

---

# Ordonnancement en temps réel d'un jobshop par métaheuristique hybride : étude comparative

Y. Houbad, M. Souier, A. Hassam, Z.Sari

Laboratoire d'automatique Tlemcen

Faculté de technologie, Université Abou Bakr Belkaid, BP 230, Tlemcen 13000, Algérie.

houbadyamina@yahoo.fr, a\_hassam@mail.univ-tlemcen.dz, m\_souier@mail.univ-tlemcen.dz, z\_sari@mail.univ-tlemcen.dz

---

**RÉSUMÉ.** Les problèmes d'ordonnancement sont souvent des problèmes d'optimisation combinatoire de type NP-difficile. Leur résolution nécessite des méthodes dédiées à leur degré de complexité, c'est pour cette raison que plusieurs heuristiques et métaheuristiques ont été conçues.

Notre étude est consacrée à l'adaptation de l'algorithme mémétique pour résoudre le problème de sélection de routages alternatifs en temps réel dans un jobshop.

Le choix de cette métaheuristique est dû au fait que les métaheuristiques hybrides fournissent une compensation des faiblesses pouvant se trouver dans d'autres algorithmes, tel que l'algorithme que nous avons utilisé (l'algorithme mémétique) qui améliore la vitesse de convergence de l'algorithme génétique en lui ajoutant une recherche locale. Ensuite on va faire une comparaison entre cet algorithme et l'algorithme génétique et d'autres métaheuristiques déjà utilisé pour résoudre ce même problème afin d'avoir une idée sur les méthodes les plus performantes pour ce problème et opter pour la plus efficace.

**ABSTRACT.** The Scheduling problems are often problems of combinatorial optimization of NP-hard. Their resolution requires methods dedicated to their degree of complexity, for this reason that several heuristics and metaheuristics have been designed. Our study is devoted to the adaptation of the memetic algorithm to solve the problem of selection of alternative routings in real time of a JobShop. The choice of this metaheuristic is because the hybrid metaheuristics provide compensation for weaknesses that can be found in other algorithms, such as the algorithm that we used (the memetic algorithm) which improves the speed of convergence of the genetic algorithm by adding a local search. Then we will make a comparison between this algorithm and the genetic algorithm and other metaheuristics previously used to solve the same problem to get an idea on the most efficient methods for this problem and choose the most effective.

**MOTS-CLÉS:** ordonnancement, jobshop, métaheuristiques, métaheuristiques hybrides, algorithme mémétique, simulation.

**KEYWORDS:** scheduling, jobshop, metaheuristics, hybrid metaheuristics, memetic algorithm, simulation.

---

## 1. Introduction

Les systèmes flexibles de production sont définis par MacCarthy et Liu (1993) : « un système flexible de production est un système de production capable de produire différents types de pièces, composé de machines à commande numérique ou à contrôle numérique et d'un système automatisé de stockage connectés par un système automatisé de manutention. Le fonctionnement du système entier est sous le contrôle et le pilotage d'un système informatique ». Donc, les systèmes flexibles de production (FMS) peuvent changer rapidement de produits et de séquences de produits sans perdre leurs productivités, à condition qu'il soit possible d'obtenir le bon produit, palette, support, ou outil, à la bonne place au bon moment.

Les problèmes liés à la technologie des FMS sont relativement complexes comparés aux systèmes de production traditionnels. C'est pour cette raison que les problèmes d'ordonnancement sont de type NP-complet. Donc, il n'existe pas de méthode exacte capable de les résoudre, c'est une raison pour laquelle plusieurs heuristiques et métaheuristiques ont été conçues. Une heuristique est un algorithme d'optimisation qui a pour but de trouver une solution réalisable de la fonction objective, mais sans garantir d'optimalité. Parmi ces heuristiques certaines sont adaptables à un grand nombre de problèmes différents sans changements majeurs dans l'algorithme, on parle alors des métaheuristiques.

Les métaheuristiques sont souvent inspirées de la biologie (algorithmes évolutionnaires...), de l'éthologie (colonies de fourmis, essaims particulaires,...), et de la physique (recuit simulé,...), elles sont généralement conçues au départ pour des problèmes discrets, mais peuvent s'adapter à d'autres types de problèmes.

Pour aller plus loin dans la recherche de solutions, il faut avant tout pouvoir détecter de nouvelles solutions. Les algorithmes à base de population présentent un intérêt particulier : le parallélisme intrinsèque. En ajoutant de nouveaux composants à ces algorithmes, on peut alors construire des algorithmes « hybrides ». L'une des faiblesses d'un algorithme génétique comme la vitesse de convergence trop lente peut, par exemple, être compensée par l'ajout d'une méthode de recherche locale. C'est le cas des algorithmes mémétiques de Moscato.

Cet article s'articule autour de quatre sections. La première est un état de l'art qui regroupe les travaux les plus importants dans la sélection de routages alternatifs. Dans la deuxième nous définissons le contexte de travail présenté. Dans la troisième, nous définissons le mode de fonctionnement des algorithmes génétiques et de l'algorithme mémétique adapté. La dernière section est consacrée aux résultats trouvés et aux interprétations.

## 2. Etat de l'art :

Plusieurs travaux ont été réalisés dans le cadre de l'ordonnancement d'un FMS depuis le début des années 80 lorsque les ateliers flexibles ont commencé à gagner l'acceptation par les pays industrialisés et ce domaine continue à attirer les intérêts des secteurs académiques et industriels.

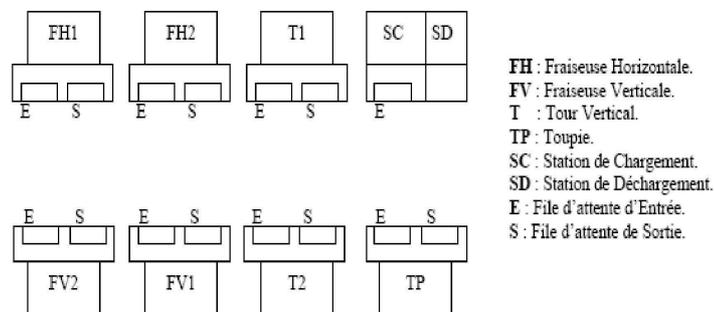
La flexibilité des routages représente la possibilité d'utiliser des machines, ou des opérations différentes, pour réaliser des produits sous la même configuration du système. On trouve dans la littérature de nombreux travaux qui se sont intéressés à l'influence de la flexibilité de routage sur les performances d'un FMS avec et sans panne de machines. On peut citer les travaux de (Tsubone et al., 1999) qui ont étudié par simulation l'impact de la flexibilité des machines et de routages en variant différentes conditions de l'atelier sur le temps de cycle moyen des pièces, on trouve aussi les travaux de (Mahmoodi et al., 1999) qui ont étudié l'effet des règles de priorité et des niveaux de flexibilité de routage sur les différentes performances d'un FMS.

Citons aussi les travaux de (Saygin et al., 1999) qui ont proposé une plateforme qui intègre le processus de planification flexible et l'ordonnancement prédictif, ils ont présenté un concept nommé Dissimilarity Maximisation Method (DMM) pour minimiser la congestion dans un FMS, l'idée de cette règle est de maximiser les dissimilarités entre les routages occupés, dans leur algorithme chaque routage ne contient qu'une seule pièce à la fois. L'efficacité de cette règle dans la résolution des problèmes de sélection de routage en temps réel a été démontré dans (Saygin *et al.*, 2001) et (Gomri et al., 2007) où elle a surpassé deux autres règles de sélection de routages alternatifs FIFO/FA (First-In First-Out/ First Available) et Equal Probability Loading (EPL) si chaque machine utilise la règle FIFO comme règle de séquençement. Dans l'étude de (Hassam et al., 2010) sur la règle DMM, dans laquelle ils ont proposé la règle DMM modifiée qui est une modification de la règle DMM qui vise à garder le même principe qui dépend de la maximisation des coefficients de dissimilarité pour la sélection des différents routages alternatifs mais en affectant plusieurs pièces à un seul routage.

On trouve aussi dans les travaux de (Souier et al., 2010) l'adaptation de plusieurs métaheuristiques (les colonies de fourmis, les algorithmes génétiques, les essais particuliers, le recuit simulé, la recherche tabou et l'électromagnétisme) pour résoudre le problème de sélection de routages alternatifs dans un FMS avec un nombre important de pièces mais en temps différé, et dans (Souier et al., 2010) ils ont réussi à appliquer ces mêmes métaheuristiques sur le même problème en temps réel. Dans (Souier et al., 2010) ils ont adapté des métaheuristiques pour résoudre le problème en temps réel sans et avec pannes des machines avec ré-ordonnancement des pièces de la station de chargement.

### 3. présentation du modèle étudié :

Le système flexible de production qu'on étudie a déjà été traité dans la littérature, il est composé de sept machines et de deux stations une de chargement et l'autre de déchargement, le système va traiter six types de pièces. Chaque machine du système comprend une file d'attente d'entrée et une file d'attente de sortie, la station de chargement comprend aussi une file d'attente d'entrée. La figure (1) représente les routages alternatifs possibles pour chaque type de pièce avec le temps de traitement pour chaque machine.



**Figure 1.** Configuration du modèle FMS étudié.

Les opérations sur le système étudié sont basées sur les suppositions suivantes :

- Les routages alternatifs pour chaque type de pièce sont connus avant le début de la production.
- Le temps de traitement est déterminé et il comprend le temps de changement des outils et le temps d'exécution de la machine.
- chaque machine peut traiter une seule pièce à la fois.

Type de parts	Taux d'arrivée	Routage et temps de traitement (min)
A	17 %	SC - T <sub>1</sub> (30) - FV <sub>1</sub> (20) - SD
		SC - T <sub>1</sub> (30) - FV <sub>2</sub> (20) - SD
		SC - T <sub>2</sub> (30) - FV <sub>1</sub> (20) - SD
		SC - T <sub>2</sub> (30) - FV <sub>2</sub> (20) - SD
B	17 %	SC - T <sub>1</sub> (20) - TP (1) - FV <sub>1</sub> (15) - SD
		SC - T <sub>1</sub> (20) - TP (1) - FV <sub>2</sub> (15) - SD
		SC - T <sub>2</sub> (20) - TP (1) - FV <sub>1</sub> (15) - SD
		SC - T <sub>2</sub> (20) - TP (1) - FV <sub>2</sub> (15) - SD
C	17 %	SC - T <sub>1</sub> (40) - FV <sub>1</sub> (25) - SD
		SC - T <sub>1</sub> (40) - FV <sub>2</sub> (25) - SD
		SC - T <sub>2</sub> (40) - FV <sub>1</sub> (25) - SD
		SC - T <sub>2</sub> (40) - FV <sub>2</sub> (25) - SD
D	21 %	SC - T <sub>1</sub> (40) - TP (1) - T <sub>1</sub> (20) - FH <sub>1</sub> (35) - SD
		SC - T <sub>1</sub> (40) - TP (1) - T <sub>1</sub> (20) - FH <sub>2</sub> (35) - SD
		SC - T <sub>1</sub> (40) - TP (1) - T <sub>2</sub> (20) - FH <sub>1</sub> (35) - SD
		SC - T <sub>1</sub> (40) - TP (1) - T <sub>2</sub> (20) - FH <sub>2</sub> (35) - SD
		SC - T <sub>2</sub> (40) - TP (1) - T <sub>1</sub> (20) - FH <sub>1</sub> (35) - SD
		SC - T <sub>2</sub> (40) - TP (1) - T <sub>1</sub> (20) - FH <sub>2</sub> (35) - SD
		SC - T <sub>2</sub> (40) - TP (1) - T <sub>2</sub> (20) - FH <sub>1</sub> (35) - SD
		SC - T <sub>2</sub> (40) - TP (1) - T <sub>2</sub> (20) - FH <sub>2</sub> (35) - SD
E	20 %	SC - T <sub>1</sub> (25) - TP (1) - T <sub>1</sub> (35) - FH <sub>1</sub> (50) - SD
		SC - T <sub>1</sub> (25) - TP (1) - T <sub>1</sub> (35) - FH <sub>2</sub> (50) - SD
		SC - T <sub>1</sub> (25) - TP (1) - T <sub>2</sub> (35) - FH <sub>1</sub> (50) - SD
		SC - T <sub>1</sub> (25) - TP (1) - T <sub>2</sub> (35) - FH <sub>2</sub> (50) - SD
		SC - T <sub>2</sub> (25) - TP (1) - T <sub>1</sub> (35) - FH <sub>1</sub> (50) - SD
		SC - T <sub>2</sub> (25) - TP (1) - T <sub>1</sub> (35) - FH <sub>2</sub> (50) - SD
		SC - T <sub>2</sub> (25) - TP (1) - T <sub>2</sub> (35) - FH <sub>1</sub> (50) - SD
		SC - T <sub>2</sub> (25) - TP (1) - T <sub>2</sub> (35) - FH <sub>2</sub> (50) - SD
F	8 %	SC - FH <sub>1</sub> (40) - SD
		SC - FH <sub>2</sub> (40) - SD

**Tableau 1.** Routages alternatifs et temps de traitement des pièces.

#### 4. Les métaheuristiques :

##### 4.1. L'algorithme génétique

L'algorithme génétique présenté ici est celui adapté par (Souier, 2009) pour résoudre ce même problème.

$n$  est la capacité des files d'attente.

Chaque chromosome artificiel représente les routages choisis des premières pièces par un code. Après chaque évaluation on modifie la meilleure solution s'il y'a une amélioration.

La mutation consiste à modifier les routages de certaines pièces parmi les  $n$  premières pièces contenues dans la file infinie.

Le croisement est fait en prenant deux génotypes, puis on choisi un endroit le long de la chaine, on coupe chacun d'eux à cet endroit et on relie la partie gauche d'une chaine à la partie droite et vice versa.

S'il y'a une place libre dans la station de chargement alors

Génération d'une population aléatoire

**Tant que** critère d'arrêt non atteint faire

**Pour** chaque individu

Evaluation de la fitness de cet individu (le produit des charges de routages)

**Si** la fonction objective est améliorée alors mettre à jour la meilleure solution

**Fin pour**

Sélection des individus pour la reproduction (opérateur de sélection)

Application de l'opérateur de croisement (on obtient un ensemble de nouveaux individus)

Application de l'opérateur de mutation sur les nouveaux individus

Constitution de la nouvelle génération

**Fin tant que**

**Finsi**

#### **4.2. L'algorithme mémétique**

Moscato introduit pour la première fois les algorithmes mémétiques. On rencontre aussi le nom d'algorithme génétique hybrides ou celui de (genetic local search). Quelque soit le nom qu'on lui donne, l'idée principale de cette technique est de rendre plus agressif un algorithme génétique par l'ajout d'une recherche locale en plus de la mutaion.

L'idée de Moscato est d'ajouter une recherche locale à l'algorithme génétique, cette recherche sera appliquée à tout nouvel individu obtenu au cours de la recherche.

Cette simple modification peut entrainer de profonds changements dans l'algorithme. Après avoir créé un nouvel individu à partir de deux parents sélectionnés, on applique une recherche locale et sous certaines conditions on applique un opérateur de mutation à cet individu.

L'algorithme mémétique utilisé est le suivant :

S'il ya une place libre dans la station de chargement alors

Génération d'une population aléatoire

**Tant que** critère d'arrêt non atteint **faire**

**Pour** chaque individu

Evaluation de la fitness de cet individu (le produit des charges de routages)

Appliquer une recherche locale sur chaque individu de la population générée

**Fin pour**

**Répéter**

Sélection des individus pour la reproduction (opérateur de sélection)

Application de l'opérateur de croisement (on obtient un ensemble de nouveaux individus)

Appliquer la recherche locale sur chaque individu obtenu

Application de l'opérateur de mutation sur les nouveaux individus

Constitution de la nouvelle génération

**Fin tant que**

**Finsi**

La méthode de recherche locale que nous avons utilisée est une méthode simple qui est la méthode de descente donnée par l'algorithme suivant :

**Pour** chaque individu (qui représente une condition initiale  $x$ ) pour cette méthode)

**Tant que** la condition d'arrêt n'est pas vérifiée faire

Modifier les routages de certaines pièces (donc trouver une autre solution voisine  $x'$  de  $x$ )

**Si** la fonction objective a été améliorée (produit des charges des routages) :  $f(x') > f(x)$  alors

Remplacer  $x$  par  $x'$

**Finsi**

**Fin tant que**

## 5. Résultats et interprétations :

Dans cette section, et afin de montrer les améliorations apportées par l'algorithme mémétique proposé, nous allons présenter les résultats obtenus pour différentes valeurs de la capacité de la file d'attente de la station de chargement et du taux de création des pièces. Les résultats obtenus seront comparés avec ceux obtenus par l'algorithme génétique, les colonies de fourmis, et la règle DMM

modifiée pour trois critères de performances (taux de sortie des pièces, temps de cycle, et les en-cours).

### 5.1. Le taux de sortie des pièces

Taux de création	1/40	1/35	1/30	1/25	1/20	1/15	1/10	1/5
DMMmodifiée	99.99	99.99	99.98	99.71	84.47	60.73	41.67	21.15
GA	99.99	99.99	99.99	99.99	98.44	71.08	47.25	23.7
ACO	99.99	99.99	99.99	99.99	90.52	64.54	43.2	21.51
MA	99.99	99.99	99.99	99.99	97.55	75.82	49.09	24.80

**Tableau 2.** Taux de sortie des pièces pour une capacité de file d'attente=2.

Taux de création	1/40	1/35	1/30	1/25	1/20	1/15	1/10	1/5
DMMmodifiée	99.99	99.99	99.99	99.98	90.61	67.79	45.42	22.58
GA	99.99	99.99	99.99	99.99	99.99	72.06	47.88	23.93
ACO	99.99	99.99	99.99	99.99	99.99	68.78	45.73	22.85
MA	99.99	99.99	99.99	99.99	99.99	72.48	49.20	24.55

**Tableau 3.** Taux de sortie des pièces pour une capacité de file d'attente=4.

Les tableaux 2 et 3 montrent que le taux de création des pièces est le même pour toutes les méthodes pour un taux de création inférieur à 1/25, au dessus de cette valeur l'algorithme mémétique a donné les meilleurs résultats, donc pour un système saturé l'algorithme mémétique domine les autres méthodes.

### 5.2. Le temps de cycle

Taux de création	1/40	1/35	1/30	1/25	1/20	1/15	1/10	1/5
DMMmodifiée	81.9	87.8	101.6	155.8	203.3	204.7	207.2	204.2
GA	89.6	81.8	88.6	98.7	132.7	169.8	168.9	168.1
ACO	90.9	83.7	89	98.8	155.2	166.6	165.4	163.5
MA	88.82	81.75	89.25	98.16	119.43	123.24	123.92	123.95

**Tableau 4.** Temps de cycle pour une capacité de file d'attente=2.

Taux de création	1/40	1/35	1/30	1/25	1/20	1/15	1/10	1/5
DMMmodifiée	81.9	87.8	101.6	155.8	203.3	204.7	207.2	204.2
GA	89.6	81.8	88.6	98.7	132.7	169.8	168.9	168.1
ACO	90.9	83.7	89	98.8	155.2	166.6	165.4	163.5
MA	88.82	81.75	89.25	98.16	119.43	123.24	123.92	123.95

**Tableau 5.** Temps de cycle pour une capacité de file d'attente=4.

Le tableau 4 et le tableau 5 montrent que pour une capacité de file d'attente égale à 2 et 4 (Système saturé) l'algorithme mémétique a donné des valeurs de temps de cycle nettement inférieures à celles obtenues par les autres méthodes, en le comparant avec l'algorithme génétique nous remarquons qu'il y'a une amélioration considérable de temps de cycle du système.

### 5.3. Les en-cours

Taux de création	1/40	1/35	1/30	1/25	1/20	1/15	1/10	1/5
DMMmodifiée	6.79	12.95	15.79	19.01	17.84	18.19	16.86	18.12
GA	3.9	4.04	4.46	5.23	7.45	8.9	8.85	8.85
ACO	3.9	4.04	4.45	5.21	7.29	8.4	8.44	8.46
MA	2.88	3.05	3.48	4.25	5.87	6.34	6.28	6.27

**Tableau 6.** Les en-cours pour une capacité de file d'attente=2.

Le tableau 6 montre l'algorithme mimétique a donné des résultats meilleurs que ceux trouvés par les autres méthodes.

## 6. Conclusion :

Dans cet article nous nous sommes intéressés à l'adaptation d'une nouvelle métaheuristique pour la résolution de sélection de routages alternatifs dans un Job Shop, en comparant cette métaheuristique avec celle de l'algorithme génétique on remarque que l'introduction de la recherche locale a amélioré les résultats trouvés par l'algorithme génétique.

Toutes les métaheuristicues s'appuient sur un équilibre entre l'intensification de la recherche et la diversification de celle-ci. D'un coté, l'intensification permet de rechercher des solutions de plus grande qualité en s'appuyant sur les solutions déjà trouvées et de l'autre, la diversification met en place des stratégies qui permettent d'explorer un grand espace de solutions et d'échapper à des minima locaux.

Dans l'algorithme mémétique, l'intensification est produite de toute évidence par l'application nouvelle de la recherche locale. L'opérateur de mutation assure la diversification de la méthode.

L'hybridation des métaheuristiques avec des méthodes de recherche locale, des méthodes exactes, ou d'autres métaheuristiques peut augmenter leur efficacité.

## 7. Bibliographie

- Ghomri L., Influence des contraintes et des perturbations sur les performances des règles de routages dans un FMS, *la conférence internationale Conception et Production Intégrées*, 2007.
- Hassam A., Manipulation des routages alternatifs en temps réel dans les systèmes flexibles de production, Thèse de Magister, Université Abou Bekr Belkaid Tlemcen, 2006.
- Mahmoodi F.MOSIER C.T., The effects of sheduling rules and routing flexibility on the performace of a random flexible manufacturing system, *The international journal of flexible manufacturing systems*, p. 271-289.
- Moscato P., On evolution, search, optimization, genetic algorithms and material arts: Towards memetic algorithms. *Technical Report C3P 826, Caltech Concurrent Computation Program*, 1989.
- Saygin C., Chen F.F., Singh J., Real time manipulation of alternative routings in flexible manufacturing systems: A simulation study. *International journal of advanced manufacturing technology*, 2001, 18, p. 755-763.
- SAYGIN C. KILICK S.E., Integrating flexible manufacturing systems with sheduling in flexible manufacturing system. *International journal of advanced manufacturing technology*, 15(4), p. 268-280.1999.
- Sevaux M., Métaheuristiques stratégies pour l'optimisation de la production de biens et de services, Habilitation à diriger des recherches, Préparée au Laboratoire d'Automatique, de Mécanique d'informatique Industrielles et Humaines du CNRS (UMR CNRS 8530) dans l'équipe Systèmes de Production, 2004.
- Souier M., Métaheuristiques pour la manipulation de routages alternatifs en temps réel dans un jobshop, Thèse du magister, Université Abou Bekr Belkaid Tlemcen, 2009.
- Souier M., Hassam. A., Sari Z. Evaluation of metaheuristics in manipulation of alternative routing, *International conference on electrical engineering design and technologies*, Hammamet Tunisia, 2008.
- Souier M., Hassam A., Sari Z., Metaheuristics for real time routing selection in FMS, Book chapter in: Lyes Benyoucef and Bernard Grabot (ED.), *Artificial intelligence techniques for networked manufacturing enterprises management*, Springer-Verlag, ch. Pp.221-247.ISNB 978-1-84996-118-9.
- Tsubone H. Horikawa M.,Comparison between Machine flexibility and routing Flexibility, *The international journal of flaxible manufacturing systems*, 11, p. 83-101.