

Résolution d'un problème d'ordonnancement de type flowshop reentrant sans-attente

Karim Amrouche¹, Mourad Boudhar¹, Mohamed Bendraouche¹ and Farouk Yalaoui²

¹ RECITS laboratory, Faculty of Mathematics, USTHB University,
BP 32 Bab-Ezzouar, El-Alia 16111, Algiers, Algeria.

amrouche-karim@hotmail.com, mmboudhar@yahoo.fr, mbendraouche@yahoo.fr

² University of Technology of Troyes, LOSI laboratory, ICD UAR CNRS 6281
12 street Marie Curie, BP 2060, Troyes 10010, France.

farouk.yalaoui@utt.fr

1 Introduction

Dans ce travail, on s'intéresse au problème d'ordonnancement d'atelier de type chain-reentrant shop sur deux machines avec recirculation sur la première machine. Chaque tâche T_j doit d'abord être traitée sur la première machine M_1 , puis sur la deuxième machine M_2 , et être retraitée par la machine M_1 pour sa dernière opération. On suppose aussi qu'aucune attente entre deux opérations d'une même tâche n'est permise. Ce genre de contrainte correspond à un enchaînement sans attente des opérations dans la gamme de fabrication, et est connue dans la littérature sous le nom de contrainte de "sans-attente" ou ("no-wait constraint" en Anglais). En pratique, on peut rencontrer ce genre de problèmes dans l'industrie métallurgique ou agroalimentaire où le stockage sur une machine n'est pas autorisé.

L'objectif est de minimiser la date de fin de traitement de l'ensemble des tâches.

Ce problème est noté $P_1 : F2 | chain - reentrant, no - wait | C_{max}$.

On note par :

a_j : Le temps de traitement de la première opération de la tâche T_j sur la première machine.

b_j : Le temps de traitement de la tâche T_j sur la deuxième machine.

c_j : Le temps de traitement de la deuxième opération sur la première machine.

Exemple

Pour illustrer ce problème, on considère l'instance suivante : 5 tâches T_1, T_2, T_3, T_4 et T_5 à ordonnancer sur 2 machines. Les temps de traitement sont donnés dans la table 1.

TAB. 1 – Temps de traitement des tâches

T_i	T_1	T_2	T_3	T_4	T_5
a_j	1	2	1	1	1
b_j	3	3	4	2	1
c_j	2	2	2	2	1

Un ordonnancement réalisable de cette instance avec $C_{max} = 16$ est reporté sur la figure 1.

2 Résolution

Dans [2] nous avons montré que le problème P_1 est NP-difficile au sens fort par une réduction du problème NMTS (Numerical Matching with Target Sum) qui est NP-difficile au sens fort [3]. Dans ce travail, un nouveau modèle mathématique est proposé pour résoudre le problème

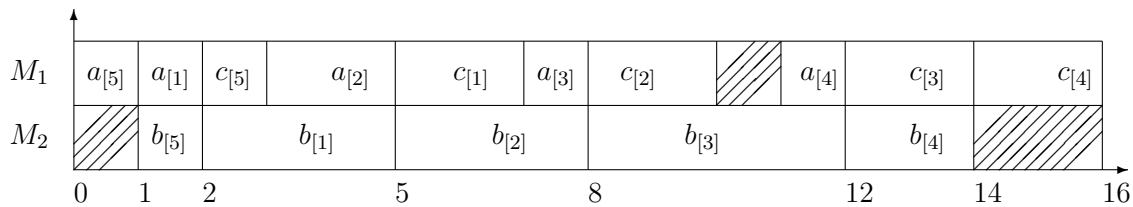


FIG. 1 – Solution du problème $P_1 : F2 |chain - reentrant, no - wait| C_{max}$.

P_1 , ce dernier a été testé sous le solveur Gurobi. Après expérimentations numériques nous avons remarqué que le modèle permet de résoudre des petites instances où le nombre de tâches est inférieur à 20. Une fois que le nombre de tâches dépasse 20, le temps d'exécution devient important (dépassant les deux heures). Un autre modèle mathématique pour ce même problème a été proposé lorsque l'ordre d'exécution des tâches est connu à l'avance. Notons par $P_2 : F2 |chain - reentrant, perm, no - wait| C_{max}$ ce problème, c.à.d. le problème P_1 auquel on ajoute une permutation faisant partie des données. La complexité de ce dernier problème est encore un problème ouvert. Les résultats des tests ont montré que le deuxième modèle permet de résoudre le problème P_2 même lorsque le nombre de tâches est grand, allant jusqu'à 500 tâches. Enfin, nous avons proposé une heuristique pour résoudre le problème P_1 , son principe est d'appliquer une recherche aléatoire dans le voisinage des solutions réalisables ; à chaque fois on génère une nouvelle solution, qui est une permutation de tâches, on fait appel au modèle mathématique pour calculer le makespan de cette permutation. Une fois le critère d'arrêt rencontré, on retourne la meilleure solution trouvée. Le critère d'arrêt utilisé pour notre heuristique est le nombre de solutions générées. Pour notre algorithme ce nombre est fixé à 1000. D'après les expérimentations numériques, nous avons observé que l'heuristique proposée donne de bonnes solutions, avec une déviation moyenne inférieure à 20% dans la plupart des cas.

Références

- [1] Karim Amrouche and Mourad Boudhar. Two machines flow shop with reentrance and exact time lag. *RAIRO, Operation Research*. 50 : 223–232, 2016.
- [2] Karim Amrouche, Mourad Boudhar, Mohamed Bendraouche and Farouk Yalaoui. Chain-reentrant shop with an exact time lag : new results. *International Journal of Production Research*. DOI : 10.1080/00207543.2016.1205235
- [3] M R. Garey and D S. Johnson. *Computers and Intractability*. W. H. Freeman and Company, New York, 1979.
- [4] M Y Wang , S P Sethi and S L Van De Velde. Minimizing makespan in a class of reentrant shops. *Oper. Res*, 45 : 702–712, 1997.
- [5] V. Lev and I. Adiri, V-shop scheduling. *European J. Opnl. Res*. 18 : 51–56, 1984.