What Multistage Stochastic Programming Can Do for Network Revenue Management*

Victor DeMiguel   Nishant Mishra †

July 3, 2008

*We thank Kristin Fridgeirsdottir, Catalina Stefanescu, and Stefanos Zenios for their help during various stages of writing this paper. We would like to thank comments from Zeger Degraeve, Guillermo Gallego, J. Michael Harrison, Tito Homem-de-Mello, Houyuan Jiang, Michal Kaut, Ioana Popescu, Kalyan Talluri, and Garrett J. van Ryzin. The paper has also benefited from the comments of participants at the following events: the INFORMS Annual meeting 2005, the 5^{th} Annual INFORMS Revenue Management Section Conference, the Tenth International Conference on Stochastic Programming, and seminars at the Sauder School of Commerce of University of British Columbia and at the Judge Institute of Management of Cambridge University.

†London Business School, 6 Sussex Place, Regent's Park, London, United Kingdom NW1 4SA; Email: {avmiguel,nmishra.phd2003}@london.edu.

What Multistage Stochastic Programming Can Do for Network Revenue Management

Abstract

Airlines must dynamically choose how to allocate their flight capacity to incoming travel demand. Because some passengers take connecting flights, the decisions for all network flights must be made simultaneously. To simplify the decision making process, most practitioners assume demand is deterministic and equal to average demand. We propose a multistage stochastic programming approach that models demand via a scenario tree and can accommodate any discrete demand distribution. This approach reflects the dynamic nature of the problem and does not assume the decision maker has perfect information on future demand. We consider four different methodologies for multistage scenario tree generation (Monte-Carlo sampling, principal-component sampling, moment matching, and bootstrapping) and conclude that the sampling methods are best. Finally, our numerical results show that the multistage approach performs significantly better than the deterministic approach and that revenue managers who ignore demand uncertainty may be losing between 1% and 2% in average revenue. Moreover, the multistage approach is also significantly better than the randomized linear programming approach of Talluri and Van Ryzin [23] provided the multistage scenario tree has a sufficiently large number of branches.

Keywords: Network revenue management, multistage stochastic programming, scenario tree generation

1 Introduction

Airlines operating a network of flights must dynamically choose how to allocate their capacity to incoming travel demand. Because some of the passengers need to take connecting flights to reach their destinations, the allocation decisions for all flights in the network must be made simultaneously. We refer to this problem as the network revenue management problem. Closely related problems can be found in the railway, shipping, and hotel industries, but in this paper we focus on the airline industry for clarity of exposition. See the book by Talluri and VanRyzin for a detailed review of research in network revenue management [24, Chapter3].

A popular approach to this problem is to use bid price policies. A bid price is an estimate of the future revenue we expect to collect from the operation of a given seat in a given flight. Given a set of bid prices, we accept a request for air travel if the revenue associated with the corresponding ticket exceeds the sum of the bid prices of all seats used by the traveller. Bid price policies are not optimal in general but they are very popular because they are intuitive and easy to implement.

The performance of a bid price policy depends crucially on the quality of the calculated bid prices. To compute bid prices, one has to first solve an optimization problem whose objective is to maximize the expected total revenue given the available capacity and the distribution of future demand. The bid prices are then computed as the Lagrange multipliers of the capacity constraints of this optimization problem. Because the dynamic program associated with the network revenue management problem is computationally intractable, we usually solve instead an optimization problem that is only an approximation to the true problem. This optimization problem is then re-solved periodically, in an attempt to obtain up-to-date bid prices and thus improve performance.

The simplest (and also perhaps the most popular) optimization model for bid price generation is the so-called deterministic linear program (DLP). The DLP maximizes total revenue assuming demand is deterministic and equal to average demand. Williamson [26] showed that the DLP usually performs better than more complicated optimization models such as the probabilistic nonlinear program (PNLP), which models uncertainty through a demand distribution. De Boer et al. [7] later argued that the PNLP performs poorly because it does not take into account the phenomenon of capacity *nesting*, by which a seat originally reserved for a particular product (itinerary-fare-class combination) may be reallocated to a different product if this is judged to be optimal once demand is realized. Talluri and VanRyzin [23] proposed the randomized linear program (RLP), an elegant extension of the DLP that captures greater demand uncertainty while circumventing some of the difficulties associated with the lack of nesting of the PNLP approach. The catch is that the RLP model assumes the decision maker has perfect information on future demand when making capacity allocation decisions. Despite this, Talluri and VanRyzin showed that the RLP usually outperforms the DLP.[1]

A natural extension to the DLP, PNLP, and the RLP models are models based on *stochastic programming* [3]. Intuitively, they should outperform the aforementioned models because they incorporate more information about demand uncertainty and they better reflect the dynamic nature of the problem. De Boer et al. [7] showed that the PNLP can be reformulated as a two-stage stochastic program (SP) and studied the impact of the lack of nesting on the performance of the PNLP. Higle and Sen [11] also proposed a two-stage SP model, but tried to overcome the difficulties

[1]Other mathematical programming approaches for generating bid prices include the use of approximate dynamic programming, as proposed by Popescu and Bertsimas [2]. They use the linear programming relaxation to a dynamic program to generate the bid prices. Adelman [1] showed that the DLP can be derived as a linear functional approximation to the optimal dynamic programming value function.

associated with the PNLP by using a different formulation in which they allocate capacity for each fare class *in each leg*; this allowed for a better representation of unused capacity. Doing so, they report the two-stage stochastic program to perform better than the DLP on average. Chen and Homem-de-Mello [5] considered a multistage stochastic programming (MSP) formulation of the problem that is basically a multi-period extension of the PNLP. They showed that the resulting model suffers from a lack of convexity, which makes it difficult to use in practice. So instead, they proposed solving a two-stage SP model (similar to De Boer et al.) “on a rolling-horizon basis” as an approximation. They also provided some theoretical insights—a key result being that re-solving the SP can only help (which Cooper’s counterexample shows is not true in the case of the DLP [6]). It must be noted that both De Boer et al. and Chen and Homem-de-Mello have primarily focussed on *two-stage* models and have assumed the demand for different products is not correlated.

Möller et al. [17] is the only other paper that proposes the use of a multistage stochastic programming (MSP) model for bid price generation. Like the model by Chen and Homem-de-Mello, this approach is a multistage extension of the PNLP approach, but Möller et al. manage to keep their problem convex by introducing cumulative-sales variables in their model. The authors, however, do not perform any out-of-sample simulations and thus, they do not give any comparisons of the performance of their multistage model with that of other existing models (DLP, PNLP, or RLP). Finally, note that because they are extensions of the PNLP approach, both approaches by Chen and Homen-de-Mello and Möller et al. are likely to suffer from the performance difficulties associated with the lack of nesting of the PNLP.

Our contribution is threefold. First, we propose a novel multistage stochastic programming methodology for bid price generation. Our approach is based on a scenario tree for travel demand that makes no assumptions on demand distribution, and thus, unlike the approaches in [7] and [5], can incorporate information on inter-temporal and inter-product demand correlation. We assume the decision maker has perfect information on the demand at the *present* stage, but knows future demand only in distribution. In this sense, our model is more realistic than the RLP approach (which assumes perfect information on all demand) but not as realistic as the models in [5] and [17] (which do not assume perfect information even on demand at the present stage). The advantage of introducing this “semi-non-anticipativity” is that by giving up a bit of realism, our approach circumvents the difficulties associated with the lack of nesting present in the PNLP and its extensions in [5] and [17]. [2]

Our second contribution is to perform a rigorous investigation on how to generate multistage scenario trees for network revenue management. In particular, we consider four scenario generation methodologies: Monte-Carlo sampling, principal-component sampling, moment matching, and finally bootstrapping, which is the methodology used in [17]. We compare the performance of these methodologies according to their in-sample and out-of-sample stability along the lines suggested by Kaut and Wallace [15]. Our analysis shows that Monte-Carlo and principal component sampling are the most stable.

Our third contribution is to perform rolling-horizon simulations to compare the MSP approach with other existing approaches such as the DLP and the RLP. To the best of our knowledge, these are the *first* out-of-sample results for MSP approaches to bid-price generation. Our results demonstrate that there are significant improvements possible by using an MSP approach for bid price generation. In particular, our results show that our approach outperforms the DLP by 1% to

[2] At this point we would like to emphasize that the objective of this paper is to study the effects of having a better optimization approach for generating bid prices and for this reason (as also for the purpose of tractability), we do not consider a more complicated demand model (such as, one involving discrete consumer choice behaviour).

2% and that this difference is statistically significant. Moreover, the MSP approach also performs *significantly better* than the RLP approach, although, the improvement is not as much as in the case of the DLP. A further sensitivity analysis shows that the relative advantage of the MSP approach is higher when the demand correlation and variability are higher.

Finally, based on our simulation experience, we would like to add that the MSP approach looks quite promising for use in real world applications. This is because, even using traditional LP solvers for solving the MSP, we were able to handle (with relative ease) large problem sizes similar to those encountered in practical settings and demonstrate significant benefits over the DLP and the RLP. Needless to say, if special purpose algorithms such as decomposition methods are implemented, we can solve much larger problems much more quickly.

The remainder of the paper is organized as follows. In Section 2 we review the existing optimization models for bid price generation and illustrate their shortcomings with a simple example. In Section 3 we propose the multistage stochastic programming approach. In Section 4 we describe various scenario generation methodologies and also discuss how the performance of these methodologies can be compared according to their in- and out-of-sample stability. In Section 5 we outline the simulation process and present the numerical results obtained. Section 6 concludes.

2 Optimization models for bid price generation

The network revenue management problem can be formulated as a dynamic program, but the astronomical size of its state space renders the problem intractable.[3] For this reason, practitioners and academics have turned to bid price policies, which are not optimal in general but offer a tractable approach to the problem. A bid price is an estimate of the future revenue we expect to collect from the operation of a given seat in a given flight. Given a set of bid prices, we accept a request for air travel if the revenue associated with the corresponding ticket exceeds the sum of the bid prices of all seats used by the traveller. Bid price policies are intuitive and easy to implement because they only require the storage of a vector of bid prices. Moreover, it has been shown that bid price policies are asymptotically optimal as demand and capacity grow large; see [22] and the references therein.

The practical performance of a bid price policy is determined by the quality of the bid prices generated. To compute bid prices, one has to first solve an optimization problem whose objective is to maximize the expected total revenue given the capacity available and the distribution of future demand. The bid prices are then computed as the Lagrange multipliers or shadow prices of the capacity constraints of this optimization problem. Moreover, because the dynamic program that represents the network revenue management problem is intractable, the optimization models we solve to generate bid prices are usually just an approximation to the true problem. Hence it is clear that the better the approximation, the higher the quality of the generated bid prices and the better the performance of the corresponding bid-price policy. In this section we review some of the most popular bid price generation procedures proposed in the literature and illustrate through a simple example how these methods can fail to give the optimal policy.

We will use the following notation. We consider a network with m resources (cabin in a given flight) and n products (a given itinerary, fare, and class combination). We denote as $A \in \Re^{m\times n}$ the *resource-product matrix*, where a_{ij} equals one if the jth product utilizes the ith resource and

[3] For instance, the size of the state space corresponding to the problem of a major airline operating a network of 5000 flights with a capacity of 300 seats each is 300^{5000}.

zero otherwise. Likewise, we denote by r the *product revenue vector*, where r_j is the revenue from the jth product.

2.1 The deterministic linear program

The DLP maximizes the total revenue based on expected demand. Namely,

$$\boxed{\begin{array}{llll} \text{DLP} & \quad\quad\quad\quad \underset{u}{\text{maximize}} & r^T u & \\ & \quad\quad\quad\quad \text{subject to} & 0 \leq u \leq E[D] & \\ & & \quad Au \leq x, & \end{array}} \tag{1}$$

where $u \in \Re^n$ is the vector of protection levels (that is, the capacity reserved for each particular product), $E[D] \in \Re^n$ is the vector of expected demand for all n products, and $x \in \Re^m$ is the capacity (seats available) for all m resources. The bid prices are the Lagrange multipliers of the resource constraints $Au \leq x$.[4] The simple and computationally efficient structure of the DLP contributed to its widespread acceptance in industry. However, because the DLP uses only the expected demand to compute the bid prices (neglecting any other distributional information), one would expect the generated bid prices to be poor approximations to the marginal value of resources. To illustrate this point consider the following simple example.

Example 1 *Consider a single flight with a capacity of two seats ($x = 2$) and demand for three fare classes (business, premier, and economy) with associated revenues $r_B > r_P > r_E$. Assume demand arrives in two different stages. In the first stage, there is deterministic demand for three economy class seats, and no demand for business or premier class seats. In the second stage, we consider three equally likely scenarios. In the first scenario, there is demand for three business class seats and no demand for premier or economy class, in the second scenario, there is demand for three premier class seats and no demand for business or economy, and in the third scenario there is no demand for any of the three classes. Concretely, if D^s_{it} denotes the demand for fare class i in period t, and scenario s, where, $i \in \{B, P, E\}$, $t \in \{1, 2\}$ and $s \in \{1, 2, 3\}$, we consider the following three demand scenarios $\{D^1_{E1} = 3, D^1_{B2} = 3\}$, $\{D^2_{E1} = 3, D^2_{P2} = 3\}$, $\{D^3_{E1} = 3, \text{and all other demand equal to zero}\}$. This example is represented in Figure 1.*

The optimal policy for the above example is dependent on the relationship between the revenues from the fare-classes. If revenue from business and premier classes is significantly higher than revenue from economy class ($\frac{r_B + r_P}{3} > r_E$), then it is optimal to reject all economy class demand in the first stage and accept all business and premier class demand in the second stage to make an expected revenue of $2(r_B + r_P)/3$. Else, if the revenue from business and premier class is not sufficiently high ($(r_B + r_P)/3 \leq r_E$), it would be optimal to accept all economy class demand at time stage 1 to obtain a total revenue of $2r_E$.

[4] Note that one could use the protection levels u to make decisions on capacity allocation. But practitioners usually prefer to use the bid prices instead because they have a clear economic interpretation as the marginal value of each seat, they are easier to implement in practice because we only need to record one bid price per resource, and they usually perform better in practice; see [24, Chapter 3]. For this reason, in this paper we focus on bid prices.

The DLP corresponding to Example 1 is:

$$\begin{array}{ll} \max\limits_{u_B,u_P,u_E} & r_B u_B + r_P u_P + r_E u_E \\ \text{subject to} & u_B + u_P + u_E \leq 2, \\ & u_B \leq 1, u_P \leq 1, u_E \leq 3, \\ & u_B, u_P, u_E \geq 0. \end{array}$$

It is easy to see that the solution to this DLP is $(u_B, u_P, u_E) = (1, 1, 0)$. If we could increase the right-hand side of the capacity constraint $u_B + u_P + u_E \leq 2$ by one unit, the objective of the DLP would increase by r_E and thus the resulting bid price would be r_E.[5] This implies that two economy class passengers must be accepted but this policy is suboptimal for the case $(r_B + r_P)/3 > r_E$. The DLP fails to find the optimal capacity allocation for this example because it only considers mean demand and thus it fails to account for the upside potential associated with the high demand for business and premier class seats in scenarios 1 and 2. As a result, it yields a bid price that is too small and does not reflect the true potential for revenue.

2.2 The probabilistic nonlinear program

The PNLP models demand through a finite set of scenarios and finds the protection levels that maximize the *expected* revenue given this finite scenario set. The problem may be formulated as:

$$\boxed{\begin{array}{lll} \text{PNLP} & \underset{u}{\text{maximize}} & E[r^T \min(u, D)] \\ & \text{subject to} & u \geq 0 \\ & & Au \leq x, \end{array}} \tag{2}$$

where D is the demand, which is assumed to follow a discrete distribution, and u are the protection levels. The PNLP can be reformulated as a stochastic program with *simple recourse* and can be solved numerically. But Williamson [26] showed that the PNLP usually does not perform as well as the DLP in practice. De Boer et al. [7] argued that the reason for this is that the PNLP does not allow for capacity *nesting*. In particular, notice that the PNLP makes a decision on how to partition the available capacity among all products in the first stage, and that it does not allow one to change that partition in the second stage depending on the realization of demand. To illustrate this point, reconsider Example 1. The corresponding PNLP is:

$$\begin{array}{ll} \max\limits_{u_B,u_P,u_E} & \frac{1}{3}(r_B \min(u_B, 3) + r_P \min(u_P, 0) + r_E \min(u_E, 3)) + \\ & \frac{1}{3}(r_B \min(u_B, 0) + r_P \min(u_P, 3) + r_E \min(u_E, 3)) + \\ & \frac{1}{3}(r_B \min(u_B, 0) + r_P \min(u_P, 0) + r_E \min(u_E, 3)) \\ \text{subject to} & u_B + u_P + u_E \leq 2, \\ & u_B, u_P, u_E \geq 0. \end{array}$$

This PNLP fails to recognize that the same capacity can be used to meet business and premier demand because these two demands occur in different scenarios. Specifically, the PNLP must make a decision at the first stage whether to reserve the capacity for premier or business, but does not account for the possibility of reserving the same capacity for both business and premier demand; a

[5] Note that the solution to the DLP is degenerate and thus its corresponding Lagrange multipliers are not unique. The Lagrange multiplier for the capacity constraint could be any value in the interval $[r_E, (r_B + r_P)/3]$. Whichever value we choose in this interval, the corresponding bid price policy will be wrong either for the case $(r_B + r_P)/3 < r_E$ or for the case $(r_B + r_P)/3 > r_E$.

situation known as capacity nesting. As a result, the solution to the PNLP is $(u_B, u_P, u_E) = (0, 0, 2)$ and the associated Lagrange multiplier or bid price is r_E, provided that $r_B/3 < r_E$. This implies that economy class demand should be accepted and this policy is clearly suboptimal for the case $(r_B + r_P)/3 > r_E > r_B/3$.

2.3 The randomized linear program

Talluri and VanRyzin [23] proposed considering a finite set of demand scenarios, all with equal probability, and then solving a different DLP for each of these scenarios. The bid prices are then computed as the average of the Lagrange multipliers corresponding to the DLPs that are solved. The resulting method is known as the randomized linear program (RLP). Namely, for each demand scenario $D(w)$ for $w = 1, \ldots, W$, solve the following linear program:

$$\text{RLP} \qquad \begin{array}{ll} \underset{u}{\text{maximize}} & r^T u(w) \\ \text{subject to} & 0 \leq u(w) \leq D(w) \\ & Au(w) \leq x, \end{array} \tag{3}$$

where $u(w)$ is the protection level. The bid prices are estimated as $\frac{1}{N}\sum_{w=1}^{N} \mu(w)$, where $\mu(w)$ are the Lagrange multipliers corresponding to the constraints $Au(w) \leq x$. This approach uses a better representation of uncertainty than the DLP approach because it considers a number of demand outcomes rather than just the mean demand. Moreover, Talluri and VanRyzin show in their experiments that the RLP outperforms the DLP, although not uniformly.

An advantage of the the RLP approach is that it does not suffer from the lack of capacity nesting associated with the PNLP. In particular, because the DLP corresponding to each realization of demand is solved independently, this method generates different protection levels for each demand scenario and thus capacity nesting occurs automatically. A shortcoming of the RLP, however, is that it is not realistic to assume that the airline can choose different protection levels for each scenario. This is basically equivalent to assuming the airline has perfect information on demand when making its capacity allocation decisions and this is clearly not the case in reality. To see the potential harmful effects of this shortcoming, reconsider Example 1. To apply the RLP method, for each of the three scenarios $s \in \{1, 2, 3\}$, we need to solve the following linear program:

$$\begin{array}{ll} \underset{u_B^s, u_P^s, u_E^s}{\text{maximize}} & r_B u_B^s + r_P u_P^s + r_E u_E^s \\ \text{subject to} & u_B^s + u_P^s + u_E^s \leq 2, \\ & u_B^s \leq D_B^{2k}, u_P^s \leq D_P^{2k}, u_E^s \leq 3, \\ & u_B^s, u_P^s, u_E^s \geq 0. \end{array}$$

The solution is $(u_B^1, u_P^1, u_E^1) = (2, 0, 0)$ for the first scenario, $(u_B^2, u_P^2, u_E^2) = (0, 2, 0)$ for the second scenario, and $(u_B^3, u_P^3, u_E^3) = (0, 0, 2)$ for the third scenario, and the resulting bid price is $(r_B + r_P + r_E)/3$. The implied solution is to reject all economy class demand and this is suboptimal for the case $(r_B + r_P)/3 < r_E$. Roughly speaking, by assuming the decision maker has perfect information on all (present and future) demand, the RLP overestimates the upside potential associated with the business and premier class demand because it assumes we can fully benefit from this demand in scenarios 1 and 2 while still accepting economy class demand under scenario 3, where there is no business and premier demand.

In the following section, we present the MSP formulation and see how it generates the optimal policy for the above example by correctly differentiating between the various fare-classes.

3 The multistage stochastic programming model

In this section, we propose a multistage stochastic programming (MSP) model for bid-price generation. The term multistage refers to the ability to make decisions at multiple stages, and thus, this type of models can reflect the dynamic nature of the network revenue management problem. This is an advantage with respect to the DLP and RLP, which are essentially single-stage models, and also with respect to the models in [7, 5, 11], which are two-stage models.

In addition, our approach models travel demand via a multistage scenario that makes no assumptions on demand distribution, and thus, unlike the approaches in [7] and [5], can incorporate information on inter-temporal and inter-product demand correlation.

The only other model from the literature that is comparable to ours is that proposed by Möller et al. [17], which is also a multistage model and allows for general demand distributions. But they only consider one scenario generation methodology (bootstrapping) and they do not provide out-of-sample simulation results comparing their model to other models such as the DLP and the RLP. We consider four scenario generation methodologies in Section 4 and provide out-of-sample results comparing our method to DLP and RLP in Section 5.

Another feature that distinguishes our approach from that by Möller et al. [17] is that we assume the decision maker has perfect information on the demand at the *present* stage, although knows future demand only in distribution. In this sense, our model is more realistic than the RLP approach (which assumes perfect information on all demand) but not as realistic as the models in [5] and [17] (which do not assume perfect information even on demand at the present stage). The advantage is that by giving up a bit of realism, our approach circumvents the difficulties associated with the lack of nesting present in the PNLP and its extensions in [5] and [17].

In the remainder of this section, we give a detailed description of our approach. We model demand via a multistage scenario tree (represented in Figure 2). The demand at time stage $t = 1$ is assumed to be the same for all scenarios. At time stage $t = 2$ we consider a number B of possible realizations of demand. Conditional on each of these realizations of demand at $t = 2$, we assume there are B possible realizations of demand at time $t = 3$, and so on and so forth. The result is that we have B^{T-1} different scenarios of demand. Note that at stage t the scenarios can be bundled into B^{t-1} information sets and all scenarios in the same information set have the same demand up to stage t. Finally, as mentioned above, this scenario tree can accommodate any demand distribution and thus can account for inter-temporal and inter-product demand correlation.

We assume that, at any time stage and scenario (that is, at any node in the multistage scenario tree), the airline knows the exact demand it faces at the present node, and decides how much to sell (i.e., how much capacity to allocate to present demand), knowing only the probability that future demand will be represented by each of the scenarios that emerge from the current tree node. That is, the airline decides on the sales at the current node with perfect information on demand at the present stage, but knowing future demand only in distribution. This contrasts with the assumption in the model by Möller et al. [17], who assume the airline always makes decisions on protection levels for future demand only, and uses the protection levels chosen at previous time stages to determine how much to sell in the current stage. Basically, the decision variables in our model are sales at the current stage, whereas the decision variables in the model by Möller et al. are protection levels for sales at the next time stage. As discussed above, the model by Möller et al. is slightly more realistic (more nonanticipative) that ours. The advantage of our model is that by using sales (as opposed to protection levels), it circumvents the lack of nesting difficulties associated with the PNLP and its multistage extensions.

The MSP model can be formulated as follows:

$$
\boxed{
\begin{array}{lll}
\text{MSP} & \underset{u_t^s}{\text{maximize}} & \sum_{s=1}^{S} p_s \sum_{t=1}^{T} r_t^T u_t^s \\
& \text{subject to} & 0 \le u_t^s \le D_t^s, \quad s = 1, \ldots, S, \\
& & \displaystyle\sum_{t=1}^{T} A u_t^s \le x, \quad s = 1, \ldots, S, \\
& & u_t^s \in \mathcal{N},
\end{array}
}
\tag{4}
$$

where S is the number of scenarios, p_s is the probability associated with each scenario, T is the number of time stages, r_t is the product revenue vector for stage t, D_t^s is the demand at time t under scenario s, u_t^s are the sales at time t under scenario s, and $\mathcal{N}$ is the set of *nonanticipative* policies; that is, the set of policies such that $u_t^{s_1} = u_t^{s_2}$ for all scenarios s_1 and s_2 that have identical demand up to time stage t. Note that the objective function is the expected revenue over all demand scenarios and the constraints must hold for every scenario in the stochastic tree. The bid prices are estimated as the *sum* of the multipliers for the constraints $\sum_{t=1}^{T} A u_t^s \le x$ corresponding to all scenarios.

To illustrate the proposed approach, consider the following MSP re-formulation of Example 1

$$
\begin{array}{ll}
\underset{u}{\text{maximize}} & r_E u_{E1} + \frac{1}{3}(r_B u_{B2}^1 + r_P u_{P2}^1) + \\
& \frac{1}{3}(r_B u_{B2}^2 + r_P u_{P2}^2) + \frac{1}{3}(r_B u_{B2}^3 + r_P u_{P2}^3) \\
\text{subject to} & u_{E1} \le 3, u_{B1} \le 0, u_{P1} \le 0 \\
& u_{B2}^1 \le 3, u_{B2}^2 \le 0, u_{B2}^3 \le 0 \\
& u_{P2}^1 \le 0, u_{P2}^2 \le 3, u_{P2}^3 \le 0 \\
& u_{E1} + u_{B2}^s + u_{P2}^s \le 2, \quad s \in \{1, 2, 3\} \\
& u_{Et}, u_{Pt}^s, u_{Bt}^s \ge 0 \quad s \in \{1, 2, 3\}, \quad t \in \{1, 2\},
\end{array}
$$

where u_{Et} is the sales for economy class at time stage t, and u_{Bt}^s and u_{Pt}^s are the sales for business and premier classes at time t and scenario s. This MSP captures the information in the scenarios correctly and gives the optimal policy depending on the revenues from each fare class. In particular, if $(r_B + r_P)/3 > r_E$, then the resulting bid price is $(r_B + r_P)/3$ and the associated policy is to reject economy demand at stage 1 and accept all demand at stage 2. If, on the other hand, $(r_B + r_P)/3 < r_E$, then the bid price is r_E and the associated policy is to accept all economy demand.

4 Multistage scenario tree generation

The practical performance of the proposed MSP approach depends crucially on the quality of the multistage scenario tree used to model demand uncertainty. In this section, we describe four different methodologies for scenario tree generation: three based on estimation and one based on bootstrapping. We also discuss the methodology proposed by Kaut and Wallace [15] to evaluate scenario tree quality. In our simulations in Section 5, we apply all four scenario generation methodologies to the network revenue management problem and evaluate their performance following Kaut and Wallace's methodology.

4.1 Related work

Scenario generation for stochastic programming has been the subject of extensive research; see [3, 9, 15] and [20, Chapter 6] and the references therein. Also, see the following papers [19, 18, 4]. Recently, a few researchers have focused on the application of stochastic programming to airline network revenue management [7, 5, 11, 17]. The issue of generating scenario trees did not arise in [7] and [5] because they assumed that the demand distribution is truncated with only finitely many scenarios and that all of them could be considered. Higle and Sen [11] used sampling from the forecasted demand to discretize the distribution. Only Möller et al. [17] used a multistage stochastic programming approach and acknowledge the issue of scenario trees in their work. They construct an initial scenario tree from historic data (bootstrapping) and then employ a scenario reduction technique to reduce the total number of scenarios. However, they did not run simulations to evaluate the performance of their methodology and provide only preliminary numerical results.

4.2 Estimation-based scenario generation

In this section, we describe three scenario generation methodologies based on estimation. We assume a time series model of demand is estimated from historical data. We then wish to generate a multistage scenario tree that mimics this estimated time series as closely as possible. In this paper, we focus on the methodologies for scenario tree generation and refer the interested reader to the paper by Stefanescu [21] for a discussion of issues related to estimation.

Assume we have an estimate of a time series model of demand $\{D_t\}_{t=1}^T$. We then wish to generate a multistage stochastic tree with T time stages and B branches per node in the tree. When generating the tree, we must be careful to preserve the correlation structure present in the time series model. Concretely, if the time series model shows correlation between the demand for different products (that is, different components in the demand vector D_t) or correlation between the demand at different time periods, then we must generate a scenario tree that preserves this correlation structure. This can be achieved with the following simple procedure:

For each time stage $t = 1, \dots, T$

For each information set $i = 1, \dots, I_t$

1. **Conditional distribution:** Compute the conditional distribution of the demand at time stage t given the demand observed by scenarios in information set i up to time stage $t-1$; that is, compute the $P(D_t|D_1, \dots, D_{t-1})$.
2. **Discretization:** Generate a discrete number of demand vectors d_k for $k = 1 : B$ that represent the conditional distribution of demand $P(D_t|D_1, \dots, D_{t-1})$.

End

End

Step 1 ensures that the generated tree preserves the correlation structure between the demand at different time stages. In particular, we compute the demand distribution conditional on the demand observed in previous time stages under this particular scenario. Also, note that we only need to compute the conditional distribution and discretize it once per *information set*, where an information set at stage t is formed by all the scenarios that have the same demand up to stage t.

We now consider three different methodologies to achieve Step 2 in the proposed scenario generation algorithm: Monte-Carlo sampling, principal-component sampling, and moment matching.

4.2.1 Monte Carlo Sampling

An obvious way to discretize the conditional distribution of demand D into B realizations is Monte-Carlo sampling. A popular approach to generate Monte-Carlo samples that match a given mean and covariance matrix is to sample the so-called *principal components*; see [8]. Concretely, assume we want to sample a multivariate distribution D with mean a and covariance matrix Σ. Because the covariance matrix is symmetric, we know there exists a diagonal matrix $\Lambda \in \Re^{n\times n}$ and an orthogonal matrix $C^{n\times n}$ such that

$$\Sigma = C^T \Lambda C. \tag{5}$$

Then the so-called vector of principal components y is defined through the orthogonal linear transformation $y = C^T D$. From this definition, it is clear that the mean of the principal components vector y is equal to $C^T a$. Moreover, from (5) we know that the covariance matrix of the principal component vector is the diagonal matrix Λ, which implies that the principal components are uncorrelated. Thus, we can use traditional univariate sampling methods to sample each of the principal components independently, and then recover samples that match the original mean and covariance matrix by applying the inverse linear transformation $D = Cy$.

Note, however, that although the resulting multivariate random variable D matches the original mean and covariance matrix, it is not guaranteed to follow a general (say, Poisson) distribution because a linear combination of random variable with any general distribution does not necessarily have the same distribution. But if the principal components are sampled from a normal distribution, then the resulting D is distributed as a multivariate normal because linear combinations of normally distributed variables are normally distributed.

4.2.2 Principal components sampling

Note that the variance associated with each of the principal components is given by the elements of the diagonal matrix Λ; that is, by λ_i for $i = 1, \ldots, n$. Assume without loss of generality that the components of y are ordered in descending order of λ_i.[6] Then, out of all principal components, the first (y_1) is the one that captures the maximum variability, the next principal component captures the second largest, and so on. Specifically, the first k ($< n$) principal components then account for (in case of standardized data) the fraction $(\Sigma_{j=1}^k \lambda_j)/(\Sigma_{j=1}^n \lambda_j)$ of the variability in the original data.

A popular approach to reduce the computational effort associated with the sampling procedure is to sample only a few of the principal components (the ones with the highest values of λ_i), set the rest of the principal components equal to their mean values, and then apply the transformation $D = Cy$. This reduces the computational effort associated with sampling and should result in samples that recover most of the variability in the original distribution. We refer to this sampling methodology as principal components sampling (PC) to distinguish it from the case where we sample all principal components, which corresponds to the traditional Monte-Carlo sampling (MC).

[6] λ_i must be nonnegative by the positive semidefiniteness of the covariance matrix.

There have been some previous attempts in using PC for scenario generation ([25], [16] and [10]). In our case, the main computational effort is not associated with sampling, but with finding an optimal solution to the resulting multistage stochastic program. Thus, the computational savings associated with principal component sampling are negligible. The reason why we implement PC is to evaluate the impact on the performance of the MSP approach of neglecting part of the correlation information; that is, the impact of ignoring a few of the principal components.

4.2.3 Moment Matching

Moment matching consists of finding a finite number of samples that match as closely as possible the first few moments of the continuous distribution of demand. This method was analyzed in depth by Hoyland and Wallace in [13].

Assume we want to approximate a demand distribution with mean a and covariance matrix Σ by a discrete distribution with B outcomes $d_1, \ldots, d_B$ and probabilities $p_1, \ldots, p_B$. We would like the expectation of the discretized distribution to be a and its covariance matrix to be as close as possible to Σ. This can be accomplished by generating B perturbation vectors $\hat{d}_1, \ldots, \hat{d}_B$ with zero sample mean and sample covariance matrix close to Σ, and then setting $d_i = a + \hat{d}_i$ for $i = 1, \ldots, B$. In particular, $\hat{d}_1, \ldots, \hat{d}_B$ may be computed as the solution to the following nonlinear optimization problem:

$$\begin{array}{ll} \min\limits_{p,D} & \|\Sigma - \sum_{i=1}^{B} p_i \hat{d}_i \hat{d}_i^T\|_F^2 \\ \text{subject to} & \sum_{i=1}^{B} p_i \hat{d}_i = 0, \\ & \sum_{i=1}^{B} p_i = 1, \\ & p_i \geq 0, \quad i = 1, \ldots, B, \end{array} \tag{6}$$

where $\|\cdot\|_F^2$ is the square of the Frobenius norm; that is, the sum of the squares of the elements of the matrix. When we have an optimal objective function value of zero, we have been able to match the first two moments and this is sufficient for the case of the multivariate normal distribution.

A few comments are in order. First, note that problem (6) is a large-scale nonlinear problem. The number of variables in problem (6) is $B(n+1)$, which can be quite large in a practical setting, say for an airline offering one thousand products ($n = 1000$), and it may take several minutes to solve under such conditions.[7]

Second, it follows from the analysis in [13] that as a first estimate, one has to use $n/2 + 1$ outcomes in order to exactly match the first and second moments of D_t; that is, the scenario tree should have $n/2 + 1$ branches emanating from each node. If fewer branches are used, the second moment of D_t will not be matched exactly by the discretized distribution. As an illustration, for a large network with 1000 products, we need at least 501 branches to match the first two moments, and this would imply that there are 501,000 variables in the nonlinear program.

Finally, the problem (6) is a nonlinear nonconvex problem (i.e., it is difficult to solve) and we can only aspire to find local minimizers. To reduce complexity, we assume that all branches have the same probability of occurrence, i.e., $p_1 = p_2 = \ldots = p_B = 1/B$. This ensures the constraints remain linear and reduces the nonlinearity of the objective function.

[7] Fortunately, for multivariate normal distributions it is enough to solve problem (6) only once per time stage in the scenario tree, which reduces the computation time. This is further elaborated in Section 5.1.

4.3 Bootstrapping-based scenario generation

The scenario generation methods considered in Section 4.2 require the estimation of a time series model of demand. As an alternative, we can generate trees by bootstrapping (or sampling) from historical demand data. This procedure has several attractions: (1) we do not need to make any distributional assumptions regarding the nature of the demand process, instead the demand is assumed to be completely specified by the empirical distribution (historical data) that is on hand, and (2) this procedure can be easily implemented in practise because most airlines keep records of the customer demand for various products. One caveat while considering this approach would be that the airline records are usually *sales data* and thus, in order to extract the customer demand information, one needs to perform *uncensoring* of this data.

Bootstrapping has been used by several authors within the stochastic programming framework (see for instance [25]); however, the most relevant work, given our context, would be that of Möller et al. [17] where they generate scenario trees using historical data. This scenario tree is then modified using scenario reduction algorithms to get a more compact representation of the original set of scenarios, which is then used to solve their MSP. We employ bootstrapping along similar lines to generate scenario trees (although we do not perform any subsequent scenario reductions).

Concretely, from the historical data available, we select B^{T-1} demand paths, one per scenario. We then need to convert these B^{T-1} demand paths into a multistage scenario tree. To do so, we set the demand in the first stage of the scenario tree equal to the average of all B^{T-1} historical demand paths for time $t = 1$. For time $t = 2$, we group the historical demand paths into B different groups, one per information set, and then use the average demand for each group as the demand for each of the B information sets in the scenario tree. This process is iterated for time stages $t = 3, \ldots, T$. The result is a scenario tree that makes no distributional assumptions and yet preserves some of the correlation structure present in the data.

4.4 Scenario evaluation

It is easy to see that in order to model the demand uncertainty in the network revenue management problem, we will need to solve very large multistage stochastic programs. For instance, consider the case of a network with 1000 products, and a stochastic tree with $B = 10$ branches per node per stage. We would expect 10 branches to be on the conservative side in order to describe the uncertainty in demand. However, this will result in 1,111 nodes in a multistage stochastic tree with $T = 4$ time stages and thus, 1,111,000 variables in the equivalent deterministic linear program. Thus, it is essential to check the performance of the MSP while using only a small number of branches in the scenario tree.

Most of the theoretical work concerning scenario tree quality has been related to the asymptotic convergence of the approximating distributions to the original distribution as the number of scenarios is increased; see [3, 9, 15] and [20, Chapter 6] and the references therein. However, there are no general results that apply when the number of branches is small—in particular, how many scenarios are required to get *good* approximations, which scenario generation scheme achieves this, and what is the error in the optimal value as a function of the scenario generation procedure and the number of scenarios. It is generally agreed that such quantitative results require stronger assumptions on the problem and depend on the model (the stochastic programming formulation and the underlying uncertainty) used.

Kaut and Wallace [15] proposed two general properties that can be checked in practice and that a scenario-generation method must satisfy to be appropriate for a given model. They term these two properties in- and out-of-sample stability. To study how each of the scenario generation methods behaves in practice, in section 5, we check the in- and out-of-sample stability of the four scenario generation methods considered.

4.4.1 In-Sample Stability

We say an scenario generation method is *stable in-sample* if the optimal objective function of a set of multistage scenario trees generated using the same scenario generation methodology are similar. For instance, if we use Monte-Carlo sampling to generate K different multistage scenario trees from the same time series model of demand, we would expect the optimal objectives of these K MSPs to be similar.

Mathematically speaking, let $\{\hat{D}_t\}_{t=1}^T$ be the multistage scenario tree that approximates the original time series model of demand $\{D(t)\}_{t=1}^T$. Also, let $g(u; D_t)$ and $g(u; \hat{D}_t)$ be the objective functions for the original and the approximated problems respectively. Then if we generate K different scenario trees $\{\hat{D}_{tk}\}_{t=1}^T$ and solve the corresponding MSPs to obtain optimal solutions u_k^*, $k = 1, \ldots, K$, *in-sample* stability will imply $g(u_k^*; \hat{D}_{tk}) \approx g(u_l^*; \hat{D}_{tl})$, for $k, l \in 1, \ldots, K$.

In our numerical experiments in Section 5, we compute the standard deviation of the objective values of a set of 30 different MSPs generated by each of the four methods considered. A condition any reasonable method must satisfy is that this standard deviation should be "small" relative to the average objective value. We also analyze how this standard deviation changes with increases in the number of time stages and branches in the scenario tree.

4.4.2 Out-of-Sample Stability

Kaut and Wallace propose an out-of-sample stability concept that requires knowledge of the true demand distribution $\{D(t)\}_{t=1}^T$. In particular, they propose generating K different scenario trees and checking if $g(u_k^*; D_{tk}) \approx g(u_l^*; D_{tl})$, for $k, l \in 1, \ldots, K$. Unfortunately this concept is not practical for network revenue management, where we actually do not use the solution to the MSPs (that is, the protection levels) for decision making, but rather the associated bid prices.

In the context of the airline network revenue management problem, and following the spirit of the methodology proposed by Kaut and Wallace, we assume the true distribution is the empirical distribution of demand implied by the historical data. We use this historical data to run rolling-horizon simulations and test the stability of our scenario generation methodologies. Specifically, for each scenario generation method we run rolling-horizon simulations where the bid prices generated by the corresponding MSPs are used to make decisions whether to accept or reject travel demand on the historical dataset. At the end we compute the average total revenue associated with each scenario generation method. Moreover, we test whether this average total revenue changes when increasing the number of time stages and branches in the scenario tree. We then say that a scenario generation method is *stable out-of-sample* if the average total revenue is not very sensitive to the number of time stages and branches. If the performance was indeed sensitive to the number of stages and branches, then this would indicate that our scenario trees are not large enough.

For instance, if the out-of-sample average total revenue is \$100M when we use a scenario tree with $B = 6$ branches per time period, while it is \$101M when using $B = 12$, then the out-of-sample average total revenue changes by only 1% when we double the number of branches. This means the method is stable out-of-sample for $B = 6$, because increasing the number of branches helps very little.

5 Numerical results

In this section we compare the performance of all four MSP approaches to that of the DLP and RLP. We also compute the perfect hindsight solution, which provides an upper bound to the total average revenue. We analyze the in- and out-of-sample stability of the MSP approaches. Finally, we perform sensitivity analysis with respect to relevant parameters such as correlation, coefficient of variation, and load factor.

To carry out the comparision, we perform rolling-horizon simulations on a small and a large randomly generated networks. We assume demand follows the linear mixed effects model proposed and analyzed by Stefanescu [21]. This allows us to evaluate the impact of correlation on the performance of the different methodologies. To the best of our knowledge, we are the first to evaluate the impact of correlation on the performance of network revenue management models. The works in [7, 5, 11] assume independent demand and Möller et al. [17] do not provide any out-of-sample simulation results.

5.1 The demand model

We assume demand follows the time series model proposed by Stefanescu [21]. This model allows for correlation between the demand for different products and the demand at different time stages. The time periods are indexed by $t = 1, \ldots, T$, where T is the length of the booking horizon. The random demand vector in period t is $D_t \in \Re^n$, where the ith component of D_t is the demand for product i. The demand model is:

$$D_t = a_t + W_t \cdot v + \varepsilon_t, \quad t = 1, \ldots T. \tag{7}$$

where $a_t \in \Re^n$ is the fixed effects term (essentially a constant) representing the mean demand in period t for all products offered. The second term $v \in \Re^n$ can be seen as a common shock that affects the demand at all time stages. The effect of v at each time stage is calibrated by the constant weight matrix W_t. We assume $W_t \in \Re^{n\times n}$ is a diagonal matrix and $v \sim \text{MVN}_n(0,\ \Sigma_v)$. The final term ε_t captures the unexplained demand variability and is assumed to be normally distributed as $\varepsilon_t \sim \text{MVN}_n(0, \Sigma_\varepsilon)$ where Σ_ε is a $n \times n$ diagonal matrix. It is assumed that ε_t are independent across time periods $t = 1, \ldots, T$, and that they are independent of v. Note that because $v \sim \text{MVN}_n(0,\ \Sigma_v)$, we have that, the demand distribution for different products is correlated. Also, for any $t \neq s$ we have

$$\begin{aligned}\text{Cov}(D_t, D_s) &= \text{Cov}(a_t + W_t v + \varepsilon_t,\ a_s + W_s v + \varepsilon_s)\\ &= \text{Cov}(W_t v, W_s v) = \text{E}[(W_t v) \cdot (W_s v)']\\ &= W_t \text{E}[vv'] W_s = W_t \Sigma_v W_s.\end{aligned} \tag{8}$$

The distributional assumptions underlying this model help keep things tractable. One important byproduct of the normality assumption is that it is easy to show that the conditional covariance matrix at stage t does *not* depend on the demand observed at stages $t = 1, \ldots, t-1$,(refer Johnson and Wichern [14] for details). To see this, we first note that the distribution of D_t conditional on $D_1, \ldots, D_{t-1}$ is still multivariate normal, denoted by $\text{MVN}_n(a_c, \Sigma_c)$. While the conditional mean demand a_c is different, in general, for each node belonging to a given stage in the scenario tree, the conditional covariance matrix Σ_c is the same for all nodes belonging to the same time stage because of the properties of the multivariate normal distribution. This implies we only need to discretize the conditional distribution of demand once per time stage in the stochastic demand tree, and then compute the demand branches for each different node as $d_i = a_c + \hat{d}_i$ for $i = 1, \ldots, B$, where $\hat{d}_i$'s are the perturbation vectors discussed in Section 4.2.3. This considerably reduces the computation time required to generate the multistage demand tree.

5.2 Test networks

We consider a small and a large test networks with a hub-and-spoke topology.

5.2.1 Small Network

The small test network that was used is shown in Figure 3. The network has six single-leg flights and we consider two cabin classes—economy and business, in each flight. There are several possible origin destination combinations in this network and in order to simulate products, we consider a subset of only 4 itineraries and 8 fare classes. Two of the itineraries correspond to multiple-leg flights and two to single-leg flights. For each one of these itineraries there are 4 economy fare classes and 4 business fare classes. We assume the booking horizon is divided into four different fare periods and at each period, there is only one economy and one business fare class being offered. This is intended to represent an increase on the economy and business class fares with time. The 4 itineraries and 8 fare classes result in a total of $4 \times 8 = 32$ products. Also, in total, we have six resources in the form of three single-leg flights and two cabin classes, the first three being economy.

The fare vector, for the 8 basic products, was considered as [90 100 50 60 290 300 150 160]. The prices can be thought of as to reflect the worth of the first economy resource to be £40, of the second to be £50 and of the third to be £60. The business class resources are worth £100 more than their corresponding economy class resources. Fare for a product using more than one resource is the sum of fares corresponding to the individual resources. Each fare in this basic fare vector is then increased by £200 every successive period to simulate price increases.

We assume that the three flights have a capacity of 200, each partitioned into business and economy, with the respective individual capacities being 40 and 160.

5.2.2 Large Network

The large test network is a hub-and-spoke network with 14 legs, 70 itineraries and 8 fare classes. The 8 fare classes comprised of a business class and an economy class, each replicated 4 times to simulate fare price increases. This resulted in a total of $70 \times 8 = 560$ products, with every succeeding group of 140 products only being available for one-fourth of the time horizon. Fares, price increases and capacities were developed in a similar fashion as for the small network. This test network is similar in size to the network used by Talluri and VanRyzin [23].

5.3 Methodology

To begin, we sample the time series model (7) to generate 30 paths of realized demand data that we use as historical demand data. For the small network we use a time horizon of $T = 20$ and for the large network a time horizon of $T = 12$.

For each of the 30 demand paths, the MSP, RLP, and DLP approaches were solved on a rolling-horizon basis. That is, for each time period $t = 1, \dots, T$, we generate multistage scenario trees using Monte-Carlo sampling, principal component sampling along only 60% of the components, moment matching, and bootstrapping [8]. We also generated random demand samples for the RLP method in a number equal to the total number of scenarios in the multistage stochastic tree, so that the comparison is fair. [9] Then we solve all four MSP problems as well as the DLP and RLP problems. The bid prices associated with each of these optimization models are then used to make booking decisions and the average total revenue made by each methodology is recorded. Finally, we also compute the perfect hindsight solution, where we assume the decision maker has perfect knowledge of future demand when making her allocation decisions. The revenue obtained by this perfect hindsight methodology is an upper bound for total revenue.

The simulations were carried out using MATLAB version 7.0.1 on a Dell Xeon PC with dual 3.2 GHz processors and 3 GB of RAM. The nonlinear optimization problem (6) was solved by calling the solver SNOPT from MATLAB. The linear programs (the deterministic equivalent for the MSP's) were solved using CPLEX 9.0, which was invoked from MATLAB using a MEX-file interface. The largest MSP problem size (corresponding to the size of the equivalent linear program) that was attempted, had 1,066,560 variables, 40,000 inequality constraints and 1,066,560 upper and lower bound constraints. Instances of this problem size were solved in about 2 minutes, while the total simulation time for *each* MSP methodology was around 12 hours. In the next section, we give the parameters of the time series model in (7) used for the simulation.

5.4 Simulation parameters and other details

For our base case, we assumed the load factor and the coefficient of variation to be along the lines of those used in the paper by Talluri and VanRyzin [23]. The mean demand a_t was then determined from the flight capacities and the load factor. The covariance matrix Σ_v was assumed to be of the form $\sigma_v^2((1-\rho) * I_n + \rho * \boldsymbol{E}_{n\times n})$, where I_n is the identity matrix of order n, E is an $n \times n$ matrix of ones, ρ is a scalar with $0 \leq \rho \leq 1$ and σ_v^2 is the variance of the shock for each product. The expression inside the parentheses ensures the positive semi-definiteness of Σ_v. The variance-covariance matrix of the error term Σ_e was a diagonal matrix with σ_e^2 chosen such that $\sigma_e^2 = \sigma_v^2$ and the total variation in demand ($\sigma_v^2 + \sigma_e^2 = \sigma^2$) being determined by the coefficient of variation. Note that the coefficient of variation was for the total demand for each product. The *basic* weight matrix was diagonal and of size ($n \times n$), with the non-zero elements corresponding to the set of products that were being offered. Table 1 lists the values for the various parameters that were used in the simulation.

The demand was scaled with respect to the time horizon (the network saw the same expected total demand for each product and the error terms were also scaled based on the length of the

[8] For bootstrapping we generated an additional set of demand paths by simulating the time series model (7) and used these paths to generate the scenarios.

[9] Note that solution times for the RLP were of the same order of magnitude or larger than those for the MSP for the same number of samples as scenarios in the stochastic tree.

horizon). So, we obtain similar results for a given network irrespective of the length of the time horizon.

In order to study the stability of the scenario generation methodologies, we ran the simulation for different values of the number of branches $B = 6, 12, 18$ (also up to $B = 1,000$ for the large network) and of the number of time stages $T = 2, 3, 4$. The number of scenarios in the MSP corresponding to each of these cases is B^{T-1}. The number of samples of the RLP was made equal to the number of scenarios in the MSP to provide a fair comparison. Thus, for $B = 12$ and $T = 3$, we took 144 samples for the RLP.

5.5 Results

We compare the total average revenue obtained by the four MSP approaches to that of the RLP and DLP approaches on the small and large networks. We also compute the perfect hindsight solution as an upper bound on the total average revenue. Below we discuss the results for the small network and the large network. For the out-of-sample stability of the different scenario generation methods, we look at the average total revenue obtained using each of the methods. For the in-sample stability, we considered the objective function value of the MSP corresponding to the first snapshot.

5.5.1 Small network

Figure 4 shows, for all methodologies, the out-of-sample results for the small network. The plot gives the average total revenue as a function of the number of branches in the scenario tree. These results are from Table 2, which also shows the variation of average total revenue with respect to the number of time stages in the MSP. For each of the MSP methods (Monte-Carlo (MC), principal component (PC), moment matching (MM), bootstrapping (BS)), Table 2 also gives three different rows containing the average total revenue and the p-values of the differences between the revenues of these policies and the revenues of the RLP (RL) and DLP (DL) methods. In particular, since we are using the same 30 demand paths for all the methodologies, the p-value corresponds to the paired t-test between estimators, i.e., whether the difference between average total revenues for an MSP approach and that for DLP or RLP is different from 0 or not. Thus, a p-value smaller than 5% means that there is a statistically significant difference between the average revenues of these policies. Finally, the last row in Table 2 gives the average revenue for the RLP approach, where for each of the six cases we use number of samples equal to the number of scenarios in the multistage tree, so that the comparison is fair.

Note that all MSP approaches outperform the DLP approach in terms of average total revenue by approximately 3%. Moreover, the p-values of the differences between the total average revenues of the MSP approaches and that of the DLP approach are statistically significant for all six cases. This confirms that ignoring demand uncertainty has an adverse effect on average revenues.

Out of the four MSP approaches considered, Monte-Carlo and principal component sampling obtain the highest average total revenue for all six cases. In addition, the small difference in the performance of these two methodologies seems to indicate that little is lost by sampling only from those principal components with highest associated variance.[10] Also, the Monte-Carlo and principal component sampling MSP approaches clearly dominate the RLP approach. In particular,

[10] For principal component sampling, we only sample from the 60% principal components with highest variance

the p-values of the differences between the average total revenue of these two MSP approaches and that of the RLP is statistically significant for all six cases considered. Moment matching follows Monte-Carlo and principal component sampling closely but the p-values of the difference with RLP are not statistically significant in all six cases. Finally, bootstrapping is the worst of all four MSP approaches and the difference with RLP is not statistically significant for any of the six cases. This seems to indicate that the *grouping* procedure employed to transform the bootstrapped demand paths into an scenario tree is not as successful at preserving the correlation structure in the data as the conditional Monte-Carlo or principal component sampling methods.

Finally, note that Monte-Carlo sampling, principal-component sampling, and moment matching are fairly stable out-of-sample. In particular, the average total revenue of these three methodologies changes by less than 2% when we increase the number of branches in the tree from $B = 6$ to $B = 18$ and the number of time stages from $T = 2$ to $T = 4$. This seems to imply that six branches and 3 time stages are enough to represent the demand uncertainty for the small network. Interestingly, the bootstrapping method is a bit less stable with respect to the number of time stages in the scenario tree. This seems to confirm our previous results that bootstrapping does not work as well as the estimation based methodologies.

Table 3 gives the in-sample stability results for the small network. For each of the four MSP approaches and for each of the six cases considered ($T = 2, 3, 4$, $B = 6, 12, 18$), we solve thirty MSP problems generated from the same time series model of demand. We then compute the average objective function value of these MSP problems as well as the standard deviation of these objective values. Note that for all cases with more than two time stages ($T > 2$), the standard deviation of these objective values is below 2%; that is, all MSP approaches are very stable in-sample for the small network. Moreover, the standard deviation for most approaches monotonically decreases with increase in the number of branches or time periods; that is, the in-sample stability in almost all cases increases with increase in the size of the MSP problems, which makes sense since the bigger the tree, the better the representation of uncertainty and the less volatility in the objective function value. This can be seen in Figure 5, with respect to principal components sampling. The plot shows that principal components sampling is quite stable in-sample, with standard deviation of the MSP objective function less than 1%, and further decreasing with increase in the number of branches.

5.5.2 Large Network

We consider eight cases with a relatively small number of branches in the scenario tree: five cases with $T = 3$ and $B = 6, 12, 18, 24, 30$ and three cases with $B = 6$ and $T = 2, 3, 4$, and then we consider four cases with a relatively large number of branches: $T = 2$ and $B = 144, 324, 576$, and $1,000$. Our results show that for all cases the MSP approaches (particularly Monte-Carlo and principal-component sampling) yield a higher average total revenue than the deterministic (DLP) approach, and the difference between these approaches is statistically significant for all cases. Also, the MSP approaches obtain a higher average total revenue than the RLP (and the difference is statistically significant) provided a sufficiently large number of branches ($B \geq 150$) is used.[11]

The out-of-sample results for the cases with small number of branches and for all methodologies are given in Table 4. Also, Figure 6 compares the out-of-sample performance of the Monte Carlo MSP approach (which is the best of all MSP approaches) with that of the RLP, DLP, and the

[11]To make a fair comparison, for the RLP we always consider a number of scenarios equal to the number of scenarios in the multistage stochastic tree.

perfect hindsight. As can be seen from the plot and the table, the results with regards to the DLP are quite similar to the ones obtained earlier for the small network. All the MSP approaches again perform significantly better than the DLP. Within the MSP approaches, Monte Carlo and principal component sampling are best and, as for the small network, the difference in the performance of these two methodologies is small. Moment matching, however, does not perform well for the large network. This is because 30 branches are not enough to approximate the first and second moments of the conditional demand distribution, and thus the moment matching methodology fails. This behavior of the moment matching algorithm has been documented in the literature before [12]. Also, the moment matching methodology is computationally expensive because of the extra time required to solve the nonlinear program.[12] However, it must be noted that certain heuristic moment matching techniques [12] may be used to address these issues.

Note finally, that even the Monte-Carlo and principal-component policies, which are the best among the MSP approaches, do not seem to achieve a very high level of stability out-of-sample. To see this, notice that their average total revenues do increase with increases in the number of branches. This seems to indicate that we might be still using too few branches given the size of the large network. Moreover, the MSP approaches do *not* perform significantly better than the RLP.[13]

To check whether better out-of-sample stability and performance relative to the RLP may be achieved with a larger number of branches in the scenario tree, we solve MSP problems with two time stages $T = 2$ and number of branches $B = 144, 324, 576, 1000$. These problems are quite large and thus take a long time to solve (particularly because for a case with 30 historical demand paths and 12 rolling-horizon periods, we need to solve 360 MSPs). For this reason, we apply only the Monte-Carlo MSP approach to solve these problems and compare it to the RLP and DLP solutions. The results are shown in Figure 7 and also documented in Table 5. Note that, for larger number of branches, Monte-Carlo MSP does indeed outperform RLP by approximately 0.5% and the difference between the two methods is indeed statistically significant.

Summarizing, in order for the MSP approaches to outperform RLP when dealing with large networks, we need to ensure that a sufficient number of branches are used and thus the corresponding representation of the demand uncertainty is accurate enough. A way to decide how many branches are needed in order to obtain such an accurate representation is to study the out-of-sample stability of the methods. From the four scenario generation methods considered, it seems like the sampling methods are the most promising and moment matching performs quite poorly for large networks.

Finally, Table 6 gives the in-sample stability results for the large network. The MSP methods achieve good in-sample stability even for small number of branches $B = 6, 12, 18$. This is somewhat surprising given the relatively poor performance of the MSP methods for small number of branches. But as Kaut and Wallace argue in their paper [15], it is possible to have in-sample stability without out-of-sample stability and vice-versa, and thus we really need to have both types of stability in order for a scenario generation method to be appropriate for a particular problem.

5.6 Sensitivity analysis

We perform a sensitivity analysis with respect to correlation, coefficient of variation, and load factor. To keep the computation time reasonable, we perform the sensitivity analysis for the small network. For the base case, we use the parameters in Section 5.4.

[12]To keep the computation time reasonable, we did not apply moment matching to the larger cases with $B = 24, 30$.

[13]Even though Monte-Carlo sampling performs better than the RLP on average for most paths, the result is not significant at the 5% significance level.

5.6.1 Correlation

Table 7 shows the sensitivity analysis with respect to demand correlation. We consider two cases, the base case where all parameters are as defined in Section 5.4 and there is inter-temporal and inter-product correlation and a case where we set $v = 0$ and thus there is no correlation in the data. We alter the rest of the parameters so that the total demand variability is the same, but the demands at different time periods and for different products are independent. For each of the two cases and for each of the methodologies, the table gives the average total revenue and the standard deviation of total revenue on the 30 historical demand paths. The methodologies considered are Monte Carlo sampling, principal component sampling, moment matching, bootstrapping, RLP, and DLP. The average total revenue in the two cases is also plotted in Figure 8. Note here that, for all the methodologies, the total revenues in the two cases for correlated and uncorrelated data are connected only for illustration purposes and *not* to represent that a continuum of simulations have been performed from the correlated to the uncorrelated case.

First of all, note that all methods perform better in the absence of correlation. This makes sense because the capacity allocation is likely to be less effective when the demand at different time periods and for different products is correlated. Also, the divergent lines in Figure 8 suggest that the relative advantage in using the MSP approaches is larger when there is demand correlation. In particular, the performance of the MSP approaches is better than that of the DLP both with and without correlation. But the relative difference between the two is slightly larger in the presence of correlation. Likewise, the performance of the MSP approaches is similar to that of the RLP in the absence of correlation. But in the presence of correlation, all MSP approaches perform slightly better than RLP. This confirms that the MSP approaches can better capture the correlation in the data than the DLP and the RLP.

5.6.2 Coefficient of Variation

We consider a base case where all parameters are as defined in Section 5.4 and two other cases where the coefficient of variation is 10% smaller and larger, respectively. The results are given in Table 8. For each of the two cases and for each of the methodologies, the table gives the average total revenue and the standard deviation of total revenue on the 30 historical demand paths. The methodologies considered are Monte-Carlo sampling, principal-component analysis, moment matching, bootstrapping, RLP, and DLP.

Note that all methods perform worse for higher coefficient of variation, as expected because of the higher uncertainty in the demand. Also, the relative difference between the Monte-Carlo MSP and the RLP and DLP approaches increases when the coefficient of variation increases. In particular, for the case where the coefficient of variation is 10% below the base case, the average total revenue of the Monte-Carlo MSP is \$5,445 higher than that of the DLP and \$1,658.75 higher than that of the RLP. But for the case where the coefficient of variation is 10% above the base case, the Monte-Carlo MSP is \$6,484.65 higher than the DLP and \$1,885.69 higher than the RLP.

5.6.3 Load Factor

We consider two cases with load factor 10% below and above the base case respectively. The results are given in Table 9. For each of the two cases and for each of the methodologies, the table gives the average total revenue and the standard deviation of total revenue on the 30 historical demand paths. The methodologies considered are Monte-Carlo sampling, principal-component analysis, moment matching, bootstrapping, RLP, and DLP.

As expected, all methods achieve higher expected revenues for higher load factors. Also, the relative advantage in using the Monte-Carlo MSP approach decreases with increases in the load factor. For the case with load factor 10% below the base case, the Monte-Carlo MSP beats the RLP and DLP by $1915.16 and $5935.68, respectively. For the case with load factor 10% above the base case, the difference is only $712.86 and $2984.34, respectively. This makes sense because with high load factors, the optimal policy should be to accept high fare demand first, and this can probably be figured out even if one does not use a very sophisticated model of demand.[14]

5.7 On bid price quality

As an illustration of the effects of using poor bid prices, Figure 9 depicts the time evolution of the average bid prices and the average capacity of one of the network resources (cabin in a given flight) for the small network. The bid prices and capacities are depicted for three different methodologies: Monte-Carlo MSP, RLP, and DLP. It is clear from the figure that the DLP underestimates the value of the bid price for this particular resource at time period 1 and, as a result, runs out of capacity faster than the RLP and MSP approaches. Then the bid price of the DLP approach increases substantially at time stage 2 and stays high ever after. The RLP and MSP approaches, on the other hand, yield much more *stable* bid prices and as a result, their overall performance is better. Summarizing, ignoring demand uncertainty may lead to poor quality bid prices and thus lower average total revenues.

6 Conclusions

Our numerical results demonstrate that the proposed MSP approaches outperform the deterministic linear programming approach (DLP), and that the difference in average revenue between the MSP and DLP approaches is statistically significant. Also, when a sufficiently high number of branches is used to represent uncertainty, the MSP approaches based on sampling also perform significantly better than the RLP. This is because the MSP approaches provide a better model of the real-world decision making process than the DLP or RLP models. Moreover, our sensitivity analysis shows that the advantage in using the MSP approaches is larger for higher demand correlation and higher coefficient of variation.

Out of all four scenario generation methods considered, Monte-Carlo and principal-component sampling are best in terms of their in- and out-of-sample stability. Also, the similar performance of these two sampling methodologies seems to indicate that not much is lost by sampling only from the principal components with largest associated variance. Bootstrapping does not perform as well as the sampling methods because it does not capture the correlation structure in the data as accurately as conditional sampling methods. Moment matching requires a prohibitively large number of branches to perform well and, in addition, is computationally very expensive. The in- and out-of-sample stability analysis have proved to be powerful tools when trying to determine the right size of the MSP scenario trees.

[14] Note that, for very low load factors there is again little difference between the performance of the MSP and other approaches because it is best to just accept every request. We do not report results for low load factors because they are not as interesting as for high load factors.

Finally, it is important to keep in mind that when applying the MSP approaches to real problems, we can solve much larger multistage stochastic programs, and thus we can use much larger numbers of branches. The reason for this is that, in reality, we only need to solve one MSP per day and then use the resulting bid prices for decision making. In our simulations, we have to solve one MSP per time stage in each historical demand path and this limits the size of the MSP problems that we could solve for our numerical experiments. In addition, decomposition methods for multistage stochastic programming [3, Chapter 7] can be used to solve very large problems.

Figure 1: Demand scenarios in Example 1.

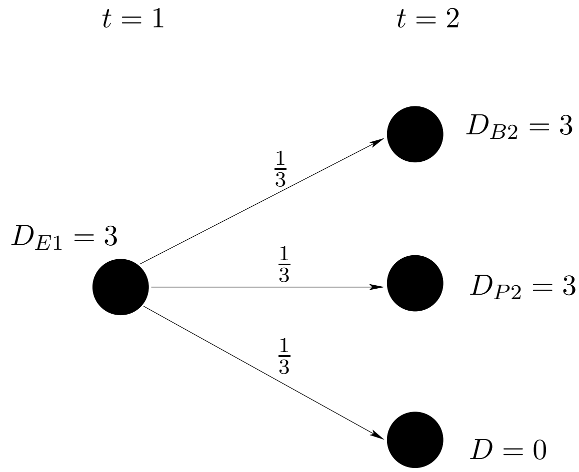


Figure 2: A multistage stochastic tree having T stages and 4 branches per stage.

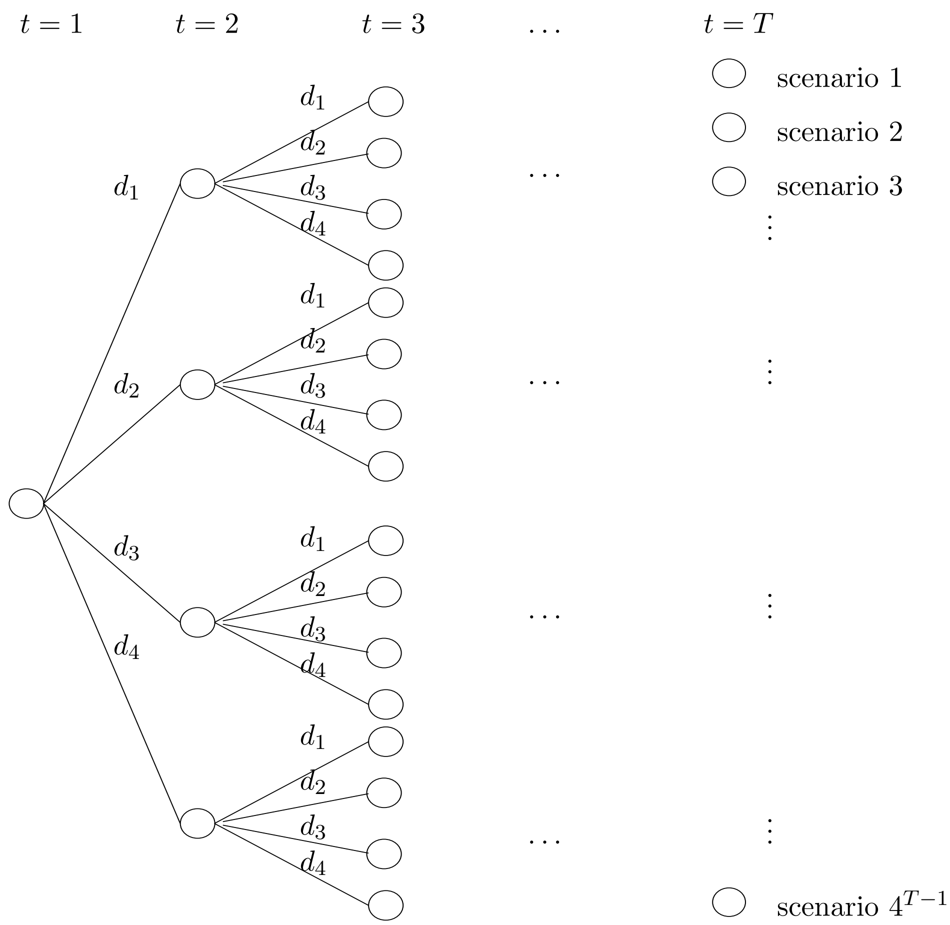


Figure 3: Topology of small network used in the simulation.

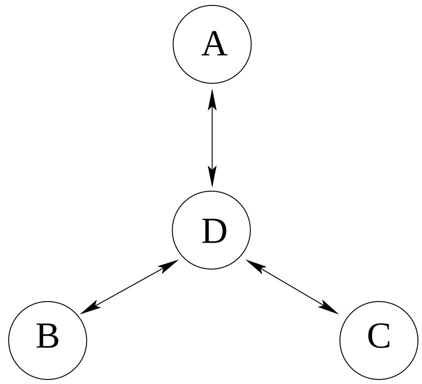


Figure 4: Small Network: Out-of-sample results with up to 18 branches.

This figure depicts the average total revenue of each of the methodologies as a function of the number of branches used in the scenario tree. The methodologies considered are Monte-Carlo sampling (MC), principal-component analysis (PC), moment matching (MM), bootstrapping (BS), RLP (RL), and DLP (DL). The figure also shows the performance of these methodologies in comparison to the perfect hindsight (PH) upper bound. The horizontal axis gives the number of branches used in the scenario tree and the vertical axis is the average total revenue obtained when using each of the methodologies.

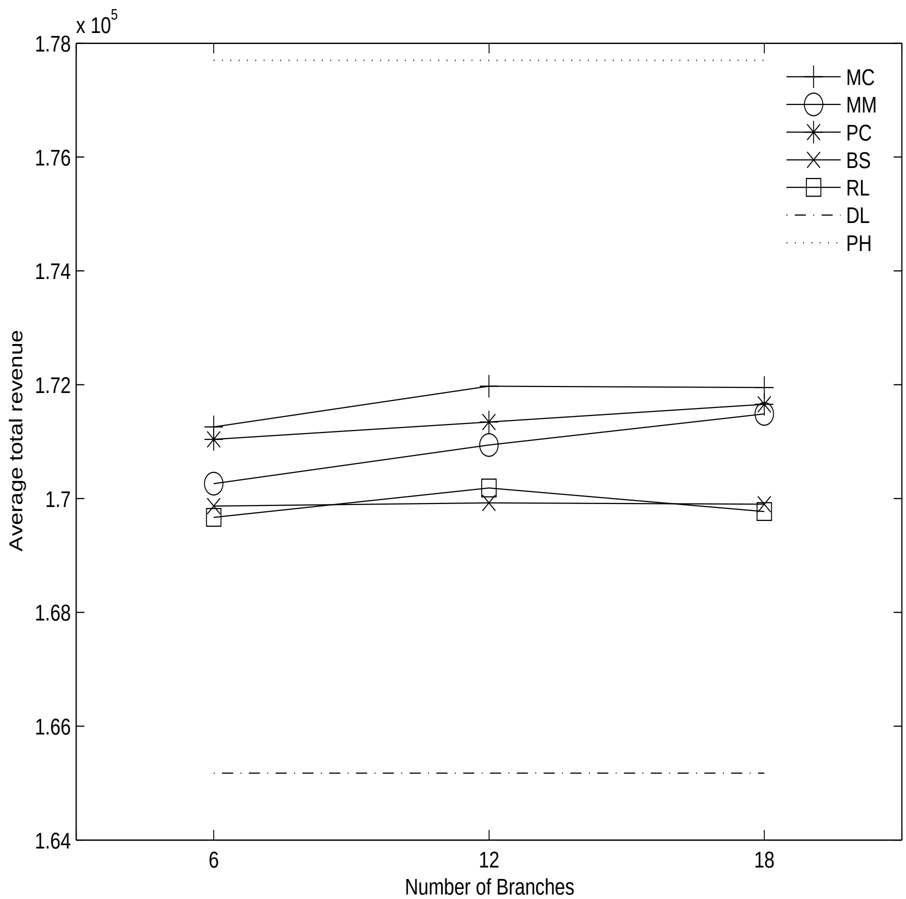


Figure 5: Small Network: In-sample results for principal components with up to 18 branches.

This figure gives the in-sample stability results of the pricipal components with respect to the number of branches. The left and right y-axes correspond to the average and the standard deviation of MSP objective function values respectively, which have been plotted against the number of branches.

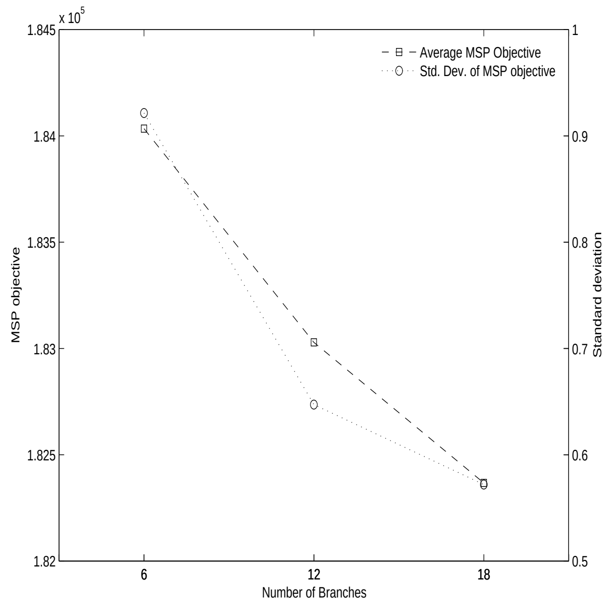


Figure 6: Large Network: Out-of-sample results for Monte-Carlo sampling with up to 30 branches.

This figure gives the out-of-sample stability results of Monte-Carlo sampling (MC), randomized linear program (RL), and the deterministic linear program (DL) along with the perfect hindsight (PH) upper bound. The horizontal axis gives the number of branches used in the scenario tree and the vertical axis is the average total revenue obtained when using each of the methodologies.

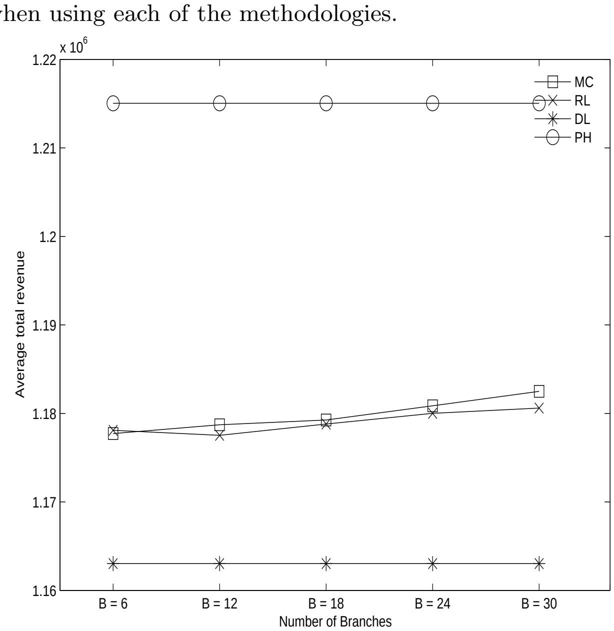


Figure 7: Large Network: Out-of-sample results with 2 time stages and up to 1000 branches.

This figure gives the out-of-sample results for the large network for a two-stage stochastic program with up to 1000 branches. Since problem sizes are large, we only study the Monte-Carlo MSP approach (MC) and compare it to the RLP (RL) and DLP (DL) solutions. We also present the perfect hindsight (PH) upper bound.

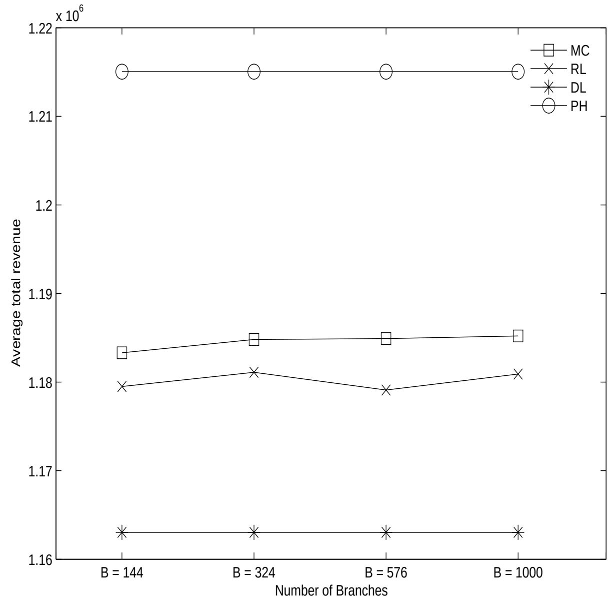


Figure 8: Small Network: Sensitivity to correlation.

This figure shows the performance of the various methodologies with respect to correlated and uncorrelated demand. The methodologies considered are Monte-Carlo sampling (MC), principal-component analysis (PC), moment matching (MM), bootstrapping (BS), RLP (RL), and DLP (DL). The horizontal axis considers two cases with correlated and uncorrelated demand, while the vertical axis shows the average total revenue.

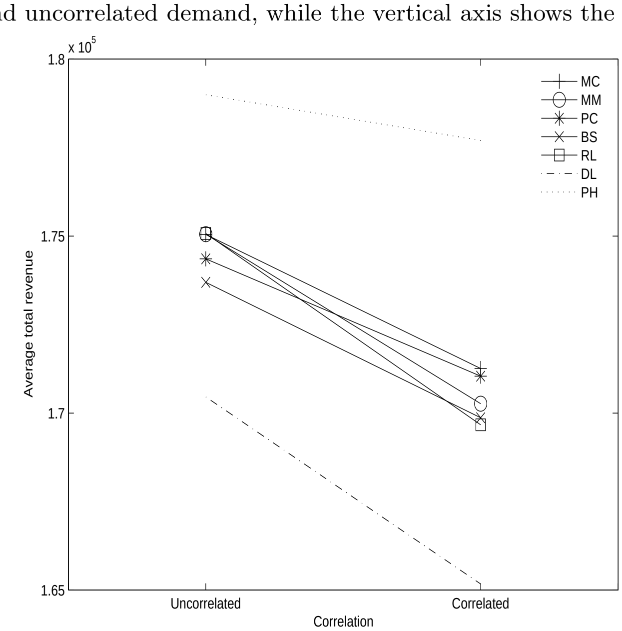
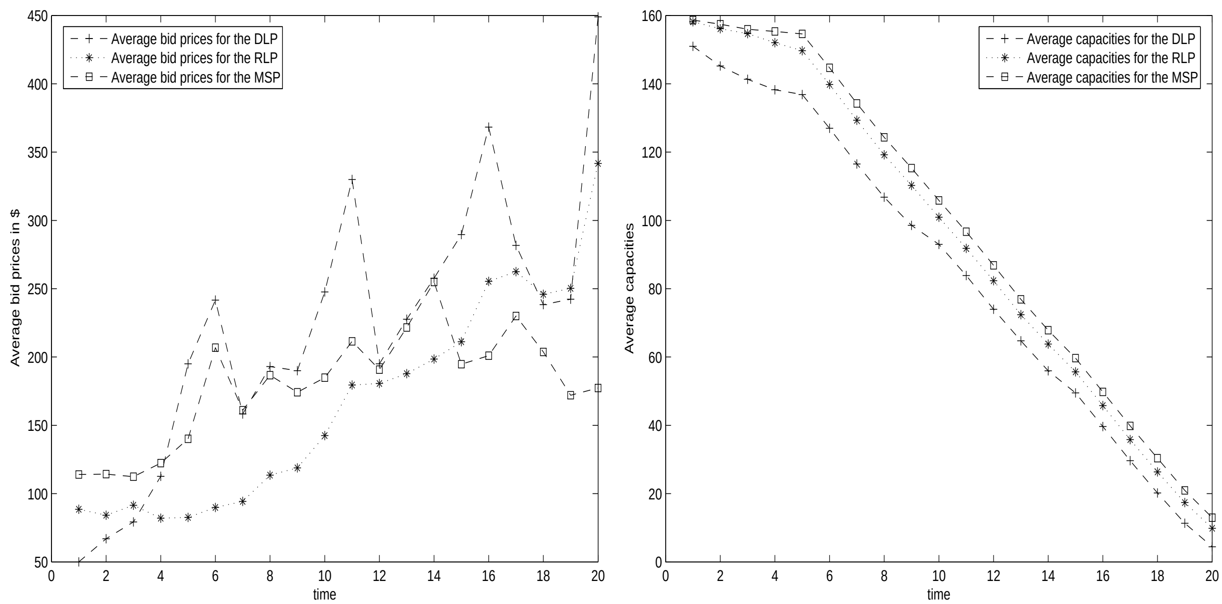


Figure 9: Average bid prices and capacities for MSP, DLP and RLP for an economy product

This figure depicts the evolution of the bid prices and capacity of one of the network resources (cabin in a given flight) for the small network. The bid prices and capacities are depicted for three different methodologies: Monte-Carlo MSP, RLP, and DLP. In both Panels (a) and (b), time is on the horizontal axis, while the vertical axes are average bid prices and average capacities respectively.



(a) Average bid prices for an economy product (b) Average capacities for the economy product

Table 1: Details of parameter values used in the simulation

Parameter	Small Network	Large Network
time horizon, S (periods)	20	12
load-factor $= \dfrac{\text{mean total demand}}{\text{mean total capacity}}$	0.77	0.77
# of products offered, n	32	560
# of resources, m	6	28
# of branches in the MSP, B	6	6
# of time periods in the MSP, T	3	3
# of demand paths, *Paths*	30	30
coefficient of variation ($= \frac{\sigma}{\mu}$)	0.4	0.4
for the var-cov matrix of the random shock, ρ_e	0.5	0.5
for the var-cov matrix of the random shock, ρ_b	0.3	0.3

Table 2: Small network: Out-of-sample stability results.

Table 2 gives the out-of-sample results for all methodologies. The first and second rows give the average total revenue of the DLP and perfect hindsight methods, respectively. For each of the MSP approaches we consider six different cases depending on the number of time stages T and the number of branches B in the multistage trees. The results for each case are given in a different column. The first three columns correspond to cases with $T = 3$ and $B = 6, 12, 18$, respectively. The last three columns correspond to cases with $B = 6$ but with $T = 2, 3, 4$, respectively. For each of the MSP methods (Monte-Carlo, principal component, moment matching, bootstrapping), we give three different rows containing the average total revenue and the p-values of the differences between the revenues of these policies and the revenues of the RLP and DLP methods.

DLP	Avg.	165172.48					
Perfect Hindsight	Avg.	177700.07					
		T = 3 time stages			B = 6 branches		
		B = 6	B = 12	B = 18	T = 2	T = 3	T = 4
Monte Carlo	Avg.	171258.96	171974.06	171951.03	170522.65	171258.96	172149.35
	pVal.-DLP	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
	pVal.-RLP	0.44%	0.31%	0.04%	0.00%	0.44%	0.33%
Principal component	Avg.	171040.28	171345.09	171654.88	169562.14	171040.28	171730.13
	pVal.-DLP	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
	pVal.-RLP	0.37%	0.70%	0.00%	0.20%	0.37%	0.15%
Moment matching	Avg.	170263.00	170940.76	171486.23	169593.35	170263.00	171060.56
	pVal.-DLP	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
	pVal.-RLP	16.91%	8.43%	0.05%	0.16%	16.91%	4.68%
Bootstrapping	Avg.	169868.08	169924.94	169900.30	167659.52	169868.08	170173.31
	pVal.-DLP	0.00%	0.00%	0.00%	0.01%	0.00%	0.00%
	pVal.-RLP	64.62%	48.66%	67.38%	65.06%	64.62%	32.68%
RLP	Avg.	169667.53	170186.60	169772.43	167948.71	169667.53	169827.29

Table 3: Small network: In-sample stability results.

The following table gives the in-sample stability results for the small network. For each of the four MSP approaches and for each of the six cases considered ($T = 2, 3, 4$, $B = 6, 12, 18$), we solve thirty MSP problems generated from the same time series model of demand. We then compute the average objective function value of these MSP problems as well as the standard deviation of these objective values.

		T = 3 time stages			B = 6 branches		
		B = 6	B = 12	B = 18	T = 2	T = 3	T = 4
Monte Carlo	Avg.	181141.23	180645.74	179214.02	178392.63	181141.23	185260.97
	Std.	1.22%	0.56%	0.84%	3.78%	1.22%	1.15%
Moment Matching	Avg.	177256.28	177991.37	178669.44	175141.99	177256.28	176506.15
	Std.	0.32%	0.16%	0.14%	0.16%	0.32%	0.17%
PCA	Avg.	184034.38	183028.58	182367.69	181670.25	184034.38	190268.89
	Std.	0.92%	0.65%	0.57%	2.42%	0.92%	0.65%
Bootstrapping	Avg.	170528.53	171624.79	171739.01	172772.92	170528.53	171748.64
	Std.	1.57%	0.60%	0.56%	5.27%	1.57%	0.64%

Table 4: Large Network: Out-of-sample stability results with up to 30 branches.

For the large network, the out-of-sample results are presented in Table 4. The first and second rows give the average total revenue of the DLP and perfect hindsight methods, respectively. For each of the MSP approaches we consider eight different cases depending on the number of time stages T and the number of branches B in the multistage trees. The results for each case are given in a different column. The first five columns correspond to cases with $T = 3$ and $B = 6, 12, 18, 24, 30$, respectively. The last three columns correspond to cases with $B = 6$ but with $T = 2, 3, 4$, respectively. For each of the MSP methods (Monte-Carlo, principal component, moment matching, bootstrapping), we give three different rows containing the average total revenue and the p-values of the differences between the revenues of these policies and the revenues of the RLP and DLP methods.

		T = 3 time stages					B = 6 branches		
		B = 6	B = 12	B = 18	B = 24	B = 30	T = 2	T = 3	T = 4
Monte Carlo	Avg.	1177722.53	1178711.74	1179248.11	1180859.00	1182496.33	1173951.55	1177722.53	1177630.85
	pVal.-DLP	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
	pVal.-RLP	82.83%	50.37%	83.89%	68.62%	27.76%	69.85%	82.83%	27.36%
PCA	Avg.	1176381.50	1179916.41	1180959.82	1181867.05	1182571.59	1175833.28	1176381.50	1179455.05
	pVal.-DLP	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
	pVal.-RLP	42.36%	16.07%	28.48%	22.12%	28.15%	77.54%	42.36%	77.47%
Moment Matching	Avg.	1169917.80	1171660.73	1173508.58			1169965.26	1169917.80	1169871.55
	pVal.-DLP	0.19%	0.01%	0.00%			0.72%	0.19%	0.25%
	pVal.-RLP	0.26%	0.06%	0.59%			25.07%	0.26%	0.03%
Bootstrapping	Avg.	1176210.46	1179800.64	1179614.13	1181122.13	1181485.29	1168707.81	1176210.46	1173893.26
	pVal.-DLP	0.00%	0.00%	0.00%	0.00%	0.00%	3.46%	0.00%	0.01%
	pVal.-RLP	33.89%	21.29%	66.03%	51.87%	60.84%	2.77%	33.89%	1.05%
RLP	Avg.	1178091.42	1177520.80	1178799.76	1180013.10	1180605.49	1175105.48	1178091.42	1179974.56
DLP	Avg.	1163036.19	1163036.19	1163036.19	1163036.19	1163036.19	1163036.19	1163036.19	1163036.19
Perfect Hindsight	Avg.	1215045.07	1215045.07	1215045.07	1215045.07	1215045.07	1215045.07	1215045.07	1215045.07
	Std.	3.04%	3.04%	3.04%	3.04%	3.04%	3.04%	3.04%	3.04%

Table 5: Large Network: Out-of-sample results with 2 time stages and up to 1000 branches.

This table presents the out-of-sample results for the large network for a two-stage stochastic program with up to 1000 branches. Since problem sizes are large, we only study the Monte-Carlo MSP approach and compare it to the RLP and DLP solutions. Here, we see that for larger number of branches, Monte-Carlo MSP does outperform RLP by approximately 0.5% and the difference between the two methods is indeed statistically significant.

DLP	Avg.	1163036.19			
Perfect Hindsight	Avg.	1215045.07			
		T = 2 stages			
		B = 144	B = 324	B = 576	B = 1000
Monte Carlo	Avg.	1183300	1184800	1184900	1185200
	pVal.-DLP	0.00%	0.00%	0.00%	0.00%
	pVal.-RLP	5.10%	8.62%	0.20%	2.34%
RLP	Avg.	1179500	1181100	1179100	1180900

Table 6: Large Network: In-sample stability results.

The following table gives the in-sample stability results for the large network. As can be seen, the MSP methods achieve good in-sample stability even for small number of branches $B = 6, 12, 18$.

		T = 3 time stages					B = 6 branches		
		B = 6	B = 12	B = 18	B = 24	B = 30	T = 2	T = 3	T = 4
Monte Carlo	Avg.	1232393.56	1230860.30	1229132.75	1228781.60	1228428.75	1214646.67	1232393.56	1248206.05
	Std.	0.59%	0.30%	0.20%	0.19%	0.18%	1.30%	0.59%	0.30%
PCA	Avg.	1245836.02	1242158.97	1240778.37	1239439.98	1239069.11	1229999.90	1245836.02	1260751.79
	Std.	0.23%	0.19%	0.14%	0.13%	0.10%	0.51%	0.23%	0.21%
Bootstrapping	Avg.	1209132.86	1207873.04	1209707.59	1209684.40	1209469.46	1206697.13	1209132.86	1208845.73
	Std.	0.57%	0.32%	0.22%	0.14%	0.13%	1.79%	0.57%	0.20%

Table 7: Small Network: Sensitivity with respect to correlation.

The following table gives the sensitivity with respect to demand correlation. For each of the two cases, with and without correlation, and for each of the methodologies, the table gives the average total revenue and the standard deviation of total revenue on the 30 historical demand paths.

		MC	PC	MM	BS	RL	DL	PH
$v = 0,\ \rho = 0$	Avg.	175046.11	174356.34	175051.27	173693.36	175073.24	170454.81	178990.26
	Std.	3.76%	3.57%	3.28%	2.40%	3.18%	2.18%	3.53%
$v \neq 0,\ \rho \neq 0$	Avg.	171258.96	171040.28	170263.00	169868.08	169667.53	165172.48	177700.07
	Std.	7.83%	7.44%	7.37%	6.84%	7.14%	6.10%	7.31%

Table 8: Small Network: Sensitivity with respect to coefficient of variation.

This table presents results regarding performance with respect to the coefficient of variation. We consider a base case where all parameters are as defined in Section 5.4 and two other cases where the coefficient of variation is 10% smaller and larger, respectively.

		MC	PC	MM	BS	RL	DL	PH
CoV 10 % below base case	Avg.	172119.50	171696.36	171332.52	170040.50	170460.75	166674.62	177864.57
	Std.	7.28%	6.83%	6.72%	5.93%	6.21%	5.49%	6.67%
CoV Base Case	Avg.	171258.96	171040.28	170263.00	169868.08	169667.53	165172.48	177700.07
	Std.	7.83%	7.44%	7.37%	6.84%	7.14%	6.10%	7.31%
CoV 10 % above base case	Avg.	170971.75	170208.74	169572.60	169008.78	169086.06	164487.10	177562.52
	Std.	8.82%	8.09%	8.11%	7.28%	7.50%	6.32%	7.93%

Table 9: Small Network: Out-of-sample sensitivity with respect to load factor.

This table shows sensitivity with respect to load factor. We consider two cases with load factor 10% below and above the base case respectively. For each of the two cases and for each of the methodologies, the table gives the average total revenue and the standard deviation of total revenue on the 30 historical demand paths.

		MC	PC	MM	BS	RL	DL	PH
Load Fact. 10 % below base case	Avg.	159961.31	158696.24	159027.02	157805.69	158045.70	154025.63	164764.52
	Std.	8.09%	7.58%	7.45%	6.93%	7.11%	6.07%	7.76%
Load Fact. Base Case	Avg.	171258.96	171040.28	170263.00	169868.08	169667.53	165172.48	177700.07
	Std.	7.83%	7.44%	7.37%	6.84%	7.14%	6.10%	7.31%
Load Fact. 10 % above base case	Avg.	181911.38	181786.31	182086.72	180854.14	181198.52	178927.04	190040.82
	Std.	7.11%	7.04%	7.06%	6.21%	6.80%	6.03%	6.58%

References

[1] D. Adelman. On approximate dynamic programming and bid-price controls in revenue management. Working paper, The University of Chicago, Graduate School of Business. Available at http://gsbwww.uchicago.edu/fac/daniel.adelman/research/.

[2] D. Bertsimas and I. Popescu. Revenue management in a dynamic network environment. *Transportation Science*, 37(3):257–277, Aug 2003.

[3] J.R. Birge and F. Louveaux. *Introduction to Stochastic Programming.* Springer-Verlag, New York, 1997.

[4] M. Casey and S. Sen. The scenario generation algorithm for multi-stage stochastic linear programs. Available at http://www.math.ups.edu/∼mcasey/, 2002.

[5] L. Chen and T. Homem de Mello. Multistage stochastic programming models for airline revenue management. Working Paper No. 04-012, Dept. of Industrial Engineering and Management Sciences, Northwestern University, Dec 2004.

[6] W. L. Cooper. Asymptotic behaviour of an allocation policy for revenue management. *Operations Research*, 50(4):720–727, Jul-Aug 2002.

[7] S. V. de Boer, R. Freling, and N. Piersma. Mathematical programming for network revenue management revisited. *European Journal of Operational Research*, 137(1):72–92, Feb 2002.

[8] L. Devroye. *Non-Uniform Random Variate Generation.* Web edition available at http://cg.scs.carleton.ca/∼luc/rnbookindex.html, 2003.

[9] J. Dupačová, G. Consigli, and S. W. Wallace. Scenarios for multistage stochastic programs. *Annals of Operations Research*, 100(1):25–53, 2000.

[10] R. Halldin. *Scenario Trees for Inflow Modelling in Stochastic Optimization for Energy Planning.* PhD thesis, Lund University, Sweden, 2002.

[11] J. L. Higle and S. Sen. A stochastic programming model for network resource utilization in the presence of multi-class demand uncertainty. To appear in *Applications of Stochastic Programming*, SIAM Series on Optimization, W.T. Ziemba and S.W. Wallace (eds.).

[12] K. Høyland, M. Kaut, and S. W. Wallace. A heuristic for moment-matching scenario generation. *Computational Optimization and Applications*, 24(2-3):169–185, Feb-Mar 2003.

[13] K. Høyland and S. W. Wallace. Generating scenario trees for multistage decision problems. *Management Science*, 47(2):295–307, Feb 2001.

[14] R. A. Johnson and D. W. Wichern. *Applied Multivariate Statistical Analysis.* Pearson Prentice Hall, Pearson Education Inc., Upper Saddle River, New Jersey, USA, 2007.

[15] M. Kaut and S. W. Wallace. Evaluation of scenario-generation methods for stochastic programming. World Wide Web, Stochastic Programming E-Print Series, http://www.speps.info, July 2003.

[16] M. Loretan. Generating market risk scenarios using principal components analysis: methodological and practical considerations. In *The measurement of aggregate market risk, CGFS Publications No. 7*, pages 23–60. Bank for International Settlements, November 1997. Available at http://www.bis.org/publ/ecsc07c.pdf.

[17] A. Möller, W. Römisch, and K. Weber. A new approach to o & d revenue management based on scenario trees. *Journal of Revenue and Pricing Management*, 3(3):265–276, 2004.

[18] T. Pennanen. Epi-convergent discretizations of multistage stochastic programs via integration quadratures. http://hkkk.fi/ pennanen/, March 2005.

[19] G. Ch. Pflug. Scenario tree generation for multiperiod financial optimization by optimal discretization. *Mathematical Programming*, 89(2):251–271, 2001.

[20] A. Ruszczyński and A. Shapiro, editors. *Stochastic Programming*, volume 10 of *Handbooks in Operations Reseach and Management Science*. Elsevier, The Netherlands, 2003.

[21] C. Stefanescu. Forecasting models for correlated demand. London Business School Working Paper, Dec 2005.

[22] K. T. Talluri and G. J. van Ryzin. An analysis of bid-price controls for network revenue management. *Management Science*, 44(11):1577–1593, Nov 1998.

[23] K. T. Talluri and G. J. van Ryzin. A randomized linear programming method for comptuting network bid prices. *Transportation Science*, 33(2):207–216, May 1999.

[24] K. T. Talluri and G. J. van Ryzin. *The Theory and Practice of Revenue Management*. Kluwer Academic Publishers, Norwell, Massachusetts, USA, 2004.

[25] N. Topaloglou, H. Vladimirou, and S. A. Zenios. Cvar models with selective hedging for international asset allocation. *Journal of Banking and Finance*, 26(7):1535–1561, 2002.

[26] E. L. Williamson. *Airline Network Seat Control*. PhD thesis, MIT, Cambridge, Massachusetts, 1992.