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 200 units of $P_1$ and 300 of $P_2$ can be sold
- Company can produce a total of 400 units
- (Cannot produce negative number of units!)

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 ...

Let \( x_1, x_2, x_3 \) denote the number of boxes of each chocolate produced daily, with

\[
\begin{align*}
\text{max} & \quad x_1 + 6x_2 + 13x_3 \\
x_1 & \leq 200 \\
x_2 & \leq 300 \\
x_1 + x_2 + x_3 & \leq 400 \\
x_2 + 3x_3 & \leq 600 \\
x_1, x_2, x_3 & \geq 0
\end{align*}
\]

Search follows these steps:

\[
(0, 0, 0) \rightarrow (200, 0, 0) \rightarrow (200, 200, 200) \rightarrow (0, 300, 100)
\]

$0 \rightarrow \$200 \rightarrow \$1400 \rightarrow \$3100$
**Example 3: Communication Network**

A communications network between three users A, B, and C. Bandwidths shown in Figure 7.3.

The connection between A and B, and $x_{AB} \text{ the long-path bandwidth for this same connection.}$

We demand that no edge’s bandwidth is exceeded and that each connection gets a bandwidth of at least 2 units.

Maximizing revenue:

$$
\text{max } 3x_{AB} + 3x'_{AB} + 2x_{BC} + 2x'_{BC} + 4x_{AC} + 4x'_{AC}
$$

Subject to:

1. $x_{AB} + x'_{AB} + x_{BC} + x'_{BC} \leq 10 \quad [\text{edge (b, B)}]$  
2. $x_{AB} + x'_{AB} + x_{AC} + x'_{AC} \leq 12 \quad [\text{edge (a, A)}]$  
3. $x_{BC} + x'_{BC} + x_{AC} + x'_{AC} \leq 8 \quad [\text{edge (c, C)}]$  
4. $x_{AB} + x'_{BC} + x_{AC} \leq 6 \quad [\text{edge (a, b)}]$  
5. $x'_{AB} + x_{BC} + x'_{AC} \leq 13 \quad [\text{edge (b, c)}]$  
6. $x'_{AB} + x_{BC} + x_{AC} \leq 11 \quad [\text{edge (a, c)}]$  

Subject to:

- $x_{AB}, x'_{AB}, x_{BC}, x'_{BC}, x_{AC}, x'_{AC} \geq 0$

- $A-B, B-C$ and $A-C$ traffic pay $3, 2, 4/\text{unit}$

- 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

$$c = \begin{pmatrix} 1 \\ 6 \end{pmatrix} \text{ and } x = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix},$$
denoted $c \cdot x$ or $c^T x$. Similarly, linear constraints can be compiled into matrix-vector form:

$$\begin{align*}
x_1 & \leq 200 \\
x_2 & \leq 300 \\
x_1 + x_2 & \leq 400
\end{align*} \implies \begin{pmatrix} 1 & 0 \\ 0 & 1 \\ 1 & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} \leq \begin{pmatrix} 200 \\ 300 \\ 400 \end{pmatrix}.$$

Here each row of matrix $A$ corresponds to one constraint: its dot product with $x$ is at most the value in the corresponding row of $b$. In other words, if the rows of $A$ are the vectors $a_1, \ldots, a_m$, then the statement $Ax \leq b$ is equivalent to

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

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

$$\begin{align*}
\max & \quad c^T x \\
Ax & \leq b \\
x \geq 0.
\end{align*}$$
Optimality of Solution

Max \( x_1 + 6x_2 \) \hspace{1cm} (1) \hspace{1cm} 5 \cdot x_2 + 1 \cdot (x_1 + x_2) \leq 5 \cdot 300 + 1 \cdot 400

\( x_1 \leq 200 \) \hspace{1cm} (2) \hspace{1cm} x_1 + 6 \cdot x_2 \leq 1900

\( x_2 \leq 300 \) \hspace{1cm} (3) \hspace{1cm} \text{Magically, we have a proof that the maximum possible value for profit is $1900}

\( x_1 + x_2 \leq 400 \) \hspace{1cm} (4) \hspace{1cm} \text{This is a certificate of optimality for the solution found by LP!}

\( x_1, x_2 \geq 0 \) \hspace{1cm} (5)
## Constructing Dual Problem

- Introduce a multiplier $y_i$ for each equation:

<table>
<thead>
<tr>
<th>Multiplier</th>
<th>Inequality</th>
</tr>
</thead>
<tbody>
<tr>
<td>$y_1$</td>
<td>$x_1 \leq 200$</td>
</tr>
<tr>
<td>$y_2$</td>
<td>$x_2 \leq 300$</td>
</tr>
<tr>
<td>$y_3$</td>
<td>$x_1 + x_2 \leq 400$</td>
</tr>
</tbody>
</table>

- After multiplying 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:

**Min**

$$200y_1 + 300y_2 + 400y_3$$

- $y_1 + y_3 \geq 1$
- $y_2 + y_3 \geq 6$
- $y_1, y_2, y_3 \geq 0$
Duality

Primal LP:

\[
\begin{align*}
\text{max } & \quad c^T x \\
\text{s.t. } & \quad Ax \leq b \\
& \quad x \geq 0
\end{align*}
\]

Dual LP:

\[
\begin{align*}
\text{min } & \quad y^T b \\
\text{s.t. } & \quad y^T A \geq c^T \\
& \quad y \geq 0
\end{align*}
\]

Theorem (Duality)

*If a linear program has a bounded optimum, then so does its 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

A polyhedron defined by seven inequalities.

\[ \begin{align*}
1 \leq 4 & \\
2 \leq 3 & \\
3 \leq 5 & \\
4 \leq 6 & \\
5 \leq 7 & \\
6 \geq 0 & \\
7 \geq 0 & \\
\end{align*} \]

Figure 7.12

7.6.1 Vertices and neighbors in \( n \)-dimensional space

Figure 7.12 recalls an earlier example. Looking at it closely, we see that each vertex is the unique point at which some subset of hyperplanes meet. Vertex \( A \), for instance, is the sole point at which constraints \( 2 \) and \( 3 \) and \( 7 \) are satisfied with equality. On the other hand, the hyperplanes corresponding to inequalities \( 4 \) and \( 6 \) do not define a vertex, because their intersection is not just a single point but an entire line. Let's make this definition precise.

Pick a subset of the inequalities. If there is a unique point that satisfies them with equality, and this point happens to be feasible, then it is a vertex.

How many equations are needed to uniquely identify a point? When there are \( n \) variables, we need at least \( n \) linear equations if we want a unique solution. On the other hand, having more than \( n \) equations is redundant: at least one of them can be rewritten as a linear combination of the others and can therefore be disregarded. In short, each vertex is specified by a set of \( n \) inequalities.

A notion of neighbor now follows naturally. Two vertices are neighbors if they have \( n-1 \) defining inequalities in common.

In Figure 7.12, for instance, vertices \( A \) and \( C \) share the two defining inequalities \{\( 3 \), \( 7 \)\} and are thus neighbors.

There is one tricky issue here. It is possible that the same vertex might be generated by different subsets of inequalities. In Figure 7.12, vertex \( B \) is generated by \{\( 2 \), \( 3 \), \( 4 \)\}, but also by \{\( 2 \), \( 4 \), \( 5 \)\}. Such vertices are called degenerate and require special consideration. Let's assume for the time being that they don't exist, and we'll return to them later.
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.