

# Méthodes approchées pour la résolution des problèmes d'ordonnancement reentrants

Naim Yalaoui<sup>1,2</sup>, Lionel Amodeo<sup>1</sup>, Farouk Yalaoui<sup>1</sup>, Halim Mahdi<sup>2</sup>

<sup>1</sup> ICD-LOSI (FRE CNRS 2848), Université de Technologie de Troyes, 12 rue Marie Curie, 10010 Troyes, France

{naim.yalaoui, lionel.amodeo, farouk.yalaoui}@utt.fr

<sup>2</sup> Ets CAILLAU, 28 rue Ernest Renan, 92130, Issy les Moulineaux, France

{nyalaoui, hmahdi}@caillau.com

**Mots-Clés :** *metaheuristiques, logique floue, ordonnancement, ateliers flow shop reentrant*

## 1 Introduction et description du problème

Ce travail porte sur la résolution des problèmes d'ordonnancement dans un atelier flow shop reentrant. Dans les systèmes de fabrication et surtout dans l'étape opérationnelle, tous les gestionnaires cherchent à optimiser leur production par la résolution des problèmes d'ordonnancement. Notre étude vise à résoudre un problème de type ordonnancement hybride reentrant.

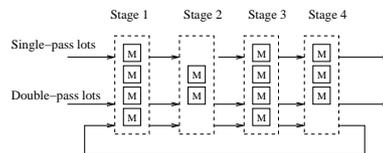


FIGURE 1 – Atelier flow shop hybride  $M=\{4,2,4,3\}$  [5]

Comme indiqué dans la figure, nous avons  $E$  étages qui contiennent  $M_j (j = 1 \dots E)$  machines parallèles, identiques. Le système traite des commandes divisées en lots dont certains doivent passer à plusieurs reprises sur les machines d'où la notion de flux réentrant. Notre problème est d'ordonner les tâches avec pour objectif de minimiser la somme des retards.

## 2 Méthodes de résolution

Afin de résoudre notre problématique, plusieurs méthodes ont été développées telles que les algorithmes génétiques (GA), les algorithmes génétiques sous contrôle de logique floue (FLCGA) [5] [3], les algorithmes d'essais particuliers (PSO), la règle de priorité EDD (Earliest Due date), et la méthode exacte ENUM (énumération de toutes les solutions). Celles-ci présentent chacune des avantages et des inconvénients, d'où la nécessité de les comparer. L'algorithme génétique (GA) utilisé dans ce travail a principalement deux paramètres de probabilités : la probabilité de mutation notée  $p_m$  et la probabilité de croisement notée  $p_c$  qui sont, en général, constants. Les valeurs de ces deux paramètres ont été modifiées dans le FLCGA et sont

definies dynamiquement à chaque itération du GA, d'où l'introduction de la logique floue. Pour rappel, la logique floue a été introduite par Lotfi Zadeh [6], d'une part, pour le traitement des données et d'autre part, utilisée pour contrôler les systèmes. Le FLC est composé de 3 étapes : *fuzzyfication*, *décision* et *defuzzyfication*. Dans notre algorithme deux contrôleurs existent : l'un est dédié à la probabilité de mutation  $p_m$  et l'autre est dédié à la probabilité de mutation  $p_c$ . Leur fonctionnement est décrit dans [5]. Un autre algorithme développé est l'optimisation à essaim particulaire (PSO) qui est une technique d'optimisation stochastique proposée par Eberhart et Kennedy (1995) [2]. L'algorithme est inspiré du comportement de la population d'abeilles. Chaque particule a une vitesse, une mémoire et des informatrices. La position et la vitesse de la première population de particules sont générées de manière aléatoire. Pour décider de se déplacer, la particule a besoin de trois informations : sa vitesse réelle, sa meilleure performance et la meilleure performance de ses voisines. Cet algorithme est basé sur trois paramètres  $w$ ,  $C1$  et  $C2$  qui lui permettent, respectivement, de poursuivre leurs propre chemin, revenir sur leurs pas, ou suivre la meilleure voisine.

### 3 Étude expérimentale

Comme dans l'étude de Choi *et al* [1] et nos études précédentes [5], [4], les algorithmes sont testés sur 324 cas : 4 problèmes pour chaque combinaison : de 3 niveaux de commandes (10, 15 et 20), 3 niveaux de nombre d'étages (4, 6 et 8), et 9 niveaux pour les paramètres utilisés pour générer des délais. Dans les problèmes testés, des délais flexibles sont utilisés et contrôlés par deux paramètres : T et R, appelés facteur de délais et le rang de délais respectivement. Les valeurs de T sont les suivantes :  $\{0.1, 0.1, 0.1, 0.1, 0.1, 0.3, 0.3, 0.3, 0.5\}$  et en ce qui suit les valeurs de R  $\{0.8, 1.0, 1.2, 1.4, 1.6, 0.8, 1.0, 1.2, 0.8\}$ . Le délai est généré selon la formule :  $DU(P(1 - T - \frac{R}{2}), P(1 - T + \frac{R}{2}))$  où  $DU(x, y)$  est une distribution unifrom dans  $[x, y]$ .  $P = \max_j \{ \min_i \{ \sum_{l=1}^{j-1} P_{il} \} + \frac{1}{M_j} \cdot \sum_{i=1}^{n_b} P_{ij} + \min_i \{ \sum_{l=j+1}^m P_{il} \} \}$   $P$  est la borne inférieure du makespan et  $n_b$  est le nombre total de lots à exécuter. Le nombre de machines ( $M_j$ ) de chaque étage et le nombre de lots de chaque commande sont générés par  $DU(1; 10)$  et  $DU(5; 15)$  respectivement. Les différentes méthodes : FLCGA, GA, PSO, EDD sont exécutées sur les 324 problèmes. Pour les problèmes de taille moyenne où la méthode exacte est possible, elle est exécutée et comparée au reste des méthodes. 100% des solutions obtenues par le FLCGA présentent une solution optimale, suivi de près par le GA avec 95% des cas. Le PSO quant à lui obtient 58% des solutions optimales, et pour le reste, il présente avec un gap moyen par rapport au FLCGA et à l'ENUM d'environ 8%. Pour les problèmes de plus grande taille ou l'énumération complète n'est pas possible, les 4 méthodes présentent chacune des résultats plus au moins intéressants, pour donner l'avantage de qualité des résultats au FLCGA et celui de la vitesse du temps de calcul au PSO.

### Références

- [1] S. Choi, Y. Kim, and G. Lee. Minimizing total tardiness of orders with reentrant lots in a hybrid flowshop. *International Journal of Production Research*, 43(11) :2149–2167, 2005.
- [2] R.C. Eberhart J. Kennedy. Particle swarm optimization. *IEEE International Conference on Neural Networks, Perth, Australia*, pages 1942–1948, 1995.
- [3] N. Yalaoui, L. Amodeo, F. Yalaoui, and H. Mahdi. Pso method to solve a scheduling problem. In *VII. International Logistics & Supply Chain Congress, Istanbul, Turkey*, 5th-6th November 2009.
- [4] N. Yalaoui, M. Camara, L. Amodeo, F. Yalaoui, and H. Mahdi. New heuristic for scheduling re-entrant production lines. In *International Conference on Computers & Industrial Engineering (CIE39), Troyes, France*, pages 199–204, July 2009.
- [5] N. Yalaoui, F. Dugardin, F. Yalaoui, L. Amodeo, and H. Mahdi. *Production Engineering and Management under Fuzziness*, chapter Fuzzy Project Scheduling. Springer-Verlag, 2009, Soumis.
- [6] Lotfi A. Zadeh. Fuzzy sets. *Information and Control*, 8(3) :338–353, 1965.