16. 9. 22

PTAS-es

KNAPSACK

$$P = \sum_{i=1}^{n} p_i \leq p_{max} \cdot n$$

$$O(nP) \sim O(n^2 p_{max})$$

Scaled Instances :-

$$\hat{p} \in [0, n/\epsilon] \qquad K = \frac{p_{max} \cdot \epsilon}{n}$$

Algorithm

1. Choose $\epsilon > 0$
2. Scale $\hat{p_i} = \left\lfloor \frac{p_i}{K} \right\rfloor$ $\qquad \cancel{\frac{p_{max}\, n}{\epsilon}}$
3. Solve scaled instance : S

Runtime :- $O(n^2 \hat{p}_{max}) \sim O(n^3/\epsilon)$

Fully - PTAS.

Fact :-

p_1	p_2	p_3	...	p_l

opt

~~For each item, we are losing $\frac{p_{max} \cdot \epsilon}{n}$~~

~~Total loss $\leq \epsilon \cdot p_{max}$~~ [

$$\text{profit}(S) = K \cdot \hat{\text{profit}}(S)$$
$$\geq K \cdot \hat{\text{profit}}(\text{Opt})$$

$\Rightarrow$ profit (s) $\geq K \cdot \left[\sum_{i \in opt} p_i/_K - n \right]$

Floor.

$\left\lfloor \frac{p_i}{K} \right\rfloor \geq \frac{p_i}{K} - 1$

$= \sum_{i \in opt} p_i - K \cdot n$

$=$ profit (opt) $- p_{max} \cdot \epsilon$

Now, use profit (opt) $\geq p_{max}$ $\Rightarrow - p_{max} \geq -$ profit(opt)

profit(s) $\geq$ profit (opt) $(1-\epsilon)$

$\Rightarrow$ profit (s) $\geq (1-\epsilon)$ profit (opt)

[Scaling is the main thing in PTAS-es]

MAKESPAN :-

m identical machines

n jobs $= \{p_1, p_2, \cdots, p_n\}$ processing time.

Assume (for now) we know the value of Opt $= T$

[This does not make the problem easier, it becomes the decision problem.]

Idea!: Large jobs : $\{j : p_j \geq \epsilon \cdot T\}$

Small jobs : $\{j : p_j < \epsilon \cdot T\}$

Claim 1: Greedy on small jobs incurs additional $< \epsilon \cdot T$ on every machine.

: Same as previous.
: volume argument fails.

<u>Detour</u> :- [Configuration Makespan]

Suppose :-

1. There are only constantly many job types: l
2. In opt, any machine has $\leq k = O(1)$ jobs

<u>Configuration of a machine</u>

Type 1	Type 2	. . .	Type l

: m/c configuration

Each Type $\leq k$ jobs.

Total # of possible configs: k^l

Out of k^l config, I have to find m config s.t. every job $\in$ some config. (all jobs covered)

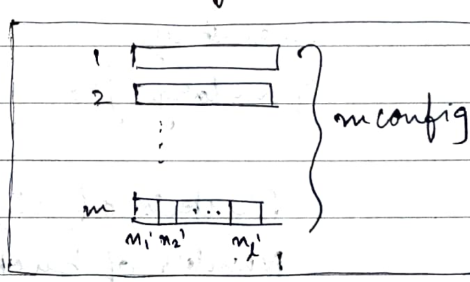
<u>Idea 2</u>: Use DP to solve Configuration Makespan

i/p: n_1 jobs of type 1

n_2 jobs of type 2

$\vdots$

n_l jobs of type l



Makespan $(\underset{n \text{ possible value}}{n_1}, n_2, n_3, \cdots, n_l)$ = Minimum no. of valid configurations that covers all the jobs.

$= \min_{\substack{\text{all valid configs} \\ (n_1', n_2', \ldots n_l')}}$ Makespan $(n_1 - n_1', n_2 - n_2', n_3 - n_3', \ldots, n_l - n_l') + 1$

Size of table : n^l

To fill each we need k^l

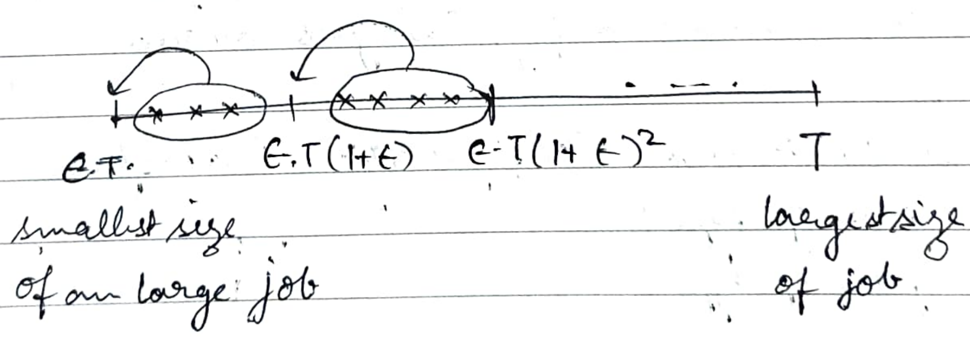
$\therefore (n^l \cdot k^l)$ running time.

Back to original problem.

Claim 2: On any machine, number of large jobs $\leq \frac{1}{\epsilon} = k$

$\downarrow$ otherwise OPT increases T

Claim 3:



Buckets of size $1+\epsilon$

Number of buckets $= l = \log_{1+\epsilon}\left(\frac{1}{\epsilon}\right)$

$$\left[\epsilon \cancel{T} (1+\epsilon)^l = \cancel{T} \quad \boxed{l = \log_{1+\epsilon} \frac{1}{\epsilon}} \right]$$

Rounding down to lower boundary.

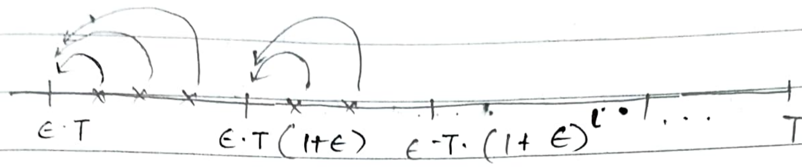
$$\text{Runtime} = n^l k^l = n^{\log_{(1+\epsilon)}\left(\frac{1}{\epsilon}\right)} \cdot \left(\frac{1}{\epsilon}\right)^{\log_{(1+\epsilon)}\left(\frac{1}{\epsilon}\right)}$$

$\boxed{\text{PTAS}}$

20.9.22

l : job types
k : max # of jobs on a m/c

$\Rightarrow$ DP : $n^l k^l$



$l = \log_{(1+\epsilon)}\left(\frac{1}{\epsilon}\right)$

$k = \frac{1}{\epsilon}$

$$DP(n_1', n_2', \dots, n_l') = \min_{(\dots)} (n_1' - s_1,\ n_2' - s_2, \dots)$$

$\downarrow$ can't be -ve, discard those config.

and Total can't be $> T$, discard those config.

answer to this DP $\leq m$.

Runtime: $n^{\log_{1+\epsilon}\left(\frac{1}{\epsilon}\right)} \cdot \left(\frac{1}{\epsilon}\right)^{\log_{1+\epsilon}\left(\frac{1}{\epsilon}\right)}$

$$= n^{\log_{1+\epsilon}\left(\frac{1}{\epsilon}\right)} \cdot 2^{\log_2\left(\frac{1}{\epsilon}\right)\log_{1+\epsilon}\left(\frac{1}{\epsilon}\right)}$$

↳ Not an efficient PTAS

O → constant factors hidden
$\tilde{O}$ → $\log_2$ factors hidden

We will improve to $\boxed{O(n) + 2^{\tilde{O}\left(\frac{1}{\epsilon}\right)}}$

ILP: (just a feasibility question)

Introduce variables $x_1, x_2, \cdots, x_\lambda$ for configurations $c_1, c_2, \cdots, c_\lambda$.

$\lambda = k^\ell$ Total possible config.

$$x_1 \cdot \begin{bmatrix} c_{11} \\ c_{12} \\ c_{13} \\ \vdots \\ c_{1\ell} \end{bmatrix} + x_2 \cdot \begin{bmatrix} c_{21} \\ c_{22} \\ c_{23} \\ \vdots \\ c_{2\ell} \end{bmatrix} + \cdots + x_\lambda \begin{bmatrix} c_{\lambda 1} \\ c_{\lambda 2} \\ c_{\lambda 3} \\ \vdots \\ c_{\lambda \ell} \end{bmatrix} = \begin{bmatrix} n_1 \\ n_2 \\ \vdots \\ n_\ell \end{bmatrix}$$

c_{11} ← # jobs of type 1

c_1 config, c_2 config, c_λ config

$$x_1 + x_2 + \cdots + x_\lambda = m$$

x_i ← non negative integer.

Algorithmic Lattice Theory.

Thm (R. Kannan): ILP can be solved in time

$$\lambda^{O(\lambda)} = 2^{O(\lambda \log \lambda)}$$

The running time does not depend on n, depends only on the dimension of lattice space (ℓ)

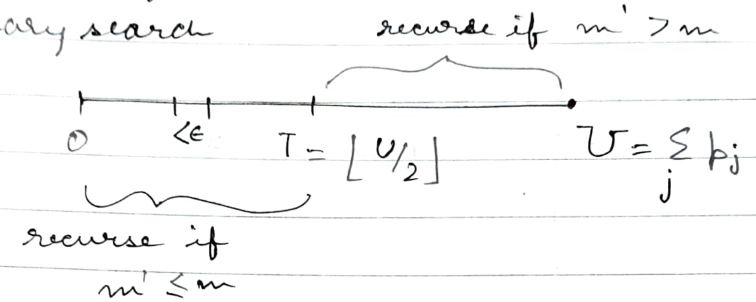
We can bring it down to $O(\lambda)$.. (For this case)

DRILL

Decide on some upper bound on T.
(maybe sum of all jobs)

$0 \le T \le$ large no.

Binary search



[m' = soln of PTAS]

when to stop $\Rightarrow$ range of searching $< \epsilon$

Figure out exact approximation.

not $\epsilon \cdot OPT$

p_j max or avg p_j whichever greater $\Leftarrow$ lower bound not 0.

[$1+\epsilon$ factor large jobs , ϵ factor small jobs , ϵ factor or 2ϵ factor Bin Search $= 1 + 4\epsilon$ around]

Later $\to$ non identical machines

LINEAR PROGRAM

Max : $x_1 + x_2$, $\{x_1, x_2\} \in \mathbb{R}$

Constraints : $2x_1 + x_2 \leq 1$ (i)

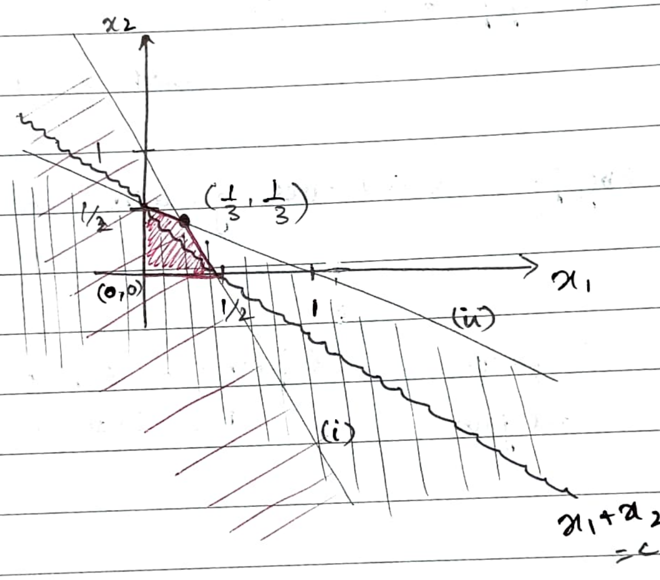
$x_1 + 2x_2 \leq 1$ (ii)

$x_1 \geq 0$

$x_2 \geq 0$

Simplex → practically efficient " " " "

Theoretical algo → efficient way to solve LP.



→ Half spaces

→ Level sets of equations

OPT = c

Keep increasing c as far as possible.

d dimensions → (d-1) dimension hyperplane

3 dimension → 2D plane

(a) Hyperplanes

(b) Half spaces

(c) Intersection of half spaces → Polygon in d dimensions → Polyhedron
(bounded / unbounded)

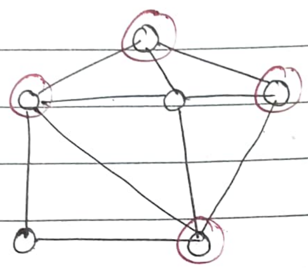
One of the vertices will always contain an optimal solution.

$\boxed{\text{VERTEX COVER}}$

art gallery problem

→ security gaurd to keep eye on all corridors

→ min no. of gaurds reqd.



Pick min no. of vertices to cover all edges.

Further → weights associated to vertices, minimize tot wt.

$x_v = \begin{cases} 1 \\ 0 \end{cases}$ indicator variable

$x_v = 1$ ← vertex taken

$x_v = 0$ ← vertex not taken.

min $\sum x_v$

$x_u + x_v \geq 1 \quad \forall (uv) \in E$

$\cancel{x_v \in \{0,1\}}$

$1 \geq x_v \geq 0$

$\xrightarrow{\text{integer}}$ NP Hard ⇒ Integer LP

$\xrightarrow{\text{fractional}}$ Purely Linear Program.

↳ we will get a better solution.

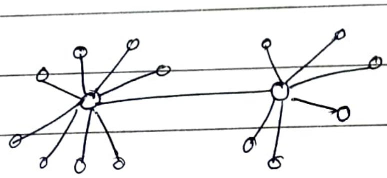
⇒ Relaxed LP

cost version:

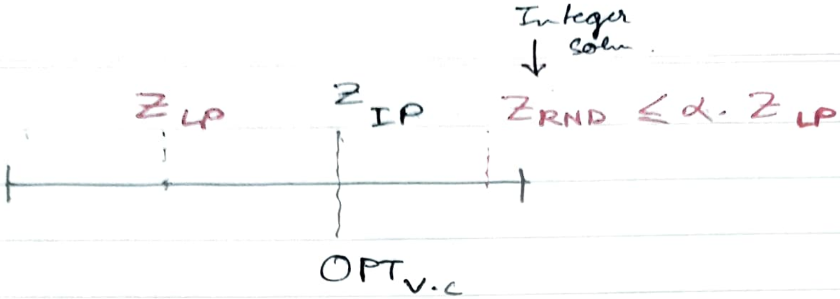
min $\sum x_v \cdot c_v$

$x_u + v_v \geq 1 \quad \forall (uv) \in E$

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



can't enforce equality.



Plan

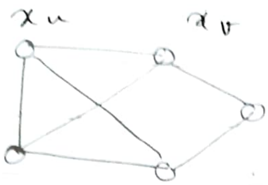
Deterministic Rounding :-

a) Solve LP : x^* is optimal

b) Round x_v^* up or down

23.9.22

LP Rounding : Vertex Cover



$\min \sum w_v \cdot x_v$ $\quad \forall\, uv \in E$

s.t. $x_u + x_v \geq 1$

$1 \cancel{\geq} x \geq 0$ $\quad [x \in \{0, 1\}]$

redundant

↓

if satisfied with 1, then satisfied with 1.5 also.

Integrality gap = ρ

Z_{LP} — $Z_{IP} = OPT(VC)$ — $Z_{RND} \leq \underbrace{\alpha}_{=2} \cdot Z_{LP}$

↑ Real $\qquad$ ↑ Integer ↑

if u and v taken then

$x_u + x_v \geq 1$

$\therefore x_u \geq \frac{1}{2}$

and $x_v \geq \frac{1}{2}$

$\therefore$ Round up if $\geq \frac{1}{2}$

Alg :-

a) Solve LP : x^*

b) $\tilde{C} = \{ \text{vertices } v : x_v^* \geq 1/2 \}$

c) Return $\tilde{C}$.

Thm : LP Rnd is a 2-approx

Proof : Cost of solution

$$= w(\tilde{C}) \leq 2 \sum_{v \in \tilde{C}} x_v^* w_v$$

$$\leq 2 Z_{LP}$$

LP gives lower bound (upper bound) on OPT.

The gap between Z_{LP} and Z_{IP} is of significance

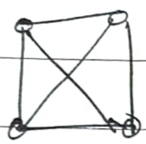
Integrality gap = ρ (ratio)

of optimal solution $\geq$ ρ times Z_{LP}.

why not exactly ρ times.

Z_{IP} will be equal to Z_{RND}.

Complete graph (K_n)



$Z_{IP} = n-1$

$Z_{LP} = n/2$ (half for every vertise)

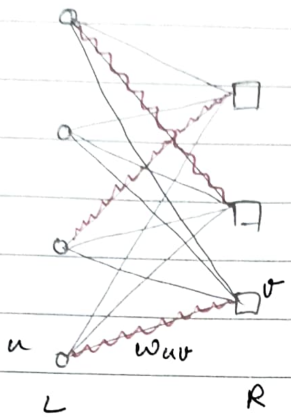
The difference is almost 2.

we cannot get a better approx ratio with this LP,
we cannot proof that the gap is better than 2, for this LP.

There could be multiple LP for one problem.

[LP can cheat.]

Maximum Bipartite Matching



matching - set of edges that do not share endpoints.

Pick matching, which maximizes tot. wt.

(a) Hungarian Algorithm.

Exact algo.
Maximum wt. matching.

(b) LP

variable for every edge

$$x_{uv} = \begin{cases} 1 \\ 0 \end{cases} \quad \forall (uv) \in E$$

$$\max \sum_{uv \in E} w_{uv} \cdot x_{uv}$$

s.t. $$\sum_{v \in R} x_{uv} \leq 1 \quad \forall u \in L$$

$$\sum_{u \in L} x_{uv} \leq 1 \quad \forall v \in R$$

$$1 \geq x \geq 0$$ $\longrightarrow$ because $x_{uv} \leq 1$ already ensured by last 2 equations

Rounding of Matching LP

0 edges → leave them
1 edges → take them
remaining edges → fractional.

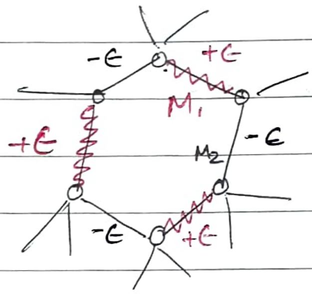
$\Downarrow$

$E_f : \{ e \in E : 0 < x_e < 1 \}$

Algo

$\boxed{a}$ Remove edges e with $x_e = 0$
Match (uv) if $X_{uv} = 1$

$\boxed{b}$ Start a BFS at a degree 1 vertex in E_f.
If there is none, start anywhere.

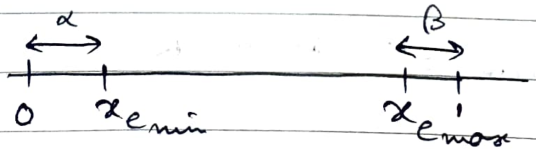


could be extra edges, we are focussing on the cycle.

The cycle is an even cycle (bipartite graph)
Can find 2 ~~perfect~~ matching in this.
Increase the red edges by ϵ.
Decrease the black edges by ϵ.

1. $$\tilde{x}_e = \begin{cases} x_e^* + \epsilon_1 & \text{for } e \in M_1 \\ x_e^* - \epsilon_1 & \text{for } e \in M_2 \end{cases}$$

↳ $\tilde{x}_e$ is feasible [ϵ_1 needs to be chosen correctly]



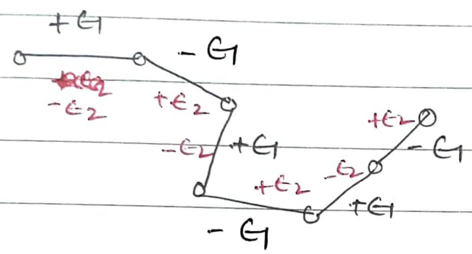
$\epsilon_1 = \min(\alpha, \beta)$

we need to consider the other matching also,

$$\tilde{x}_e^2 = \begin{cases} x_e^* - \epsilon_2 & \forall \; e \in M_1 \\ x_e^* + \epsilon_2 & \forall \; e \in M_2 \end{cases}$$

(ϵ_1 and ϵ_2 different)

if no cycle exists



end vertices
choose ϵ_1 and ϵ_2 such that within slack of end vertices.

Thm: There is one optimal solution to LP-Match where $x_{uv} = \{0, 1\}$ for all $uv \in E$.

Integrality gap $= \rho = 1$