Evening Lecture - April 7th, 2005

- LP so far
- Parametric Linear Programming
- Upper Bound Technique
- Where the Simplex Method can be inefficient
- Interior Points Method
- Network Simplex Methods

1  **Linear Programming so far:**

- Formulating them
- Forms
- Basic solutions
- Geometry
- Optimality
- Simplex
- Duality & Sensitivity
- What happens when you change the problem?
- How to use the theoretical results

2  **Dual Simplex Method**

- Dual feasible but no longer primal feasible
- Find the basic solution to primal with the first line of the table all >= 0
- Feasibility Test: -Select worst exit variable vs. basic variable -Select entering variable to transform first line to a zero (min ratio test)
- Solve to get table in proper form

\[
\begin{align*}
\text{max } & y^T b \\ 
\text{dual } & \text{min } q^T c \\
\end{align*}
\]

Infeasible Solution
p.295 of the text

<table>
<thead>
<tr>
<th></th>
<th>z</th>
<th>y_1</th>
<th>y_2</th>
<th>y_3</th>
<th>y_4</th>
<th>y_5</th>
<th>RHS</th>
</tr>
</thead>
<tbody>
<tr>
<td>z</td>
<td>1</td>
<td>4</td>
<td>12</td>
<td>18</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>y_4</td>
<td>0</td>
<td>-1</td>
<td>0</td>
<td>-3</td>
<td>1</td>
<td>0</td>
<td>-3</td>
</tr>
<tr>
<td>y_5</td>
<td>0</td>
<td>0</td>
<td>-3</td>
<td>-2</td>
<td>0</td>
<td>1</td>
<td>-5</td>
</tr>
</tbody>
</table>

-5 in the \( y_5 \) row is the biggest violation by being the most negative
- With the exiting variable only look at columns < 0
- Columns \( y_2 \) & \( y_3 \) could become basic entry variables
\[
\frac{-12}{2} < \frac{-18}{2}
\]

Is this feasible? No, perform the ratio test on \( y_4 \)

\[
\begin{array}{cccccc}
  & z & y_1 & y_2 & y_3 & y_4 & y_5 & \text{RHS} \\
 z & 1 & 4 & 0 & 6 & 0 & 6 & -30 \\
y_4 & 0 & -1 & 0 & -3 & 1 & 0 & -3 \\
y_2 & 0 & 0 & 1 & 1 & 0 & \frac{1}{2} & \frac{9}{2} \\
\end{array}
\]

3 Parametric Linear Programming \( P(\lambda) \)

\[
\text{max}(1 - 2\lambda)x_1 + (4 - \lambda)x_2
\]

\[
\text{s.t. } x_1 + 2x_2 \leq 6 \\
- x_1 + 3x_2 \leq 6 \\
x_1, x_2 \geq 0
\]

Use Simplex Method & leave \( \lambda \) as a variable

\[
\begin{array}{cccccc}
  & z & x_1 & x_2 & x_3 & x_4 & \text{RHS} \\
 z & 1 & (1 - 2\lambda) & (4 - \lambda) & 0 & 0 & 0 \\
x_3 & 0 & 1 & 2 & 1 & 0 & 6 \\
x_4 & 0 & -1 & 3 & 0 & 1 & 6 \\
\end{array}
\]

For \( \lambda \leq \frac{1}{2}x^4 = (0, 0, 6, 6) \)

Pivot Steps

\[
\begin{array}{ccccccc}
  & z & x_1 & x_2 & x_3 & x_4 & \text{RHS} \\
 z & 1 & 0 & 0 & (\frac{7}{5} - \frac{7}{5}\lambda) & (\frac{3}{5} + \frac{3}{5}\lambda) & (\frac{-56}{5} - \frac{24}{5}\lambda) \\
x_3 & 0 & 1 & 0 & \frac{3}{5} & \frac{3}{5} & \frac{6}{5} \\
x_4 & 0 & 0 & 1 & \frac{1}{5} & \frac{1}{5} & \frac{12}{5} \\
\end{array}
\]

\[1 - \lambda \geq 0 \text{ i.e. } \lambda \leq 1\]

\[3\lambda \geq -2\]

\[x_3, x_4, x_1, x_2\]

\[0 \leq \lambda \leq \frac{1}{2} \quad \frac{1}{2} \leq \lambda \leq 1\]

- The optimal solution depends on lambda
- Parametric Linear Programming is used for sensitivity analysis
- Simplex Method can get inefficient as constraints get large
4 Interior Point Algorithms

- Search through interior of feasible region
- “Gradient Descent” idea is to improve as fast as possible
- Transform Linear Programming problems into a feasibility problem

\[
\begin{align*}
\text{max} \quad & z = E_x \\
\text{min} \quad & y^T b \\
\text{s.t.} \quad & A_x \leq b \\
& x \geq 0 \\
\text{s.t.} \quad & A^T y \leq c \\
& y \geq 0
\end{align*}
\]

Find a feasible \( \hat{x} \) to \( \hat{A} \hat{x} \leq \hat{b} \)

\[
\hat{A} \begin{pmatrix}
-c^T & y^T \\
A & 0 \\
0 & -A \\
-I & 0 \\
0 & -I
\end{pmatrix} \begin{pmatrix}
x \\
y
\end{pmatrix} \leq \begin{pmatrix}
0 \\
b \\
c \\
0
\end{pmatrix}
\]

Which is equivalent to:

\( \hat{A} \cdot \hat{x} \leq \hat{b} \)

- Reduces maximization to finding a feasible solution
- Minimum cost flows:
  \( x_{ij} = \text{flow along arc } i \to j \text{ nodes} \)

Typical problem:

\[
\begin{align*}
\text{min} \quad & z = c^T x \\
\text{s.t.} \quad & A_x = b \\
\ell \leq x \leq \mu \quad \text{with } (-\infty & \infty) \text{ ok}
\end{align*}
\]

(\text{flow out of node}) - (\text{flow into node}) = \( b_i \)

\( b_i > 0 \) source \( b_i \leq \sinh b_i = 0 \) transshipment

min \( 12x_{12} + 8x_{26} + \cdots \)

s.t. \( \sum_j x_{ij} - \sum_k x_{ik} = b_i \)

\[
\begin{pmatrix}
1 & 1 & 0 & 0 & 0 \\
-1 & 0 & 1 & 1 & -1 \\
1 & -1 & 0 & 0 & 1 \\
1 & 1 & -1 & -1 & 0
\end{pmatrix}
\]

These are made up values in the matrix

5 Setting Up Networks

Make sure supply = demand
  - Image should go here
  - Assign people to tasks
  - Image should go here
  - \( c_{ij} = \text{cost for person } i \text{ to perform task } j \)

\[
\begin{align*}
\text{min} \quad & 11x_{1A} + 5x_{1B} + \cdots \\
\text{s.t.} \quad & x_{1A} + x_{1B} + x_{1C} = 1
\end{align*}
\]

- Shortest Path (cost can be distance)
- Maxflow through a network

3
• Cuts in a graph (network)

• Consider the max flow problem: if we pick a ‘cut’ in the network then the capacity through that out is an upper bound or max flow. Further, min cut = max flow.

insert image here
find the bottleneck and all the ways to cut it