

CONDUITE D'UNE CELLULE DE PRODUCTION FLEXIBLE PAR UNE MÉTA-HEURISTIQUE BASÉE SUR LA RECHERCHE TABOU

Khaloudna Zahia, Djeghaba Messaoud

Laboratoire d'automatique et de signaux de Annaba (LASA),
Université Badji Mokhtar, BP12, Annaba, Algérie
zayakhaled@yahoo.fr , djeghaba_mes@yahoo.com

Abstract : The organization and the conduct of the flexible manufacturing systems are confronting the variation of the request and at uncertain environment economics. In this paper, we suggest to use a method meta-heuristic based on taboo search, that permit to solve the problem of flexible manufacturing systems scheduling, modeled by Petri net, in which objective to minimized the cost of production.

Résumé : La gestion et la conduite d'un atelier de production flexible sont confrontées aux variations rapides de la demande et à l'incertitude de l'environnement économique. Dans ce papier on propose l'utilisation d'une méthode méta-heuristique basée sur la recherche tabou qui permet de résoudre le problème d'ordonnancement dans lequel l'objectif est de minimiser le coût de production.

Mots clés : Modélisation, Ordonnancement, Conduite, Méta-heuristiques, Réseau de Pétri, Recherche tabou

1- Introduction :

Les problèmes d'ordonnancement sont présent dans tous les secteurs de l'économie et constituent une fonction importante en gestion de production. Un problème d'ordonnancement consiste à allouer dans le temps des travaux à des ressources existantes en quantité limitée tout en satisfaisant un ensemble de contraintes. Les ateliers de production flexible s'avère difficile a ordonnancer.

En effet, pour de tels problèmes ou les méthodes exactes requièrent un effort calculatoire qui croit exponentiellement avec leur taille. C'est pourquoi, des méthodes approchées ont été proposées. Elles apportent des solutions acceptables, en des temps raisonnables. Parmi ces méthodes on mentionne les algorithmes génétiques, recuit simulé, colonie de fourmis, recherche tabou..., et c'est cette dernière que nous nous proposons d'appliquer aux problèmes de conduite de la cellule de production flexible.

Les questions soulevées, sont liées aux conflits d'accès aux ressources partageables, et en nombre limités.

Il s'agit de les affecter de façon à réduire aux maximum leur temps d'oisiveté, ainsi augmenter la productivité, et la production.

2- Conduite :

Le rôle de la conduite d'un système de production est de piloter les différentes ressources du système pour accomplir les travaux planifiés au préalable. La conduite est rendue difficile par les aléas de fonctionnement des différentes ressources et d'autres perturbations.

De fait, la cellule de production est un système à événement discret, pour la conduire il y a nécessité d'un modèle. L'outil le mieux adapté actuellement pour cette opération de modélisation est le réseau de pétri.

Il permet la gestion, de la concurrence, de la synchronisation, du parallélisme entre processus, et peut visualiser l'état du système à tout moment.

L'aspect aléatoire des événements est pris en charge par un des extensions du R.d.P, qui est le R.d.P stochastique.

Il s'agit pour nous d'obtenir des solutions acceptable, à défaut d'être optimale, au problème d'ordonnancement dans la cellule de production, en dépit du comportement incertain de ses divers constituants, et ses événements imprévus.

3- L'ordonnancement

Un problème d'ordonnancement consiste à organiser dans le temps la réalisation de tâches, compte tenu de contraintes temporelles (délais, contraintes d'enchaînement) et de contraintes portant sur la disponibilité des ressources requises.

Il est défini par l'allocation des ressources et vise à satisfaire un ou plusieurs objectifs, ensuite une procédure d'optimisation, choisie la "meilleure" séquence au sens du (des) critère(s) donné (s).

4- Optimisation

Un problème d'optimisation combinatoire consiste à trouver la *meilleure* solution dans un [ensemble discret](#) dit ensemble des solutions réalisables. En général, cet ensemble est [fini](#) mais il lui est associé une liste, relativement courte, de contraintes que doivent satisfaire les solutions réalisables.

Clairement, un problème d'optimisation peut avoir plusieurs solutions optimales, Pour un ensemble de solutions, la meilleure solution (ou *solution optimale*) est celle qui minimise ou maximise la fonction objectif. L'[optimisation](#) suppose que les solutions candidates à un problème puissent être ordonnées selon un ou plusieurs critères d'évaluation, construits sur la base d'indicateurs de performances. On cherchera donc à minimiser ou maximiser de tels critères.

L'évaluation des performances devient dans ce cas un élément essentiel dans la recherche de la meilleure manière de prendre les décisions. Le réseau de Pétri est un outil très efficace pour l'analyse et l'évaluation de ces systèmes.

5- Réseau de pétri

Le terme «réseau de pétri» désigne une famille de graphes orientées, munis d'un formalisme mathématique qui fait intervenir la manipulation des nombres entiers ou réels positifs. Plusieurs classes de réseau de pétri ont été développées et étudiées. Parmi celles-ci on distinguera les réseaux de pétri autonomes et les réseaux de pétri dépendant du temps. Pour les premiers, seul l'ordre d'apparition des événements est pris en compte alors que pour les seconds les instants d'occurrence des événements interviennent également. Pour représenter le comportement des systèmes dynamiques, il est nécessaire de considérer le temps. La modélisation s'effectue à l'aide des réseaux de Pétri temporisées RdPT. Le temps peut être associé indifféremment aux places ou aux transitions. La fonction d'évaluation est basée sur l'équation fondamentale des R.d.P. Soit :

$$M_{k+1} = M_k + C(p, t) * D \quad (1)$$

Avec C (p, t) : Matrice d'incidence donnée par :

$$C_{(p_i, t_j)} = \begin{cases} -1 & \text{si } p \text{ est placée entrée de } t_j \\ +1 & \text{si } p \text{ est placée sortie de } t_j \end{cases} \quad (2)$$

Et D est le vecteur de franchissement contenant toutes les transitions franchies par le marquage M.

Des conflits apparaissent lorsqu'une place est en amont de plusieurs transitions, et que le marquage de la place commune ne permet pas de franchir toutes les transitions qui la partagent. Dans ce cas, un outil de prise de décisions complète le réseau de Pétri. Il résout le conflit en précisant quelles transitions franchir, et ceci par association d'une méthode méta-heuristique qui assure l'efficacité de décision.

6- Heuristiques et méta-heuristiques

Une [heuristique](#) est une méthode pour résoudre de manière approchée un type de problème particulier, dont la solution optimale ne peut être obtenue

Une [méta-heuristique](#) est une méthode, ou plus précisément, un canevas de méthodes, pour résoudre de manière approchée tous les problèmes dont la solution optimale ne peut être obtenue. La méthode ne dépend donc plus du type de problème auquel on est confronté.

a- Les heuristiques

Pour certains problèmes, les algorithmes ont une complexité beaucoup trop grande pour obtenir un résultat en temps raisonnable, même si l'on pouvait utiliser une puissance de calcul phénoménale. On est donc amené à rechercher une solution la plus proche possible d'une solution optimale en procédant par essais successifs. Puisque toutes les combinaisons ne peuvent être essayées, certains choix stratégiques doivent être faits. Ces choix, généralement très dépendants du problème traité, constituent ce qu'on appelle une [heuristique](#). Le but d'une [heuristique](#) est donc de ne pas essayer toutes les combinaisons possibles avant de trouver celle qui répond au problème, afin de trouver une solution approchée convenable (qui peut être exacte dans certains cas) dans un temps raisonnable.

Dans un objectif de résolution de problèmes et de *prise de décision*, les heuristiques trouvent cependant leur place dans les algorithmes qui nécessitent l'exploration d'un grand nombre de cas, car celles-ci permettent de réduire leur complexité moyenne en examinant d'abord les cas qui ont le plus de chances de donner la réponse. .

b- Méta-heuristique

Les méta-heuristiques forment une famille d'[algorithmes d'optimisation](#) visant à résoudre des problèmes d'[optimisation difficile](#) pour lesquels on ne connaît pas de méthode classique plus efficace, ils sont souvent employées en optimisation [combinatoire](#).

Les méta-heuristiques sont généralement des algorithmes [stochastiques](#), qui progressent vers un [optimum](#) par [échantillonnage](#) d'une [fonction objectif](#).

Les méta-heuristiques ne nécessitent pas de connaissance particulière sur le problème optimisé pour fonctionner, le fait de pouvoir associer une (ou plusieurs) valeurs à une solution est la seule information nécessaire. En pratique, elles sont utilisées sur des problèmes ne pouvant être optimisés par des méthodes mathématiques.

Les méta-heuristiques les plus classiques sont celles fondées sur la notion de parcours. Dans cette optique, l'algorithme fait évoluer une seule solution sur l'espace de recherche à chaque itération. La notion de [voisinage](#) est alors primordiale, les plus connues dans cette classe sont le [recuit simulé](#), la [recherche avec tabous](#), l'autre approche utilise la notion de population elle manipule un ensemble de solutions en parallèle, à chaque itération, on peut citer les [algorithmes génétiques](#), les [algorithmes de colonies de fourmis](#).

La méthode proposée dans cet article est basée sur la recherche taboue.

Dans ce qui suit nous allons présenter et appliquer la méthode de recherche tabou sur une cellule de production flexible pilote, développée au LAGIS (LILLE). L'application de la recherche tabou consiste, à partir d'une liste d'ordre de fabrication et de gammes à calculer les plans de charges occupant chaque machine et chaque opérateur avec le moins possible de temps mort. L'objectif visé est la minimisation du temps d'exécution globale du plan de charges [Le MAKESPAN].

7- Recherche tabou

La recherche taboue est une [métaheuristique](#) d'optimisation présentée par [Fred Glover](#) en 1986. On trouve souvent l'appellation *recherche avec tabous* en français.

Cette méthode est une méta-heuristique itérative qualifiée de *recherche locale* au sens large.

a- Principe

Son principe consiste à explorer l'espace de recherche composé de toutes les solutions réalisables dans le but d'aboutir à la solution optimale et qui minimise la fonction objective.

Commençant à partir d'une solution initiale réalisable, le processus consiste, à chaque itération à choisir la meilleure solution dans le voisinage de la solution courante, même si cette solution n'entraîne pas une amélioration.

Il est essentiel de noter que cette opération peut conduire à augmenter la valeur de la fonction : c'est le cas lorsque tous les points du voisinage ont une valeur plus élevée. C'est à partir de ce mécanisme que l'on échappe aux minima locaux.

Afin d'éviter le piège des optima locaux dans lequel ce processus peut être facilement attrapé, la recherche tabou utilise une structure de mémorisation temporaire dans laquelle elle sauvegarde les dernières solutions visitées : la liste tabou.

Le mécanisme consiste à interdire (d'où le nom de *tabou*) de revenir sur les dernières positions explorées.

Les positions déjà explorées sont conservées dans une pile [FIFO](#) (appelée souvent *liste tabou*) d'une taille donnée, qui est un paramètre ajustable de l'heuristique. Cette pile doit conserver des positions complètes, ce qui dans certains types de problèmes, peut nécessiter l'archivage d'une grande quantité d'informations. Cette difficulté peut être contournée en ne gardant en mémoire que les mouvements précédents, associés à la valeur de la fonction à minimiser.

En effet, une solution reste interdite pendant un nombre d'itérations égal à la taille de cette liste. Par suite le meilleur voisin non tabou sera choisi pour la prochaine itération.

b- Liste taboue dynamique :

L'un des aspects critiques de l'utilisation d'une recherche taboue est la nécessité d'ajuster la longueur de la liste taboue pour parcourir efficacement l'espace des solutions. Si la liste est trop courte, la recherche finit par explorer un optimum local de rayon légèrement supérieur. Les différentes solutions explorées forment un *cycle* qui va se répéter indéfiniment. A l'inverse, si la liste est trop longue, tous les mouvements peuvent devenir tabous et sans nouveaux voisins la recherche s'arrête, c'est un *blocage*. Le réglage de la longueur de la liste taboue dépend essentiellement de la topologie de l'espace des solutions, pour laquelle le voisinage et le critère tabou sont nos seuls éléments de mesure. En général, la liste taboue doit être maintenue à une longueur minimale permettant d'éviter un cycle. En cas de blocage, tous les mouvements sont tabous. Cette situation est facile à détecter. Il suffit de diminuer la longueur de la liste pour autoriser de nouveaux mouvements.

c- Fonction voisinage :

La complexité d'une approche de résolution basée sur la recherche tabou dépend essentiellement de la taille du voisinage de la solution courante et de la méthode d'évaluation de chacun de ces voisins afin déterminer celui qui minimise la fonction coût.

d- Évaluation du voisinage :

Le meilleur voisin non tabou parmi le voisinage de la solution courante sera sélectionné pour l'itération suivante. Pour cela il faudrait pouvoir évaluer tous les voisins, cependant une évaluation complète.

e- Algorithme :

Étape 1 : Choisir une solution initiale i dans S (l'ensemble des solutions)

Appliquer $i^*=i$ et $k=0$,

Étape 2 : Appliquer $k=k+1$, et générer un sous-ensemble de solution en $N(i, k)$ pour que :

- les mouvements tabous ne soient pas choisis

- un des critères d'aspiration $a(i, m)$ soit applicable

Étape 3 : choisir la meilleure solution i' parmi l'ensemble de solutions voisines $N(i, k)$

Appliquer $i=\text{meilleur } i'$

Étape 4 : Si $f(i) \leq f(i^*)$, alors nous avons trouvé une meilleure solution

Appliquer $i^*=i$

Étape 5 : Mettre à jour la liste T et les critères d'aspiration

Étape 6 : Si une condition d'arrêt est atteinte, stop.

Si non, retour à Étape 2

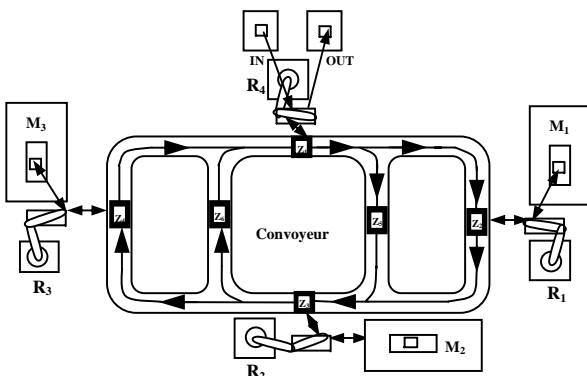
Condition d'arrêt : condition qui régira l'arrêt de l'algorithme.

8- Application :

Dans les systèmes industriels, les opérations appartiennent à des catégories différentes : opérations de transferts, d'usinage, et de stockage. Elles sont définies au niveau des ressources. Plusieurs opérations peuvent rendre effectif les mêmes fonctions dans une FMS (cas de ressources redondantes). Une ressource peut s'exécuter aussi sur plusieurs opérations (cas de ressources polyvalentes). Soit la cellule de production flexible pilote, développée au LAGIS (LILLE). Les ressources principales de ce processus sont trois machines appelées M1, M2, M3. Elles exécutent respectivement les fonctions d'usinage : M1 (f1, f3), M2 (f1, f3), et M3 (f2, f3). R14 est un robot qui exécute des opérations du transfert de FIFO IN à Z1, de Z1 à FIFO OUT, de Z1 à Z2, de Z2 à Z1, de Z1 à Z4, et de Z4 à Z1. R11 est un robot de chargement /déchargement affecté à M1, R12 est un robot de chargement /déchargement affecté à M2, R13 est un robot de chargement /déchargement affecté à M3. CV est un convoyeur, permettant le transfert de Z1 à Z2 ou Z5, de Z2 ou Z5 à Z3, de Z3 à Z4 ou Z6, de Z4 ou Z6 à Z1.

Opération exécutée par une machine de la FMS : $OP_{Mj}^{f_i}$ est l'opération exécutée par la machine M_j pour le fonctionnement f_i . Pour simplifier, les opérations de transfert seront notées $TRE_{R_j}^{L1 \rightarrow L2}$ et signifie que c'est une opération de transfert élémentaire exécutée par le robot R_j de la région $L1$ vers la région $L2$.

Ce processus autorise une certaine flexibilité. Des ressources différentes réalisent les mêmes fonctions d'usinage et plusieurs fonctions peuvent être réalisées par une seule machine. Exemple, la fonction $f1$ est réalisée par M1 et M2. et M1 réalise les opérations $f1$ et $f3$, OP_{M1}^{f1} et OP_{M1}^{f3} . Le convoyeur est une ressource de transport complexe. Des chemins différents peuvent être suivis pour atteindre le même emplacement. R14 peut exécuter le même transfert comme le convoyeur, de Z1 à Z2. C'est un exemple de flexibilité du transport.



Il y a trois types de fonctions d'opérations :

f_1 : opération de perçage, f_2 : opération de fraisage1, f_3 : opération de fraisage2

Nous proposons six gammes d'opérations : par exemple :

gamme1 :

$$IN \xrightarrow{R_{14}} Z_2 \xrightarrow{OP_{M_1}^{f_1}} Z_2 \xrightarrow{CV} Z_3 \xrightarrow{R_{13}} Z_4 \xrightarrow{OP_{M_3}^{f_2}} Z_4 \xrightarrow{CV} Z_1 \xrightarrow{R_{14}} OUT$$

gamme2 :

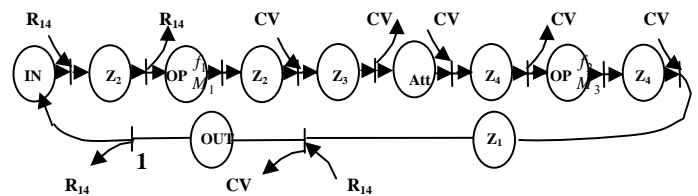
$$IN \xrightarrow{R_{14}} Z_1 \xrightarrow{CV} Z_5 \xrightarrow{OP_{M_2}^{f_3}} Z_3 \xrightarrow{R_{12}} Z_4 \xrightarrow{OP_{M_3}^{f_2}} Z_4 \xrightarrow{CV} Z_1 \xrightarrow{R_{14}} OUT$$

gamme3 :

$$IN \xrightarrow{R_{14}} Z_2 \xrightarrow{OP_{M_1}^{f_1}} Z_2 \xrightarrow{CV} Z_3 \xrightarrow{OP_{M_2}^{f_3}} Z_3 \xrightarrow{R_{12}} Z_4 \xrightarrow{OP_{M_3}^{f_2}} Z_4 \xrightarrow{CV} Z_1 \xrightarrow{R_{14}} OUT$$

Ces gammes sont réalisées par un robot de transport, un convoyeur, trois machines d'usinages M_1 (f_1, f_3), M_2 (f_1, f_3), M_3 (f_2, f_3). L'arrivée des pièces étant aléatoire, nous avons besoin d'un modèle d'ordonnancement dynamique capable de prendre en compte toutes les contraintes et les perturbations endogènes ou exogènes (panne des machines, rupture des stocks, modifications sur le marché...).

Une modélisation compacte a conduit au modèle global construit sur la base d'un R.d.P stochastique.



A chaque jeton est associée, une variable temporelle stochastique, qui représente l'évolution des opérations de transport et d'usinage modélisée par la place correspondante. A chaque transition T_j . Il correspond un état de conflit qu'il faut résoudre par le lancement de l'algorithme de recherche tabou.

9- Résultats :

Ces tableaux donnent le nombre de pièces produites par unité de temps :

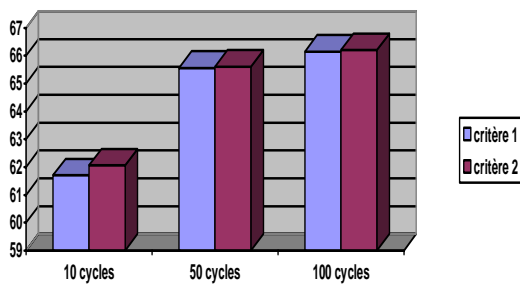
Taux de production = nombre de pièces produites / temps moyen d'exécution %

Critère 1 = La fonction d'évaluation f est la durée d'exécution effectuée sans le temps morts.

	RECHERCHE TABOU		
	CRITÈRE 1		
N cycles	10	50	100
Pas de panne	61.70	65.57	66.15
Dépannage du R ₁₄	59.00	64.94	65.84
Dépannage du CV	61.70	65.57	65.97

Critère 2 = La fonction d'évaluation f est la durée d'exécution effectuée avec le temps morts

	RECHERCHE TABOU		
	CRITÈRE 2		
N cycles	10	50	100
Pas de panne	62.02	65.61	66.20
Dépannage du R ₁₄	59.00	64.94	65.84
Dépannage du CV	62.06	65.61	65.88



Histogrammes des deux critères

10- Conclusion

La recherche de la séquence optimale par La recherche taboue a donnée des résultats prometteurs

- Si en augmente le nombre de cycles, tous les paramètres s'améliorent, ceci s'interprète par le fait que l'espace de recherche augmente, et donne à l'algorithme une meilleure chance de trouver l'optimum.
- Le taux de production et le nombre de pièces produites varie selon l'adaptation de la fonction d'évaluation.
- Cependant, quelques problèmes reste à étudier, surtout une meilleure adaptation de la liste taboue, c'est-à-dire de bien choisir la taille de la liste taboue, une liste trop longue peut être restrictive, par contre une liste trop courte risque de s'avérer inutile.
- Ne génère pas toujours la solution optimale, vue quelle permet les mouvements non améliorateurs.

11-Références

[1] C.Khanphang Bousathing, "Algorithmes heuristiques et évolutionnistes. Application à la résolution du

problème de déplacement de formes irrégulières" *Thèse de doctorat, université des sciences et technologie de Lille*, 19/10/98

- [2] R.David. " Du grafcet aux réseaux de pétri», Paris 1992.
- [3] D.Lefebvre "Contribution à la modélisation des systèmes dynamiques à événements discrets pour la commande et la surveillance" *Habilitation à diriger des recherches*. Belfort, novembre 2000
- [4] R.Ghoul., M.Djeghaba "Modelling and Performance Evaluation of FMS Using SCWN", *4th Jordanian International Electrical and Electronic Engineering Conference*, Amman, Jordan, Avril 2001.
- [5] R.Ghoul., M.Djeghaba "Modélisation et Conduite d'une FMC par Décoloration des RdP bien-formés", *Conférence internationale sur la productique*, Alger, juin 2001.
- [6] R.Ghoul, H.Tebbikh, M.Djeghaba "Conduite d'un système de production flexible par les SCWN", *APII-JESA*, Volume 36 – n°10/2002, pages 1399 à 1411.
- [7] L. Haudot " Une approche orientée utilisation pour la conception de systèmes coopératifs en ordonnancement de production", *Thèse de doctorat*, LAAS, Toulouse, 1996.
- [8] E.Taillard, "Méta-heuristiques et outils nouveaux en R.O", *Thèse de doctorat* de l'institut d'informatique, Haute École spécialisée de suisse occidentale.1999.
- [9] Ghoul R, Djeghaba M., «Gestion des Ressources Partagées dans une Cellule de Production Flexible par des Algorithmes Heuristiques », *Conférence Euro-Maghrébine sur la mécanique*, Annaba, Novembre
- [10] Ghoul R.,Tebbikh H., «Production management in a flexible Manufacturing cell with a Petri net model And meta-Heuristic methods». *Third assiut university international conference on mechanical engineering advanced technology for industrial production*, Assiut, December 24-26, 2002.-
- [11] Marc Sevaux, Philippe Thomin, " Recherche taboue améliorée pour l'ordonnancement sur machines parallèles", 3ème Conférence Francophone de MODélisation et SIMulation, MOSIM'01 du 25au 27 avril 2001 France.
- [12] Meriem Ennigrou , Khaled Ghérida, "Approche MultiAgents basée sur la recherche taboue pour job shop flexible
- [13] Joseph Ayas, Marc André Viau, « La recherche Taboue ». 16 novembre 2004.