Network Flows

Flow network: directed graph $G=(V,E)$ with source vertex (s) and sink (t); each edge $(u,v)\in E$ has non-negative capacity $c(u,v)$; if $(u,v)\notin E$, then $c(u,v)\equiv 0$.

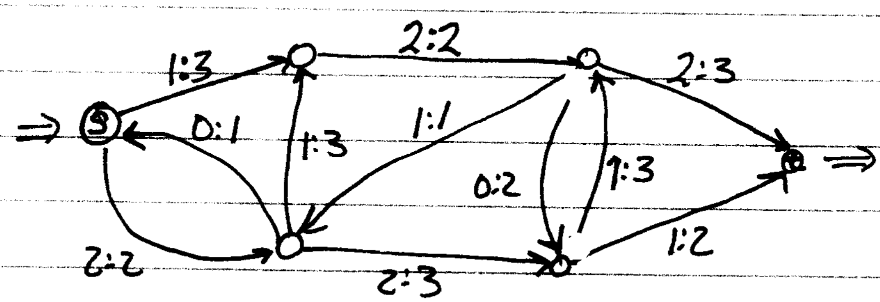
Flow on G is a function $f: V\times V \to \mathbb{R}$ satisfying

1. Capacity constraint: $f(u,v)\le c(u,v) \quad \forall u,v \in V$
2. Flow conservation: $\sum_{v\in V} f(u,v) = 0 \quad \forall u \in V-\{s,t\}$ (no buildup)
3. Skew symmetry: $f(u,v) = -f(v,u)$

Flow Value, $|f| = \sum_{v\in V} f(s,v) = f(s,V)$ is total flow out of source. notation: Cap. letter indicates implicit summation

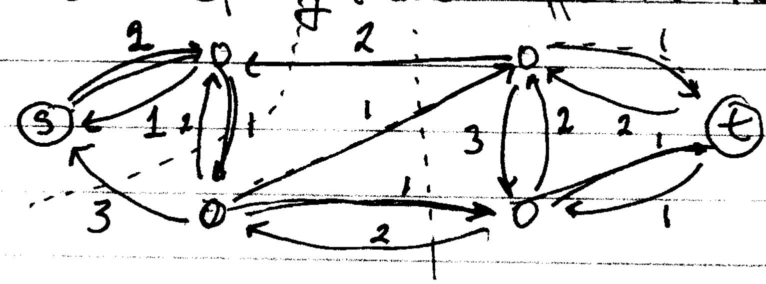
Can be shown that this is equal to flow into sink

Goal: FINDING MAXIMUM FLOW VALUES



Residual Capacity of (u,v) = $c_f(u,v) = c(u,v) - f(u,v) \ge 0$

Residual Network G_f is graph of edges with strictly positive residual capacity (that can support more flow)



<u>Augmenting Path</u> is a path from s to t in G_f.

Because the residual network G_f contains only edges that have excess capacity, any augmenting path in G_f represents a perturbation of the flow f with <u>increased</u> flow value.

Flow value can be increased along augmenting path p by $c_f(p) = \min_{(u,v) \in p} \{ c_f(u,v) \}$

Suggests an algorithm ("method")

① Start with Zero Flow
② Find an augmenting path p in corresponding G_f
③ Augment f by $c_f(p)$ and loop

⟹ FORD - FULKERSON ALGORITHM

<u>Potential Problems</u>

Convergence? Can it enter infinite loop? No, because we only accept augmenting paths.

Can we be guaranteed of finding an augmenting path if one exists? Breadth-First Search finds all shortest paths from a source. If none are found to sink, then none exist.
- Do we want a shortest path?
- Do we want a longest path?
- We really want a path of maximum capacity.

Global optimum? When no more augmenting paths can be found, how do we know we haven't found a local optimum rather than the global optimum?

Properties of Flows

Lemma (26.1 in CLRS)

For $G=(V,E)$ a flow network and f a flow in G.

① $f(X,X)=0$ for all $X \subseteq V$ → double summation includes $f(x,x')+f(x,x')=f(x,x')-f(x',x)=0$ (skew)

② $f(X,Y)=-f(Y,X)$ for all $X,Y \subseteq V$ (skew)

③ $f(X \cup Y, Z) = f(X,Z)+f(Y,Z)$ for all $X,Y,Z \subseteq V$ and $X \cap Y = \emptyset$

(expand summation, combine terms, recontract summations)

Illustrate Use:

Flow is defined as net flow leaving source; show equal to that entering sink

demonstrate: $|f| \equiv f(s,V) = f(V,t)$

book

$$
\begin{aligned}
|f| &\equiv f(s,V) \\
&= f(V,V) - f(V-\{s\},V) \quad \text{Lemma ③} \\
&= 0 + f(V, V-\{s\}) \quad \text{lemma ①, lemma ②} \\
&= f(V,t) + \underbrace{f(V, V-\{s,t\})}_{0 \text{ by flow conservation}} \quad \text{lemma ③}
\end{aligned}
$$

slightly more intuitive

$$
\begin{aligned}
|f| &\equiv f(s,V) + f(V,t) - f(V,t) \\
&= f(V,t) - \left(f(V,s) + f(V,t)\right) \quad \text{Lemma ②, Lemma ③} \\
&= f(V,t) - f(V,\{s,t\}) + f(V,V) \quad \text{lemma ①} \\
&= f(V,t) + \underbrace{f(V, V-\{s,t\})}_{0 \text{ by flow conservation}} \quad \text{Lemma ③}
\end{aligned}
$$

Cut is **partition** of V into ~~$S \cup T$ $(S \cap T = \emptyset)$~~ S and $T = V - S$
with $s \in S$ and $t \in T$.

Capacity of cut $c(S,T) = \sum_{u \in S} \sum_{v \in T} c(u,v) = c(S,T)$ "forward" direction only

- Small capacity cuts serve as "barriers" to large flows.
- A minimum cut of a network is that whose capacity is a minimum over all cuts in network.

Lemma: Given any flow f, $|f| = f(S,T)$ for any cut (S,T)

Proof:

$$
\begin{aligned}
f(S,T) &= f(S,V) - \cancel{f(S,S)}^{\to 0} \\
&= f(s,V) + \underbrace{f(S-s,V)}_{0 \text{ by flow conservation}} \\
&= |f|
\end{aligned}
$$

→ This is like another conservation of flow idea
- by definition for flow out of source $(S = \{s\})$
- already proven for flow into sink $(T = \{t\})$
- here proven for any partition (cut) dividing s, t

The Max-flow Min-Cut Theorem

Corollary: The value of any flow is bounded above by the capacity of any cut:
Proof: $|f| = f(S,T) = \sum_{u \in S} \sum_{v \in T} f(u,v) \le \sum_{u \in S} \sum_{v \in T} c(u,v) = c(S,T)$

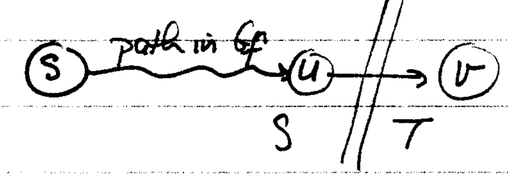
The following are equivalent
1. f is a maximum flow in G
2. The residual network G_f contains no augmenting paths
3. $|f| = c(S,T)$ for some cut (S,T) of G (minimum cut)

Proof Sketch

1 ⇒ 2 Augmenting path would allow flow in G to be increased above f, contradicting its being a maximum flow

3 ⇒ 1 By corollary, $|f|$ is bounded from above by the capacity of any cut, so $|f| = c(S,T)$ means it can't be increased

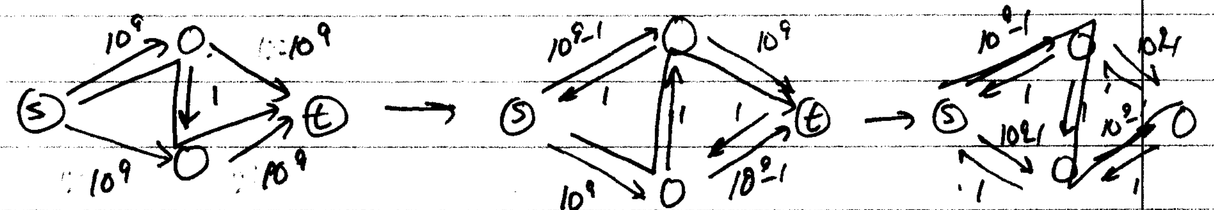
(2) ⇒ (3) G_f has no augmenting paths. Let $S = \{u \in V : \exists$ path from s to u in $G_f\}$ and $T = V - S$. $s \in S$ and $t \in T$ (otherwise would be augmenting path). (S,T) is a cut. $c_f(u,v) = 0$ for $u \in S, v \in T$ (otherwise v would be in S) $\therefore f(u,v) = c(u,v)$, since $c_f(u,v) = c(u,v) - f(u,v)$. $|f| = f(S,T) = c(S,T)$. ← summation



Thus, the Ford-Fulkerson Algorithm is guaranteed to result in successive improvements. When no more improvements are possible, the maximum flow has been found.

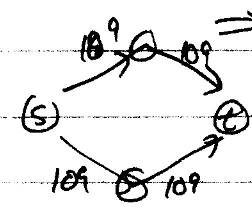
Analysis on integer capacities. $|f^*|$ is max flow. $O(E)$ is time to find augmenting path $[O(E+V)]$ $\Rightarrow O(|f^*| E)$

Possible Efficiency Problem:



$\Rightarrow 2 \times 10^9$ iterations

Correct Soln



Edmonds-Karp Algorithm

- Use BFS to find breadth-first augmenting path (positive edges in G_f have weights equal to unity)
- Time to find an augmenting path $\Theta(V+E) \Rightarrow \Theta(E)$ if all vertices reachable from source
- Claim number of augmentations limited to $O(VE)$ ← see below

  $\Rightarrow$ Running time $O(VE^2)$

Monotonicity Lemma

Let $\delta(v) = \delta_f(s,v)$ be breadth-first distance $s \to v$ in G_f. Suppose algorithm augments f to produce f'

[During E-K alg., $\delta(v)$ increases monotonically.]

Prove $\delta'(v) \geq \delta(v)$ by induction on $\delta(v)$

Base: $\delta'(s) = \delta(s) = 0$

Inductive Step: Consider BF path $s \to \cdots u \to v$ in $G_{f'}$

$\delta'(v) = \delta'(u) + 1$ because subpaths of shortest paths are shortest paths

$(u,v) \in E_{f'}$ [What about E_f?] $\to$ 2 cases

Case 1: $(u,v) \in E_f$

$$\begin{aligned} \text{Then } \delta(v) &\leq \delta(u) + 1 && \triangle \text{ inequal} \\ &\leq \delta'(u) + 1 && \text{induction} \\ &= \delta'(v) && \text{BF (above)} \checkmark \end{aligned}$$

Case 2: $(u,v) \notin E_f$

Augmenting path must have included (v,u) ← How else could this edge have appeared in the residual network?

$$s \to \cdots v \to u \to \cdots t$$

$$\begin{aligned} \text{Then } \delta(v) &= \delta(u) - 1 && \text{BF path} \\ &\leq \delta'(u) - 1 && \text{induction} \\ &= \delta'(v) - 2 && \text{BF path (above)} \\ &< \delta'(v) && \checkmark \end{aligned}$$

Theorem: # Augmentation Steps in Edmonds-Karp alg is $O(VE)$.

Proof:

Let p be an augmenting path

Residual capacity of some edge $(u,v) \in p$ is $c_f(p) = c_f(u,v)$

$\Rightarrow$ (u,v) called <u>critical</u>

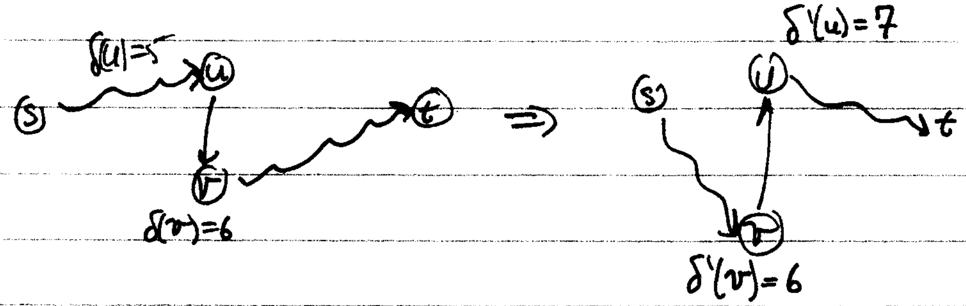
Augmentation removes (u,v) from critical path (for now)

Must wait until (v,u) augmented before reappears.

Before p augmented (u,v): $\delta(v) = \delta(u) + 1$ (p: BF path)

Before augmenting (v,u) later:

$$
\begin{aligned}
\delta'(u) &= \delta'(v) + 1 && \text{(BF path)} \\
&\geq \delta(v) + 1 && \text{(monotonicity lemma)} \\
&= \delta(u) + 2 && \text{(above)}
\end{aligned}
$$

Eg. 

$\Rightarrow \delta(u)$ increases ≥ 2 between times when (u,v) is critical.

Each $\delta(u)$ starts ≥ 0, never decreases, and remains $\leq |V|-1$ until unreachable.

$\Rightarrow$ (u,v) is critical at most $O(V)$ times

$O(E)$ edges in $G_f \Rightarrow O(VE)$ augmentations

Corollary: $O(VE^2)$

<u>Fastest Known Algorithm:</u> King, Rao, Tarjan: $O(VE \log_{E/(V \lg V)} V)$ time