The multi-agent project scheduling problem : the fair share of stress

Cyril Briand[1,2] **and Thomas Lehaux**[1,2]

[1]CNRS ; LAAS ; 7 avenue du colonel Roche, F-31077 Toulouse, France
[2]Université de Toulouse ; UPS, INSA, INP, ISAE ; LAAS ; F-31077 Toulouse, France
`{briand, tlehaux}@laas.fr`

Keywords: project scheduling, robustness, multi-agent scheduling, fairness.

1 Introduction

The Multi-Agent Project Scheduling Problem (MAPSP) is a project scheduling problem where a set $\mathcal{V}$ of n activities is distributed among a set $\mathcal{A}$ of m agents ($m \leq n$). Classically, activities are linked together by a set $\mathcal{P}$ of precedence constraints $(i, j) \in \mathcal{P}$ (i precedes j). A project completion time $C_{\max}$ is set. The duration of activity i is uncertain and modeled by an interval $[\underline{p}_i, \overline{p}_i]$. The value $\underline{p}_i$ can be interpreted as the best duration of i, when the agent is able to perform in the most favorable situation. Conversely, $\overline{p}_i$ gives the duration of i in the worst case scenario, when the agent dwells in the most degraded situation. Ressource constraints are not considered.

Each agent A_u is supposed to have its own decision autonomy (its own local objective functions). Its knowledge is assumed restricted. It only knows the subset of precedence constraints and the processing times which are linked to the activities it manages. Moreover, when Activity j, managed by A_u, has a precedence relation with Activity i (i.e., $(i, j) \in \mathcal{P}$), managed by A_v ($u \neq v$), i is called a *border activity* and it is mandatory that A_v communicates to A_u a completion time interval $[\underline{C}_i, \overline{C}_i]$ for i. The agent engages to complete i within this interval, i.e., $C_i \in [\underline{C}_i, \overline{C}_i]$. Eventually, this completion time interval can be reduced to a single date.

The MAPSP consists in assigning to each activity a completion time interval so that, on the one hand, the worst completion time of any activity remains lower or equal to the maximum project duration (i.e., $\overline{C}_i \leq C_{\max}$, $\forall i \in \mathcal{V}$) and, on the other hand, the agents are satisfied. This notion of agent satisfaction is detailed below.

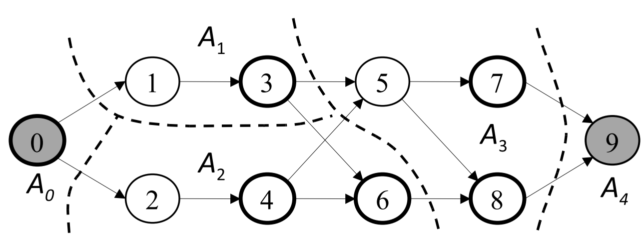

Fig. 1. A MAPSP example

A MAPSP example with $n = 8$ activities is depicted in Fig. 1. It is represented by an activity-on-node graph, where each vertex corresponds to an activity and each arc corresponds to a precedence relation between a pair of activities belonging to $\mathcal{V} = \{0, \ldots, 9\}$.

The dotted lines indicate the assignment of activities to the agents. Typically, the activities 0 and 9 are dummies and correspond to the beginning of the project and its end, respectively. They are assigned to fictitious agents A_0 and A_4 that can be seen as the project launcher and the project client, respectively. The non-dummy activities are distributed among a set of three agents, $\mathcal{A} = \{A_1, A_2, A_3\}$. The sets $\mathcal{V}_u$ of activities allocated to Agent A_u are $\mathcal{V}_1 = \{1, 3\}$, $\mathcal{V}_2 = \{2, 4, 6\}$ and $\mathcal{V}_3 = \{5, 7, 8\}$. The vertices corresponding to the border activities are in bold in the graph.

Given a valid assignment of completion time intervals to the border activities, each Agent A_u, $\forall u = 1, \ldots, m$, can determine for any activity $i \in \mathcal{V}_u$ a maximum authorized processing time, denoted $\tilde{p}_i$, with $\tilde{p}_i \in [\underline{p}_i, \overline{p}_i]$. This processing time corresponds to the maximum time the agent allocates itself in order to carry out the activity i. Each $\tilde{p}_i$ is set so that local precedence constraints are satisfied and, in the worst case (i.e., $p_i = \tilde{p}_i \ \forall i \in \mathcal{V}_u$), the worst completion times which were announced by u to its downstream agents remains satisfied, i.e :

$$C_j \geq \overline{C}_i - \Delta_i + \tilde{p}_j \ , \forall (i,j) \in \mathcal{P} \text{ such that } i \notin \mathcal{V}_u \text{ and } j \in \mathcal{V}_u \tag{1}$$

$$C_j \geq C_i + \tilde{p}_j \ , \forall (i,j) \in \mathcal{P} \text{ such that } i \in \mathcal{V}_u \text{ and } j \in \mathcal{V}_u \tag{2}$$

$$C_i \leq \overline{C}_i \ , \forall i \in \mathcal{P} \text{ such that } i \in \mathcal{V}_u \tag{3}$$

In Eq. (1), the variable $\Delta_i \in [0, \overline{C}_i - \underline{C}_i]$ is introduced. It models the fact that A_u is authorized to anticipate by Δ_i the completion of the border activity i that is managed by the upstream agent A_v. As a consequence, it exists an inconsistency risk since, when the realized completion time of i is greater than $\overline{C}_i - \Delta_i$, A_u organization can become temporally inconsistent. If A_u is optimistic (i.e., $\Delta_i = \overline{C}_i - \underline{C}_i$), it trusts A_v to be able to complete i at the earliest, but the inconsistency risk is high. At the opposite, in the pessimistic point of view (i.e., $\Delta_i = 0$), A_u prefers to protect itself against the worst behavior of A_v, and the inconsistency risk is void. In this paper, the inconsistency risk R_u taken by A_u is expressed by Formula (4).

$$R_u = \max_{\{(i,j) \in \mathcal{P} | i \notin \mathcal{V}_u \wedge j \in \mathcal{V}_u\}} \Delta_i \tag{4}$$

After agent A_u has fixed the variables Δ_i, it is able to determine the values $\tilde{p}_j$ for its own activities, provided the chosen values respect Equations (1-3). Considering Activity $j \in \mathcal{V}_u$, the stress that A_u gets for achieving j can be measured by the following ratio :

$$\gamma_j = \overline{p}_j - \tilde{p}_j$$

When γ_j is close to 0, the prescribed worst processing time is close to the maximal one and the agent is not stressed, while, the higher it gets, the more the agents are stressed. In this paper, the stress of the Agent A_u, denoted S_u, refers to the maximum value of γ_j, i.e. $S_u = \max_{j \in \mathcal{V}_u} \gamma_j$.

Under such hypothesis, the MAPSP is a multi-objective optimization problem where each agent A_u wants to minimize its inconsistency risk and its stress, these objectives being conflicting. In this paper, we also consider two global objective functions (GOF): the minimization of $\max_{(u,v) \in \mathcal{A}^2}(S_v - S_u)$ and the minimization of $\max_{(u,v) \in \mathcal{A}^2}(R_v - R_u)$. Those GOFs favor the fairness of the agents against their selfishness since, when the stress and the inconsistency risk are equally shared between agents, both GOFs are optimal (they equal 0).

2 Discussion and state-of-the-art

Any real life project that involves several decision-independent partners, each in charge of the execution of a different part of the project, can be viewed as a MAPSP. In this kind of project, due to the privacy nature of the data handled by the partners, only a restricted set of information is shared among the actors. One natural mode of collaboration between partners precisely consists in communicating proposals of intervals for the completion time of border activities, these values being negotiated until a trade-off satisfying both partners is found. Once a globally satisfactory assignment of the completion time intervals is found, the agents become free to organize themselves as they intend, provided they succeed in completing the activities within the contracted intervals. During this phase, any agent can be more or less pessimistic depending on whether it assumes that the upstream agents will be more or less efficient. In case of inconsistency, completion intervals can be negotiated back.

Looking at the scheduling literature, we do not find many works dealing with the MAPSP. Several papers propose to solve resource-constrained multi-project scheduling problems in a decentralized way, using a software architecture inspired from Multi-Agent Systems (MAS) (Confessore *et. al.* 2007, Homberger 2007), but here, we distinguish between a distributed scheduling method based on MAS and a MAPSP, as defined above, for which we do not care about the nature of the solving method.

Nonetheless, scheduling with agents is not a new problem since several authors already took an interest in this topic, considering the presence of resource constraints and trying to solve the problem in a centralized way. In (Agnetis *et. al.* 2000, Agnetis *et. al.* 2004), the authors consider the job shop environment where two agents compete for performing their jobs on a common set of resources, trying to optimize their respective objective (the difference between agent objective values being ϵ-constrained). Several papers (see (Agnetis *et. al.* 2009, Cheng *et. al.* 2006, Cheng *et. al.* 2008)) focus on the single machine scheduling problem with two or more agents and provide complexity results and exact algorithms for particular cases. Several authors also focus on grid scheduling problems where agents, corresponding to cluster of processors, are able to negotiate their load all together (see (Pascual *et. al.* 2009) for a recent approach). Like us, some authors additionally suggest to introduce a GOF, also called social objective, and define important concepts such as the *price of anarchy* (Koutsoupias and Papadimitriou 1999) (ratio between the worst Nash equilibrium and the optimum GOF value) or the *price of stability* (Angel *et. al.* 2006) (ratio between the best Nash equilibrium and the optimum GOF value).

3 The one-per-one case

In this section we assume that any agent manages one and only one activity, i.e., Activity i is managed by A_i (see Fig. 2). Moreover, we also suppose that agents are fair : inconsistency risks are identical for any agent (i.e., $R_i = \Delta, \forall A_i \in \mathcal{A}$) and stress is equally shared (i.e., $S_j = \gamma$, $\forall i \in \mathcal{A}$). Under these assumptions, the GOFs are optimal since $\max_{(u,v)\in\mathcal{A}^2}(S_v - S_u) = \max_{(u,v)\in\mathcal{A}^2}(R_v - R_u) = 0$. Therefore, the MAPSP turns into a 2-objective optimization problem, for which we take an interest in finding the Pareto solutions minimizing both Δ and γ criteria.

For this case, the Pareto solutions can be found in polynomial time. To solve this problem, we use the activity-on-node graph as seen in Fig. 2. The value of the arc between an activity i and an activity j is : $max(\underline{p_i}, \overline{pi} - \gamma - \Delta)$. Every activities having the same γ and Δ, the first idea is to set $(\Delta + \gamma)$ to a certain value and to compute the longest path by using the Ford-Bellman algorithm, which runs in $\mathrm{O}(n \log n)$ on graphs without circuit. We compute the longuest paths for every possible value of $(\Delta + \gamma)$ verifying $\Delta + \gamma \leq \overline{p_i} - \underline{p_i}$. We

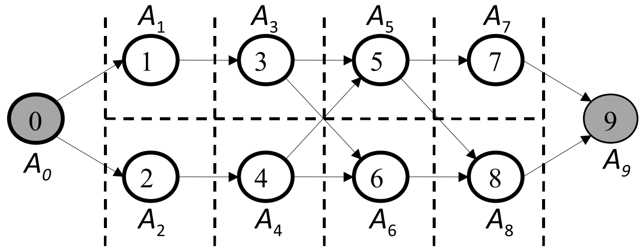

Fig. 2. A one-per-one MAPSP example

consider the $\overline{p_i} - \underline{p_i}$ values in the decreasing order and compute the longest path until its value is superior to C_{max}. This will gives us the minimum allowed value of L. Every integer couple of values (Δ, γ) such that $\Delta + \gamma = L$ are Pareto optimum. The determination of L requiring at most n longest path computation (i.e, there are n $(\overline{p_i} - \overline{p_i})$ values to consider at the most), the Pareto solutions can be found in $O(n^2 \log n)$.

At the conference, we will discuss the complexity of the general MAPSP and present some other particular cases which can be solved in polynomial time.

Acknowledgements

This work was supported by the ANR project no. 08-BLAN-0331-02 named “ROBO-COOP”.

References

Agnetis A., Mirchandani P. B., Pacciarelli D., Pacifici A., 2000, “Nondominated schedules for a job-shop with two competing agents", *Computational and Mathematical Organization Theory*, 6(2), pp. 191-217.

Agnetis A., Mirchandani P. B., Pacciarelli D., Pacifici A., 2004, “Scheduling problems with two competing agents", *Operations Research*, 52(2), pp. 229-242.

Agnetis A., de Pascale G., Pacciarelli D., 2009, “A Lagrangian approach to single-machine scheduling problems with two competing agents", *Journal of Scheduling*, 12(4), pp. 401-415.

Angel E, Bampis E, Pascual F, 2006, “ The price of approximate stability for a scheduling game problem", *Proceedings of Euro-Par (Lecture Notes in Computer Science)*, Springer: Berlin.

Cheng T. C. E., Ng C. T., Yuan J. J., 2006, “Multi-agent scheduling on a single machine to minimize total weighted number of tardy jobs", *Theoretical Computer Science*, 362, pp. 273-281.

Cheng T. C. E., Ng C. T., Yuan J. J., 2006, “ Multi-agent scheduling on a single machine with max-form criteria", *European Journal of Operational Research*, 188, pp. 603-609.

Confessore G., Giordani S., Rismondo S., 2007, “ A Market-based Multi-Agent System Model for Decentralized Multi-Project Scheduling", *Annals of Operations Research*, 150(1), pp. 115-135.

Homberger J., 2007, “A Multi-agent System for the Decentralized Resource-constrained Multi-project Scheduling Problem", *International Transactions in Operational Research*, 14(6), pp. 565-589.

Koutsoupias E., Papadimitriou C., 1999, “Worst-case equilibria", *in Proceedings of the 16th Annual Symposium on Theoretical Aspects of Computer Science*, pp. 404-413.

Pascual F., Rzadca K., Trystram D., 2009, “"Cooperation in Multi-Organization Scheduling", *Concurrency and Computation: Practice and Experience*, 21(7), pp. 905-921.