



Modeling Reconfigurable Shop Floor Problems

Xavier Delorme, Gérard Fleury, Philippe Lacomme and
Damien Lamy

EasyChair preprints are intended for rapid dissemination of research results and are integrated with the rest of EasyChair.

February 5, 2020

Modélisation des problèmes d'ateliers reconfigurables

Xavier Delorme¹, Gérard Fleury², Philippe Lacomme², et Damien Lamy³

¹ Mines Saint-Etienne, Univ Clermont Auvergne, CNRS, UMR 6158 LIMOS,
Institut Henri Fayol, F - 42023 Saint-Etienne France
delorme@emse.fr

² Université Clermont Auvergne, LIMOS UMR 6158, 63178 Aubière, France
placomme@isima.fr, fleury@isima.fr

³ Mines Saint-Etienne, Institut Henri Fayol, F - 42023 Saint-Etienne, France
damien.lamy@emse.fr

Mots-clés : *ordonnancement, systèmes de production reconfigurables, programmation linéaire, programmation par contrainte.*

Introduction

Les problèmes d'ordonnancement se caractérisent par l'exécution d'opérations sur des machines et dans la majorité des cas, les données relatives aux opérations sont connues et fixes pendant la période de planification considérée. On s'intéresse ici à des modélisations permettant de résoudre des problèmes multimodes c'est-à-dire des problèmes dans lesquels la durée des opérations dépend du mode de fonctionnement du système. Le passage d'un mode à l'autre requiert un temps de reconfiguration ou de setup qui dépend du mode actuel et du mode qui suit. On trouve ces problèmes de manière importante dans les systèmes de production reconfigurables (RMS) qui visent à fournir une bonne adaptabilité aux modifications du marché [1]. Dans les RMS, les reconfigurations ne sont pas seulement logicielles mais également matérielles, induisant des temps de reconfigurations importants pouvant impacter plusieurs ressources simultanément. De manière générale, les problèmes d'ordonnancement sont peu abordés dans les RMS [3]. Le problème traité ici est donc différent de la conception et de l'équilibrage de tels systèmes tel que proposé dans [2] et vise à ordonnancer les opérations dans l'objectif de réduire le makespan.

Formalisation

Le problème étudié consiste à ordonnancer un ensemble J de n jobs, $J = \{J_1, J_2 \dots J_n\}$ sur un ensemble $M = \{M_1, \dots, M_m\}$ de m machines. L'ensemble du système de production est soumis à des configurations. Changer de configuration peut impliquer plusieurs machines, résultant en des temps de traitements des opérations différents. Chaque job i est constitué d'un ensemble de m opérations, notées O_{ij} . Chaque opération O_{ij} a une durée P_{ij}^k où k correspond à la configuration choisie. Les configurations diffèrent des temps de setup, en cela que les configurations peuvent affecter plusieurs machines et ces configurations ne peuvent être activées que lorsque les machines sont inactives. Considérant deux configurations k_1 et k_2 , identifier les machines qui sont concernées par un passage k_1 à k_2 est effectué grâce aux vecteurs $R_{k_1 k_2}^{M_u}$, où chaque valeur du vecteur est égale à 0 si la machine M_u n'est pas impliquée, 1 sinon. Un temps de reconfiguration T_{k_1, k_2} est requis lorsque l'on passe de la configuration k_1 à k_2 . L'objectif est d'ordonnancer les opérations et de définir les configurations utilisées afin de minimiser le temps de traitement de l'ensemble des opérations (makespan). Le problème est une généralisation du

Job-shop qui nécessite à la fois la résolution des disjonctions entre opérations utilisant la même machine et la résolution de l'affectation des configurations aux opérations, tout en tenant compte de l'arrêt simultané d'un ensemble de ressources entre le passage d'une configuration à une autre.

Une solution du problème est un planning donnant les dates de début des opérations et pour chaque opération la configuration dans laquelle elle est effectuée. Une solution se compose donc d'intervalles de temps pendant lesquels une configuration particulière est active et toutes les opérations exécutées dans cette fenêtre de temps le sont selon cette configuration. Sur la Fig. 1 par exemple, l'opération 1 de la pièce 1, l'opération 1 de la pièce 3 et l'opération 1 de la pièce 4 démarrent toutes les trois à la date 0 dans la configuration 2. On passe de la configuration 2 à la configuration 1 entre la date 10 et la date 11, la durée de la reconfiguration valant 1 unité de temps. Ce passage se fait avec un arrêt des machines M1 et M2 mais l'arrêt de la machine M3 n'est pas requis par cette modification. De la même manière, une reconfiguration est appliquée à la date 31, la machine M3 n'étant pas impliquée, la troisième opération de la pièce 4 peut débuter dès la fin de l'opération précédente dans la gamme opératoire de la pièce.

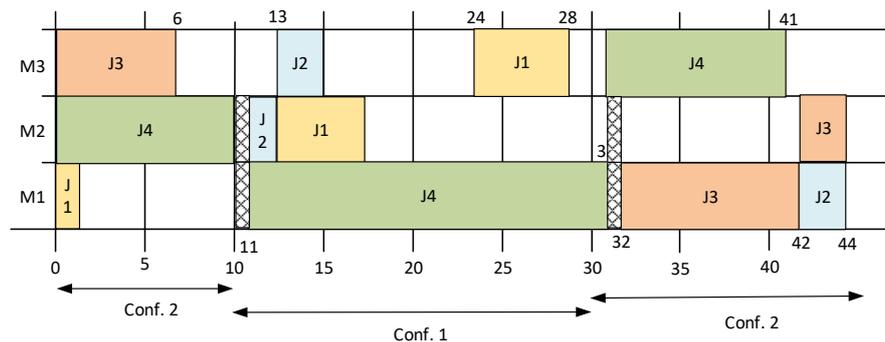


FIG. 1. Exemple de solution

Il est proposé une formalisation linéaire indexée sur le temps et une formalisation de type Programmation par Contrainte qui sont comparées sur 10 exemples avec IBM ILOG CPLEX et le solveur OR-Tools CP-SAT. Les premiers résultats montrent que le solveur OR-Tools possède des performances bien meilleures que CPLEX sur les instances de petites tailles considérées.

Conclusion

La première contribution de ce travail est la proposition d'une nouvelle formalisation des problèmes de type Job-Shop avec Reconfiguration sous la forme d'un problème linéaire et d'un problème de type programmation par contrainte. Actuellement, une généralisation de la notion de graphe disjonctif est étudiée afin de pouvoir définir des méthodes d'exploration de types heuristiques et des recherches locales reposant sur l'exploration du chemin critique. Considérer des durées opératoires suivant des lois de probabilités est également une des perspectives de recherche.

Références

1. Koren Y, Heisel U, Jovan F, et al (1999) Reconfigurable Manufacturing Systems. CIRP Ann 48:527–540
2. Essafi M, Delorme X, Dolgui A (2012) A reactive GRASP and Path Relinking for balancing reconfigurable transfer lines. Int J Prod Res 50:5213–5238. <https://doi.org/10.1080/00207543.2012.677864>
3. Azab A, Naderi B (2015) Modelling the Problem of Production Scheduling for Reconfigurable Manufacturing Systems. Procedia CIRP 33:76–80. <https://doi.org/10.1016/j.procir.2015.06.015>