CSE 548: (*Design and*) Analysis of Algorithms

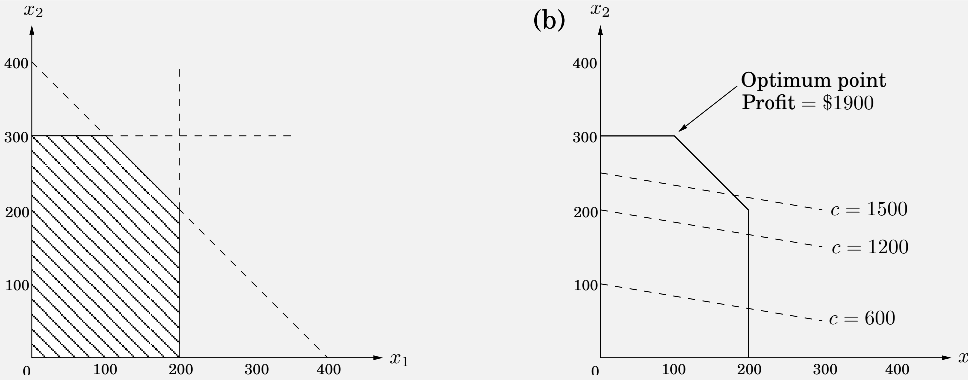
Linear Programming

R. Sekar

Overview

- A technique for modeling a diverse range of optimization problems
  - LP is more of a modeling technique: You are not being asked to develop new “LP algorithms,” but to model existing problems using LP.
  - Existing solvers can solve these problems
- We cover the intuition behind the solver, but not in great depth.

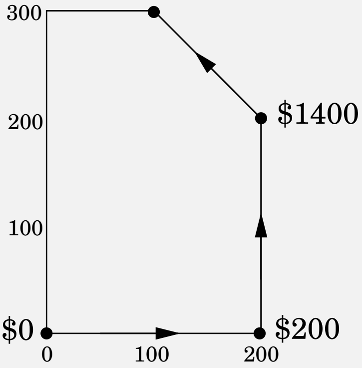
Example 1: Profit Maximization



- Product P_1 generates \$1/unit, P_2 generates \$6/unit — Max $x_1 + 6x_2$
- Max 200 units of P_1 and 300 of P_2 can be sold — $x_1 \leq 200, x_2 \leq 300$
- Company can produce a total of 400 units — $x_1 + x_2 \leq 400$
- (Cannot produce negative number of units!) — $x_1, x_2 \geq 0$

Note: It is easy to see that a maximum should be at a vertex

Simplex Method

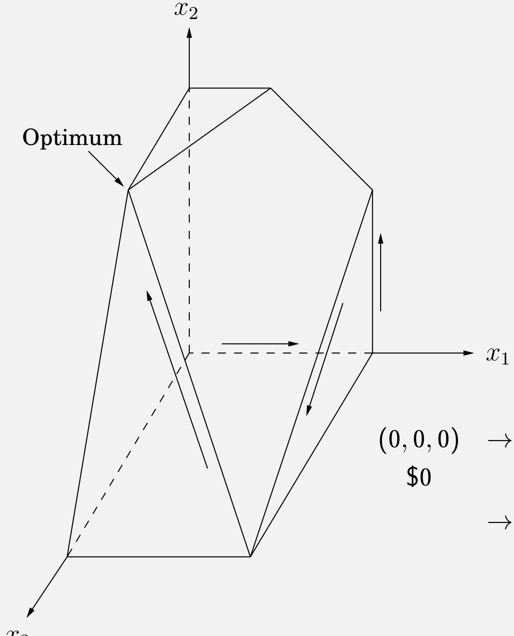


- Applicable to *convex problems,* i.e., conjunctions, and *linear constraints,* i.e., no squaring/multiplication of variables.
- Feasible regions are *convex polygons*

Simplex

- Start at the origin
- Switch to neighboring vertex if objective function $f(\bar{x})$ is higher
- Repeat until you reach a local maxima
  - which *will* be a global maxima
  - Consider the line $f(\bar{x}) = c$ passing through the vertex. Rest of the polygon must be below this line.

Example 2: On to more products ...



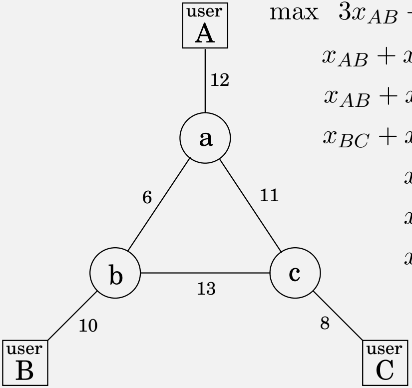
$$
\begin{aligned}
\max \quad x_1 + 6x_2 + 13x_3 & \\
x_1 &\le 200 \\
x_2 &\le 300 \\
x_1 + x_2 + x_3 &\le 400 \\
x_2 + 3x_3 &\le 600 \\
x_1, x_2, x_3 &\ge 0
\end{aligned}
$$

Search follows these steps:

$(0,0,0)$ → $(200,0,0)$ → $(200,200,200)$
\$0 — \$200 — \$1400

→ $(200,0,200)$ → $(0,300,100)$
\$2800 — \$3100

Example 3: Communication Network



$$\begin{aligned}
\max \quad & 3x_{AB} + 3x'_{AB} + 2x_{BC} + 2x'_{BC} + 4x_{AC} + 4x'_{AC} & \\
& x_{AB} + x'_{AB} + x_{BC} + x'_{BC} \le 10 & \text{[edge } (b, B)\text{]} \\
& x_{AB} + x'_{AB} + x_{AC} + x'_{AC} \le 12 & \text{[edge } (a, A)\text{]} \\
& x_{BC} + x'_{BC} + x_{AC} + x'_{AC} \le 8 & \text{[edge } (c, C)\text{]} \\
& x_{AB} + x'_{BC} + x'_{AC} \le 6 & \text{[edge } (a, b)\text{]} \\
& x'_{AB} + x_{BC} + x'_{AC} \le 13 & \text{[edge } (b, c)\text{]} \\
& x'_{AB} + x'_{BC} + x_{AC} \le 11 & \text{[edge } (a, c)\text{]} \\
& x_{AB} + x'_{AB} \ge 2 & \\
& x_{BC} + x'_{BC} \ge 2 & \\
& x_{AC} + x'_{AC} \ge 2 & \\
& x_{AB}, x'_{AB}, x_{BC}, x'_{BC}, x_{AC}, x'_{AC} \ge 0 &
\end{aligned}$$

- A–B, B–C and A–C traffic pay \$3, \$2, \$4/unit
- Minimum 2 units per connection
- x and x' refer to traffic on short path and long path, resp.
- Sol: $x_{AB} = 0$, $x'_{AB} = 7$, $x_{BC} = x'_{BC} = 1.5$, $x_{AC} = 0.5$, $x'_{AC} = 4.5$

Matrix-vector notation

A linear function like $x_1 + 6x_2$ can be written as the dot product of two vectors

$$\mathbf{c} = \begin{pmatrix} 1 \\ 6 \end{pmatrix} \text{ and } \mathbf{x} = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix},$$

denoted $\mathbf{c} \cdot \mathbf{x}$ or $\mathbf{c}^T\mathbf{x}$. Similarly, linear constraints can be compiled into matrix-vector form:

$$\begin{array}{rcl} x_1 & \leq & 200 \\ x_2 & \leq & 300 \\ x_1 + x_2 & \leq & 400 \end{array} \implies \underbrace{\begin{pmatrix} 1 & 0 \\ 0 & 1 \\ 1 & 1 \end{pmatrix}}_{\mathbf{A}} \underbrace{\begin{pmatrix} x_1 \\ x_2 \end{pmatrix}}_{\mathbf{x}} \underbrace{\leq}_{\leq} \underbrace{\begin{pmatrix} 200 \\ 300 \\ 400 \end{pmatrix}}_{\mathbf{b}}.$$

Here each row of matrix $\mathbf{A}$ corresponds to one constraint: its dot product with $\mathbf{x}$ is at most the value in the corresponding row of $\mathbf{b}$. In other words, if the rows of $\mathbf{A}$ are the vectors $\mathbf{a}_1, \ldots, \mathbf{a}_m$, then the statement $\mathbf{Ax} \leq \mathbf{b}$ is equivalent to

$$\mathbf{a}_i \cdot \mathbf{x} \leq b_i \text{ for all } i = 1, \ldots, m.$$

With these notational conveniences, a generic LP can be expressed simply as

$$\begin{gathered} \max\ \mathbf{c}^T\mathbf{x} \\ \mathbf{Ax} \leq \mathbf{b} \\ \mathbf{x} \geq 0. \end{gathered}$$

Optimality of Solution

$$
\begin{array}{rcll}
\text{Max} \quad x_1 & + & 6x_2 & (1) \\
x_1 & \leq & 200 & (2) \\
x_2 & \leq & 300 & (3) \\
x_1 + x_2 & \leq & 400 & (4) \\
x_1, x_2 & \geq & 0 & (5)
\end{array}
$$

- Multiply (3) by 5 and (4) by 1 and add:

$$
\begin{aligned}
5 \cdot x_2 + 1 \cdot (x_1 + x_2) &\leq 5 \cdot 300 + 1 \cdot 400 \\
x_1 + 6 \cdot x_2 &\leq 1900
\end{aligned}
$$

- Magically, we have a proof that the maximum possible value for profit is \$1900
- This is a *certificate of optimality* for the solution found by LP!

Constructing Dual Problem

- Introduce a multiplier y_i for each equation:

Multiplier	Inequality		
y_1	x_1	$\leq$	200
y_2	x_2	$\leq$	300
y_3	$x_1 + x_2$	$\leq$	400

- After muliplying and adding, we get

$$(y_1 + y_3)x_1 + (y_2 + y_3)x_2 \leq 200y_1 + 300y_2 + 400y_3$$

- To get optimality proof, we need $y_1 + y_3 \geq 1$, $y_2 + y_3 \geq 6$. In other words, we have the dual problem:

$$\begin{array}{rcl} \text{Min} & & 200y_1 + 300y_2 + 400y_3 \\ y_1 + y_3 & \geq & 1 \\ y_2 + y_3 & \geq & 6 \\ y_1, y_2, y_3 & \geq & 0 \end{array}$$

Duality

Primal LP:

$$\max\ \mathbf{c}^T\mathbf{x}$$
$$\mathbf{A}\mathbf{x} \leq \mathbf{b}$$
$$\mathbf{x} \geq 0$$

Dual LP:

$$\min\ \mathbf{y}^T\mathbf{b}$$
$$\mathbf{y}^T\mathbf{A} \geq \mathbf{c}^T$$
$$\mathbf{y} \geq 0$$

Theorem (Duality)

If a linear program has a bounded optimum, then so does it dual, and the two optima coincide.

Simplex Algorithm

"Pebble falling down:"

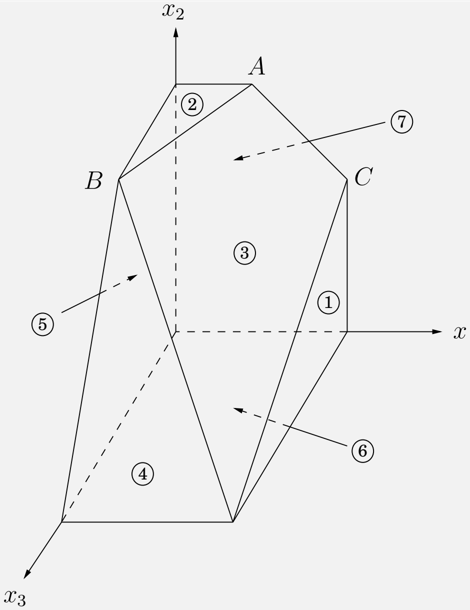
- If you rotate the axes so that the normal to the hyperplane represented by the objective function faces down,
- then simplex operation resembles that of a pebble starting from one vertex, sliding down to the next vertex down and the next vertex down,
- until it reaches the minimum.

For simplicity, we consider only those cases where there is a unique solution, i.e., ignore degenerate cases.

Simplex Algorithm

- What is the space of feasible solutions?
  - A convex polyhedron in n-dimensions ($n =$ number of variables)
- What is a vertex?
  - A point of intersection of n inequalities ("hyperplanes")
- What is a neighboring vertex?
  - Two vertices are neighbors if they share $n-1$ inequalities.
  - Vertex found by solving n simultaneous equations
- How many times can it fall?
  - There are m inequalities and n variables, so $\binom{m+n}{n}$ vertices can be there.
  - This is an exponential number, but simplex works exceptionally well in practice.

Simplex Algorithm



History and Main LP Algorithms

Fourier (1800s) Informal/implicit use

Kantorovich (1930) Applications to problems in Economics

Koopmans (1940) Application to shipping problems

Dantzig (1947) Simplex method.

Nobel Prize (1975) Kantorovich and Koopmans, not Dantzig

Khachiyan (1979) Ellipsoid algorithm, polynomial time but not competitive in practice.

Karmarkar (1984) Interior point method, polynomial time, good practical performance.