Linear Programming

A political problem

Standard formulation of LP

Shortest paths, maximum flow &
multi-commodity flow as LP

A Political Problem

Politician trying to win an election

<table>
<thead>
<tr>
<th>Area</th>
<th>Population</th>
</tr>
</thead>
<tbody>
<tr>
<td>Urban</td>
<td>100,000</td>
</tr>
<tr>
<td>Suburban</td>
<td>200,000</td>
</tr>
<tr>
<td>Rural</td>
<td>50,000</td>
</tr>
</tbody>
</table>

Issues: More roads
        Gun control
        Farm subsidies
        Gasoline Tax

To govern effectively, you want to win a majority in each of the three areas

Campaign staff estimates how many votes you can win by spending $1000 in advertising in support of a particular issue.
# ESTIMATES

<table>
<thead>
<tr>
<th>Policy</th>
<th>Urban</th>
<th>Suburban</th>
<th>Rural</th>
</tr>
</thead>
<tbody>
<tr>
<td>Building Roads</td>
<td>-2</td>
<td>5</td>
<td>3</td>
</tr>
<tr>
<td>Gun Control</td>
<td>8</td>
<td>2</td>
<td>-5</td>
</tr>
<tr>
<td>Farm Subsidies</td>
<td>0</td>
<td>0</td>
<td>10</td>
</tr>
<tr>
<td>Gasoline Tax</td>
<td>10</td>
<td>0</td>
<td>-2</td>
</tr>
</tbody>
</table>

Table represents 1000's of votes won by advertising worth $1000.

Goal: Figure out the minimum amount of money that you need to spend in order to win 50,000 urban votes, 100,000 suburban votes, and 25,000 rural votes.

## TRIAL AND ERROR.

$20k$ to B.R., $0$ to G.C., $4k$ to F.S., $9k$ to H.T.

Total = $33k$

$20(-2) + 0(8) + 4(0) + 9(10) = 50k$ urban votes

$20(5) + 0(2) + 4(0) + 9(0) = 100k$ suburban votes

$20(3) + 0(-5) + 4(10) + 9(-2) = 82k$ rural votes more votes than voters

Is $33k$ minimum cost? No.
FORMULATION

4 Variables $x_1, x_2, x_3, x_4$

100s of dollars spent on advertising for building roads (B.R.)

G.C. F.S. G.T.

At least 50k urban votes: $-2x_1 + 8x_2 + 0x_3 + 10x_4 \geq 50$

"100k Suburban": $5x_1 + 2x_2 + 0x_3 + 0x_4 \geq 100$

"25k rural": $3x_1 - 5x_2 + 10x_3 - 2x_4 \geq 25$

Minimize $x_1 + x_2 + x_3 + x_4$

(No such thing as negative-cost advertising)

$x_1 \geq 0$, $x_2 \geq 0$, $x_3 \geq 0$, $x_4 \geq 0$

Q: What if you want a simple majority $\geq 175K$ votes?

General Linear Programs.

Linear function $f(x_1, x_2, \ldots, x_n) = \sum_{j=1}^{n} a_j x_j$

$f(x_1, x_2, \ldots, x_n) = b$ linear equality

$\geq b$ linear inequality

$\leq b$ (not $< or >$)

Linear programming: Minimizing or maximizing a linear function subject to a finite set of linear constraints.
Linear Programming (LP) vs. Integer-Linear Programming (ILP)

**LP:** variables \( x_i \) are real-valued

**ILP:** variables \( x_i \) constrained to be integers

**LP:** \( 1 \leq x_1 \leq 3 \), \( x_1 = \sqrt{2} \), \( x_1 = 1.21 \)

**ILP:** \( 1 \leq x_1 \leq 3 \), \( x_1 = 1, 2 \) or 3

Can translate \( 1 < x_1 < 3 \) to \( 2 \leq x_1 \leq 3 \) in ILP but will have to approximate constraint in LP as \( 1 + \varepsilon \leq x_1 \leq 3 \) for some fixed \( \varepsilon \)

LP is polynomial-time solvable \( \rightarrow \) ILP is NP-hard

**STANDARD FORM**

Maximize \( \sum_{j=1}^{n} c_j x_j \) \( \rightarrow \) objective function

Subject to \( \sum_{i=1}^{n} a_{ij} x_j \leq b_i \), for \( i = 1, 2, \ldots, m \)

\( \sum_{i=1}^{n} a_{ij} x_j \geq b_i \), for \( j = 1, 2, \ldots, n \)

Constraints

\( x_j > 0 \) \( \rightarrow \) non-negativity constraints

1) Want to minimize \( -2x_1 + 3x_2 \). Negate coefficients & maximize \( 2x_1 - 3x_2 \)

2) If \( x_j \) does not have a non-negativity constraint, \( x_j \) replaced by \( x_j' - x_j'' \), \( x_j' \geq 0 \), \( x_j'' \geq 0 \)

3) Equality constraint \( x_1 + x_2 = 7 \) translates to \( x_1 + x_2 \leq 7 \), \( x_1 + x_2 \geq 7 \)

4) \( \geq \) constraint translated to \( \leq \) by multiplication of \( -1 \)

\( x_1 + x_2 \geq 7 \) \( \rightarrow \) \( -x_1 - x_2 \leq -7 \)
**Maximum Flow**

![Graph with nodes and edges]

- \( c(u,v) \): given capacities
- Recall \( c(u,v) = 0 \) if \((u,v)\) doesn't exist

Work with positive flows \( p(u,v) \) (Standard form)

**Maximize** \[\sum_{v \in V} p(s,v)\] (outflow from source)

**Subject to** \[p(u,v) \leq c(u,v) \quad \forall (u,v) \in E\]

\[\sum_{(w,u) \in E} p(w,u) = \sum_{(u,w) \in E} p(u,w) \quad \forall u \in \{s-t\}\]

\[\sum_{(s,w) \in E} p(s,w) = \sum_{(w,t) \in E} p(w,t)\]

- Can maximize weighted flows instead & numerous other cost functions

Maximize \[\sum_{v \in V} p(s,v)\] for example

**Edmonds-Karp** much more efficient for conventional maximum flow.
Single-pair Shortest Paths

\[ d[s] = 0 \]
\[ d[v] \]

Assume \( w(u,v) \geq 0 \)

How to represent as LP?
Don't want to enumerate paths → could be exponential!

If there is an edge between \( u \) and \( v \), we have a relationship between \( d[u] \) and \( d[v] \):
\[ d[v] \leq d[u] + w(u,v) \]

If a shortest path includes \((u,v)\), \[ d[v] = d[u] + w(u,v) \]

Maximize \[ d[t] \]
Subject to
\[ d[v] \leq d[u] + w(u,v) \quad \forall (u,v) \in E \]
\[ d[s] = 0, \quad d[u] \geq 0, \quad \forall u \neq s \in V \]

Q: What about negative-edge weights?
NEGATIVE WEIGHTS

Still works! Can use LP to detect negative cycles.

\[ x_A, x_B, x_C, x_D, x_E \] represent \( d[A], d[B], \ldots \)

\[ x_A = 0 \]

\[ x_B \leq x_A - 1 \]
\[ x_C \leq x_B + 3 \]
\[ x_D \leq x_B + 1, \quad x_B \leq x_D + 2, \]

etc.

Thm: Iff the constraint graph contains a negative-weight cycle, then the system of differences is unsatisfiable.

If part:

\[ x_B - x_A \leq w(A, B) \]
\[ x_C - x_B \leq w(B, C) \quad \text{Cycle } A \rightarrow B \rightarrow C \rightarrow A \]
\[ x_A - x_C \leq w(C, A) \]

\[ 0 \leq \text{weight of cycle} \]
\[ < 0 \quad \text{contradiction.} \]

Only if part more involved. CLRS p.604, Thm 24.9
Multi-commodity flow

Real power of LP is in solving new problems for which no specific poly-time algorithm is known.

Cars & trucks on a highway system

\[
\begin{aligned}
\text{(u,v)} & \quad \text{p}_1(u,v) \quad \text{cars} \\
\text{(u,v)} & \quad \text{p}_2(u,v) \quad \text{trucks}
\end{aligned}
\]

Flow for cars is conserved. Will work with positive flows.
Flow for trucks is conserved.

Want to push \(d_1\) flow through network, i.e. \(d_1\) for cars, \(d_2\) for trucks.

Maximize \(0 \leq \sum_{i=1}^{k} p_i(u,v) \leq c(u,v) + \gamma(u,v)\)

Subject to \(\forall i, \forall u \neq s_i, t_i \quad (u,v) \in E\)

\(\forall i \quad \sum_{(s_i,w) \in E} p_i(s_i,w) = \sum_{(w,t_i) \in E} p_i(w,t_i) = d_i\)

Can maximize weighted flows instead & numerous other cost functions

\[
\sum_{i=1}^{k} a_i \cdot p_i(s_i,w) \leq c(s_i,w) \quad \forall (s_i,w) \in E
\]