Examples of Network Flow Problems

John E. Mitchell

1 Shortest path problem

Definition 1. Given a directed graph \( D = (V, E) \) with a weight function \( w : E \to \mathbb{R} \) and
start and end vertices \( s, t \in V \). The weight of a \((s,t)\)-path is the sum of the weights over all
arcs in the path. The **shortest path problem** is to find the \((s,t)\)-path of minimum weight.

If all the edge weights \( w_e \) are nonnegative, the problem can be solved using Dijkstra’s
algorithm, a dynamic programming approach.

The problem can be expressed as a linear program. Let \( A \) denote the node-arc incidence
matrix, so

\[
A_{ve} = \begin{cases} 
-1 & \text{if arc } e \text { leaves vertex } v \\
+1 & \text{if arc } e \text { enters vertex } v \\
0 & \text{otherwise}
\end{cases}
\]

Assume the rows of \( A \) are ordered with \( s \) the first row and \( t \) the last row. The LP formulation
is then

\[
\min_{x \in \mathbb{R}^{|E|}} \sum_{e \in E} w_e x_e \\
\text{subject to} \quad Ax = \begin{bmatrix} -1 \\ 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix} \\
x \geq 0
\]

The structure of the constraint matrix is such that any optimal BFS to this LP is binary.
(The constraint matrix is totally unimodular: the determinant of any submatrix is 0, 1,
or -1.)

The dual LP is

\[
\max_{y \in \mathbb{R}^{|V|}} \sum_{i \in V} y_i - y_s \\
\text{subject to} \quad y_j - y_i \leq w_{ij} \quad \text{for all arcs } (i,j).
\]

For any optimal solution and for any node \( j \) on the optimal path, \( y_j - y_s \) is equal to the
length of the shortest path from node \( s \) to node \( j \). This can be shown using complementary
slackness.

Exercise 1. If the graph has a negative length cycle then the dual problem is infeasible.
2 Maximum flow in a network

Definition 2. Given a directed graph $D = (V, E)$ with start and end vertices $s, t \in V$. Let each edge $e \in E$ have a capacity $u_e$ (possibly infinite). The maximum flow problem is to maximize flow from $s$ to $t$ in the network.

This problem can be written as a linear program. First, we define a vector

$$d := (1, 0, \ldots, 0, -1)^T \in \mathbb{R}^{|V|}$$

The LP formulation is then

$$\max_{x \in \mathbb{R}^{|E|}, r \in \mathbb{R}} \quad r$$

subject to

$$Ax + dr = 0$$

$$x \leq u$$

$$x \geq 0$$

2.1 Cuts

Definition 3. An $s$–$t$ cut is a partition $(W, \bar{W})$ of the nodes of $V$ into two sets $W \ni s$ and $\bar{W} \ni t$. The capacity of the cut is

$$C(W, \bar{W}) := \sum_{(i,j) \in E, i \in W, j \in \bar{W}} u_{ij}.$$

The dual to the max flow problem is

$$\min_{\pi \in \mathbb{R}^{|V|}, y \in \mathbb{R}^{|E|}} \quad \sum_{(i,j) \in E} u_{ij} y_{ij}$$

subject to

$$\pi_j - \pi_i + y_{ij} \geq 0 \quad \text{for all } (i,j) \in E$$

$$\pi_s - \pi_t = 1$$

$$y \geq 0$$

Theorem 1. Every $(s,t)$-cut determines a feasible solution with cost $C(W,\bar{W})$ to the dual of the max flow problem through the assignment

$$y_{ij} = \begin{cases} 1 & \text{if } i \in W, j \in \bar{W} \\ 0 & \text{otherwise} \end{cases}$$

$$\pi_i = \begin{cases} 1 & \text{if } i \in W \\ 0 & \text{otherwise} \end{cases}$$

The proof of this theorem and the next one are left as exercises.

Theorem 2. The value of the maximum flow from $s$ to $t$ equals the capacity of the minimum $(s,t)$-cut. A flow $x$ and a cut $(W,\bar{W})$ are jointly optimal if and only if

$$x_{ij} = 0 \quad \text{for all } (i, j) \in E \text{ with } i \in \bar{W}, j \in W$$

$$x_{ij} = u_{ij} \quad \text{for all } (i, j) \in E \text{ with } i \in W, j \in \bar{W}$$
2.2 Augmenting path algorithm

The augmenting path algorithm exploits the relationship between cuts and flows to find an optimal flow.

Let \( W \) be the vertices to which we can push flow from \( s \), namely vertices \( \{s, 3, 5\} \). Arc \((4, 5)\) flows backwards across the cut and carries positive flow, so this lets us augment the flow.

The largest possible value is \( \epsilon = 2 \). This gives an updated flow:  

Now the cut is saturated. The value of the flow is equal to the capacity of the cut. The augmenting path algorithm runs in polynomial time provided:

- the capacities \( u_{ij} \) are all integer
- the augmented path with fewest arcs is chosen
3 Minimum cost circulation problem

Definition 4. Given a directed graph $D = (V, E)$ with a weight function $w : E \rightarrow \mathbb{R}$, with each vertex having a net demand $b_v$. Assume $\sum_{v \in V} b_v = 0$. Let each edge $e \in E$ have a capacity $u_e$ (possibly infinite). A feasible flow $x$ satisfying $0 \leq x \leq u$ has a net flow of $b_v$ into node $v$ for all $v \in V$. The **minimum cost circulation problem** is to meet all the demands at minimum cost.

The LP formulation for this problem is:

$$\min_{x \in \mathbb{R}^{|E|}} \quad w^T x$$

subject to

$$Ax = b$$
$$x \leq u$$
$$x \geq 0$$

The maximum flow problem can be cast as a minimum cost circulation problem:

introduce a return arc from $t$ to $s$ with infinite capacity and weight $-1$; give all other edges weight $w_e = 0$; give all vertices demand $b_v = 0$.

The network simplex algorithm (see later) is geared to the minimum cost circulation problem.