Network Flows (Contd.)

Graph $G = (V, E)$ with capacities on edges.

s-t graph $\quad u \to v \in E$

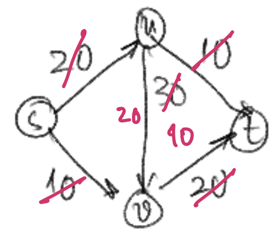
DAG $\quad c(u \to v)$

1. $0 \leq f(u \to v) \leq c(u \to v)$ } constraint on flow.

2. $\sum_{u} f(u \to v) = \sum_{w} f(v \to w) \quad \forall\, v \in V \setminus \{s,t\}$ } conservation of flow.

3. Throughput of the network:

$$\sum_{u} f(s \to u) \quad \text{subj to 1 and 2.}$$

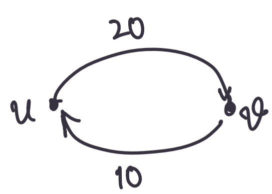


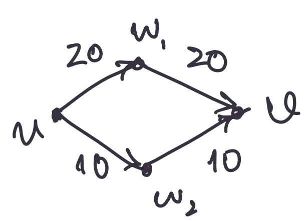
Flow that satisfies 1 and 2 is "feasible".

Want: Feasible flow that maximizes $\sum_{u} f(s \to u)$ subj to 1 and 2.

→ Suppose f is the flow through the network.

$$c_f(u \to v) = \begin{cases} c(u \to v) - f(u \to v) & \text{if } u \to v \in E \\ f(v \to u) & \text{if } v \to u \in E \\ 0 & \text{o/w.} \end{cases}$$





Cut of a graph:

↳ Let S and T be the partition of vertices of the graph.

$$\text{cut}(S,T) = \{e=(u,v) \mid u\in S \text{ and } v\in T\}.$$

Assume that $s\in S$ and sink $t\in T$.

Capacity of a cut:

$$\text{Capacity}(S,T) = \sum_{u\in S}\sum_{v\in T} c(u\to v).$$

$$\text{minCut}(G) = \min_{\substack{S\subseteq V \\ s\in S}} \{\text{capacity}(S, V\setminus S)\}.$$

<u>Theorem</u>: maximum flow in G = min Cut of G.

↳ If f is any feasible flow and (S,T) is any cut s.t source $\in S$ and sink $\in T$, then the total flow $|f|$ is at most the capacity (S,T).

$$|f| \leq \text{capacity}(S,T).$$

$$|f| = \sum_{w\in V} f(s\to w) + \sum_{v\in V\setminus\{s,t\}} \left(\sum_w f(v\to w) - \sum_u f(u\to v) \right)$$

Conservation of flow

$$= \sum_{v\in S}\sum_{w} f(v\to w) - \sum_{v\in S}\sum_{u} f(u\to v)$$

$$= \sum_{v\in S}\sum_{w\in T} f(v\to w) - \underbrace{\sum_{v\in S}\sum_{v\in T} f(u\to v)}_{\geq 0}.$$

flow is non-neg.

$$\leq \sum_{v\in S}\sum_{w\in T} f(v\to w)$$

$$= \text{capacity}(S,T).$$

Any feasible flow $\leq$ capacity of any cut.

$\max flow(G) \leq \min Cut(G)$.

→ $\max flow(G) \geq \min Cut(G)$.

$F \leftarrow$ zero.

1. Find a $s \rightsquigarrow t$ path π in the graph. Find the bottleneck capacity of this path.
   Flow $F \leftarrow F +$ bottleneck(π).

2. Construct a residual graph G_F and repeat the step 1.
   Do this until no $s \rightsquigarrow t$ paths can be found in G_F.

$$F(u \to v) \leftarrow \begin{cases} F(u \to v) + \text{bottleneck}(\pi) & \text{if } u \to v \in E \cap \pi \\ F(u \to v) - \text{bottleneck}(\pi) & \text{if } v \to u \in \pi \wedge u \to v \in E \\ F(u \to v) & \text{o/w}. \end{cases}$$

Obs: Updated F is still feasible.

F^* be the flow when the algorithm terminates.

$$0 = C_{F^*}(u \to v) = C(u \to v) - F^*(u \to v).$$

$u \in S$, $u \to v \in E$
$v \in T$

No more $s \rightsquigarrow t$ paths are found in residual graph

$$\sum_{(u,v) \in Cut(S,T)} c(u \to v) = \sum_{(u,v) \in Cut(S,T)} F^*(u \to v)$$

$Capacity(S,T) = |F^*|$

F^* is in fact optimal flow.

Recall that $\underbrace{\text{max flow}}_{\text{any}}(G) \leq \text{min Cut}(G)$.

TODO:

1. Algo that runs in $O(\log|C| \cdot m \cdot (m+n))$ instead of $O(|C| \cdot (m+n))$.

2. Bipartite matching.