Network Flow Problem
(Max-flow problem)

G, a digraph with n nodes and m arcs.
Each arc (i,j) has a maximum capacity c_{ij}.
Problem: find a maximum flow from s to t.

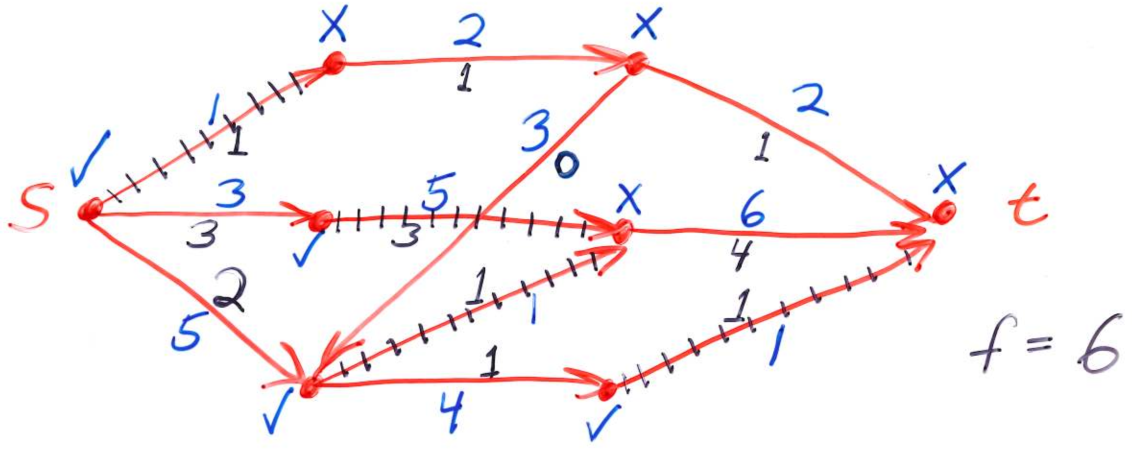
- The flow x_{ij} on an edge (i,j) must be $\le c_{ij}$.
- For every node v (other than s or t), flow is conserved, i.e. the sum of the flows coming into v is equal to the sum of the flows going out of v.

The value of the flow is the sum of all the flows going out of s
(= the sum of all the flows going into t).

max f

s.t.
$$\sum_{j=1}^{n} x_{ij} - \sum_{k=1}^{n} x_{ki} = \begin{cases} f & \text{if } i=1 \quad s \\ 0 & \text{if } i \neq 1 \text{ or } n \\ -f & \text{if } i=n \quad t \end{cases}$$

$$x_{ij} \le c_{ij}$$
$$x_{ij} \ge 0$$



capacities
flows

A cut is a subset of the edges such that removing them leaves no directed path from s to t.

A partition of the vertices $(P, \bar{P})$ with $s \in P$ and $t \in \bar{P}$ defines a cut — it contains all edges from vertices in P to vertices in $\bar{P}$.

$K(P, \bar{P})$ = capacity of the cut $(P, \bar{P})$

$$= \sum_{\substack{i \in P \\ j \in \bar{P}}} c_{ij}$$

The flow, $F(N) = \sum_j x_{sj} - \sum_i x_{is}$

Thm For any cut $(P, \bar{P})$, the flow is

$$F(N) = \sum_{\substack{i \in P \\ j \in \bar{P}}} x_{ij} - \sum_{\substack{i \in \bar{P} \\ j \in P}} x_{ij}$$

Pf

$$F(N) = \sum_{i \in P} \left(\sum_j x_{ij} - \sum_j x_{ji} \right)$$ (0 contribution from all but s)

$$= \left(\cancel{\sum_{\substack{i \in P \\ j \in P}} x_{ij}} + \sum_{\substack{i \in P \\ j \in \bar{P}}} x_{ij} \right) - \left(\cancel{\sum_{\substack{i \in P \\ j \in P}} x_{ji}} + \sum_{\substack{i \in P \\ j \in \bar{P}}} x_{ji} \right)$$

$$= \sum_{\substack{i \in P \\ j \in \bar{P}}} x_{ij} - \sum_{\substack{i \in P \\ j \in \bar{P}}} x_{ji}$$ ■

Corollary For any cut $(P, \bar{P})$ and any flow $F(N)$, $K(P, \bar{P}) \geq$ the value of $F(N)$.

Pf $$K(P, \bar{P}) = \sum_{\substack{i \in P \\ j \in \bar{P}}} c_{ij} \geq \sum_{\substack{i \in P \\ j \in \bar{P}}} x_{ij}$$

$$\geq \sum_{\substack{i \in P \\ j \in \bar{P}}} x_{ij} - \sum_{\substack{i \in \bar{P} \\ j \in P}} x_{ij}$$

$$= F(N)$$ ■

(i,j) is a forward edge $\Rightarrow$

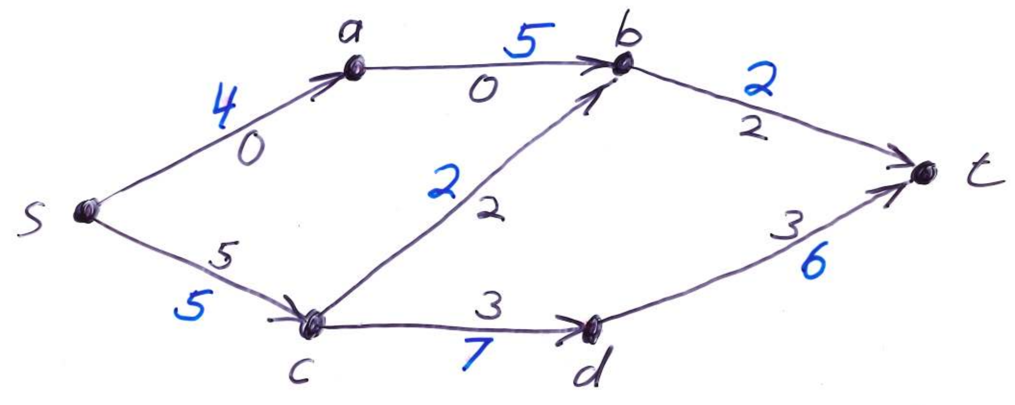
(i,j) is a reverse edge $\Rightarrow$

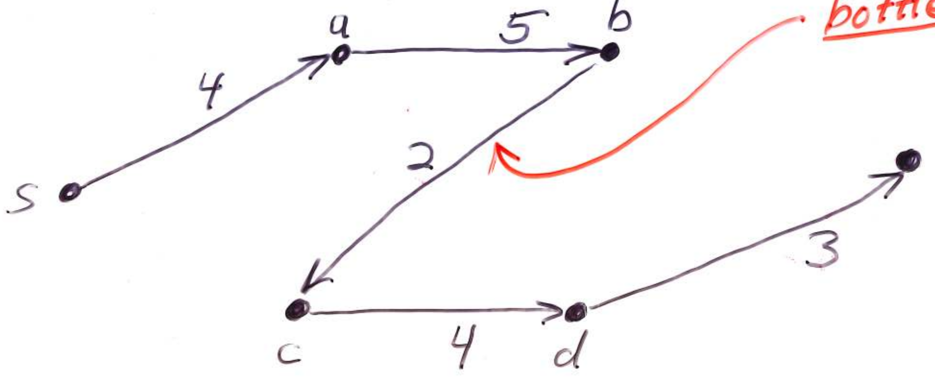
Def Given a flow $F(N)$, an augmenting path is a path $Q = (s = v_0, v_1, v_2, \ldots, v_k = t)$ in the underlyind undirected graph s.t.

a.) if (v_i, v_{i+1}) is a forward edge, then $\Delta_i = c_{v_i, v_{i+1}} - x_{v_i, v_{i+1}} > 0$

b.) if (v_i, v_{i+1}) is a reverse edge, then $\Delta_i = x_{v_{i+1}, v_i} > 0$.

$\Delta = \min \Delta_i \qquad \Delta > 0$

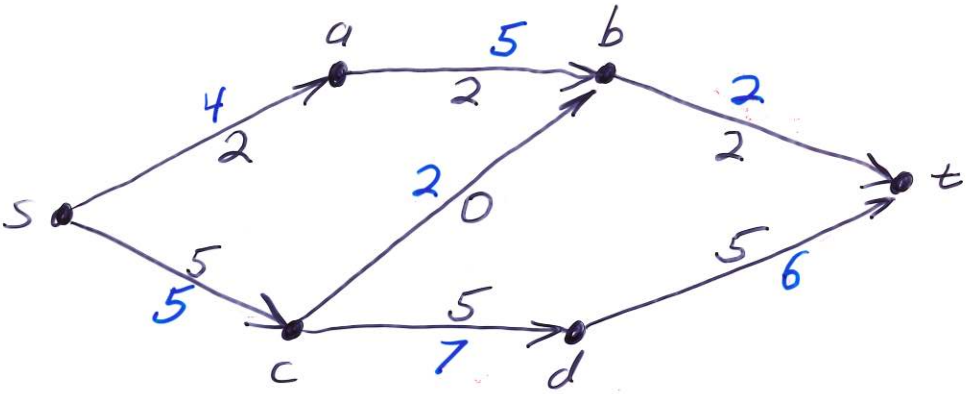




bottleneck edge

$\Delta_i = \Delta$

$\Delta = 2$



Now there are no more augmenting paths, so this is a maximum flow.

Given an augmenting path, we can augment the flow by

a.) if (v_i, v_{i+1}) is a forward edge, then
$$x_{v_i, v_{i+1}} \leftarrow x_{v_i, v_{i+1}} + \Delta$$

b.) if (v_i, v_{i+1}) is a reverse edge, then
$$x_{v_i, v_{i+1}} \leftarrow x_{v_i, v_{i+1}} - \Delta$$

(Preserves conservation of flow.)

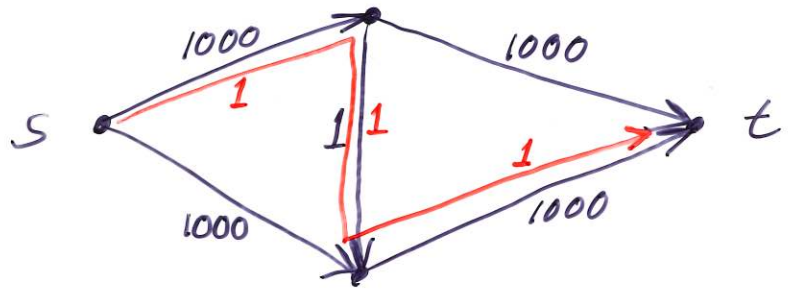
Ford-Fulkerson Algorithm

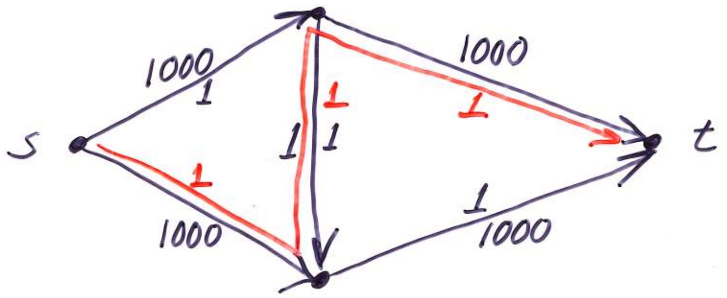
Keep finding augmenting paths (augmenting the flow by the Δ for that path) until none exists.

If the capacities are integer, it is easy to see that it eventually terminates.

- The flow $\leq$ the sum of the capacities on edges leaving s
- The flow gets incremented by ≥ 1 each time.

It can take a long time!





Doing it this way requires 2000 augmentations!

Thm If no augmenting path exists for some $F(N)$, then the value of $F(N)$ is a maximum.

Pf Let's label vertices "reachable" from s.

a.) s is labeled.

b.) If for $(u,v) \in E$, u is labeled and v is unlabeled, then if $x_{u,v} < c_{u,v}$, label v.

c.) If for $(u,v) \in E$, v is labeled and u is unlabeled, then if $x_{u,v} > 0$, label u.

If $\nexists$ an augmenting path, t is not labeled. Let P be the labeled vertices.

$x_{u,v} = c_{u,v}$ if $u \in P$ and $v \in \bar{P}$

$x_{u,v} = 0$ if $u \in \bar{P}$ and $v \in P$

$$\text{Thus } F(N) = \sum_{\substack{u \in P \\ v \in \bar{P}}} x_{u,v} - \sum_{\substack{u \in \bar{P} \\ v \in P}} x_{u,v}$$

$$= \sum_{\substack{u \in P \\ v \in \bar{P}}} c_{u,v} = K(P, \bar{P})$$

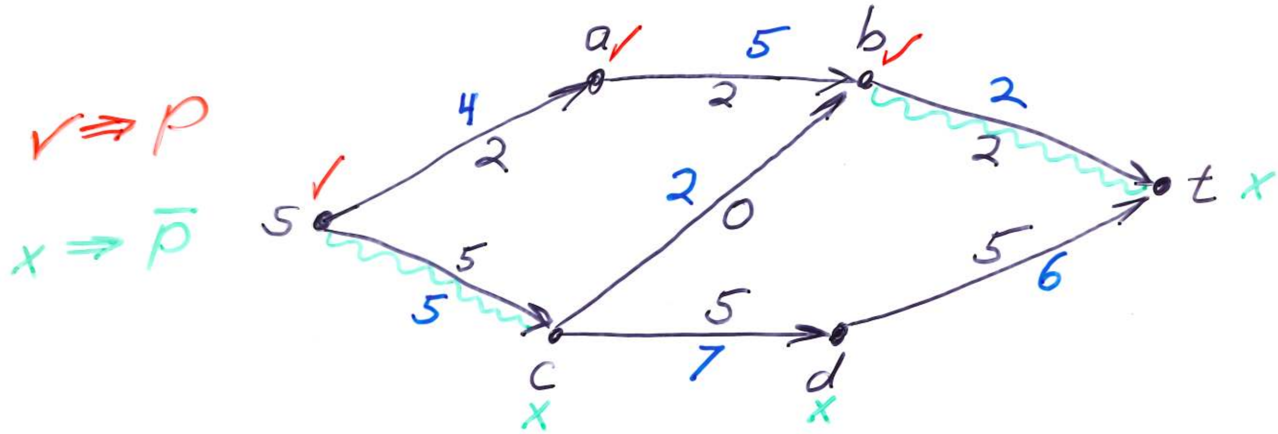
So $F(N)$ is maximum.

Note that $(P, \bar{P})$ must be a cut of minimum capacity, or the value of $F(N)$ would exceed the capacity of some cut.

∎

Thm (Max-flow, Min-cut) For a given network, the maximum possible flow is equal to the minimum capacity of all cuts, i.e. $\max F(N) = \min K(P, \bar{P})$.

Note This can also be proven using duality.



$= (P, \bar{P})$

<u>Integrality Thm</u> If the capacities are integer, then $\exists$ a max flow s.t. $\forall (i,j)$ x_{ij} is integer.

<u>Pf</u> By the algorithm - it produces a max flow with integer values.

Or by the fact that the matrix for the linear program is TUM. $\blacksquare$

Edmonds and Karp showed that if you use breadth-first-search (BFS) to find the shortest path and always choose the shortest path, the paths never get shorter, so there are at most $|V|\cdot|E|$ paths chosen and the algorithm takes $O(|V|^3 \cdot |E|)$. (because if we do BFS with an adjacency matrix, it takes $O(|V|^2)$).

<u>Thm</u> (Edmonds and Karp) If each flow augmentation is made along an augmenting path with a minimum number of arcs, then a maximal flow is obtained after no more than $mn/2 \leq (n^3-n^2)/2$ augmentations, where m is the number of arcs in the network and n is the number of nodes.

$\sigma_i^{(k)}$ = min # of arcs in an augmenting path from s to i after k flow augmentations

$\tau_i^{(k)}$ = min # of arcs in an augmenting path from i to t after k flow augmentations

Lemma If each augmentation is made along an augmenting path with min # of arcs, then

$$\sigma_i^{(k+1)} \geq \sigma_i^{(k)} \qquad \forall i, k.$$

and $\tau_i^{(k+1)} \geq \tau_i^{(k)}$

Pf Assume that $\sigma_i^{(k+1)} < \sigma_i^{(k)}$ for some i, k. Let $\sigma_i^{k+1} = \min\{\sigma_j^{(k+1)} \mid \sigma_j^{(k+1)} < \sigma_j^{(k)}\}$. Look at the last arc, (i,j) or (j,i) in a shortest augmenting path from s to i. Suppose it's a forward arc with $x_{ji} < c_{ji}$.

$\sigma_i^{(k+1)} = \sigma_j^{(k+1)} + 1$ and
$\sigma_i^{(k+1)} \geq \sigma_j^{(k)} + 1$. Thus $x_{ji} = c_{ji}$ after the k^{th} augmentation; otherwise $\sigma_i^{(k)} \leq \sigma_j^{(k)} + 1 \leq \sigma_i^{(k+1)}$. So (j,i) was a reverse edge in the $k+1^{st}$ flow augmenting path. That path contained a minimum number of arcs, so $\sigma_j^{(k)} = \sigma_i^{(k)} + 1$. Since $\sigma_j^{(k)} + 1 \leq \sigma_i^{(k+1)}$, $\sigma_i^{(k)} + 2 \leq \sigma_i^{(k+1)}$

$\therefore \sigma_i^{(k)} \leq \sigma_i^{(k+1)}$

The proofs for (i,j) a reverse edge and for $\tau_i^{(k+1)} \geq \tau_i^{(k)}$ are similar. ∎

Pf of Thm Each time an augmentation is made, there is at least one bottleneck edge. The flow through this edge (i,j) is either increased to capacity or decreased to zero. The next time this path is in an augmenting path, it must be with the opposite orientation.

Assume it was forward in the $k+1^{st}$ augmentation.

$$\sigma_j^{(k)} = \sigma_i^{(k)} + 1$$

Suppose that next time it appears is in the $l+1^{st}$. Then it's a reverse edge.

$$\sigma_i^{(l)} = \sigma_j^{(l)} + 1$$

By the lemma, $\sigma_j^{(l)} \geq \sigma_j^{(k)}$ and $\tau_i^{(l)} \geq \tau_{i}^{(k)}$
so $\sigma_i^{(l)} + \tau_i^{(l)} \geq \sigma_i^{(k)} + \tau_i^{(k)} + 2$.

So each path in which (i,j) is a bottleneck edge is at least 2 longer than the previous one.

Since all paths have length $\leq n-1$, no edge is bottleneck more than $n/2$ times.

$\therefore$ There are $\leq mn/2$ flow augmentations $\blacksquare$