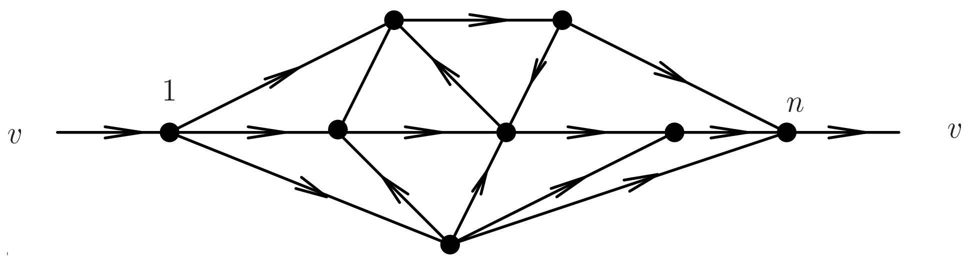
10 Maximal Flow in a Network

10.1 Max-flow/min-cut theory

Consider a network consisting of n nodes labelled $1, \dots, n$ and directed edges between them with capacities c_{ij} on the arc from node i to node j. Let x_{ij} denote the flow in the arc $i \to j$, where $0 \le x_{ij} \le c_{ij}$.



Problem: Find maximal flow from node 1 to node n subject to the conservation of flow at nodes, i.e.,

$$\text{maximize } v \quad \text{s.t. } 0 \le x_{ij} \le c_{ij},$$

and

$$\boxed{\begin{array}{c}\text{flow out of}\\ \text{node } i\end{array}} - \boxed{\begin{array}{c}\text{flow into}\\ \text{node } i\end{array}} = \sum_j x_{ij} - \sum_j x_{ji} = \begin{array}{rcl} v & & i = 1 \\ 0 & \text{as} & i \neq 1, n \\ -v & & i = n \end{array}$$

where the summations are understood to be over existing arcs only. v is known as the **value of the flow**.

This is obviously an LP problem, but with lots of variables and constraints. We can solve it more quickly (taking advantage of the special network structure) as follows.

Definition 10.1 *A* **cut** $(S, \bar{S})$ *is a partition of the nodes into two disjoint subsets S and $\bar{S}$ with $1 \in S$ and $n \in \bar{S}$.*

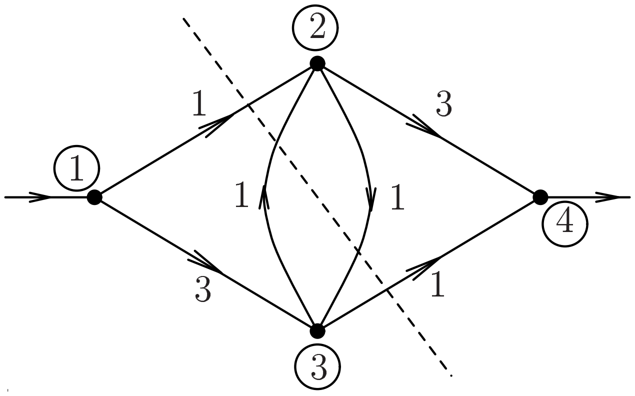
Definition 10.2 *The* **capacity** *of a cut*

$$C(S, \bar{S}) = \sum_{i \in S, j \in \bar{S}} c_{ij}.$$

Thus given a cut $(S, \bar{S})$ the capacity of the cut is the maximal flow from nodes in S to nodes in $\bar{S}$. It is intuitively clear that any flow from node 1 to node n must cross the cut $(S, \bar{S})$, since in getting from 1 to n at some stage it must cross from S to $\bar{S}$. This holds for any flow and any cut. In fact, we have:

Theorem 10.1 (Max flow/min cut Theorem) *The maximal flow value through the network is equal to the minimal cut capacity.*

Example.



Cut $S = \{1, 3\}$, $\bar{S} = \{2, 4\}$, $C(S, \bar{S}) = 3$. Check that the maximal flow is 3.

Proof. Let us define $f(X, Y) = \sum_{i \in X, j \in Y} x_{ij}$, the flow from X to Y. We will use $N = \{1, \ldots, n\}$ to represent the set of nodes in the network. Let $(S, \bar{S})$ be a cut and $\{x_{ij}\}$ a feasible flow with value v. Note that $f(X, N) = f(X, S) + f(X, \bar{S})$. Since x_{ij} is feasible, we have

$$\sum_j x_{ij} - \sum_j x_{ji} = \begin{array}{rcl} v & & i = 1 \\ 0 & \text{as} & i \neq 1, n \\ -v & & i = n \end{array} .$$

Summing this equality over i in S we obtain

$$\begin{aligned} v &= \sum_{i \in S} \sum_j (x_{ij} - x_{ji}) \\ &= \sum_{i \in S} \sum_j x_{ij} - \sum_{i \in S} \sum_j x_{ji} \\ &= f(S, N) - f(N, S) \\ &= f(S, S) + f(S, \bar{S}) - f(\bar{S}, S) - f(S, S) \\ &= f(S, \bar{S}) - f(\bar{S}, S) \\ &\leq f(S, \bar{S}) \\ &\leq C(S, \bar{S}) \end{aligned}$$

So any flow $\leq$ any cut capacity, (and in particular max flow $\leq$ min cut).

Now let f be a maximal flow, and define $S \subseteq N$ recursively as follows:

(1) $1 \in S$.
(2) If $i \in S$ and $x_{ij} < c_{ij}$, then $j \in S$.
(3) If $i \in S$ and $x_{ji} > 0$, then $j \in S$.
Keep applying (2) and (3) until no more can be added to S.

So S is the set of nodes to which we can increase flow. Now if $n \in S$ we can increase flow along a path in S and f is not maximal. So $n \in \bar{S} = N \setminus S$ and $(S, \bar{S})$ is a cut. From the definition of S we know that for $i \in S$ and $j \in \bar{S}$, $x_{ij} = c_{ij}$ and $x_{ji} = 0$, so in the formula above we get

$$v = f(S, \bar{S}) - f(\bar{S}, S) = f(S, \bar{S}) = C(S, \bar{S}).$$

So max flow = min cut capacity. ■

Corollary 10.2 *If a flow value v = cut capacity C then v is maximal and C minimal.*

The proof suggests an algorithm for finding the maximal flow.

10.2 Ford-Fulkerson algorithm

1. Start with a feasible flow (e.g., $x_{ij} = 0$).

2. Construct S recursively by the algorithm defined in the box above.

3. If $n \in S$ then there is a path from 1 to n along which we can increase flow by

$$\epsilon = \min_{(ij)} \max[x_{ji}, c_{ij} - x_{ij}] > 0.$$

   where the minimum is taken with respect to all arcs $i \to j$ on the path.

   Replace the flow by this increased flow. Return to 2.

   If $n \notin S$ then the flow is optimal.

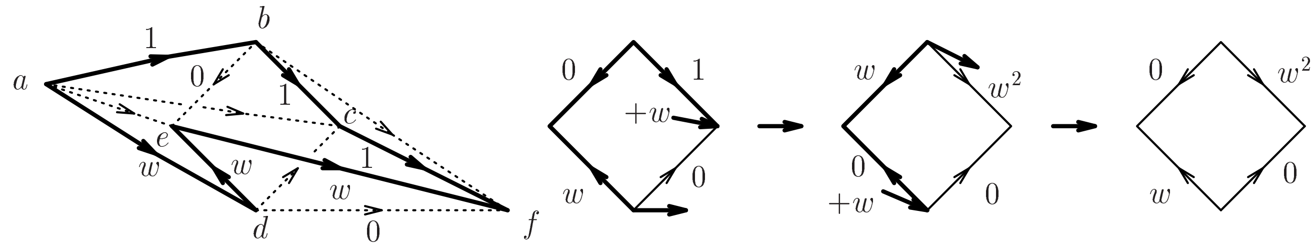
The algorithm is crude and simple; we just push flow through where we can, until we can't do so anymore. There is no guarantee that it will be very efficient. With hand examples it is usually easy to 'see' the maximal flow. You just demonstrate that it is maximal by giving a cut with the same capacity as the flow and appeal to the min cut = max flow theorem.

The algorithm can be made not to converge if the capacities are not rational multiples of one another. However,

Theorem 10.3 *If capacities and initial flows are rational then the algorithm terminates at a maximal flow in a finite number of steps. (Capacities are assumed to be finite.)*

Proof. Multiply by a constant so that all capacities and initial flows are integers. The algorithm increases flow by at least 1 on each step. ■

Example: Failure to stop when capacities and initial flows are not rational



The network consists of a square $b\,c\,d\,e$ of directed arcs of capacity 1. The corners of the square are connected to a source at a and a sink at f by arcs of capacity 10. The initial flow of $1+w$ is shown in the first picture, where $w = (\sqrt{5}-1)/2$, so $1-w = w^2$. The first iteration is to increase flow by w along $a \to c \to b \to e \to d \to f$. The second increases it by w along $a \to d \to e \to b \to f$. The flow has increased by $2w$ and the resulting flow in the square is the same as at the start, but multiplied by w and rotated through 180^o. Hence the algorithm can continue in this manner forever without stopping and never reach the optimal flow of 40.

10.3 Minimal cost circulations

Definition 10.3 *A network is* **closed** *if there is no flow into or out of the network.*

Definition 10.4 *A flow in a closed network is a* **circulation** *if* $\sum_j x_{ij} - \sum_j x_{ji} = 0$ *for each node* i.

Most network problems can be formulated as the problem of finding a **minimal cost circulation** in a closed network where there are capacity constraints $c_{ij}^- \le x_{ij} \le c_{ij}^+$ on arcs (i,j) and a cost per unit flow of d_{ij} in arcs (i,j). The full problem is

$$\text{minimize} \sum_{ij} d_{ij} x_{ij}$$

$$\text{subject to} \sum_j x_{ij} - \sum_j x_{ji} = 0, \text{ each } i, \text{ and } c_{ij}^- \le x_{ij} \le c_{ij}^+.$$

Definition 10.5 *A circulation which satisfies the capacity constraints is called a* **feasible circulation**.

There is a beautiful algorithm called the out-of-kilter algorithm which will solve general problems of this kind. It does not even require a feasible solution with which to start. (See Part IIA 'Algorithms and Networks'.) In the next lecture we shall just derive conditions for a flow to be optimal.

We shall also see, although it should be obvious already, that the max flow problem that is studied in this lecture can be formulated as a minimal cost circulation problem.