Flows and cuts:

we have a graph $V = \{\text{vertices}\}$
$E = \{\text{edges}\}$

make this graph directed

there is a unique vertex s (the source): this has only outgoing edges

and d (the drain): this has only incoming edges

- each edge has an associated capacity (which is integer; ≥ 0)
- flow through the edge may not exceed its capacity
- count incoming flows +ve, outgoing -ve
- Kirchoff's Law applies for all but s and d

$$\sum_{e \in e(v)} f_e = 0$$

<u>Problem</u>: what is the maximum flow from s to d. ?

- many practical applications
- many algorithms follow.

<u>Augmenting path</u>:

- a path from s to d
  - undirected
  - all forward $(s \to d)$ arrows below capacity
  - all backward $(d \to s)$ arrows greater than zero

1) if an augmenting path exists, flow is not maximal
   <u>Because</u> we can increase flow along this path.

2) if flow is maximal, there is no A.P.
   <u>Because</u> if there were, we could augment

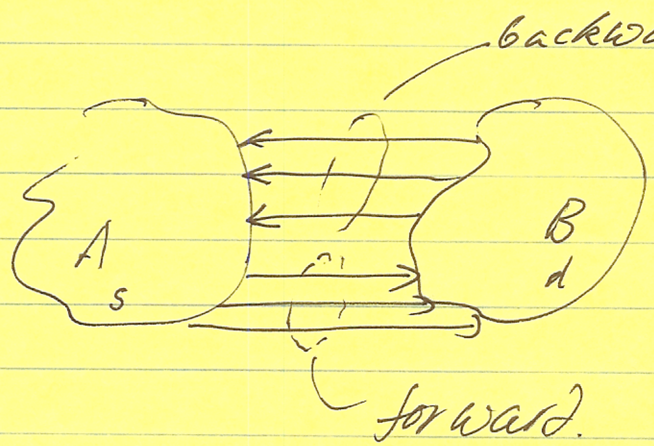
Algorithm outline

- find augmenting path
- max flow on this path

- Efficiency depends on details of how path is found

Now, consider a disconnecting cut

- set of edges that disconnects s, d
- divide verts into A, B st $A \cup B = V$ and $A \cap B = \emptyset$ and $s \in A$, $d \in B$



- total flow $A \to B = \Sigma\text{forward} - \Sigma\text{backward}$

Value of the cut

$= \sum$ capacities forward

~~$- \sum$ capacities backward~~

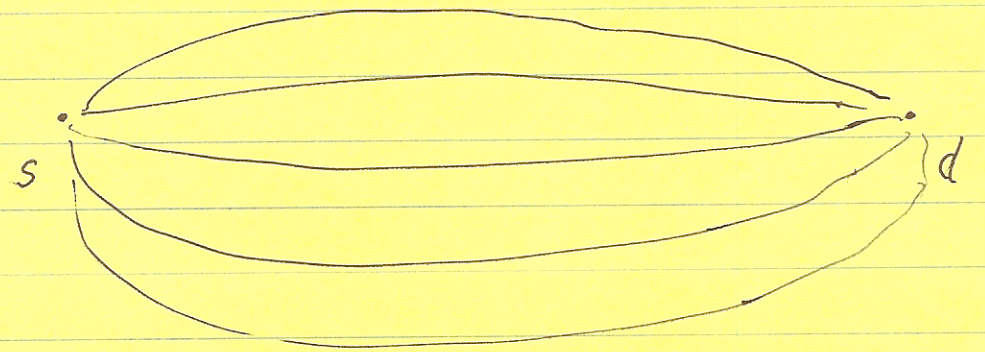
$\Rightarrow$ If flow is maximal, there is at least 1 cut so that

- all forward are at capacity
- all backward are zero

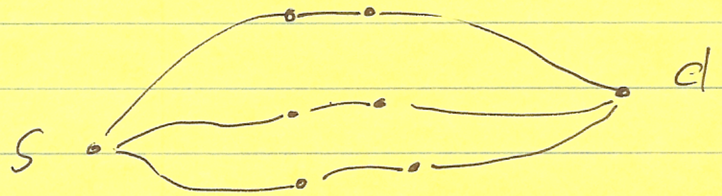
$\Leftarrow$ if such a cut exists, flow is maximal.

(Easy; there can be no augmenting path) □

$\Rightarrow$: consider all paths, $s \to d$



- Each has, at least, either an edge which is forward and at capacity or backward and at zero



(there might be more than one)

Now cut each path at some such edge.

→ is this a cut?

<u>No</u>: Because some verts could appear on both s-side and d side

<u>But</u>: we can get a cut out of it

assume $v_1 \in V(s)$, and $v_1 \in V(d)$

~~can't~~ move from $V(d)$ to $V(s)$

<u>because</u> otherwise there would be an augmenting path □

Notice

Value of this cut

= Max flow.

But there cannot be a cut with <u>lower</u> value, because then flow would be smaller.

(Getting a min-cut from a max-flow:

⇒ wasn't constructive (cycles)

cut all capacity, 0 edges.

→ if this is a cut, stop

→ otherwise, put other CC's into $V(s)$, $V(d)$ at will

→ restore edges in $V(s)$, $V(d)$

→ this is a cut)

Cuts, flows and linear programs

for a directed graph, write the incidence matrix A

$$
\begin{array}{cc}
 & \xrightarrow{\text{edges}} \\
\text{verts}\downarrow & \begin{bmatrix} a_{ij} = 0 \text{ - if } e_j \text{ not incident on } v_i \\ a_{ij} = 1 \text{ if } e_j \to v_i \\ \phantom{a_{ij} =}\ -1 \text{ if } e_j \leftarrow v_i \end{bmatrix}
\end{array}
$$

we will make row 1 cf to s
$\qquad\qquad\qquad\quad$ 2 $\quad$ to d.

Now f is a vector of flows

$$Af = \begin{bmatrix} v \\ -v \\ 0 \\ 0 \\ 0 \end{bmatrix} \vdots$$

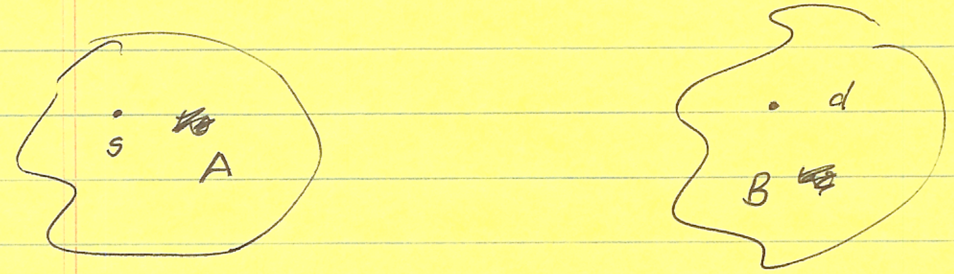
and max flow is

$$\max \ v$$

$$\text{st} \quad Af + v\begin{bmatrix} \cancel{s}\,-1 \\ \cancel{d}\,1 \\ 0 \\ \vdots \end{bmatrix} = 0$$

$$f \geq 0 \qquad\qquad f \leq c$$

<u>cuts</u> as a linear program



for each vert, x, $\pi(x) = \begin{cases} 1 & x \in A \\ 0 & x \in B. \end{cases}$

$\gamma(E) = \begin{cases} 1 & \text{if edge in cut} \\ 0 & \text{otherwise}. \end{cases}$

this means:

$$\gamma(x,y) + \pi(x) - \pi(y) \geq 0$$

(i.e. if $x \in A$, $y \in B$ $\gamma(x,y) = 1$)

$$\min \quad c'\gamma$$

st $\quad \pi_i \in \{0,1\} \qquad \pi_x - \pi_y + \gamma_{xy} \geq 0$

$\quad \gamma_e \in \{0,1\}$

$\pi_s = 1, \quad \pi_d = 0$

Notice similarity to 01 QP

Q: are they the same thing?