Efficiency and Equity in the Multiple Organization Scheduling Problem

Martin Durand, Fanny Pascual

Efficiency and Equity in the Multiple Organization Scheduling Problem

Martin Durand, Fanny Pascual

Sorbonne Université, CNRS, LIP6, 4 place Jussieu, 75005 Paris, France
martin.durand@lip6.fr,fanny.pascual@lip6.fr

Keywords: multi agents scheduling, fairness, makespan minimization.

1 Introduction

The Multi Organization Scheduling Problem (MOSP) (Pascual *et al.* 2007) deals a set of n organizations $\{O_1, \ldots, O_n\}$ which each owns both a set of identical parallel machines, and a set of sequential tasks to execute. The objective is to minimize the completion time of the last task completed on the machines shared by the organizations (the makespan), under an additional constraint: no organization should increase the last completion time of its tasks in the shared system, compared to the case where it executes its own tasks on its own machines. This last constraint is called the *rationality constraint*, and ensures that all the organizations have incentive to share their machines. More formally, let us denote by C^i_{loc} is the makespan of Organization O_i if it schedules its own tasks on its owns machines - this scheduled is assumed to be given by the organization (it can minimizes the makespan of Organization O_i ,or not) - and is called the *local makespan of* O_i. Given any schedule S of all the tasks on all the machines, we denote by $C_i(\mathcal{S})$ the makespan of O_i, i.e. the maximum completion time of a task of O_i in S. Our problem is the following one:

$$\text{minimize } C_{\max}(\mathcal{S}) \text{ such that, for each } i \in \{1, \ldots, n\}, C_i(\mathcal{S}) \leq C^i_{loc}.$$

Interest of cooperation. Let us first show that sharing machines does not only allow organizations which have many tasks and few machines to decrease theirs makespans by given tasks to the other organizations, but that each organization may decrease its makespan. Let us consider the following instance, in which all organizations can benefit from cooperating.

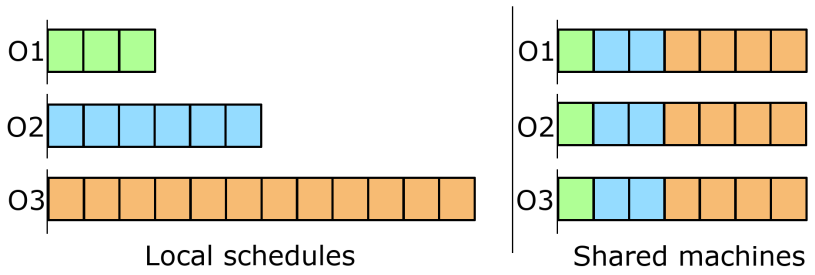

Fig. 1. An example in which each organization benefits from cooperating

There are $n = 3$ organizations, each one having only one machine. All the tasks are of length 1. Organization O_1 owns 3 tasks, O_2 owns 6 tasks, and O_3 owns 12 tasks. On figure 1 we can see on the left the local schedules (schedules in which each organization schedules its own tasks on its owns machines), and on the right a schedule in which the organizations

share their machines. In this example, sharing the machines allow each organization to decrease its makespan. This example can be extended for higher values of n. We have shown that the best improvement which can be obtained for each organization simultaneously is a factor n, and that this ratio can be obtained on some instances.

Focus of this paper and map of the paper. Besides analyzing the best possible benefit that organizations can mutually have by sharing their machines, our aim is to focus on the *efficiency* of algorithms (where the efficiency is thought in term of makespan – the date at which all the tasks have been computed), and on the *equity* of algorithms for MOSP (it is not suitable that, even if the returned schedule fulfills the rationality constraint, the machines which are free are used only for the tasks of a single organization while some tasks of the other organizations are waiting). These two aspects may be antagonist, and our aim is to see to which extent, since what we want would be a schedule with a small makespan and in which machines are shared with equity.

In Section 2, we focus on efficiency: we analyze the highest possible increase of a local makespan which may occur if we want a schedule which minimizes the (global) makespan. We then look at the problem where each organization agrees to increases its makespan (compared to its local makespan) by a factor $(1 + \epsilon)$.

In Section 3, we focus on fairness: we introduce a new problem which consists in maximizing the minimal gain (decrease of its makespan) of an organization. Before presenting these results, we start by reviewing existing work on MOSP.

State of art. The Multi Organization Scheduling Problem (Pascual *et al.* 2007) has been introduced with parallel rigid tasks (tasks that need to be executed in parallel on several machines) and has mainly been studied from an approximation viewpoint. The best approximate algorithm is a 3-approximation algorithm when the organizations schedule locally the tasks in decreasing order of their heights (the height of a task is the number of machines needed to execute the task), or a 4-approximation algorithm in the general case (Dutot *et. al.* 2011). For sequential tasks (tasks that need to be executed on one machine only), the best known algorithm is a 2-approximate algorithm (Cohen *et. al.* 2010) (in the sequel, all the papers – as well as our results – deal with sequential tasks).

Some papers also consider a relaxed version of MOSP: it is assumed that the organizations tolerate a bounded degradation on the makespan of their own tasks, and the aim is to minimize the global makespan. This problem is denoted by $(1 + \alpha)$-MOSP (Ooshita *et. al.* 2009) when it is assumed that each organization accepts to increase the maximum completion time of its tasks by a factor at most $(1+\alpha)$. A $\frac{3}{2}$-approximate algorithm for 2-MOSP has been given (Cordeiro *et. al.* 2011). The closest work in spirit to what we will do in Section 2 is a study of $(1+\alpha)$-MOSP on unrelated machines (Ooshita *et. al.* 2009, Ooshita *et. al.* 2012). In this setting, Ooshita et al. show that, when there is no cooperation ($\alpha = 0$), the makespan can be m times higher than in the optimal makespan without the rationality constraint. When $\alpha > 0$, the authors also give a $(2 + \frac{2}{\alpha})$-approximate algorithm for $(1 + \alpha)$-MOSP.

2 Efficiency vs. increase of the local makespans

In this section, we study how the aim of minimizing the makespan is in opposition with the rationality constraint.

2.1 Necessary trade-off between the (global) makespan and the increase of local makespans.

We have shown that in an optimal schedule for the makespan minimization, an organization may increase its makespan up to a factor m, where m is the number of machines (due to lack of space the proof is omitted). This value, that could be called "the price of efficiency", is high. We will now assume that organizations may accept to increase their makespans in order to get an efficient schedule, but only if this does not increase to much their makespans. We will now assume that each organization agrees to increase a little bit its makespan: it will accept a schedule in which its makespan is increased by a factor at most $(1+\epsilon)$ compared to its local makespan.

Let $\epsilon \geq 0$. We assume that each organization O_i agrees to have a makespan at most equal to $(1+\epsilon)C^i_{loc}$. If $\epsilon = 0$, this is the MOSP. Otherwise, each organization agrees to increase a little bit its makespan (the higher α is, the higher an organization agrees to increase its makespan). We call $(1+\epsilon)$-MOSP, the problem where we wish to minimize the makespan with these relaxed constraints:

$$\text{minimize } C_{\max}(\mathcal{S}) \text{ such that, for each } i \in \{1,\ldots,N\}, C_i(\mathcal{S}) \leq (1+\epsilon)C^i_{loc}.$$

Thanks to a specific instance, we give a lower bound on the approximation ratio of any algorithm for $(1+\epsilon)$-MOSP with respect to the optimal makespan when there is no rationality constraints: this shows what we loose, in term of makespan, due to the relaxed rationality constraint. When $\epsilon = 0$, we obtain the following proposition.

Proposition 1 *There is no algorithm which returns schedules which fulfill the rationality constraint, and which is less than* 2*-approximate with respect to the global makespan.*

This bound improves the previous one, $\frac{3}{2}$, which had been given (Pascual *et al.* 2007) first for two organizations and then (Cohen *et. al.* 2010) for more than two organizations. Furthermore, in (Cohen *et al.* 2011) the authors show that no approximation algorithm for MOSP has a ratio asymptotically better than 2 w.r.t. the global makespan (when m tends towards the infinity) when we add the constraint that on the returned schedule, each machine schedules the tasks of its organization (if any) before the tasks of other organizations. This constraint is thus not necessary to obtain the asymptotic ratio of 2.

2.2 A PTAS for the makespan minimization with a bounded increase on the local makespan

We adapt the polynomial approximation scheme (PTAS) presented (Hall and Shmoys 1989) for a scheduling problem (makespan minimization with delivery times), to get a PTAS with resource augmentation for our problem. More precisely: given a fixed $\epsilon > 0$, and a fixed number of organizations n, we will get a polynomial time algorithm which returns a schedule with a makespan at most $(1+\epsilon)$ times the optimal makespan, and in which the makespan of each organization is at most $(1+\epsilon)$ times its local makespan. The rationality constraint may thus be violated, but the increase of the makespans of the organizations is bounded, and may be acceptable if ϵ is small.

In the previous sections, we have assumed either that the rationality constraint should be fulfilled (but we then had as only objective function to minimize the makespan, and the gains for the organizations – the decrease of their makespans – in the returned schedule could be very different), or we have even assumed than we can relax (in a bounded way) the rationality constraint to get a schedule with an even smaller makespan. In the following section, we focus on fairness issues: we will keep the rationality constraint, and our focus

will not be to decrease the makespan, but to get schedule in which *all* the organizations decrease their makespans by a factor as large as possible.

3 Maximizing the Minimal Decrease of the Local Makespans

Given a schedule $\mathcal{S}$, the gain $g_i(\mathcal{S})$ of Organization O_i represents how much Organization O_i has decreased its makespan in $\mathcal{S}$ in comparison to its local schedule:

$$g_i(\mathcal{S}) = \frac{C^i_{loc}}{C_i(\mathcal{S})}.$$

The Maximal Minimal Gain problem, denoted as MaxMinGain, takes the same input as MOSP. It builds a schedule of all the tasks of all the organizations on the m machines of the organizations, in order to maximize the minimum gain among the organizations. The returned schedule is thus $\mathcal{S} = \arg\max_{\mathcal{S}} \min_{i\in\{1,\dots,N\}} g_i(\mathcal{S})$

Problem MaxMinGain can be solved in polynomial time when all the tasks have the same length. Moreover, in this case, it is possible to find a schedule $\mathcal{S}$ which is optimal for MaxMinGain and which is optimal for problem $(P||C_{\max})$: the global makespan is minimized while the minimal gain of an organization is maximized. The algorithm is very simple: it consists in scheduling the tasks greedily, by increasing local makespans.

When tasks can have different lengths, MaxMinGain is strongly NP-hard and hard to approximate. We also show that there is no algorithm which is optimal for MaxMinGain and which has an approximation smaller than 2 for MOSP. Naturally, this implies that no algorithm can be optimal for MaxMinGain and have an approximation ratio smaller than 2 for $(P||C_{\max})$.

As seen in the previous section, a list scheduling by increasing local makespan is optimal for MaxMinGain when tasks all have the same length. However, in the general case, such a schedule can break the rationality constraint. We complete our results with a heuristic that aims at returning a schedule as close as possible from a list schedule by increasing local makespan but respecting the rationality constraint. On tested instances, this heuristic returns a schedule with an average makespan below 1.07 times the optimal makespan and an average minimum gain above 0.92 times the optimal one.

References

Cohen J., D. Cordeiro, D. Trystram, F. Wagner, 2010, "Analysis of Multi-Organization Scheduling Algorithms", *Euro-Par*, Vol. 2, pp. 367-379.

Cohen J., D Cordeiro, D. Trystram, F. Wagner, 2011, "Multi-organization scheduling approximation algorithms", *Concurrency and Computation: Practice and Experience*, Vol. 23, pp. 2220-2234.

Cordeiro D., P. Dutot, G. Mounié, D. Trystram, 2011, "Tight Analysis of Relaxed Multi-organization Scheduling Algorithms", *IPDPS*, pp. 1177-1186.

Dutot P., F. Pascual, K. Rzadca, D. Trystram, 2011, "Approximation Algorithms for the Multi-organization Scheduling Problem", *IEEE Transactions on Parallel and Distributed Systems*, Vol. 22, pp. 1888-1895.

Hall L. A., Shmoys D. B., 1989, "Approximation Schemes for Constrained Scheduling Problems", *FOCS*, pp. 134-139.

Ooshita F., T. Izumi, T. Izumi, 2009, "A Generalized Multi-Organization Scheduling on Unrelated Parallel Machines", *PDCAT*, pp. 26-33.

Ooshita F., T. Izumi, T. Izumi, 2012, "The Price of Multi-Organization Constraint in Unrelated Parallel Machine Scheduling", *Parallel Process Letters*, Vol. 22.

Pascual F., K. Rzadca, D. Trystram, 2011, "Cooperation in Multi-organization Scheduling", *Euro-Par*, pp. 224-233.