18.453 Lecture 3

Lecture plan

1. Min-weight perfect matching

2. Linear/integer program formulation

3. Primal-Dual algorithm.

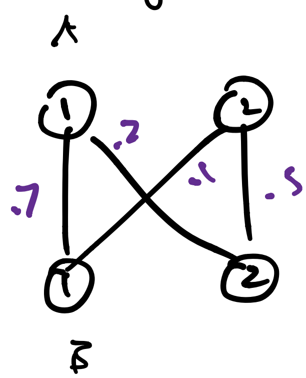
Springer - my copy (Korte book)
24 Euros

Minimum Weight Perfect Matching (MWPM)

Consider bipartite graph with $|A| = |B| = \frac{n}{2}$

edge ij costs $c_{ij} \in \mathbb{R}$.

E.g.



$$\begin{bmatrix} c_{11} & c_{12} \\ c_{21} & c_{22} \end{bmatrix} = \begin{bmatrix} .7 & .2 \\ .1 & .5 \end{bmatrix}$$

Goal: find matching M in G of least cost

$$c(M) := \sum_{ij \in M} c_{ij}.$$

Exercise: can reduce cardinality mathing to MWPM.

by allowing $c_{ij} = \infty$, can assume G is complete bipartite graph.

Application: n machines, n tasks, costs c_{ij} for machine i to do task j.

Today: Hungarian algorithm

- uses linear programming
- is strongly polynomial time: # steps independent of sizes of c_{ij}; polynomial in n.

Linear/integer programs

- First, express problem as <u>integer program</u>.
- Associate vector with matching.

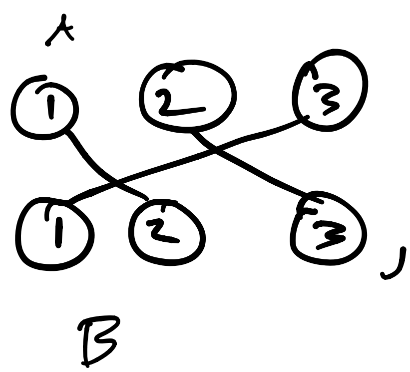
<u>incidence vector</u> of matching M is vector x s.t.

$$x_{ij} = \begin{cases} 1 & \text{if } ij \in M \\ 0 & \text{else.} \end{cases}$$

(confusingly, also a matrix)

E.g.

$M =$



$$X = \begin{bmatrix} x_{11} & x_{12} & x_{13} \\ x_{21} & x_{22} & x_{23} \\ x_{31} & x_{32} & x_{33} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix}$$

note: X permutation matrix.

Integer program: (IP)

min-weight perfect matching has cost

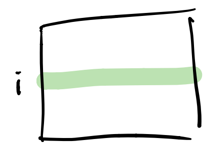
$$\min \sum_{ij} c_{ij} x_{ij}$$

objective

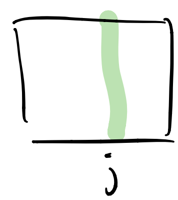
subject to

constraints

$$\sum_{j} x_{ij} = 1 \quad \forall i \in A$$



$$\sum_{i} x_{ij} = 1 \quad \forall j \in B$$



$$x_{ij} \geq 0 \quad \forall i \in A, j \in B$$

$$\boxed{x_{ij} \in \mathbb{Z} \quad \forall i \in A, j \in B}$$

not linear program!

Any solution to IP is valid matching & vice versa.

Linear program (LP)

Get linear program (P) by dropping integrality constraint.

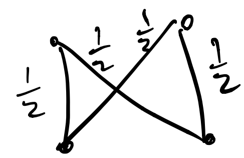
$$
(P) \qquad
\begin{array}{ll}
\min & \sum c_{ij} x_{ij} \\
\text{subject to} & \left.\begin{array}{ll}
\sum_j x_{ij} = 1 & \forall i \in A \\
\sum_i x_{ij} = 1 & \forall j \in B \\
x_{ij} > 0 & \forall i \in A, j \in B
\end{array}\right\} \text{constr.} \\
& (x_{ij} \in \mathbb{R})
\end{array}
$$

Called the linear programming **relaxation** of the integer program.

Say x **feasible** if satisfies constraints.

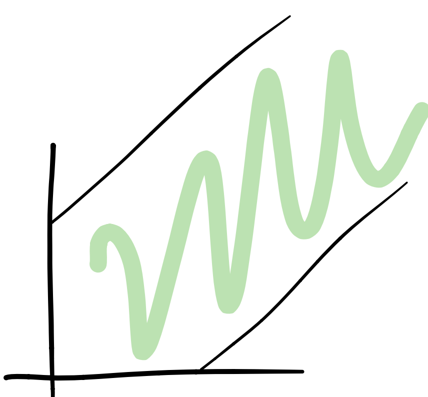
In contrast to IP: not all feasible x are matchings!

x_{ij} can be fractional.

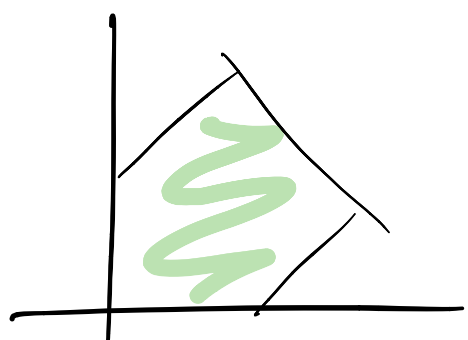


$x = \begin{bmatrix} \frac{1}{2} & \frac{1}{2} \\ \frac{1}{2} & \frac{1}{2} \end{bmatrix}$, is feasible.

Set of feasible solution is a **polytope** (bounded polyhedron).



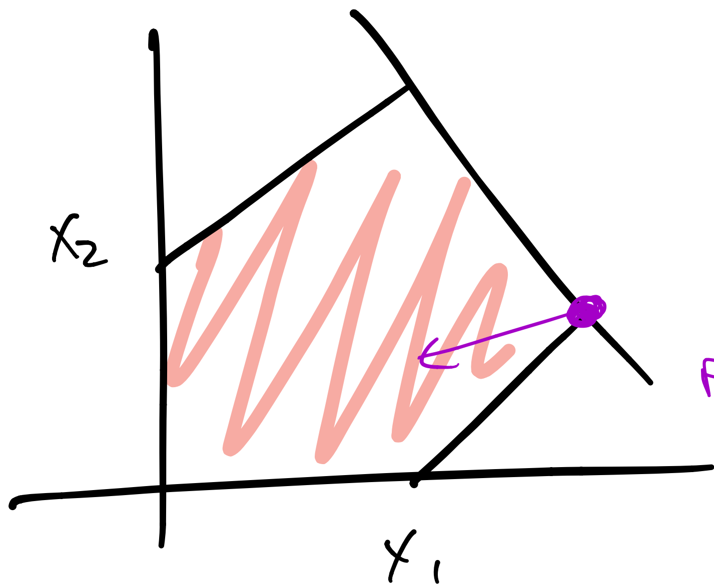
polyhedron



polytope

optimum of a linear function will occur at an <u>extreme point</u> (corner).

<u>E.g.</u> if $c = \leftarrow$

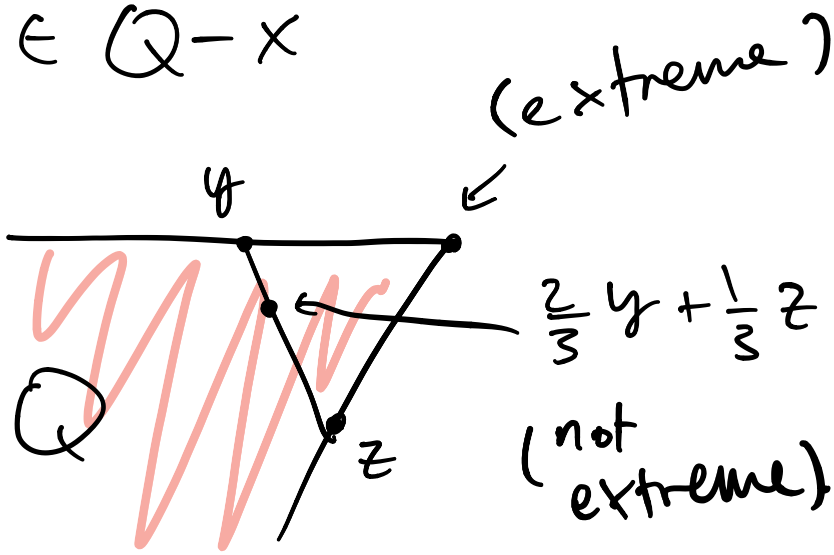


minimizes $c \cdot x$ over polytope.

Extreme point x of set Q is point that can't be written as

$$\lambda y + (1-\lambda) z \ , \ \lambda \in (0,1)$$

for $y, z \in Q - x$



(more on this when we get to polyhedral combinatorics).

In general, extreme points need not be integral (even if constraints all have coefficients in $\{0,1\}$.)

No surprise: L.P. solvable in polynomial time, I.P. NP-hard.

Say z_{IP} = value of some IP

z_{LP} = value of its relaxation.

In general

$$z_{IP} \neq z_{LP}.$$

But! IP is more constrained, so

$$\boxed{z_{IP} \geq z_{LP}.}$$

for minimization problems.

Moreover: if x is optimum for LP, and x integral, then x opt for IP!

Exercises: 1. prove this

* 2. find example where $z_{IP} \neq z_{LP}$.

For perfect matching, we are **lucky**! Constraints special.

Consider the polytope P

cut out by constraints of (P).

$$P = \left\{ x \text{ s.t. } \sum_j x_{ij} = 1 \quad \forall i \in A \quad \sum_i x_{ij} = 1 \quad \forall j \in B \quad x_{ij} \geq 0 \quad \forall i \in A, j \in B \right\}$$

Theorem: every extreme point of P is integral.

(in particular, is a 0-1 vector and hence is the incidence matrix of p.m.).

We give 2 proofs:

1. algorithmic (today)
2. algebraic (later); uses total unimodularity

First: duality for LP's.
(informal version).

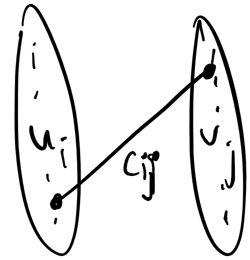
LP duality

Dual of (P): family of obstructions for (P) to have small value.

Recall:

$$(P)\quad \begin{aligned} \min \quad & \sum c_{ij} x_{ij} \\ \text{subject to} \quad & \sum_j x_{ij} = 1 \quad \forall i \in A \\ & \sum_i x_{ij} = 1 \quad \forall j \in B \\ & x_{ij} \geq 0 \quad \forall i \in A, j \in B \end{aligned}$$

obstruction: values



$$u_i \quad i \in A,$$

$$v_j \quad j \in B$$

$$\text{s.t.} \quad u_i + v_j \le c_{ij} \quad \forall i \in A, \; \forall j \in B.$$

Then: for any matching M,

$$\underbrace{\sum_{ij \in M} c_{ij}}_{} \ge \sum_{ij \in M} u_i + v_j = \boxed{\sum_{i \in A} u_i + \sum_{j \in B} v_j}$$

$$z_{IP} \ge (D)$$

this value is our obstruction.

want to <u>maximize</u> this value; doing this gives us the dual (D) of (P).

$$\boxed{\begin{array}{ll} & \max \sum_{i \in A} u_i + \sum_{j \in B} v_j \\ (D) & u_i + v_j \leq c_{ij} \quad \forall i \in A \\ & \qquad\qquad\qquad \forall j \in B. \end{array}}$$

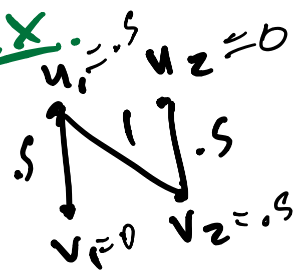
In fact, $\sum_{i \in A} u_i + \sum_{j \in B} v_j$ is not just a lower bound on $c(M)$, but on (P).

$(z_{IP} \geq z_{LP} \geq (D))$

Indeed, we can calculate:

$$\sum_{ij} c_{ij}x_{ij} \geq \sum_{ij}(u_i+v_j)x_{ij}$$

Ex.



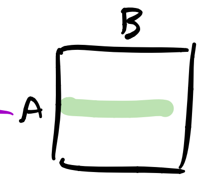
$u_1+v_1 \leq .5$

$u_1+v_2 \leq 1$

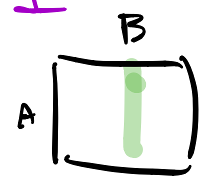
$u_2+v_2 \leq .5$

$$= \sum_{i\in A}\sum_{j\in B} u_i x_{ij} + \sum_{ij} v_j x_{ij}$$

$$= \sum_{i\in A} u_i \Big(\overbrace{\sum_{j\in B} x_{ij}}^{1}\Big)$$



$$+ \sum_{j\in B} v_j \Big(\overbrace{\sum_{i\in A} x_{ij}}^{1}\Big)$$



(constraint) $\Rightarrow$ $$= \boxed{\sum_{i\in A} u_i + \sum_{j\in B} v_j}.$$

Construction (P) $\rightsquigarrow$ (D) is example of more

general 'recipe' for taking dual of LP's.

Summary:

primal linear program

$$\boxed{\min_{M \text{ perfect matching}} \sum_{ij \in M} c_{ij}} \geq \boxed{\min_{x \in P} \sum c_{ij} x_{ij}}$$

integer program

dual linear program

$$\geq \boxed{\max_{x \in D} \sum_{i \in A} u_i + \sum_{j \in B} v_j}$$

D polyhedron. (D).

Recall: used $\sum_{ij \in M} c_{ij} \geq \sum_{ij \in M} u_i + v_j$

$\boxed{\text{When equality ??}}$

M must only have edges (i,j) s.t.

$$c_{ij} = u_i + v_j .$$

"complementary slackness" in LP lingo.

- Let $w_{ij} := c_{ij} - u_i - v_j$.
- Are matchings on $\{(i,j) : w_{ij} = 0\}$, but no guarantee they are perfect.
- primal-dual alg uses such (non-perfect matchings) to update dual solution u_i, v_j.

Primal-Dual

Outline:

1. start w/ any dual feasible solution

(need $u_i + v_j \le c_{ij}$)

$$u_i = 0, \quad v_j = \min_i c_{ij}$$

Repeat the following until done:

2. In any iteration, alg. has dual feas. soln. (u, v, w)
$w_{ij} = c_{ij} - u_i - v_j$

3. Want a matching on
$E = \{(i,j) : w_{ij} = 0\}$.

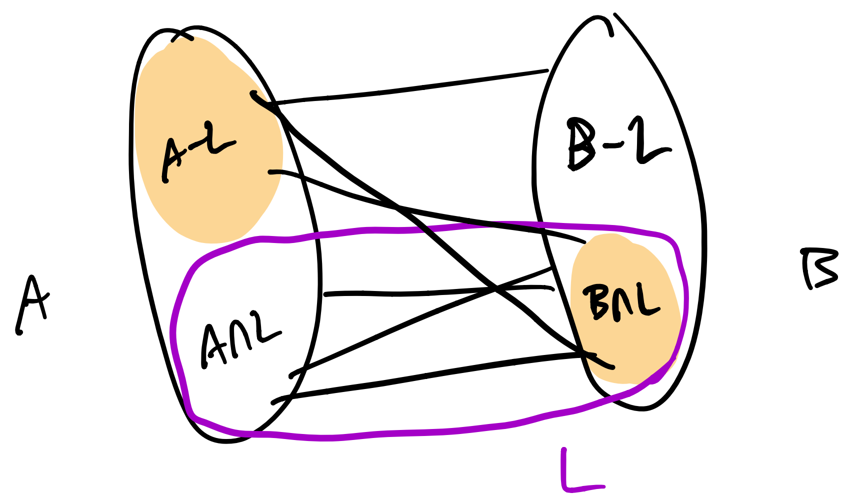
Use cardinality matching alg. to output largest matching M in E.

- If M perfect, is optimal by complementary slackness. ✓
- If not, use the vertex cover output mg alg. to find new dual feasible soln w/ larger value. ✓

Details of Step 3 :

- Suppose M not perfect.
- Recall set L output by the aug. paths algorithm.

C^*

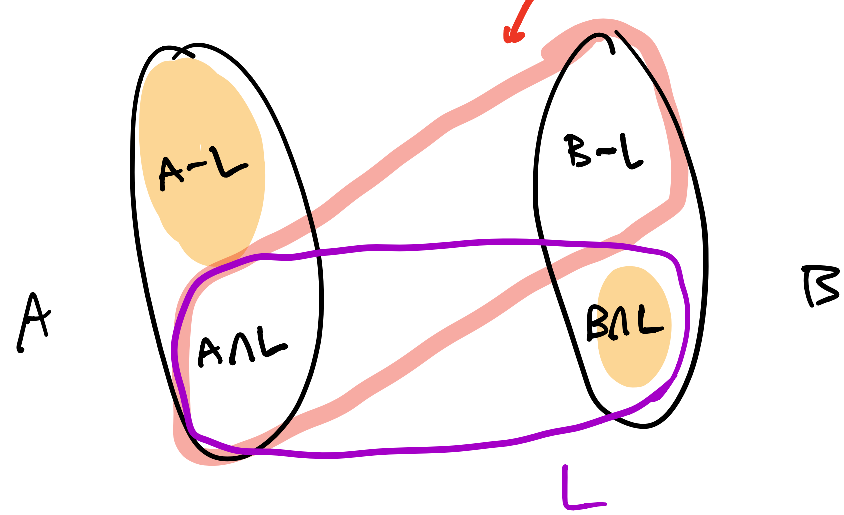


$C^* = (A-L) \cup (B \cap L)$ is optimal vertex cover of E.

In particular:

C^*

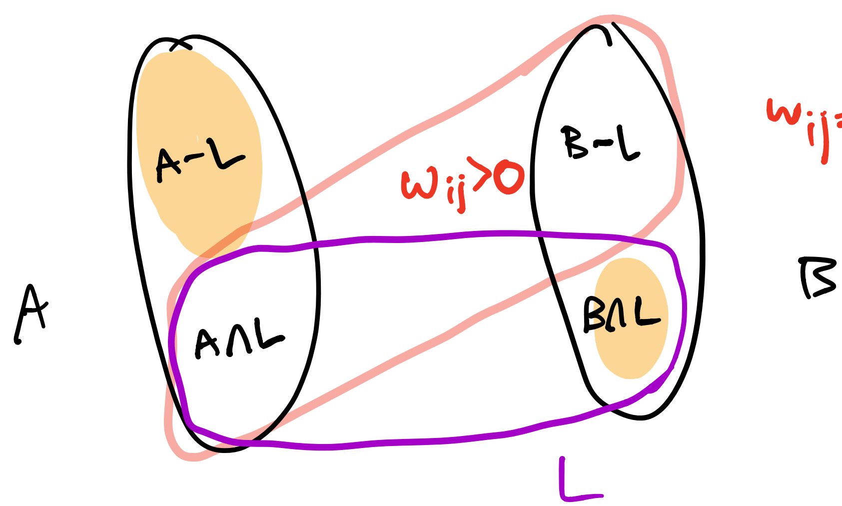
no edges of E.



Equivalently: $w_{ij} > 0$ for $i \in A \cap L$, $j \in B - L$.

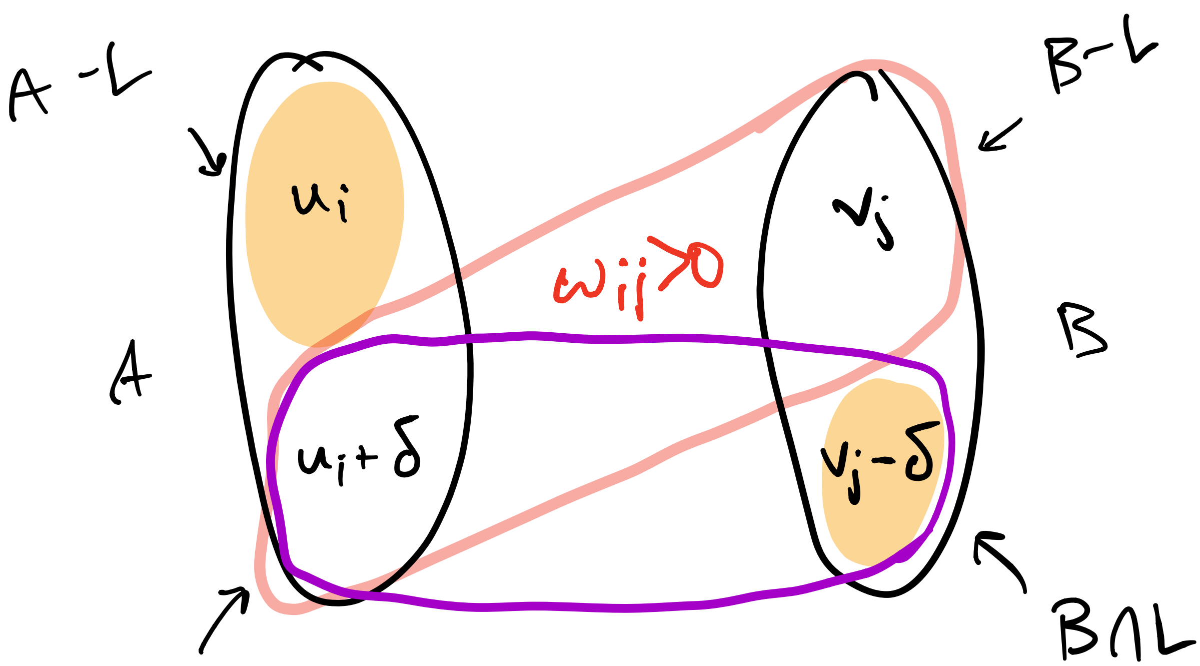
$E = \{(i,j) : w_{ij} = 0\}$.

$w_{ij} = c_{ij} - u_i - v_j > 0$.



Updating u,v : Set

$$\boxed{\delta = \min_{\substack{i \in (A \cap L) \\ j \in (B - L)}} w_{ij},} \quad (\delta > 0)$$



constraints: $u_i + v_j \le c_{ij}$

formally:

$$u_i := \begin{cases} u_i & i \in A - L \\ u_i + \delta & i \in A \cap L \end{cases}$$

Network simplex:

$$v_j := \begin{cases} v_j & i \in B - L \\ v_j - \delta & j \in B \cap L. \end{cases}$$

New solution is feasible!

New value? $\sum_{i \in A} u_i + \sum_{j \in B} v_j$

$$\text{New} - \text{OLD} = \delta\big(|A \cap L| - |B \cap L|\big).$$

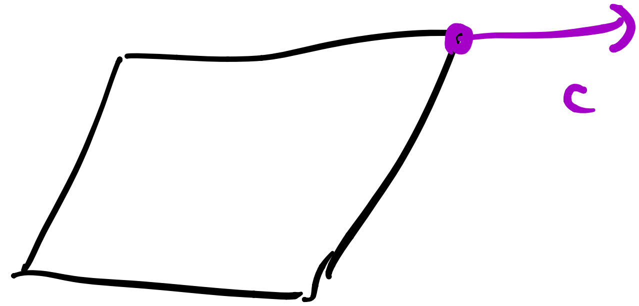
$$= \delta\big(\underbrace{|A \cap L| + |A - L|}_{|A|} - \underbrace{|A - L| - |B \cap L|}_{|C^*|}\big)$$

$$= \delta\Big(\underbrace{\frac{n}{2} - |C^*|}_{\geq 1}\Big) \geq \delta.$$

Thus, dual value increases!
Repeat until termination -
then M is perfect; done.

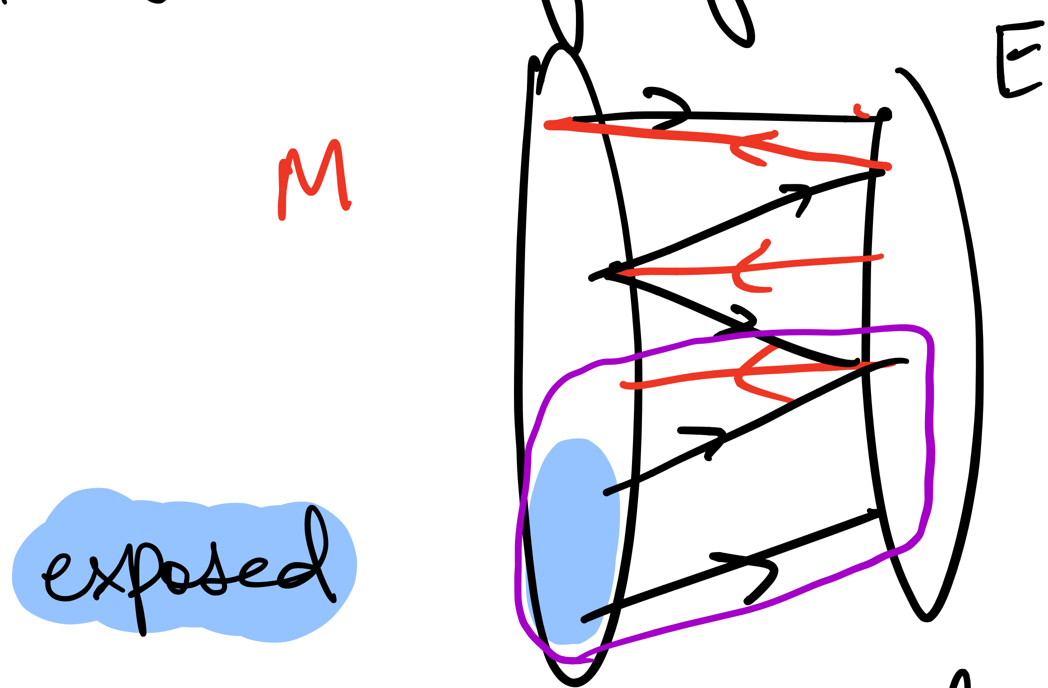
Proves Theorem: for any extreme pt x^*, can choose c to make x^* unique optimum.

$z_{IP} = z_{LP}$ for all c $\iff$ P has integral extreme pts.



Termination? how do we know it terminates?

Recall def of L.

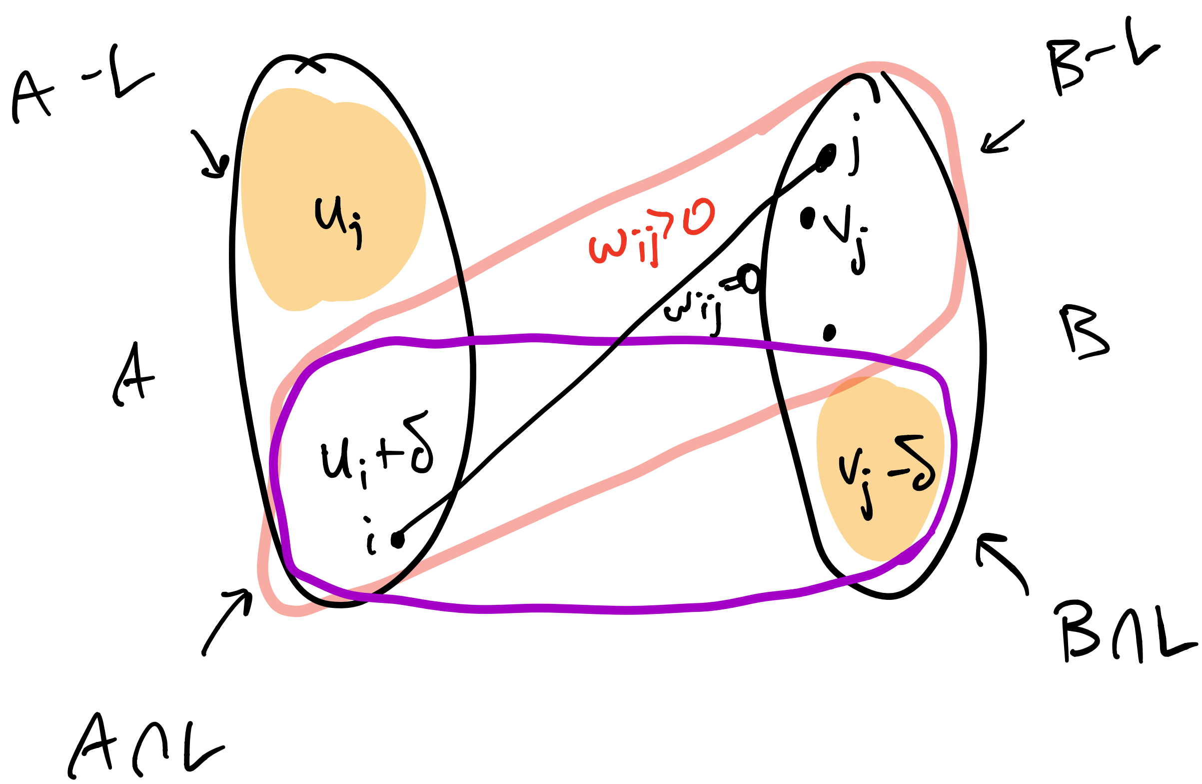


everything reachable from exposed in A.

Claim: New vertex $j \in B$ reachable.

$$\boxed{\text{for } \underline{\text{some }} i \in A \cap L,\ j \in B - L,\ w_{ij} = 0 \text{ by our choice of } \delta.}$$

$E = \{(i,j) : w_{ij} = 0\}$



thus, in $\leq \frac{n}{2}$ iterations either

Analysis:

- "<u>outer loop</u>": matchings M.
  if M not perfect, is exposed vertex in B
  - "<u>Inner loop</u>:"
    each time u, v change, ≥ 1 edge added to E & ≥ 1 new vertex of B reachable. Thus need to change dual $\leq \frac{n}{2}$ times before exposed vertex reached. Once this happens, can increase $|M|$.

find new <u>larger</u> M; either M perfect (done) or re-enter inner loop.

outer loop can happen $\leq \frac{n}{2}$ times
inner loop happens $\leq \frac{n}{2}$ times per outer loop

$\frac{n}{2} \cdot \frac{n}{2} \simeq O(n^2)$ iterations

Total running time

$$O(n^4)$$

b/c takes $O(n^2)$ time to compute L. $\square$

Exercise: By tracking more carefully how L changes, show $O(n^3)$.

Remark: strongly polynomial time: poly in n, assuming arith. operations free.*

* and that space to run algorithm is poly in # input bits.