

DES ALGORITHMES BASEES SUR LES PRINCIPES DES METAHEURISTIQUES POUR LE RE-ORDONNANCEMENT TEMPS REEL DANS UN FMS AVEC FLEXIBILITE DE ROUTAGE

Souier Mehdi, Hassam Ahmed, Sari Zaki

Laboratoire d'automatique
BP. 230 Faculté des sciences de l'ingénieur
Université Abou Bekr Belkaid –Tlemcen.
{m_souier,a_hassam,z_sari}@mail.univ-tlemcen.dz

RESUME : Cet article présente les résultats d'une étude de simulation d'un système flexible de production (FMS) 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. L'objectif de cette étude est d'analyser l'impact du ré-ordonnement en temps réel des pièces contenues dans la station de chargement sur les performances du système lorsque les métaheuristiques suivantes (les colonies de fourmis, les algorithmes génétiques, le recuit simulé, la recherche tabou, les essais particuliers et l'électromagnétisme) sont appliquées pour résoudre le problème de sélection de routages alternatifs en temps réel. Les résultats obtenus ont montré que pour chaque métaheuristique que nous avons étudiée, le ré-ordonnement améliore les 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.

MOTS-CLES : Système flexible de production, routage alternative, règle de routage alternative, métaheuristique, ré-ordonnement

1 INTRODUCTION

L'une des premières études sur l'ordonnement des systèmes flexibles de production est le travail de Nof et al. (1979) où ils démontrent l'importance et l'effet des décisions d'ordonnement sur les performances des systèmes de production. La définition classique et traditionnelle dit que l'ordonnement est une activité faite en temps différé où les opérations de production sont ordonnées avant le début de la production.

Plusieurs chercheurs proposent différentes méthodes pour fournir une flexibilité à l'ordonnement en temps différé afin d'augmenter les performances des systèmes (Saygin and Kilick, 1996), (Saygin *et al.*, 2001). Mais dans leurs approches les opérations de production sont ordonnées avant le début de la production.

Néanmoins, 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 et de tirer pleinement partie de la flexibilité offerte par les systèmes flexibles de production, les décisions d'allocation et d'ordonnement des opérations et des plans de processus sont généralement prises dynamiquement et à très court terme, en fonction de l'état du système de produc-

tion, des caractéristiques du plan de production et des objectifs de production (Mebarki, 1994).

Les problèmes liés à la technologie des FMS sont relativement complexes comparés aux systèmes de production traditionnels. C'est la raison pour laquelle les problèmes d'ordonnement dans ces systèmes sont généralement NP complet. Il n'existe donc pas d'algorithme capable de résoudre ces problèmes de manière exacte.

Comme méthodes de résolution de problèmes d'optimisation difficile, les métaheuristiques sont des algorithmes de type stochastique visant à résoudre une large gamme de ces problèmes, 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, essais particuliers...). Elles sont généralement conçues au départ pour des problèmes discrets, mais peuvent s'adapter aux autres types de problèmes.

Notre travail se situe dans le cadre de sélection de routages, et nous nous sommes intéressés à analyser l'impact du ré-ordonnement temps réel de la station de chargement sur les performances du système de production lorsque certaines métaheuristiques (les colonies de fourmis, les algorithmes génétiques, le recuit simulé, la recherche tabou, les essais particuliers, l'électromagnétisme) ont été appliquées pour résoudre le

problème de sélection de routages alternatifs dans un FMS en temps réel, et nous allons présenter une étude comparative entre les deux cas avec et sans ré-ordonnement des pièces contenues dans la station de chargement

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 du travail présenté. Dans la troisième section, nous définissons le mode de fonctionnement des algorithmes des métaheuristiques étudiées avec et sans ré-ordonnement de la station de chargement. La dernière section est consacrée aux résultats.

2 ETAT DE L'ART :

Beaucoup de travaux qui concernent l'ordonnement d'un FMS ont été réalisés depuis le début des années 80 où 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.

Bien que plusieurs travaux existants dans la littérature ne considèrent pas la flexibilité de routages, qui est la possibilité d'utiliser des machines, ou des opérations différentes, pour réaliser des produits sous la même configuration du système. 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, Tsubone and Horikawa (1999) 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, Mahmoodi et Mosier (1999) ont étudié l'effet des règles de priorité et des niveaux de flexibilité de routage sur les différentes performances d'un FMS (temps de cycle moyen, retard moyen...).

Saygin et Kilic (1999) ont proposé une plateforme qui intègre le processus de planification flexible et l'ordonnement prédictif (en temps différé), ils 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, 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 (Ghomri et Sari, 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, et dans (Saygin et kilic, 2004) où les trois règles ont été combinée avec d'autres règles de priorité. Dans l'étude de (Hassam et Sari, 2010) sur la règle DMM, Ils ont remarqué que pour une faible capacité de files d'attentes, le système de production se sature rapidement

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 qui dépend de la maximisation des coefficients de dissimilitude pour la sélection des différents routages alternatifs mais en affectant plusieurs pièces à un seul routage. Alors si tous les routages sont sélectionnés par une pièce, 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.

Devant le succès rencontré par ces techniques dans le cadre de résolution des autres problèmes NP-difficiles, nous avons les utilisés pour la manipulation de routages alternatifs dans les FMSs. Dans (Souier *et al.*, 2008) 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 routage dans un FMS avec un nombre important de pièces mais en temps différé. Dans (Souier *et al.*, 2010) nous avons réussi à les appliquer sur le même problème en temps réel, durant ce 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 optimaux donc ils peuvent être améliorés si on ré-ordonne les pièces contenues dans la station de chargement.

Le présent travail concerne l'adaptation des métaheuristiques pour résoudre le problème en temps réel sans et avec pannes des machines avec le ré-ordonnement en temps réel des pièces contenues dans la station de chargement.

3 PRESENTATION DU MODELE FMS ETUDIEE

Pour faire la comparaison entre les deux cas avec et sans ré-ordonnement de la station de chargement, nous avons étudié et simulé un système flexible de production pris de la littérature (Saygin *et al.*, 2001), (Ghomri et Sari, 2007), (Saygin et kilic, 2004) et (Hassam et Sari, 2010). Ce système comprend :

- Deux fraiseuses verticales (FV_1 et FV_2).
- Deux fraiseuses horizontales (FH_1 et FH_2).
- Deux tours (T_1 et T_2).
- Une toupie (TP).
- Une station d'entrée (SE) et une station de sortie (SS)
- Chaque machine comporte une file d'attente d'entrée et une file d'attente de sortie qui ont des capacités égales (n).

La disposition de ces machines dans le système est illustrée dans la figure 1. Un chariot autoguidé bidirection-

nel qui a une durée de disponibilité fixe est utilisé pour le transport des pièces entre les stations. Six types de pièces affluent dans le système, leurs types, leurs taux d'arrivée, leurs routages alternatifs ainsi que leurs temps opératoires sont donnés dans le tableau 1.

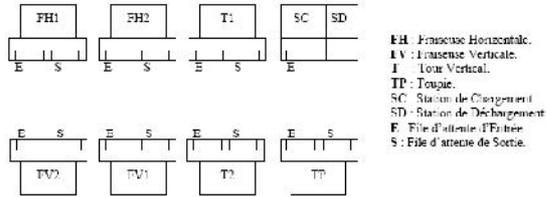


Figure 1 : Configuration du modèle FMS étudié (Saygin et al., 2001).

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

Tableau 1 : Routages alternatifs et temps de traitement des pièces (Saygin et al., 2001).

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.
- Avant d'être traitée par la machine, la pièce doit être passée par sa station d'entrée. Et après le traitement, elle doit être mise dans la file de sortie de la station courante avant qu'elle passe à la file d'entrée de la station suivante.
- Comme il est mentionné dans le tableau, chaque type de pièce arrive avec un pourcentage, c'est-à-dire pour cent pièces qui arrivent dans le système on trouve 17 pièces de type A, 17 pièces de type B.... Toute ces pièces entrent dans le système dans un ordre aléatoire et chaque type de ces pièces a ses propres routages alternatifs, avec l'existence de machines communes entre les routages des différentes pièces et du changement de l'état de l'atelier, ce qui rend le choix des routages alternatifs une tâche difficile.

4 LES ALGORITHMES DES META HEURISTIQUES

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 n 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.

4.1 Le recuit simulé (Simulated Annealing : SA)

La méthode de recuit simulée a été conçue par 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 Metropolis et al. (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 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 de chaque file 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

Sinon 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

4.2 Recherche avec tabous (Taboo Search : TS)

La méthode de recherche tabou, a été formalisée en 1986 par Glover et 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 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 nbre_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.

4.3 Les algorithmes génétiques (Genetic Algorithms : GA)

Les algorithmes génétiques ont été proposés par 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.

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) (Goldberg, 1989).

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.

4.4 Les colonies de fourmis (Ant Colony Optimization : ACO)

Cette métaheuristique a été introduite pour la première fois par Dorigo (1992) et a été inspirée 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 $\Delta \tau_{ij}^k(t)$ qui dépend de la qualité de la solution trouvée (le produit des charges 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 $\Delta \tau_{ij}^k(t)$ piste sur le trajet $T^k(t)$ (pour chaque routage j choisi pour la pièce i par la fourmi k).

Fin Pour.

Évaporer les pistes et modifier les intensités.

Fin Pour.

Finsi.

4.5 Les essais particuliers (Particle Swarm Optimization : PSO)

PSO est une métaheuristique proposée par Kennedy et Eberhart (1995). 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 (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 voisins B_{li} , et B_g sont les bests locaux et globaux. n est la taille des files d'attente.

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 $F(x_i) > F(B_{li})$ alors : (mise à jour du best local)

$B_{li} = x_i$

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

$X_i(t) = c_2 \oplus F_3(c_1 \oplus F_2(w \oplus F_1(X_i(t-1)), B_{li}(t-1)), B_g(t-1))$

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 .

4.6 L'électromagnétisme (electromagnetism like method : EM)

Electromagnétisme est une métaheuristique basée sur une population de solution, elle a été proposée par Birbil et al. (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 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

m : la taille de la population.

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

$avg = \text{calcAvgObjectiveValues}()$ (calcul de moyen des fonctions objectifs).

Pour $i = 1$ à m

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

$j =$ particule sélectionné.

croisement_uniforme (x_i, x_j).

Finsi

Sinon Si $f(x_i) < avg$ Alors

Calcul force et déplace (x_i).

Finsi

Fin pour

Fin tant que

Finsi.

4.7 L'adaptation des métaheuristiques avec ré-ordonnement

Durant le cas sans ré-ordonnement, si une pièce entre dans la station de chargement on l'affecte à un routage selon son type et l'état du système actuel en utilisant les algorithmes présentés précédemment. Mais dès qu'elle est dans la station de chargement on ne peut pas changer son routage même si l'état du système peut changer avant qu'elle passe à la première machine du routage choisi, ce qui va influencer sur les performances du système de production.

Pour cela nous proposons de ré-ordonner les pièces (modifier leurs routages) existantes dans la station de chargement. Notre proposition vise à garder le même principe des algorithmes décrits précédemment, avec la permission de changer les routages des pièces qui sont à l'intérieur de la station de chargement avec les premières pièces de la file infinie à chaque passage d'une nouvelle pièce à cette station et en fonction du nouvel état de l'atelier. Pour ce faire, dans chaque algorithme nous effectuons à chaque itération les modifications suivantes :

- En plus de n premières pièces contenues dans la file infinie, la solution est composée de routages de pièces contenues dans la station de chargement.
- Les opérations de construction de nouvelles solutions tel que les opérateurs de croisement mutation, recherche local ..., doivent prendre en considération toutes ces pièces.

5 RESULTATS ET INTERPRETATIONS

Pour pouvoir analyser l'influence du ré-ordonnement des pièces contenues dans la station de chargement sur les performances du système flexible de production, lorsque les métaheuristiques sont utilisées pour la sélection de routages alternatifs et le système est en état de saturation. Nous avons proposé des études en simulation effectuées sur un PC (Core (TM) 2 Duo CPU avec 2.2 GHZ et 1 GO de RAM) avec des variations sur les critères du système étudié (le taux de création ou d'arrivée de pièces et la taille des files d'attente d'entrée et de sortie des stations). Dans chaque cas deux scénarios ont été pris en compte avec et sans présence de panne. 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 et les interprétations de certains résultats trouvés pour une capacité de file d'attente $n=2$ (chaque file d'attente du système soit une file d'entrée et de sortie d'une station de travail ou une file d'entrée de la station de chargement à une capacité $=2$) sur un temps de simulation de 20000 heures et un régime transitoire de 3000 heures, avec 10 réplications (essais) pour chaque simulation.

5.1 Taux de production

Taux de création	1/35	1/30	1/25	1/20	1/15	1/10	1/5
ACO_Sans_réor	99.9	99.9	99.9	90.5	64.5	43.2	21.5
ACO_Avec_réor	99.9	99.9	99.9	95.3	67.0	44.8	22.3
SA_Sans_ré_or	99.9	99.9	99.9	94.7	61.9	41.2	20.6
SA_Avec_ré_or	99.9	99.9	99.9	94.9	62.4	41.8	20.9
PSO_Sans_ré_or	99.9	99.9	99.9	99.2	69.3	46.1	23.1
PSO_Avec_ré_or	99.9	99.9	99.9	99	71.3	47.6	23.7
GA_Sans_ré_or	99.9	99.9	99.9	98.4	71.0	47.2	23.7
GA_Avec_ré_or	99.9	99.9	99.9	99.9	72.2	48	24
TS_Sans_ré_or	99.9	99.9	99.9	95.2	64.9	43.2	21.7
TS_Avec_ré_or	99.9	99.9	99.9	99.9	66	44	22
EM_Sans_ré_or	99.9	99.9	99.9	98.5	66.9	44.7	22.4
EM_Avec_ré_or	99.9	99.9	99.9	98.9	67.7	45.2	22.7

Tableau 2 : Taux de sortie des pièces.

Taux de création	1/35	1/30	1/25	1/20	1/15	1/10	1/5
ACO_Sans_réor	99.9	99.9	99.9	84.4	63.1	42.1	21.1
ACO_Avec_réor	99.9	99.9	99.9	87.0	65.2	43.6	21.8
SA_Sans_ré_or	99.9	99.9	99.9	80.7	60.2	40.2	20.2
SA_Avec_ré_or	99.9	99.9	99.9	81.4	60.9	40.7	20.3
PSO_Sans_ré_or	99.9	99.9	99.9	90	67.5	44.9	22.6
PSO_Avec_ré_or	99.9	99.9	99.9	92.6	69.5	46.2	23.2
GA_Sans_ré_or	99.9	99.9	99.9	92	69	46.2	23.1
GA_Avec_ré_or	99.9	99.9	99.9	93.3	70.0	46.7	23.3
TS_Sans_ré_or	99.9	99.9	99.9	84.5	63.4	42.2	21.1
TS_Avec_ré_or	99.9	99.9	99.9	85.8	64.2	42.9	21.5
EM_Sans_ré_or	99.9	99.9	99.9	89.7	65.6	43.7	21.9
EM_Avec_ré_or	99.9	99.9	99.9	90.1	66.2	44.3	22.1

Tableau 3 : Taux de sortie des pièces obtenu avec présence de pannes.

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

Les tableaux 2 et 3 nous montrent que le taux de sortie des pièces obtenu par les métaheuristiques étudiées avec ré-ordonnement (avec_réor) est supérieur à celui trouvé sans ré-ordonnement (sans_réor) pour un taux de création de pièces supérieur à 1/25 (un système saturé) et même avec la présence de pannes. Hors de cet intervalle, les résultats obtenus dans les deux cas sont identiques. Ces tableaux montrent aussi que si on compare les résultats obtenus par les métaheuristiques pour un système saturé les algorithmes génétiques sont les plus efficaces.

Par conclusion, on peut dire que le ré-ordonnement de pièces contenues dans la station de chargement a amélioré le taux de sortie des pièces pour un système de production saturé.

5.2 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.

Les tableaux 4 et 5 nous montrent l'augmentation du taux d'utilisation des machines T₁ et T₂ obtenu par les métaheuristiques étudiées avec ré-ordonnement par rapport à celui trouvé sans ré-ordonnement pour un taux de création de pièces supérieur à 1/25. Hors de cet intervalle, nous ne pouvons pas dire qu'un cas est meilleur que l'autre. Ces tableaux montrent que le taux d'utilisation des machines T₁ et T₂ est plus important pour les algorithmes génétiques que pour les autres métaheuristiques étudiées.

Généralement, Les taux d'utilisation des machines T₁ et T₂ obtenus par les métaheuristiques avec ré-ordonnement de pièces contenues dans la station de chargement sont meilleurs que ceux obtenus sans ré-ordonnement pour un taux de création de pièces important et même avec introduction de pannes.

Taux de création	1/35	1/30	1/25	1/20	1/15	1/10	1/5
ACO_Sans_réor	54.5	63.9	76.3	85.7	81.9	82.2	82.0
ACO_Avec_réor	54.7	63.7	76.1	90.4	86.5	86.8	86.6
SA_Sans_ré_or	54.7	64.1	76.2	89.4	80.4	80.4	80.4
SA_Avec_ré_or	55.2	64.1	76.3	90	81.3	81.8	82.0
PSO_Sans_ré_or	54.7	63.9	76.5	93.2	89.2	88.8	88.9
PSO_Avec_ré_or	54.9	63.9	76.1	92.8	91.3	91.3	91.3
GA_Sans_ré_or	54.5	64.3	75.6	92.7	90.9	90.8	90.9
GA_Avec_ré_or	54.5	63.9	76.5	93.2	92.3	92.1	92.2
TS_Sans_ré_or	54.4	63.7	76.4	90.3	84.7	84.7	85.2
TS_Avec_ré_or	54.7	63.8	75.9	93.7	85.9	85.9	86.1
EM_Sans_ré_or	54.4	63.9	76.1	92.3	86.1	86.5	86.7
EM_Avec_ré_or	54.3	64	76.3	92.7	87.5	87.7	87.9

Tableau 4 : Le taux d'utilisation des machines T₁ et T₂

Taux de création	1/35	1/30	1/25	1/20	1/15	1/10	1/5
ACO_Sans_réor	54.5	63.9	76.1	80.4	80.2	80.2	80.5
ACO_Avec_réor	54.7	63.9	76	84.3	84.2	84.6	84.6
SA_Sans_ré_or	54.4	64.2	76.2	78.7	78.3	78.5	79.8
SA_Avec_ré_or	54.5	63.9	75.9	79.6	79.4	79.1	79.6
PSO_Sans_ré_or	54.8	64.2	76.4	86.7	86.6	86.6	86.9
PSO_Avec_ré_or	54.7	63.9	76.1	88.9	88.9	88.9	89.1
GA_Sans_ré_or	54.9	64.0	76.1	88.1	88.4	88.6	88.6
GA_Avec_ré_or	54.8	63.9	76.1	89.6	89.7	89.7	89.7
TS_Sans_ré_or	54.6	63.9	76.0	82.7	82.8	82.7	82.7
TS_Avec_ré_or	54.5	63.8	76.1	83.7	83.6	83.7	84
EM_Sans_ré_or	54.6	64	75.9	86.1	84.6	84.6	84.9
EM_Avec_ré_or	54.7	64	76.2	86.6	85.6	85.9	85.9

Tableau 5 : Le taux d'utilisation des machines T₁ et T₂ obtenu avec présence de pannes.

5.3 Taux d'utilisation de l'AGV

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

Les tableaux 6 et 7 nous montrent que pour un système saturé, l'utilisation de l'AGV est plus importante pour chaque métaheuristique dans le cas avec ré-ordonnement que pour le cas sans ré-ordonnement, même avec la présence de pannes.

Taux de création	1/35	1/30	1/25	1/20	1/15	1/10	1/5
ACO_Sans_réor	19.5	22.8	27.3	30.6	29.3	29.4	29.4
ACO_Avec_réor	19.6	22.8	27.2	32.3	31.2	31.3	31.2
SA_Sans_ré_or	19.6	23	27.3	31.9	29.1	29	29.1
SA_Avec_ré_or	19.9	23.1	27.3	32.1	29.4	29.6	29.6
PSO_Sans_ré_or	19.6	22.9	27.4	33.2	32	31.9	31.9
PSO_Avec_ré_or	19.7	22.9	27.2	33	32.8	32.8	32.8
GA_Sans_ré_or	19.5	23.1	27	33.0	32.6	32.6	32.6
GA_Avec_ré_or	19.5	22.9	27.4	33.6	33.1	33.1	33.1
TS_Sans_ré_or	19.5	22.8	27.4	32.3	30.6	30.6	30.8
TS_Avec_ré_or	19.6	22.9	27.1	33.3	31.1	31.1	31.1
EM_Sans_ré_or	19.5	22.9	27.2	32.8	31.1	31.2	31.2
EM_Avec_ré_or	19.4	23	27.3	33	31.5	31.6	31.7

Tableau 6 : Le taux d'utilisation de l'AGV.

Taux de création	1/35	1/30	1/25	1/20	1/15	1/10	1/5
ACO_Sans_réor	19.5	22.9	27.2	28.8	28.7	28.7	28.8
ACO_Avec_réor	19.6	23	27.2	30.5	30.3	30.5	30.5
SA_Sans_ré_or	19.5	23.1	27.3	28.4	28.3	28.3	2.88
SA_Avec_ré_or	19.5	22.9	27.1	28.8	28.7	28.6	28.8
PSO_Sans_ré_or	19.6	23.1	27.4	31.2	31.1	31.1	31.2
PSO_Avec_ré_or	19.6	22.9	27.2	31.9	31.9	31.9	3.01
GA_Sans_ré_or	19.7	23	27.2	31.7	31.7	31.8	31.8
GA_Avec_ré_or	19.6	22.9	27.2	32.2	32.2	32.2	32.2
TS_Sans_ré_or	19.6	22.9	27.2	29.9	29.9	29.9	29.9
TS_Avec_ré_or	19.5	22.9	27.2	30.2	30.2	30.3	30.3
EM_Sans_ré_or	19.5	23	27.1	30.9	30.4	30.5	30.6
EM_Avec_ré_or	19.6	23	27.3	31.1	30.8	30.9	30.9

Tableau 7 : Le taux d'utilisation de l'AGV obtenu avec présence de pannes.

6 CONCLUSIONS ET PERSPECTIVES

Dans ce travail, nous avons adapté les métaheuristiques étudiées pour résoudre le problème de sélection de routage en temps réel dans un FMS avec et sans ré-ordonnement de la station de chargement. Nous nous sommes intéressés dans cette étude aux différences entre le taux de sortie des pièces, le taux d'utilisation des machines et de système de transport.

L'analyse présentée dans ce travail démontre l'effet du ré-ordonnement des pièces contenues dans la station de chargement sur les performances du système flexible de production si on utilise les métaheuristiques pour la sélection de routages alternatifs. De cette analyse nous avons constaté que pour chaque métaheuristique que

nous avons étudiée, le ré-ordonnement améliore les 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'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 :

- **L'utilisation des métaheuristiques hybrides**

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.

- **Le ré-ordonnement en temps réel des pièces en cours de traitement**

L'étude expérimentale a montré que les en-cours sont très faible par rapport à la capacité de l'atelier, ce qui peut être justifié par la présence de machines goulots (particulièrement T_1 et T_2) donc il est possible d'approfondir le principe de ré-ordonnement en temps réel en permettant de changer les routages de toutes ou certaines pièces qui sont à l'intérieur des files d'entrée et de sortie des machines, afin de pouvoir minimiser ou éviter ces impasses.

- **Combinaison avec les autres règles de priorités**

Durant notre travail nous avons choisi la règle FIFO pour le séquençement 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.

REFERENCES

Birbil, S.I. and S.C. Fang, 2003. An Electromagnetism like Mechanism for Global Optimization. *Journal of Global Optimization*, 25, 263-282.
Clerc, M. et P. Siarry, 2004. Une nouvelle métaheuristique pour l'optimisation difficile: la méthode des essais particuliers, *J3eA*. 3 (7).

Dorigo. M., (1992). *Optimization, learning and natural algorithms*. Italy, PhD thesis, DEI, Politecnico di Milano.
Eberhart R. C. 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.
Ghomri, L. et Z. Sari, 2007. Influence des Contraintes et des Perturbations sur les Performances des Règles de Routage dans un FMS, *la conférence internationale Conception et Production Intégrées*, Rabat,
Glover F. and M. Laguna, 1997. Tabu Search. *Article was principally adapted from the book « Tabu Search » of Glover*, Kluwer Academics publishers
Goldberg, E.E., 1989. *Genetic Algorithms in Search, Optimization, and Machine Learning*. Addison Wesley, Reading, MA.
Hassam A. and Z. Sari, 2010. Selection of alternative routings in real time: DMM and modified DMM rules, *International Journal of Product Development*, Vol. 10, N°1/2/3, pp 241-258
Holland, J.H, 1975. *Adaptation in Natural and Artificial Systems*. University of Michigan Press, Ann Arbor.
Kirkpatrick, S., C. D. Gelatt, and M. P. Vecchi, 1983. Optimization by Simulated Annealing, *Science*, vol. 220 (4598), p. 671-680.
Mahmoodi, F. and C.T. Mosier 1999. The Effects of Scheduling Rules and Routing Flexibility on the Performance of a Random Flexible Manufacturing System, *The International Journal of Flexible Manufacturing Systems*, 11, p. 271-289.
Mebarki, N., 1995. *Une approche d'ordonnement temps réel basée sur les règles de priorité des files d'attente*. PhD thèse, Université de Claude Bernard Lyon1, France.
Metropolis, N., 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.
Nof, S., M. Barash and J. Solberg, 1979. Operational control of item flow in versatile manufacturing system. *International journal of production research*, 17, p. 479-489.
Saygin C. and S.E. Kilick, 1996. Effect of flexible process plans on performance of flexible manufacturing system. *Proceedings of 7th International DAAM Symposium*, Vienna, p.393-394.
Saygin, C. and S.E. Kilic, 1999. Integrating flexible manufacturing systems with scheduling in flexible manufacturing system, *International journal of advanced Manufacturing Technology*, 15(4), p. 268-280.
Saygin, C. and S.E. Kilic, 2004. Dissimilarity Maximization Method for Real-time Routing of Parts in Random Flexible Manufacturing Systems, *The International Journal of Flexible Manufacturing Systems*, 16, p. 169-182.

- Saygin, C., 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, p 755-763.
- Souier, M., A., Hassam et Z., Sari, 2008. Evaluation of Metaheuristics performances in manipulation of alternative routing, International conference on Electrical Engineering Design and Technologies, Hammamet Tunisia.
- Souier, M., A., Hassam et Z., Sari, 2010. Meta-heuristics 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. 8, pp. 221-247. ISBN 978-1-84996-118-9
- Tsubone, H. and M. Horikawa, 1999. A Comparison between Machine Flexibility and Routing Flexibility, *The International Journal of Flexible Manufacturing Systems*, 11, p 83-101.