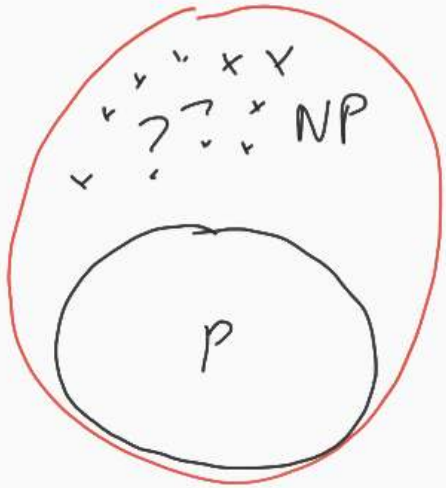
Lecture 21 : NP completeness and Approximations



① P - problems with efficient algos

② NP - problems with efficiently verifiable witnesses

Fact: $P \subseteq NP$ ✓

Question: is $P = NP$ or not ?

Assumption: $P \neq NP$

If $Q \in P$ then $NP = P$

↑ my problem

↑ something really wierd is happening.

Q' is NP-complete (as a hardest problem in NP)

① $Q' \in NP$

② if $Q' \in P$ then $NP = P$

Q ← your problem

Q ?? Q'

Q' is NP complete

① $Q' \in NP$

② ¬ Q is NP-complete

if $Q' \in P$

$\Rightarrow Q \in P$

$NP = P$

Fact: 4 COLORABLE is NP complete

3COLORABLE is NP complete ✓

if 4COLOR $\in P$

$\Rightarrow$ 3COLOR $\in P$.

$Q \Longrightarrow Q'$ ✓

3COLOR

4COLOR ↑

5 COLOR ↑

$x_1 x_2 \ldots x_n$ Boolean vars.

Cook Levin Theorem.

3SAT is NP-complete

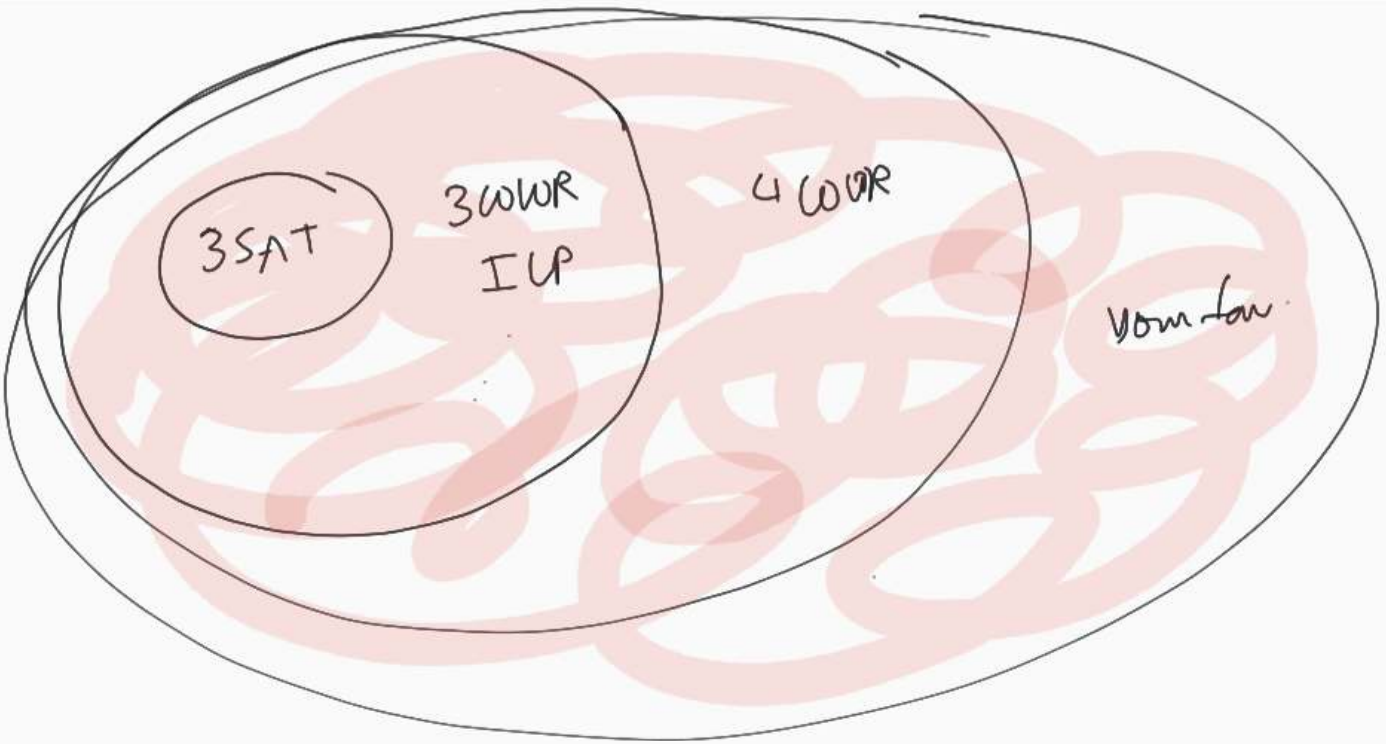
Qn: does ∃ setting of T/F for $x_1, x_2 \ldots x_n$ st φ true.

formula $\varphi = (x_1 \vee x_2 \vee \bar{x}_7) \wedge (x_3 \vee \bar{x}_1 \vee x_5)$

$\wedge$ ( )

$\wedge$

$\wedge$ $(\bar{x}_2 \vee x_9 \vee \bar{x}_{37})$



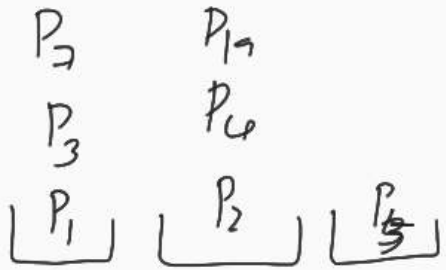
• Q

Makespan Minimization (MM)

n jobs $p_1, p_2, \ldots, p_n \in \mathbb{Z}_{\geq 0}$

m machines

Assign jobs to machines to minimize the makespan

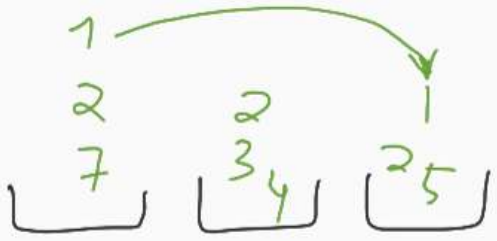


makespan = time @ which all jobs done.

7 3 4 2 5 2 1 2 1

27 ~~stau~~ units of work

$\Rightarrow \geq 9$ in some machine



makespan = ~~10~~ 9

Q: Is MM NP-complete?

NO. MM is not a decision problem.

Redefine MM

given $n, m, p_1 p_2 \dots p_n, \in \mathbb{Z}_{\geq 0}$

and a target M

does $\exists$ an assignment of jobs to machines s.t.

makespan (assignment) $\leq M$?

① $MM \in NP$

② $\exists$ problem Q which is NP-complete s.t. if $MM \in P$ $\Rightarrow Q \in P$

$\Rightarrow$ which is weird

Q1 again: $MM \in NP$ ✓

- witness = assignment
- verifier: checks easily in linear time.

Part 2: Find Q s.t. Q is NP-complete $\Rightarrow$ if $MM \in P \Rightarrow Q \in P$ (which is weird)

Thm: if $MM \in P \Rightarrow$ PARTITION $\in P$

Pf: Assume $\exists$ algo A to solve MM that runs in poly time

take any instance of Partition $I = (c_1, c_2 \ldots c_s)$ ↑ # of

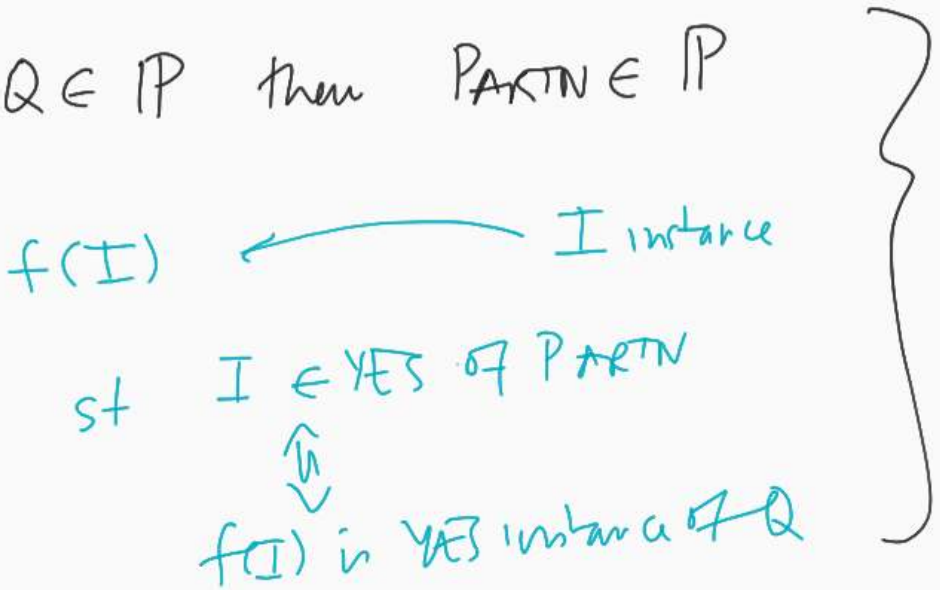
construct an instance of MM = $f(I)$

$$\begin{cases} n = s \\ m = 2 \\ P_i = c_i \\ M = \left(\sum_{i=1}^{s} c_i\right)/2 \end{cases}$$

I is a YES instance of Partition

$\Leftrightarrow$ $f(I)$ is a YES instance of MM

$Q \in \mathbb{P}$ then PARTN $\in \mathbb{P}$



Karp reduction
↑
encourage you to use

Q: Can you approximate the min makespan

Algorithms for MM

$P_1, P_2 \ldots P_n$

$P_{avg} = \frac{\sum_i P_i}{m} \leq OPT$

$P_{max} \leq OPT$

Greedy

$\forall i = 1 \ldots n$

put P_i on some machine with least load.

Thm: Makespan (Greedy) $\leq$ 2 · OPTimal makespan

$\forall$ instances

$\leq$

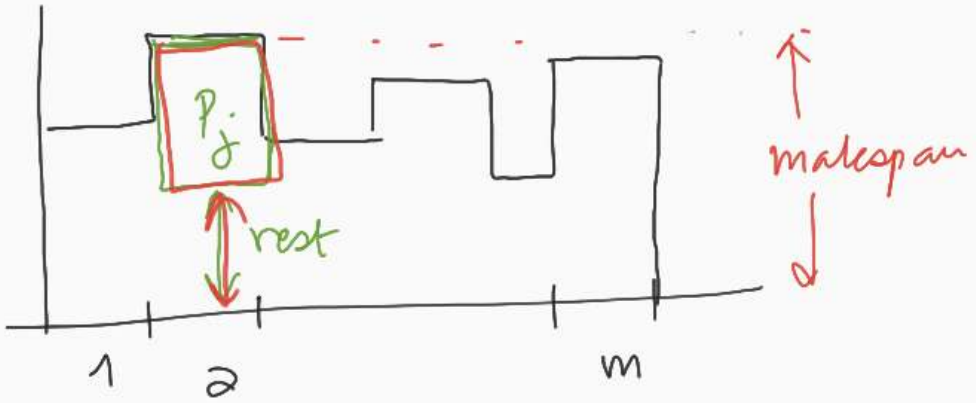
$LB_1 + LB_2$ difficult?

$\forall$ instances

$LB(I) \leq OPT(I)$

LB easier to argue about

Claim: $\text{Makespan}(\text{Greedy}(I)) \leq P_{max}(I) + P_{avg}(I)$



Better?

Makespan $= P_j +$ (rest) load of least loaded m/c @ when j beij added.

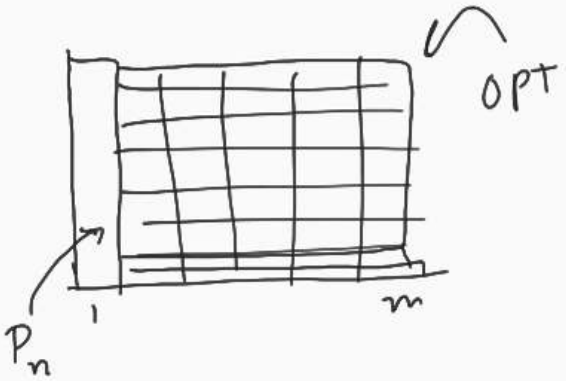
$\leq P_{max} + \left(\sum_{i=1}^{j-1} P_i\right)/m \leq P_{max} + P_{avg} \leq 2OPT.$

2-approximation. ☺

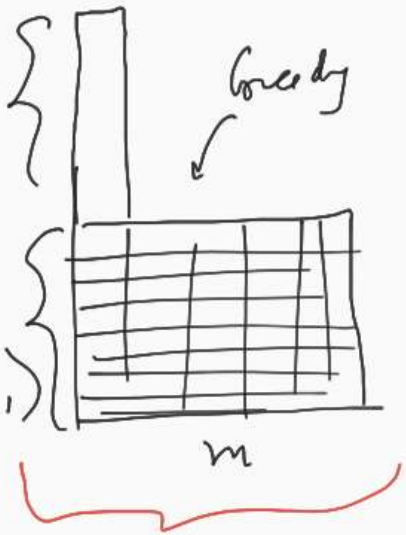
$\forall \varepsilon > 0$

Fact $\exists$ instances where Makespan (Greedy) $\geq (2-\varepsilon)$ OPT

Pf:



$p_1 p_2 \ldots p_{n-1} = 1$, $p_n = (n/m-1)$



OPT in case with sorted greedy.

Sorted Greedy

(1) Sort so that $p_1 \geq p_2 \geq \ldots \geq p_n$

(2) Run greedy.

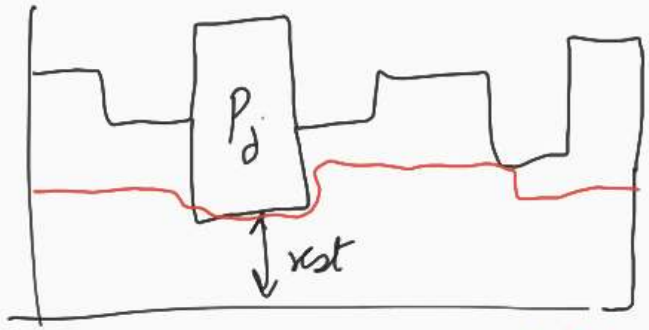
Thm: Makespan (Sorted Greedy) $\leq \underline{1.5}$ OPT.

Thm: Makespan (Sorted G) $\leq$ $\leq 1.5 OPT$

Pf: Makespan $\leq P_j + rest$

sps $rest = 0 \Rightarrow P_j + rest \leq P_{max} + 0$

$\leq OPT$



n # jobs
m # m/cs

else claim: $\exists \geq m+1$ jobs of size $\geq P_j \Rightarrow OPT \geq 2P_j$

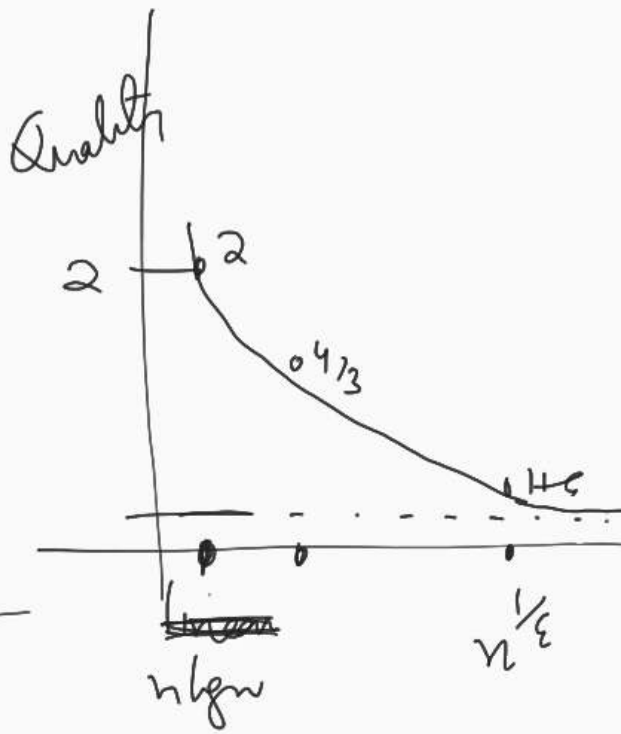
$$P_j + rest \leq \frac{OPT}{2} + P_{avg} \leq \frac{3}{2} OPT$$

Makespan:

2 Greedy

$\frac{3}{2} \rightarrow 4/3$ sorted greedy

$\forall \varepsilon > 0$ $(1+\varepsilon)$ apx algo, runtime $n^{O(1/\varepsilon)}$



Algo: 2-apx for Vertex Cover

< 1.36 apx $\Rightarrow$ P = NP (which is ~~weird~~ weird)