

METAHEURISTIQUES POUR LA MANIPULATION DE ROUTAGES ALTERNATIFS EN TEMPS REEL DANS UN FMS

M. Souier, A. Hassam, Z. Sari

Laboratoire d'automatique de tlemcen

Faculté des sciences de l'ingénieur, Université Abou Bekr Belkaid, BP 230, Tlemcen 13000, Algérie
m_souier@mail.univ-tlemcen.dz, a_hassam@mail.univ-tlemcen.dz, z_sari@mail.univ-tlemcen.dz

Résumé--Cet article présente certaines métaheuristiques (les colonies de fourmis (ant colony optimization (ACO)), les algorithmes génétiques (genetic algorithm (GA)), le recuit simulé (simulated annealing (SA)), la recherche tabou (taboo search (TS)), les essaims particulaires (particle swarm optimization (PSO)) et l'électromagnétisme (electromagnetism like method (EM))) et compare leurs performances avec les règles de sélection de routages alternatifs en temps réel DMM (Dissimilarity Maximization Method) et DMM modifiée (MDMM), afin d'avoir une idée sur l'efficacité de ces techniques et de sélectionner la plus efficace.

Pour valider les résultats de cette comparaison, nous avons simulé les règles DMM et DMM modifiée sur un FMS (Flexible manufacturing system) modèle en utilisant ARENA, et les métaheuristiques ont été simulées en utilisant Java sur le même modèle, qui se compose de sept machines, une station de chargement, une station de déchargement, et six types de pièces différentes. En raison de l'existence de centres d'usinage identiques dans le système, les types de pièces ont des routages alternatifs. De plus, au niveau de chaque machine, FIFO (First In First Out) est employé comme règle de séquençement.

Les résultats de simulation ont montré l'amélioration des performances des métaheuristiques jugées par le taux de production, le taux d'utilisation des machines et du système de transport pour un système flexible de production saturé, même avec la présence de pannes.

Mots clés--Système de production flexible, Routage alternatif, Règle de sélection de routage, Métaheuristiques, Simulation.

I. INTRODUCTION

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. C'est pour cela, que pour ce genre de systèmes, il faut développer des algorithmes efficaces.

Plusieurs chercheurs proposent différentes méthodes pour fournir une flexibilité à l'ordonnancement en temps différé où les opérations de production sont ordonnancées

avant le début de la production afin d'augmenter les performances des systèmes [1].

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 tirer pleinement partie de la flexibilité offerte par les systèmes flexibles de production (FMS), les décisions d'allocation et d'ordonnancement des opérations et des plans de processus sont généralement prises dynamiquement et à très court terme, en fonction de l'état du système de production (disponibilité des ressources, disponibilité du système de manutention, présence de goulots d'étranglement), des caractéristiques du plan de production (date échues des ordres de fabrication) et des objectifs de production (augmentation du taux de production, réduire les en cours) [2].

La flexibilité de routages est l'une des types de flexibilité étudiées par la communauté scientifique, cette flexibilité offre au système les moyens d'un aiguillage plus souple, de façon à servir les différents segments de procédés libres ou sous engagés.

De nombreux travaux se sont intéressés à l'influence de la flexibilité de routage sur les performances d'un FMS avec et sans pannes de machines, Saygin et Kilic en 1999 [3] ont proposé une plateforme qui intègre le processus de planification flexible et l'ordonnancement prédictif (en temps différé), il ont présenté un concept nommé dissimilarity maximisation method (DMM) pour minimiser la congestion dans un FMS, l'idée de cette règle c'est de maximiser les dissimilarités entre les routages occupés. L'efficacité de cette règle dans la résolution des problèmes de sélection de routage en temps réel a été démontré dans [1] et [4] où elle a surpassé d'autres règles tel que FIFO/FA (first-in first-out/ first available) et equal probability loading (EPL) si chaque machine utilise la règle FIFO, et dans [5] où les trois règles ont été combinée avec d'autres règles de priorité. Dans l'étude de Hassam et Sari [6] sur la règle DMM, Il ont remarqué que pour un taux d'arrivée de pièces important et pour une faible capacité de

files d'attentes, le système de production est saturé et le taux d'utilisation des machines et du transporteur est assez faible, ce qui va influencer sur les performances du système de production. Pour cela ils ont proposé la règle DMM modifiée qui est une modification de la règle DMM qui vise à garder le même principe mais si tous les routages sont sélectionnés par des pièces, la pièce suivante va être acheminée dans le routage où la file d'attente de la première machine, de ce routage, contient au moins une place libre.

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 efficace. 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, essaims particulaires...). Elles sont généralement conçues au départ pour des problèmes discrets, mais peuvent s'adapter aux autres types de problèmes.

Devant le succès rencontré par ces techniques dans le cadre de résolution des autres problèmes NP-difficiles, notre but a été leur adaptation pour résoudre notre problème de sélection de routages alternatifs. Dans [7] nous avons adapté plusieurs métaheuristiques (les colonies de fourmis, les algorithmes génétiques, les essaims de particules, le recuit simulé, la recherche tabou et l'électromagnétisme) pour résoudre le problème de sélection de routages dans un FMS avec un nombre important de pièces mais en temps différé. Dans ce travail, nous avons adapté ces techniques pour résoudre le problème en temps réel.

Le reste de cet article s'articule autour de trois sections. La première est réservée à la présentation du contexte du travail présenté. Dans la deuxième, nous définissons les règles DMM, DMM modifiée et le mode de fonctionnement des algorithmes des métaheuristiques étudiées. La dernière partie est consacrée aux résultats et interprétations.

II. PRESENTATION DU MODEL FMS ETUDIEE

Ce système flexible de production qui a été déjà utilisé dans la littérature [1], 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.

TABLEAU I. ROUTAGES ALTERNATIFS ET TEMPS DE TRAITEMENT DES PIECES [1]

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) – FV2 (15) – SD
C 17%	SC – T1 (40) – FV1 (25) – SD
	SC – T1 (40) – FV2 (25) – SD
	SC – T2 (40) – FV1 (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
F 8%	SC – FH1 (40) – SD
	SC – FH2 (40) – SD

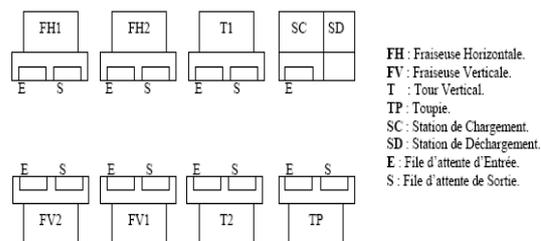


Fig. 1. Configuration du modèle FMS étudiée [1]

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.

III. MÉTAHEURISTIQUES ET METHODS SIMULÉES

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.

A. Méthodes simulées

A. 1. DISSIMILARITY MAXIMIZATION METHOD

DMM [1] 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.

A. 2. MODIFIED DISSIMILARITY MAXIMIZATION METHOD [6]

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 règle 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.

B. Les algorithmes des métaheuristiques

Toutes les métaheuristiques présentent des algorithmes itératifs, donc c'est très difficile de trouver la solution optimale à partir de la première itération ce qui justifie l'échec d'adaptation de ces techniques pour la résolution des problèmes complexes en ligne.

Pour résoudre notre problème en ligne, nous avons pris en considération à chaque itération des critères concernant l'état du système, à chaque fois qu'une nouvelle pièce entre dans la station de chargement. À l'aide des métaheuristiques nous avons essayé d'équilibrer les charges de routages non pas en terme de nombre de pièces mais en terme de temps opératoire (maximiser le produit des charges de routages). Pour ce faire nous avons pris en considération les types des premières pièces existantes (le nombre de ces pièces égale à la taille de la station de chargement) dans la file d'entrée (avant qu'ils passent à la station de chargement) en essayant d'affecter chaque pièce à un routage selon son type. Les routages de ces pièces

sont modifiés à chaque itération selon le principe de chaque métaheuristique et pour chaque routage nous avons calculé la charge en vue de maximiser leur produit.

B. 1. RECUIT SIMULE (SA)

La méthode de recuit simulé a été conçue par Kirkpatrick *et al* [8]. 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 Metropolis *et al* en 1953 [9] 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 objective 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.

L'algorithme de recuit simulé adapté pour résoudre notre problème de sélection de routage se présente comme suit :

n est la capacité des files d'attente.

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

Construire l'état initial (Affecter les n premières pièces à des routages d'une façon aléatoire).

Calculer le produit des charges de routages.

Pour t = 1 à t_{max}

Modifier les routages de certaines pièces parmi les n premières pièces contenues dans la file infinie.

Calculer le produit des charges de routages.

Si la fonction objectif a été améliorée alors cette solution est acceptée

Finsi

Si non générer un nombre aléatoire

Si ce nombre est inférieur ou égale à $\exp(-\Delta E)$:

ΔE est la variation de la fonction objectif alors cette solution est acceptée

Finsi

Finsi

Fin Pour.

Finsi

B. 2. LA RECHERCHE TABOU (TS)

La méthode de recherche tabou, a été formalisée en 1986 par Glover et Laguana [10]. 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

appartient à $N(s')$ 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.

L'algorithme suivant explique comment cette métaheuristique a été appliquée pour résoudre notre problème:

n est la capacité des files d'attente.

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

Construire l'état initial S

Initialiser les paramètres.

Calculer les produits des charges de routages.

Tant que (critère d'arrêt est non atteint)

Pour $t = 1$ à $n_{\text{bre_voisins}}$

Modifier les routages de certaines pièces choisies aléatoirement parmi les n premières pièces de la file infinie avec des mouvements non tabous (modification de S).

Calculer le produit des charges de routages.

Si la fonction objectif (produit des charges de routages) est supérieure à la meilleure solution, alors mettre à jour la meilleure solution (L'élitisme).

Finsi

Fin pour

Soit T le meilleur des voisins.

Insertion de mouvements $T \rightarrow S$ dans la liste tabou.

$S = T$.

Fin tant que

Finsi.

B. 3. LES ALGORITHMES GENETIQUES (GA)

Les algorithmes génétiques ont été proposés par Holland en 1975 [11]. 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.

Avec 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) [12].

L'algorithme suivant est l'algorithme basé sur cette métaheuristique appliqué pour résoudre notre problème :

n est la capacité des files d'attente.

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

Génération d'une population aléatoire.

Tant que (critère d'arrêt est non atteint)

Pour chaque individu

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

Si la fonction objectif est supérieure à la meilleure solution 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

Dans cet algorithme, chaque chromosome artificiel représente les routages choisis des premières pièces par un code. Après chaque évaluation des individus 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 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.

B. 4. LES COLONIES DE FOURMIS (ACO)

Cette métaheuristique a été introduite pour la première fois par Dorigo en 1992 [13] 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 basée sur un compromis entre l'intensité de la trace et la visibilité qui représente l'inverse de la distance entre i et j . (d_{ij})

L'importance relative des deux éléments est contrôlée par deux coefficients α et β . 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 qui dépend de la qualité de la solution trouvée sur l'ensemble de son

parcours.

L'algorithme original a été adapté à notre problème en remplaçant la ville i par la pièce i et la ville j par le routage j .

Pour chaque pièce i , le choix du routage j est basé sur un compromis entre l'intensité de la trace $\tau_{ij}^k(t)$ et la visibilité η_{ij} (dépend du nombre de pièces dans la file d'entrée de la première machine de ce routage et sa charge).

L'importance relative des deux éléments est toujours contrôlée par deux coefficients α et β . Si le nombre total de fourmis est m et la taille de la station de chargement est n , un cycle est réalisé lorsque chacune des m fourmis affecte les n premières pièces de la file infini à des routages j .

Après un tour complet (l'affectation de toutes les n premières pièces de la file infini aux routages par les fourmis), chaque fourmi laisse une certaine quantité de phéromone qui dépend $\Delta \tau_{ij}^k(t)$ de la qualité de la solution trouvée (le produit des charge de routages) sur l'ensemble des routages sélectionnés pour les pièces.

Le pseudo code de l'algorithme est :

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

Pour $t = 1$ à t_{\max}

Pour chaque fourmi $k = 1$ à m

Choisir pour la première pièce de la file infini un routage au hasard suivant son type.

Pour chaque pièce i contenu dans la deuxième place jusqu' au la n^{eme} place de la file infini.

Choisir un routage j , parmi les routages possibles selon une probabilité dépendant de l'intensité de la trace et du nombre de pièces dans la file d'entrée de la première machine de ce routage et sa charge.

Fin Pour

Evaluation de la fonction objectif. (Produit des charges de routages)

Déposer une piste $\Delta \tau_{ij}^k(t)$ sur le trajet $T^k(t)$ (pour chaque routage j choisi pour la pièce i par la fourmis k).

Fin Pour.

Évaporer les pistes et modifier les intensités.

Fin Pour.

Finsi.

B. 5. LES ESSAIS PARTICULAIRES (PSO)

PSO est une métaheuristique proposée par Kennedy et Eberhart en 1995 [14]. Elle est basée sur la métaphore des interactions et communications sociales. Au départ 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 voisins 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 [15].

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 voisins

Bl_i , et B_g sont les bests locaux et globaux.

Le pseudo code de l'algorithme de cette métaheuristique adapté pour résoudre notre problème est le suivant :

S'il y'a une place libre dans la station de chargement alors **Initialiser** la population.

Initialiser les paramètres.

Tant que (critère d'arrêt est non atteint)

Pour chaque particule X_i

Calculer le produit des charges de routages de cette particule.

Si $F(x_i) > F(Bl_i)$ alors : (mise à jour du best local)

$Bl_i = x_i$

Finsi

Si $F(x_i) > F(B_g)$ alors : (mise à jour du best global)

$B_g = x_i$

Finsi

Fin pour

Pour chaque particule X_i

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

Fin pour

Fin tant que

Finsi.

F_1 représente un opérateur qui modifie les routages de quelques pièces parmi les premières pièces de la file infinie 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 même manière, on applique F_2 et F_3 qui représentent les croisements avec les bests locaux et globaux selon des probabilités C_1 et C_2 .

B. 6. L'ELECTROMAGNETISME (EM)

Electromagnétisme est une métaheuristique basée sur une population de solution, elle a été proposée par Birbil *et al* en 2003 [16] 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 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. Le pseudo code suivant explique comment cette métaheuristique a été appliquée pour résoudre notre problème :

S'il y'a une place libre dans la station de chargement alors **Génération** d'une population aléatoire.

Tant que (critère d'arrêt est non atteint)

Pour chaque particule X_i

Recherche locale en modifiant les routages de certains pièces parmi les n premières de la file infini.

Evaluation de la fonction objectif (Produit des charges de routages).

Si la fonction objectif est supérieure à la meilleure solution alors mettre à jour la meilleure solution.

Fin pour

avg = calcAvgObjectiveValues() (calcul de moyen des fonctions objectifs).

Pour $i = 1$ à m (m : la taille de la population)

Si $i \neq \text{best}$ et $f(x_i) > \text{avg}$ alors

$j =$ particule sélectionné.

croisement_uniforme (x_i, x_j).

Finsi

Sinon Si $f(x_i) < \text{avg}$ Alors

Calcul force et déplace (x_i).

Finsi

Fin pour

Fin tant que

Finsi.

IV. RÉSULTATS ET INTERPRÉTATIONS

Afin de montrer les améliorations apportées par les métaheuristiques étudiées lorsque le système est en état de saturation nous avons réalisé plusieurs études en simulation avec des variations sur les critères du système étudié (le taux de création des pièces, la taille des files d'attente d'entrée et de sortie des stations). Pour chaque étude deux scénarios ont été prise en compte avec et sans introduction de pannes. Les pannes sont introduites sur les ressources toutes les 100 heures. La panne va durer 2 heures suivant une loi exponentielle.

Cette section est réservée à la présentation des résultats trouvés et leurs interprétations sur un temps de simulation de 20000 heures et un régime transitoire de 3000 heures, avec 10 réplifications (essais) pour chaque simulation.

A. Taux de production

TABLEAU 2. TAUX DE SORTIE DES PIÈCES POUR UNE CAPACITÉ DE FILE D'ATTENTE =2.

Taux de création	1/35	1/30	1/25	1/20	1/15	1/10	1/5
ACO	99.9	99.9	99.9	90.5	64.5	43.2	21.5
SA	99.9	99.9	99.9	94.7	61.9	41.2	20.6
PSO	99.9	99.9	99.9	99.2	69.3	46.1	23.1
GA	99.9	99.9	99.9	98.4	71.0	47.2	23.7
TS	99.9	99.9	99.9	95.2	64.9	43.2	21.7
EM	99.9	99.9	99.9	98.5	66.9	44.7	22.4
MDM M	99.9	99.9	99.7	84.4	60.7	41.6	21.1
DMM	99.9	99.9	81.4	24.6	32.0	15.4	8.88

TABLEAU 3. TAUX DE SORTIE DES PIÈCES POUR UNE CAPACITÉ DE FILE D'ATTENTE =2 ET AVEC PRÉSENCE DE PANNES.

Taux de création	1/35	1/30	1/25	1/20	1/15	1/10	1/5
ACO	99.9	99.9	99.9	84.4	63.1	42.1	21.1
SA	99.9	99.9	99.9	80.7	60.2	40.2	20.2
PSO	99.9	99.9	99.9	90.0	67.5	44.9	22.6
GA	99.9	99.9	99.9	92	69.0	46.2	23.1
TS	99.9	99.9	99.9	84.5	63.4	42.2	21.1
EM	99.9	99.9	99.9	89.7	65.6	43.7	21.9
MDMM	99.9	99.6	85.0	66.7	50.9	34.2	17.0
DMM	99.9	99.9	53.0	32.4	29.2	15.4	8.55

Le tableau 2 et le tableau 3 montrent que pour un taux de création de pièces supérieur ou égale à 1/30, les taux de production obtenus par les métaheuristiques étudiées sont meilleurs que ceux des méthodes DMM et DMM modifiée (sauf le recuit simulé) et qu'en dessous du taux de création 1/30 le taux de production est pratiquement le même pour les deux méthodes et les métaheuristiques, même avec la présence de pannes. Ces tableaux montrent que pour un taux de création de pièces supérieur à 1/25 que les algorithmes génétiques sont plus efficaces que les autres métaheuristiques et les méthodes DMM modifiée et DMM. Et pour un taux de création inférieur à 1/20, les résultats obtenus par les métaheuristiques sont les mêmes, même avec introduction de pannes.

B. Taux d'utilisation des machines.

TABLEAU 4. TAUX D'UTILISATION DES MACHINES T_1 ET T_2 POUR UNE CAPACITÉ DE FILE D'ATTENTE =2.

Taux de création	1/35	1/30	1/25	1/20	1/15	1/10	1/5
ACO	54.5	63.9	76.3	85.7	81.9	82.2	82.0
SA	54.7	64.1	76.2	89.4	80.4	80.4	80.4
PSO	54.7	63.9	76.5	93.2	89.2	88.8	88.9

GA	54.5	64.3	75.6	92.7	90.9	90.8	90.9
TS	54.4	63.7	76.4	90.3	84.7	84.7	85.2
EM	54.4	63.9	76.1	92.3	86.1	86.5	86.7
MDMM	56.9	66.5	79.8	84.5	82.7	83.5	84.7
DMM	57	66.5	65.0	24.6	42.5	30.9	35.5

TABLEAU 5. TAUX D'UTILISATION DES MACHINES T₁ ET T₂ POUR UNE CAPACITÉ DE FILE D'ATTENTE =2 ET AVEC PRÉSENCE DE PANNE.

Taux de création	1/35	1/30	1/25	1/20	1/15	1/10	1/5
ACO	54.5	63.9	76.1	80.4	80.2	80.2	80.5
SA	54.4	64.2	76.2	78.7	78.3	78.5	79.8
PSO	54.8	64.2	76.4	86.7	86.6	86.6	86.9
GA	54.9	64.0	76.1	88.1	88.4	88.6	88.6
TS	54.6	63.9	76.0	82.7	82.8	82.7	82.7
EM	54.6	64.0	75.9	86.1	84.6	84.6	84.9
MDM M	56.9	66.7	77.1	77.7	78.1	78.5	78.7
DMM	57.2	66.8	42.7	33.0	39.3	31.2	34.8

Les tableaux 4 et 5 montrent que le taux d'utilisation des machines T₁ et T₂ est plus important pour les algorithmes génétiques que pour DMM, DMM modifiée et les autres métaheuristiques étudiées, pour un taux de création important (supérieur à 1/25), même avec la présence de pannes.

Pour un taux de création inférieur à 1/25, DMM et DMM modifiée sont plus efficaces que les métaheuristiques étudiées, dans cet intervalle, certaines métaheuristiques sont meilleures que d'autres mais il n'existe pas une métaheuristique qui donne les meilleurs taux d'utilisation des machines pour tous les taux d'arrivée des pièces.

C. Taux d'utilisation de l'AGV.

Le tableau 6 et le tableau 7 montrent que l'utilisation de l'AGV est plus importante pour les métaheuristiques (sauf le recuit simulé et les colonies de fourmis dans le cas sans présence de pannes) que la méthode DMM et DMM modifiée pour un système saturé et de petites files d'attente.

Ces tableaux montrent aussi que pour un taux de création de pièces supérieur à 1/20 et même avec la présence de pannes, le taux d'utilisation d'AGV pour les algorithmes génétiques est plus important que pour les autres métaheuristiques et règles DMM et DMM modifiée.

TABLEAU 6. TAUX D'UTILISATION DE L'AGV POUR UNE CAPACITE DE FILE D'ATTENTE =2.

Taux de création	1/35	1/30	1/25	1/20	1/15	1/10	1/5
ACO	19.5	22.8	27.3	30.6	29.3	29.4	29.4
SA	19.6	23.0	27.3	31.9	29.1	29.0	29.1
PSO	19.6	22.9	27.4	33.2	32	31.9	31.9
GA	19.5	23.1	27.0	33.0	32.6	32.6	32.6
TS	19.5	22.8	27.4	32.3	30.6	30.6	30.8
EM	19.5	22.9	27.2	32.8	31.1	31.2	31.2
MDMM	17.4	21	27.3	30.2	29.1	30.1	30.4
DMM	17.4	20.9	21.6	8.43	14.3	10.5	12.0

Taux de création	1/35	1/30	1/25	1/20	1/15	1/10	1/5
ACO	19.5	22.9	27.2	28.8	28.7	28.7	28.8
SA	19.5	23.1	27.3	28.4	28.3	28.3	28.8
PSO	19.6	23.1	27.4	31.2	31.1	31.1	31.2
GA	19.7	23.0	27.2	31.7	31.7	31.8	31.8
TS	19.6	22.9	27.2	29.9	29.9	29.9	29.9
EM	19.5	23	27.1	30.9	30.4	30.5	30.6
MDMM	17.5	21.1	24.2	24.0	24.5	24.6	24.5
DMM	17.6	21.0	14.2	10.9	12.9	10.4	11.5

TABLEAU 7. TAUX D'UTILISATION DE L'AGV POUR UNE CAPACITÉ DE FILE D'ATTENTE =2 ET AVEC PRÉSENCE DE PANNES.

V. CONCLUSION

Dans ce travail, nous nous sommes intéressés à l'adaptation des métaheuristiques pour résoudre le problème de sélection de routages alternatifs dans un FMS en temps réel. En particulier nous avons étudié les colonies de fourmis, les algorithmes génétiques, le recuit simulé, la recherche tabou, les essais particulières, 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.

De cette étude comparative un certain nombre de conclusions peuvent être faites. Nous résumons ici, l'essentiel:

- Les résultats obtenus ont montré que la plupart des 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 même avec la présence de pannes.
- 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, le taux d'utilisation des machines et du transport, les algorithmes génétiques sont les plus efficaces pour un taux de production supérieur à 1/20.

L'ordonnement temps réel est un domaine qui reste ouvert, et dont l'importance ne cesse de croître. Notre travail doit être approfondie et enrichie pour proposer d'autres solutions pratiques au problème de sélection de routages alternatifs en temps réel. Donc, les perspectives ouvertes par ces travaux sont multiples :

- Durant notre travail, nous avons supposé que si une pièce entre dans la station de chargement, on ne peut pas changer son routage, et nous n'avons aucune garantie que les résultats trouvés sont les optimaux donc ils peuvent être améliorés si on ré-ordonne les pièces contenues dans la station de chargement.
- Toutes les métaheuristiques partagent l'avantage qu'elles prêtent à toutes sortes d'extensions. Mais il est souvent impossible de prévoir avec certitude l'efficacité d'une technique donnée quand elle est appliquée. Donc les métaheuristiques hybrides qui s'efforcent de tirer partie des avantages spécifiques de métaheuristiques différentes en les combinant peuvent améliorer les performances de notre système.
- Durant notre travail nous avons choisi la règle FIFO pour le séquençage de chaque machine, or l'efficacité des règles de priorité dépend étroitement des critères de performance évalués et des conditions opératoires effectives dans l'atelier... Ainsi, nous suggérons pour une sélection de routages alternatifs en temps réel de combiner les algorithmes proposés avec d'autres règles de priorité ou avec des approches de sélection dynamique des règles de priorités

the International Conference on Industrial and Systems Management, China, 2007.

- [7] M. Souier, A. Hassam and Z. Sari. "Evaluation of Metaheuristics performances in manipulation of alternative routing," presented at the international conference on Electrical Engineering Design and Technologies, Hammamet Tunisia, 2008.
- [8] S. Kirkpatrick, C. D. Gelatt and M. P. Vecchi, "Optimization by Simulated Annealing," *Science Number 4598*, volume 220, 4598, p .671-680, 1983.
- [9] N.Metropolis, A.W. Rosenbluth, M.N. Rosenbluth, A.H. Teller and E. Teller, "Equations of state calculations by fast computing machines," *Journal of Chemical Physics*, 21:1087-1091, 1953.
- [10] F. Glover and Manual Laguana. "Tabu Search," Article was principally adapted from the book « *Tabu Search* » of Glover, Laguana , Kluwer Academics publishers, 1997.
- [11] J.H. Holland, *Adaptation in Natural and Artificial Systems*, University of Michigan Press, Ann Arbor, 1975.
- [12] E.E. Goldberg. *Genetic Algorithms in Search, Optimization, and Machine Learning*, Addison Wesley, Reading, MA, 1989.
- [13] M. Dorigo, "Optimization, learning and natural algorithms," PHD thesis, DEI, Politecnico di Milano, Italy, 1992.
- [14] R. C. Eberhart and J. Kennedy, " New optimizer using particle swarm theory," in *Proceedings of the 6th International Symposium on Micro Machine and Human Science*, pp. 39-43, Nagoya, Japan, October 1995.
- [15] M., Clerc et P. Siarry, "Une nouvelle métaheuristique pour l'optimisation difficile: la méthode des essaims particulaires," *J3eA*, 3 (7), 2004.
- [16] S.I. Birbil and S.C. Fang, "An Electromagnetism like Mechanism for Global Optimization," *Journal of Global Optimization*, 25, 263-282, 2003.

REFERENCES

- [1] C. Saygin, F.F. Chen and J. Singh, "Real-Time Manipulation of alternative Routings in Flexible Manufacturing Systems: A simulation Study," *International journal of advanced Manufacturing Technology*, 18, 755-763, 2001.
- [2] N. Mebarki, "Une approche d'ordonnement temps réel basée sur les règles de priorité des files d'attente," PhD thèse, Université de Claude Bernard Lyon1, 1995.
- [3] C. Saygin and S.E. Kilic, "Integrating flexible manufacturing systems with scheduling in flexible manufacturing system," *International journal of advanced Manufacturing Technology*, 15(4), 268-280, 1999.
- [4] L. Ghomri, et Z. Sari, "Influence des Contraintes et des Perturbations sur les Performances des Règles de Routage dans un FMS," la conférence internationale Conception et Production Intégrées, Rabat, 2007.
- [5] C. Saygin and S.E. Kilic, "Dissimilarity Maximization Method for Real-time Routing of Parts in Random Flexible Manufacturing Systems," *The International Journal of Flexible Manufacturing Systems*, 16, 169-182, 2004.
- [6] A. Hassam and Z. Sari, "Real-Time Selection of Process Plan in Flexible Manufacturing Systems: Simulation Study," presented at