

# Métaheuristiques pour la manipulation de routages alternatifs dans un job shop

MEHDI SOUIER, AHMED HASSAM, ZAKI SARI

LABORATOIRE D'AUTOMATIQUE DE TLEMCCEN

Faculté des sciences de l'ingénieur, Université d'Abou Bekr Belkaid, BP 230, Tlemcen 13000, Algérie

{m\_souier,a\_hassam,z\_sari}@mail.univ-tlemcen.dz

---

**Résumé** -Le problème d'ordonnancement de type Job Shop est une classe très connue des problèmes d'optimisation NP-difficile. En raison de leur complexité, plusieurs heuristiques et métaheuristiques ont été proposées pour le résoudre. Cet article présente les résultats d'une étude de simulation d'un job shop avec flexibilité de routages.

L'objectif de cette étude est de comparer les performances de quelques métaheuristiques (les colonies de fourmis, les algorithmes génétiques, le recuit simulé, la recherche avec tabou, les essais particuliers et l'électromagnétisme) avec les méthodes de sélection de routage alternative en temps réel DMM (Dissimilarity Maximization Method) et DMM modifiée afin d'avoir une idée sur l'efficacité de ces métaheuristiques et de choisir la plus efficace.

Pour valider les résultats de cette étude, nous avons simulé la DMM et la DMM modifiée sur un modèle de Job shop en utilisant le logiciel de simulation ARENA, et les métaheuristiques ont été simulées par Java sur le même modèle.

Les résultats obtenus ont montré que toutes les métaheuristiques ont nettement amélioré le taux de production, le taux d'utilisation des différentes machines et le taux d'utilisation du système de transport, pour un système de production saturé et même en présence de pannes.

**Abstract** - The job shop scheduling problem is a very known class of Np-hard optimization problems. Because of their complexity, several heuristics and metaheuristics were proposed to solve it. This paper presents the results of a simulation study of a job shop with routing flexibility.

The objective in this study is to compare some metaheuristics performances (Ant colony, genetic algorithms, simulated annealing, taboo search, particle swarm and electromagnetism) with methods of selection of alternative routing in real time: DMM (Dissimilarity Maximization Method) and modified DMM in order to have an idea on the effectiveness of these metaheuristics and select the most effective.

To validate the results of this study, we simulated DMM and modified DMM on a Job shop model using ARENA and metaheuristics were simulated using Java on the same model.

The obtained results showed that all metaheuristics clearly improved the production rate, the utilization ratio of the various machines and the utilization ratio of the material handling system, for overloaded production systems and even in the presence of breakdowns

**Mots clés** – Job shop, Routage alternatif, Règle de sélection de routage, Métaheuristiques, Simulation.

**Keywords** – Job shop, Alternative routing, Routing selection rule, Metaheuristics, Simulation

---

## 1 INTRODUCTION

L'ordonnancement de la fabrication d'un ensemble de produits, c'est répartir leurs réalisations dans le temps et dans l'espace [Doumeingts et al, 1983]. Ceci signifie : la détermination de la séquence d'opérations dans le temps et la répartition des ressources de production (à capacité limitée) aux différentes opérations prévues.

La définition classique et traditionnelle dit que l'ordonnancement est une activité faite en temps différé où les opérations de production sont ordonnancées avant le début de la production.

Plusieurs chercheurs proposent différentes méthodes pour fournir une flexibilité à l'ordonnancement en temps différé afin d'augmenter les performances des systèmes [Saygin and Kilick, 1996] [Saygin et al, 2001].

Mais 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. Donc afin de prendre en compte ces changements les décisions d'ordonnancement doivent être prises en temps réel pour améliorer les performances du système de production.

Le manque de méthodes d'ordonnancement des FMS en temps réel, qui utilisent la flexibilité des opérations et des routages sont la force motrice de règles de sélection des routages alternatifs en temps réel DMM [Saygin et al, 2001] et DMM modifiée [Hassam and Sari, 2007]

Les ateliers flexibles avec des ressources limités, des pannes de machines aléatoires ou des critères de production multiples ont un aspect qui explique que les problèmes d'ordonnancement dans ces systèmes sont généralement de type NP complet, et il n'existe pas de méthodes universelles permettant de résoudre efficacement tous les cas [Garey and

Johson, 1979]. C'est pour cela, que pour ce genre de système, il faut développer des algorithmes efficaces.

Les métaheuristiques sont des algorithmes de type stochastique visant à résoudre une large gamme de problèmes d'optimisation difficile, pour lesquels on ne connaît pas de méthodes classiques plus efficaces. Souvent inspirées d'analogies avec la réalité comme la physique (recuit simulé, diffusion simulée,...) la biologie (algorithmes évolutionnaires, recherche tabou,...) et l'éthologie (colonies de fourmis, essais particuliers...). Elles sont généralement conçues au départ pour des problèmes discrets, mais peuvent s'adapter aux autres types de problèmes.

Dans cet article nous nous intéressons à certaines métaheuristiques (les colonies de fourmis, les algorithmes génétiques, le recuit simulé, la recherche tabou, les essais particuliers, l'électromagnétisme) et nous allons présenter une étude comparative entre ces métaheuristiques et les règles DMM et DMM modifiée.

Le plan de cet article s'articule autour de trois points. Dans le premier nous définissons le contexte du travail présenté. Dans le deuxième point, nous définissons les règles DMM, DMM modifiée et le mode de fonctionnement des algorithmes des métaheuristiques étudiés. La dernière partie est consacrée aux résultats et interprétations.

## 2 PRESENTATION DU MODELE FMS ETUDIEE

Pour faire la comparaison entre les deux règles DMM et DMM modifiée, nous avons étudié et simulé un système flexible de production déjà utilisé dans la littérature [Saygin et al, 2001].

Ce système contient sept machines et deux stations : une station de chargement et une de déchargement. Six types de pièces différentes sont traités dans le système. Chaque machine comporte une file d'attente d'entrée et une file d'attente de sortie, la station de chargement contient aussi une file d'attente d'entrée.

La configuration du système flexible de production est donnée dans la figure 1. Les routages alternatifs et les temps de traitement de chaque type de pièce sont présentés dans le tableau 1.

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

- Les routages alternatifs de 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.

- Le temps de traitement d'une opération est le même sur les machines alternatives identifiées pour cette opération.

- Chaque machine peut traiter une seule pièce à la fois.

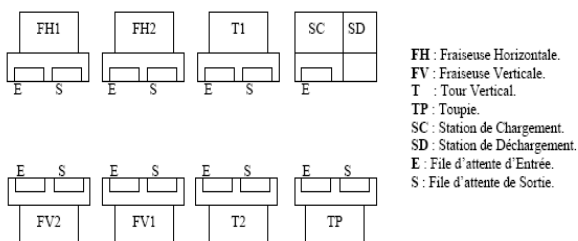


Figure 1. configuration du modèle FMS étudié [Saygin et al, 2001].

Tableau 1. Routages alternatifs et temps de traitement des pièces [Saygin et al, 2001]

Types de pièces et taux d'arrivée	Routage et (durées opératoires (min))
A 17%	SC – T1 (30) – FV1 (20) – SD
	SC – T1 (30) – FV2 (20) – SD
	SC – T2 (30) – FV1 (20) – SD
	SC – T2 (30) – FV2 (20) – SD
B 17%	SC – T1 (20) – TP (1) – FV1 (15)–SD
	SC – T1 (20) – TP (1) – FV2 (15)– SD
	SC – T2 (20) – TP (1) – FV1 (15) – SD
	SC – T2 (20) – TP (1) – FV2 (15) – SD
C 17%	SC – T1 (40) – FV1 (25) – SD
	SC – T1 (40) – FV2 (25) – SD
	SC – T2 (40) – FV1 (25) – SD
	SC – T2 (40) – FV2 (25) – SD
D 21%	SC – T1 (40) – TP (1) – T1 (20) – FH1 (35)–SD
	SC – T1 (40) – TP (1) – T1 (20) – FH2 (35)–SD
	SC – T1 (40) – TP (1) – T2 (20) – FH1 (35)–SD
	SC – T1 (40) – TP (1) – T2 (20) – FH2 (35)–SD
	SC – T2 (40) – TP (1) – T1 (20) – FH1 (35)–SD
	SC – T2 (40) – TP (1) – T1 (20) – FH2 (35)–SD
	SC – T2 (40) – TP (1) – T2 (20) – FH1 (35)–SD
	SC – T2 (40) – TP (1) – T2 (20) – FH2 (35)–SD
E 20%	SC – T1 (25) – TP (1) – T1 (35) – FH1 (50)–SD
	SC – T1 (25) – TP (1) – T1 (35) – FH2 (50)–SD
	SC – T1 (25) – TP (1) – T2 (35) – FH1 (50)–SD
	SC – T1 (25) – TP (1) – T2 (35) – FH2 (50)–SD
	SC – T2 (25) – TP (1) – T1 (35) – FH1 (50)–SD
	SC – T2 (25) – TP (1) – T1 (35) – FH2 (50)–SD
	SC – T2 (25) – TP (1) – T2 (35) – FH1 (50)–SD
	SC – T2 (25) – TP (1) – T2 (35) – FH2 (50)–SD
F 8%	SC –FH1 (40) – SD
	SC –FH2 (40) – SD

## 3 META HEURISTIQUES ET METHODES SIMULEES

Dans cette section, nous allons définir les méthodes DMM et DMM modifiée qui sont utilisées pour la sélection de routages alternatifs dans un FMS et les métaheuristiques étudiées (recuit simulé, algorithmes génétiques, recherche tabou, colonies de fourmis, essaims particuliers et électromagnétisme).

### 3.1 Dissimilarity Maximization Method

DMM [Saygin et al, 2001] a été développée pour le but de réduire la congestion dans un système. Le concept de cette méthode est basé sur l'objectif de la maximisation des dissimilarités entre les routages alternatifs. DMM utilise un coefficient de dissimilarité qui est basé sur les types de machines dans le routage. Il sélectionne un routage pour une pièce donc la dissimilarité cumulée est maximisée.

### 3.2 Modified dissimilarity Maximization Method [Hassam and Sari, 2007]

Cette méthode est aussi utilisée dans la sélection des routages alternatifs en temps réel. Pour la règle DMM, après la sélection d'un routage alternatif pour une pièce, ce routage ne peut pas être utilisé par une autre pièce tant que la première n'a pas quitté le système. La DMM modifiée consiste à garder le même principe mais avec l'assignation de plusieurs pièces à un même routage. Donc si tous les routages sont occupés la prochaine pièce sera transférée au routage dont la file d'attente de la première machine contient une place libre.

### 3.3 Les colonies de fourmis (ACO)

Cette méta-heuristique a été introduite pour la première fois par Marco Dorigo [Dorigo, 1992] [Colorni et al, 1992] et a été inspirée par des études du comportement de fourmis réelles pour résoudre naturellement des problèmes relativement complexes.

Le premier algorithme de cette métaheuristique a été appliqué pour résoudre le problème du voyageur de commerce, le principe de cet algorithme est simple. Lorsqu' une fourmi  $k$  se déplace de la ville  $i$  à la ville  $j$ , elle laisse une trace sur le chemin. De plus, elle choisit la prochaine ville à visiter à l'aide d'une probabilité  $P_{ij}^k$  basée sur un compromis entre l'intensité de la trace  $\Gamma_{ij}^k$  et la visibilité  $\eta_{ij}$  qui représente l'inverse de la distance entre  $i$  et  $j$ . ( $d_{ij}$ ). L'importance relative des deux éléments est contrôlée par deux coefficients  $\alpha$  et  $\beta$ .

Chaque fourmi  $k$  possède une forme de mémoire tabou $_k$ , lui rappelant la liste ordonnée des villes déjà visitées afin d'obliger celle-ci à former une solution admissible.

Après un tour complet, chaque fourmi laisse une certaine quantité de phéromone  $\Delta \Gamma_{ij}^k$  qui dépend de la qualité de la solution trouvée sur l'ensemble de son parcours.

### 3.4 Le recuit simulé (SA)

La méthode de recuit simulée a été conçue par S.Kirkpatrick, C.D Gellat, M.P Vechi en 1983 [Kirkpatrick et al, 1983]. C'est une métaheuristique inspirée par un processus employé en métallurgie pour obtenir un état solide bien ordonné avec énergie minimale appelée le processus de recuit.

Cette technique consiste à porter le matériau à haute température  $t$  puis abaisser lentement celle-ci. Cette méthode d'optimisation est basée sur les travaux de N. Metropolis [Metropolis, 1953] qui permet de décrire le comportement d'un système dans l'équilibre thermodynamique à une certaine température. Cette technique transporte le procédé de recuit à la résolution d'un problème d'optimisation: la fonction objectif à minimiser étant l'énergie  $E$  du matériau. La température  $T$  est également introduite, qui est dans ce cas un paramètre de contrôle de l'algorithme.

A partir d'un état initial à une température  $T$ , on génère une autre solution par une modification élémentaire d'une manière aléatoire. Si cette solution améliore la fonction objectif, cette dernière est automatiquement acceptée. Si elle dégrade la fonction objectif elle peut être acceptée avec une probabilité  $\exp(-\Delta E)$  où  $\Delta E$  est la variation de la fonction objectif, une fois que l'équilibre thermodynamique est atteint, on abaisse la température légèrement avant d'effectuer une nouvelle itération.

### 3.5 Les essaims particulaires (PSO)

PSO est une métaheuristique proposée par Kennedy et Eberhart en 1995[Eberhart and Kennedy, 1995]. Elle est basée sur la métaphore des interactions et communications sociales.

Au début de l'algorithme, un essaim est réparti au hasard dans l'espace de recherche, chaque particule ayant également une vitesse aléatoire. Ensuite, à chaque pas de temps :

- chaque particule est capable d'évaluer la qualité de sa position et de garder en mémoire sa meilleure performance, c'est-à-dire la meilleure position qu'elle a atteinte jusqu'ici (qui peut en fait être parfois la position courante) et sa qualité (la valeur en cette position de la fonction à optimiser).

- chaque particule est capable d'interroger un certain nombre de ses voisines et d'obtenir de chacune d'entre elles sa propre meilleure performance (et la qualité afférente).

- A chaque pas de temps, chaque particule choisit la meilleure des meilleures performances dont elle a connaissance, modifie

sa vitesse en fonction de cette information et de ses propres données et se déplace en conséquence [Clerc et Siarry, 2004].

A chaque itération, La vitesse et la position sont mise à jour suivant deux forces best local et global. La première l'attire au best local qui est la position qui a donné la meilleure fitness pour la particule (i.e c'est la position la plus proche de l'objectif que cette particule a pu atteindre). L'autre best est le "Global Best", c'est la meilleure position trouvée par la particule et ses voisines, comme suit :

$$V_i(t) = V_i(t-1) + c_1 r_1 (p_i + x_i(t-1)) + c_2 r_2 (p_g + x_i(t-1))$$
$$X_i(t) = X_i(t-1) + V_i(t)$$

Où  $v_i(t)$  est la vitesse de la particule  $i$  à l'itération  $t$ .  $X_j(t)$  représente la position de la particule  $j$  à l'itération  $t$ . Le vecteur  $p_i$  est la mémoire de la particule  $i$  à la génération courante, et le vecteur  $p_g$  est la meilleure solution trouvée par l'essaim. Le coefficient cognitif  $c_1$  et le coefficient social  $c_2$  sont connus. Les coefficients d'accélération  $r_1$  et  $r_2$  sont deux nombres aléatoires avec une distribution uniforme. À l'origine, PSO a été conçu pour résoudre les problèmes d'optimisation continus, mais elle peut être adaptée aux problèmes discrets comme notre problème de sélection de routage, où les deux équations précédentes ont été remplacées par l'équation suivante proposée par Pan et al [Pan et al, 2005] :

$$X_i(t) = c_2 \oplus F_3(c_1 \oplus F_2(w \oplus F_1(X_i(t-1)), p_i(t-1)), G(t-1))$$

L'équation précédente se compose de trois composants: Le premier composant est la vitesse de la particule.  $F_1$  représente un opérateur qui modifie le routage de quelques pièces avec une probabilité  $w$ , un nombre aléatoire uniforme  $r$  est généré entre 0 et 1. Si  $r$  est inférieur à  $w$  alors  $F_1$  est appliqué pour produire une permutation perturbée de la particule, sinon la permutation courante est gardée. De la même manière, le deuxième composant qui est la partie cognitive de la particule et le troisième composant qui est la partie sociale de la particule ont été modifiés par les opérateurs  $F_2$  et  $F_3$  représentant les croisements avec les probabilités  $c_1$  et  $c_2$ .

### 3.6 Les algorithmes génétiques (GA)

Les algorithmes génétiques ont été proposés par Holland [Holland, 1975]. Ils ont été inspirés des principes de la génétique naturelle et de la théorie d'évolution (La présence ou l'absence des gènes et de leur ordre dans le chromosome décident des caractéristiques d'une espèce. Différents traitements sont passés d'une génération à la prochaine à travers différents processus biologiques qui déterminent la structure génétique...)

Dans un GA, chaque solution est stockée dans un chromosome artificiel représenté par un code. Chacun de ces chromosomes est défini par deux caractéristiques. Le premier est leur génotype, qui est la chaîne des symboles qui définit le chromosome. Il s'appelle comme ceci en raison de l'analogie avec un ordre génétique dans la biologie. Le second est le phénotype, le phénotype est une solution du problème dans une représentation naturelle.

Pour chacun des chromosomes, les paramètres sont décodés et évalués par la fonction fitness pour déterminer la qualité du phénotype.

De nouveaux candidats sont produits d'un ensemble de population existant en appliquant des opérateurs artificiels génétiques choisis (croisement et mutation) [Goldberg, 1989].

Le croisement est exécuté en prenant deux génotypes, puis on choisit un endroit le long de la chaîne, on coupe chacun d'eux à cet endroit et on relie la partie droite d'une chaîne à la partie gauche de l'autre et vice versa. Ceci produit deux nouveaux chromosomes, qui sont une combinaison des deux parents. La reproduction consiste en le choix des chromosomes de la nouvelle génération à choisir entre enfants et parents de la génération courante. La mutation est faite en choisissant aléatoirement deux bits du même chromosome et en les permutant.

### 3.7 Recherche tabou (TS)

La méthode de recherche tabou, a été formalisée en 1986 par F. Glover [Glover and Laguana, 1997]. Sa principale particularité tient dans la mise en œuvre de mécanismes inspirés de la mémoire humaine.

Le principe de la Recherche Tabou est de choisir à chaque itération la meilleure solution  $s' \in N(s)$ , même si  $f(s') > f(s)$ . Lorsqu'on atteint un minimum local  $s$  par rapport au voisinage  $N$ , la Recherche Tabou va donc se déplacer vers une solution  $s'$  plus mauvaise que  $s$ . Le danger est alors de revenir à  $s$  immédiatement si  $s \in N(s')$  puisque  $s$  est meilleure que  $s'$ . Pour éviter de tourner ainsi en rond, on crée une liste T qui mémorise les dernières solutions visitées et interdit tout déplacement vers une solution de cette liste. Cette liste T est appelée liste taboue, Les solutions ne demeurent dans T que pour un nombre limité d'itérations. La liste T est donc une mémoire à court terme. Si une solution  $s'$  est dans T on dit que  $s'$  est une solution taboue. De même, tout mouvement qui nous mène de la solution courante à une solution de T est appelé mouvement tabou.

### 3.8 Electromagnétisme (EM)

Electromagnétisme est une métaheuristique basée sur une population de solution, elle a été proposée par S. Birbil et al. [Birbil and Fang, 2003] pour résoudre efficacement des problèmes d'optimisation difficile. Elle a été inspirée d'une analogie avec le mécanisme attraction-répulsion de la théorie d'électromagnétisme.

Dans cette approche la charge de chaque point est relative à la valeur de la fonction objectif, ce que nous essayons d'optimiser. Cette charge détermine également l'attraction ou la répulsion du point.

D'ailleurs, la force électrostatique entre deux points est directement proportionnelle aux charges de ces points et inversement proportionnelle au carré de la distance entre ces points. La charge fixe de chaque particule est donnée comme suit :

$$q_i = \exp\left(-n * (f(x_i) - f(x_{best})) / \left(\sum (f(x_i) - f(x_{best}))\right)\right), \forall i$$

Où  $q_i$  est la charge de particule  $i$ ,  $f(x_i)$ ,  $f(x_{best})$ , et  $f(x_k)$  sont les valeurs des fonctions objectifs des particules  $i$ , la meilleure solution, et la particule  $k$ , tandis que  $n$  est la dimension de problème. La qualité de solution ou la charge de chaque particule détermine l'effet d'attraction et de répulsion dans la population. Une meilleure solution encourage d'autres particules à converger aux vallées attrayantes. Tandis qu'une mauvaise solution décourage les autres particules pour se déplacer vers sa région. Ces particules se déplacent avec toute la force et ainsi des solutions diversifiées sont produites. La formulation suivante est la force de la particule  $i$ .

$$F_i = \sum \left\{ \begin{array}{l} (x_j - x_i) * \frac{(q_i * q_j)}{\|x_j - x_i\|^2} : f(x_j) < f(x_i) \\ (x_i - x_j) * \frac{(q_i * q_j)}{\|x_j - x_i\|^2} : f(x_j) \geq f(x_i) \end{array} \right\}, \forall i$$

Le générique pseudo-code de cette métaheuristique est le suivant:

1. initialise ()
2. while (critère d'arrêt non atteint) do
3. recherche local ()
4. calculer la force total F ()
5. déplacer la particule par F ()
6. Evaluer les particules ()
7. End While

## 4 RESULTATS ET INTERPRETATIONS

Il a été montré dans [Hassam and Sari, 2007] [Ghomri et Sari, 2007] que les méthodes DMM et DMM modifiée présentent les meilleurs résultats pour la sélection alternative de routage lorsque comparés avec d'autres règles classiques de sélection de routage.

Pour cela nous avons utilisé les méthodes DMM et DMM modifiée comme base de comparaison pour évaluer les performances des métaheuristicues étudiées (les colonies de fourmis, les algorithmes génétiques, le recuit simulé, la recherche tabou, les essais particuliers et l'électromagnétisme).

Afin de montrer les améliorations apportées par les métaheuristicues étudiées nous avons réalisé plusieurs études en simulation avec des variations sur les critères du système étudié.

Le but de notre travail est d'évaluer les performances des métaheuristicues lorsque le système est en état de saturation. C'est pour ça que les résultats que nous allons présenter reposent sur deux critères principaux :

- Le taux de création des pièces.
- La taille des files d'attente d'entrée et de sortie des stations.

Dans cette section nous avons montré les résultats de simulation des deux méthodes, l'exécution des programmes et les interprétations des résultats.

Les méthodes et les algorithmes ont été simulés pendant 20000 heures avec un régime transitoire de 3000 heures sur un PC (Core (TM) 2Duo CPU avec 2.2 GHZ et 1 GO de RAM).

### 4.1 Etude comparative sans introduction de pannes

#### 4.1.1 Taux de production

Le taux de sortie des pièces est calculé en divisant le nombre de pièces sorties du système sur le nombre de pièces créées.

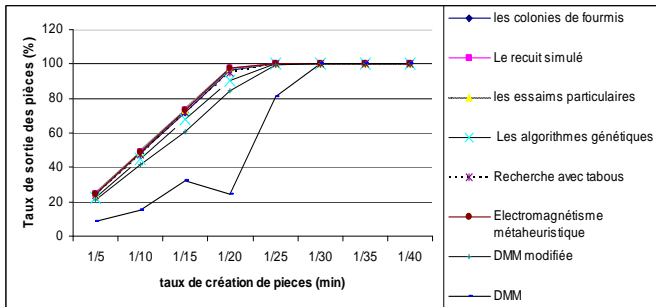
Le tableau 2 montre que pour un taux de création des pièces important les résultats obtenus par les métaheuristicues étudiées sont meilleurs que celles des méthodes DMM modifiée et DMM et qu'en dessous du taux de création 1/25 le taux de production est pratiquement le même pour toutes les méthodes.

Le tableau 2 nous montre que le recuit simulé, essaim de particules et électromagnétisme sont plus efficaces que les

autres métaheuristiques et les méthodes DMM modifiée (MDMM) et DMM.

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

Taux de création	1/5	1/10	1/15	1/20	1/25	1/30	1/35	1/40
ACO	24.0	48.1	72.1	96.1	99.9	99.9	99.9	99.9
SA	24.3	48.7	73.1	97.7	99.9	99.9	99.9	99.9
PSO	24.3	48.8	73.0	97.5	99.9	99.9	99.9	99.9
GA	22.3	45.0	67.5	90.0	99.9	99.9	99.9	99.9
TS	23.8	47.6	71.5	95.2	99.9	99.9	99.9	99.9
EM	24.2	48.6	73.1	97.3	99.9	99.9	99.9	99.9
MDMM	21.1	41.6	60.7	84.4	99.7	99.9	99.9	99.9
DMM	8.88	15.4	32.0	24.6	81.4	99.9	99.9	99.9

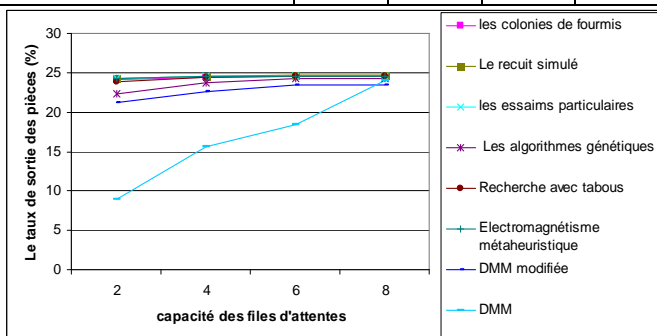


**Figure 2. Taux de sortie des pièces pour une capacité de file d'attente = 2.**

La figure 3 nous montre qu'en variant les capacités des files d'attente et en maintenant le temps de création fixe, le recuit simulé, les essais de particules et électromagnétisme métaheuristique restent toujours plus efficaces que les autres métaheuristiques et les méthodes DMM et DMM modifiée

**Tableau 3. Taux de sortie des pièces pour taux de création=1/5 (1/min).**

Capacité de file d'attente	2	4	6	8
ACO	24	24.39	24.55	24.59
SA	24.3	24.62	24.66	24.71
PSO	24.3	24.51	24.61	24.62
GA	22.3	23.76	24.21	24.25
TS	23.8	24.46	24.52	24.55
EM	24.2	24.56	24.58	24.59
MDMM	21.1	22.58	23.44	23.51
DMM	8.8	15.67	18.42	24.01



**Figure 3. Taux de sortie des pièces pour taux de création=1/5 (1/min).**

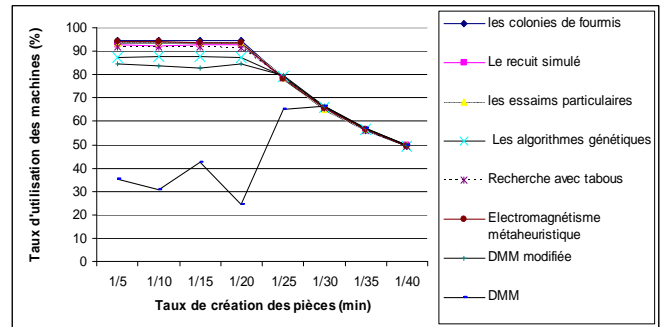
#### 4.1.2 Taux d'utilisation des machines

Le taux d'utilisation des machines est un critère très important dans la mesure des performances d'un système de production. Le tableau 4 montre que le taux d'utilisation pour les machines T<sub>1</sub> et T<sub>2</sub> est plus important pour les métaheuristiques étudiées que pour DMM et DMM modifiée pour un taux de création

important (supérieur à 1/25), ce tableau montre aussi que l'électromagnétisme et les colonies de fourmis sont les plus efficaces.

**Tableau 4. Taux d'utilisation des machines T<sub>1</sub> et T<sub>2</sub> pour une capacité de file d'attente =2.**

Taux de création	1/5	1/10	1/15	1/20	1/25	1/30	1/35	1/40
ACO	94.7	94.7	94.6	94.6	78.7	65.6	56.2	49.2
SA	92.6	92.5	92.9	92.9	78.2	65.7	56.6	49.8
PSO	93.7	93.7	93.6	93.7	78.6	65.2	56.5	49.6
GA	87.1	87.7	87.6	87.4	79.0	66.0	56.6	49.3
TS	91.8	91.7	91.8	91.6	78.7	65.6	56.2	49.2
EM	94.0	94.1	93.4	93.4	78.4	65.4	56.3	49.5
MDMM	84.7	83.5	82.7	84.5	79.8	66.5	56.9	49.9
DMM	35.5	30.9	42.5	24.6	65.0	66.5	57.0	49.9



**Figure 4. Taux d'utilisation des machines T<sub>1</sub> et T<sub>2</sub> pour une capacité de file d'attente = 2.**

#### 4.1.3 Taux d'utilisation de l'AGV

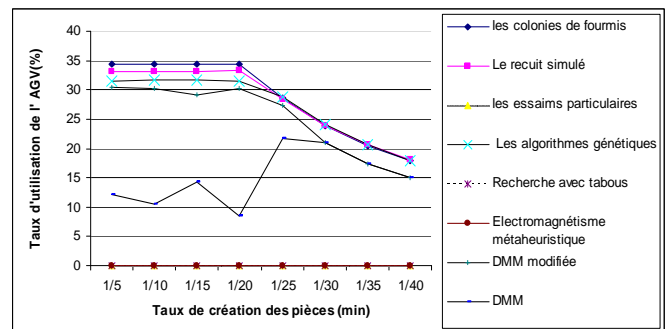
Dans cette partie nous avons illustré le taux d'utilisation de l'AGV,

Le tableau 5 montre que l'utilisation de l'AGV est plus importante pour les métaheuristiques que la méthode DMM et DMM modifiée.

Ce tableau montre aussi qu'il y'a certaine amélioration pour certains métaheuristiques comme les colonies de fourmis si on les compare avec autres comme les algorithmes génétiques.

**Tableau 5. Taux d'utilisation de l'AGV pour une capacité de file d'attente =2.**

Taux de création	1/5	1/10	1/15	1/20	1/25	1/30	1/35	1/40
ACO	34.3	34.3	34.3	34.3	28.5	23.8	20.4	17.8
SA	33.2	33.1	33.1	33.2	28.3	23.8	20.6	18.1
PSO	33.7	33.7	33.6	33.7	28.5	23.6	20.5	18.0
GA	31.4	31.6	31.6	31.5	28.7	24.0	20.5	17.9
TS	33.0	33.0	33.0	32.9	28.5	23.8	20.4	17.8
EM	33.9	33.9	33.5	33.5	28.4	23.7	20.4	18.0
MDMM	30.4	30.1	29.1	30.2	27.3	21.0	17.4	14.9
DMM	12.0	10.5	14.3	8.43	21.6	20.9	17.4	15.0

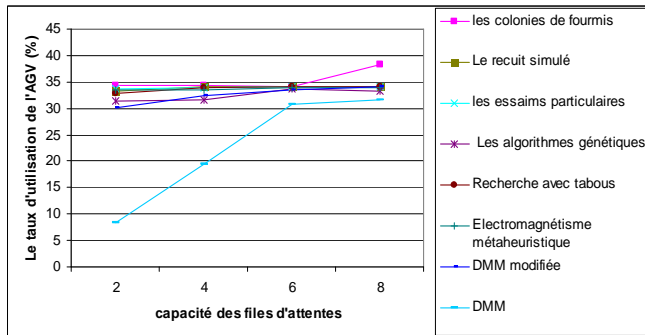


**Figure 5. Taux d'utilisation de l'AGV pour une capacité de file d'attente = 2.**

La figure 6 nous montre que pour un système saturé le taux d'utilisation de l'AGV est plus important pour les colonies de fourmis même si on augmente les tailles des files d'attente.

**Tableau 6. Taux d'utilisation de l'AGV pour un taux de création de pièces= 1/20.**

Capacité de file d'attente	2	4	6	8
ACO	34.3	34.25	34.22	38.2
SA	33.2	34.03	34.02	34.07
PSO	33.7	33.92	34.03	34.06
GA	31.5	31.57	33.61	33.26
TS	32.9	33.89	34.07	34.11
EM	33.5	33.59	33.98	34.01
MDMM	30.2	32.46	33.53	34.03
DMM	8.43	19.56	30.67	31.69



**Figure 6. Taux d'utilisation de l'AGV pour un taux de création de pièces= 1/20.**

#### 4.2 Etude comparative avec introduction de pannes

Dans cette partie nous allons donner les résultats et les interprétations de l'étude comparative entre les métaheuristiques et les règles DMM et DMM modifiée présentés déjà dans le paragraphe précédent avec l'introduction de pannes. Les pannes sont introduites sur les ressources toutes les 100 heures. La panne va durer 2 heures.

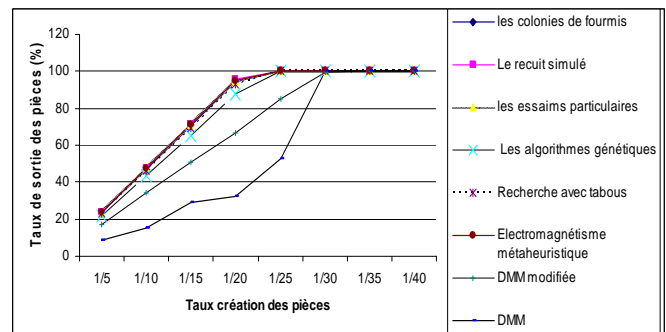
##### 4.2.1 Taux de production

La simulation du système en introduisant des pannes nous montre que les taux de production pour les métaheuristiques étudiées sont nettement supérieurs à ceux des règles DMM et DMM modifiée pour un taux de création supérieur ou égale à 1/25.

De ces résultats nous pouvons constater que le taux de production en utilisant PSO et EM est plus important que celui obtenu en utilisant les autres métaheuristiques et les règles DMM et DMM modifiée pour des taux de création élevés. Donc le taux de production d'un système saturé est plus élevé pour ces deux métaheuristiques même avec la présence de pannes.

**Tableau 7. Taux de sortie des pièces pour une capacité de file d'attente =2.**

Taux de création	1/5	1/10	1/15	1/20	1/25	1/30	1/35	1/40
ACO	23.5	47.1	70.6	94.2	99.9	99.9	99.9	99.9
SA	23.7	47.5	71.3	95.5	99.9	99.9	99.9	99.9
PSO	23.7	47.5	71.5	95.1	99.9	99.9	99.9	99.9
GA	21.6	43.2	65.4	87.9	99.9	99.9	99.9	99.9
TS	23.1	46.2	69.5	92.6	99.9	99.9	99.9	99.9
EM	23.7	47.4	71.4	95.1	99.9	99.9	99.9	99.9
MDMM	17.0	34.2	50.9	66.7	85.0	99.6	99.9	99.9
DMM	8.55	15.4	29.2	32.4	53.0	99.9	99.9	99.9

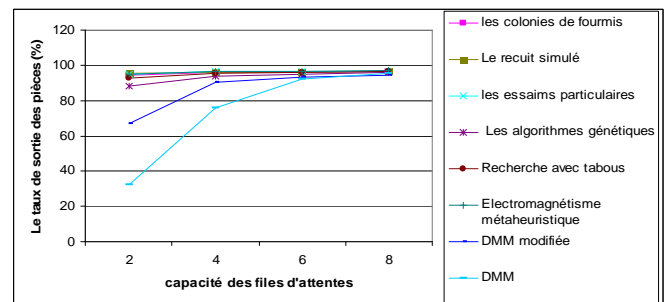


**Figure 7. Taux de sortie des pièces pour une capacité de file d'attente = 2.**

La figure 8 nous montre que si on augmente les capacités des files d'attente le taux de production est toujours plus important pour les essais de particules et électromagnétisme métaheuristique que pour les autres métaheuristiques et les règles DMM et DMM modifiée.

**Tableau 8. Taux de sortie des pièces pour un taux de création = 1/20.**

Capacité de file d'attente	2	4	6	8
ACO	94.23	95.73	96.23	96.43
SA	95.56	95.9	96.1	96.32
PSO	95.12	96.6	96.62	96.81
GA	87.91	93.66	94.62	96.12
TS	92.64	95.44	96.09	96.43
EM	95.15	96.6	96.63	96.87
MDMM	66.7	90.6	93.11	94.24
DMM	32.46	76.07	91.94	95.69



**Figure 8. Taux de sortie des pièces pour un taux de création = 1/20.**

##### 4.2.2 Taux d'utilisation des machines

Comme nous pouvons bien le remarquer dans le tableau 9, avec un taux de création de pièces supérieur à 1/25, le taux d'utilisation des machines est toujours plus important en utilisant les métaheuristiques.

Ce tableau montre aussi que certaines métaheuristiques comme les colonies de fourmis sont toujours meilleurs que d'autres comme les algorithmes génétiques.

**Tableau 9. Taux d'utilisation des machines T<sub>1</sub> et T<sub>2</sub> pour une capacité de file d'attente =2.**

Taux de création	1/5	1/10	1/15	1/20	1/25	1/30	1/35	1/40
ACO	92.7	92.7	92.7	92.7	78.7	65.6	56.2	49.2
SA	90.6	90.	90.5	90.7	78.4	65.7	56.6	49.7
PSO	92.2	91.6	91.7	91.6	77.6	65.7	56.4	49.6
GA	84.2	84.1	84.7	85.4	79.1	65.9	56.5	49.4
TS	89.2	89.1	89.2	89.0	78.7	65.6	56.1	49.2
EM	92	91.6	91.8	92.2	78.2	65.3	56.4	49.6
MDMM	78.7	78.5	78.1	77.7	77.1	66.7	56.9	49.8
DMM	34.8	31.2	39.3	33	42.7	66.8	57.2	49.9

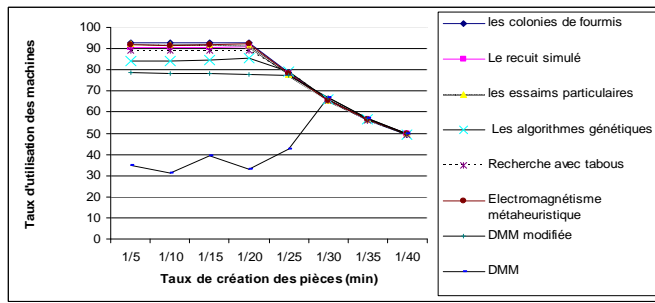


Figure 9. Taux d'utilisation des machines  $T_1$  et  $T_2$  pour une capacité de file d'attente = 2.

#### 4.2.3 Taux d'utilisation de l'AGV

Le tableau 10 nous montre que pour un système saturé le taux d'utilisation de l'AGV est plus important pour les colonies de fourmis même avec la présence de pannes.

Tableau 10. Taux d'utilisation de l'AGV pour une capacité de file d'attente =2.

Taux de création	1/5	1/10	1/15	1/20	1/25	1/30	1/35	1/40
ACO	33.6	33.6	33.6	33.6	28.5	23.8	20.4	17.8
SA	32.4	32.4	32.4	32.4	28.4	23.8	20.6	18.1
PSO	33.2	33.0	33.0	32.9	28	23.8	20.4	18.0
GA	30.4	30.4	30.6	30.8	28.7	23.9	20.5	17.9
TS	32.0	32.0	32.0	32.0	28.5	23.8	20.3	17.8
EM	33.1	32.9	33.0	33.2	28.3	23.6	20.5	18.0
MDMM	24.5	24.6	24.5	24.0	24.2	21.1	17.5	14.9
DMM	11.5	10.4	12.9	10.9	14.2	21.0	17.6	15.0

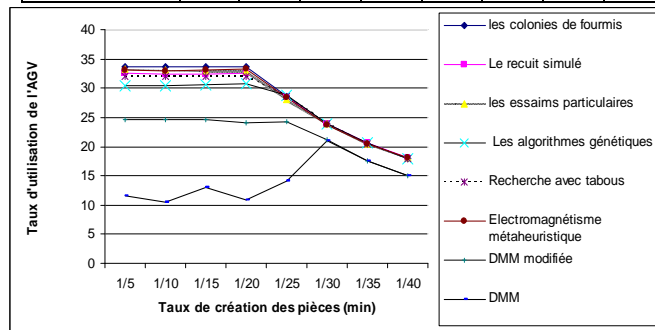


Figure 10. Taux d'utilisation de l'AGV pour une capacité de file d'attente = 2.

La figure 11 nous montre que pour un système saturé et de petites files d'attentes (inférieur à 6) le taux d'utilisation de l'AGV est plus important pour les colonies de fourmis même avec la présence de pannes. Si on augmente les files d'attentes la règle DMM modifiée est la meilleure.

Tableau 11. Taux d'utilisation de l'AGV pour un taux de production = 1/20.

Capacité de file d'attente	2	4	6	8
ACO	33.65	33.59	33.57	33.55
SA	32.49	33.56	33.43	33.44
PSO	32.96	33.28	33.34	33.36
GA	30.85	32.42	32.81	33.04
TS	32.03	33.16	33.37	33.45
EM	33.26	33.38	33.32	33.41
MDMM	24.07	32.46	33.58	34.08
DMM	10.94	25.28	30.66	31.68

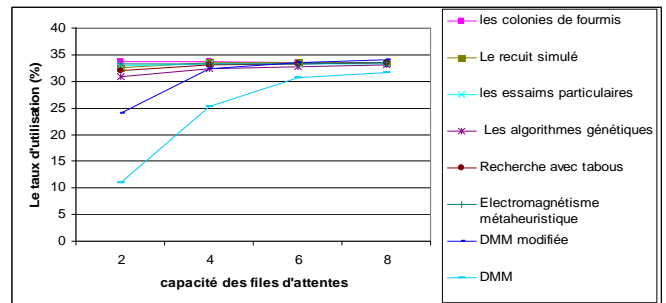


Figure 11. Taux d'utilisation de l'AGV pour un taux de production = 1/20.

## 5 CONCLUSION

Dans ce travail, nous nous sommes intéressés à l'ordonnancement des systèmes flexibles de production, aux règles de sélection de routage et aux métaheuristiques. En particulier nous avons étudié les colonies de fourmis, les algorithmes génétiques, le recuit simulé, la recherche tabou, les essais particuliers, l'électromagnétisme et nous avons présenté une étude comparative entre ces métaheuristiques et les méthodes DMM et DMM modifiée qui sont des règles de sélection de routages alternatifs en temps réel dans les FMS afin d'avoir une idée sur l'efficacité de ces techniques.

Pour cela, nous avons simulé DMM et DMM modifiée dans un jobshop à l'aide du logiciel ARENA et les métaheuristiques à l'aide du langage de programmation Java.

Nous avons d'abord commencé par varier le temps d'arrivée (le taux de création) des pièces et garder la capacité des files d'attentes fixes. Puis nous avons fixé le temps de création des pièces et nous avons varié les capacités des files d'attente. Ensuite, nous avons simulé le système avec les deux critères de simulation déjà mentionnés, mais en ajoutant des pannes aux machines du système.

Les résultats obtenus ont montré que les métaheuristiques que nous avons étudiées ont donné de meilleures performances pour le taux de production, le taux d'utilisation des machines et du système de transport pour un système flexible de production saturé surtout pour de petites capacités de files d'attente.

L'efficacité de ces métaheuristiques varie d'un objectif à un autre et d'un critère à un autre par exemple pour le taux de production, les métaheuristiques les plus efficaces sont le recuit simulé, EM et PSO dans le cas sans panne. Tandis que dans le cas avec panne EM et PSO ont donnée les meilleurs résultats. Pour le taux d'utilisation des machines, les colonies de fourmis et l'électromagnétisme sont les meilleures dans le cas sans panne tandis que les colonies de fourmis sont les plus efficaces dans le cas avec panne. Pour le taux d'utilisation de l'AGV on trouve les colonies de fourmis comme meilleure alternative, pour de petites files d'attentes.

Une des perspectives immédiates de ce travail est d'adapter ces techniques pour résoudre le problème en temps réel, et de travailler sur d'autres types d'ateliers afin de pouvoir mieux cerner le problème et éventuellement trouver des conclusions générales.

## 6 REFERENCES

- M. Clerc, et P. Siarry, (2004) Une nouvelle métaheuristique pour l'optimisation difficile: la méthode des essais particuliers, J3eA - Vol. 3 - 7 (2004).
- A. Colomi, , M.Dorigo and V. Maniezzo, (1992). Distributed Optimization by Ant Colonies. In Varela, F. and Bourgine, P., editors, Proceedings of ECAL'91 - First European



Conference on Artificial Life, pages 134–142, Paris, France. Elsevier Publishing.

- M. Dorigo. (1992) Optimization, learning and natural algorithms. Italy, PhD thesis, DEI, Politecnico di Milano.
- G. Doumeingts et al (1983) La gestion de production assistée par ordinateur. Hermès.
- R. C. Eberhart and J. Kennedy (1995) New optimizer using particle swarm theory". Proceedings of the 6th International Symposium on Micro Machine and Human Science, pp. 39–43, Nagoya, Japan.
- M.R Garey and D.S. Johnson, (1979) Computers and intractability a guide of the theory of NP- completeness. W.H. Freeman and company, San Francisco.
- L. Ghomri et Z. Sari (2007) Influence des Contraintes et des Perturbations sur les Performances des Règles de Routage dans un FMS. Conception et Production Intégrées, Rabat
- F. Glover and Manual Laguna, 1997 Tabu Search. Article was principally adapted from the book « Tabu Search » of Glover, Kluwer Academics publishers
- E.E. Goldberg, (1989) Genetic Algorithms in Search, Optimization, and Machine Learning. Addison Wesley, Reading, MA.
- A. Hassam and Z. Sari (2007) Real-Time Selection of Process Plan in Flexible Manufacturing Systems: Simulation Study. International Conference on Industrial and Systems Management.
- J.H Holland (1975) Adaptation in Natural and Artificial Systems. University of Michigan Press, Ann Arbor.
- S.I. Birbil and S.C. Fang (2003) An Electromagnetism like Mechanism for Global Optimization. Journal of Global Optimization vol 25: 263–282.
- S. Kirkpatrick, C. D. Gelatt and M. P. Vecchi (1983) Optimization by Simulated Annealing. Science Number 4598, volume 220, 4598, p .671-680.
- N.Metropolis, A.W. Rosenbluth, M.N. Rosenbluth, A.H. Teller and E. Teller. (1953) Equations of state calculations by fast computing machines. Journal of Chemical Physics, 21:1087–1091.
- QK.Pan, MF .Tasgetiren, YC .Liang (2005) A discrete particle swarm optimization algorithm for the no-wait flowshop scheduling problem with makespan criterion. Proceedings of the international workshop on UK planning and scheduling special interest group, UK PLANSIG2005. City University, London, pp: 31–41.
- C.Saygin and S.E. Kilick (1996) Effect of flexible process plans on performance of flexible manufacturing system. Proceedings of 7th International DAAM Symposium, p.393-394, Vienna.
- C. Saygin, F.F. Chen and J. Singh (2001) Real-Time Manipulation of alternative Routings in Flexible Manufacturing Systems: A simulation Study. International journal of advanced Manufacturing Technology, 18, 755-763.