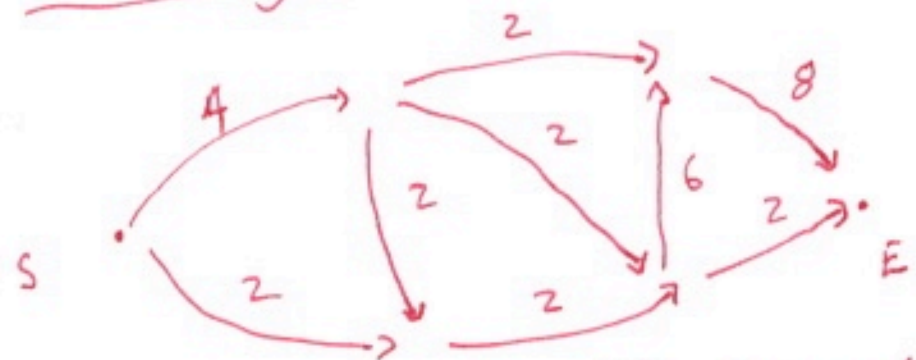
Discrete or Combinatorial optimization:

Many examples:

- Intrinsically combinatorial



A railway network, directed from (say) producer (S) to consumer (E); with capacities

- Cut edges to disconnect S and E
- Find the largest flow between S and E

- As an optimization problem

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

$$\text{st} \quad x \in \{0,1\}^n$$ i.e. 0-1 vectors

$$Cx = d$$

or even

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

$$\text{st} \quad x \in \mathbb{Z} \quad \leftarrow \text{integers}$$

$$Cx = d$$

Some tools:

Dynamic programming.

(setup in AML-notes)

easy case:

$$\begin{aligned} \text{cost}(x_1 \dots x_K) &= F(x_1, x_2) + F(x_2, x_3) + \dots F(x_{K-1}, x_K) \\ &\quad + G(x_1) + G(x_2) + \dots G(x_K) \end{aligned}$$

Q: find $x_1 \dots x_K \in \mathbb{Z}^K$ to max cost.

A: easier with a picture.

A trellis

- one col for each x_K; one node for each value of x_K
- label nodes w/ $G(x_i = m)$

  m'th node in i'th col gets this.

  directed edges w/ $F(x_i = m, x_{i+1} = n)$

  m'th node in i'th col to n'th node in $i{+}1$'th

- Then: the cost of directed path $1 \to K$
  $=$ cost of ($x_1 = a$, $x_2 = b$, etc).

($\to$ AML Notes

Pretty clearly, this won't work
~~for~~

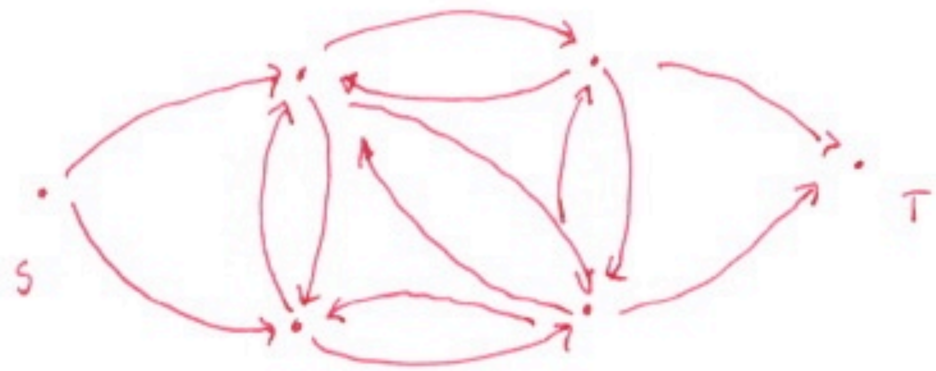
$$F(x_1, x_2) + F(x_2, x_3) + F(x_1, x_3)$$

(try it!)

Criterion

- each var gets a node
- ~~if~~ each pair(i,j) of vars gets an edge if there is an $F(x_i, x_j)$ term
- Resulting graph is a forest $\Leftrightarrow$ D.P. will work.

An abstracted picture of a railway network from S to T



each edge has a capacity, the most stuff you can move along that edge

- at each node, the amount of stuff arriving ~~can~~ must be the same as the amount leaving – no storage! [Kirchoff's laws]
- Q: how much stuff can go from S to T?

Another case:

⑤

- Assume we have a directed graph G, with two distinguished vertices

  S - source, only outgoing edges
  T - target, only incoming edges

- Each edge e has a non-neg capacity $c_e \geq 0$
- A flow from S to T is a labelling of each edge e with a weight w_e such that:

  - Kirchoffs laws are satisfied at all but S and T

  - $w_{e_i} < c_{e_i}$

We can set this up as a linear program ⑥

- each edge $(u \to v)$ gets $\omega_{u \to v}$
- Kirchoffs laws apply, except at S, T

$$\sum_u \omega_{u \to v} - \sum_w \omega_{v \to w} = 0$$

for all v not S, T

- $\omega_{u \to v} \leq c_{u \to v}$
- $\omega_{u \to v} \geq 0$

Total flow is:

$$\sum_v \omega_{S \to v}$$

[Recall Kirchoffs laws !

Now imagine trying to cut this network. ⑦

- Construct two sets of ~~edges~~ vertices

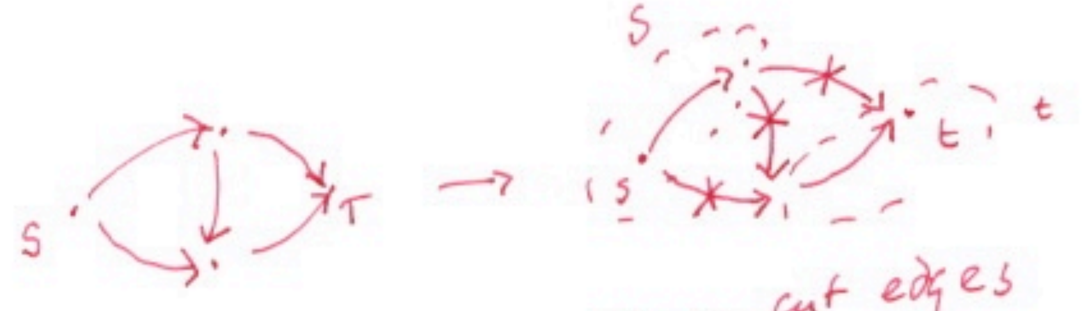
$$S \quad , \quad T$$

where $S \cup T = V$

$S \cap T = \phi$

$s \in S \quad , \quad t \in T$

~~AND there~~ By cutting edges from S to T



x - cut edges

Do this by cutting a set of edges (8)
w/ minimal capacity — min cut.

One formulation:

$x_v \quad \leftarrow \quad$ one per vertex,

$$x_s = 1, \quad x_t = 0$$

$$x_v = 1 \Leftrightarrow v \in S$$
$$x_v = 0 \Leftrightarrow v \in T$$

Cost function

$$\sum c_{uv}(x_u)(1 - x_v)$$

Constraints:

$$x_u \in \{0,1\}$$

$$x_s = 1$$
$$x_t = 0$$

Notice the cost is quadratic.

but if
$x_u \in \{0,1\}$ $x_v \in \{0,1\}$

then $x_v x_u \in \{0,1\}$.

notice:
$$x_u x_v \leq x_u$$
$$x_u x_v \leq x_v$$
$$x_u x_v \geq x_u + x_v - 1$$

So make a new variable $q_{uv} \in \{0,1\}$
which represents x_{uv}

leading to

$$-\sum_{uv} c_{uv} q_{uv} + \left[\sum_{u} c_{uv} x_u\right](n_v - 2)$$

s.t.

$$q_{uv} \in \{0,1\}, \quad x_u \in \{0,1\}, \quad x_v \in \{0,1\}$$

$$q_{uv} - x_u \leq 0$$

$$q_{uv} - x_v \leq 0$$

$$q_{uv} - x_u - x_v + 1 \geq 0$$

Now what?

Turns out, rather remarkably, this is easy.