

# Complexité des problèmes d’ordonnancement parallèle avec tâches unitaires et contraintes de précédence

Tianyu Wang<sup>1</sup>, Odile Bellenguez-Morineau<sup>1</sup>

1, IMT Atlantique, LS2N, UMR CNRS 6004, 4 rue Alfred Kastler, B.P. 20722 F-44307 Nantes  
Cedex 3. France

{tianyu.wang, odile.morineau}@imt-atlantique.fr

**Mots-clés :** *ordonnancement, contraintes de précédence, complexité*

L’état de l’art sur la complexité d’une famille de problèmes d’ordonnancement proposé par[5] met en évidence des cas encore ouverts. Nous proposons ici une mise à jour intégrant de nouveaux résultats.

## 1 Famille de problèmes

Les problèmes considérés contiennent  $n$  tâches et  $m$  machines parallèles identiques. Les tâches ont une même durée, noté  $p_j = 1$ . Dans certains problèmes, la préemption est autorisée.

De plus, il existe des contraintes de précédence entre les tâches : la tâche  $j_2$  ne peut pas commencer avant que la tâche  $j_1$  soit terminée si  $j_1 \prec j_2$ . Ces contraintes de précédence peuvent se représenter par un ordre partiel ou un graphe acyclique. Nous considérons les cas particuliers où ces graphes admettent une forme spécifique, telle que *in-tree*(noté i.t.), *out-tree*(o.t.), *opposing-forests*(o.f.), ou *level-order*(l.o.). Enfin, on considère différentes fonctions objectif, telles que : le *makespan*  $C_{\max}$ , la somme des délais de réalisation  $\sum C_j$ , le retard algébrique maximum  $L_{\max}$ , et le retard absolu maximum  $\sum T_j$ .

## 2 Complexité et algorithmes

Prot et Bellenguez-Morineau[5] ont proposé un état de l’art et ont notamment montré que la complexité du problème d’ordonnancement varie selon la classe du graphe de précédence, la fonction objectif et la préemption, et mettent en évidence des cas toujours ouverts. Notre objectif est ici de proposer une synthèse de nouveaux résultats sur ces cas.

L’un des problèmes *minimum open* dans[5] est noté  $P|p_j = 1, i.t.|\sum C_j$ , selon la notation à 3-champs[4]. Nous avons cependant pu adapter les travaux de[3], pour proposer une transformation de 3-Partition vers ce problème d’ordonnancement, en créant des in-trees avec une chaîne très longue, tels que les tâches initiales correspondant à chaque partition doivent être exécutées dans chaque unité de temps pour que  $\sum C_j$  soit bornée. Cela nous a permis de proposer une preuve établissant la complexité de ce problème.

Les problèmes mettant en jeu un graphe de type level-order, tel que  $P|p_j = 1, l.o.|\sum C_j$  et  $P|p_j = 1, l.o.|C_{\max}$  étaient également ouverts. Cependant, en utilisant une transformation similaire de[7], on garde la structure des graphes et on inverse les précédence pour obtenir des out-trees et un level-order de tâches fictives à partir d’une instance de 3-Partition. Les tâches fictives forment un profil de machines disponibles à chaque moment, qui force les chaînes correspondant à une partition à être exécutées dans une période de temps spécifique. Cela permet de montrer l’existence d’un ordonnancement plein, i.e. sans *idle time*, obtenu par une structure issue de la partition lorsqu’elle existe. On montre ainsi que ces problèmes sont  $\mathcal{NP}$ -complet.

Par la suite, nous montrerons comment ces résultats peuvent être étendus au cas préemptifs,  $P|p_j = p, l.o., pmtn|C_{\max}$  et  $P|p_j = p, i.t., pmtn|\sum C_j$ . En changeant les tâches fictives par un opposing-forest ou out-tree, on montre également que les problèmes  $P|p_j = 1, l.o., pmtn|\sum C_j$ ,  $P|p_j = 1, l.o., pmtn|C_{\max}$ ,  $P|p_j = p, o.t.|L_{\max}$  et  $P|p_j = p, o.t.|\sum T_j$  sont  $\mathcal{NP}$ -complet. Enfin, nous avons prouvé que  $P|p_j = 1, i.t., pmtn|\sum C_j$  est aussi  $\mathcal{NP}$ -complet. Les preuves complètes de ces résultats sont en cours de rédaction.

Les articles présentant la complexité du  $P|p_j = 1, i.t.|\sum C_j$ [7] et  $P|p_j = 1, l.o.|C_{\max}$ [6] sont soumis.

### 3 Conclusion et perspectives

Dans cette mise à jour de l'état de l'art, nous apportons des conclusions sur la complexité de différents cas jusqu'ici ouverts, renforçant ainsi l'idée qu'il est nécessaire de bien caractériser le type de contraintes de précédence afin de mettre en œuvre la meilleure méthode de résolution. Ce travail a pour suite une recherche d'amélioration de méthodes existantes, dont le développement est en cours sur les problèmes  $P|p_j = p, l.o.|C_{\max}$  et  $P|p_j = p, i.t.|\sum C_j$ , en vue d'améliorer les algorithmes de  $O((nm)^m)$ [2] et  $O(n^m)$ [1] à  $O((2h)^m(qn)^2)$  et  $O((hm)^m)$ , où  $h$  est la longueur de la plus longue chaîne dans le graphe. Ces travaux sont en cours de rédaction.

### Références

- [1] Philippe Baptiste, Peter Brucker, Sigrid Knust, and Vadim G Timkovsky. Ten notes on equal-processing-time scheduling. *Quarterly Journal of the Belgian, French and Italian Operations Research Societies*, 2(2) :111–127, 2004.
- [2] Danny Dolev and Manfred K Warmuth. Profile scheduling of opposing forests and level orders. *SIAM Journal on Algebraic Discrete Methods*, 6(4) :665–687, 1985.
- [3] MR Garey, DS Johnson, RE Tarjan, and M Yannakakis. Scheduling opposing forests. *SIAM Journal on Algebraic Discrete Methods*, 4(1) :72–93, 1983.
- [4] Ronald L Graham, Eugene L Lawler, Jan Karel Lenstra, and AHG Rinnooy Kan. Optimization and approximation in deterministic sequencing and scheduling : a survey. *Annals of discrete mathematics*, 5 :287–326, 1979.
- [5] Damien Prot and Odile Bellenguez-Morineau. A survey on how the structure of precedence constraints may change the complexity class of scheduling problems. *Journal of Scheduling*, pages 1–14, 2017.
- [6] Tianyu Wang and Odile Bellenguez-Morineau. Complexity of parallel scheduling problem unit-processing-time jobs under level-order precedence constraints. Soumis à Journal of Scheduling, 2017.
- [7] Tianyu Wang and Odile Bellenguez-Morineau. Complexity of parallel scheduling problem with equal processing time and particular precedence constraints while minimizing the mean flow time. Soumis à Journal of Scheduling, 2017.