Lecture 6

Wednesday, April 12, 2017 1:57 PM

Facility Location Problem

Input:
- $X = (F \cup C, d)$ metric space
- $f : F \to \mathbb{R}_{\geq 0}$ facility opening costs

Output:
- $A \subseteq F$ set of open fac
- $\sigma : C \to A$ assgn clients to open fac.

Obj: Minimize $\sum_{i \in A} f_i + \sum_{j \in C} d(j, \sigma(j))$

A "simpler" version of the k-median problem.

LP-relxn

i is open or not

j goes to i or not.

$$\min \sum_{i \in F} f_i y_i + \sum_{j \in C} d_{ij} x_{ij}$$

(1) $\forall j \in C : \sum_{i \in F} x_{ij} \geq 1$ → all clients are assigned

(2) $\forall i \in F, j \in C : x_{ij} \leq y_i$ → clients go to open facilities

$$0 \leq \bar{x}, \bar{y} \leq 1$$

Algorithm:

Algorithm:

1. Solve LP to get (x, y).

$$C_j := \sum_{j \in C} d_{ij} x_{ij}$$

$$LP = F^* + C^* \quad \text{where} \quad F^* = \sum_i f_i y_i, \quad C^* = \sum_j C_j$$

2. $\boxed{\text{Filtering}}$

- Order clients $C_1 \leq C_2 \leq \cdots \leq C_m$
- For each $j \in C$, define

$$F_j = \{ i \mid d_{ij} \leq 2 C_j \}$$

Claim: $\displaystyle\sum_{i \in F_j} y_i \geq \frac{1}{2}, \ \forall j$

Pf: $\forall j$, $\displaystyle \cancel{C_j} \geq \sum_{i \notin F_j} d_{ij} x_{ij} > 2 \cancel{C_j} \sum_{i \notin F_j} x_{ij}$

$$\Rightarrow \forall j, \ \sum_{i \notin F_j} x_{ij} < 1/2 \ \underset{LP(1)}{\Rightarrow} \ \sum_{i \in F_j} x_{ij} > 1/2$$

$$\underset{LP(2)}{\Rightarrow} \ \sum_{i \in F_j} y_i > \frac{1}{2}$$

3. $\boxed{\text{Clustering}}$

③ Clustering

- Go over clients in the above order.
- While considering j, if $\exists\, j' > j$ still "alive" st $F_{j'} \cap F_j \neq \emptyset$, kill j' & remove from order.
- Call j responsible for j'
- Let $C' \subseteq C$ be the alive clients
  - every client $j' \notin C'$ has a responsible $j \in C'$.
  - $\forall\, j_1, j_2 \in C'$, $F_{j_1} \cap F_{j_2} = \emptyset$

    by design.

④ Open Fac

For each $j \in C'$:

- Open cheapest facility $i \in F_j$.
- Assign j and all clients $j' \notin C'$ who j is responsible for to i

Analysis :-

• FACILITY OPENING COST

$$ALG_{fac} = \sum_{j \in C'} f_i = \sum_{j \in C'} \min_{i \in F_j} f_i$$

$$\leq 2 \sum_{j \in C'} \sum_{i \in F_j} f_i y_i$$

uses fact $\sum_{i \in F_j} y_i \geq \frac{1}{2}$ $\forall j$.

$$\leq 2 \sum_{i \in F} f_i y_i \underbrace{(\#\text{ of } j \in C' \text{ st } i \in F_j)}_{\leq 1}$$

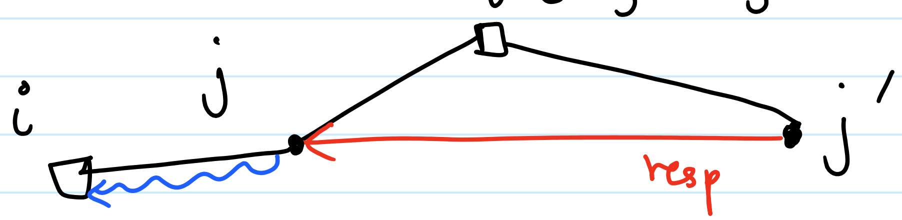
$$\leq 2F^*$$

• Connection Costs

– All $j \in C'$, go to a facility in F_j

$\therefore \quad d(i,j) \leq 2C_j$

– All $j' \notin C'$ $\quad \tilde{i} \in F_j \cap F_{j'}$



$$\Rightarrow \quad d(j,j') \leq d(\tilde{i},j) + d(\tilde{i},j')$$

$$\leq 2c_j + 2c_{j'}$$

$$\leq 4c_{j'} \quad \because \text{of order.}$$

$$d(i,j') \leq d(i,j) + d(j,j')$$

$$\leq 6c_{j'}$$

$\therefore$ Total conn cost $\leq 6 \sum_{j \in C} c_j = 6C^*$

$$\therefore \text{ALG} \leq 2F^* + 6C^*$$

Thm: Above Algorithm is a 6-appx.

Min-Cost Perfect Matchings in Bipartite Graphs

Input: - $G = (A \cup B, E)$

- $c : E \to \mathbb{R}_{\geq 0}$

Output: - A perfect matching M of G

Objective: - Minimize $\sum_{e \in M} c(e)$

LP:

$$\min \sum_{e=(u,v)} c_e x_{uv}$$

$$\forall u \in A: \quad \sum_{v \in B : (u,v) \in E} x_{uv} = 1$$

$$\forall v \in B: \quad \sum_{u \in A, (u,v) \in E} x_{uv} = 1$$

$$0 \leq x \leq 1$$

Thm:- There is an <u>integral</u> opt. soln to the above LP.

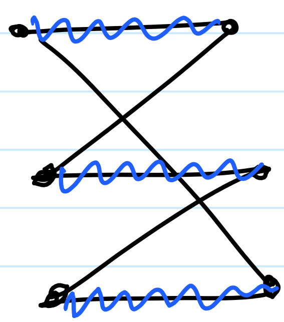
Proof:
- Let x be an optimum soln.
- Let $F \subseteq E$ be the set of edges with

$$0 < x_{uv} < 1$$

- If $F = \emptyset$, we are done, nothing to prove.
- $\deg_v(F) \neq 1$. .... Convince yourself.

$\Downarrow$

- Each connected component has a cycle. C



$C = M_1 \cup M_2$

Wlog, assume $c(M_1) \leq c(M_2)$

- Consider $$\tilde{x} = x + \delta \cdot \mathbb{1}_{M_1} - \delta \cdot \mathbb{1}_{M_2}$$

ie, $\forall\, e \in M_1$, $\tilde{x}_{uv} = x_{uv} + \delta$

$e \in M_2$, $\tilde{x}_{uv} = x_{uv} - \delta$

- $\tilde{x}$ is feasible
- $\text{cost}(\tilde{x}) \leq \text{cost}(x)$ ... since x is opt, we have $=$
- choose $$\delta = \min_{e \in C} \min(x_e, 1 - x_e)$$

$\therefore$ $\tilde{x}$ has <u>one</u> less edge in F

Keep repeating to get the <u>integral</u> optimum.

$\blacksquare$

---

③ <u>Unrelated Machine Scheduling</u> $(R \| C_{max})$

<u>Input</u>: - m - machines
n - jobs
p_{ij} : running time of job j on machine i

<u>Output</u>: Assignment / schedule of jobs to m/c's.

Objective:- $\min \max_{i=1}^{m} p_i(S_i) := \sum_{j \in S_i} p_{ij}$

jobs assigned to m/c i

LP-reln

$$\min T$$

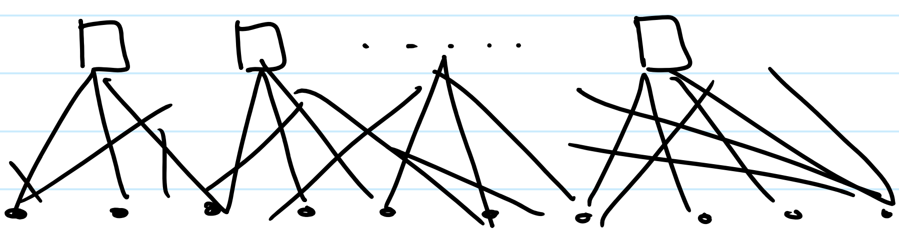
$$\forall i: \quad T \geq \sum_{j \in J} p_{ij} x_{ij}$$

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

$$\text{“} T \geq P_{max} \text{”}$$

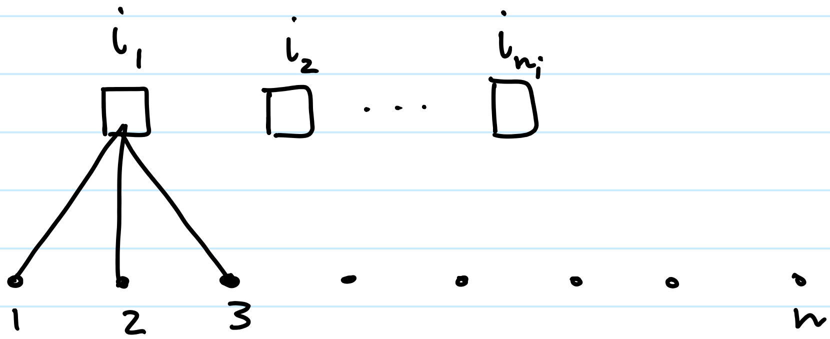
Algorithm:-

- Solve LP to get x



- For each m/c i, define $n_i := \left\lceil \sum_{j \in J} x_{ij} \right\rceil$
- let $N_i := \{i_1, i_2, \ldots, i_{n_i}\}$
- Construct a bipartite graph H with one side j, and the other $\bigcup_i N_i$

- Fix a machine i. We describe the edges from $N_i \to J$
- Order jobs s.t $P_{i1} \geq P_{i2} \geq \cdots \geq P_{in}$



- Find the smallest j_1 s.t. $\sum_{1 \leq j < j_1} x_{ij} < 1$ but

$\sum_{1 \leq j \leq j_1} x_{ij} > 1$. Connect $\{1, 2, \ldots, j_1\}$ to i_1

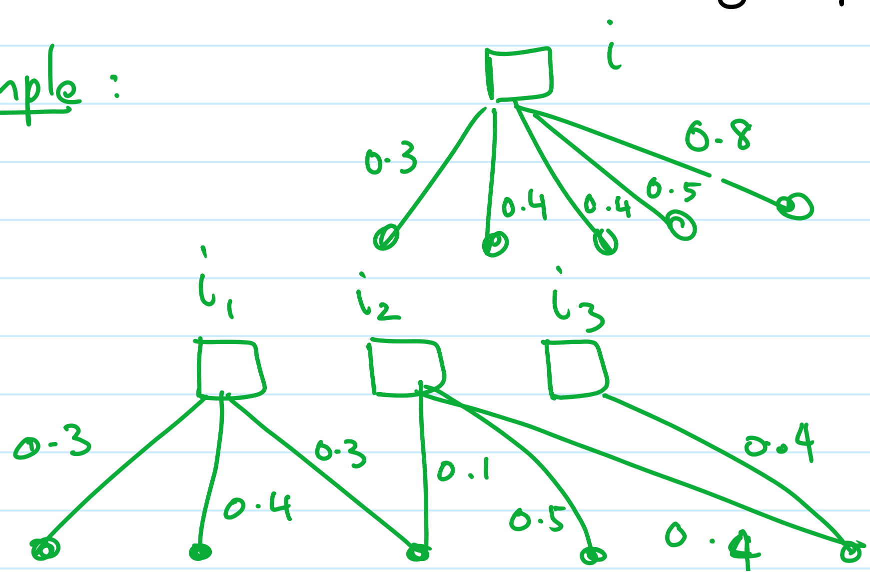
Also define $z_{i_1, j} = x_{ij} \quad \forall\ 1 \leq j < j_1$

& $z_{i_1, j_1} = 1 - \sum_{j < j_1} z_{i_1, j}$

Then do a similar thing for i_2 and so on.

Example:

$n_i = 3$



- After we do this, for all but $k = n_i$ we have

$$\sum_j z_{i_k, j} = 1$$

&

$$\sum_j z_{i_{n_i}, j} \leq 1$$

- Once we do this for all m/c's

$$\sum_{i \in \cup N_i} z_{i,j} = 1 \quad , \forall j$$

- That is, in the bipartite graph H, z is a fractional soln on its edges s.t

$$\forall j \in J : \sum_{i \in \cup N_i} z_{ij} = 1$$

$$\forall i \in \cup N_i : \sum_{j \in J} z_{ij} \leq 1$$

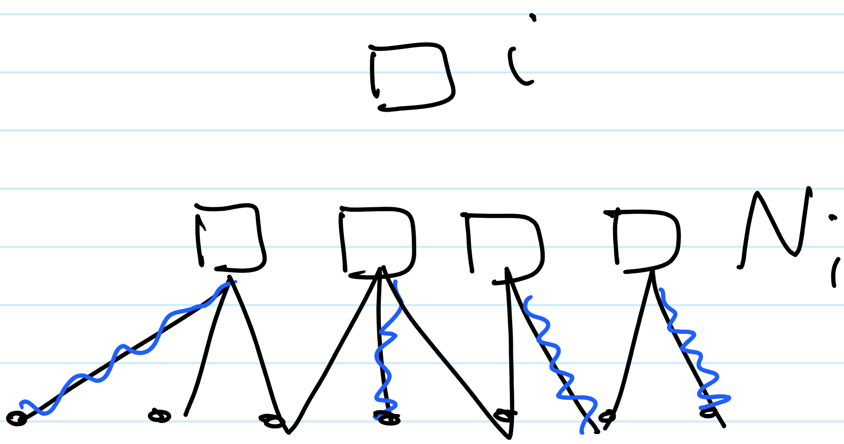
- This implies (like in the prev result) a matching M in H which matches all jobs in J.

–

– Again look @ machine i and assign all jobs that M assigns to vertices in N_i, to i. This completes the description of the algo.

Analysis

Fix i.



$$p_{i1} \geq p_{i2} \geq \cdots \cdots \geq p_{in}$$

Consider $k > 1$, and the copy i_k. Say it gets job j in M.

Note : $$p_{ij_k} = p_{ij_k} \cdot \sum_{j \in J} z_{i_{k-1}, j}$$
$$\leq \sum_{j \in J} p_{ij} \, z_{i_{k-1}, j}$$
$\because$ i_{k-1} gets $z_{i_{k-1}, j} > 0$ only for jobs with $p_{ij} \geq p_{ij_k}$

$\therefore$ Total load of jobs assigned on copies $k > 1$ is
$$\sum_{k>1} p_{ij_k} \leq \sum_{k>1} \sum_{j} p_{ij} \, z_{i_{k-1}, j}$$
$$= \sum_{j} p_{ij} \left(\sum_{k>1} z_{i_{k-1}, j} \right)$$
$$\leq \sum_{j} p_{ij} \, x_{ij} \leq T$$

$$\therefore C_{max} \leq T + p_{ij_1} \leq 2T$$

$\blacksquare$