

CONSTRUCTION D'HEURISTIQUES BASEES SUR DES REGLES DE PRIORITE POUR JOB SHOP PAR OPTIMISATION VIA SIMULATION

W.MOUELHI, H.PIERREVAL

LIMOS, UMR CNRS n°6158, site de l'IFMA, BP 165, F-63175 Aubière Cedex, France.

wiem.mouelhi@ifma.fr, henri.pierreval@ifma.fr

RESUME : Dans cet article, nous nous intéressons au cas d'un atelier de type job shop, l'objectif étant de déterminer dynamiquement la séquence de passage des pièces sur chaque machine, en considérant l'état réel du système à chaque instant t de prise de décision. L'approche que nous proposons consiste à construire des heuristiques basées sur des règles de priorité à partir d'une logique déterminée par un réseau de neurones. Cette logique est améliorée par optimisation via simulation. En effet, après optimisation via simulation de sa logique, déterminée principalement par ses poids, par rapport à la fonction objectif : minimiser le retard moyen, le réseau de neurones possède la capacité de décider quelle pièce de la file d'attente passera sur la machine, en fonction de l'état de l'atelier. Nous utilisons ARENA pour modéliser l'atelier job shop et le recuit simulé comme méthode d'optimisation. Nous avons mené des expérimentations sur l'atelier job shop étudié, ce dernier est inspiré du travail de (Fonseca et Navarrese, 2002). Les résultats ont montré qu'après optimisation, le réseau de neurones est devenu capable de choisir les combinaisons des règles sans se baser ni sur une expertise préalable, ni sur un ensemble d'apprentissage.

MOTS-CLES : job shop, ordonnancement dynamique, optimisation via simulation, règle de priorité, réseau de neurones

1. INTRODUCTION

Les systèmes de production sont de plus en plus sujets à des perturbations (pannes, absentéisme), des décalages entre ce qui était prévu et ce qui est réalisé, des informations erronées, etc. Dans un tel contexte, le besoin d'outils d'ordonnancement très réactifs est souvent exprimé. Dans cet article, nous nous intéressons plus particulièrement au cas d'ateliers de type job shop, l'objectif étant de déterminer dynamiquement la séquence de passage des pièces sur une machine en se basant sur les règles de priorité. Le choix en temps réel des tâches par règles de priorité (RP) constitue une des approches les plus simples et les plus étudiées dans la littérature (Blackstone et al., 1982, Boucon, 1991, Sabuncuoglu, 1998). Plusieurs auteurs se sont intéressés au choix des RPs dans un atelier job shop (e.g., Haupt, 1989, Ramasesh, 1990, Rajendran et Holthaus, 1999, Mohanasundaram et al., 2002). Haupt (1989) a présenté une étude de la littérature existante sur la gestion des files d'attentes des ateliers job shop et il a classifié les RPs par leur efficacité pour différentes mesures de performance. En 1990, Ramasesh a présenté un état de l'art et une classification des RPs qui peuvent être statiques, dynamiques, simples, combinées, tronquées ou basées sur des heuristiques. Cette classification inclut aussi des critères de sensibilité à certaines propriétés de l'atelier.

Dans un certain nombre de cas, ces RPs utilisent des seuils, c'est le cas de la règle tronquée SPT/FIFO qui bascule de SPT (shortest processing time) à FIFO (first in first out) lorsque une pièce attend depuis plus de x unités de temps. Pierreval et Mebarki (Pierreval et Mebarki, 1997) ont utilisé des seuils à l'intérieur d'un ensemble de règles destinées à changer dynamiquement de RP selon l'état du système. Par exemple : si le retard d'une pièce devient supérieur à x , alors gérer en EDD (earliest due date). Dans une telle approche, les seuils ont un impact très fort sur les performances. C'est pourquoi des méthodes d'optimisation via simulation ont été proposées pour optimiser ces règles en choisissant un seuil approprié. On cherche à déterminer le meilleur vecteur de seuils qui permet d'optimiser des performances exprimées par exemple en termes de retard moyens des pièces. Il a alors été montré, sur des problèmes de job shop, que d'excellentes performances pouvaient être obtenues.

L'approche présentée dans la suite pour aborder un problème de job shop est également basée sur des principes l'optimisation via simulation. Dans les articles cités précédemment, seuls des seuils numériques sont optimisés. La logique de changement des RP est prédéterminée de façon « experte ». L'ambition de ce travail consiste, à partir d'une approche d'optimisation via simulation à déterminer également une logique de changement de règles de priorité performante, de sorte à obtenir une heuristique d'ordonnancement dynamique

des pièces, selon une approche globale exposée dans (Mouelhi et al., 2007). Ce problème est abordé sur la base d'un job shop à 4 machines dont on cherche à minimiser le retard moyen.

Dans un premier temps nous formulons le problème du job shop étudié. Nous introduisons ensuite une approche de construction d'heuristiques basée sur l'optimisation via simulation pour le choix des règles de priorité dans le job shop. Nous montrons enfin sa pertinence et son efficacité à travers les expérimentations menées et les résultats obtenus.

2. FORMULATION DU PROBLEME DU JOB SHOP

Nous considérons un atelier job shop inspiré du travail de Fonseca et Navarrese (2002), fonctionnant de manière stochastique. Cet atelier est constitué de quatre machines différentes {M1, M2, M3, M4}, et nous avons quatre types de pièces différents {P1, P2, P3, P4}.

Les pièces suivent les cheminements suivants en fonction de leur type :

Type de pièces	Séquences
P1	M1 – M3 – M2
P2	M2- M3 - M1 – M4
P3	M2- M3 - M4
P4	M4- M2- M1

Tableau 1. Séquences des pièces en fonction de leur type

La loi d'arrivée des pièces est du type exponentiel de moyenne 5,5 minutes. Les taux de production des pièces et les lois régissant les temps opératoires par type de pièces et par machine en minutes suivant une loi normale [moyenne, écart type] sont présentés dans le tableau suivant :

Type de pièces	Proportion d'arrivée par type (%)	M1	M2	M3	M4
P1	40	[9, 1.3]	[7, 1]	[8, 1.2]	-
P2	15	[7, 1]	[8, 1.2]	[9, 1.3]	[7, 1]
P3	20	-	[9, 1.3]	[7, 1]	[9, 1.5]
P4	25	[9, 1.5]	[4, 1]	-	[8, 1]

Tableau 2. Paramètres des lois des durées opératoires

Chaque machine dans l'atelier possède sa propre file d'attente.

Le problème que nous considérons dans cet atelier est de minimiser le retard moyen. Nous recherchons pour cela des heuristiques capables de déterminer la séquence de passage des pièces sur chaque machine, en considérant l'état réel du système à chaque instant t de prise de décision.

3. APPORT DE L'OPTIMISATION VIA SIMULATION

Pour résoudre le problème que nous avons posé précédemment, nous proposons la construction d'heuristiques basées sur des règles de priorité à partir d'une logique déterminée par un réseau de neurones (Mouelhi et al., 2007). Cette logique sera au départ aléatoire, puis améliorée par optimisation via simulation, afin de devenir progressivement performante.

Dans notre approche, nous n'avons donc pas seulement des seuils numériques à optimiser, mais plus globalement l'ensemble de la logique de décision.

Le réseau de neurones est en fait considéré comme le décideur : il gère le passage des pièces sur les quatre machines de l'atelier en se basant sur l'état réel du système à chaque instant t de prise de décision. Le raisonnement (logique) qu'il utilise dépend de sa topologie (supposée fixe ici) et de ses poids, que l'on caractérise ici par un vecteur noté *W*.

Pour déterminer les poids de ce réseau, est donc la logique sous jacente, nous nous basons sur une méthode bien connue : l'optimisation via simulation (Azadivar, 1992, Fu, 1994, Andradottir, 1998, Fu, 2002). L'optimisation propose des solutions à la simulation qui les évaluent, en tenant compte de la dynamique de fonctionnement du système. Dans notre approche, nous recherchons la meilleure logique de prise de décision du réseau de neurones (figure 1) au regard de notre fonction objectif : le retard moyen.

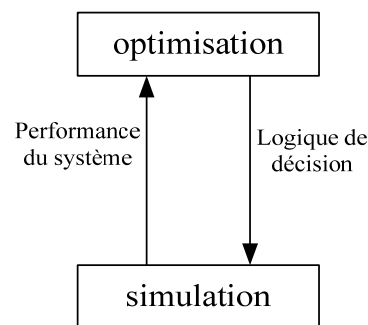


Figure 1. Optimisation de la logique de prise de décision via simulation

Les décisions étant basées sur l'état courant de l'atelier, nous définissons un vecteur D qui représente les variables d'état de celui-ci. Nous avons étudié les paramètres et les variables de l'atelier qui reflètent au mieux son état courant et qui influent sur les différentes règles de priorité qui composeront nos heuristiques. Les différentes variables de décision que nous avons retenues sont :

- Pourcentages des pièces de chaque type dans la file concernée.
- Nombre de pièces dans toutes les autres files.
- Temps opératoire moyen.
- Temps opératoire maximum.
- Temps opératoire minimum.
- Numéro de la file traitée.
- Date de livraison minimum.
- Date de livraison maximum.
- Temps de livraison moyen.
- Pourcentage de pièces dans la file qui ont une marge inférieure à 10.

S'agissant de la mise en œuvre, nous distinguons trois modules (figure 2) : le réseau de neurones, le modèle de simulation et le module d'optimisation. A chaque fois le réseau de neurone doit décider quelle pièce de la file passera, la simulation lui communique l'état $D(t)$ de l'atelier à l'instant t de la prise de décision. Ce dernier lui renvoie sa décision : la règle de priorité qu'il jugera meilleure pour faire passer la bonne pièce. La simulation applique alors cette décision. A la fin de la période de simulation, la valeur de la fonction objectif est calculée et communiquée au module d'optimisation qui modifie ainsi la logique du réseau de neurones, pour optimiser la fonction objectif de l'atelier. Plusieurs méthodes peuvent être utilisées pour résoudre ce problème d'optimisation, tels que la méthodologie de surface de réponse (RSM), des méthodes basées les méta-heuristiques, etc. Nous avons choisi le Recuit Simulé comme méthode d'optimisation (Siarry *et al.*, 1997).

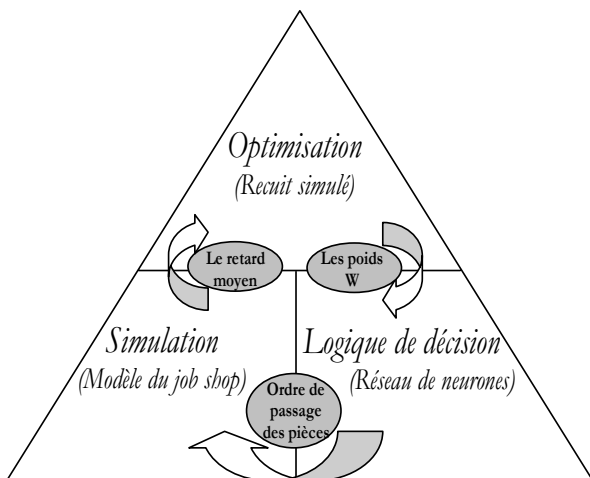


Figure 2. Interaction entre les trois modules : réseau de neurones, optimisation et simulation.

4. DESCRIPTION DU MODELE

Dans notre Job shop, les pannes et les arrêts de maintenance ne sont pas pris en compte et les temps de transferts entre les différentes parties de l'atelier sont considérés comme nuls. On considère qu'il n'y a jamais de rupture d'approvisionnement.

Les règles que nous avons utilisées pour notre atelier sont : *FIFO* (First In First Out), *LIFO* (Last In Last Out), *SPT* (Shortest Processing Time), *LPT* (Longest Process Time), *EDD* (Earliest Due Date), *PPM* : Plus Petite Marge ; la marge correspond à la différence entre la durée restant avant la livraison et la somme des temps opératoires à effectuer.

Le Passage des pièces dans les files d'attente est assuré par ces six règles de priorité. Elles ne présentent pas toutes le même intérêt mais elles permettent de réaliser des heuristiques de gestion de production variée.

Notre modèle de représentation de l'atelier est présenté ci-dessous :

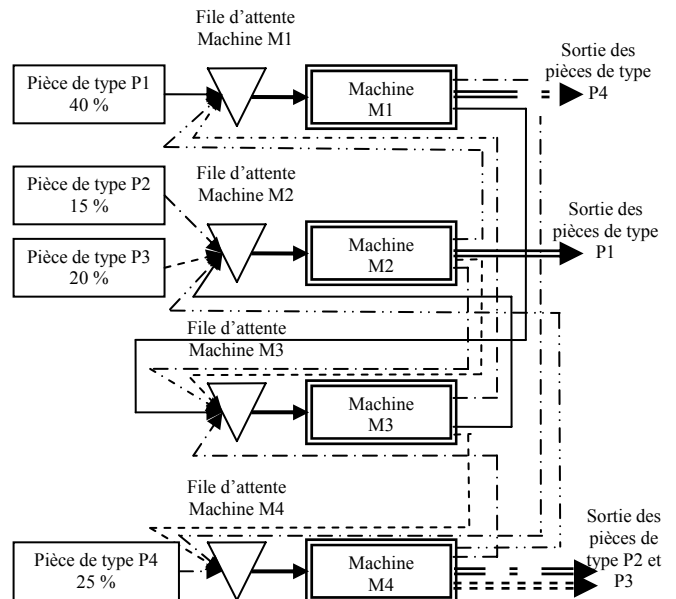


Figure 3. Représentation de l'atelier à cheminement multiples étudié et des différents flux

Pour modéliser l'atelier nous avons utilisé le logiciel de simulation ARENA 10.0. Les modules RN et optimisation sont implémentés en Visual C++. La figure 4 présente la connexion entre les trois modules qui constituent notre modèle. Le RN communique avec la simulation via des variables globales communes aux deux modules. Le recuit simulé reçoit l'information du module simulation et envoie les poids optimisés au module RN.

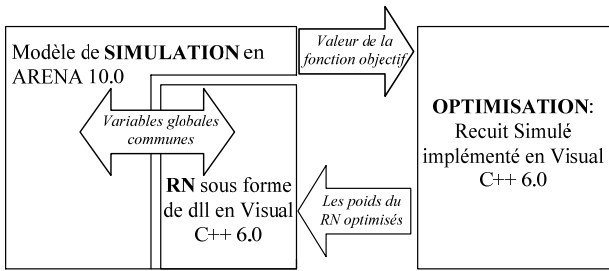


Figure 4. Connexion des différents modules du modèle.

5. EXPERIMENTATIONS

Le modèle précédent a été utilisé pour réaliser plusieurs expérimentations. L'unité de temps de la simulation est la minute et notre atelier fonctionne 24 heures sur 24 et 7 jours sur 7, sans pause même pendant les changements d'équipe. La durée de la simulation est de 40320 minutes soit 1 mois comprenant une période de transition de 1 semaine durant lesquels les statistiques de l'atelier ne sont pas comptabilisées, nous considérons 30 répliques pour chaque simulation.

Dans les expérimentations que nous avons réalisées, le réseau de neurones est constitué de trois couches : une couche d'entrée de 13 neurones qui représentent le vecteur D de l'état de l'atelier lors de la prise de décision et qui sont communiqués par la simulation, une couche cachée de 4 neurones et une couche de sortie d'un seul neurone qui représente la règle de priorité choisie (figure 5).

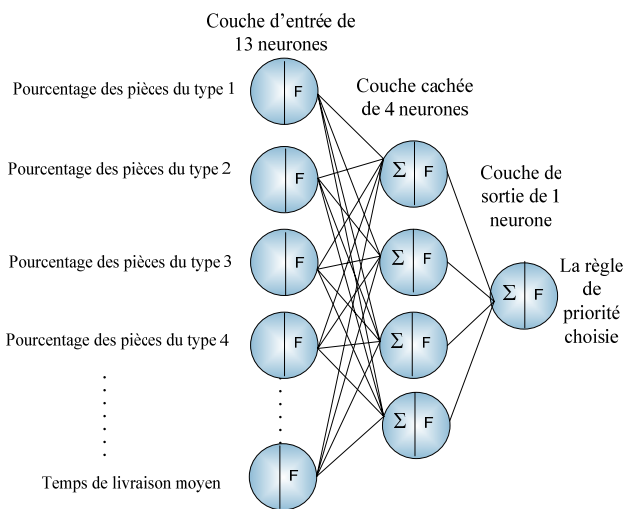


Figure 5. Le réseau de neurones utilisé pour le choix des règles de priorité

Ces couches sont reliées entre elles par 56 connexions qui constituent notre matrice W de 56 poids que nous cherchons à optimiser. Nous utiliserons une fonction tangentielle comme fonction d'activation.

Dans ces expérimentations, nous avons utilisé un recuit simulé comme méthode d'optimisation de la matrice W . Une solution initiale W_0 est tirée aléatoirement. La perturbation des solutions est basée sur des principes de voisinage variables. A partir de la matrice de la solution courante W_t à l'itération t , nm poids sont aléatoirement choisis (avec $nm \leq 56$). Ces derniers sont modifiés par l'addition de valeurs qui suivent une loi gaussienne de moyenne 0 et d'écart type σ : $N(0, \sigma)$. nm et σ sont progressivement réduits tout au long des itérations du recuit simulé, suivant la baisse de température. Ainsi, le nombre de poids choisis et les modifications sur ces derniers sont plus importants au début que vers la fin de l'optimisation.

L'approche choisie assure la connexité de l'espace de recherche. La fonction objectif, le retard moyen, est évalué par la simulation. Après plusieurs expérimentations, le paramètre de la diminution de température noté r et la température initiale ont été empiriquement déterminés. Le recuit simulé s'arrête quand la température est inférieure à ϵ , avec ϵ initialement choisie.

6. RESULTATS

Les résultats que nous présentons ci-dessous correspondent à nos premières expérimentations. Nous avons considéré une température initiale de 50000, le paramètre de diminution de température r est de 0.9 et ϵ (Condition d'arrêt) est égal à 0.001. Le temps de calcul en CPU est de 4248 secondes pour un processeur Intel Pentium® 4, CPU 3,80 GHz et 2 Go de mémoire RAM.

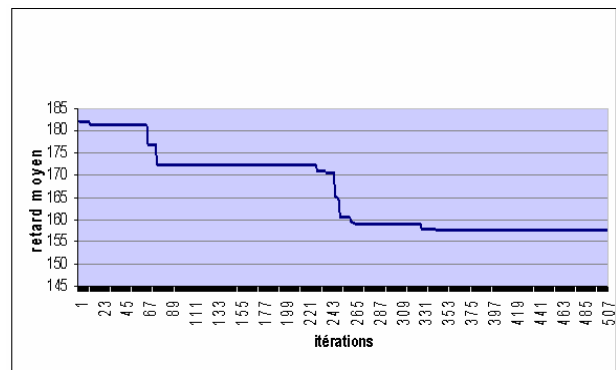


Figure 6. Evolution du retard moyen

La Figure 6 présente les résultats de la diminution du retard moyen durant les différentes itérations du recuit simulé. D'abord, nous avons calculé le retard moyen avec une matrice initiale de poids tirés aléatoirement.

Les résultats obtenus (dans les premières itérations) sont très mauvais, tels qu'ils sont attendu puisque le RN n'a pas encore appris. Ensuite, tout au long de l'optimisation de ses poids par le recuit simulé, le RN permet de diminuer le retard moyen dans l'atelier. Nous nous apercevons qu'après optimisation le RN arrive à combiner les six règles de priorité et proposer une heuristique dans le but de minimiser le retard moyen dans l'atelier. Alors, il est devenu capable de choisir les règles de priorité sans se baser ni sur une expertise préalable, ni sur un ensemble d'apprentissage.

Dans un second temps, et pour pouvoir comparer les décisions du RN aux performances des meilleures règles considérées dans l'atelier (*FIFO, LIFO, SPT, LPT, EDD et PPM*), nous avons considéré l'impact de chacune de ces règles sur le retard moyen. Le tableau 3 présente le retard moyen des six règles de priorité associées à chacune des files d'attente des quatre machines de l'atelier. Le dernier résultat dans le tableau présente la performance des décisions prises par le RN.

Règle de priorité	Le retard moyen
SPT	298,305
PPM	156,496
EDD	157,813
FIFO	182,128
LPT	581,188
FIFO	412,569
RN	152,664

Tableau 3. Comparaison entre les résultats des règles de priorité et les résultats du réseau de neurones

En se basant sur ces résultats (tableau 3), nous constatons que les règles PPM et EDD sont les règles les plus performantes pour minimiser le retard moyen dans notre atelier jobshop. Cependant, et après optimisation de ses poids, le RN propose une heuristique qui fournit des résultats performants.

7. CONCLUSION

Dans cet article, nous avons présenté une méthode pour construire des heuristiques de passages des pièces sur les différentes machines d'un atelier job shop, basée sur les règles de priorité. Pour cela nous avons recherché une logique exprimée par un réseau de neurones en utilisant une approche d'optimisation via simulation.

Les expérimentations que nous avons menées sur l'atelier job shop étudié révèlent qu'à partir d'une matrice tirée aléatoirement le RN a pu apprendre à

prendre des décisions adéquates sans disposer d'expertise préalable. Nous pensons que ces résultats peuvent encore être améliorés. Nos recherches futures intègrent de nombreuses directions. Parmi celles-ci on relève l'amélioration de l'optimisation dans R^n (Siarry *et al.*, 1997).

La limite majeure de cette méthode réside principalement dans les temps de calcul nécessaires. Nous envisageons donc une meilleure gestion des problèmes de temps de calcul en se basant sur le parallélisme et la distribution de nos expérimentations sur plusieurs processeurs (Paris et Pierreval, 2000).

REFERENCES

- Andradottir, S., 1998. *Simulation optimization*. Handbook of Simulation, Chapter 9, Wiley and Sons, New York, p. 307-333.
- Azadivar, F., 1992. A tutorial on simulation optimization. *Proceedings of the 1992 Winter Simulation Conference*, p. 198-204.
- Blackstone, J.H., D.T., Philipps, and G.L., Hogg, 1982. A State of the art survey of dispatching rules for manufacturing job shop operations, *International Journal of Production Research* 20 p. 27-45.
- Boucon, D., 1991. *Ordonnancement d'atelier, aide au choix de règles de priorité*. Thèse de Doctorat, ENSAE, Toulouse, France.
- Dusonchet F., 2003. Dynamic scheduling for production systems operating in a random environment, Thèse n° 2825 de l'école Fédérale Polytechnique de Lausanne, EPFL, Suisse.
- Fonseca, D. J. and D. Navarrese, 2002. Artificial neural networks for job shop simulation, *Advanced Engineering Informatics*, 16 p. 241-246.
- Fu, M. C., 1994. Optimization via simulation: a review. *Annals of Operation research*, 53 p. 199-247.
- Haupt, R., 1989. A survey of priority rule-based scheduling. OR Spektrum.
- Mouelhi, W., H. Pierreval and M. B. Gningue, 2007. Dynamic Selection of Dispatching Rules: a Neural Network Approach. *International Conference on Industrial Engineering and Systems management*, 324 CD-ROM.
- Mouelhi, A.-L. Huyet and H. Pierreval, 2007. Combining simulation and artificial neural networks : an overview. 6th EUROSIM Congress on Modeling and Simulation, (CD-ROM).
- Pierreval, H. and N. Mebarki, 1997. A real-Time Scheduling Approach Based on a Dynamic Selection of Dispatching Rules, *International Journal of Production Research*, 35 (6) p.1575-1591.
- Pierreval, H. and J. L. Paris, 2000. Distributed evolutionary algorithms for simulation optimization, *IEEE Transaction on Systems, Man and Cybernetics*, 20(1) p.15-24.
- Rajendran, C., and O. Holthaus, 1999. A comparative study of dispatching rules in dynamic flowshops and

- job shops. *European Journal of Operational Research*, 116 p. 156-170.
- Ramasesh, R., 1990. Dynamic Job Shop Scheduling: A Survey of Simulation research. *OMEGA International Journal of Management Science*, 18(1), p. 43-57.
- Sabuncuoglu, I., 1998. A study of scheduling rules of flexible manufacturing systems: a simulation approach, *International Journal of Production Research* 36(2) p. 527-546.
- Siarry, P., G. Berthiau, F. Durbin, J. Haussy, 1997. Enhanced simulated annealing for globally minimizing functions of many-continuous variables, *ACM Transactions on Mathematical Software*, 23(2), p. 209-228