Distributed Learning of Equilibria in a Scheduling Problem

Johanne Cohen

PRiSM/CNRS, Versailles, France.

Join work with Olivier Bournez (LIX, Ecole Polytechnique).

Load balancing game.

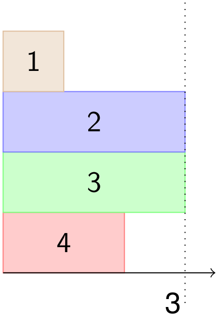
- m machines with speeds $s_1, \ldots, s_m$
- n tasks with weights $w_1, \ldots, w_n$.
  - each task is managed by a selfish agent.
  - each agent aims to placing the task on the machine with smallest load.
- the load ℓ_j of machine j under assignment A :

$$\ell_j = \frac{\sum_{i:A[i]=j} w_i}{s_j}$$

- The cost c_i of an agent i = load of the machine $A[i]$.

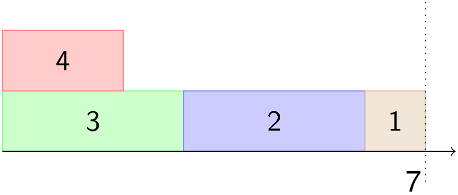
Load balancing game

- 4 jobs : $w_1 = 1$, $w_2 = w_3 = 3$, $w_4 = 2$,
- 2 identical machines



Load balancing game

- 4 jobs : $w_1 = 1$, $w_2 = w_3 = 3$, $w_4 = 2$,
- 2 identical machines

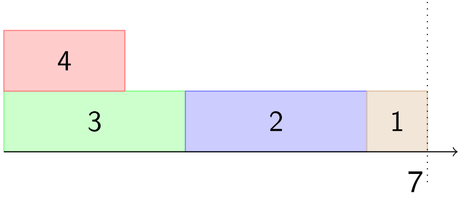


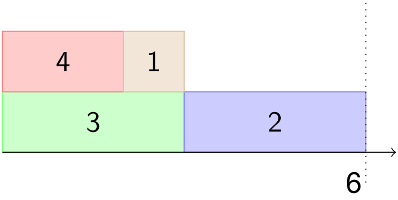
Assignment A_1 where

- $c_1 = c_2 = c_3 = 7$
- $c_4 = 2$

Load balancing game

- 4 jobs : $w_1 = 1$, $w_2 = w_3 = 3$, $w_4 = 2$,
- 2 identical machines





Assignment A_1 where

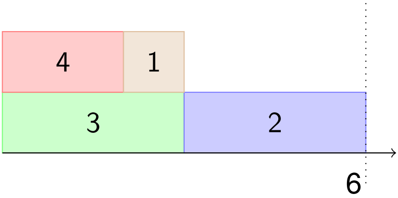
- $c_1 = c_2 = c_3 = 7$
- $c_4 = 2$

Assignment A_2 where

- $c_2 = c_3 = 6$
- $c_4 = c_1 = 3$

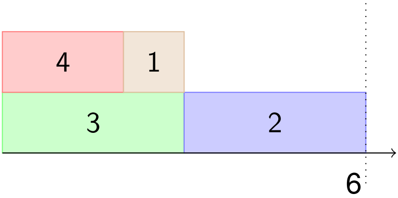
Nash Equilibrium (NE)

An assignment A is a (pure) Nash Equilibrium if and only if no agent can improve its cost by unilaterally moving its task to another machine.



Nash Equilibrium (NE)

An assignment A is a (pure) Nash Equilibrium if and only if
for all agents i, and for all machines k,
we have : $c_i^{A[i]} \leq c_i^k$



Mixed strategy

A mixed strategy q_i for the agent i is a probability vector

$$q_i = (q_{i,1}, \ldots, q_{i,m})$$

where i chooses the machine k with the probability $q_{i,k}$.

The strategy profil $Q = (q_1, \ldots, q_n)$ corresponds to the mixed strategies of all agents.

Mixed strategy

A mixed strategy q_i for the agent i is a probability vector

$$q_i = (q_{i,1}, \ldots, q_{i,m})$$

where i chooses the machine k with the probability $q_{i,k}$.

The strategy profil $Q = (q_1, \ldots, q_n)$ corresponds to the mixed strategies of all agents.

Consequence :

- Each profil Q induces a random mapping
- Each agent aims at minimizing its expected cost.
- The notion of Nash equilibrium can be extended.

Previous works

1. Do the load balancing games have pure Nash Equilibria ?

Previous works

1. Do the load balancing games have pure Nash Equilibria ?

Theorem [Fotakis2002,Even-Dar2003]

Every instance of the load balancing game admits at least one pure Nash equilibrium.

Theorem [Fotakis2002]

The Largest Processing Time algorithm computes a pure Nash equilibrium in the load balancing games.

Previous works

1. Do the load balancing games have pure Nash Equilibria ?

2. What are their performance ?

Previous works

1. Do the load balancing games have pure Nash Equilibria ?

2. What are their performance ?

   Price of Anarchy $= max_G max_{P \ is \ NE} \frac{cost(P)}{opt(G)}$

	Identical machines	Uniformaly related machines
Pure NE	$2 - \frac{1}{m}$	$\Theta\left(\frac{\log m}{\log\log m}\right)$
Mixed NE	$\Theta\left(\frac{\log m}{\log\log m}\right)$	$\Theta\left(\frac{\log m}{\log\log\log m}\right)$

- 1999 : Koutsoupias & Papadimitriou
- 2001 : Mavronicolas & Sirakis,
- 2002 : Czumaj & Vöcking

Previous works

1. Do the load balancing games have pure Nash Equilibria ?

2. What are their performance ?

3. How to learn equilibria ? How fast is the convergence ?

    - n agents with a set of strategies

    - The game is repeated :
      at each step t, the agents play their strategy.

    - Can the system converge to a Nash equilibrium ?

Learning Equilibria : Definition

- $q_i(t)$: a mixed strategy for agent i at time t

  (a probability vector over pure strategies)

- $q_i(t+1) = learning\_algorithm(\{q_j(t')\}_{j \text{ agent}, t' \leq t})$

- Find a learning algorithm such that for all i

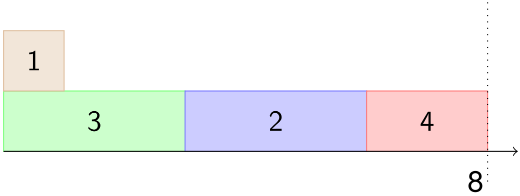
  When $t \to \infty$, $\qquad q_i(t) \to q_i^*$

  where $(q_1^*, \cdots, q_N^*)$ is an Nash Equilibrium

Best Response Learning Algorithm : Principle

The best response policy activates the *unsatisfied* agents one after the other :

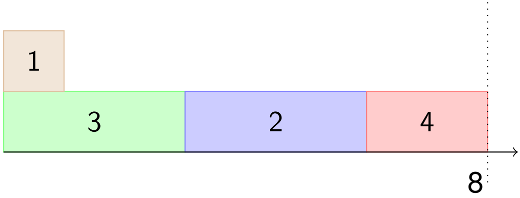
> the active agent plays a best reponse (machine with minimum load).



Best Response Learning Algorithm : Principle

The best response policy activates the *unsatisfied* agents one after the other :

> the active agent plays a best reponse (machine with minimum load).

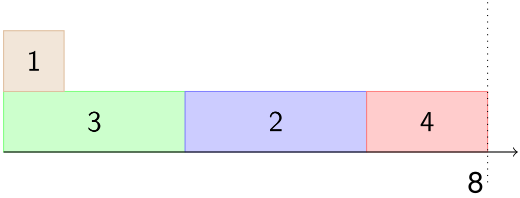


the *unsatisfied* agents are 2, 3 and 4

Best Response Learning Algorithm : Principle

The best response policy activates the *unsatisfied* agents one after the other :

> the active agent plays a best reponse (machine with minimum load).

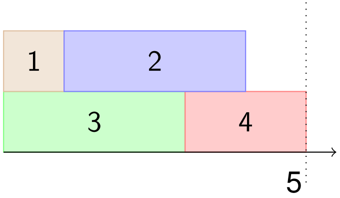


agent 2 is activated

Best Response Learning Algorithm : Principle

The best response policy activates the *unsatisfied* agents one after the other :

> the active agent plays a best reponse (machine with minimum load).



assignment after the move

Best Response Learning Algorithm : Properties

Theorem [Even-Dar2003]

Let A be any assignement of n tasks to m identical machines. Starting from A, best response policy eventually reaches a pure Nash equilibrium.

Theorem [Berenbrink2006]

A stochatic extension of this policy reaches eventually a pure Nash equilibrium in an expected number of rounds $O(\log \log n + m^4)$ whenever all tasks are identical.

Our Dynamic : replication dynamic

Algorithm :

- For all agents i, choose any initial mixed strategy $q_i(0)$
- At each step t, activate randomly an unique task i.
- For the active agent i :
  - Choose a machine a_i according to distribution $q_i(t)$
  - Move i to this machine, and measure cost c_i of this choice.
  - Update the mixed strategy

$$q_i(t+1) = q_i(t) + \boxed{b} \times \boxed{r_i(t)} \times (\boxed{\mathbf{1}_{a_i}} - q_i(t))$$

- $\boxed{b}$: a constant step size
- $\boxed{r_i(t)} = 1 - c_i$
- $\boxed{\mathbf{1_{a_i j}}} = \begin{cases} 1 & if\ j = a_i \\ 0 & otherwise \end{cases}$

The Main Ideas

- $Q = (q_1, \ldots, q_n)$: the global state of the system (q_i is the mixed strategy of i).
- F : a function over the global state Q of the system where

$$F(Q) = \sum_{k=1}^{m} 1/s_k \left[\frac{1}{2}(\sum_{j=1}^{N} q_{j,k} w_j)^2 + \sum_{j=1}^{N} q_{j,k} w_j^2 (1 - \frac{q_{j,k}}{2}) \right]$$

Property 1 : Each step of the algorithm changes Q (one component q_i) in a way such that

$$\Delta F(Q) = \mathbb{E}[F(Q(t+1)) - F(Q(t)|Q(t)]) \leq 0$$

Corollary : F will decrease with high probability, and F will converge with high probability (almost surely). A limit point must vanishes ΔF.

Property 2 : Nash Equilibria correspond to such points.

Property 1 : $\Delta F \leq 0$

- Fix the probabilistic choices (the i and machine a_i).

$$F(Q(t+1)) - F(Q(t)) = w_i \sum_{k=1}^{m} \Delta_{i,k} \times h_{i,k}$$

where

$$\Delta_{i,k} = b \times r_i(t) \times (1_{k=a_i} - q_{i,k}(t))$$
$$h_{i,k} = w_i + \sum_{j \neq i} q_{j,k} w_j.$$

Property 1 : $\Delta F \leq 0$

- Fix the probabilistic choices (the i and machine a_i).

$$F(Q(t+1)) - F(Q(t)) = w_i \sum_{k=1}^{m} \Delta_{i,k} \times h_{i,k}$$

where

$$\Delta_{i,k} = \text{modification of } q_i(t)$$
$$h_{i,k} = \text{expected cost of } i \text{ when } i \text{ chooses } k$$

Property 1 : $\Delta F \leq 0$

- Fix the probabilistic choices (the i and machine a_i).

$$F(Q(t+1)) - F(Q(t)) = w_i \sum_{k=1}^{m} \Delta_{i,k} \times h_{i,k}$$

where

$$\Delta_{i,k} = \text{modification of } q_i(t)$$
$$h_{i,k} = \text{expected cost of } i \text{ when } i \text{ chooses } k$$

- Now,

$$\Delta F = \mathbb{E}[F(Q(t+1)) - F(Q(t)) | Q(t)]$$

$$\Delta F = -\frac{b}{n} \sum_{i=1}^{n} w_i \sum_{k=1}^{m} \sum_{k'>k} q_{i,k} q_{i,k'} [h_{i,k} - h_{i,k'}]^2$$

$$\Delta F \leq 0$$

Corollary : F Will Decrease

Since $\Delta F = \mathbb{E}[F(Q(t+1)) - F(Q(t))|Q(t)] \leq 0$,

F will decrease with high probability (almost surely).

Formal Statement (Martingale Theory / Foster's Theorem) :
Let $Q_0, Q_1, \cdots, Q_n, \cdots$ be a sequence of random variables with values in E such that for some function $F : E \rightarrow \mathbb{R}^{\geq 0}$, $\mathbb{E}[F(Q_{n+1})|Q_n] \leq F(Q_n) - \epsilon$ whenever $Q_n \notin S$, where S is a set.

Then, for any Q_0, Q_n will reach S almost surely.

Corollary : F Will Decrease Rapidly

Since $\Delta F = \mathbb{E}[F(Q(t+1)) - F(Q(t))|Q(t)] \leq 0$,

F will decrease with high probability (almost surely).

Formal Statement (Martingale Theory / Foster's Theorem) :
Let $Q_0, Q_1, \cdots, Q_n, \cdots$ be a sequence of random variables with values in E such that for some function $F : E \rightarrow \mathbb{R}^{\geq 0}$, $\mathbb{E}[F(Q_{n+1})|Q_n] \leq F(Q_n) - \epsilon$ whenever $Q_n \notin S$, where S is a set.

Then, for any Q_0, Q_n will reach S almost surely.

Furthermore

$$\mathbb{E}[Reach(S)] \leq \frac{F(Q_0)}{\epsilon}$$

where $Reach(S)$ is the corresponding stopping time (time before reaching S for the first time).

Corollary : Speed of Convergence

- Recall $\Delta F = -b\frac{1}{n}G$ where

$$G = \sum_{i=1}^{n} w_i \sum_{k=1}^{m} \sum_{k'>k} q_{i,k} q_{i,k'} [h_{i,k} - h_{i,k'}]^2$$

- Fix some $\epsilon > 0$. Denote by

$$Inf(\epsilon) = \{X | G(X) \leq \epsilon\}.$$

- For all $\epsilon > 0$

$$\mathbb{E}[Reach(Inf(\epsilon))] \leq \frac{nF(Q(0))}{b\epsilon}$$

$$\mathbb{E}[Reach(Inf(\epsilon))] \leq \frac{3n(\sum_{j=1}^{N} w_j^2)(\sum_{k=1}^{m} 1/s_k)}{2b\epsilon}$$

Property 2 : Limit Points Correspond to Nash Equilibria

- F decreases almost surely whenever

$$\Delta F = -b\frac{1}{n}\sum_{i=1}^{n} w_i \sum_{k=1}^{m}\sum_{k'>k} q_{i,k}q_{i,k'}[h_{i,k} - h_{i,k'}]^2 < 0$$

- Hence, limit points ($\Delta F = 0$) are those such that for all i $h_{i,k} = h_{i,k'}$ for all k, k' such that $q_{i,k} \neq 0$, $q_{i,k'} \neq 0$.

- The limit points correspond to Nash equilibria. (more details)

Conclusion

- We considered a particular algorithm to learn Nash equilibria.
- We proved almost-sure convergence of our algorithm to Nash equilibria.

  previous proofs were only (complicated) proofs that it can not do anything else + weak convergence arguments instead of almost-sure convergence.

- Only pure Nash equilibria are stable : all non-pure equilibria will be left almost-surely (combined with some results [Coucheney-Gaujal-Touati 2009]).
- This is the first time a speed of convergence for this algorithm is proved formally.

Other well-known characterization of NE.

A strategy profile Q is a Nash equilibrium if and only if

for all agents i, for all machines j,

$q_{i,k} > 0 \rightarrow$ for all other machines k, $h_{i,j} \leq h_{i,k}$

Where

$h_{i,k} =$ expected cost of i when i chooses k

back