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 w/ flow (integer, ≥ 0)

. 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.

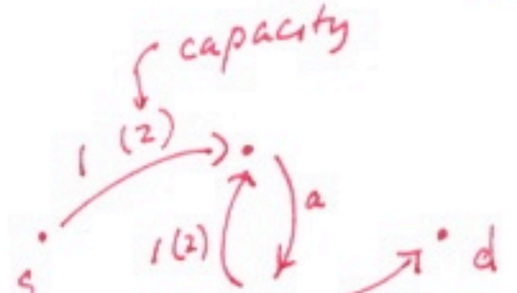
Q: what is max flow out of s?
(equiv: into d).

<u>An elementary alg</u>:

- 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.



$$s \xrightarrow{1} \xleftarrow{1} \xrightarrow{1} d$$

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

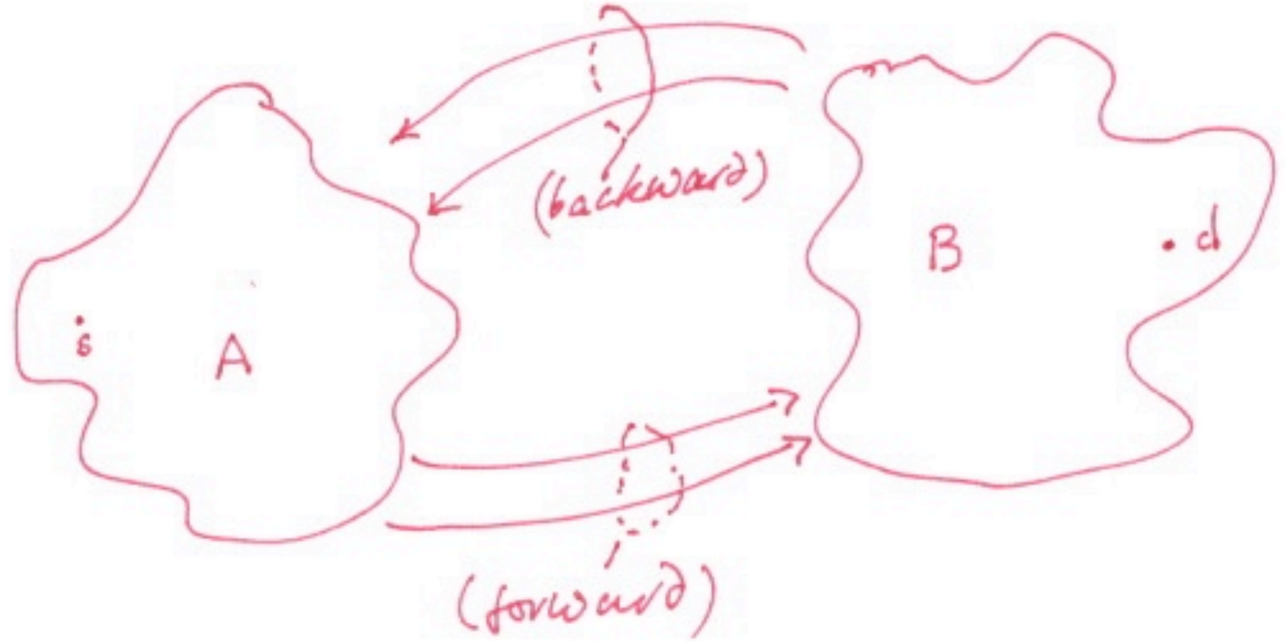
Notice also we are taking
a feasible soln (original flow)
and improving it; but always feasible
cause always (a) meets ineq constraints
(b) meets eq. constraints (Kirchoffs laws).

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>:

$$\left[\begin{array}{l}\text{There is at least} \\ \text{one cut so that} \\ \cdot \text{ all forward are} \\ \quad \text{at capacity} \\ \cdot \text{ all backward are} \\ \quad \text{zero}\end{array}\right] \iff \left[\begin{array}{l}\text{Flow is} \\ \text{maximal.}\end{array}\right]$$

$\Rightarrow$ 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)

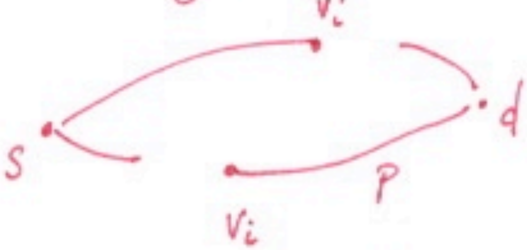
- ~~now cut each of these edges~~
- now cut each path at some such edge.

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.

<u>Notice</u> : 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 $\longrightarrow$

verts $\downarrow$

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

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

Can write max flow as

$$\max \quad v$$

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

$$f \geq 0$$

$$f \leq c$$

vector of capacities

flow

Another LP version

$$A_R = \begin{bmatrix} \text{all rows of} \\ \text{A but the first} \\ \text{two} \end{bmatrix}$$

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

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

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

$$f \geq 0 \quad f \leq c$$

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
    $\longrightarrow \det(B) = \det\begin{bmatrix} 1 & b^T \\ 0 & \tilde{B} \\ 0 & \\ \vdots & \end{bmatrix} \times (\pm 1)$
    so $\det(B) = \pm \det(\tilde{B})$.
  - B has all cols with 2 non-zeros
    $\longrightarrow 1^T B = 0$, $\det B = 0$

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

Obtain the Dual :

L.P.

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

Dual :

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

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

where $z \geq 0$, $u \geq 0$

So $\min \; u^T c$

st $\; w - A_R^T y + z - u = 0$

$z \geq 0$

$u \geq 0$

we can clean up to :

$\min \; u^T c$

st $\; A_R^T y + u \geq w$

$u \geq 0$ (z is non-neg.

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

constraints are

$$\begin{bmatrix} \mathrm{Id} & A_R^T \end{bmatrix} \begin{bmatrix} y \\ u \end{bmatrix} \geq w$$

but $\begin{bmatrix} \mathrm{Id} & A_R^T \end{bmatrix}$ is TUM (cause minors are minors of A_R which is TUM)

and w is integer

now consider a new variable

$$\hat{u}^T = [\underbrace{0}_{\text{source}};\ \underbrace{-1}_{\text{drain}};\ \underbrace{u}_{1 \text{ per vert}}]^T$$

we have

$$\begin{aligned} \hat{u}^T A = \ & 0 \times \text{first row of } A \\ & + -1 \times \text{second row of } A \\ & + u^T A_R \\ = \ & -w^T \\ & + \\ & + u^T A_R \end{aligned}$$

so that

$$y^T + \hat{u}^T A = y^T - w^T + u^T A_R$$

but $y^T + u^T A_R \geq w$

so $y^T + \hat{u}^T A \geq 0$

We can interpret a soln to the dual as a cut.

at soln, look at $\hat{u}$ values

$U = \{ \text{verts st } \hat{u}_v \geq 0 \}$ but not drain.

contains source,

The value of the cut is

$$\sum_{\substack{i \in \text{edges} \\ \text{leaving} \\ u}} c_i = c(\delta^{out}(u))$$

Now if we can show that

$$c(\delta^{out}(u)) \leq y^T c$$

this is the max flow value cause soln's of primal and dual have same value.

we have that $\hat{u}$ is a min cut
(because $c(\delta^{out}(u)) \geq$ flow for any flow)

Notice our dual vars y, $\hat{u}$ can be interpreted

$$y^T + \hat{u}^T A \geq 0$$

$$\xrightarrow{\text{edges}}$$

verts | Incidence

this means there is one y per edge.
(already seen one $\hat{u}$ per vert).

$\delta^{out}(u)$ = set of edges leaving u.

- want to show $y^T c \geq c(\delta^{out}(u))$
- enough to show that for each edge a leaving u, $y_a \geq 1$
- then $y^T c \geq \sum_a c_a = c(\delta^{out}(u))$

Now consider

$$a : \underset{\underset{u}{\in}}{i} \longrightarrow \underset{\underset{v}{\in}}{j}$$

an edge that leaves u.

must have

$\hat{u}_i \geq 0$ (because its in u)

$\hat{u}_j \leq -1$ (integer, and less than zero.

$$y^T + \hat{u}^T A \geq 0$$

implies

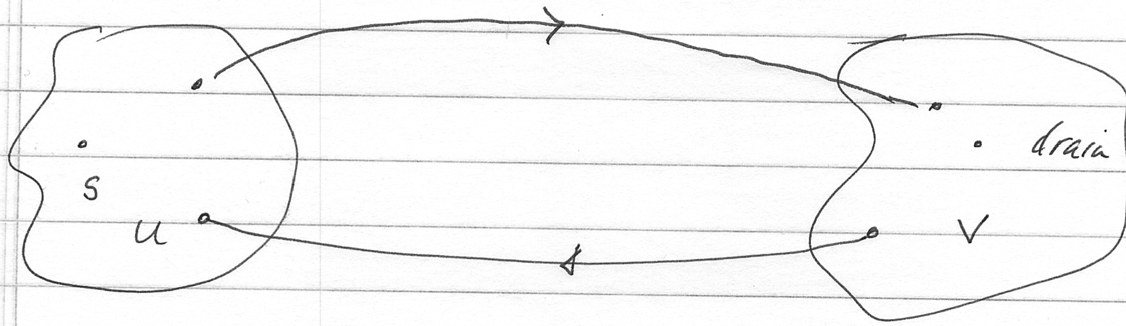
$$y_a + \hat{u}_j - \hat{u}_i \geq 0 \quad \text{(recall defn of } A\text{)}$$

$$y_a \geq \hat{u}_i - \hat{u}_j \geq 1$$

So our dual vars give a cut!

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 <u>vertex</u>:

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

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

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

<u>But</u> this is

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

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

<u>and we have</u>

$$\max \quad x^T \underset{=}{A}\, x \quad + \quad b^T x$$

$$\uparrow$$

non neg

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