COL 758: Advanced Algorithms

Lectures 34, 35

Load Balancing

Given:

- m machines, n jobs
- Parameters p_{ij} = Time taken by job j on machine i (Here $j \leq n$, $i \leq m$)

Question: Find an assignment of jobs to machines that minimizes maximum load.

Constraint:

- Each job must run contiguously on one machine.
- A machine can process at most one job at a time.

Load of a machine:

If a machine i is assigned jobs $j_1, \ldots, j_k$, then the $\text{Load}(i) = p_{ij_1} + \ldots + p_{ij_k}$

Load Balancing

Given:

- m machines, n jobs
- Parameters p_{ij} = Time taken by job j on machine i (Here $j \leq n$, $i \leq m$)

Question: Find an assignment of jobs to machines that minimizes maximum load.

Example:

Jobs

	1	2	3	4	
M: 1	99	10	20	99	20
M: 2	5	15	99	99	20
M: 3	99	99	25	20	20

max load = 20

"Online" Load Balancing

$$(j) \quad \begin{bmatrix} p_{1j} \\ p_{2j} \\ \vdots \\ p_{mj} \end{bmatrix} \begin{array}{c} 1 \\ 2 \\ \vdots \\ m \end{array}$$

Given:

- m machines, n jobs (jobs arriving in online manner)
- Parameters p_{ij} = Time taken by job j on machine i (Here $j \leq n$, $i \leq m$)

Question: Find an assignment of jobs to machines (in an online manner) with $O(\log m)$ competitive ratio!

Example:

Greedy

Jobs

	1	2	3	4	
M: 1	99	10	20	99	10
M: 2	5	15	99	99	5
M: 3	99	99	25	20	45

Simpler Problem:

Threshold Load Balancing

Threshold Load Balancing

Given:

- m machines, n jobs, threshold α
- Parameters p_{ij} = Time taken by job j on machine i (Here $j \leq n$, $i \leq m$)

Question: ~~Can we~~ find an assignment of jobs to machines so that load on each machine is $\leq \alpha$?

Example:

Jobs

	1	2	3	4
M: 1	99	10	20	99
M: 2	5	15	99	99
M: 3	99	99	25	20

$\alpha = 15$ Ans = No

$\alpha \geq 20$ Ans = Valid assignment exists!

“Online” Threshold Load Balancing

Given:

- m machines, n jobs (jobs arriving in online manner), threshold = α
- Parameters p_{ij} = Time taken by job j on machine i (Here $j \leq n$, $i \leq m$)

Algorithm:

jobs 1, 2, 3, . . . → “Online” Threshold Load Balancing ← threshold α

→ Claim that NO (offline) scheduling is possible with maximum load at most α

OR

→ Find an Online scheduling of jobs with maximum load $O(\alpha \log m)$.

Natural Question:

Why is ‘Threshold’ Load Balancing important?

"Online" Threshold Load Balancing → "Online" Load Balancing

Theorem : If there exists an algorithm for "Online Threshold Load Balancing" with competitive ratio $c \cdot \log n$, then there exists an algorithm for "Online Load Balancing" with competitive ratio $4c \cdot \log n$

"Online" Threshold Load Balancing ➡ "Online" Load Balancing

Algorithm

Arrival of Job 1: Set $\alpha = \min_i(p_{i1})$, and schedule job 1 on machine $\arg\min_i(p_{i1})$

Arrival of Job j: Run the black-box algorithm of "Online Threshold Load Balancing" with parameter α.

Two possibilities are

1. Assign job j to some machine so that maximum load remains bounded by '$c \cdot \alpha \log n$'
   (Continue to next job).

2. Claim that no assignment is possible with maximum load $\leqslant \alpha$
   (Set $\alpha_{new} = 2\alpha$. Forget about job assignment till now, and continue with job $j+1$ as first job.)

"Online" Threshold Load Balancing → "Online" Load Balancing

Round 1

1, 2, 3 → "Online" Threshold Load Balancing ↑ threshold α

starting point

4, 5, 6, 7, 8 → "Online" Threshold Load Balancing ↑ threshold 2α

9, 10, 11, 12 → "Online" Threshold Load Balancing ↑ threshold 4α

Round r

$\ldots, n$ → "Online" Threshold Load Balancing ↑ threshold $2^r\alpha$

$$\left(c\alpha \log(m) + c(2\alpha)\log m + c(4\alpha)\log m + c(2^r a \log m)\right)$$

$$\binom{\text{Max}}{\text{Load}} \leq c \cdot 2^{r+1} \alpha \log m$$

$$\boxed{\text{"}c.r\text{"} = 4c \cdot \log m}$$

Initialization:

Job 1

$$\alpha = \min_{i=1}^{m} (p_{i1})$$

Claim:
No scheduling possible with maximum load $\leqslant 2^{r-1}\alpha$

Next Question:

Algorithm for Online ‘Threshold’ Load Balancing?

"Online" Threshold Load Balancing

$(\alpha = 1)$

Why we can assume $\alpha = 1$?

Given:

- m machines, n jobs (jobs arriving in online manner), threshold $= \alpha$ $\qquad \alpha^{new} = \frac{\alpha}{\alpha} = 1$
- Parameters p_{ij} = Time taken by job j on machine i (Here $j \leq n$, $i \leq m$)

$$p_{ij}^{new} = \frac{p_{ij}}{\alpha}$$

Algorithm:

jobs 1, 2, 3, . . → "Online" Threshold Load Balancing ← threshold α

→ Claim that NO (offline) scheduling is possible with maximum load at most α

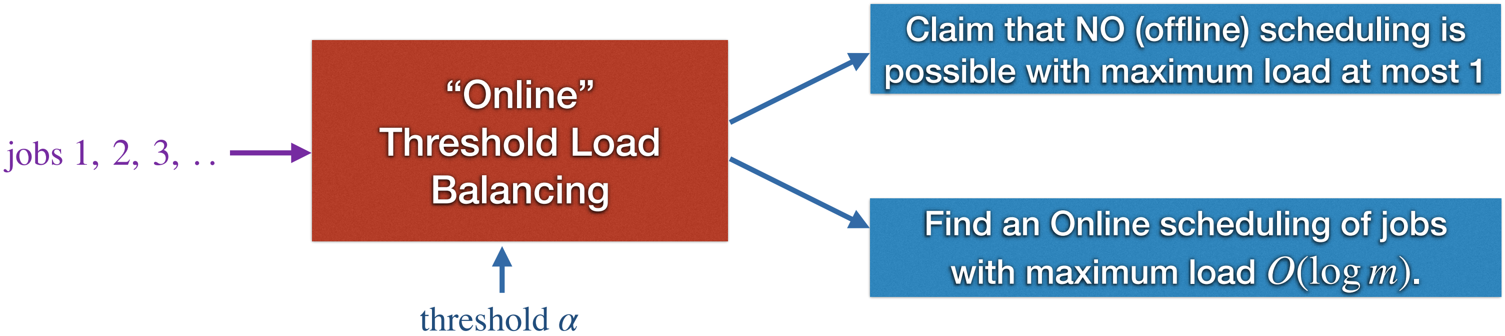
→ Find an Online scheduling of jobs with maximum load $O(\alpha \log m)$.

"Online" Threshold Load Balancing — Normalized version

Given:

- m machines, n jobs (jobs arriving in online manner), threshold = 1
- Parameters p_{ij} = Time taken by job j on machine i (Here $j \le n$, $i \le m$)

Algorithm:



Modeling Offline problem

$\min\ c^T x$ s.t. $Ax \geq b$, $x \geq 0$ $\longleftrightarrow$ $\max\ b^T y$, $A^T y \geq c$, $y \geq 0$

Integer Program

Optimize $\max \left(\sum_{(i,j) \in Q} y_{ij} \right)$

Subject to: $\sum_{i \mid (i,j) \in Q} y_{ij} \leqslant 1$, for each job $j \in [1,n]$ — z_j

$\sum_{j \mid (i,j) \in Q} p_{ij}\, y_{ij} \leqslant 1$, for each m/c $i \in [1,m]$ — x_i

$y_{ij} \in \{0,1\}$ $\quad y_{ij} \geq 0$

p_{ij} = Processing time of job j on machine i

$Q = \{(i,j) \mid p_{ij} \leq 1\}$

$$y_{ij} = \begin{cases} 1 & \text{Job } j \text{ is assigned to machine } i \\ 0 & \text{otherwise} \end{cases}$$

No. of variables $= |Q|$

No of constraints $= m + n$

$$\min \left(\sum_{i=1}^{m} x_i + \sum_{j=1}^{n} z_j \right)$$

$\forall\, i,j \in Q$

$$x_i\, p_{ij} + z_j \geq 1$$

$x_i \geq 0 \qquad y_j \geq 0$

y_{ij}

$z_j \longrightarrow 1$

$x_i \longrightarrow p_{ij}$

Primal and Dual

Primal Program

Optimize $\max \left(\sum_{(i,j)\in Q} y_{ij} \right)$

Subject to: $\sum_{i \mid (i,j)\in Q} y_{ij} \leqslant 1$, for each job $j \in [1,n]$

$\sum_{i \mid (i,j)\in Q} p_{ij}\, y_{ij} \leqslant 1$, for each m/c $i \in [1,m]$

$y_{ij} \geqslant 0$

Dual Program

Optimize $\min \left(\sum_{i=1}^{m} x_i + \sum_{j=1}^{n} z_j \right)$

Subject to: $p_{ij}x_i + z_j \geqslant 1$, for $(i,j) \in Q$

$x_i \geqslant 0$, for $i \in [1,m]$

$z_j \geqslant 0$, for $j \in [1,n]$

$$z_j \geq 1 - p_{ij}\, x_i$$

$$z_j \geq \max_{\substack{i \\ (i,j)\in Q}} \left(1 - p_{ij}\, x_i \right)$$

Primal and Dual

M/c dual variables

Job dual variables.

Primal Program

Optimize $\max \left(\sum_{(i,j) \in Q} y_{ij} \right)$

Subject to: $\sum_{i \mid (i,j) \in Q} y_{ij} \leqslant 1$, for each job $j \in [1,n]$

$\sum_{i \mid (i,j) \in Q} p_{ij}\, y_{ij} \leqslant 1$, for each m/c $i \in [1,m]$

$y_{ij} \geqslant 0$

Dual Program

Optimize $\min \left(\sum_{i=1}^{m} x_i + \sum_{j=1}^{n} z_j \right)$

Subject to: $z_j \geqslant \max_{i \mid (i,j) \in Q} (1 - p_{ij} x_i)$, for $j \in [1,m]$

$x_i \geqslant 0$, for $i \in [1,m]$

$z_j \geqslant 0$, for $j \in [1,n]$

Added when job j arrives

Maintain: A dual feasible D, and choice of integral y_{ij}'s that satisfy first primal constraint.

Maintaining Dual Feasible and Assignment of Jobs

Algorithm

Before arrival of Jobs: Initialise $x_1, \ldots, x_m = \frac{1}{m}$

$D = \sum x_i = \frac{m}{m} = 1$

Job j arrives:

If $p_{1,j}, \ldots, p_{m,j} > 1$, or ______________________, then NO scheduling with load at most 1 possible.

Else:

1. Let i_0 be the index maximizing $(1 - p_{ij}x_i)$. Here (i_0, j) lies in Q.
2. Assign job j to machine i_0 , and set $y_{i_0 j} = 1$.
3. Set $z_j = 1 - (p_{i_0 j})(x_{i_0})$.
4. Update x_{i_0} to $x_{i_0} + \Delta x_{i_0}$.

$D_{obj} = \sum_{i=1}^{m} x_i + \sum_{j=1}^{n} z_j$

Change in Dual is $\Delta D =$ "$\underbrace{1 - (p_{i_0 j})(x_{i_0})}_{\Delta z_j}$" + "$\Delta x_{i_0}$."

A favourable requirement is: $\Delta D + \Delta x_{i_0} = 1$.

$\Delta D = 1 - \Delta x_{i_0}$ $\Big\}$ $\Delta x_{i_0} = \frac{(p_{i_0 j})(x_{i_0})}{2}$

After arrival of n jobs:

(Sum total change in D) + (Sum total change in $\sum_{i \leqslant m} x_i$) $= n$

Solving this gives,

$$(D - 1) + (\sum_{i \leqslant m} x_i - 1) = n$$

$$D \;=\; n + 2 - \sum_{i \leqslant m} x_i \quad \longleftarrow \text{If } x_i \text{ becomes larger than 2, then } D_{offline-opt} < n.$$

Bcoz we find some dual feasible whose obj value $< n$

What does this imply?

If any x_i becomes larger than 2, then $P_{offline-opt} < n$.

This in turn imply we can't schedule all n jobs under the guarantee that max load is at most 1.

Maintaining Dual Feasible and Assignment of Jobs

Algorithm

Before arrival of Jobs: Initialise $x_1, \ldots, x_m = \dfrac{1}{m}$

Job j arrives:

If $p_{1,j}, \ldots, p_{m,j} > 1$, or any x_i is larger than 2, then NO scheduling with load at most 1 possible.

Else:

1. Let i_0 be the index maximizing $(1 - p_{ij}x_i)$. Here (i_0, j) lies in Q.
2. Assign job j to machine i_0 , and set $y_{i_0 j} = 1$.
3. Set $z_j = 1 - (p_{i_0 j})(x_{i_0})$.
4. Update $x_{i_0} = x_{i_0} + (p_{i_0 j})(x_{i_0})/2$.

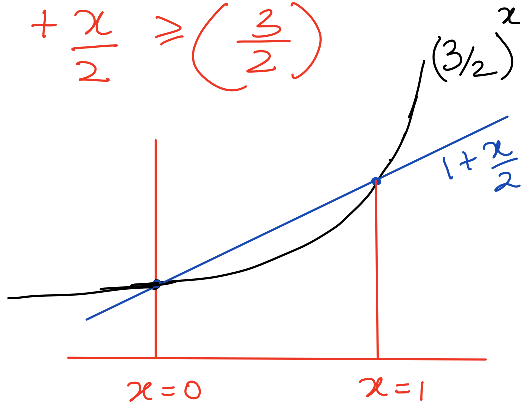
$x_{i_0}^{new} = x_{i_0}\left(1 + \dfrac{p_{i_0 j}}{2}\right)$

Change in Dual is $\Delta D = \; 1 - (p_{i_0 j})(x_{i_0}) + \Delta x_{i_0}$.

A favourable requirement is: $\Delta D + \Delta x_{i_0} = 1$.

Bound on load on any machine i:

If $x \in [0,1]$, then $1 + \frac{x}{2} \geq \left(\frac{3}{2}\right)^x$



Suppose jobs $j_1, \ldots, j_k$ are assigned to machine i.

We know at any stage $x_i \leqslant 3$.

$$3 \geq x_i \geq \frac{1}{m}\left(1 + \frac{p_{ij_1}}{2}\right)\left(1 + \frac{p_{ij_2}}{2}\right)\ldots\left(1 + \frac{p_{ij_k}}{2}\right)$$

$$3 \geq x_i \geq \frac{1}{m}\left(\frac{3}{2}\right)^{(p_{ij_1} + \cdots + p_{ij_k})}$$

$(p_{ij_1} + \cdots + p_{ij_k}) = \mathrm{LOAD}(i)$

So we get..

$$(p_{ij_1} + p_{ij_2} + \ldots + p_{ij_k}) \leqslant \log_{\frac{3}{2}}(3m) = O(\log m)$$

Online Load Balancing

Theorem: There exists an algorithm for "Online Threshold Load Balancing" with competitive ratio $c \cdot \log m$ for some constant c.

log(m)