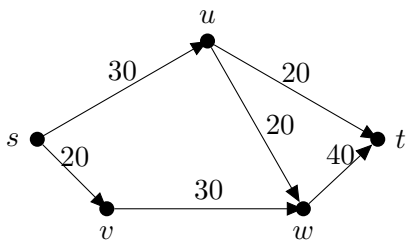
Not in Textbook - Network Flows

Original application (product of cold war)

Suppose country X enters into a war. How quickly can X move tanks from storage s to the target t (battle ground)? The tanks are moved on a railroad. Every link gives the capacity how many tank a day can be transported.



1: How many tanks per day can be delivered to the battleground? Is the solution unique?

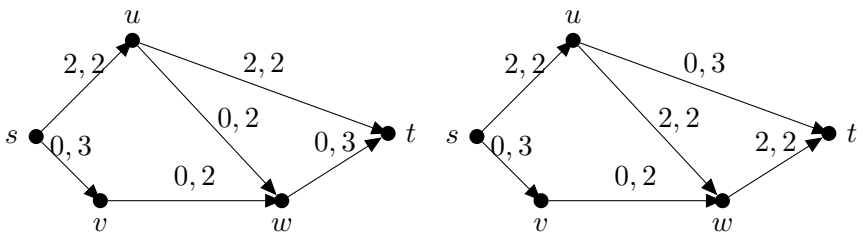
Problem: (Directed) graph G, source s, sink t, capacities $c : E(G) \to \mathbb{R}^+$.
Network is (G, c, s, t).

Input: Network (G, c, s, t)
Output: s-t-flow of maximum value

s-t-flow f is a function $f : E(G) \to \mathbb{R}^+$
Value of f is $\sum_{su \in E} f(su) - \sum_{us \in E} f(us)$ i.e. leaving $-$ entering to s.

2: How f looks around one vertex of the network? (what must f satisfy?)

3: How to improve these flows to be maximum? Description on edges are values of f, c.



4: After the improvement, how do you argue that nobody can further improve?

Let $A \subset V(G)$ such that $s \in A$ and $t \notin A$. Use $\delta^+(A)$ to denote set of edges uv, where $u \in A$ and $v \notin A$ (edges leaving A). Use $\delta^-(A)$ to denote set of edges uv, where $u \notin A$ and $v \in A$ (edges entering A).
Capacity of s-t-cut A is $\sum_{e \in \delta^+(A)} c(e)$.

5: Prove that for A and any flow f holds
(a) value$(f) = \sum_{e \in \delta^+(A)} f(e) - \sum_{e \in \delta^-(A)} f(e)$
(b) value$(f) \leq \sum_{e \in \delta^+(A)} c(e)$

This proves the *obvious* observation that maximum flow cannot exceed capacity of minimum cut.

Notice in 3. we were improving flow be reducing the flow on uw. We "sent flow in the opposite direction".

For a digraph G, define $\overleftrightarrow{G}$ by adding for every edge e also its **reverse** $\overleftarrow{e}$.

For f and c define **residual capacities** $c_f : E(\overleftrightarrow{G}) \to \mathbb{R}_+$

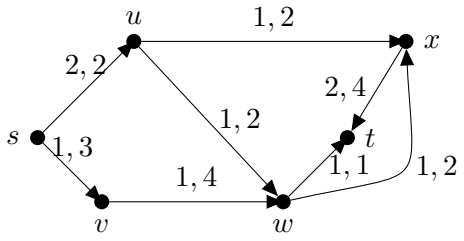
$$c_f(e) = c(e) - f(e) \qquad\qquad c_f(\overleftarrow{e}) = f(e)$$

Residual capacities ... how much extra we can send in each direction.

Residual graph G_f is obtained from $\overleftrightarrow{G}$ by removing edges $e \in E(\overleftrightarrow{G})$ with $c_f(e) = 0$.

Augmenting path is an s-t path in G_f.

6: Construct the residual graph for



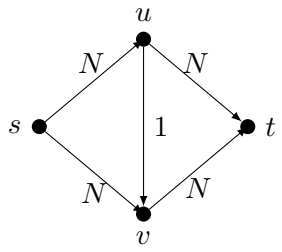
and find an augmenting path and increase the flow using the augmenting path.

7: How to update (to **augment** the flow f using augmenting path in $\overleftrightarrow{G}$?

Ford-Fulkerson Algorithm
Input: Network (G, c, s, t).
Output: and s-t-flow f of maximum value

1. $f(e) = 0$ for all $e \in E(G)$
2. while f-augmenting path P exists:
3. compute $\gamma := \min_{e \in E(P)} c_f(e)$
4. augment f along P by γ (as much as possible)

8: How many iterations (at most) the algorithm needs at the following network:



(N is a big integer. Try to trick the algorithm to do many steps by picking P.)

9: Show that s-t-flow f is maximum if and only if there is no f-augmenting path. (That is, Ford-Fulkerson algorithm is correct.)

The question gives proof to
Theorem (Ford Fulkerson) Maximum value of an s-t-flow equals minimum capacity of an s-t-cut.

10: If $c : E \to \mathbb{Z}$, is it true that the flow produced from Ford-Fulekrson is integral and that the algorithm finishes in a finite time?

This proves
Theorem Dantzig Fulkerson: If the capacities are integral, then there exists an integral maximum flow.

11: If capacities are integral, is it true that every maximum flow is integral?

Flows in graphs

Given a weighted directed graph (i.e., each directed edge has a value assigned which we think of as capacity and denote by c) with one vertex identified as the sink (s) and a second vertex identified as the source (t), a *flow*, denoted f, is an assignment of values to the edges satisfying:

1. The assignment of each edge is bounded above by the weight of each edge.
2. For each vertex other than the source and the sink, the sum of the assignments for the edges coming in to a vertex is equal to the sum of the assignments of the edges coming out from that same vertex.

The total flow is equal to the net outflow from the source, or equivalently the net inflow to the sink. We can bound the flow by looking at a set of edges which separates the source and the sink (i.e., so there is no directed path from the source to the sink); the total sum of the edge weights of any such separating set is an upper bound for the flow. In fact, more is true.

Max Flow-Min Cut. *The maximum flow in a network equals the minimum capacity of a cut.*

To find an optimal flow we look for an *augmenting path*. This is a path from the sink to the source such that for forward edges we have that $f(e) < c(e)$ (the current flow is less than the capacity of the edge, i.e., it is not being fully utilized) and for backward edges we have $f(e) > 0$ (i.e., the edge is currently being used for flow). Then by adding this path into the network we can increase the flow (the amount of flow that we can increase by is the minimum for edges in the augmenting path of the difference $c(e) - f(e)$ for forward edges and $f(e)$ for backward edges). If there is no augmenting path then the flow is maximal.

Algorithm (Ford-Fulkerson (1962))**.** *Given a flow f in a network find either an augmenting path or a cut showing that the flow is maximal. This is done by starting at the source and looking for all vertices which can be reached by an augmenting path. Either we get to the sink and have some path that augments or we get stuck in which case we have our desired cut.*

Start with $R = \{s\}$ (vertices we have reached) and $S = \emptyset$ (vertices we have searched).

Iteratively do the following:

- *Pick a vertex $v \in R - S$.*
- *For each edge $v \to w$ with $f(vw) < c(vw)$ and $w \notin R$, add w to R.*
- *For each edge $w \to v$ with $0 < f(vw)$ and $w \notin R$, add w to R.*
- *Add v to S.*
- *If $t \in R$ then there is an augmenting path (this can be found by tracing how edges were added to S), and we are done. If $R = S$, then S and $\overline{S}$ is our desired cut, and we are done. Otherwise, continue iterating.*

We note in the special case that the capacities are all integral that the maximum flow can be taken as integral on each edge. This is because in taking the augmenting paths we can always take an integral value when finding the augmenting path.

By Steve Butler