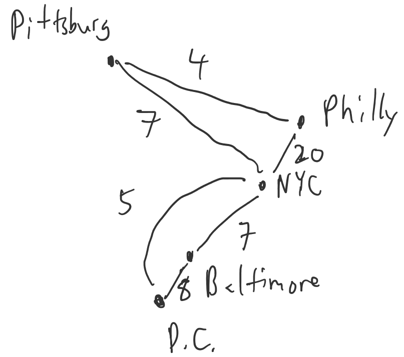
Lec22-network-flow

KT 7.1

Tuesday, October 31, 2023 8:07 PM

Suppose are Andrew Carnegie, and you want to ship steel from Pittsburg to Washington D.C.

Each railroad track has a capacity.



How can you send as much steel as possible.

A flow network is a connected, directed graph $G=(V,E)$
- each edge has a capacity $c_e \in \mathbb{N}$. (pos integer)
- source vertex $s \in V$ (no in-edges)
- sink vertex $t \in V$. (no out-edges)

An s-t flow is a function $f: E \longrightarrow \mathbb{R}^{\geq 0}$ representing amt of material carried on each edge, where

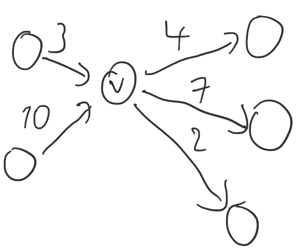
$$0 \leq f_e \leq c_e$$

$\forall v \in V$ except s & t, we have

$$f^{in}(v) = \sum_{e \in In(v)} f(e) = \sum_{e \in Out(v)} f(e) = f^{out}(v)$$

← in- and out-edges of v respectively.

Ex.

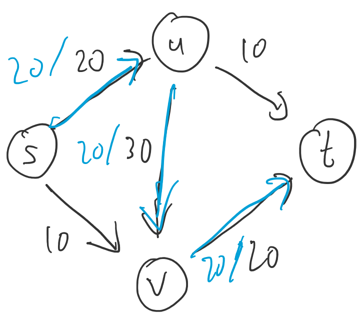


The value of flow f is $v(f) = \sum_{e \in Out(s)} f(e) = f^{out}(s)$ } total amt produced

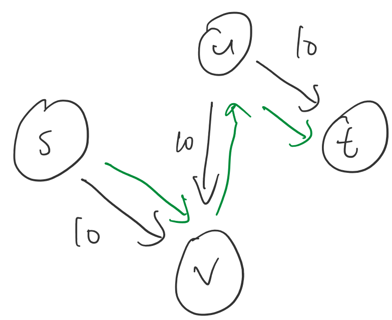
Problem (Maximum flow): Given a flow network G, find a flow f to maximize $v(f)$.

Algorithm idea: (greedy start)

1. Let $f(e) = 0 \ \forall e \in E$.
2. Repeat until stuck. (no path with remaining capacity)
   Choose an $s \to t$ path and push maximum flow possible along it (min remaining capacity edge along path)
3. Undo some flow along certain edges to create more paths to push flow along?

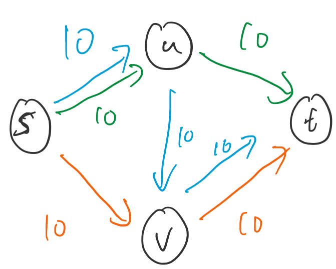


$v(f) = 20$



No more paths left.
What if we undo 10 units of $u \to v$ though

Want:



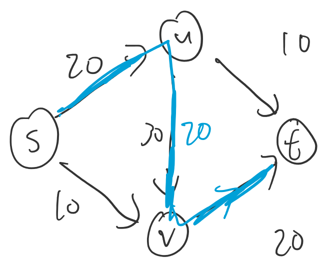
$v(f) = 10$

Residual graph

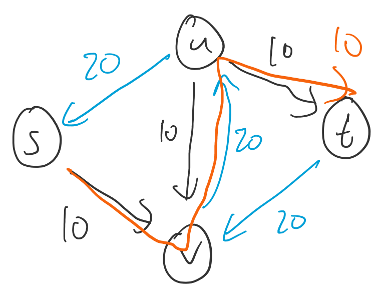
Given flow f on graph G, define the residual graph G_f on the same nodes but modified/new edges

forward edges: $\forall\ e=(u,v)\in G$ where $f(e)<c_e$,
include edge $e'=(u,v)\in G_f$ with capacity $c_e - f(e)$ } capacity remaining

backward edges: $\forall\ e=(u,v)\in G$ where $f(e)>0$,
include edge $e'=(v,u)\in G$ with capacity $f(e)$. } can erase allocated flow
(if this is already existing edge, add to capacity)

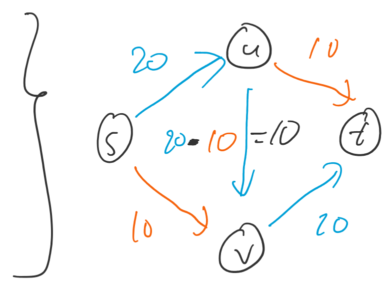


flow f



residual graph G_f.

path P with capacity 10 in residual graph



augmented flow $f+P$.

If P is an $s \to t$ path in G_f with smallest capacity edge bottleneck$(P,f)>0$, can create new flow $f+P$ by sending bottleneck(P,f) along P. (increased value(f))

augment(f,P):
  $b=$ bottleneck(P,f)
  $\forall\ e=(u,v)\in P$:
    If e is a forward edge: (with capacity $c_e - f(e)$)
      Increase $f(e)$ by b ($f'(e)=f(e)+b \le f(e)+c_e-f(e) \le c_e$)

If e is a backward edge: (has capacity $f(e')$)

Let $e' = (v, u)$.

Decrease $f(e')$ by b (still ≥ 0)

return f.

Capacity constraint still met:

$$s \xrightarrow{\text{for}} v_1 \xrightarrow{\text{for}} v_2 \xrightarrow{\text{back}} v_3 \xrightarrow{\text{back}} v_4 \xrightarrow{\text{for}} t \quad \in G_f$$

$$s \xrightarrow{+b} \underbrace{v_1}_{\text{increase}} \xrightarrow{+b} \underbrace{v_2}_{\text{no change}} \xleftarrow{-b} v_3 \xleftarrow{-b} v_4 \xrightarrow{+b} t \quad \in G$$

$f^{in}(v_1) + f^{out}(v_1)$

Ford-Fulkerson alg

MaxFlow(G):

Set $f(e) = 0 \quad \forall e \in G$

While $P :=$ FindPath$(s, t,$ Residual$(G, f)) \neq$ None:

$f :=$ augment(f, P)

Return f

Running time:

- All values are integers.
- Always augment value(f) by at least 1 unit.
- Never send more than $C = \sum_{e \in Out(s)} c_e$.

$\Rightarrow$ Ford-Fulkerson terminates in at most C iterations of augmenting.

- G has m edges $\Rightarrow$ G_f has $\leq 2m$ edges

- G has m edges $\Rightarrow$ G_f has $\leq 2m$ edges.
- Can find $s \to t$ path in G_f in $O(m+n)$ time.

WLOG, $n = O(m)$ because any node without an edge can be pruned.

$\Rightarrow$ Ford-Fulkerson is $O(mC)$.

Minimum cuts

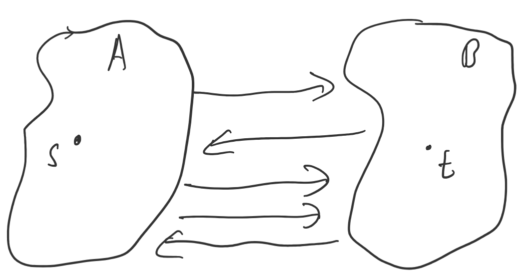
Recall that a cut of a undirected graph $G=(V,E)$ is a partition of the nodes into sets A, B, (where $B = V - A$).

The weight of a cut $(A,B) = \sum_{\substack{(a,b)\in E \\ \text{where } a\in A, \\ b \in B}} w(a,b)$.

In a flow network, an $s-t$ cut (A,B) is a cut s.t. $s \in A$, $t \in B$.

The capacity of an $s-t$ cut $(A,B) = \sum_{\substack{(a,b)\in E \\ a\in A \\ b\in B}} c_{(a,b)}$.

Thm Let f be an $s-t$ flow and (A,B) an $s-t$ cut.
Then $v(f) = f^{out}(A) - f^{in}(A)$.



pf. Any flow that leaves A must either return to A or end up in B, but if it ends in B, it must end at t. $\boxtimes$

Thm Let f be any $s-t$ flow and (A,B) be any $s-t$ cut.

Then $v(f) \le \text{capacity}(A, B)$

pf. $$v(f) = f^{out}(A) - f^{in}(A)$$
$$\le f^{out}(A)$$
$$= \sum_{e \in Out(A)} f(e) \le \sum_{e \in Out(A)} c_e = \text{capacity}(A, B)$$ $\boxtimes$

$\Rightarrow$ all cuts are at least as big as any flow.

$\Rightarrow$ minimum cut constrains maximum flow capacity.

Story: Soviet scientists like A.N. Tolstoi [1930] wanted to maximize flow of goods, like cement on Soviet railways. That is a max-flow problem.

The US Air Force was also studying Soviet railways, but because they wanted to break it. [1950, Ross, Harris]. This is a min-cut problem.

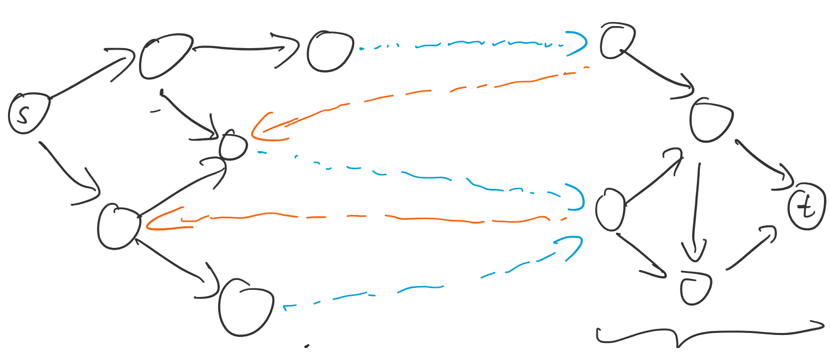
Turns out both problems are "dual" to each other.

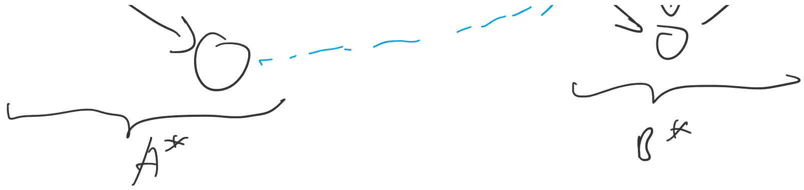
Min Cut = Max Flow.

Consider f^* returned by Ford-Fulkerson.

Then G_{f^*} defines a cut in G by the nodes reachable from s A^*
and the nodes not reachable from s B^*

($B^* \ne \emptyset$ because $t \in B^*$, or else we wouldn't terminate augments)





Blue edges must be saturated, (or else in G_{f^*} that edge is present)

Red edges must have 0 flow (or else in G_{f^*} there would be an opposite direction edge connecting A^* & B^*)

$\Rightarrow v(f^*) = \text{capacity}(A^*, B^*)$

$\Rightarrow$ No flow can have value bigger than capacity(A^*, B^*)

$\Rightarrow f^*$ is a max-flow & (A^*, B^*) is a min-cut.

Thm (Min-cut Max-flow) The value of the maximum flow in any flow graph is equal to the capacity of the minimum cut.

Can find a min-cut by constructing max flow f^* & using BFS to find nodes reachable in G_{f^*}.

Ford-Fulkerson only is guaranteed to terminate with integer weights

Variation Edmunds-Karp works for any weights in $O(VE^2)$ time.