Lecture #12:

0.0.1 Graph Algorithms: Maximum Flow (Chapter 26)

Given a directed graph $G = (V, E)$ in which each edge $(u, v) \in E$ has a **capacity** $c(u, v) \geq 0$. If $(u, v) \notin E$ we assume that $c(u, v) = 0$. We also distinguish two vertices s (origin = source) and t (destination = sink). A **flow** in G is a function $f : V \times V \longrightarrow \mathbf{R}$ that satisfies the following three properties:

Capacity Constraint: $f(u, v) \leq c(u, v) \forall u, v \in V$

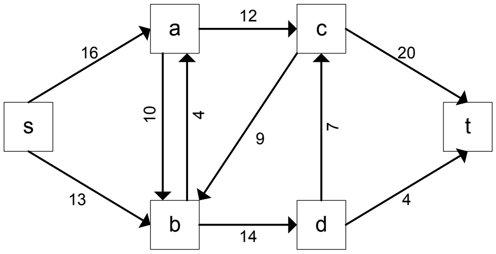
Skew symmetry: $f(v, u) = -f(u, v) \forall u, v \in V$

Flow conservation: $\sum_{v \in V} f(u, v) = 0 \forall u \in V - \{s, t\}$

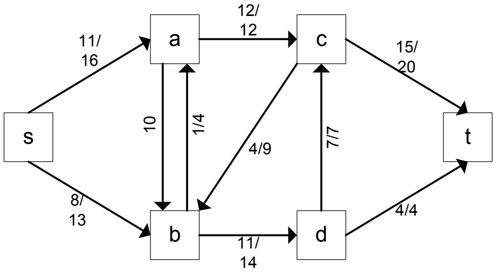
$f(u, v)$ is called the net flow from u to v. The value of f denoted by $|f|$ is equal to $\sum_{v \in V} f(s, v)$.

In the **maximum flow problem**, we want the flow f with maximum $|f|$.

For example, consider the following problem:

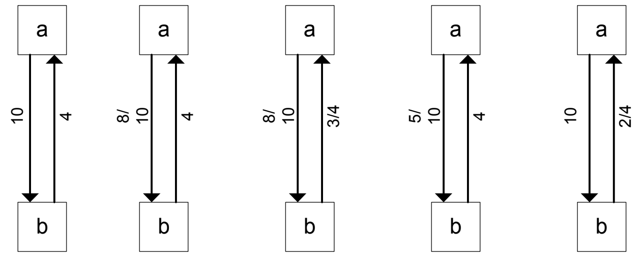


The following is a flow:



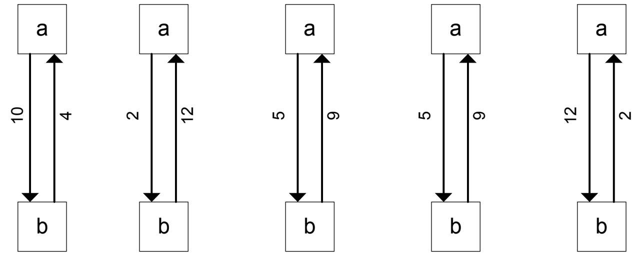
The value of this flow is 19.

Cancellation: Consider the following diagram:

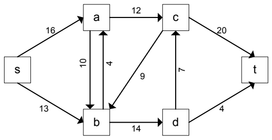


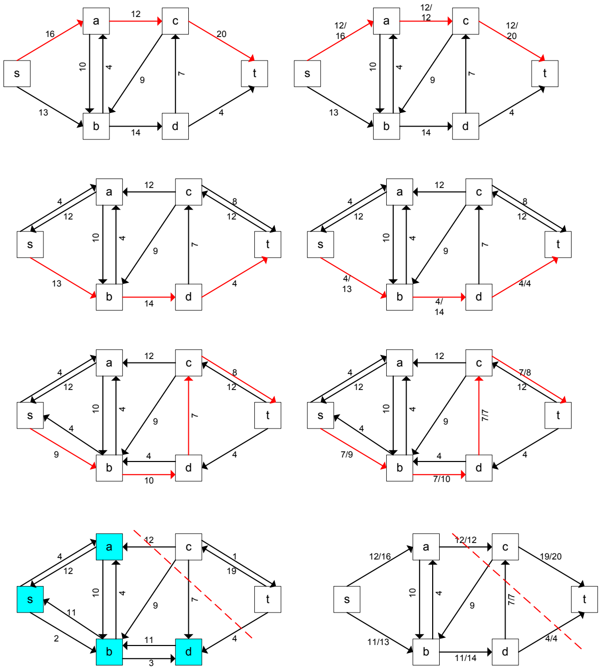
In the first diagram, we have the capacities; the second has a flow of 8 from a to b; the third has an additional flow of 3 from b to a; cancelling the flow in both directions gives us the next diagram with 5 from a to b, and none in the other direction. Finally, an additional flow of 7 from b to a, cancels the previous flow and introduces a flow of 2 from b to a.

Because of this, we define the concept of **residual capacity** as follows: $c_f(u,v) = c(u,v) - f(u,v)$. The residual capacity for the above case would look like:



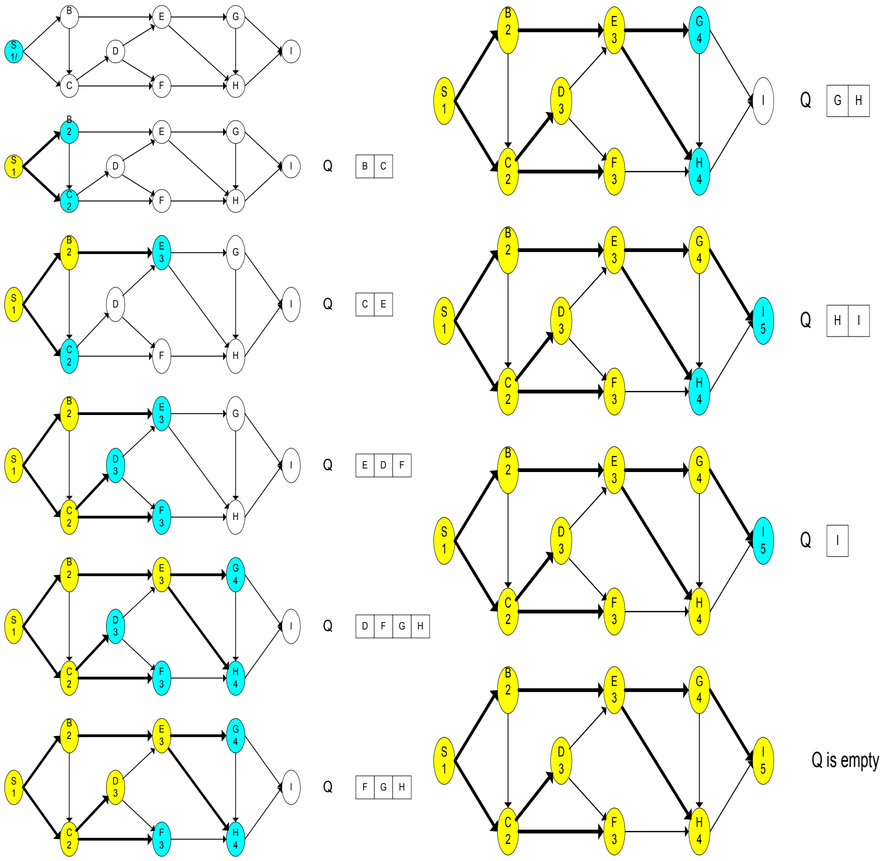
And a residual network as follows: Given a flow network $G = (V,E)$, $c : E \longrightarrow \mathbf{R}_+$, and a flow f , the residual network $G_f = (V, E_f)$ where $E_f = \{(u,v) \in V \times V : c_f(u,v) > 0\}$. **Please note that E_f may contain edges not present in E.** The main step of the algorithm is to find a directed path from s to t in G_f called an **augmenting path** and **saturate** it. To saturate a path P, we add a flow equal to $\min_{(u,v)\in P}[c_f(u,v)]$ along the path. This step is repeated until there is no such path. Please note that after any flow change, you must redefine the residual network. We show all this by an example below:





Residual networks are on the left column and flows on the right. The last step does not have a path in the residual network. The solution at this time is optimal.

The path is the residual network is found using BFS and BFS is shown below:



Proofs:
Let $f(X,Y) = \sum_{\substack{u\in X\\ v\in Y}}[f(u,v)]$. Then it is easy to show:

$$\begin{aligned}
f(X,X) &= 0 \forall X \subseteq V \\
f(X,Y) &= -f(Y,X) \forall X,Y \subseteq V \\
f(X\cup Y, Z) &= f(X,Z) + f(Y,Z) \forall X,Y,Z \subseteq V; X\cap Y = \phi \\
f(Z, X\cup Y) &= f(Z,X) + f(Z,Y) \forall X,Y,Z \subseteq V; X\cap Y = \phi
\end{aligned}$$

Using this it is easy to show that

$$|f| = f(s,V) = f(V,t)$$

Lemma 1 *Let f be a flow in $G = [V,E]$ from s to t. Let f' be a flow in the residual network G_f induced by f. Then $f + f'$ is a flow in G with a value $|f + f'| = |f| + |f'|$*

Proof.

$$\begin{aligned} (f+f')(u,v) &= f(u,v)+f'(u,v) \\ &= -f(v,u)-f'(v,u) \\ &= -(f(v,u)+f'(v,u)) \\ &= -(f+f')(u,v) \end{aligned}$$

$$\begin{aligned} (f+f')(u,v) &= f(u,v)+f'(u,v) \\ &\le f(u,v)+[c_f(u,v)-f(u,v)] \\ &= c(u,v) \end{aligned}$$

since $f'(u,v) \le c_f(u,v) = c(u,v) - f(u,v)$. For all $u \in V - \{s\} - \{t\}$

$$\begin{aligned} \sum_{v\in V}(f+f')(u,v) &= \sum_{v\in V} f(u,v) + \sum_{v\in V} f'(u,v) \\ &= 0+0=0 \end{aligned}$$

Finally

$$\begin{aligned} |f+f'| &= \sum_{v\in V}(f+f')(s,v) \\ &= \sum_{v\in V} f(s,v) + \sum_{v\in V} f'(s,v) \\ &= |f|+|f'| \end{aligned}$$

■

A cut $(S, V-S)$ separates s and t if $s \in S$ and $t \in V-S$. If we let $T = V-S$, the capacity of the cut (S,T) is $c(S,T)$. Recall by our notation

$$c(S,T) = \sum_{\substack{u\in S \\ v\in T}} c(u,v)$$

Lemma 2 *For any cut (S,T) and any flow f, we have $f(S,T) = |f|$*

Proof.

$$\begin{aligned} f(S,T) &= f(S,V) - f(S,S) \\ &= f(S,V) \\ &= f(s,V) + f(S-s,V) \\ &= f(s,V) \\ &= |f| \end{aligned}$$

■

Lemma 3 *For any flow f and any cut (S,T), $|f| \le c(S,T)$*

Proof.

$$\begin{aligned} |f| &= f(S,T) = \sum_{\substack{u \in S \\ v \in T}} f(u,v) \\ &\leq \sum_{\substack{u \in S \\ v \in T}} c(u,v) = c(S,T) \end{aligned}$$

■

Theorem 4 *Given a flow f in $G = [V, E]$ and s, t,as above, the following are equivalent"*

1. *f is a maximum flow in G from s to t.*
2. *The residual network G_f contains no directed path from s to t. [Such a path is called an augmenting path]*
3. *$|f| = c(S,T)$ for some cut (S,T) separating s and t.*

Proof. (1) $\Longrightarrow$ (2) : Suppose f is maximum and there is an augmenting path p. Let f_p be the flow that this path can take – note

$$f_p = c_f(p) = \min_{(u,v) \in p} [c_f(u,v) > 0$$

The $f + f_p$ is a flow in G with a greater value and this is a contradiction to $|f|$ being maximum.

(2) $\Longrightarrow$ (3) : Suppose G_f has no directed path from s to t. Let $S = \{v \in V :$ there is a directed path from s to v in $G_f\}$. Let $T = V = S$. (S,T) is a cut separating s and t.

$$\begin{aligned} [u &\in S, v \in T] \Longrightarrow [c_f(u,v) = 0] \\ &\Longrightarrow [f(u,v) = c(u,v)] \end{aligned}$$

Hence

$$\begin{aligned} |f| &= f(S,T) \\ &= c(S,T) \end{aligned}$$

(3) $\Longrightarrow$ (1) : For any feasible flow g, and any cut $(X, V - X)$ separating s and t, we have $|g| \leq c(X, V - X)$. Hence if we have a flow f and a cut (S,T) separating s and t with $|f| = c(S,T)$, then this flow is maximum and this cut is minimum. ■

0.0.2 Edmonds-Karp Algorithm

This algorithm finds at each step a flow augmenting path which has as few edges as possible. This is done by using BFS on the graph G_f. We will now show that if augmentations are done in this manner, the number of flow augmentations (and the whole algorithm) are polynomially bounded.

Lemma 5 *If the E-K algorithm is run on a flow network $G = [V, E]$ with source s and sink t, then for all vertices $v \in V - \{s\}$, the shortest distance $\delta(s, v)$ in the residual network G_f (measured by the number of edges) never decreases.*

Proof. Suppose not. Let $\delta_{f'}(s, v) < \delta_f(s, v)$ where f is the flow before and f' is the flow after some augmentation. Further suppose

$$[\delta_{f'}(s, u) < \delta_{f'}(s, v)] \Longrightarrow [\delta_f(s, u) \leq \delta_{f'}(s, v)]$$

Let p' be a shortest path from s to v in $G_{f'}$ with u as the node before v. Hence

$$\delta_{f'}(s, u) = \delta_{f'}(s, v) - 1$$

and

$$\delta_f(s, u) \leq \delta_{f'}(s, u)$$

If before the augmentation f', $f[u, v] < c[u, v]$, then

$$\delta_f(s, v) \leq \delta_f(s, u) + 1 \leq \delta_{f'}(s, u) + 1 = \delta_{f'}(s, v)$$

which contradicts our supposition. So $f[u, v] = c[u, v]$ before augmentation f'. This implis that $(u, v) \notin E_f$. But augmenting path p in G_f chosen to produce $G_{f'}$ must contain edge (v, u) in the direction from v to u since $(u, v) \in E_{f'}$ (by supposition) and $(u, v) \notin E_f$ as we have shown. Since p is a shortest path from s to t, its subpaths are also shortest paths. Hence

$$\delta_f(s, v) = \delta_f(s, u) - 1 \leq \delta_{f'}(s, u) - 1 = \delta_{f'}(s, v) - 2 < \delta_{f'}(s, v)$$

and this is a contradiction to the supposition. ■

Theorem 6 *The total number of augmentations performed by E-K algorithm is $O(|V|\,|E|)$.*

Proof. An edge (u, v) in the residual network G_f is said to be *critical* on an augmenting path p is the residual capacity of the path equals the residual capacity of the edge; that is $c_f(p) = c_f(u, v)$. After we augment, any critical edge is not a part of the next residual network. Moreover, at least one edge of an augmenting path is critical.

Next we show that each edge can be critical at most $\frac{|V|}{2}$ times. This gives us the required bound on the number of augmentations.

When an edge (u, v) is critical, since all augmentations are done along shortest paths, we have

$$\delta_{f'}(s, v) = \delta_{f'}(s, u) + 1$$

Since this edge disappears after this augmentation, it can not reappear in the residual network until edge (v, u) appears in some augmenting path at a later point in time. If f' is the flow when this event occurs,

$$\delta_{f''}(s, u) = \delta_{f''}(s, v) + 1$$

Since $\delta_f(s, v) \le \delta_{f'}(s, v)$, we have

$$\delta_{f''}(s, u) = \delta_{f''}(s, v) + 1 \ge \delta_f(s, v) + 1 = \delta_{f'}(s, u) + 2$$

Thus, from the time (u, v) becomes critical to when it becomes critical again, $\delta(s, u)$ increases by 2.

$$1 \le \delta(s, u) \le |V| - 1$$

Hnece the claim holds and therefore, the algorithm does $O(|V|\,|E|)$ augmentations and is therefore strongly polynomial. Complexity is $O\left|V\right|\left|E\right|^2)$ since BFS has complexity $\Theta(|E| + |V|)$. $\blacksquare$