Linear Programming and Network Optimization

Jon Turner
Computer Science & Engineering
Washington University

www.arl.wustl.edu/~jst
Linear Programming

- A linear program seeks values for a set of non-negative real variables $x_i$ that optimize a linear function of the $x_i$ subject to linear constraints
  - maximize $\sum_i c_i x_i$ subject to $\sum_j a_{ij} x_j \leq b_i$ for all $i$
  - or in matrix form, maximize $C^T X$ subject to $AX \leq B$
- Linear programs can be solved efficiently
  - classical simplex method has exponential worst-case but is fast in practice
  - interior point method runs in polynomial time
- If some or all of the $x_i$ are constrained to be integers, we get an *Integer Linear Program*
  - in general, these are *NP*-hard
Max Flow as LP

- Defined over flow variables $f_e$ for each edge $e$
  - maximize $\Sigma_{e=(s,u)} f_e$ subject to $0 \leq f_e \leq cap_e$ for all $e$ and $\Sigma_{e=(u,v)} f_e = \Sigma_{e=(u,v)} f_e$ for all $u$ is $s$ or $t$
  - to put this into the standard matrix form
    - let $F$ be column vector with an entry per edge
    - let $S$ be a column vector with a 1 entry for every edge leaving the source vertex and a 0 entry for all other edges
    - let $I$ be the identity matrix with $m$ rows and columns
    - let $G=[g_{ue}]$ be an edge incidence matrix where for $u$ is $s$ or $t$, $g_{ue}=1$ if $u$ is the tail of $e$ and $g_{ue}=-1$ if $u$ is the head of $e$
    - define coefficient matrix $A$ by “stacking” $I$ above $G$ above $-G$
    - let $B$ be a column matrix with $m+2(n-2)$ entries where first $m$ are the edge capacities and remainder are all 0
    - so LP becomes: maximize $S^TF$ subject to $F \geq 0$ and $AF \leq B$
Min Cost Flow as LP

- Also defined over flow variables $f_e$ for each edge $e$
  - minimize $\Sigma_e cost f_e$ subject to $\Sigma_{e=(s,u)} f_e = f_s^u$ and $0 \leq f_e \leq cap_e$ for all $e$ and $\Sigma_{e=(w,u)} f_e = \Sigma_{e=(u,v)} f_e$ for all $u \neq s$ or $t$
  - this can also be put into standard matrix form by
    - switching to a maximization problem (maximize $-\Sigma_e cost f_e$)
    - expanding coefficient matrix and constraint vector from max flow by adding two rows to express constraint on total flow

- Integrality property for max flow and min cost flow
  - if capacities are integers then optimal flows are also
  - consequence of a general property of coefficient matrix
    - a coefficient matrix is totally unimodular if every square sub-matrix has a determinant equal to 0, 1 or $-1$
  - any LP with integer coefficients and bounds, and a totally unimodular coefficient matrix, has an integral optimum
Multicommodity Flow Problem

- Several types of “stuff” (called commodities) to be moved through a network
  - can define a separate source and sink for each commodity
  - each edge can have total flow capacity plus (optional) limits on individual commodity flows
  - non source/sink nodes must preserve flow of each commodity

- Can be formulated as LP
  - meaning that it can be solved reasonably efficiently even if we generalize by adding costs, more flow constraints
  - in general, does not satisfy integrality property
    - coefficient matrix is not totally unimodular
  - no substantially better solution method than LP
Shortest Path Problem as LP

- For single-source, single-sink version, costs > 0
  - use \{0, 1\} selection variables \(x_e\) to define path (so ILP)
  - minimize \(\Sigma_e \text{ cost}_e \times x_e\) subject to \(\Sigma_{e=(u,t)} x_e \geq 1\) and \(\Sigma_{e=(w,u)} x_e \geq \Sigma_{e=(u,v)} x_e\) for all \(u \neq s\) or \(t\)

\[
\text{minimize } C = 3x_{a_0} + 6x_{b_0} + 2x_{a_6} + 7x_{x_1} + 4x_{b_4}
\]
subject to \(AX \geq B\)

- can also formulate as a min-cost flow problem (\(x_e = f_e\))
  - because capacities are all 1, integrality property for min cost flows implies \(x_e\) values of an optimal solution are integers
  - so can find optimal solution of shortest path ILP using LP
Alternate LP for Shortest Path

- Imagine a graph as a set of balls connected by strings of different length
  - pull the source and sink balls as far apart as possible
    - distance separating them is the shortest path distance
- Leads to maximization problem
  - maximize $d_t$ subject to $d_s \leq d_u + \text{cost}_{uv}$ for all edges $(u,v)$ and $d_s = 0$

$$
\begin{align*}
\text{maximize } d_t & \text{ subject to } \\
& 
\begin{bmatrix}
1 & 0 & 0 \\
0 & 1 & 0 \\
-1 & 1 & 0 \\
-1 & 0 & 1 \\
0 & -1 & 1 \\
\end{bmatrix}
\begin{bmatrix}
d_s \\
d_t \\
\end{bmatrix}
\leq 
\begin{bmatrix}
3 \\
6 \\
2 \\
7 \\
4 \\
\end{bmatrix}
\end{align*}
$$

- this is the dual of the original LP
Duality

- Standard form LP: maximize $C^TX$ subject to $AX \leq B$
- **Dual:** minimize $B^TZ$ subject to $A^TZ \geq C$ where the vector $Z$ is made up of dual variables
- The optimal solution values of the primal and dual forms are equal – $C^TX^* = B^TZ^*$
  - sometimes the dual is easier to solve than primal
- Alternate forms
  - can convert to minimization by negating $C$
  - can change $\leq$-bounds to $\geq$-bounds by negating $A$ and $B$
  - so for example, if primal expressed as minimize $C^TX$
    subject to $AX \leq B$, dual is minimize $B^TZ$ subject to $A^TZ \geq -C$
Complementary Slackness

- Primal: maximize $C^TX$ subject to $AX \leq B$
  - $B-AX$ is referred to as slack in primal variables
- Dual: minimize $B^TZ$ subject to $A^TZ \geq C$
  - $A^TZ-C$ is referred to as slack in dual variables
- Complementary slackness condition states that $X^*$ and $Z^*$ are optimal solutions if and only if
  $$(B-AX^*) = [s_i] \Rightarrow s_i z_i^* = 0 \text{ for all } i \text{ and}$$
  $$(A^TZ^*-C) = [t_j] \Rightarrow t_j x_j^* = 0 \text{ for all } j$$
  - so each non-zero slack value in primal (dual) corresponds to a zero dual (primal) variable
  - *primal-dual algorithms* adjust values of primal and dual variables with objective of making these conditions true
Shortest Path & Complementary Slackness

\[
\begin{align*}
\text{minimize } & \quad CX = 3x_{st} + 6x_{tb} + 2x_{sd} + 7x_{at} + 4x_{st} \\
\text{subject to } & \quad \begin{bmatrix} 1 & 0 & -1 & -1 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & -1 & 0 & 1 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & -1 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_{sa} \\ x_{bc} \\ x_{ab} \\ x_{bt} \\ x_{ad} \\ x_{tc} \\ x_{ta} \end{bmatrix} \geq \begin{bmatrix} 0 \\ 0 \\ 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} \quad \begin{bmatrix} 3 \\ 7 \\ 6 \\ 4 \end{bmatrix}
\end{align*}
\]

\[
\begin{align*}
\text{maximize } & \quad d_i \quad \text{subject to } \quad \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 1 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & -1 & 0 \\ 0 & 1 & 1 \end{bmatrix} \begin{bmatrix} d_s \\ d_t \\ d_a \\ d_b \end{bmatrix} \leq \begin{bmatrix} 3 \\ 6 \\ 2 \\ 7 \end{bmatrix}
\end{align*}
\]

- For primal, optimal solution \(X^* = [1 \ 0 \ 1 \ 0 \ 1]\)
- For dual, optimal solution \(D^* = [3 \ 5 \ 9]\)
- Complementary slackness conditions
  \( (A^TD^* - C)^T X^* = [0] \) and \((B - AX^*)^T D^* = [0] \)

\[
\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ -1 & 1 & 0 \\ -1 & 0 & 1 \\ 0 & -1 & 1 \end{bmatrix} \begin{bmatrix} 3 \\ 5 \\ 9 \\ 7 \\ 4 \end{bmatrix} \begin{bmatrix} \begin{bmatrix} 3 \\ 6 \\ 2 \\ 7 \\ 4 \end{bmatrix}^T \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 1 \\ 0 & 0 & 1 \\ 1 & 0 & 1 \\ 1 & 0 & 1 \end{bmatrix} \begin{bmatrix} 3 \\ 5 \\ 9 \\ 7 \\ 4 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \end{bmatrix}
\]
Max Matching as ILP

- ILP for maximum size matching problem using 0-1 selection variables $X = [x_e]$
  - maximize $\sum_u x_u$ subject to $\sum_e = (u,v) x_e \leq 1$ for all $u$
  - to get matrix form, let $G = [g_{ue}]$ be incidence matrix of graph where $g_{ue} = 1$ if $u$ is an endpoint of $e$, else 0
  - maximize $[1]^T X$ subject to $X \geq 0$ and $GX \leq [1]$

- For weighted matching, let $W$ be column vector of edge weights, then
  - maximize $W^T X$ subject to $X \geq 0$ and $GX \leq [1]$
    - can get LP with same optimal solutions by adding constraints
  - dual: minimize $[1]^T Z$ subject to $Z \geq 0$ and $G^T Z \geq W$
    - the variables $z_u$ can be thought of as vertex labels and the constraints take form $z_u + z_v \geq w_e$ for all $e = (u,v)$