Game Theoretic Analysis of the Multi-Organization Scheduling Problem

Johanne Cohen[2] **Daniel Cordeiro**[1] Denis Trystram[1]
Frédéric Wagner[1]

[1]Laboratoire d'Informatique de Grenoble
Grenoble University

[2]Laboratoire d'informatique PRiSM
Versailles Saint-Quentin-en-Yvelines University

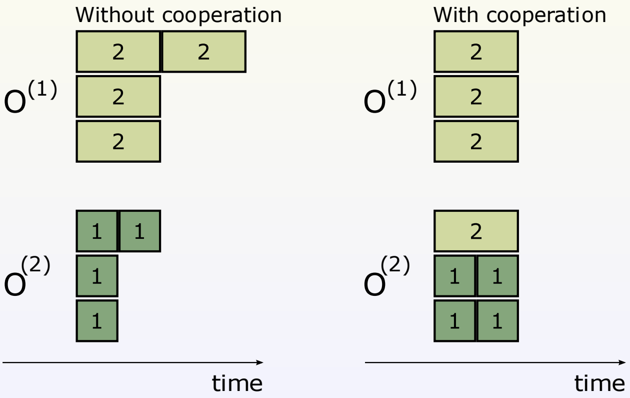
Workshop on Algorithms and Techniques for Scheduling on Clusters and Grids

Outline

Outline

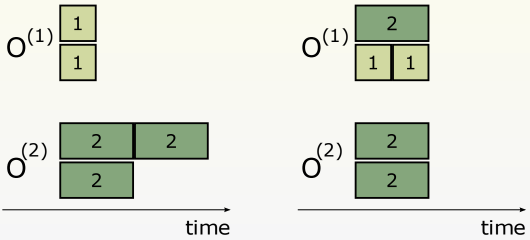
The importance of cooperation

- Current global computing technology (e.g. grid computing systems) makes very clear the importance of creating coalitions of computational resources.



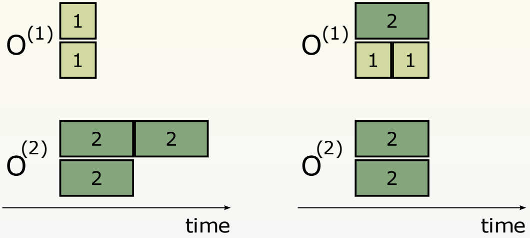
Goal: encourage collaboration

- If each organization cooperates unconditionally, we can achieve the best utilization possible of the available resources.



Goal: encourage collaboration

- If each organization cooperates unconditionally, we can achieve the best utilization possible of the available resources.



- Although (if you look closely) sometimes some concessions must be made:
  - C_{max} that $O^{(1)}$ can achieve by itself: **1**
  - C_{max} of $O^{(1)}$ in the global optimum configuration: **2**

Goal: encourage collaboration

What if we have only *selfish* organizations with specific performance goals?

- An organization could just leave the coalition and do all the work by itself instead of helping others (which is even worse for the entire community).

Our goal is to provide a scheduling mechanism that can improve the global performance of the system while assuring that the local performance of each organization will not be penalized for cooperating with others.

Outline

The problem

The multi-organization scheduling problem can be defined as the problem of minimizing the maximum completion time (makespan) of all jobs and, at the same time, minimize locally:

- the makespan of k organizations
  MOSP($k : C_{max}$)
- the average completion time of k organizations
  MOSP($k : \sum C_i$)

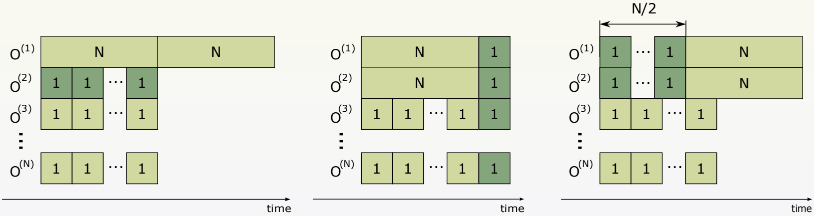
Under the additional constraint that
no local schedule criterion can be increased.

Model

- N organizations, where each organization $O^{(k)}$ has $m^{(k)}$ identical processors and $n^{(k)}$ jobs to be executed;
- Each job $J_i^{(k)} (1 \leqslant i \leqslant n^{(k)})$ requires exactly $q_i^{(k)}$ processors for $p_i^{(k)}$ units of time;
- Each user submits his/her own jobs locally in his/her organization.

Impact of the local constraint

- What makes this problem interesting is the additional constraint that no local schedule can be worsen if compared with the schedule that one organization can obtain by itself.



- The ratio between the optimal solution and the optimal without the local constraints is asymptotically equal to $\frac{3}{2}$.

Previous work

- This problem was first introduced by [Pascual et al., Europar'07], that proposed an algorithm and a load-balancing heuristic called ILBA for parallel rigid jobs;
- Dutot et al. refined the algorithm and obtained a 3-approximation algorithm with tight bound for parallel rigid jobs.

Related work

Without the local constraint introduced by MOSP, this problem is equivalent to the *Multiple Strip Packing Problem*.

- [Schwiegelshohn et al., IPDPS'08] studied this problem in the context of grid computing systems. They proposed an 3-approximation algorithm for the offline case and a 5-approximation for the online case;
- Christina Otte and Klaus Jansen just presented their new results on this problem.

Outline

Introduction

- We are working on modeling MOSP as a non-cooperative game;
- MOSP constraint of not worsening the local objective makes the problem tricky;
- We will focus in the case where all organizations have only one machine ($m^{(k)} = 1, 1 \leqslant k \leqslant N$).

Less jobs makes the problem easier?

- The general MOSP problem is NP-hard. Taking $N = 1$, $m^{(k)} = 2$ and $q_i^{(k)} = 1$, $(\forall i, k)$ we have the classical $P2||C_{\max}$ scheduling problem;
- What if we have one machine per organization ($m^{(k)} = 1$), only 2 jobs per organization ($n^{(k)} = 2$) and sequential jobs ($q_1^{(k)} = q_2^{(k)} = 1$)?

NP-completeness

- Even with less jobs, the problem is NP-Complete in the strong sense.
- Proof: reduction from 3-PARTITION problem.

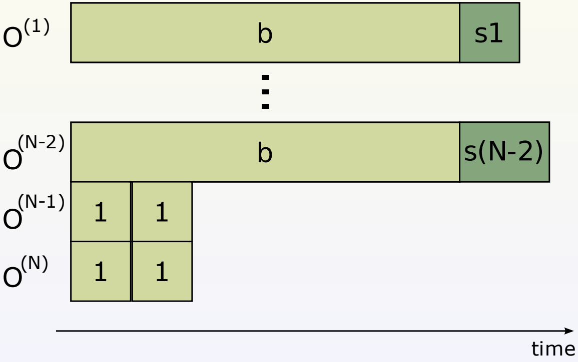
The decision problem version can be defined as follows:

Instance: the number N of organizations, the size of all jobs $p_i^{(k)}$ and an integer K;

Question: does there exist a feasible scheduling with $C_{max} = \max\limits_{i,k}\{p_i^{(k)}\} \leqslant K$?

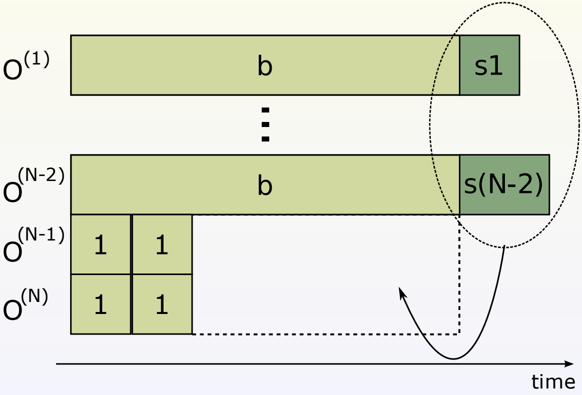
Sketch of the proof

- First, lets see how to reduce from the 2-PARTITION problem:



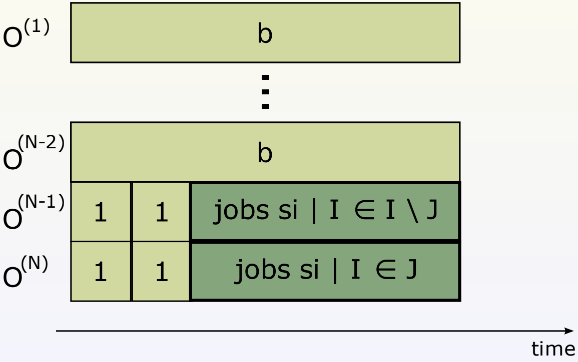
Sketch of the proof

- First, lets see how to reduce from the 2-PARTITION problem:



Sketch of the proof

- First, lets see how to reduce from the 2-PARTITION problem:

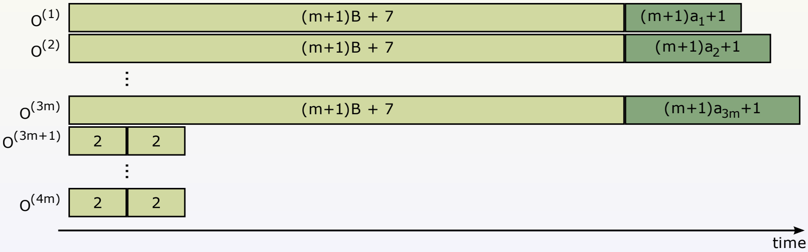


Sketch of the proof

- In the 3-PARTITION problem we want to partition a set of $3m$ integers (that sums up to mB) into m disjoint sets composed of exactly three elements (that sums up to B).
- To extend this proof to reduce from 3-PARTITION we must take:
  - An instance of 3-PARTITION $(\{a_1, \ldots, a_{3m}\},\ B)$, where $\sum_{i=1}^{3m} a_i = mB$;
  - $N = 4m$ organizations;
  - For the first $3m$ organizations, we set $p_1^{(k)} = (\mathbf{m} + \mathbf{1})\mathbf{B} + \mathbf{7}$ and $p_2^{(k)} = (\mathbf{m} + \mathbf{1})\mathbf{a_k} + \mathbf{1}, \forall k \in [1; 3m]$;
  - For the remaining organizations ($3m + 1$ to $4m$), we set $p_1^{(k)} = p_2^{(k)} = \mathbf{2}, \forall k \in [3m + 1; 4m]$ (the last m organizations have two jobs of size 2).

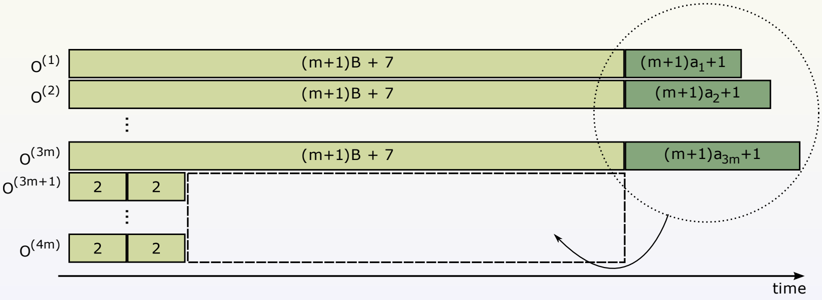
Sketch of the proof

- We can build an optimal schedule for the described instance with makespan exactly equal to $(m+1)B+7$:



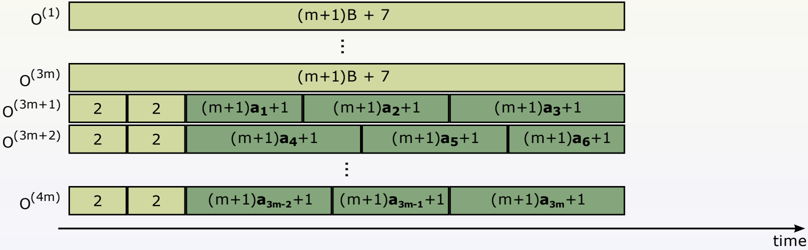
Sketch of the proof

- We can build an optimal schedule for the described instance with makespan exactly equal to $(m+1)B+7$:



Sketch of the proof

- We can build an optimal schedule for the described instance with makespan exactly equal to $(m+1)B+7$:



Proposed model

- We are studying a non-cooperative game defined as follows:
  - Each player is an organization responsible for an "application" (a set of $n^{(k)}$ jobs) and wants to minimize its $cost^{(k)}$ (completion time of its last job, average completion time, etc.);
  - Each organization applies some schedule algorithm locally (LPT, SPT, etc.) putting its own jobs first;
  - A strategy $S^{(k)}$ for player k is a vector of $n^{(k)}$ elements such that $S^{(k)}(i)$ corresponds to the organization chosen by player k for job $J_i^{(k)}$;
  - A configuration (profile) M is the vector $(S^{(1)}, S^{(2)}, \ldots, S^{(N)})$ such that $S^{(k)}$ is a strategy of player k.

Nash equilibrium

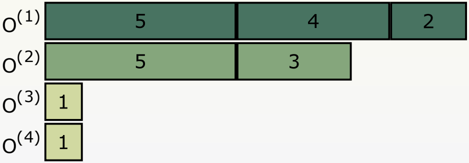
- A configuration $M = (S^{(1)}, S^{(2)}, \ldots, S^{(N)})$ is a Nash equilibrium if all players k (applications) satisfies the following property:

  $\forall s \in \mathcal{S}^{(k)}, cost^{(k)}(M) \leqslant cost^{(k)}(s, M_{-k})$, where M_{-k} is a vector $(S^{(1)}, S^{(2)}, S^{(k-1)}, S^{(k+1)} \ldots, S^{(N)})$

- Do there always exist Nash Equilibria for MOSP($k : C_{max}$) or MOSP($k : \sum C_i$)?

Nash equilibrium and MOSP($k : C_{max}$)

If every organization uses LPT and puts its jobs first, then there are instances of MOSP($k : C_{max}$) where we do not have equilibrium:

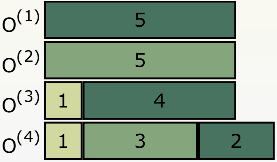


$$\mathbf{C}^{(1)}_{\mathbf{max}} = \mathbf{11}$$
$$\mathbf{C}^{(2)}_{\mathbf{max}} = \mathbf{8}$$

Suppose this initial configuration.

Nash equilibrium and MOSP($k : C_{max}$)

If every organization uses LPT and puts its jobs first, then there are instances of MOSP($k : C_{max}$) where we do not have equilibrium:

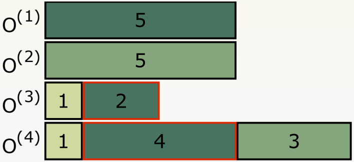


$$\mathbf{C_{max}^{(1)} = 6}$$
$$\mathbf{C_{max}^{(2)} = 5}$$

What if $O^{(1)}$ changes its strategy?

Nash equilibrium and MOSP(k : C_{max})

If every organization uses LPT and puts its jobs first, then there are instances of MOSP(k : C_{max}) where we do not have equilibrium:

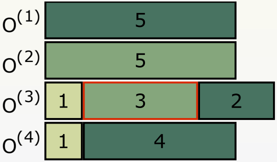


$$\mathbf{C}^{(1)}_{\mathbf{max}} = \cancel{6}\ \mathbf{5}$$
$$\mathbf{C}^{(2)}_{\mathbf{max}} = \cancel{5}\ \mathbf{8}$$

What if $O^{(2)}$ changes its strategy?

Nash equilibrium and MOSP($k : C_{max}$)

If every organization uses LPT and puts its jobs first, then there are instances of MOSP($k : C_{max}$) where we do not have equilibrium:

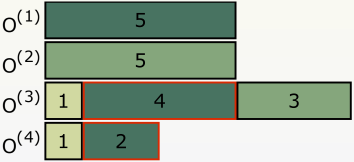


$$\mathbf{C^{(1)}_{max}} = \cancel{5}\ \mathbf{6}$$
$$\mathbf{C^{(2)}_{max}} = \cancel{8}\ \mathbf{5}$$

What if $O^{(1)}$ changes its strategy?

Nash equilibrium and MOSP($k : C_{max}$)

If every organization uses LPT and puts its jobs first, then there are instances of MOSP($k : C_{max}$) where we do not have equilibrium:

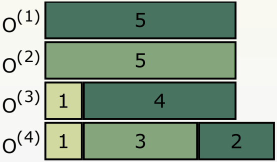


$$\mathbf{C_{max}^{(1)}} = \cancel{6}\ \mathbf{5}$$
$$\mathbf{C_{max}^{(2)}} = \cancel{5}\ \mathbf{8}$$

What if $O^{(2)}$ changes its strategy?

Nash equilibrium and MOSP($k : C_{max}$)

If every organization uses LPT and puts its jobs first, then there are instances of MOSP($k : C_{max}$) where we do not have equilibrium:

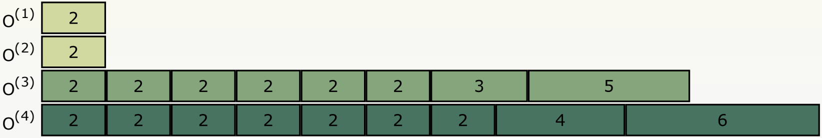


$$\mathbf{C}_{\mathbf{max}}^{(1)} = \cancel{5}\ \mathbf{6}$$
$$\mathbf{C}_{\mathbf{max}}^{(2)} = \cancel{8}\ \mathbf{5}$$

Loop!

Nash equilibrium and MOSP($k : \sum C_i$)

If every organization uses SPT and puts its jobs first, then there are instances of MOSP($k : \sum C_i$) where we do not have equilibrium:

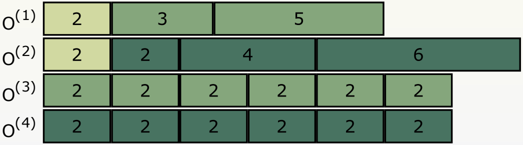


$$\sum \mathbf{C_i^{(3)}} = \mathbf{77}$$
$$\sum \mathbf{C_i^{(4)}} = \mathbf{98}$$

Suppose this initial configuration.

Nash equilibrium and MOSP($k : \sum C_i$)

If every organization uses SPT and puts its jobs first, then there are instances of MOSP($k : \sum C_i$) where we do not have equilibrium:

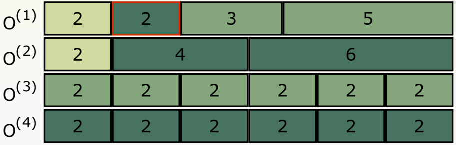


$$\sum \mathbf{C_i^{(3)}} = \mathbf{57}$$
$$\sum \mathbf{C_i^{(4)}} = \mathbf{68}$$

What if $O^{(4)}$ changes its strategy?

Nash equilibrium and MOSP($k : \sum C_i$)

If every organization uses SPT and puts its jobs first, then there are instances of MOSP($k : \sum C_i$) where we do not have equilibrium:

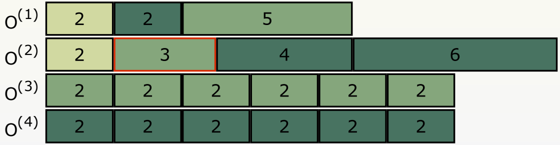


$$\sum \mathbf{C_i^{(3)}} = \cancel{57}\ \mathbf{61}$$
$$\sum \mathbf{C_i^{(4)}} = \cancel{68}\ \mathbf{60}$$

What if $O^{(3)}$ changes its strategy?

Nash equilibrium and MOSP($k : \sum C_i$)

If every organization uses SPT and puts its jobs first, then there are instances of MOSP($k : \sum C_i$) where we do not have equilibrium:

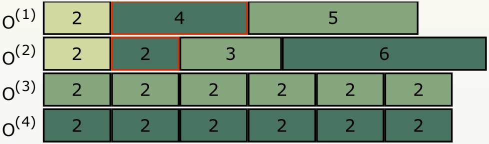


$$\sum \mathbf{C_i^{(3)}} = \cancel{61}\ \mathbf{56}$$
$$\sum \mathbf{C_i^{(4)}} = \cancel{60}\ \mathbf{70}$$

What if $O^{(4)}$ changes its strategy?

Nash equilibrium and MOSP($k : \sum C_i$)

If every organization uses SPT and puts its jobs first, then there are instances of MOSP($k : \sum C_i$) where we do not have equilibrium:

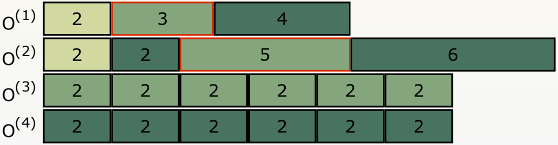


$$\sum \mathbf{C_i^{(3)}} = \cancel{56}\ \mathbf{60}$$
$$\sum \mathbf{C_i^{(4)}} = \cancel{70}\ \mathbf{65}$$

What if $O^{(3)}$ changes its strategy?

Nash equilibrium and MOSP($k : \sum C_i$)

If every organization uses SPT and puts its jobs first, then there are instances of MOSP($k : \sum C_i$) where we do not have equilibrium:

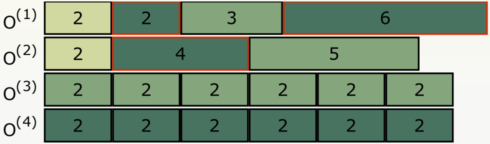


$$\sum \mathbf{C_i^{(3)}} = \cancel{60}\ \mathbf{56}$$
$$\sum \mathbf{C_i^{(4)}} = \cancel{65}\ \mathbf{70}$$

What if $O^{(4)}$ changes its strategy?

Nash equilibrium and MOSP($k : \sum C_i$)

If every organization uses SPT and puts its jobs first, then there are instances of MOSP($k : \sum C_i$) where we do not have equilibrium:



$$\sum \mathbf{C_i^{(3)}} = \cancel{56}\ \mathbf{60}$$
$$\sum \mathbf{C_i^{(4)}} = \cancel{70}\ \mathbf{65}$$

Loop!

Outline

Future work

- Study of:
  - *Price of Anarchy* (ratio between the worst objective function value of an equilibrium and the optimal)
  - *Price of Stability* (ratio between the best objective function value of one of its equilibria and the optimal outcome)
- ϵ-approximate Nash Equilibrium
- Fairness