Flows, cuts, linear programs, quadratic programs

Recall setup:

we have a directed graph

$V = \{\text{vertices}\}$

$E = \{\text{edges}\}$

- two special verts: s (source) only outgoing edges
  d (drain) only incoming edges
- each edge has a capacity; label each edge (integer, ≥ 0) w/ flow
- flow through edge may not exceed capacity
- at a vertex v

$$\sum_{v_i} f_{v_i \to v} - \sum_{v_j} f_{v \to v_j} = 0$$

(Kirchoffs laws)

- flow ≥ 0 on each edge.

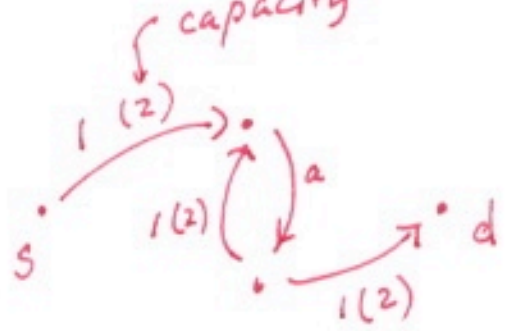
Q: what is max flow out of s?
(equiv: into d).

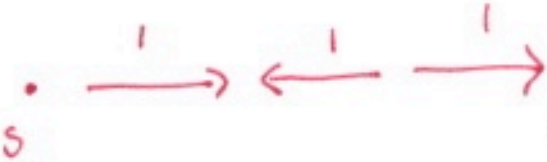
An elementary alg:

- assume we have a feasible flow f
- an augmenting path for this flow is an undirected path from s to d such that:
  - all forward arrows ($s \to d$) are below capacity
  - all backward arrows ($d \to s$) are greater than zero.

(Q: what is flow, capacity on a..?

eg. capacity





is aug.

(Q: why no probs w/ a?)

- if an augmenting path exists, flow is NOT maximal (Because we can increase flow along this path)
- if flow is maximal, there is no A.P. (Because if there were, we could increase flow).

Alg outline:

- find augmenting path
- max flow on this path,

Notice there is some housekeeping here

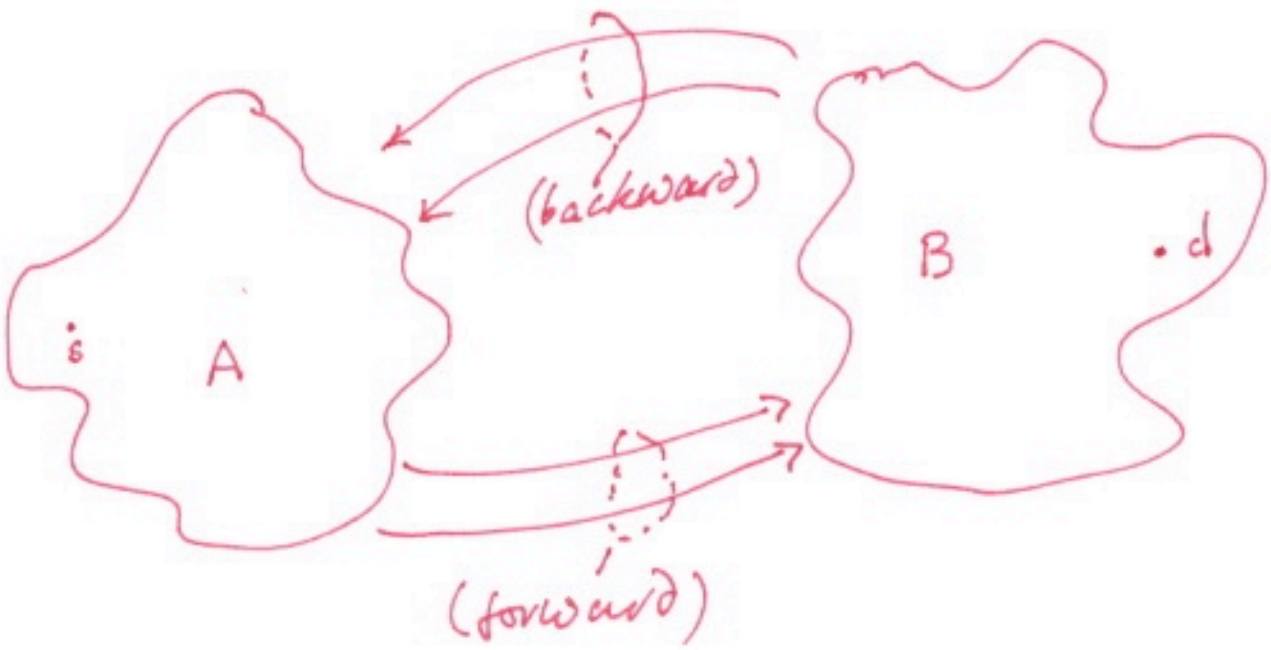
Notice also we are taking a feasible soln (original flow) and improving it; but always feasible cause always ⓐ meets ineq constraints ⓑ meets eq. constraints (Kirchoffs laws).

Now consider a <u>disconnecting cut</u>

- a set of edges that disconnects s, d.
- divides verts into A, B

st $A \cup B = V$

$A \cap B = \emptyset$ $\quad s \in A, d \in B$



the total flow from $s \to d$ is

$$\sum \text{forward} \quad - \sum \text{backward}.$$

we say the <u>value</u> of the cut is

$$\sum \text{capacities forward}.$$

<u>Thm</u>:

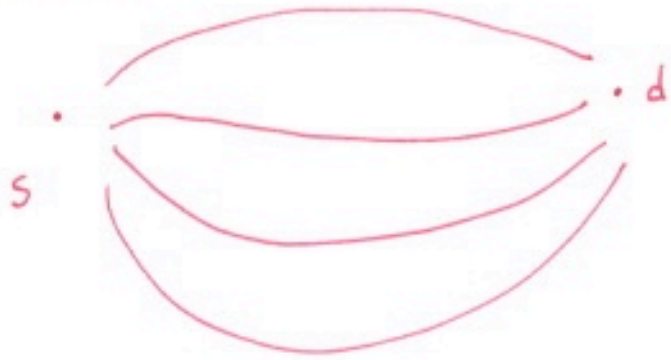
There is at least one cut so that
- all forward are at capacity
- all backward are zero

$\Leftrightarrow$ Flow is maximal.

$\Rightarrow$ easy; there can be no augmenting path $\square$

$\Leftarrow$

consider all paths, $s \to d$



- each has ~~a~~ either at least one edge forward and at capacity or at least one edge backward and w/ flow 0 (else there would be an augmenting path)

~~now cut each of these edges~~

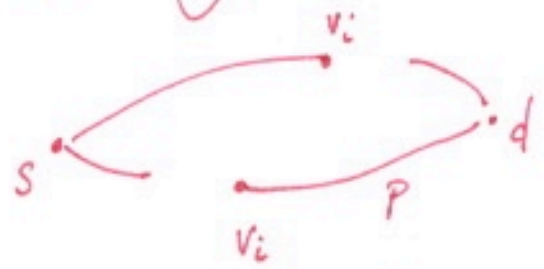
- now cut each path at some such edge.

Q: is this a cut?

A: No, cause some verts could be on both S-side and t-side

Now assume $v_i \in V(s)$ and $v_i \in V(d)$

after cutting



we can move it from V(d) to V(s) w/o problems.

(Why: – there must be a downstream edge on P that we could have cut, but didn't, else there is an aug path!)

. restore edges as required.

Notice: value of this cut

=

$\sum$ capacities forward

=

Max flow

Notice

there cannot be a cut w/ lower value because if there were, the flow would be smaller.

In linear programming language:

for directed graph, write an incidence matrix A

edges →

verts ↓

$$a_{ij} = \begin{cases} 0 & \text{- if } e_j \text{ not incident on } v_i \\ 1 & \text{- if } e_j \rightarrow v_i \\ -1 & \text{- if } e_j \leftarrow v_i \end{cases}$$

- arrange A so row 1 is s
  row 2 is d

Can write max flow as

$$\max \quad v$$

$$\text{st} \quad Af + v\begin{bmatrix} -1 \\ 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} = 0 \quad \leftarrow \text{Kirchoffs laws}$$

$$f \geq 0$$

$$f \leq c$$

f: flow

c: vector of capacities

Another LP version

A_R = [all rows of A but the first two]

w^T = second row of A, csp to target / drain vertex.

$$\max \quad w^T f$$

$$\text{st} \quad A_R f = 0 \qquad \text{(Kirchoff's laws}$$

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

Thm:

A, A_R are TUM.

Proof: (induction)

- consider some $K \times K$ minor
- 3 cases:
  - some col is all zeros
    $\longrightarrow \det(B) = 0$
  - B has 1 col that has exactly 1 non-zero

$$\longrightarrow \det(B) = \det\begin{bmatrix} 1 & 6^T \\ 0 & \tilde{B} \\ \vdots & \end{bmatrix} \times (\pm 1)$$

so $\det(B) = \pm \det(\tilde{B})$.

  - B has all cols with 2 non-zeros

$\longrightarrow 1^T B = 0$, $\det B = 0$

Careful This argument does not mean that $\det(B)$ is always zero!

Obtain the Dual :

L.P.

$$\max \; w^T f$$
$$\text{st} \quad A_R f = 0$$
$$c - f \geq 0 \qquad f \geq 0$$

Dual:

$$Q(y, z, u) = \sup_f \left[w^T f - y^T A_R f + z^T f + u^T (c - f) \right]$$

$$= \begin{cases} u^T c & \text{if } \left[w^T - y^T A_R + z^T - u^T = 0 \right] \\ \infty & \text{otherwise} \end{cases}$$

Where $z \geq 0$, $u \geq 0$

So $\min \; u^T c$

$$\text{st} \quad w - A_R^T y + z - u = 0$$

$$z \geq 0$$

$$u \geq 0$$

we can clean up to:

$$\min \; u^T c$$

$$\text{st} \quad A_R^T y + u \geq w$$

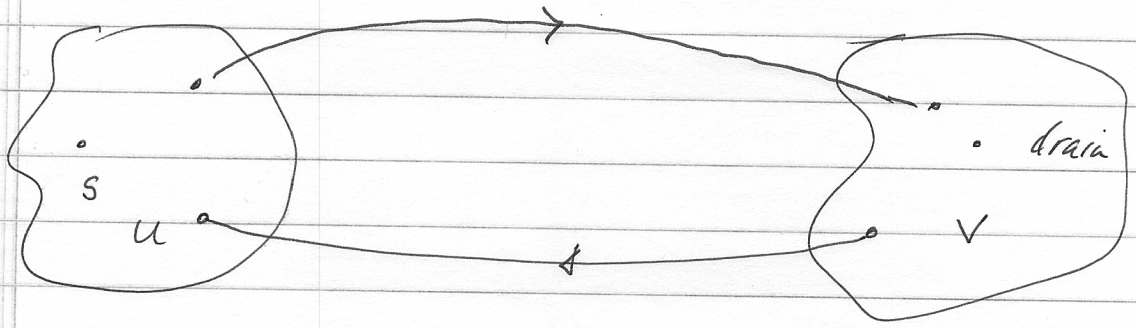
$$u \geq 0$$

(z is non-neg.

At a solution to this dual, y and u are integer

Min-cut = easy 0-1 QP

- Consequence - if we have an easy 0-1 QP, we can do it with any fast min-cut alg we have.



have x, 1 per vertex:

u: $x_i = 0$

v: $x_i = 1$

$$\text{value of cut} = \sum_{\substack{a \in \text{edges} \\ \text{from} \\ u \text{ to } v}} c_a$$

$$= \sum_{\text{all edges}} (1 - x_i)(x_j) \cdot C_{i \to j}$$

So we have

$$\min \sum_{i,j} (1 - x_i)(x_j) C_{i \to j}$$

$$\text{s.t.} \quad x \in \{0, 1\}$$

But this is

$$\max \sum_{i,j} (x_i - 1) x_j C_{i \to j}$$

$$\text{st} \quad x \in \{0, 1\}$$

and we have

$$\max \quad x^T \underbrace{\frac{A}{2}}_{\uparrow} x + b^T x$$

non neg

$$\text{st} \quad x \in \{0, 1\}.$$