Comparing Nash equilibria to Optimal Solutions in the Load Balancing Game with randomized strategies.

Abstract: A randomized version of the (atomic) load balancing game is presented. Some similarities and differences with the continuous version of the same game are commented. The concept of Nash equilibrium for this game is explicit-ed in terms of the expected load experienced by jobs. It is proven that the maximum expected load experienced by any user in a Nash equilibrium is at most two times the minimum possible.

1 The Load Balancing Game — Randomized version

Recall from the first lectures the setting of the load balancing game. In this game we have a set of n parties, so called *jobs*, which we will denote by j, and a set S containing m *servers* or machines, that will be denoted by i.

As before each job j has an associated subset of servers $S_j \subset S$ to which it can access. However, for the main result of this lecture it will be assumed that $S_j = S$ for all j.

Each job j has a *weight*[1] w_j, with which it is going to load *one* of the servers. So we are dealing here with an *atomic* version of the load balancing game with one major modification. We will allow jobs to flip coins to decide to which server to go. In other words we will be considering randomized strategies instead of pure ones. Formally, each job j decides (independently of the others) a probability distribution between the machines in S_j. For convenience we think of this as a vector

$$\mathbf{p}^j = (p_1^j, p_2^j, \ldots, p_m^j)$$

where p_i^j is the probability that job j goes *as a whole* to machine i. Clearly, we must require $\sum_{i=1}^m p_i^j = 1$, for every job j, and $p_k^j = 0$ when $k \notin S_j$.

Finally, each server has a *response time* function $r_i(L)$, that is a function of the total load on it. By the total load L_i we simply mean the sum of the weights of the jobs on it. For this lecture and the following we will assume

$$r_i(x) = x, \quad \text{(this lecture and the following)}$$

The fact that the jobs can now choose their strategies from a continuum, in a sense splitting their weights between different machines, can let one think that this game is identical as the corresponding continuous version of the load balancing game. As we will show this is not exactly true. A significant difference will be apparent when we explicitly describe what the goal of the game is and how a Nash equilibrium for it looks like.

[1] This weight is conceptually the same as our former processing time. We have changed the terminology just for clarity in the notation.

In this game each job wants to maximize its own welfare in a selfish manner. The goal of each job stated in English it would be: "to be on a machine with small load (response time)."

Since there is randomness involved we must be a little careful when stating mathematically what this means. We will assume each player is *risk neutral*. This means that its *payoff* is just the expected load on the machine it is on.

By *expected load on machine* i we mean, of course,

$$L_i = \sum_j p_i^j w_j.$$

It is tempting to guess that the expected load experienced by job j is $c'_j = \sum_i p_i^j L_i$.

Despite the appealing simplicity of this expression, it does not correspond to reality. Its major fault consists in not taking into account the fact that, once a job chooses a machine, its full weight is completely there. This can be more clearly seen with an example. Suppose there is only one job with weight $w_1 = 1$ that chooses to go to each of the m machines with equal probability $p_i^1 = \frac{1}{m}$. We then have $L_i = \frac{1}{m} \cdot 1 = \frac{1}{m}$ and the expression above would give $c'_j = \sum_i p_i^j \frac{1}{m} = \frac{1}{m}$. On the other hand the load experienced by job j is 1 in all situations, because "it is always in the machine it went to". Thus, the expected load experience by j cannot be given by c'_j.

A few moments of thought are enough to realize that the correct expression for the *expected load experienced by job* j is

$$c_j = \sum_i p_i^j \left(w_j + \sum_{k \neq j} p_i^k w_k \right) = \sum_i p_i^j \left(L_i + (1 - p_i^j) w_j \right) \quad (1)$$

The expression in parenthesis is nothing more than the expected load of machine i conditioning on the event that job j goes there. Summing overall machines i with the corresponding probability of j going to machine i gives the desired expected load experienced by j. The last term in the last expression corresponds to the correction to the expected load L_i that originates from conditioning j on itself.

2 Nash Equilibria

From the last comments of the preceding section it is easy to infer that a set of probability distributions (mixed strategies) form a *Nash equilibrium* if and only if for any job j and any machine machine i, for which $p_i^j > 0$, the expected load of this machine, assuming j goes to i, is at most the expected load of any other machine s, assuming j goes there. Formally:

A set $\{\mathbf{p}^j\}_j$ *of probability distributions is a Nash equilibrium if and only if for all jobs* j *and all machines* $i, s \in S_j$ *we have the implication*

$$p_i^j > 0 \quad \Rightarrow \quad L_i + (1 - p_i^j) w_j \leq L_s + (1 - p_s^j) w_j. \quad (2)$$

An important point here is that for any i the expression $L_i + (1 - p_i^j) w_j$ does not depend on $\mathbf{p}^j$, as can be easily checked from the definition of L_i. In words, the probability distribution belonging to job j has no effect on the expected load experienced by j on a particular server i. Thus, the changing of the probabilities $\mathbf{p}^j$ will not affect any of the expected experienced loads of j. It will only affect other job's experiences.

But does a Nash equilibrium exist? Sure, this follows immediately from the general theorem that assures the existence of Nash equilibria for games with a finite number of pure strategies.

Last, we note that any deterministic Nash is also a randomized Nash in which for each job one of the probabilities is set to 1 and the rest to 0. This can be checked directly from the characterization (2).

3 Measuring the quality of a solution

Given that c_j measures the cost of a particular solution to player j, the following are some of the natural measures of the social cost that would allow us to define a social optimum.

1. $\sum_j c_j$ (average social cost)
2. $\max_j c_j$

In what follows we are going to compare the quality of the worst Nash to that of the optimal solution with respect to the second measure $\max_j c_j$, for the special case in which $S_j = S$ for all j. In fore-coming lectures we are going to analyze still a third measure of quality which consists in taking the expected value of the maximum load on any machine.

4 Comparing the quality of the worst Nash to that of the optimal solution with respect to $\max_j c_j$

Recall that in the first lecture we had proved for the atomic version of the load balancing game with deterministic (pure) strategies, that the maximum load on any machine in a Nash equilibrium is at most twice the minimum possible. Here we prove an analogous statement for the present game by essentially copying the proof of the old result.

Theorem 1 *Let c_j denote the expected load experienced by job j in any Nash equilibrium and let c_j^* denote the same quantity for the best possible random assignment in the load balancing game. If $S_j = S$, then*

$$\max_j c_j \leq 2 \cdot \max_j c_j^*.$$

Proof. Set $OPT := \max_j c_j^*$. Noticing from (1) that $c_j^* = w_j + \sum_i p_i^{*j} \left(\sum_{k \neq j} p_i^{*k} w_k \right) \geq w_j$ we conclude get

$$OPT \geq \max w_j \tag{3}$$

we argue now that

$$OPT \geq \frac{1}{m} \sum_j w_j \tag{4}$$

To see this recall that, by definition, we have $c_j^* \geq \sum_i p_i^j L_i^*$, in the optimal solution. Multiplying these inequalities by w_j and then adding them up gives

$$\begin{aligned}
\sum w_j c_j^* &= \sum_j \sum_i w_j p_i^j L_i^* \\
&= \sum_i \left(\sum_j w_j p_i^j \right) L_i^* \\
&= \sum_i L_i^{*2} \\
&\geq \frac{1}{m} (\sum_i L_i^*)^2 \qquad (5) \\
&= \frac{1}{m} (\sum_j w_j)^2,
\end{aligned}$$

where, in (5), we have used the *power mean- arithmetic mean* inequality $\sum_i L_i^{*2} \geq \frac{1}{m} \left(\sum_i L_i^*\right)^2$ [2] and then we have used the fact

$$\sum_i L_i = \sum_{i,j} p_i^j w_i = \sum_i w_i,$$

when summing first over j. Now, since the value of OPT is by definition the biggest experienced load among the c_j's, using the result just showed we can write

$$\sum_j w_j OPT \geq \sum_j w_j c_j \geq \frac{1}{m} \left(\sum_j w_j \right)^2$$

which gives (4) upon dividing by $\sum_j w_j$ on both sides.

Now denote by J the machine with worst expected experience load in the Nash, that is, $c_J = \max_j c_j$. We know, from the characterization of Nash equilibria, that

$$c_J \leq L_s + (1 - p_s^J) w_J$$

for all other servers $s \in S \, (= S_J)$. Summing over all s we get

$$m c_J \leq \sum_s \left[L_s + (1 - p_s^J) w_J \right] = \left(\sum_{k=1}^n w_k \right) + (m-1) w_J \leq \sum_k w_k + m w_J$$

After dividing by m, the result now follows from (3) and (4)

$$c_J \leq \frac{1}{m} \sum_k w_k + w_J \leq OPT + OPT.$$

[2] This inequality is a special case of Jensen's inequality applied to the convex function $f(x) = x^2$, but has also a quick independent proof. We start with the generic inequality $0 \leq \sum_i (L_i^* + t)^2 = \sum_i L_i^{*2} + 2 \left(\sum L_i^* \right) t + m t^2$ that holds for any real t. Now putting $t = -(\frac{1}{m} \sum_i L_i^*)$ gives $\sum_i L_i^{*2} - \frac{1}{m} \left(\sum L_i^* \right)^2 \geq 0$ which easily yields the desired inequality.

$\blacksquare$

The moral of the story is that deterministic Nashes are not much worse than randomized ones. Also, if one wants to prove something about Nash equilibria in general a first step should be to prove it for deterministic Nashes, which are special cases of the former and are also easier to work with. After that one might think of translating the proof for the case of mixed strategies.

5 Complementary References

- E. Koutsoupias, C. H. Papadimitriou. *Worst-case equilibria*, Symposium on Theoretical Aspects of computer science (STACS) 99E.