$P \subseteq ZPP \subseteq RP \subseteq BPP$

open problem $\quad P = BPP$?

Approximation alg   4/11/2022

Maximisation

α - approx $\quad (\alpha \le 1)$

Value (solution) $\ge \alpha \cdot$ Value (opt)

multiplicative approx.

Minimisation

β - approx $\quad (\beta \ge 1)$

Value (solution) $\le \beta \cdot$ Cost (opt)

aim $\rightarrow$ to reduce β as much as possible $(1+\varepsilon)$

Load Balancing Problem (NP Hard)

m Jobs — $t_1, t_2 \, \ldots \, t_m$

n processors $\rightarrow$ Identical [For now]

Jobs $\rightarrow$ indivisible

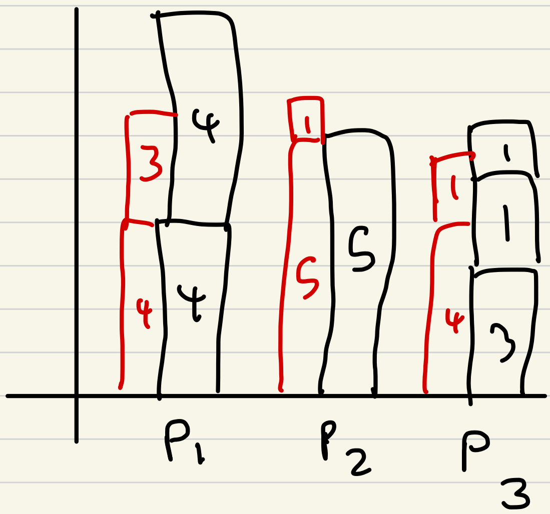
assign Jobs to processors evenly

makespan := max. load on any processor

Goal :- Minimise makespan

eg: $n = 3$ processors
6 jobs

$t_{i's}$ - 4, 4, 5, 1, 1, 3



makespan = 8

makespan = 7

7 is optimal
(Total = 18 , ideally (6, 6, 6) not possible here ⇒ (7, 6, 5))

NP - Hard (even for 2 processors)

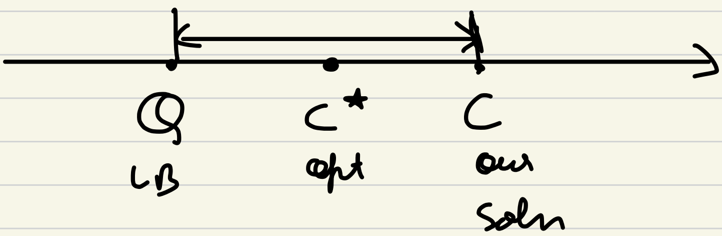
Q. succinct certificate?

H.W → Knapsack / subset sum reduce to min makespan (to conclude NP Hard)

For our approx. alg.

$$\text{makespan (solution)} \leq \beta \cdot \text{OPT makespan}$$

Finding some lower bound and bounding it rather than bounding c^* (optimal)



Suggestion 1

Round robin approach

Sort jobs

then do 1, 2,
2, 1

$1 \to P_1$
$2 \to P_2$

eg 32, 16, 8, 4, 2, 1

32	16
4	8
2	1
38	25

$\beta \approx \frac{3}{2}$

Suggestion 2

Greedy 1

Go over the jobs in the given order

At any time, allocate the processor with min. load.

eg

4, 4, 5, 1, 1, 3

can go anywhere

P_1	P_2	P_3
4	4	5
1	1	3

max span = 8

Greedy 1 gives a 2-approximate soln.

(For 2 processors, makespan = $\frac{1}{2}$ · ~~total~~ sum,
so all alg give 2approximations
Hence not very interesting)

Bounds on OPT

makespan will be atleast

① Avg load $\dfrac{\left(\sum_{i=1}^{m} t_i\right)}{n}$

② $\max\{t_1, t_2 \ldots t_n\}$

Attempt - 1

Greedy 1 $\leq$ 2 · (avg load)

Counter

3 processors

3 jobs 1000, 1, 1

makespan = 1000 (also optimal)

But avg = $\frac{1002}{3}$ = 334

and so our soln $\not\leq 2 \cdot (\text{avg})$

eventhough its optimal

So not a great L.B

Attempt 2

Greedy 1 $\leq 2 \cdot \{\text{max load of a job}\}$

3 processors

1, 1, 1 . . . . . 1 (1000 jobs)

Greedy 1 makespan $= \left\lceil \frac{1000}{3} \right\rceil = 334$

(also optimal)

max load $= 1$

$\therefore$ Greedy 1 $\not\leq 2 \cdot$ LB

$\therefore$ Bad LB

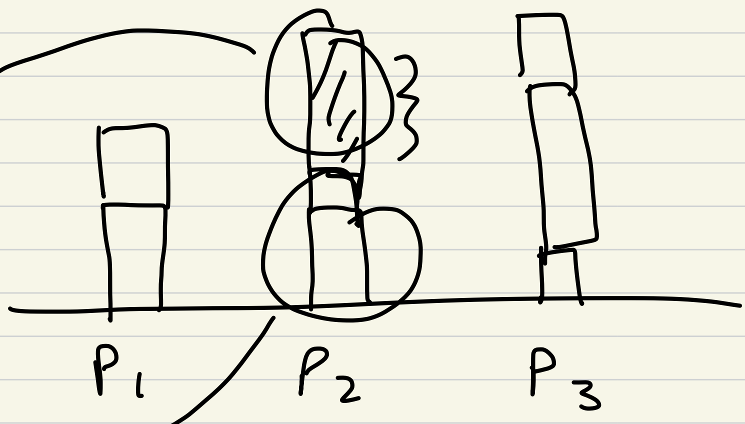
($\because$ eventhough our alg gives optimal, doesn't do well with LB)

Claim Greedy 1 $\leq$ avg load + max. load of any job

$\leq 2 \cdot \max(\text{avg}, \text{max})$

$\Rightarrow$ Greedy 1 $\leq 2 \cdot$ optimal

<u>HW</u> do the same proof for Round Robin



$\leq \frac{\sum_{i=1}^{n} t_i}{n}$ (avg)

when the last node was assigned, load on P_2 was minimum among all processors

$\leq max_L$

$\therefore$ load on $P_2 \leq \frac{\sum t_i}{n} + max_L$ (This concludes the proof)

average is over processors (not jobs)

for a $\frac{3}{2}$ - approx

Greedy 2

$\rightarrow$ Sort the jobs + Greedy 1
in decreasing order

Claim :

Greedy 2 gives a $\frac{3}{2}$ approximation eg where Greedy 1 does badly

5, 5, 5, 5, 1, 1, 1, 1, 1, 5

P_1	P_2	P_3	P_4	P_5
5	5	5	5	1
				1
				1
				1
				1
5				

max = 10 opt = 6

(max = $2n$
opt = $n+1$
ratio = $\frac{2n}{n+1}$)

proof of claim

$\boxed{k}$ } $\le 1/2$ (opt)

} $\le$ opt

total load before $k \le$ avg $\le$ opt

P_1 P_2 . . . P_n

Case 1 $k > n$ (no of processors)

By PHP (Pigeon hole)
Some processor gets 2 jobs among first k.
($\because$ array sorted, 2 jobs $\ge k$
($\because$ sort)

$\therefore$ optimal $\ge 2\,t_k$

$$\therefore \quad t_h \le \frac{opt}{2}$$

<u>Case 2</u> $\quad k \le n$

Then this processor has only 1 job

$$k \le opt$$