1 Ford-Fulkerson Method

Not really an algorithm but a class of algorithms.

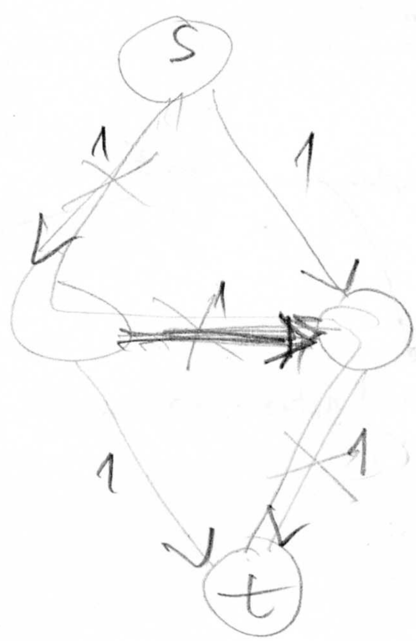
- Suppose we started with some flow and wanted to determine whether it was the maximum. How would we do it?
- Look for a path from s to t that can admit more flow and then increase the flow along that path (this is called an **augmenting path**).

Algorithm 1.0: Ford-Fulkerson(G, s, t).

```
initialize flow f to 0.
while there exists an augmenting path p
  do augment flow f along p
return f.
```

What do we need to show to prove that Ford-Fulkerson is correct if it terminates?

- if there is no augmenting path then f is a max flow. x



Residual networks

So how do we find an augmenting path?

- Let's just create a graph showing the excess capacity we have at each edge.
- The excess capacity of an edge (u, v) is called the **residual capacity** of (u, v):

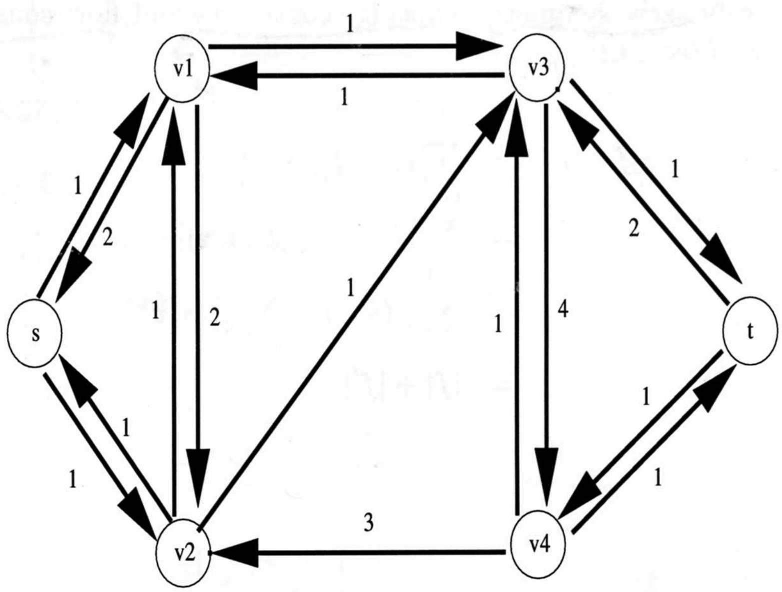
$$c_f(u, v) = c(u, v) - f(u, v)$$

- The graph is called the **residual network** of G induced by f denoted by $G_f = (V, E_f)$ where

$$E_f = \{(u, v) : c_f(u, v) > 0\}$$

- Each edge in E_f is called a residual edge.

Draw the residual network for graph previously shown:



- Can we have $(u, v) \in E_f$ if $(u, v) \notin E$?
- Yes! notice edge (v_1, s) in the residual graph drawn.
- But $(u, v) \in E_f$ if and only if at least one of (u, v) or (v, u) are in E.
- $|E_f| \leq 2|E|$

Definition 1.1 *Define* $(f_1 + f_2)(u, v) = f_1(u, v) + f_2(u, v)$

Lemma 1.2 (CLR Lemma 27.2)

- *Let G be flow network with source s and sink t.*
- *Let f be a flow in G.*
- *Let G_f be the residual network induced by f.*
- *Let f' be a flow in G_f.*

Then:

- *$f + f'$ is a flow.*
- $|f + f'| = |f| + |f'|$

Proof. Need to verify skew symmetry, capacity contraints and flow conservation to show that $f + f'$ is a flow. (See CLR.)

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

∎

How to find flow in residual network? $\Rightarrow$

1.0.1 Augmenting paths

Definition 1.3 *Given a flow network G and a flow f, an augmenting path is a simple path from s to t in the residual network G_f.*

Definition 1.4 *The residual capacity of a path p in a residual network G_f is:*

$$c_f(p) = \min\{c_f(u,v) : (u,v) \in p\}$$

Lemma 1.5 (CLR Lemma 27.3) *Let*

- *G be a flow network.*
- *f be a flow.*
- *and p be an augmenting path in G_f.*

Define $f_p(u,v) = \begin{cases} c_f(p) & \text{if } (u,v) \in p \\ -c_f(p) & \text{if } (v,u) \in p \\ 0 & \text{otherwise} \end{cases}$ *Then f_p is a flow in G_f and $|f_p| = c_f(p) > 0$.*

Proof. Exercise 27.2-7. ∎

Corollary 1.6 (CLR Cor 27.4) *Let $f' = f + f_p$* ; i.e. $\forall u,v,\ f'(u,v) = f(u,v) + f_p(u,v)$

- *Then f' is a flow in G with value $|f| + |f_p| > |f|$.*

Proof. Immediate from previous two lemmas. ∎

- So we can strictly increase flows by adding flow along an augmenting path.

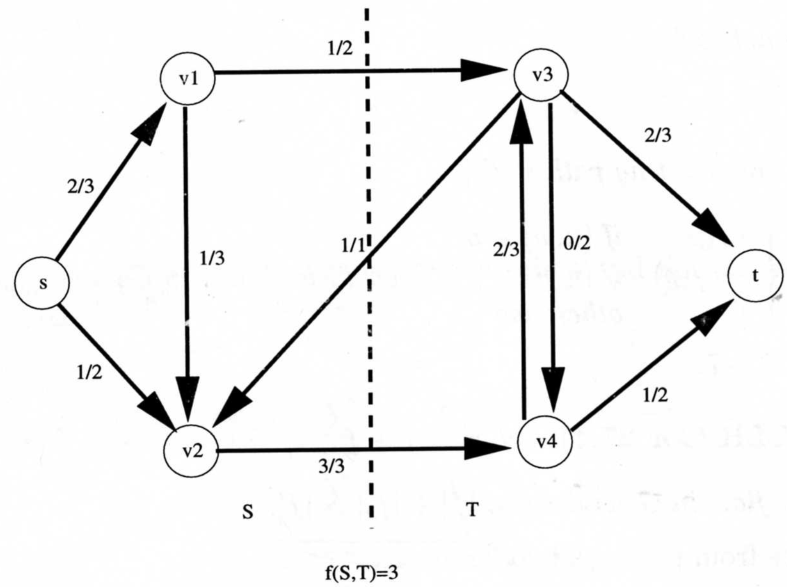
Q: when there are no more augmenting paths, is flow maximal?

Cuts of flow networks

Definition 1.7 *A* **cut** (S, T) *of a flow network* $G = (V, E)$ *is a partition of* V *into* S *and* $T = V - S$ *such that* $s \in S$ *and* $t \in T$.

- *The* **net flow across a cut** (S, T) *is* $f(S, T)$. $= \sum_{u \in S, v \in T} f(u,v)$
- *The* **capacity of a cut** (S, T) *is* $c(S, T) = \sum_{u \in S, v \in T} c(u, v)$.

Show example using network already drawn and a cut down the middle.



$$c(S, T) = 5$$

Lemma 1.8 (CLR Lemma 27.5) *Let* (S, T) *be any cut in* G. *Then*

$$f(S, T) = |f|$$

Proof. Intuitively, we are counting the flow that moves from one side to the other. There is no place for the flow to go once it reaches the t side except to t.

$$\begin{aligned}
f(S, T) &= f(S, V - S) \\
&= f(S, V) - f(S, S) \qquad \text{by Lemma 27.1} \\
&= f(S, V) \qquad \text{since } f(S, S) = 0 \\
&= f(s, V) + f(S - s, V) \qquad \text{by Lemma 27.1} \\
&= f(s, V) \qquad \text{since} t \notin S - \{s\} \\
&= |f|
\end{aligned}$$

Corollary 1.9 (CLR Cor 27.6) *Let (S,T) be any cut. then*

$$|f| \le c(S,T)$$

Proof.

$$\begin{aligned} |f| &= f(S,T) && \text{By Lemma 27.5} \\ &= \sum_{u \in S} \sum_{v \in T} f(u,v) \\ &\le \sum_{u \in S} \sum_{v \in T} c(u,v) \\ &= c(S,T) \end{aligned}$$

∎

Theorem 1.10 (Max-flow Min-cut Theorem) *Let f be a flow in G. The following conditions are equivalent.*

(##)

1. *f is a maximum flow in G.*
2. *G_f contains no augmenting paths.*
3. *$|f| = c(S,T)$ for some cut (S,T) of G.*

Proof.

(1) $\Rightarrow$ (2) $(\neg 2 \Rightarrow \neg 1)$

- Let f be a max flow and suppose G has an augmenting path.
- by Cor CLR 27.4, $|f + f_p| > |f|$.
- contradiction.

(2) $\Rightarrow$ (3)

- Suppose G_f contains no path from s to t.
- Define $S = \{v \in V : \text{there is a path from} s \to v \text{ in } G_f\}$ and $T = V - S$.
- $s \in S$ (obvious) ; $t \in T$ since $\nexists$ $s \rightsquigarrow t$ path in G_f
- $\Rightarrow$ (S,T) is a cut ~~since there is no path from s to t in G_f.~~
- For each $u \in S$ and $v \in T$, $f(u,v) = c(u,v)$ since otherwise $(u,v) \in E_f$ which implies $v \in S$.
- So $c(S,T) = f(S,T) = |f|$.

(3) $\Rightarrow$ (1)

- By Cor CLR 27.6, $|f| \le c(S,T)$ for all cuts (S,T).
- So $|f| = c(S,T)$ for some cut implies f is a max flow.

∎

Ford-Fulkerson Algorithm

Algorithm 1.0: Ford-Fulk(G, s, t).

$\Theta(E)$

```
for each (u,v) ∈ E
  do f[u,v] ← 0
     f[v,u] ← 0
```

$\Theta(V+E) = \Theta(E)$

```
while there is a path p from s to t in G_f
  do c_f(p) = min{c_f(u,v) : (u,v) ∈ p}
     for each (u,v) ∈ p
       do f[u,v] ← f[u,v] + c_f(p)
       do f[v,u] ← −f[u,v]
return f.
```

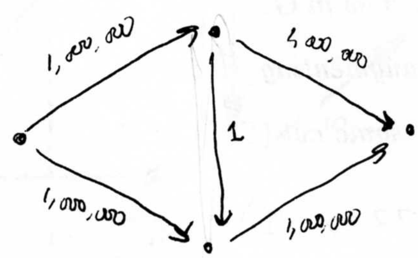
$O(E)$

how many times?

Flow is always strictly increasing - how long to get max flow?

Do bad example of running time from the book:

PS. 600



Running Time Suppose all capacities are integers.

- Find a path using DFS or BFS in $O(E)$ time (since connected).
- At most $|f^*|$ iterations since value increases by at least 1 each time
- So $O(E|f^*|)$ time where f^* is a max flow.
- Problem: What if max flow is huge.

Draw long example:

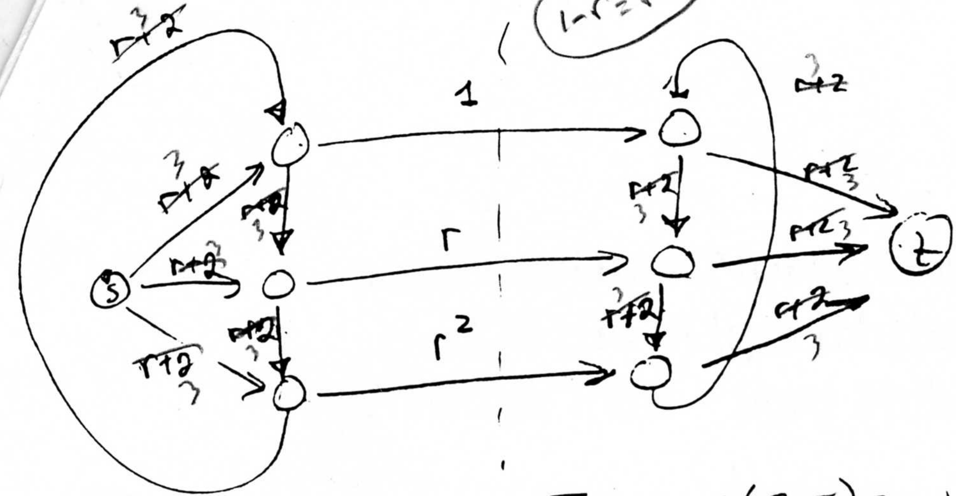
① If all capacities are integral, $O(E\,|f^*|)$

② If all capacities rational, $O(E\,c\,|f^*|)$ where $c =$ LCM of denominators of rational capacities (reduced to normal form)

(Note: all numbers representable on computers are effectively rational)

③ If non-rational capacities, can converge arbitrarily slowly...

$(1-r=r^2$



$$|f^*| = 1+r+r^2 = 1+r+(1-r) = 2$$

S T $c(S,T) = 1+r+r^2.$

This is min cut capacity

so $|f^*| = c(S,T)$

all other cuts have capacity > 2

$r^2+r-1=0$

$\cancel{x^2+x-1} \Rightarrow r = \frac{-1+\sqrt{5}}{2} \approx 0.618 = \hat{\phi}$

$= \phi - 1$

golden ratio

$\forall$ we have $\boxed{1-r=r^2}$

In general $r^{n+2} = r^n - r^{n+1} \quad \forall n \geq 0.$

$r^n - r^{n+1} = r^{n+2}$

$1 > r > r^2 > r^3 > \cdots > 0$

$r+2 = \frac{1}{1-r} = \sum_{n=0}^{\infty} r^n$

$r+2-r^2-2r = 2-r^2-r = 1.$

c_f

$1+r^2+r^3+r^4+\cdots$

$= \sum_{k=0}^{\infty} r^k - r$

$= \frac{1}{1-r} - r$

$= \frac{1-r(1-r)}{1-r}$

$= \frac{1-r+r^2}{1-r}$

$= \frac{1-r+(1-r)}{1-r}$

$= 2$ ← but this is max flow; i.e., it will take forever to get there.

total flow is

$1+r^2+r^3+\cdots$

$= \left(\sum_{n=0}^{\infty} r^n\right) - r$

$= 2.$

step 1 send 1 thru top edge

gives following residual

step 2 send r^2 over lowest edge back over upper and thru middle to give