# 8 Graphs and Flows

Lectures 8 through 11 will be concerned with flow problems on graphs and networks.

A directed *graph*, or *network*, $G = (V, E)$ consists of a set $V$ of *vertices* and a set $E \subseteq V \times V$ of *edges*. When the relation $E$ is symmetric, $G$ is called an undirected graph, and we can write edges as unordered pairs $\{i, j\} \in E$ for $i, j \in V$. The *degree* of vertex $i \in V$ in graph $G$ is the number $|\{j \in V : (i, j) \in E \text{ or } (j, i) \in E\}|$ of other vertices connected to it by an edge. A *walk* from $u \in V$ to $w \in V$ is a sequence of vertices $v_1, \dots, v_k \in V$ such that $v_1 = u$, $v_k = w$, and $(v_i, v_{i+1}) \in E$ for $i = 1, \dots, k-1$. In a directed graph, we can also consider an undirected walk where $(v_i, v_{i+1}) \in E$ or $(v_{i+1}, v_i) \in E$ for $i = 1, \dots, k-1$. A walk is a *path* if $v_1, \dots, v_k$ are pairwise distinct, and a *cycle* if furthermore $v_1 = v_k$. A graph that does not contain any cycles is called *acyclic*. A graph is called *connected* if for every pair of vertices $u, v \in V$ there is an undirected path from $u$ to $v$. A *tree* is a graph that is connected and acyclic. A graph $G' = (V', E')$ is a subgraph of graph $G = (V, E)$ if $V' \subseteq V$ and $E' \subseteq E$. In the special case where $G'$ is a tree and $V' = V$, it is called a *spanning tree* of $G$.

## 8.1 Minimum Cost Flows

Consider a network $G = (V, E)$ with $|V| = n$, and let $b \in \mathbb{R}^n$. Here, $b_i$ denotes the amount of flow that enters or leaves the network at vertex $i \in V$. If $b_i > 0$, we say that $i$ is a *source* supplying $b_i$ units of flow. If $b_i < 0$, we say that $i$ is a *sink* with a demand of $|b_i|$ units of flow. Further let $C, \underline{M}, \overline{M} \in \mathbb{R}^{n \times n}$, where $c_{ij}$ denotes the cost associated with one unit of flow on edge $(i, j) \in E$, and $\underline{m}_{ij}$ and $\overline{m}_{ij}$ respectively denote lower and upper bounds on the flow across this edge. The minimum cost flow problem then asks for flows $x_{ij}$ that conserve the flow at each vertex, respect the upper and lower bounds, and minimize the overall cost. Formally, $x \in \mathbb{R}^{n \times n}$ is a *minimum cost flow* of $G$ if it is an optimal solution of the following optimization problem:

$$
\begin{aligned}
\text{minimize} \quad & \sum_{(i,j) \in E} c_{ij} x_{ij} \\
\text{subject to} \quad & b_i + \sum_{j:(j,i) \in E} x_{ji} = \sum_{j:(i,j) \in E} x_{ij} && \text{for all } i \in V, \\
& \underline{m}_{ij} \leqslant x_{ij} \leqslant \overline{m}_{ij} && \text{for all } (i, j) \in E.
\end{aligned}
$$

Note that $\sum_{i \in V} b_i = 0$ is required for any feasible flows to exist, and we make this assumption in the following. We further assume without loss of generality that the network $G$ is connected. Otherwise the problem can be decomposed into several smaller problems that can be solved independently. An important special case is that of *uncapacitated flows*, where $\underline{m}_{ij} = 0$ and $\overline{m}_{ij} = \infty$ for all $(i, j) \in E$.

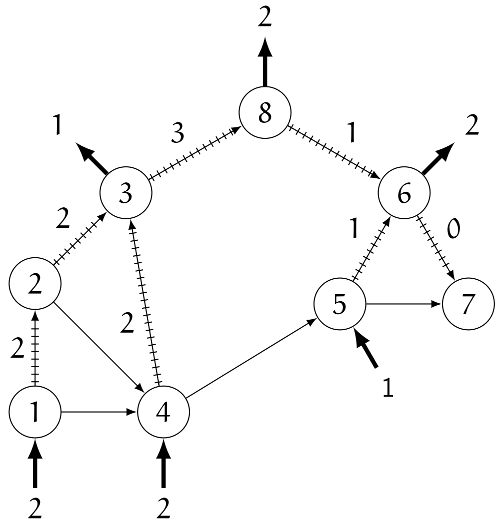


Figure 8.1: A flow network with a spanning tree T indicated by hatched edges. Since the network is uncapacitated, we have to set $L = E \setminus T$ and $U = \emptyset$, and thus flows are zero for edges not in T. Flows for the edges in T can be determined inductively starting from the leafs. Note that the resulting spanning tree solution is feasible.

The minimum cost flow problem is a linear programming problem, with constraints of the form $Ax = b$ where

$$a_{ik} = \begin{cases} 1 & \text{kth edge starts at vertex } i, \\ -1 & \text{kth edge ends at vertex } i, \\ 0 & \text{otherwise.} \end{cases}$$

Given this rather simple structure, we may hope that minimum cost flow problems are easier to solve than general linear programs. Indeed, we will see that basic feasible solutions of a minimum cost flow problem take a special form, and will obtain an algorithm that exploits this form.

## 8.2 Spanning Tree Solutions

Consider a minimum cost flow problem for a connected network $G = (V, E)$. A solution $x$ to this problem is called *spanning tree solution* if there exists a spanning tree $(V, T)$ of G and two sets $L, U \subseteq E$ with $L \cap U = \emptyset$ and $L \cup U = E \setminus T$ such that $x_{ij} = \underline{m}_{ij}$ if $(i, j) \in L$ and $x_{ij} = \overline{m}_{ij}$ if $(i, j) \in U$. For every choice of T, L and U, the flow conservation constraints uniquely determine the values $x_{ij}$ for $(i, j) \in T$. An example is shown in Figure 8.1.

It is not hard to show that the basic solutions of a minimum cost flow problem are precisely its spanning tree solutions.

THEOREM 8.1. *A flow vector is a basic solution of a minimum cost flow problem if and only if it is a spanning tree solution.*

## 8.3 The Network Simplex Method

We will now derive a variant of the simplex method, the network simplex method, that works directly with spanning tree solutions. The network simplex method maintains a feasible solution for the primal and a corresponding dual solution, but unlike the simplex method does not guarantee that these two solutions satisfy complementary slackness. Rather, it uses a separate condition to either establish both dual feasibility and complementary slackness, and thus optimality, or identify a new spanning tree solution.

The Lagrangian of the minimum cost flow problem is

$$\begin{aligned} L(x,\lambda) &= \sum_{(i,j)\in E} c_{ij}x_{ij} - \sum_{i\in V} \lambda_i \Bigg( \sum_{j:(i,j)\in E} x_{ij} - \sum_{j:(j,i)\in E} x_{ji} - b_i \Bigg) \\ &= \sum_{(i,j)\in E} (c_{ij} - \lambda_i + \lambda_j)x_{ij} + \sum_{i\in V} \lambda_i b_i \end{aligned} \tag{8.1}$$

Let $\bar{c}_{ij} = c_{ij} - \lambda_i + \lambda_j$ be the *reduced cost* of edge $(i,j) \in E$. Dual feasibility requires that $\bar{c}_{ij} \geqslant 0$ whenever $\overline{m}_{ij} = \infty$, and holds trivially if all edges are subject to finite capacities. Minimizing $L(x,\lambda)$ subject to the regional constraints $\underline{m}_{ij} \leqslant x_{ij} \leqslant \overline{m}_{ij}$ for $(i,j) \in E$ further yields the following complementary slackness conditions:

$$\begin{aligned} \bar{c}_{ij} > 0 &\quad \text{implies} \quad x_{ij} = \underline{m}_{ij}, \\ \bar{c}_{ij} < 0 &\quad \text{implies} \quad x_{ij} = \overline{m}_{ij}, \text{ and} \\ \underline{m}_{ij} < x_{ij} < \overline{m}_{ij} &\quad \text{implies} \quad \bar{c}_{ij} = 0. \end{aligned}$$

Assume that $x$ is a basic feasible solution associated with sets $T$, $U$, and $L$. Then the system of equations

$$\lambda_{|V|} = 0, \qquad \lambda_i - \lambda_j = c_{ij} \quad \text{for all } (i,j) \in T$$

has a unique solution, which in turn allows us to compute $\bar{c}_{ij}$ for all edges $(i,j) \in E$. Note that $\bar{c}_{ij} = 0$ for all $(i,j) \in T$ by construction, so the third complementary slackness condition is always satisfied.

### Pivoting

If $\bar{c}_{ij} \geqslant 0$ for all $(i,j) \in L$ and $\bar{c}_{ij} \leqslant 0$ for all $(i,j) \in U$, dual feasibility and the first two complementary slackness are satisfied as well, meaning that the solution is optimal. Otherwise, consider an edge $(i,j)$ that violates these conditions, and observe that this edge and the edges in $T$ forms a unique cycle $C$. Since $(i,j)$ is the only edge in $C$ with non-zero reduced cost, we can decrease the objective by pushing flow along $C$ to increase $x_{ij}$ if $\bar{c}_{ij}$ is negative and decrease $x_{ij}$ if $\bar{c}_{ij}$ is positive. Doing so will change the flow on all edges in $C$ by the same amount, with the direction of the change depending on whether a specific edge is oriented in the same or the opposite direction as $(i,j)$.

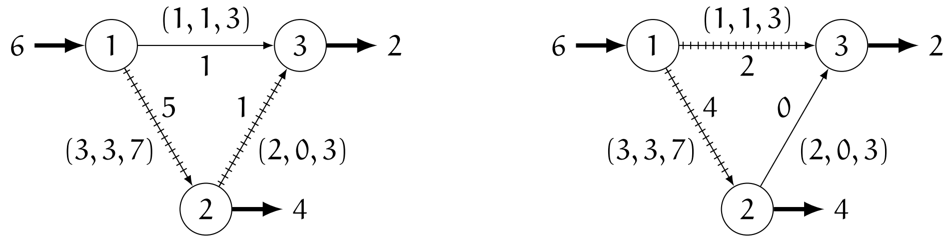


Figure 8.2: Flow network before and after a pivoting step. Edge $(i,j)$ is labeled with the vector $(c_{ij}, \underline{m}_{ij}, \overline{m}_{ij})$ and the current flow $x_{ij}$, and spanning trees are indicated by hatched edges. In the situation shown on the left, we have $\lambda_3 = 0$, $\lambda_2 = c_{23} + \lambda_3 = 2$, and $\lambda_1 = c_{12} + \lambda_2 = 5$, and thus $\bar{c}_{13} = c_{13} - \lambda_1 + \lambda_3 = -4$. If we push one unit of flow around the cycle $1,3,2,1$, the flow on $(2,3)$ reaches the lower bound of $\underline{m}_{23} = 0$ and we obtain a new spanning tree with edges $(1,2)$ and $(1,3)$. The new situation is shown on the right. Now, $\lambda_3 = 0$, $\lambda_1 = c_{13} + \lambda_3 = 1$, and $\lambda_2 = \lambda_1 - c_{12} = -2$, and thus $\bar{c}_{23} = c_{23} - \lambda_2 + \lambda_3 = 4$. Since this is positive and $x_{23} = \underline{m}_{23}$, we have found an optimal solution.

Let $\underline{B}, \overline{B} \subseteq C$ respectively denote the sets of edges whose flow is to decrease or increase, and let

$$\delta = \min\left\{ \min_{(k,\ell)\in\underline{B}} \{x_{k\ell} - \underline{m}_{k\ell}\}, \min_{(k,\ell)\in\overline{B}} \{\overline{m}_{k\ell} - x_{k\ell}\} \right\}.$$

be the maximum amount of flow that can be pushed along $C$. If $\delta = \infty$, the problem is unbounded. If $\delta = 0$ or if the minimum is attained for more than one edge, the problem is degenerate. Otherwise, pushing $\delta$ units of flow along $C$ yields a unique edge $(k,\ell) \in C$ whose flow is either $\underline{m}_{k\ell}$ or $\overline{m}_{k\ell}$. If $(k,\ell) \in T$, we obtain a new BFS with spanning tree $(T \setminus \{(k,\ell)\}) \cup \{(i,j)\}$. If instead $(k,\ell) = (i,j)$, we obtain a new BFS where $(i,j)$ has moved from $U$ to $L$, or vice versa. An example of the pivoting step is given in Figure 8.2.

In the absence of degeneracies the value of the objective function decreases in every iteration of the network simplex method, and an optimal solution or a certificate of unboundedness is found after a finite number of iterations. If a degenerate solution is encountered it will still be possible to identify a new spanning tree or even a new BFS, but extra care may be required to ensure convergence. This is beyond the scope of this course.

## Finding an initial feasible spanning tree solution

Consider a minimum cost flow problem for a network $(V,E)$ and assume without loss of generality that $\underline{m}_{ij} = 0$ for all $(i,j) \in E$. If this is not the case, we can instead consider the problem obtained by setting $\underline{m}_{ij}$ to zero, $\overline{m}_{ij}$ to $\overline{m}_{ij} - \underline{m}_{ij}$, and replacing $b_i$ by $b_i - \underline{m}_{ij}$ and $b_j$ by $b_j + \underline{m}_{ij}$. A solution with flows $x_{ij}$ for the new problem then corresponds to a solution with flows $x_{ij} + \underline{m}_{ij}$ for the original problem.

We now modify the problem such that the set of optimal solutions remains the same, assuming that the problem was feasible, but an initial feasible spanning tree solution is easy to find. For this, we introduce a dummy vertex $d \notin V$ and uncapacitated dummy edges $E' = \{(i,d) : i \in V, b_i \geqslant 0\} \cup \{(d,i) : i \in V, b_i < 0\}$ with cost equal to $\sum_{(i,j)\in E} c_{ij}$. It is easy to see that a dummy edge has positive flow in some optimal solution of the new problem if and only if the original problem is infeasible. Furthermore, a feasible spanning tree solution is now easily obtained by letting $T = E'$, $x_{id} = b_i$ for all $i \in V$ with $b_i > 0$, $x_{di} = -b_i$ for all $i \in V$ with $b_i < 0$, and $x_{ij} = 0$ otherwise.

## 8.4 Integrality of Optimal Solutions

Since the network simplex method does not require any divisions, any finite optimal solution it obtains for a problem with integer constants is also integral.

THEOREM 8.2. *Consider a minimum cost flow problem that is feasible and bounded. If* $b_i$ *is integral for all* $i \in V$ *and* $\underline{m}_{ij}$ *and* $\overline{m}_{ij}$ *are integral for all* $(i,j) \in E$*, then there exists an integral optimal solution. If* $c_{ij}$ *is integral for all* $(i,j) \in E$*, then there exists an integral optimal solution to the dual.*