

Filtrage basé sur les coûts pour la minimisation du délai moyen.

Arnaud Malapert¹ et Margaux Nattaf²

¹ Université Côte d'Azur, CNRS, I3S, France arnaud.malapert@unice.fr

² Univ. Grenoble Alpes, CNRS, Grenoble INP[†] G-SCOP, 38000 Grenoble, France
{margaux.nattaf}@grenoble-inp.fr

Mots-clés : *Ordonnancement à machine parallèles, Familles de tâches, Flow time, Disqualification des machines, Filtrage basé sur les coûts.*

Introduction Ce travail porte sur un problème d'ordonnancement de familles de jobs sur des machines parallèles avec des contraintes liées à la qualification des machines. Dans ce problème, chaque job appartient à une famille et une famille ne peut être exécutée que par un sous-ensemble de machines dites qualifiées. De plus, les machines peuvent perdre leurs qualifications au cours de l'ordonnancement. En effet, si aucun job d'une famille n'est ordonnancé sur une machine pendant un laps de temps donné, la machine perd sa qualification pour cette famille. Le but est de minimiser la somme des temps d'exécution des travaux, c'est-à-dire le délai moyen, et de maximiser le nombre de qualifications restantes à la fin de l'ordonnancement. Ce problème est appelé PTC (scheduling Problem with Time Constraints).

Plusieurs modèles de Programation Linéaire en Nombres Entiers (PLNE) et de Programmation Par Contraintes (PPC) ont été proposés dans [4, 3, 1]. Les résultats expérimentaux ont permis de dégager les deux meilleurs modèles, un PLNE et un modèle PPC, et ont montré que leur performances étaient orthogonales, i.e. difficilement comparables. Un axe d'amélioration pour le modèle PPC est une meilleure prise en compte du délai moyen de l'affectation partielle pendant la résolution.

L'objectif de ces travaux est d'améliorer les performances du modèle PPC pour l'objectif de délai moyen. Pour ce faire, une version relâchée de PTC, dans laquelle les contraintes de qualification sont supprimées, est prise en compte. Pour cette relaxation, les résultats de [2] sont adaptés pour définir un algorithme minimisant le délai moyen sur une machine où les contraintes de qualifications sont relâchées en temps polynomial. Cet algorithme est ensuite utilisé pour définir plusieurs algorithmes de filtrage pour PTC.

Relaxation et séquençement Une façon de relâcher le problème est de supprimer les contraintes liées à la qualification des machines. Le problème revient alors à un problème d'ordonnancement à machines parallèles avec familles de jobs et temps de transition indépendant de la séquence. Le seul objectif est donc la minimisation du délai moyen.

Pour ce problème, nous adaptons les résultats de [2] qui définissent des propriétés et règles de dominance pour un problème d'ordonnancement à une machine avec famille de jobs où les jobs d'une même famille n'ont pas forcément la même durée. Ces résultats sont adaptés à PTC et dans notre cas permettent de définir un algorithme séquençant optimalement les jobs sur les machines. Cet algorithme vise à être utilisé sur des affectations partielles jobs-machines afin de calculer une borne inférieure sur le délai moyen.

Nous montrons que dans une séquence optimale les jobs d'une même famille sont ordonnancés en blocs les uns à la suite des autres. L'enjeu est donc de définir l'ordre dans lequel ces blocs vont être ordonnancés. Pour cela, les blocs sont triés par durée moyenne et nous pouvons

[†]Institute of Engineering Univ. Grenoble Alpes

montrer que, dans une solution optimale, tous les blocs excepté le premier sont ordonnancés suivant cet ordre. Pour trouver la solution optimale, il suffit donc d'ordonnancer les blocs par durée moyenne et d'essayer de déplacer chaque bloc en première place. La solution obtenant le meilleur délai moyen est la solution optimale.

Modèle PPC et règles de filtrage Le modèle utilisé pour modéliser le problème d'ordonnancement à machine parallèles avec familles de jobs est un modèle classique en PPC. Par souci de place, il ne sera pas détaillé ici. Cependant, nous présentons brièvement les variables dont les domaines seront filtrés par nos algorithmes : $flowtime_m$ est une variable permettant de modéliser le flow time par machine ; $nbJobs_{f,m}$ modélisent le nombre de jobs de la famille f ordonnancés sur une machine m ; et $nbJobs_m$ représente le nombre de jobs sur la machine m .

Trois règles de filtrage peuvent être définies pour PTC. Ces règles se basent sur une affectation partielle des jobs aux machines. La première règle vise à augmenter la borne supérieure sur $flowtime_m$ en calculant directement une solution réalisable avec les jobs déjà affectés aux machines. Cette règle peut être renforcée de la manière suivante. La borne supérieure sur $nbJobs_m$ nous donne le nombre de jobs maximum pouvant être ordonnancés sur m . Nous complétons donc l'ensemble des jobs déjà affectés à m avec les jobs contribuant le moins à l'objectif de délai moyen jusqu'à obtenir un ensemble ayant une taille égale à la borne supérieure sur $nbJobs_m$. Les jobs contribuant le moins au délai moyen sont les jobs de plus petite durée pour lesquels nous considérons un temps de transition nul¹ Les jobs sont ensuite séquencés de manière optimale grâce à l'algorithme décrit précédemment. Si ce séquençement optimal conduit à un délai moyen supérieur à la borne sur $flowtime_m$, alors le nombre de jobs maximal sur m peut être diminué. La seconde règle permet de diminuer la borne supérieure sur $nbJobs_{f,m}$ et la dernière règle permet de réduire la borne supérieure sur $nbJobs_m$.

Expérimentations Des expérimentations sont conduites en utilisant les instances de [1]. Ces 570 instances comportent entre 20 et 70 jobs, 3 à 5 machines et 2 à 5 familles. Les expérimentations, réalisées avec IBM ILOG CPLEX Optimization Studio 12.10, montrent l'efficacité des règles de filtrage. Ces règles permettent de résoudre une centaine d'instances supplémentaires de manière optimale (passant de 50 à 65% d'instances résolues à l'optimal). La rapidité de résolution est aussi grandement augmentée par les règles de filtrage. Cependant aucune règle ne semble vraiment dominer l'autre. Enfin les performances du modèle PPC deviennent plus compétitive avec celles de la PLNE bien que le nombre de solutions optimales obtenues par le modèle PLNE reste légèrement supérieur (+67). Le modèle PPC est cependant capable de trouver 32 solutions réalisables supplémentaires par rapport à la PLNE.

Références

- [1] Arnaud Malapert and Margaux Nattaf. A new cp-approach for a parallel machine scheduling problem with time constraints on machine qualifications. In Louis-Martin Rousseau and Kostas Stergiou, editors, *Integration of Constraint Programming, Artificial Intelligence, and Operations Research*, pages 426–442. Springer International Publishing, 2019.
- [2] A. J. Mason and E. J. Anderson. Minimizing flow time on a single machine with job classes and setup times. *Naval Research Logistics (NRL)*, 38(3) :333–350, 1991.
- [3] Margaux Nattaf, Stéphane Dauzère-Pérès, Claude Yugma, and Cheng-Hung Wu. Parallel machine scheduling with time constraints on machine qualifications. *Comput. Oper. Res.*, 107 :61–76, 2019.
- [4] Ali Obeid, Stéphane Dauzère-Pérès, and Claude Yugma. Scheduling job families on non-identical parallel machines with time constraints. *Annals of Operations Research*, 213(1) :221–234, Feb 2014.

1. Pour être sur d'obtenir une borne inférieure sur le flow time.