Given a flow network

$$G = (V, A), \quad s, t \in V,$$

$$c : A \to \mathbb{R}_+.$$

Want to compute a max value flow $f : A \to \mathbb{R}_+$ s.t.

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

(Value of f is $|f| := \sum_{s \to w} f(s \to w) - \sum_{u \to s} f(u \to s)$)

$f(a) \leq c(a) \quad \forall\ a \in A$

Augmenting Path Algorithms

Start with $f = \vec{0}$

(i.e. $f(a) = 0 \ \forall a$)

Repeatedly:

Create residual graph

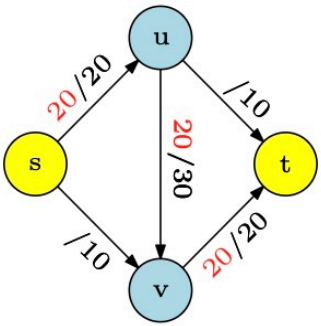
$G_f = (V, A_f)$

(For each $a \to v \in A$,

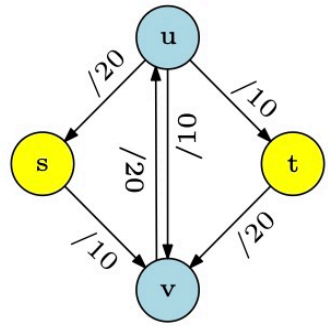
if $f(a \to v) = 0$, add

$a \to v$ to A_f with

$c_f(u \to v) = c(u \to v)$



G



G_f

if $f(u \to v) = c(u \to v)$,

add $v \to u$ to A_f

$c_f(v \to u) = c(u \to v)$

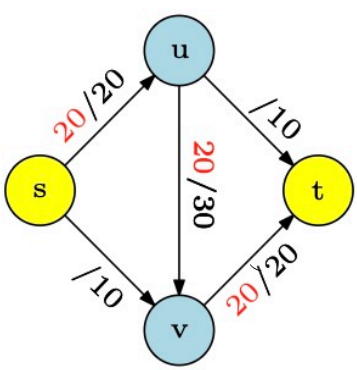
o.w. add $u \to v$ & $v \to u$ to A_f

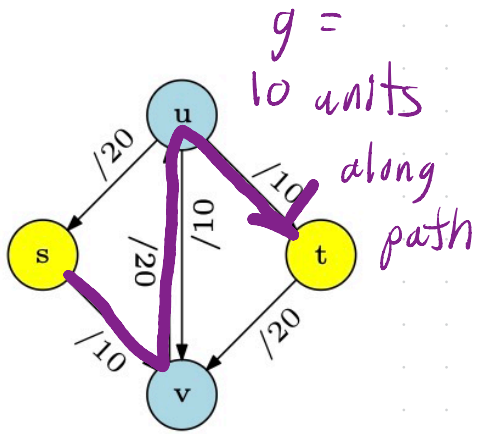
$c_f(u \to v) = c(u \to v) - f(u \to v)$

$c_f(v \to u) = f(u \to v)$

If there is an (s,t)-path P in G_f...

Let g be the flow in G_f that sends min capacity of P's arcs units of flow along P.

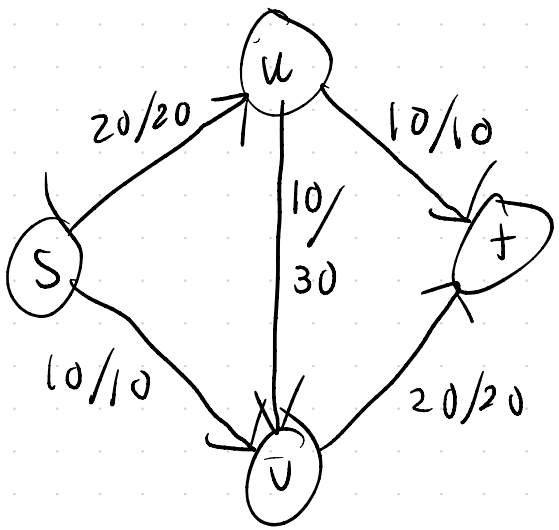




$f \leftarrow f + g$

$((f+g)(u \to v) = f(u \to v) + g(u \to v)$ if $g(u \to v) \geq 0$

$$((f+g)(u \to v) = f(u \to v) - g(v \to u)$$
$$\text{if } g(v \to u) = 0)$$



If no path, return f.

$O(m)$ time per iteration

If capacities are integers, add ≥ 1 unit to flow value every

iteration.

Let f^* be max value flow.

$\leq |f^*|$ iterations

$\Rightarrow O(m|f^*|)$ time total.

pseudo polynomial time

(poly in # objects & number magnitude)

What if you take max bottleneck path?

Can find path in $O(m \log n)$ time.

Can argue # iterations $\in O(m \log |S^*|)$

So total time $O(m^2 (\log n)(\log |S^*|))$

weakly polynomial time

Shortest (in # edges) augmenting paths (Edmonds-Karp '72)

Lemma: If we augment along shortest (s,t)-path in G_f, the shortest path length is not lower in new residual graph.

Lemma: We can do at most m iterations before path length increases.

$\Rightarrow \leq n-1$ edges on shortest paths

$\Rightarrow n-1$ distance increases

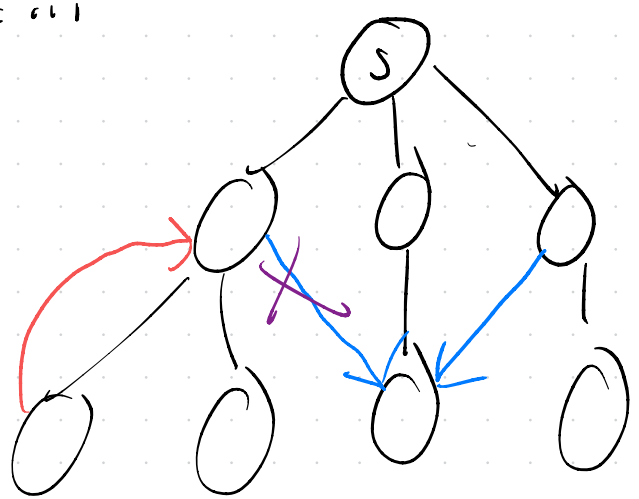
$\Rightarrow O(mn)$ iterations

Use BFS each iteration in $O(m)$ time.

So $O(m^2 n)$ time total.

strongly polynomial time

Suppose we do a BFS in G_s...



L_0

L_1

L_2



L_i nodes are distance i from s.

forward arcs go from some $u \in L_i$ to $v \in L_{i+1}$

backward arcs go from $u \in L_i$ to $v \in L_j$ with $j < i$

Shortest (s,t)-paths use only forward arcs.

When we augment along a path P, its min res. capacity arc leaves G_f.

Removing arc will not decrease distance.

We may add backward edges to G_s as we augment along the path. Those won't decrease the distance.

After $\leq m$ iterations we've removed all forward arcs implying t is a particular distance.

We don't <u>have</u> to pick individual paths from G_f...

If g is <u>any</u> flow in G_f, then $f+g$ is a flow in G, & $|f+g| = |f| + |g|$.

f is a <u>blocking flow</u> if for every (s,t)-path P, at least one edge of P is saturated ($f(u \to v) = c(u \to v)$).

Lemma: Suppose f is a blocking flow in G & t is distance d from s in G. t is distance $> d$ in G_f.

$\Rightarrow$ Max (s,t)-flow can be computed with $O(n)$ blocking flow computations.

Turns out we can do a BFS, find distances, & use only forward edges to find a blocking flow in $O(mn)$ time total.

$\Rightarrow O(mn^2)$ time total

(Dinitz)

$\vdots$

Orlin: $O(mn)$

2024: $m^{1+o(1)}$ time assuming

$1/2$

$\subseteq$ poly(n, m)

integer capacites

Connectivity:

Given undirected $G = (V, E)$.

$$\alpha_G(s,t) = \min_{U \subset V, s \in U, t \in V-U} |\delta(U)|$$

$|\delta(U)|$: # edges leaving U

i.e., min # edge to remove to separate s & t

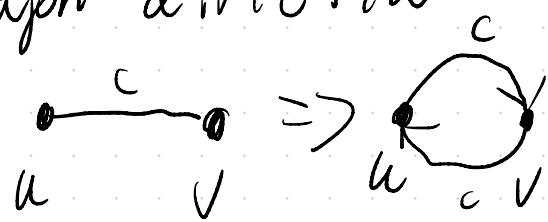
Everything today & Wed applies to capacities $c: E \to \mathbb{R}_+$

$\delta(U)$:= sum of capacities leaving

u

$\alpha_G(s,t)$ = max value of (s,t)-flows

(can make graph directed by having



G is undirected, so

$$\alpha_G(s,t) = \alpha_G(t,s)$$

$\binom{n}{2}$ values for $\alpha_G(s,t)$ across choices for $s \neq t$?

Actually $n-1$ values.

And can represent them as the cuts themselves (choices of (u))

using a Gomory-Hu Tree.

Actually applies to any "symmetric submodular function".