Lecture 8.1

Selfish Load Balancing Game

n jobs to be scheduled on m machines.

- Players : $[n]$ the set of jobs.
- Strategy set of each player : $[m]$ the set of machines.
- Strategy profile : an assignment

$$A : [n] \longrightarrow [m]$$

- Cost function : suppose i-th job is assigned to

j-th machine. Each job $i \in [n]$ has a length w_i and each machine j has a speed of s_j. Time taken to run job i in machine j is $\frac{w_i}{s_j}$. The load l_j of machine j in an assignment $A : [n] \to [m]$ is $\sum_{i \in [n] : A(i) = j} \frac{w_i}{s_j}$. The cost of job i in a strategy profile $A : [n] \to [m]$ is l_j where $A(i) = j$.

Theorem: Every selfish load balancing game has a PSNE.

Proof: Associate every assignment $A:[n]\to[m]$ with its sorted "load vector" $(\lambda_1,\ldots,\lambda_m)$ $\lambda_1\geq\lambda_2\geq\cdots\geq\lambda_m$.

We say $(\lambda_1,\ldots,\lambda_m) \underset{\to}{>} (\lambda_1',\ldots,\lambda_m')$ if there exists $j\in[m]$ such that $\lambda_k=\lambda_k'$ $\forall\ k\in[j-1]$,

$$\lambda_j > \lambda_j'.$$

Let $(\lambda_1, \ldots, \lambda_m)$ be the smallest sorted load vector, let the assignment A corresponds to $(\lambda_1, \ldots, \lambda_m)$. We claim that A is a PSNE. Suppose <u>not</u>. Then, there exists a job $i \in [n]$ which benefits by moving from machine s to machine t.

$$\left(\lambda_1, \ldots, \underset{s}{\lambda_a}, \ldots, \underset{t}{\lambda_b}, \ldots, \lambda_m\right)$$

$$> \left(\lambda_1, \ldots \lambda_{a-1}, \ldots \quad \ldots\right) \quad \text{new sorted load vector}$$

This contradicts our assumption that $(\lambda_1, \ldots, \lambda_m)$ is the smallest sorted load vector. $\blacksquare$

Theorem: The PoA of any selfish load balancing game with n jobs and m identical (same speed) machines is at most $2 - \frac{2}{m+1}$.

Proof: Since the machines are identical, wma wlog that $s_j = 1 \; \forall j \in [m]$, Let A be a PSNE assignment. Social welfare function is the makespan i.e. $\max_{j \in [m]} l_j$. Let OPT be the minimum makespan possible.

Let $c(A) = l_{j^*}$

There must be at least 2 jobs assigned to j^*.

Otherwise, $c(A) = l_{j^*} = OPT$ since $OPT \geq \max_{i \in [n]} w_i$

and $PoA = 1$

Let $i^* \in [n]$ be a job assigned to j^* in A.

wma wlog $\quad w_{i^*} \leq \frac{1}{2} c(A)$

$j \in [m] \setminus \{j^*\}$, $\quad l_j + w_{i^*} \geq l_{j^*}$

$$\Rightarrow l_j \geq c(A) - w_{i^*}$$

$$\geq c(A) - \frac{1}{2} c(A)$$

$$= \frac{1}{2} c(A)$$

$$
\begin{aligned}
OPT &\geq \frac{\sum_{i=1}^{n} w_i}{m} \\
&= \frac{\sum_{j=1}^{m} l_j}{m} \\
&= \frac{l_{j^*} + \sum_{\substack{j=1 \\ j \neq j^*}}^{m} l_j}{m} \\
&\geq \frac{c(A) + \frac{1}{2}(m-1)\, c(A)}{m} \\
&= \frac{m+1}{2m}\, c(A)
\end{aligned}
$$

$$PoA = \frac{c(A)}{OPT} \leq \frac{2m}{m+1} = 2 - \frac{2}{m+1}$$ $\blacksquare$