Computational Learning Theory	**Spring Semester, 2005/6**

Lecture 2: March 21 2006

Lecturer: Yishay Mansour *Scribe: Adi Adiv, Michal Rosen and Ricky Rosen*[1]

2.1 Price of Anarchy

In this lecture we examine "players - machine" games, where each player chooses a machine to place her job at. We look at a global *optimum* function. We basically deal with two models: *pure*, where the players choose a deterministic strategy, and *mixed*, where the players choose a stochastic strategy (i.e., each chooses a probability distribution over machines). We farther examine different types of games such as identical machines (equal speed to all the machines), non-identical machine (different speed). We introduce the measure *price of anarchy* to capture the "quality" of equilibria and specify the price of anarchy in each model.

The main goal is to compare the "quality" of *Nash equilibrium* (*NE*) to the "quality" of a global optimum (OPT). The following examples will help us understand the notion of the *Price of Anarchy*:

2.1.1 Routing on parallel lines

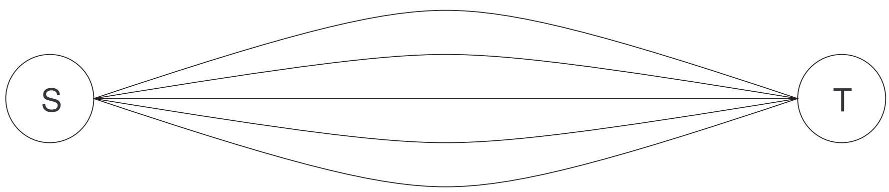

Figure 2.1.1: Routing on parallel lines

- Assume there is a network of parallel lines from an origin S to a destination T as shown in figure 2.1.1. Several agents want to send a particular amount of traffic along a path from the source S to the destination T. The more traffic on a specific line, the longer the traffic delay.
- Allocation jobs to machines as shown in figure 2.1. Each player (job) has to choose a resource (machine). The machines may be of a different speed. The performance of

[1]This scribe is based in port on the scribe notes of Noa Bar-Yosef and Eitan Yaffe 2004

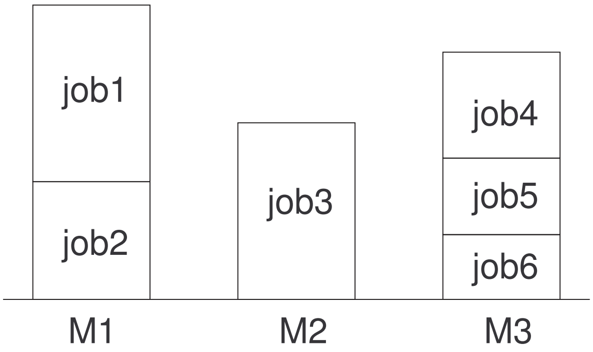

Figure 2.1: Scheduling jobs on machines

each machine reduces as more jobs are allocated to it. The players aim is to run on the least loaded machine. We are interested in specifying the Nash equilibria in such games. This is a situation where no player can gain by switching to different machine. The *global optimum* function, in this case, is either the minimum over the load on the most loaded machine or the sum of costs over all machines.

In these scribe we will use only the terminology of the job scheduling problem.

2.2 The Model

- A set of n users (or players), denoted $N = \{1, 2, ..., n\}$
- A set of m machines: $M_1, M_2, ..., M_m$
- A vector of $\vec{s}$ speeds: $s_1, s_2, ..., s_m$ where s_i is the speed of machine M_i
- A vector of weights, $w_1, ..., w_n$, where each user i has a weight: $w_i > 0$
- Let A_i be the actions of player i, i.e, $A_i =$ M. Mapping of user i to machines. Let $A = \times_{i=1}^{n} A_i$ be the joint action. By $a_i = M_i$ we mean that the i^{th} player runs on machine M_j.
- The *cost* for a given joint action $a \in A$ for a given player i will be defined as follows: $C_i(a) = \frac{\sum_{j:a_i=a_j} w_j}{s_{a_i}}$

- The load on machine M_i in joint action $a \in A$ is,

$$L_i(a) = \frac{\sum_{j:A(j)=i} w_j}{s_i}$$

  It is easy to verify that $C_i(a) = L_{a_i}(a)$.

We can define different measures for the global optimum function:

1. MakeSpan $MS(a) = \max_i L_i(a)$ (this is the $\L_\infty$ norm of a).
2. $SC(a) = \sum_j C_j(a)$ (this is the $\L_1$ norm of a, since $C_j(a)$'s are non-negative).

Note that $SC(a) = \sum_j C_j(a) = \sum_i (L_i(a))^2$. for the social optimum we have,

$$OPTSC = \min_{a \in A} SC(a)$$

,

$$OPT_{MS} = \min_{a \in A} MS(a)$$

2.3 Points of equilibria

Defenitions :

$\Delta(A_i)$ - is a collection of random variables above A_i.
$\Delta(A) = \times\Delta(A_i)$.
The expected cost is:
$E_{a\sim p}[c_j(a)] = E_{a\sim p}[L_k(a)|a_j = k] = E_{a\sim p,k\sim p_j}[\frac{w_j}{S_k} + L_k(a^{-j})]$.
$P = \{p_1, ...p_n\} \in \Delta(A)$.

In our discussion we will consider two types of equilibria, in both the interpretation is that no player can choose another machine and decrease her cost:

- Pure Nash equilibria: $a \in A$ is a pure Nash equilibria if for every player $j \in N$ and for every machine $M_i \in M : c_j(a_{-j}, a_j) \leq c_j(a_{-j}, a_j = M_i)$
- Mixed Nash equilibria : for every $j \in N$ and for every $M_i \in M$, $E[c_j(a)] \leq E[c_j(a^{-j}, a_j = M_i)]$.

Claim 2.1 *For each "job-machine" game there is a pure Nash equilibria.*

Note that not every optimal solution is an equilibria.

Proof:[of Claim 2.1] We define the order $a \leq b$ iff $L(a) \leq_L L(b)$, where $\leq_L$ is lexicographic order on the sorted vector (lexicographic order: $w \leq_L v$ if $w_i = v_i$ for $i = 0, ..., k$, and $w_{k+1} \leq v_{k+1}$) . Let $a^* \in A$ such that for all $b \in A$, $a^* \leq b$.

- a^* exists, (since it is a complete order).
- a^* is an optimal solution for MS (makeSpan), since the first coordinate in the sorted order is the most loaded machine.

We show that a^* is an equilibria. Assume for contradiction that player j gains by moving from M_k to M_l resulting in a joint action b. The load on M_l after the change is smaller than M_k before the change. In addition, M_k after the change is smaller than M_k before the change. Therefor $L(a^*) \geq L(b)$ and we have reached a contradiction to the minimality of a^*. □

2.4 Price of Anarchy

We would like to bound the relation between the worst equilibria and the optimal solution (measured according to MS).

We define the **Price of Anarchy** on pure strategy as

$$PoA = \max_{a \in PNE} \frac{MS(a)}{OPT_{MS}}$$

, where PNE is the set of pure Nash equilibria. And for mixed strategy

$$PoA = \max_{a \in MNE} \frac{E_{a \sim p}[MS(a)]}{OPT_{MS}}.$$

When MNE is the set of mixed Nash equilibria.

Theorem 2.2 *For m machines, $PoA \leq m$.*

Proof: Let $s^* = \max_j s_j$. In the worst case any *Nash equilibrium* is bounded by:

$$MS(a) \leq \frac{\sum_{i=1}^n w_i}{s^*} = W$$

(Otherwise, a player that observes a higher load than W can move to a machine with speed s^* for which its load after the migration is always less than W).

We also have that

$$MS(a) \geq \frac{\sum_{j=1}^n w_j}{\sum_{i=1}^m s_i}.$$

(Which is the case if we can distribute each player's weight in an equal manner over all the machines).

Using the above bounds, we get:

$$PoA \leq \frac{\sum_{i=1}^n w_i / s^*}{\sum_{i=1}^n w_i / \sum_{j=1}^m s_j} = \frac{\sum_{j=1}^m s_j}{s^*} \leq m$$

Since $S_j \leq S^*$, for every machine M_j. □

Claim 2.3 *For every pure Nash equilibria a,*

$$MS(a) \leq m \cdot OPT_{MS}$$

2.5 Two Identical Machines, Deterministic Model

As can be seen in Figure 2.2, at a pure *Nash Equilibrium*, the maximal load is 4. However, the maximal load of the *optimal solution* is only 3. Therefore $PoA = \frac{4}{3}$, in this example we show that this is the worst case.

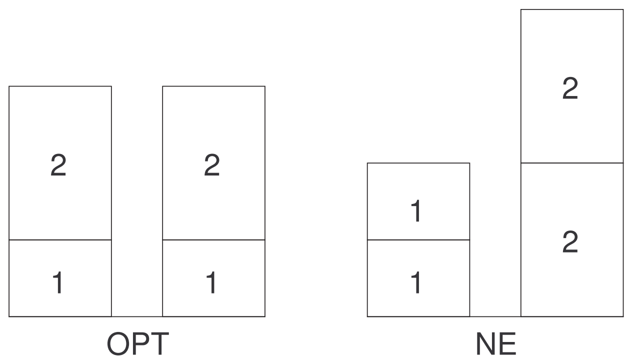

Figure 2.2: Example of $PoA = \frac{4}{3}$

Claim 2.4 *For 2 identical machines and pure Nash equilibria,* $PoA \leq \frac{4}{3}$.

Proof: Without loss of generality, let us assume that $L_1 > L_2$. We define $v = L_2 - L_1$. We have two cases:

a. If $L_2 \geq v$:
By definition$L_1 = L_2 + v$. Therefore $MS = L_2 + v$, and OPT is at least $\frac{L_1+L_2}{2} = L_2 + \frac{v}{2}$. Hence,

$$PoA \leq \frac{L_2 + v}{L_2 + \frac{v}{2}} = 1 + \frac{\frac{v}{2}}{L_2 + \frac{v}{2}} \leq 1 + \frac{\frac{v}{2}}{v + \frac{v}{2}} = \frac{4}{3}.$$

b. If $L_2 < v$:
As before $L_1 = L_2 + v$. Therefore $2L_2 < L_1 < 2v$. If L_1 consists of the weight of more than one player, we will define w to be the weight of the user with the smallest

weight in M_1. Since this is a pure *Nash Equilibrium*, $w > v$. (Otherwise the player would rather move). However, $L_1 < 2v$, hence it is not possible to have two or more players on M_1. Because of this, there is at most one player on M_1 which is the optimal solution, and $PoA = 1$ accordingly.

$\square$

2.6 Identical machines, deterministic users

First we define some variables:

$$w_{max} = \max_i w_i \tag{2.1}$$

$$L_{max} = \max_j L_j \tag{2.2}$$

$$L_{min} = \min_j L_j \tag{2.3}$$

Claim 2.5 *In a* Nash equilibrium*,* $L_{max} - L_{min} \leq w_{max}$

Proof: Otherwise there would be some user j s.t. $w_j \leq w_{max}$, which could switch to the machine with load L_{min}. $\square$

Theorem 2.6 *In identical machines and deterministic users (pure strategies),* $PoA \leq 2$

Proof: We shall distinguish between to cases:

- $L_{min} \leq w_{max}$ In this case $L_{max} \leq L_{min} + w_{max} \leq 2w_{max}$ and since $OPT_{MS} \geq w_{max}$ we conclude that $PoA \leq \frac{L_{max}}{OPT_{MS}} \leq \frac{2w_{max}}{w_{max}} = 2$
- $L_{min} > w_{max}$ Then $L_{max} \leq L_{min} + w_{max} \leq 2L_{min}$, Since the average is greater than its smallest term, i.e., $OPT_{MS} \geq \frac{1}{m}\sum_i L_i \geq L_{min}$, we conclude that $OPT_{MS} \geq L_{min}$ Therefore: $PoA \leq \frac{L_{max}}{OPT} \leq \frac{2L_{min}}{L_{min}} = 2$

$\square$

The upper bound bound is tight

We will give an example in which PoA is $(1 - o(1))2$ and therefore one should not expect a better bound than 2. Consider the following game: m machines and $\frac{m-1}{\epsilon}$ users with a weight of ϵ and two users with jobs of weight 1 as shown in figure 2.3. One can easily verify that this is a *Nash equilibrium* with a cost of 2. The optimal configuration is obtained by scheduling the two "heavy" users (with $w = 1$) on two separate machines and dividing the other users among the rest of the machines. In this configuration we get: $C = OPT = 1 + \frac{1}{m} = 1 + o(1)$

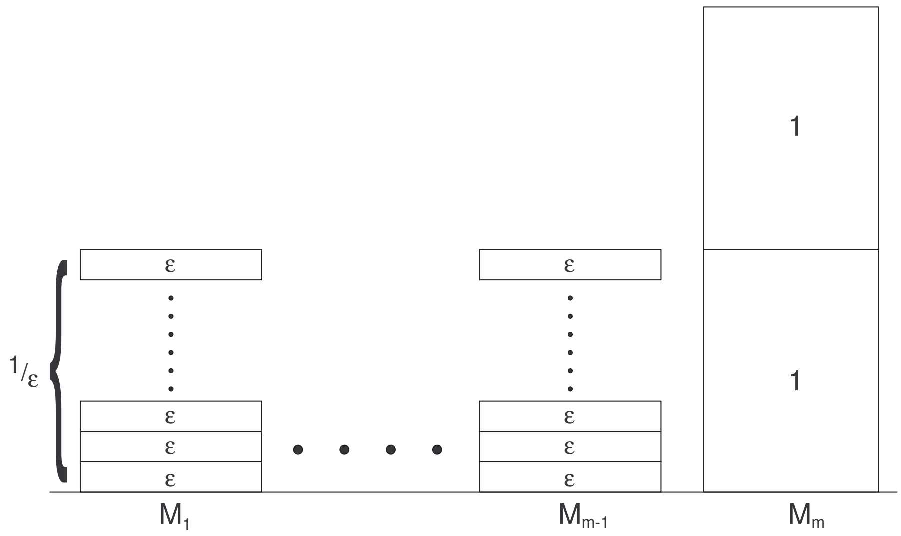

Figure 2.3: PoA comes near to 2

2.7 Two Identical Machines, Stochastic Model

we first consider two identical users, for which $w_1 = w_2 = 1$, as shown in figure 2.4. Each of the players chooses a machine at random. With a probability of 1/2, the players will

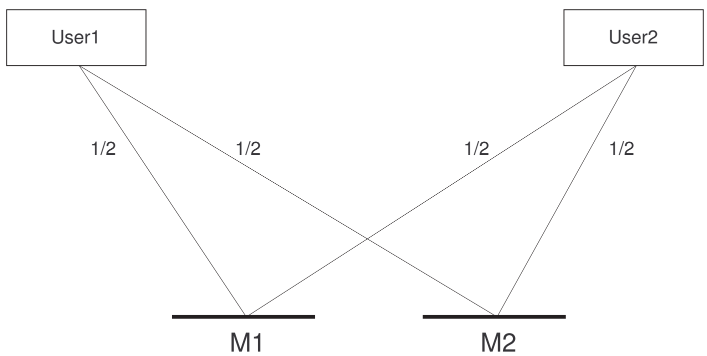

Figure 2.4: Stochastic model example

choose the same machine and with a probability of 1/2,the players choose different machines. Therefore $MS = 1/2{\cdot}2{+}1/2{\cdot}1 = 3/2$. The cost of OPT is 1 and so it follows that $PoA = 3/2$.

2.8 Identical machines, stochastic users

Consider the following example: m machines, $n = m$ users, $w_i = 1$, $p_i(j) = \frac{1}{m}$. What is the maximal expected load? This problem is identical to the following problem: m balls are thrown randomly into m bins; What is the expected maximum number of balls in a single bin? Let us first see what is the probability that k balls will fall into a certain bin:

$$\Pr = \binom{m}{k} \cdot \left(\frac{1}{m}\right)^k \left(1 - \frac{1}{m}\right)^{m-k} \approx \left(\frac{c \cdot m}{k}\right)^k \left(\frac{1}{m}\right)^k = \left(\frac{c}{k}\right)^k$$

The probability that there exists a bin with at least k balls is $1-(1-(\frac{c}{k})^k)^m$. For $(\frac{c}{k})^k \geq \frac{1}{\sqrt{m}}$ the probability that there exist a bin with k balls is $(1 - \frac{1}{\sqrt{m}})^m = e^{-\sqrt{m}}$. For $(\frac{c}{k})^k \leq \frac{1}{m^2}$ this probability is $m \cdot \left(\frac{c}{k}\right)^k < m \cdot \frac{1}{m^2} = \frac{1}{m}$. Therefore for $k \sim \frac{\ln m}{\ln \ln m}$ this probability is a constant and the maximal load is roughly $\frac{\ln m}{\ln \ln m}$.

2.8.1 Upper bound

Similar to the pure Nash equilibrium case, we can bound the expected load in a mixed Nash equilibrium (MNE).

Theorem 2.7 *Let $p \in \Delta$ be MNE then*

$$\overline{L_j} = E[L_j] \leq 2OPT$$

We first state Azuma-Hoeffding Lemma that will be used later in the proof of the theorem.

Lemma 2.8 (Azuma-Hoeffding) *For some random variable $X = \sum x_i$, where x_i are random variables with values in the interval $[0, z]$, is:*

$$P[X \geq \lambda] \leq \left(\frac{e \cdot E[X]}{\lambda}\right)^{\frac{\lambda}{z}}$$

Proof:[of Theorem 2.7] Let us define $\lambda = 2\alpha OPT$, $z = w_{max}$ and $x_i = \begin{cases} w_i & \text{if } p_i(j) > 0 \\ 0 & \text{otherwise} \end{cases}$
Using theorem 2.6 from the deterministic part we know that:

$$\bar{L_j} = E[L_j] \leq 2OPT$$

We wish to prove that the probability of having a machine M_j for which $L_j \gg \bar{L_j}$ is negligible. By applying the inequality we get:

$$P[L_j \geq 2\alpha OPT] \leq \left(\frac{e \cdot E[L_j]}{2\alpha OPT}\right)^{\frac{2\alpha OPT}{w_{max}}} \leq \left(\frac{e}{\alpha}\right)^{2\alpha}$$

which results in

$$P[\exists j\ L_j \geq 2\alpha OPT] \leq m\left(\frac{e}{\alpha}\right)^{2\alpha}$$

Note that for $\alpha = \Omega(\frac{\ln m}{\ln\ln m})$ the probability is smaller than $\frac{1}{2m}$. Since for any $a \in A, MS(a) \leq m \cdot OPT$, we obtain that $E[MS(a)] \leq \alpha \cdot OPT + \left(\frac{1}{m}\right) m \cdot OPT = (\alpha + 1) \cdot OPT$. □

2.9 Non-identical machines, deterministic users

We shall first examine a situation with a 'bad' *Price of Anarchy* of $\frac{\ln m}{\ln\ln m}$, and then establish an upper bound.

2.9.1 Example

Let us have $k + 1$ groups of machines, with N_j machines in group j. The total number of machines $m = N = \sum_{j=0}^{k} N_j$. We define the size of the groups by induction:

- $N_k = \sqrt{N}$
- $N_j = (j + 1) \cdot N_{j+1}$
- $N_0 = k! \cdot N_k$

From the above it results that:

$$k \sim \frac{\ln N}{\ln\ln N}$$

the speed of the machines in group N_j is defined $s_j = 2^j$.

First we set up an equilibrium with a high cost. Each machine in group N_j receives j users, each with a weight of 2^j. It is easy to see that the load in group N_j is j and therefore the make span is k. Note that group N_0 received no users.

Claim 2.9 *This is a Nash equilibrium.*

Proof: Let us take a user in group N_j. If we attempt to move him to group N_{j-k} he will see a load of

$$(j - k) + \frac{2^j}{2^{j-k}} > j$$

On the other hand, on any group N_{j+k} the load is $j + k > j$ even without this job and therefore he has no reason to move there. □

To bound the optimum we simply need to move all the users of group N_j to group N_{j-1} (for $j = 1...k$). Now there is a separate machine for each user and the load on all machines is $\frac{2^j}{2^{j-1}} = 2$. Therefore $OPT \leq 2$.

Corollary 2.10 *The coordination ratio is* $\sim \frac{\ln m}{\ln\ln m}$

2.9.2 Upper Bound

The machines have different speeds; Without loss of generality let us assume that $s_1 \geq s_2 \cdots \geq s_m$. The make span is defined $C = \max L_j$.

For $k \geq 1$, define J_k to be the smallest index in $\{0, 1, \dots, m\}$ such that $L_{J_k+1} < k \cdot OPT$ or, if no such index exists, $J_k = m$. We can observe the following:

- All machines up to J_k have a load of at least $k \cdot OPT$
- The load of the machine with an index of $J_k + 1$ is strictly less than $k \cdot OPT$

Let C^* be defined:

$$C^* = \lfloor \frac{C - OPT}{OPT} \rfloor$$

Our goal is to show that $C^*! < J_1$ which will result in

$$C = O\left(\frac{\log m}{\log \log m}\right) \cdot OPT$$

We will show this using induction.

Claim 2.11 *(The induction base)* $J_{C^*} \geq 1$

Proof: By the way of contradiction, assume $J_{C^*} = 0$. This implies (from the definition of J_k) that $L_1 < C^* \cdot OPT \leq C - OPT$. Let M_q denote the machine with the maximum expected load. Then $L_1 + OPT < C = L_q$.

We observe that any user that uses j on M_q must have a weight w_j larger than $s_1 \cdot OPT$, otherwise j could switch to the fastest machine,M_1, reaching a cost of $L_1 + \frac{w_j}{s_1} \leq L_1 + OPT < L_q$. However, $OPT \geq \frac{w_j}{s_1}$ in contradiction to the stability of the *Nash equilibrium.* □

We shall divide the proof of the induction step into two claims. Let S be the group of users of the machines $M_1, \dots, M_{J_{k+1}}$.

Claim 2.12 *An optimal strategy will not assign a user from group S to a machine M_r such that $r > J_k$.*

Proof: From the definition of J_k, the users in S have a load of at least $(k+1) \cdot OPT$. Machine $J_k + 1$ has a load of at most $k \cdot OPT$. No user from S will want to switch to $J_k + 1$. Therefore, the minimal weight in S is larger than $s_{J_k+1} \cdot OPT$,which implies that if any job in S is run on M_{J_k+1}, then $L_{J_k+1} > OPT$. Switching to machine $r > J_k + 1$ will result in an even larger load because $s_r < s_{J_k+1}$. □

Claim 2.13 *If an optimal strategy assigns users from group S to machines* $1, 2, \dots, J_k$ *then* $J_k \geq (k+1)J_{k+1}$

Proof: Let $W = \sum_{i \in S} w_i$.

$$W = \sum_{j \leq J_{k+1}} s_j \cdot E[L_j] \geq (k+1) OPT \sum_{j \leq J_{k+1}} s_j$$

Since an optimal strategy uses only machines $1, 2, \ldots, J_k$ we get:

$$OPT \sum_{j \leq J_k} s_j \geq W$$

$$\sum_{j \leq J_k} s_j \geq (k+1) \cdot \sum_{j \leq J_{k+1}} s_j$$

Since the sequence of the speeds is non-increasing, this implies that $J_k \geq (k+1) J_{k+1}$, the induction step. $\square$

Now we can combine the two claims above using induction to obtain:

Corollary 2.14 $C^*! < J_1$

By definition $J_1 \leq m$. Consequently $C^*! \leq m$, which implies the following:

Corollary 2.15 *(Upper bound)* $C = O(\frac{\log m}{\log \log m})$