Le problème d'ordonnancement de projet multi-agent : un partage équitable du stress

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

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

Mots-Clés : *ordonnancement de projet, ordonnancement multi-agent, equité, robustesse.*

1 Introduction

Beaucoup de projets font intervenir plusieurs acteurs indépendants, ayant leur propre autonomie decisionnelle, chacun étant responsable d'une partie du projet. Pour des raisons de confidentialité, ceux-ci n'ont généralement qu'une connaissance minimale des données gérées par les autres. Une forme de collaboration classique est alors l'échange de dates de fin d'activités, afin que chacun puisse se coordonner. Ces dates sont négociées jusqu'à ce que chaque acteur soit satisfait. Le problème d'ordonnancement multi-agent n'est pas nouveau et plusieurs auteurs ont déjà montré un intérêt pour ce sujet (voir les références [1, 2, 3]). Nous proposons dans cet article un modèle pour ce problème où l'on cherche à partager équitablement le stress et les risques entre les agents sans prise en compte des ressources (celles-ci n'étant pas supposées partagées entre les agents du système).

2 Formalisation du problème

Un *problème d'ordonnancement de projet multi-agent* (noté MAPSP) est défini par un jeu $\mathcal{V}$ de n activités, distribuées parmi un ensemble $\mathcal{A}$ de m agents ($m \leq n$). Typiquement, les activités sont liées par un ensemble $\mathcal{P}$ de contraintes de précédence : $(i, j) \in \mathcal{P}$ signifie que l'activité i précède l'activité j. Le projet possède une durée maximale C_{max}. Le temps d'exécution d'une activité est incertain et est modélisé par un intervalle temporel $[\underline{p}_i, \overline{p}_i]$. $\underline{p}_i$ correspond à la durée minimale pour réaliser l'activité i : celle réalisée lorsque l'agent travaille dans les meilleures conditions possibles. A contrario, la valeur $\overline{p}_i$ est la durée dans le pire des cas.

Il est supposé que chaque agent A_u possède une fonction objectif locale propre. De plus, chaque agent a une connaissance limitée du système : il ne connait que les contraintes de précédence et les intervalles de durée liés aux activités qu'il gère. Nous définissons comme *activité frontière* une activité i, gérée par un agent A_u, ayant une contrainte de précédence avec une activité j, gérée par un agent A_v (avec $u \neq v$). Nous supposons obligatoire que, pour chaque activité frontière i, les agents concernés se communiquent entre eux des fenêtres de temps $[\underline{C}_i, \overline{C}_i]$. Ces fenêtres correspondent à un engagement de la part des agents gérant les activités frontières d'achever ces taches dans l'intervalle de temps communiqué, i.e., $C_i \in [\underline{C}_i, \overline{C}_i]$ $\forall i$.

Pour une affectation valide des intervalles $[\underline{C}_i, \overline{C}_i]$ des activités frontières, chaque agent A_u peut déterminer pour chaque activité $j \in \mathcal{V}_u$ une durée maximale autorisée notée $\tilde{p}_j$, telle que $\tilde{p}_j \in [\underline{p}_j, \overline{p}_j]$. Cette durée est une durée maximale que l'agent s'alloue pour effectuer la tache j. Nous partons du principe que, gérant ses propres ressources, l'agent à un certain controle sur le temps nécessaire pour réaliser ses tâches. Cette durée est une durée maximale, la tâche peut être effectuée en un temps plus court (sans jamais prendre moins de temps que $\underline{p}_j$). Le stress de l'agent est alors défini comme $S_u = \max_{j \in \mathcal{V}_u}(\overline{p}_j - \tilde{p}_j)$. Il est d'autant plus grand que $\tilde{p}_j$ est inférieur à $\overline{p}_j$.

Nous introduisons également la variable $\Delta_i \in [0, \overline{C}_i - \underline{C}_i]$. Elle modélise le fait que l'agent A_u peut anticiper d'une valeur Δ_i la date de fin de l'activité i gérée par l'agent A_v (i étant une activité frontière entre u et v). De ce fait, si A_v finit i après la date $\overline{C}_i - \Delta_i$ alors le planning de A_u peut devenir temporairement inconsistant. Plus A_u est optimiste ($\Delta_i = \overline{C}_i - \underline{C}_i$), plus le risque d'incohérence est élevé. Au contraire, si A_u est pessimiste ($\Delta_i = 0$), alors le risque d'incohérence est nul et la robustesse maximale (au sens où l'organisation de A_u devient indépendante des performances des autres agents). Ce risque, noté $\mathcal{R}_u$ peut être exprimé par la formule suivante : $R_u = \max_{i \in \mathcal{P} | i \notin \mathcal{V}_u} \Delta_i$. Au final, on peut exprimer ce problème comme un programme linéaire en nombres entiers avec les contraintes suivantes :

$$
\begin{array}{ll}
\min(R_u, S_u, \max(S_v - S_u), \max(R_v - R_u)) & \\
\text{s.c} & \\
R_u \geq \Delta_i \ , \forall i \in \mathcal{P} \text{ avec } i \notin \mathcal{V}_u & (1) \\
S_u \geq \overline{p}_j - \tilde{p}_j \ , \forall j \in \mathcal{P} \text{ avec } j \in \mathcal{V}_u & (2) \\
C_j \geq \overline{C}_i - \Delta_i + \tilde{p}_j \ , \forall (i,j) \in \mathcal{P} \text{ avec } i \notin \mathcal{V}_u \text{ et } j \in \mathcal{V}_u & (3) \\
C_j \geq C_i + \tilde{p}_j \ , \forall (i,j) \in \mathcal{P} \text{ avec } i \in \mathcal{V}_u \text{ et } j \in \mathcal{V}_u & (4) \\
C_i \leq \overline{C}_i \ , \forall i \in \mathcal{P} \text{ avec } i \in \mathcal{V}_u & (5) \\
\Delta_i \leq \overline{C}_i - \underline{C}_i \ , \forall i \in \mathcal{P} \text{ avec } i \notin \mathcal{V}_u & (6) \\
S_u \geq 0, R_u \geq 0, \Delta_i \geq 0, \underline{p}_j \leq \tilde{p}_j \leq \overline{p}_j & (7)
\end{array}
$$

Le MAPSP est un problème d'optimisation multi-objectif car chaque agent veut minimiser son risque d'incohérence tout en minimisant son stress. Or, ces deux objectifs sont antagonistes. En plus de ces objectifs locaux, nous considérons deux fonctions objectives globales ayant pour but d'harmoniser stress et risque d'incohérence entre les agents, et ainsi éviter les comportements égoistes : $\max_{(u,v) \in \mathcal{A}^2}(S_v - S_u)$ et $\max_{(u,v) \in \mathcal{A}^2}(R_v - R_u)$. Durant la conférence, nous discuterons de la complexité de ce problème et présenterons quelques cas particuliers polynomiaux.

Remerciements

Ce travail est supporté par le projet ANR no. 08-BLAN-0331-02 appellé “ROBOCOOP”.

Références

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

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

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