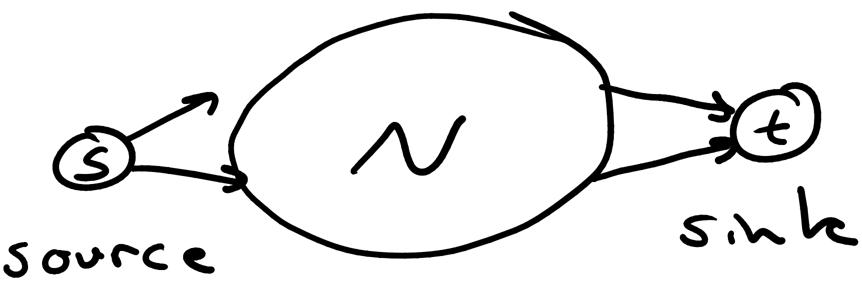
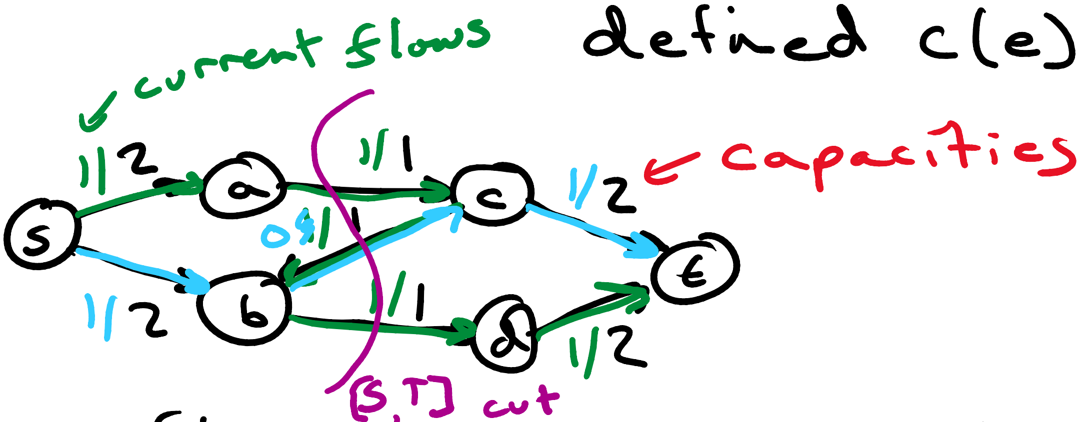
Lecture 15 - Flow Networks

Monday, March 20, 2023 1:28 PM

Flow network



$\forall e \in E(G)$: we have a capacity defined $c(e) \geq 0$



a flow on G assigns to each edge some <u>flow</u> value $f(e)$

these flow values must be <u>feasible</u>

$\forall e \in E(G)$:

$$0 \leq f(e) \leq c(e)$$

$\forall v \in V(G)$, except for s,t:

$f^-(v) =$ sum of fl

$f^-(v)$ = sum of flows on incoming edges

$f^+(v)$ = sum of flows on outgoing edges

$$f^-(v) = f^+(v)$$

flow on all of G

val(f) = total flow

$\uparrow$ current flow

$$val(f) = f^+(s) - f^-(s) \swarrow$$
$$= f^-(t) - f^+(t)$$

usually 0

<u>maximum flow</u>: feasible flow where val(f) is maximum

Given feasible flow f,

we define f-augmenting path P_f
where P_f goes from $s \to t$
$\forall e \in P_f$:

if P_f follows direction of e

if P_f follows direction of e

then $f(e) < c(e)$

if P_f goes against direction of e

then $f(e) > 0$

$\epsilon(e) = c(e) - f(e) =$ tolerance of e for forward edges in P_f

$\epsilon(e) = f(e) =$ tolerance of e for backword edges in P_f

Given P_f, we consider the minimum tolerance $\forall e \in P_f \rightarrow z$

To augment our flow:

$\forall e \in P_f$: $f(e) += z$ for forward e

$f(e) -= z$ for backward e

Note: when we augment a flow,

Note: when we augment a flow, we increase val(f) by 2

---

source-sink cut

$$[S, T]$$

S = source set of vertices

T = sink set of vertices

note: $s \in S$, $t \in T$

The size of $|[S,T]|$ is just the sum of the capacities of the cut edges

$$= \sum c(e) \quad \forall e \in [S,T]$$

$S = \{$vertices that can be reached from s along a pseudo-augment. path$\}$

$T = \{$

$T = \{V(G) - S\}$

path following a potential f-aug path but can't reach to t

Note: the size of any cut gives us a bound on flow

$$|[S,T]| \geq val(f)$$

? ? Big Question ? ? ? ? ? ? ? ? ?

Does min cut = max flow

A: yes

to prove this via a few equivalences

1. f is a max flow

2. no f-aug paths

3. $|[S, T]| = \text{val}(f)$

We'll show $1 \Rightarrow 2 \Rightarrow 3 \Rightarrow 1$

$(1 \Rightarrow 2)$ contrapositive

$\neg 2 \Rightarrow \neg 1$

$\exists$ f-aug path $\Rightarrow$ f is not a max flow

$\Rightarrow$ we've already seen how to increase flow given f-aug path

$(2 \Rightarrow 3)$

no f-aug paths $\Rightarrow$ cut is equal to flow on the network

S = set of reachable vertices from s following pseudo f-aug paths

Note: $s \in S$, $t \notin S$

all edges from $S \to T$ have

$c(e) = f(e)$

all edges from $T \to S$ have

$f(e) = 0$

$$\text{val}(f) = \sum \text{flows from } S \to T \underbrace{- \sum \text{flows from } T \to S}_{=0}$$

$$= \sum \text{flows from } S \to T$$

$$= \sum_{e \in [S,T]} c(e) = |[S,T]| \checkmark$$

(3 => 1) cut = flow => flow is max

Note: the capacities on edges gives us cut = flow

Q: can we increase our flow?

A: No. We've observed that forward edges are at capacity and backward edges are at zero flow

$\Rightarrow$ we cannot increase our flow ✓

Combined with our prior inequality → cut ≥ max flow and cut = max flow

$\Rightarrow$ $\boxed{\text{min cut} = \text{max flow}}$

QED

To get max-flow / min cut (min s,t-cut)

Initialize all $f(e)$ to zero

while $\exists$ some f-aug path P_f

   Find z = min tolerance on P_f

   update $f(e), e \in P_f$ by z

→ we're done

To get min cut:

   define our cut as $[S, T]$

   S = vertizes reachable from

      s on pseudo-f-aug. paths

   T = everything else

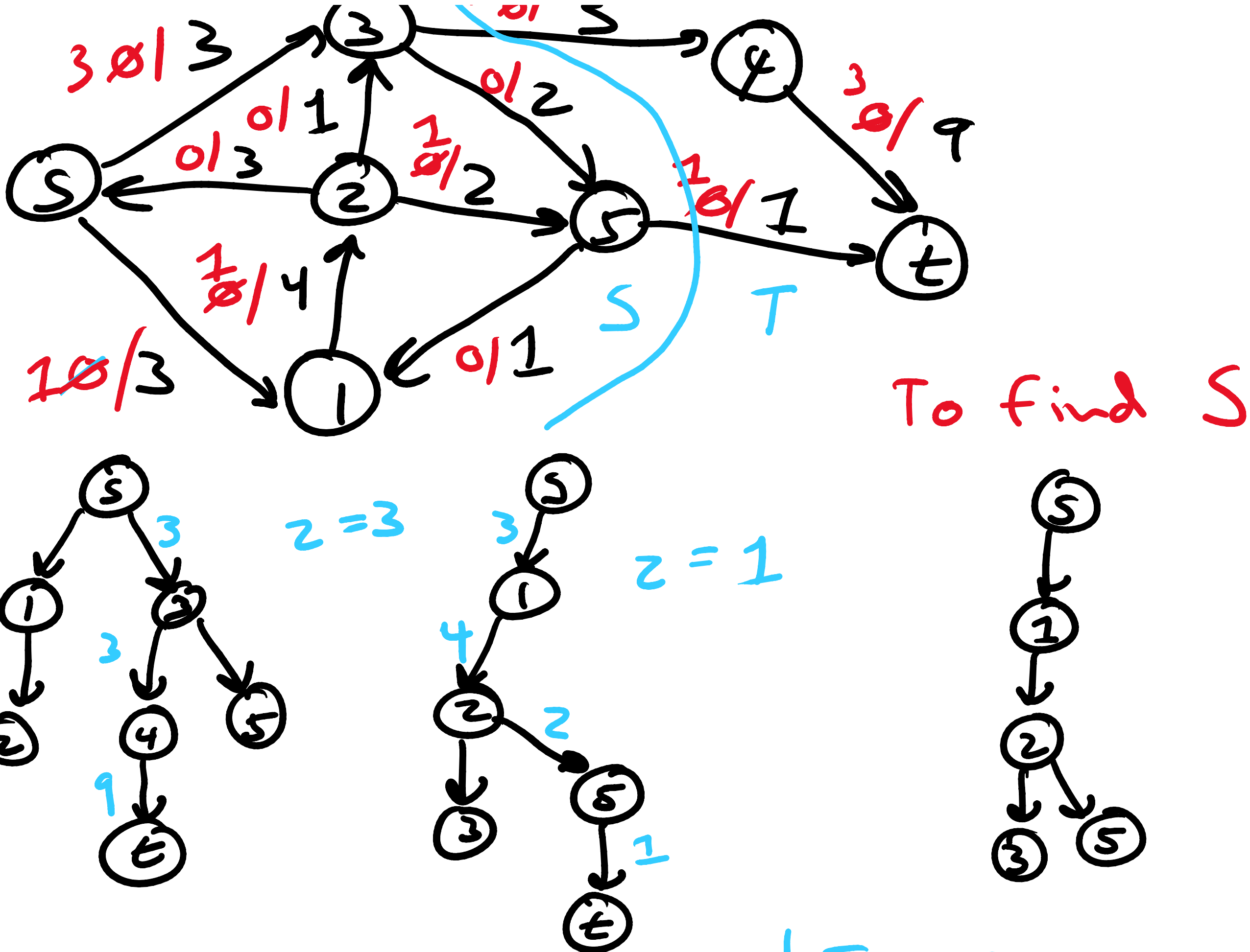
Basic algorithm:

   Ford-Fulkerson algorithm

If we use BFS to find P_f:

   Edmonds-Karp algorithm

Edmonds-Karp example



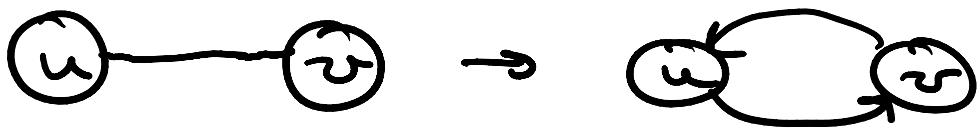
$|[S,T]| = 4$

$val(f) = 4$

Quiz 8 stuff

To make undirected G
to directed G'

* replace each $e = (u,v) \in E(G)$
with $e = (u,v)$, $f = (v,u) \in E(G')$



G G'

For the quiz problem

* replace each vertex v with a single directed edge with unit capacity
* this edge goes from $v_0 \rightarrow v_1$ where v_0 has all of v's original incoming edges and v_1 has v's outgoing edges