Tijms and J. Wessels (eds.), *Markov decision theory*, Mathematical Centre Tract no. 93, 1977, Amsterdam.

[300] Van Hee, K.M., A. Hordijk and J. Van der Wal: *Successive approximations for convergent dynamic programming*, in: H.C. Tijms and J. Wessels (eds.) *Markov decision theory*, Mathematical Centre Tract no.93, Mathematical Centre, Amsterdam, 1977, 183–211.

[301] Van Nunen, J.A.E.E.: *A set of successive approximation method for discounted Markovian decision problems*, Zeitschrift für Operations Research 20 (1976) 203–208.

[302] Van Nunen, J.A.E.E.: *Contracting Markov decision processes* Mathematical Centre Tract 71, Mathematical Centre, Amsterdam, 1976.

[303] Van Nunen, J.A.E.E. and J. Wessels: *A principle for generating optimization procedures for discounted Markov decision processes*, Colloquia Mathematica Societatis Bolyai Janos, Vol. 12, North Holland, Amsterdam, 1976, 683–695.

[304] Van Nunen, J.A.E.E. and J. Wessels: *The generation of successive approximations for Markov decision processes using stopping times*, in: H. Tijms and J. Wessels (eds.) *Markov decision theory*, Mathematical Centre Tract no.93, Mathematical Centre, Amsterdam, 1977, 25–37.

[305] Van Nunen, J.A.E.E. and J. Wessels: *Markov decision processes with unbounded rewards*, in: H.C. Tijms and J. Wessels (eds.) *Markov decision theory*, Mathematical Centre Tract no.93, Mathematical Centre, Amsterdam, 1977, 1–24.

[306] Varaiya, P.P., J.C. Walrand and C. Buyukkoc: *Extensions of the multi-armed bandit problem: the discounted case*, IEEE Transactions on Automatic Control 30 (1985) 426–439.

[307] Veinott, A.F. Jr.: *Optimal policy for a multi-product, dynamic nonstationary inventory problem*, Management Science 12 (1965) 206–222.

[308] Veinott, A.F. Jr.: *On finding optimal policies in discrete dynamic programming with no discounting*, Annals of Mathematical Statistics 37 (1966) 1284–1294.

[309] Veinott, A.F. Jr.: *On the optimality of (s, S) inventory policies: new conditions and a new proof*, SIAM Journal on Applied Mathematics 14 (1966) 1067–1083.

[310] Veinott, A.F. Jr.: *Discrete dynamic programming with sensitive discount optimality criteria (preliminary report)*, Annals of Mathematical Statistics 39 (1968) 1372.

[311] Veinott, A.F. Jr.: *Discrete dynamic programming with sensitive discount optimality criteria*, Annals of Mathematical Statistics 40 (1969) 1635–1660.

[312] Veinott, A.F. Jr.: *Markov decision chains*, in: G.B.Dantzig and B.C.Eaves (eds.) *Studies in Mathematics, vol. 10: studies in optimization*, The Mathematical Association of America (1974) 124–159.

[313] Von Neumann, J. and O. Morgenstern: *The theory of games and economic behaviour*, Princeton University Press, 1950.

[314] Vrieze, O.J.: *Linear programming and undiscounted stochastic games*, OR Spektrum 3 (1981) 29–35.

[315] Vrieze, O.J.: *Stochastic games with finite state and action spaces*, CWI Tracts 33, 1987.

[316] Vrieze, O.J., S.H. Tijs, T.E.S. Raghavan and J.A. Filar: *A finite algorithm for the switching controller stochastic game*, OR Spektrum 5 (1983) 15–83.

[317] Wagner, H.M. and T. Whithin: *Dynamic problems in the theory of the firm*, T. Whithin (ed.): *Theory of inventory management*, App. 6, 2nd ed., Princeton University Press, 1957.

[318] Walrand, J.: *An introduction to queueing networks*, Prentice-Hall, Englewood Cliffs, New Jersey, 1988.

[319] Weber, R.R.: *Scheduling jobs with stochastic processing requirements on parallel machines to minimize makespan or flowtime*, Journal of Applied Probabitily 19 (1982) 167–182.

[320] Weber, R.R.: *On the Gittins index for multi-armed bandits*, Annals of Applied Probabitily 2 (1992) 1024–1033.

[321] Weiss, G.: *Multiserver stochastic scheduling*, in: Dempster, M.A.H., J.K. Lenstra and A.H.G. Rinnooy Kan (eds.), *Deterministic and stochastic scheduling*, Reidel, Dordrecht, Holland (1982) 157–179.

[322] Weiss, G.: *Braching bandit processes*, in: Probability in the Engineering and Informational Sciences 2 (1988) 269–278.

[323] Wessels, J.: *Stopping times and Markov programming*, in: Transactions of the 7-th Prague conference on information theory, statistical decision functions and random processes, Academia, Prague (1977) 575–585.

[324] Wessels, J. and J.A.E.E. Van Nunen: *Discounted semi-Markov decision processes: linear programming and policy iteration*, Statistica Neerlandica 29 (1975) 1–7.

[325] White, D.J.: *Dynamic programming, Markov chains and the method of successive approximations*, Journal of Mathematical Analysis and Applications 6 (1963) 373–376.

[326] White, D.J.: *Dynamic programming and probabilistic constraints*, Operations Research 22 (1974) 654–664.

[327] White, D.J.: *Multi-objective infinite-horizon discounted Markove decision processes*, Journal of Mathematical Analysis and Applications 89 (1982) 639–647.

[328] White, D.J.: *Monotone value iteration for discounted finite Markov decision processes*, Journal of Mathematical Analysis and Applications 109 (1985) 311-324.

[329] White, D.J.: *Mean, variance and probabilistic criteria in finite decision processes: a review*, Journal of Optimization Theory and Applications 56 (1988) 1–30.

[330] White, D.J.: *Computational approaches to variance-penalized Markov decision processes*, OR Spektrum 14 (1992) 79–83.

[331] White, D.J.: *A mathematical programming approach to a problem in variance penalised Markov decision processes*, OR Spektrum 15 (1994) 225–230.

[332] Whittle, P.: *Multi-armed bandits and the Gittins index*, Journal of the Royal Statistical Society, Series B 42 (1980) 143–149.

[333] Whittle, P.: *Optimization over time*, Wiley, 1982.

[334] Widder, D.V.: *The Laplace transform*, Princeton University Press, Princeton, New Jersey, 1946.

[335] Winston, W.: *Optimality of the shortest line discipline*, Journal of Applied Probability 14 (1977) 181–189.

[336] Ye, Y.: *A new complexity result on solving the Markov decision problem*, Mathematics of Operations Research 30 (2005) 733–749.

[337] Ye, Y.: *The simplex and policy-iteration methods are strongly polynomial for the Markov decision problem with a fixed discount rate*, Mathematics of Operations Research 36 (2011) 593–603.

[338] Yu, P. and M. Zeleny: *The set of all nonrandomized solutions in linear cases and multicriteria simplex method*, Journal of Mathematical Analysis and Applications 49 (1975) 430–468.

[339] Zangwill, W.I.: *A deterministic multi-period production scheduling model with backlogging*, Management Science 13 (1966) 105–119.

[340] Zangwill, W.I.: *A backlogging model and a multi-echelon model of a dynamic economic lot size production system - a network approach*, Management Science 15 (1969) 506–527.

[341] Zoutendijk, G.: *Mathematical programming methods*, North Holland, 1976.

Index