

MAINTENANCE PREDICTIVE AVEC DES DATES DE DEBUT FLEXIBLES DANS UN FLOW SHOP HYBRIDE

Walid Besbes

Unité de Recherche GIAD-FSEG
B.P. 1081 – 3018, Sfax, Tunisie
Walid.Besbes@fsegs.rnu.tn

Taïcir Loukil

Unité de recherche LOGIQ-ISGI
B.P. 954 – 3018, Sfax, Tunisie
Taïcir.Loukil@fsegs.rnu.tn

Jacques Teghem

FPMs/MATHRO, Rue de
Houdain 9 B7000 Mons Belgique
Jacques.Teghem@fpms.ac.be

RESUME : Dans cet article, nous étudions un atelier d'ordonnancement de la production de type flow shop hybride (FSH) à k étages. Chaque étage comprend M machines dédiées. Chaque job nécessite k opérations. Contrairement à la plupart des travaux, nous supposons que chaque machine de l'atelier n'est pas disponible durant une période prédéterminée. La fonction objectif est la minimisation du makespan. Dans la littérature, la plupart des auteurs traitent le problème à deux étages tout en supposant que la préemption est autorisée. Dans ce travail, nous considérons la contrainte de non préemption tout en intégrant une nouvelle variante qui est la flexibilité de la date de début de la période de maintenance (PM). Une analyse mathématique est élaborée sur base des temps morts sur les machines et aussi des temps d'attente des jobs. Cette analyse nous a permis de comparer les deux approches, avancement et recul de la PM pour les différents scénarios dégagés dans ce type de problème.

MOTS-CLES : ordonnancement, flow shop hybride, maintenance, flexibilité.

1. INTRODUCTION

Dans la plupart des travaux proposés dans la littérature relative à la théorie de l'ordonnancement, il est souvent supposé que les ressources nécessaires à la réalisation des tâches sont toujours disponibles durant l'horizon d'étude. Cette hypothèse peut être vraie dans certains cas. Cependant, dans le contexte industriel, les machines peuvent ne pas être disponibles d'une façon continue. Cette indisponibilité peut être due à des raisons déterministes (maintenance préventive, etc.) ou aléatoires (une machine qui tombe en panne, etc.).

Dans ce papier, nous nous situons dans un contexte déterministe où les périodes d'indisponibilité des machines correspondent à des périodes de maintenance prédéterminées.

Selon la nature des tâches à exécuter, trois cas ont été présentés dans la littérature. Le premier cas appelé « résumable », (noté r) est rencontré quand un job ne peut pas se terminer avant la période de maintenance suivante ; il est alors interrompu mais ensuite, repris et continue une fois que la machine redevient disponible. Si le job peut être repris partiellement après avoir été interrompu, le cas est dit « semi résumable », (noté sr). Finalement, si l'exécution du job ne peut pas être interrompue une fois qu'elle a commencé, le cas est dit « non résumable », (noté nr).

Dans ce travail, nous considérons le troisième cas à savoir le cas non résumable. Plus précisément, un job j peut être exécuté sur une machine m^i qu'à la condition qu'il se termine avant que m^i ne devienne indisponible.

Le flow shop hybride considéré dans ce papier se compose de k étages et chaque étage comporte M machines parallèles et identiques. Chaque machine ne subit qu'une seule période de maintenance durant l'horizon de temps considéré. Nous attribuons à chaque date de début des périodes maintenance un degré de flexibilité, autrement dit, nous pouvons, dans une certaine mesure avancer ou reculer la période de maintenance par rapport à sa date de début préfixée. La flexibilité des dates de début des périodes de la maintenance préventive nous paraît très intéressante pour ce genre de problème vu que cette variante peut être appliquée dans le secteur industriel d'une part et d'une autre part, la valeur du C_{\max} est logiquement inférieure à celle dans le cas où les dates de début des périodes de la maintenance sont fixes (c.f. figures 1 et 2). L'étude proposée dans ce document peut être généralisée aussi job shop flexible. Le critère à optimiser est la minimisation du makespan qui est une mesure de performance souvent considérée dans la littérature garantissant une utilisation efficace des ressources ainsi que le désir d'assurer une satisfaction des demandes des clients. Le problème considéré peut être noté, en nous basant sur (Kubiak et al., 2002), comme suit : $FHK(P), h_{iM,1} / nr / C_{\max}$, où :

- FHK: flow shop hybride à K étages.
- P : machines parallèles identiques.
- $h_{iM,1}$: il y a M machines à chaque étage i , avec une seule période de maintenance pour chaque machine.

Nous présentons dans la section 2 un aperçu de la revue de la littérature relative aux problèmes d'ordonnancement de production sous contrainte

d'indisponibilité des machines, les travaux traitant le cas non résumable et les études proposées pour traiter le FSH lorsque les machines ne sont pas toujours disponibles. Dans la section 3, nous discutons de la complexité de ce problème. La section 4 est consacrée à la description du problème traité et aux notations utilisées. La description et l'analyse des différents scénarios dégagés en tenant compte de la flexibilité des dates des périodes de maintenance font l'objet de la section 5. Finalement, nous terminons ce papier par la conclusion et les perspectives de recherche.

2. ETAT DE L'ART

Cette deuxième section est composée de trois sous-sections. Tout d'abord, nous discutons quelques études proposées dans la littérature pour traiter des problèmes d'ordonnancement de la production sous contrainte d'indisponibilité des machines où la préemption n'est pas autorisée. Par la suite, nous survolons les travaux traitant ce même problème tout en intégrant la flexibilité au niveau des dates de début des périodes de maintenance. Finalement, nous présentons les quelques travaux proposés dans la littérature pour étudier le FSH sous contrainte d'indisponibilité des machines. Pour une revue exhaustive de la littérature en ce qui concerne le FSH en général, nous renvoyons le lecteur aux travaux de (Vignier et al., 1999), (Lin et Zhang, 1999) et plus récemment (Ruiz et Maroto, 2006).

Le problème d'ordonnancement sous contrainte d'indisponibilité des machines a été largement étudié par (Lee, 1996, 1997, 1999) et plus récemment dans (Blazewicz et al., 2001) et dans (Kubiak et al., 2002). Dans (Schmidt, 2000), l'auteur présente un panorama des algorithmes proposés avant 1998 pour résoudre ce type de problème dans le cas déterministe.

Contrairement à la plupart des travaux traitant de l'ordonnancement de la production sous contrainte d'indisponibilité des machines, nous supposons que la préemption des opérations n'est pas autorisée. Seulement quelques auteurs ont considéré cette contrainte. Dans (Espinouse et al., 1999), les auteurs ont traité un flow shop sans attente où les opérations ne sont pas préemptives en se basant sur la l'algorithme de Gilmore et Gomory (Gilmore et Gomory, 1964). Par la suite, ils ont démontré qu'il est impossible d'obtenir une solution avec une garantie de performance pour ce type de problème pour plus qu'une période d'indisponibilité. Plus récemment, dans (Liao et al., 2005), les auteurs traitent un problème d'ordonnancement à deux machines parallèles. Les périodes d'indisponibilité sont prédéfinies. Les deux modèles, résumable et non résumable, ont été considérés.

Dans le cas déterministe, la plupart des auteurs considèrent les périodes de maintenance comme étant des données du problème, autrement dit, les dates de début et les durées des périodes de maintenance sont

préfixées. Seulement quelques travaux récents ont intégré la flexibilité au niveau des dates des périodes de maintenance. Aggoune traite le problème de flow shop classique sous contrainte d'indisponibilité des machines (Aggoune, 2004). La fonction objectif est la minimisation du makespan. Dans ce travail, deux cas ont été considérés. Dans le premier, les dates de début et de fin des périodes de maintenance sont préfixées. Dans le second, les dates de début des périodes de maintenance sont supposées flexibles et elles sont déterminées tout au long de l'ordonnancement des jobs. L'auteur propose deux métaheuristiques pour résoudre le problème : un algorithme génétique et une recherche tabou. Gao et al., ont proposé un algorithme génétique hybride pour résoudre le problème de type job shop flexible avec des périodes d'indisponibilité des machines qui ne sont pas préfixées (Gao et al., 2006). Les auteurs cherchent à programmer les périodes de maintenance au mieux lors de l'ordonnancement des jobs. Plus récemment, dans (Ruiz et al., 2007), les auteurs traitent un problème d'ordonnancement de type flowshop sous contrainte d'indisponibilité des machines. A travers ce travail, Ruiz et al., cherchent à démontrer l'intérêt de considérer implicitement l'ordonnancement des périodes de maintenance dans l'exécution des jobs. Six adaptations de méthodes approchées ont été testées. Les auteurs ont démontré que toutes ces approches donnent de bons résultats pour ce type de problème.

Certes les chercheurs s'intéressent de plus en plus aux problèmes d'ordonnancement de production sous contrainte d'indisponibilité des machines, cependant, il apparaît que la plupart des travaux proposés traitent les ateliers à une machine, machines parallèles et le flow shop classique. Quelques chercheurs seulement se sont intéressés à l'étude des ateliers flexibles, notamment le flow shop hybride. Parmi les premiers travaux proposés dans ce sens, nous relevons celui d'Allaoui et Artiba. Dans cette étude, les auteurs cherchent à optimiser plusieurs critères: C_{\max} , \bar{C} , T_{\max} , \bar{T} et N_T dans un FSH à K étages sous contraintes d'indisponibilité des machines (Allaoui et Artiba, 2004). Ils proposent trois versions du recuit simulé où ils se basent à chaque fois pour générer la solution initiale sur l'une des heuristiques suivantes : SPT, LPT et EDD. Pour assurer une meilleure performance, ils intègrent un module de simulation à leur métaheuristique. Dans leur papier, Allaoui et Artiba comparent les performances de ces trois versions de leur recuit simulé proposé et aussi l'heuristique NEH dans le cas résumable et le cas non résumable. En 2005, Xie et Wang considèrent un FSH à 2 étages sous contrainte d'indisponibilité des machines (Xie et Wang, 2005). Le critère à optimiser est le makespan mais uniquement le cas non résumable a été considéré. Ils démontrent que ce problème est plus difficile qu'un FSH à 2 étages sans contrainte d'indisponibilité des machines sauf dans le cas : $F_2(P), h_{1,1} | M_1 = 1, M_2 \geq 2 | C_{\max}$, autrement dit, il s'agit du cas où il y a seulement une période

d'indisponibilité de l'unique machine du premier étage alors que les 2 machines du deuxième étage sont disponibles tout au long de l'horizon de l'ordonnancement. Récemment, Allaoui et Artiba traitent un flow shop hybride à 2 étages sous contrainte d'indisponibilité des machines pour minimiser le makespan (Allaoui et Artiba, 2006). Le FSH considéré se compose d'une machine au premier étage et M machines au second. Ils supposent que chaque machine de l'atelier ne peut pas être sujette à plus qu'une période de maintenance. Les périodes de maintenance sont préfixées et seulement le cas non résumable est considéré. Tout d'abord, ils proposent un branch and Bound pour traiter ce problème. Puis, ils calculent la performance de trois heuristiques : « LIST algorithm » (LS), « LPT algorithm » et « H-Heuristic », proposé dans (Lee et Vairaktarkis, 1994). Ils concluent que dans le pire cas, la « H-Heuristic » est plus performante que la « LPT » dans le cas où le deuxième étage ne comporte pas plus que 4 machines.

3. ANALYSE DE LA COMPLEXITE

Nous avons remarqué que la plupart des travaux proposés dans la littérature traitant le flow shop hybride considèrent ce problème à un étage (machines parallèles) ou à deux étages seulement. En effet, peu d'auteurs se sont intéressés au FSH à k étages ($k \geq 3$). Concernant le FSH à un étage ou encore le problème à machines parallèles, Karp a démontré en 1972 que le problème à deux machines et avec la minimisation du makespan comme critère d'optimisation est NP-difficile au sens ordinaire (Karp, 1972). Le problème de FSH à deux étages, où il y a une machine au premier étage et M machines ($M \geq 2$) au second est NP-difficile (Gupta, 1988). De plus, dans (Hoogeveen et al., 1996), les auteurs ont démontré que même si la préemption est autorisée, ce problème demeure NP-difficile au sens fort. Dans ces deux derniers travaux, (Gupta, 1988) et (Hoogeveen et al., 1996), les auteurs supposent que les machines sont toujours disponibles. Etant donné que le flow shop hybride considéré dans ce travail est plus général que ce problème, en effet, nous avons ($K \geq 3$) étages en plus de l'intégration de la contrainte d'indisponibilité des machines, nous pouvons dire que le problème $FHK(P), h_{iM,1} / nr / C_{\max}$ est NP-difficile au sens fort.

4. NOTATIONS ET DESCRIPTION DU PROBLEME

Nous définissons ci-dessous les notations utilisées tout au long de notre étude :

- n : le nombre de jobs ; $j = 1, 2, \dots, n$.
- k : le nombre d'étages ; $i = 1, 2, \dots, k$.
- M_i : le nombre de machines sur l'étage i ; $i = 1, 2, \dots, K$.
- m_i : l'indice de la machine m sur l'étage i ; $i = 1, 2, \dots, K$.
- $O_{i,j}$: l'opération i du job j.

- PMD : la première machine disponible sur l'étage courant.
- $[e_m, f_m]$: la fenêtre de temps qui contient la durée de la période de maintenance de la machine m.
- d_m : la durée de la période de maintenance de la machine m avec $d_m < f_m - e_m$.
- t_m^- : la date de disponibilité de la machine m avant l'exécution de la tâche de maintenance.
- t_m^+ : la date de disponibilité de la machine m après l'exécution de la tâche de maintenance.
- t_m : la date de disponibilité de la machine m.
- $p_{i,j}$: le temps d'exécution du job j à l'étage i.
- $C_{i,j}$: la date de fin du job j à l'étage i.
- tmm : le temps mort sur la machine m.
- taj : le temps d'attente du job j.
- (A) : avancement de la période de maintenance.
- (R) : recul de la période de maintenance.

L'atelier qui fait l'objet de cette étude est un flow shop hybride à k étages. Le FSH peut être considéré comme une généralisation du flow shop classique où il y a n jobs à ordonnancer sur K étages en respectant le même ordre de passage. Chaque étage se compose d'une ou plusieurs machines parallèles. Il faut qu'au moins l'un de ces étages comporte plus qu'une machine. Les n jobs et les M machines du premier étage sont tous disponibles à l'instant $t = 0$. Une machine donnée ne peut exécuter plus d'une opération à la fois. Un job donné ne peut être exécuté sur plus d'une machine à la fois.

Nous supposons que chaque machine subit une et une seule période de maintenance de durée préfixée au préalable : $d_m \subset [e_m, f_m]$. La fenêtre de temps $[e_m, f_m]$ est aussi prédéterminée au début de l'exercice. Seulement le cas non résumable est considéré dans ce papier.

5. FLEXIBILITE DES PERIODES DE MAINTENANCE : DESCRIPTION ET ANALYSE

Dans la majorité des travaux proposés dans la littérature pour traiter les problèmes d'ordonnancement de production sous contrainte d'indisponibilité des machines dans le cas déterministe, l'exécution d'une tâche de maintenance débute à un instant préfixé au préalable (figure 1). Dans la figure 1, S_m (resp. T_m) est la date de début (resp. la date de fin) de la tâche de maintenance sur la machine m. L'opération, non préemptive, O_{31} (resp. O_{22}) ne peut être exécutée sur m_1 (resp. m_2) qu'une fois la tâche de maintenance se termine. Par conséquent, nous avons un temps mort ou un temps d'inactivité de m_1 (resp. m_2) entre la date de fin de l'opération O_{21} (resp. O_{12}) et la date de début S_1 (resp. S_2) de la tâche de maintenance. Cette dernière remarque nous fait penser à une autre approche qui nous paraît plus intéressante. Dans cette approche, nous

donnons un degré de flexibilité au niveau des dates de début des périodes de maintenance. Ainsi, ces dates de début vont être déterminées implicitement au cours de l'ordonnancement des jobs. Cependant, les durées des périodes de maintenance (d_m) restent fixes et prédéterminées au préalable. A chaque d_m , nous assignons une fenêtre de temps $[e_m, f_m]$ avec $d_m < f_m - e_m$.

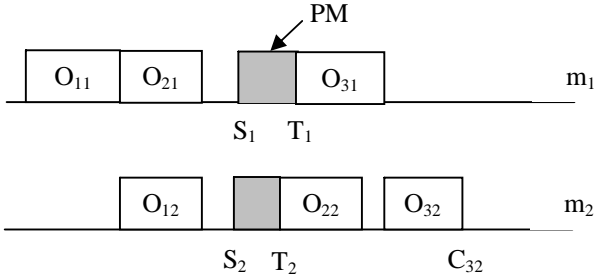


Figure 1. Périodes de maintenance avec des dates de début fixes

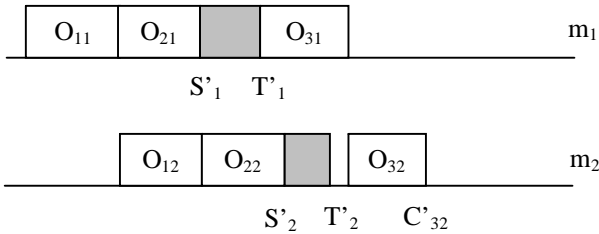


Figure 2. Périodes de maintenance avec des dates de début flexibles

Dans la figure 2, relative à des dates de début flexibles, nous supposons que :

- $C_{2,1} > e_{m1}$, de sorte que la période de maintenance de m_1 peut être avancée : (S'_1, T'_1).
- $C_{1,2} + p_{2,2} + d_{m2} < f_{m2}$, de sorte que la période de maintenance m_2 peut être retardée : (S'_2, T'_2).

Ainsi le temps mort sur m_1 est éliminé et celui sur m_2 est réduit, il en découle un avancement du job 3 ($C'_{32} < C_{32}$).

L'idée de base sera de vérifier si le premier job, non encore affecté, d'une séquence considérée peut être assigné à la PMD de l'étage courant en respectant la contrainte d'indisponibilité.

A cet égard, (c.f. l'exemple ci-dessus) deux possibilités sont à considérer lorsque $C_{i,j} > e_m$ et $C_{i,j} + d_m \leq f_m$.

Dans ce cas, nous avons le choix entre avancer la PM au plus tôt, avant l'exécution du job j ou la reculer et exécuter j en premier lieu. De plus, si l'avancement de la PM est appliqué, est ce que le job sera exécuté automatiquement sur m après la tâche de maintenance ou bien nous allons vérifier tout d'abord si cette machine m reste la première machine disponible sur cet étage après la mise à jour de sa date de disponibilité ? De même en cas de recul de la PM, faut-il exécuter automatiquement la tâche de maintenance après le job j ou bien nous allons

vérifier tout d'abord s'il y a moyen de placer un autre job sur cette machine tout en respectant sa PM ?

Par conséquent, dans le cas où l'avancement et le recul de la PM sont tous les deux possibles, nous dégagons 4 scénarios envisageables.

Remarque : pour tous les $C_{i,j}$ utilisés dans cette section, nous supposons qu'ils sont calculés après vérification de la date de disponibilité de la PMD (t_m) par rapport à la date de fin du job j ($C_{i-1,j}$) sur l'étage précédent. En effet, $C_{i,j} = \max(C_{i-1,j}, t_m) + p_{i,j}$

Dans ce qui suit, nous allons analyser les 4 scénarios dégagés sur base du temps mort sur la PMD (tmm) et du temps d'attente du job j (taj) dans le cas où $C_{i,j} > e_m$ et $C_{i,j} + d_m \leq f_m$. Pour toute cette analyse, nous plaçons donc dans le cas où $C_{i,j} > e_m$ et $C_{i,j} + d_m \leq f_m$

5.1. Scénario 1

En cas d'avancement de la PM, l'exécution de la tâche de maintenance est suivie directement par l'exécution du job j ; $t_m^+ = C_{i,j}$. Et en cas de recul de la PM, le job j est exécuté sur m avant la tâche de maintenance mais celle-ci n'est pas ensuite nécessairement exécutée. Aussi, $t_m^- = C_{i,j}$.

Dans ce premier scénario, nous avons 4 sous cas en comparant la valeur de $C_{i-1,j}$ à celles de e_m et t_m^- . Le tableau 1 représente les résultats de la comparaison entre les deux approches d'avancement et du recul de la PM pour le premier scénario en termes de tmm et taj. Les valeurs de ces deux facteurs lorsque l'avancement de la PM est appliqué, sont représentées respectivement par les lignes 2 et 3. Celles du tmm et du taj lorsque le recul est appliqué sont représentées respectivement par les lignes 6 et 7. La dernière ligne de ce tableau représente la meilleure approche à suivre dans chaque sous cas en comparant les valeurs trouvées pour le tmm et le taj pour les deux approches (A) et (R).

Dans les trois derniers sous cas, la meilleure approche à suivre est clairement le recul de la PM. Par contre, dans le premier sous cas ($C_{i-1,j} > e_m$ & $t_m^- < C_{i-1,j}$) nous ne pouvons pas dire en nous basant sur les valeurs trouvées du tmm et taj, laquelle des deux approches est meilleure vu que nous avons un tmm plus petit dans le cas d'avancement alors que le taj est plus important que dans le cas du recul.

Remarque : dans le cas où le recul est appliqué dans ce scénario, nous ne pouvons pas utiliser la notation t_m^+ comme étant la date de disponibilité de la machine vu que la tâche de maintenance n'est pas systématiquement

exécutée après $O_{i,j}$. Dans ce cas, la date de disponibilité de la machine est toujours notée t_m^- .

5.2. Scénario 2

En cas d'avancement de la PM, nous procédons de la même façon que dans le premier scénario. En cas de recul de la PM, le job j sera exécuté suivi de l'exécution de la tâche de maintenance. Mise à jour de $t_m^+ = C_{i,j} + d_m$.

Remarque : Pour ce deuxième scénario, la possibilité que $t_m^- > e_m$ ne peut pas être envisagée car une fois que nous avons $C_{i,j} > e_m$, la tâche de maintenance sera exécutée automatiquement sur cette machine et ce, quelque soit l'approche utilisée.

Il faut signaler tout d'abord que pour l'analyse de chaque scénario, nous considérons uniquement le même scénario dans tout le problème et c'est pourquoi les sous cas définis dans le second scénario sont différents de ceux du premier. En effet, pour le premier scénario et aussi pour le troisième (c.f. 5.3) et en cas de recul de la

PM (c'est seulement le job qui sera exécuté), nous aurons peut être par la suite la possibilité d'affecter un deuxième job j2 à la même machine m avant la PM en appliquant le recul de la PM une deuxième fois tout en vérifiant évidemment que $C_{i,j_2} > e_m$ & $C_{i,j_2} + d_m \leq f_m$

Donc, dans cette situation, nous avons $t_m^- \geq e_m$ puisqu'il y a déjà un premier job qui a été exécuté ou partiellement exécuté durant la fenêtre de temps $[e_m, f_m]$ allouée à la PM de la machine m. Par contre, pour le scénario 2 et aussi 4 (c.f. 5.4), nous ne pouvons pas avoir : $t_m^- \geq e_m$ étant donné que dès que la problème de la PM se pose (autrement dit, nous arrivons à la fenêtre de temps $[e_m, f_m]$), la tâche de maintenance va être exécutée quelque soit l'approche suivie.

Dans ce deuxième scénario, nous avons 3 sous cas en comparant la valeur de $C_{i-1,j}$ à celle de e_m .

Similairement au premier tableau, le tableau 2 représente les résultats de la comparaison entre les deux approches d'avancement et du recul de la PM pour le deuxième scénario.

		$C_{i-1,j} > e_m$ & $t_m^- < C_{i-1,j}$	$C_{i-1,j} > e_m$ & $t_m^- \geq C_{i-1,j}$	$C_{i-1,j} \leq e_m$ & $t_m^- < C_{i-1,j}$	$C_{i-1,j} \leq e_m$ & $t_m^- \geq C_{i-1,j}$
Avancement (A)	tmm	$\max(e_m - t_m^-, 0)$	0	$e_m - t_m^-$	$\max((e_m - t_m^-), 0)$
	taj	$\max((\max(e_m, t_m^-) + d_m) - C_{i-1,j}, 0)$	$(t_m^- + d_m) - C_{i-1,j}$	$(e_m + d_m) - C_{i-1,j}$	$(\max(e_m, t_m^-) + d_m) - C_{i-1,j}$
	$C_{i,j}$	$\max((e_m, t_m^-) + d_m) - C_{i-1,j} + p_{i,j}$	$t_m^- + d_m + p_{i,j}$	$e_m + d_m + p_{i,j}$	$\max(e_m, t_m^-) + d_m + p_{i,j}$
	t_m^+	$C_{i,j}$	$C_{i,j}$	$C_{i,j}$	$C_{i,j}$
Recul (R)	tmm	$C_{i-1,j} - t_m^-$	0	$C_{i-1,j} - t_m^-$	0
	taj	0	$t_m^- - C_{i-1,j}$	0	$t_m^- - C_{i-1,j}$
	$C_{i,j}$	$C_{i-1,j} + p_{i,j}$	$t_m^- + p_{i,j}$	$C_{i-1,j} + p_{i,j}$	$t_m^- + p_{i,j}$
	t_m^-	$C_{i,j}$	$C_{i,j}$	$C_{i,j}$	$C_{i,j}$
Choix		indécis	(R)	(R)	(R)

Tableau 1. Récapitulation des résultats du cas ($C_{i,j} > e_m$ & $C_{i,j} + d_m \leq f_m$) pour le scénario1

En nous basant sur la minimisation du tmm et du taj, le recul de la PM paraît plus intéressant dans les deux derniers sous cas : $C_{i-1,j} = e_m$ et $C_{i-1,j} < e_m$. Par contre, pour le premier sous cas, aucune des deux approches ne l'emporte sur l'autre.

Dans le cas où l'avancement de la PM est appliqué, la valeur du tmm est toujours la même pour les trois sous cas : $tmm = e_m - t_m^-$ vu que $\forall j \in n, e_m > t_m^-$.

Donc, en appliquant l'avancement de la PM pour ce scénario, l'exécution de la tâche de maintenance commence toujours au début de la fenêtre de temps allouée (e_m).

Dans le cas où le recul de la PM est appliqué :

$$tmm = C_{i-1,j} - t_m^- \text{ si } C_{i-1,j} > t_m^- \\ = 0, \text{ sinon.}$$

5.3. Scénario 3

L'exécution de la PM ne sera pas nécessairement suivie de l'exécution du job en cas d'avancement de la PM. Le job j ne sera pas exécuté. Mise à jour de $t_m^+ = \max(e_m, t_m^-) + d_m$. En cas de recul, nous procédons de la même façon que dans le premier scénario. Dans ce troisième scénario, nous avons 4 sous cas en comparant la valeur de $C_{i-1,j}$ à celle de e_m et

t_m^- . L'organisation du tableau 3 est très similaire à celle du premier tableau surtout au niveau des 4 sous cas dégagés pour ce scénario. Le tableau 3 représente les résultats de la comparaison entre les deux approches d'avancement et du recul de la PM pour ce scénario.

Dans le premier sous cas ($C_{i-1,j} > e_m$ & $t_m^- < C_{i-1,j}$), la meilleure approche à suivre est l'avancement de la PM.

Dans le deuxième sous cas ($C_{i-1,j} > e_m$ & $t_m^- \geq C_{i-1,j}$), il est difficile de dire quelle approche est meilleure que l'autre en nous basant uniquement sur le temps mort de la machine ; en effet $tmm = 0$ aussi bien en appliquant l'avancement ou le recul de la PM. Pour les deux derniers sous cas, il paraît que le recul de la PM est plus intéressant.

Remarque : le facteur temps d'attente du job (taj) n'est pas pris en considération dans ce scénario dans le cas où l'avancement de la PM est choisi. En effet, une fois la tâche de maintenance a été exécutée, nous retournons chercher la première machine disponible sur l'étage en cours. Donc, dans ce troisième scénario, le choix de la meilleure approche à suivre entre l'avancement et le recul de la PM est basé uniquement sur la valeur de tmm.

Similairement au premier scénario, nous ne pouvons pas parler de t_m^+ si le recul de la PM est appliqué.

		$C_{i-1,j} > e_m$	$C_{i-1,j} = e_m$	$C_{i-1,j} < e_m$
Avancement (A)	tmm	$e_m - t_m^-$	$e_m - t_m^-$ ou $C_{i-1,j} - t_m^-$	$e_m - t_m^-$
	taj	$\max((e_m + d_m - C_{i-1,j}), 0)$	d_m	$(e_m + d_m) - C_{i-1,j}$
	$C_{i,j}$	$\max((e_m + d_m), C_{i-1,j}) + p_{i,j}$	$C_{i-1,j} + d_m + p_{i,j}$ ou $e_m + d_m + p_{i,j}$	$e_m + d_m + p_{i,j}$
	t_m^+	$C_{i,j}$	$C_{i,j}$	$C_{i,j}$
Recul (R)	tmm	$C_{i-1,j} - t_m^-$	$e_m - t_m^-$ ou $C_{i-1,j} - t_m^-$	$\max((C_{i-1,j} - t_m^-), 0)$
	taj	0	0	$\max((t_m^- - C_{i-1,j}), 0)$
	$C_{i,j}$	$C_{i-1,j} + p_{i,j}$	$C_{i-1,j} + p_{i,j}$ ou $e_m + p_{i,j}$	$\max(C_{i-1,j}, t_m^-) + p_{i,j}$
	t_m^+	$C_{i,j} + d_m$	$C_{i,j} + d_m$	$C_{i,j} + d_m$
Choix		Indécis	(R)	(R)

Tableau 2. Récapitulation des résultats du cas ($C_{i,j} > e_m$ & $C_{i,j} + d_m \leq f_m$) pour le scénario2

		$C_{i-1,j} > e_m$ & $t_m^- < C_{i-1,j}$	$C_{i-1,j} > e_m$ & $t_m^- \geq C_{i-1,j}$	$C_{i-1,j} \leq e_m$ & $t_m^- < C_{i-1,j}$	$C_{i-1,j} \leq e_m$ & $t_m^- \geq C_{i-1,j}$
Avancement (A)	tmm	$\max(e_m - t_m^-, 0)$	0	$e_m - t_m^-$	$\max((e_m - t_m^-), 0)$
	t_m^+	$\max(e_m, t_m^-) + d_m$	$t_m^- + d_m$	$e_m + d_m$	$\max(e_m, t_m^-) + d_m$
Recul (R)	tmm	$C_{i-1,j} - t_m^-$	0	$C_{i-1,j} - t_m^-$	0
	taj	0	$t_m^- - C_{i-1,j}$	0	$t_m^- - C_{i-1,j}$
	$C_{i,j}$	$C_{i-1,j} + p_{i,j}$	$t_m^- + p_{i,j}$	$C_{i-1,j} + p_{i,j}$	$t_m^- + p_{i,j}$
	t_m^-	$C_{i,j}$	$C_{i,j}$	$C_{i,j}$	$C_{i,j}$
Choix		(A)	(A) ou (R)	(R)	(R)

Tableau 3. Récapitulation des résultats du cas ($C_{i,j} > e_m$ & $C_{i,j} + d_m \leq f_m$) pour le scénario 3

5.4. Scénario 4

En cas d'avancement de la PM, nous procédons de la même façon que dans le scénario 3 : c'est seulement la tâche de maintenance qui sera exécutée. En cas de recul de la PM, nous procédons de la même façon que dans le deuxième scénario : l'exécution du job sera suivie directement de l'exécution de la tâche de maintenance.

Remarque : Similairement au second scénario, la tâche de maintenance va être automatiquement exécutée dans les deux approches. Donc, nous ne pouvons pas avoir une date de disponibilité de la machine (t_m^-), au niveau des sous cas, qui est supérieure à la date de début de la fenêtre de temps (e_m) allouée à la tâche de maintenance.

Nous avons défini dans ce scénario deux sous cas : $C_{i-1,j} \geq e_m$ et $C_{i-1,j} < e_m$. Le tableau 4 représente une comparaison entre les deux approches (l'avancement et le recul de la PM) en termes de minimisation du temps mort sur la machine (tmm) seulement. Similairement au troisième scénario, le temps d'attente du job (taj) ne peut pas être pris en considération dans ce scénario en cas d'avancement de la PM vu que le job ne sera pas exécuté après la tâche de maintenance.

Dans le premier sous cas ($C_{i-1,j} \geq e_m$), l'approche de l'avancement de la PM est plus profitable. Dans le deuxième sous cas ($C_{i-1,j} < e_m$), le recul de la PM paraît plus intéressant (c.f. tableau 4).

D'après le tableau 4, nous remarquons que l'interprétation des résultats est la même pour les 2 sous cas lorsque l'avancement de la PM est appliqué vu que nous avons $t_m^- < e_m$ dans ce cas. Donc, l'exécution de la tâche de maintenance débute toujours à e_m en appliquant cette approche (A).

5.5. Constatations et récapitulation

En comparant les deux approches (l'avancement et le recul de la PM) dans chaque scénario en termes de minimisation du temps mort sur la machine et de temps d'attente du job, nous essayons de chercher, pour chaque scénario et pour chaque sous cas laquelle des deux approches est la plus intéressante. Notre idée de départ était de trouver le meilleur scénario et de chercher dans ce scénario la meilleure approche à suivre pour chaque sous cas. Il s'est avéré que c'est plus intéressant d'essayer de définir des sous cas identiques pour les quatre scénarios et de voir pour chaque sous cas le meilleur scénario et la meilleure approche à suivre. Pour cela, il faut redéfinir les différents sous cas en supposant que c'est possible d'appliquer différents scénarios dans le même problème. Concernant ce travail, nous avons des sous cas différents pour nos scénarios. Donc, à ce stade, nous allons essayer d'aborder ce point en comparant les scénarios deux à deux, les scénario 1 et 3 d'une part, 2 et 4 d'autre part, et par approche.

En ce qui concerne le premier et le troisième scénarios, nous obtenons pour les différents sous cas les mêmes résultats lorsque le recul de la PM est appliqué. Aussi, en cas d'avancement de la PM, nous trouvons les mêmes valeurs de tmm pour ces deux scénarios. Donc, pour chaque sous cas, ces deux scénarios présentent les mêmes valeurs de tmm. Donc, en nous basant uniquement sur le facteur tmm, nous pouvons regrouper les deux scénarios 1 et 3 en un seul. La dernière ligne du tableau 5, relative à la meilleure approche à suivre dans ce cas est identique à celle du tableau 3.

Dans le premier sous cas ($C_{i-1,j} > e_m$ & $t_m^- < C_{i-1,j}$), la meilleure approche à suivre est l'avancement de la PM. Dans le sous cas suivant ($C_{i-1,j} > e_m$ & $t_m^- \geq C_{i-1,j}$), les

deux approches (A) et (R) nous donnent les mêmes valeurs de tmm.

En ce qui concerne les deux derniers sous cas ($C_{i-1,j} \leq e_m$ & $t_m^- < C_{i-1,j}$ et $C_{i-1,j} \leq e_m$ & $t_m^- \geq C_{i-1,j}$), le recul de la PM est plus intéressant (c.f. tableau 5).

De la même façon, si nous considérons les deux scénarios 2 et 4, nous avons les mêmes valeurs de tmm si nous favorisons le recul de la PM. Aussi, en cas d'avancement de la PM, nous obtenons les mêmes résultats pour ces deux scénarios. Donc, nous ne pouvons pas comparer ces deux scénarios puisqu'ils sont identiques en termes de tmm. Donc, en nous basant uniquement sur le facteur tmm, nous pouvons regrouper les deux scénarios 2 et 4 en un seul. Par la suite, nous allons chercher la meilleure approche à suivre dans chaque sous cas. Etant donné, qu'à ce niveau, c'est seulement le tmm qui est considéré, nous pouvons grouper les deux sous cas ($C_{i-1,j} > e_m$ et

$C_{i-1,j} = e_m$) en un seul : $C_{i-1,j} \geq e_m$ pour le scénario 2 vu que nous avons les mêmes valeurs de tmm. Dans le sous cas ($C_{i-1,j} \geq e_m$), la meilleure approche à suivre est l'avancement de la PM. Dans l'autre sous cas ($C_{i-1,j} < e_m$), le recul de la PM paraît plus intéressant (c.f. tableau 6). Nous remarquons que la dernière ligne du tableau 6, relative à la meilleure approche à suivre dans ce cas est identique à celle du tableau 4.

Cette deuxième comparaison nous a permis de trouver la meilleure approche à suivre en considérant les scénarios deux à deux. Elle nous a permis aussi d'avancer la constatation suivante : $\forall t_m^-$, si $C_{i-1,j} < e_m$, le recul de la PM est plus profitable en nous basant sur la minimisation du tmm et ce, indépendamment du scénario appliqué.

		$C_{i-1,j} \geq e_m$	$C_{i-1,j} < e_m$
Avancement (A)	tmm	$e_m - t_m^-$	$e_m - t_m^-$
	t_m^+	$e_m + d_m$	$e_m + d_m$
Recul (R)	tmm	$C_{i-1,j} - t_m^-$	$\max((C_{i-1,j} - t_m^-), 0)$
	taj	0	$\max((t_m^- - C_{i-1,j}), 0)$
	$C_{i,j}$	$C_{i-1,j} + p_{i,j}$	$\max(C_{i-1,j}, t_m^-) + p_{i,j}$
	t_m^+	$C_{i,j} + d_m$	$C_{i,j} + d_m$
Choix		(A)	(R)

Tableau 4. Récapitulation des résultats du cas ($C_{i,j} > e_m$ & $C_{i,j} + d_m \leq f_m$) pour le scénario 4

			$C_{i-1,j} > e_m$ & $t_m^- < C_{i-1,j}$	$C_{i-1,j} > e_m$ & $t_m^- \geq C_{i-1,j}$	$C_{i-1,j} \leq e_m$ & $t_m^- < C_{i-1,j}$	$C_{i-1,j} \leq e_m$ & $t_m^- \geq C_{i-1,j}$
Avancement (A)	Scénario 1 (S1)	tmm	$\max(e_m - t_m^-, 0)$	0	$e_m - t_m^-$	$\max((e_m - t_m^-), 0)$
	Scénario 3 (S3)		$\max(e_m - t_m^-, 0)$	0	$e_m - t_m^-$	$\max((e_m - t_m^-), 0)$
Recul (R)	Scénario 1 (S1)	tmm	$C_{i-1,j} - t_m^-$	0	$C_{i-1,j} - t_m^-$	0
	Scénario 3 (S3)		$C_{i-1,j} - t_m^-$	0	$C_{i-1,j} - t_m^-$	0
Meilleur scénario			(S1) ou (S3)	(S1) ou (S3)	(S1) ou (S3)	(S1) ou (S3)
Meilleure approche			(A)	(A) ou (R)	(R)	(R)

Tableau 5. Comparaison des scénarios 1 et 3 en terme de tmm dans le cas $C_{i,j} > e_m$ & $C_{i,j} + d_m \leq f_m$

			$C_{i-1,j} \geq e_m$	$C_{i-1,j} < e_m$
Avancement (A)	Scénario 2 (S2)	tmm	$e_m - t_m^-$	$e_m - t_m^-$
	Scénario 4 (S4)		$e_m - t_m^-$	$e_m - t_m^-$
Recul (R)	Scénario 2 (S2)	tmm	$C_{i-1,j} - t_m^-$	$\max((C_{i-1,j} - t_m^-), 0)$
	Scénario 4 (S4)		$C_{i-1,j} - t_m^-$	$\max((C_{i-1,j} - t_m^-), 0)$
Meilleur scénario			(S2) ou (S4)	(S2) ou (S4)
Meilleure approche			(A)	(R)

Tableau 6. Comparaison des scénarios 2 et 4 en terme de tmm dans le cas $C_{i,j} > e_m$ & $C_{i,j} + d_m \leq f_m$

6. CONCLUSION

Dans cet article, nous avons analysé le problème de flow shop hybride à k étages, dans lequel chaque machine est contrainte à une période d'indisponibilité. Nous supposons que les dates de début des périodes de maintenance sont flexibles ce qui nous permet d'avoir un makespan plus intéressant.

Dans le cas ($C_{i,j} > e_m$ & $C_{i,j} + d_m \leq f_m$), deux alternatives sont possibles. En effet, nous avons le choix entre avancer ou reculer la période de maintenance par rapport à l'exécution du job en cours. En plus, dans ce cas bien précis, nous avons défini quatre scénarios envisageables concernant l'étape suivante une fois que l'exécution de la tâche de maintenance ou du job a été décidée. Nous avons élaboré une analyse en nous basant sur les valeurs du temps d'attente du job et surtout du temps mort sur la machine pour voir quelle est la meilleure approche à suivre pour chaque scénario et pour chaque sous cas défini.

Une deuxième réflexion sur la redéfinition des sous cas dégagés qui doivent être identiques pour tous les scénarios, nous permettra de trouver le meilleur scénario et la meilleure approche à suivre dans chaque sous cas.

Une implémentation d'un algorithme génétique basée sur la dernière analyse sera par la suite élaborée afin d'effectuer des expérimentations numériques pour valider les résultats trouvés.

Nous nous intéressons par la suite à la robustesse de la meilleure approche trouvée dans chaque cas. Nous nous basons sur cette étude pour traiter ce type de problème dans un contexte stochastique.

REFERENCES

Aggoune, R., 2004. Minimizing the makespan for the flow shop scheduling problem with availability constraints. *European Journal of Operational Research*, 153, p. 534-543.

Allaoui, H., and A. Artiba, 2004. Integrating simulation and optimization to schedule a hybrid flow shop

with maintenance constraints. *Computers & Industrial Engineering*, 47, p. 431-450.

Allaoui, H., and A. Artiba, 2006. Scheduling two-stage hybrid flow shop with availability constraints. *Computers & Operations Research*, 33, p. 1399-1419.

Blazewicz, J., J. Breit, P. Formanowicz, W. Kubiak, and G. Schmidt, 2001. Heuristic algorithms for the two-machine flow shop problem with limited machine availability. *Omega*, 29, p. 599-608.

Espinouse, M.L., P. Formanowicz, and B. Penz, 1999. Minimizing the makespan in the two-machine no-wait flow-shop. *Computers and Industrial Engineering*, 37, p. 497-500.

Gao, J., M. Gen, and L. Sun, 2006. Scheduling jobs and maintenance in flexible job shop with hybrid genetic algorithm. *Journal of Intelligent Manufacturing*, 17, p. 493-507.

Gilmore, P.C., and R.E. Gomory, 1964. Sequencing a one state-variable machine: a solvable case of the traveling salesman problem. *Operations Research*, 12, p. 655-679.

Gupta, J.N.D., 1988. Two stage hybrid flow shop scheduling problem. *Journal of Operational Research Society*, 39, p. 359-364.

Hoogeveen, J.A., J.K. Lenstra, and B. Veltman, 1996. Preemptive scheduling in a two-stage multiprocessor flow shop is NP-hard. *European Journal of Operational Research*, 89, p. 172-175.

Karp, R.M., 1972. *Reducibility among combinatorial problems*. In: Miller RE, Thatcher JW, editors. Complexity of computers computations. New York: Plenum Press, p. 85-103.

Kubiak, W., J. Blazewicz, P. Formanowicz, J. Breit, and G. Schmidt, 2002. Two-machine flow shop with limited machine availability. *European Journal of Operational Research*, 136, p. 528-540.

Lee, C.Y., and G.L. Vairaktarkis, 1994. Minimizing makespan in hybrid flowshops. *Operational Research Letters*, 16, p. 149-158.

Lee, C.Y., 1996. Machine scheduling with an availability constraint. *Journal of Global Optimization*, 9, p. 395-416.

- Lee, C.Y., 1997. Minimizing the makespan in two-machine flowshop scheduling problem with an availability constraint. *Operations Research Letters*, 20, p. 129-139.
- Lee, C.Y., 1999. Two-machine flowshop scheduling with availability constraints. *European Journal of Operational Research*, 114, p. 129-139.
- Liao, C.J., D.L. Shyur, and C.H. Lin, 2005. Makespan Minimization for two parallel machines with an availability constraint. *European Journal of Operational Research*, 160, p. 445-456.
- Linn, R. and W. Zhang, 1999. Hybrid flowshop scheduling: A survey, *Computers and Industrial Engineering*, 37, p. 57-61.
- Ruiz, R., and C. Maroto, 2006. A genetic algorithm for hybrid flowshops with sequence dependent setup times and machine eligibility. *European Journal of Operational Research*, 169, p. 781-800.
- Ruiz, R., J.C. Garcia-Diaz, and C. Maroto, 2007. Considering scheduling and preventive maintenance in the flowshop sequencing problem. *Computers & Operations Research*, 34, p. 3314-3330.
- Schmidt, G., 2000. Scheduling with limited machine availability. *European Journal of Operational Research*, 121, p. 1-15.
- Vignier, A., J. C. Billaut, and C. Proust, 1999. Les problèmes d'ordonnancement de type flow-shop hybride : état de l'art, *RAIRO Rech. Opér.*, 33, p. 177-183.
- Xie, J., and X. Wang, 2005. Complexity and algorithms for two-stage flexible flowshop scheduling with availability constraints. *Computers and Mathematics Applications*, 50, p. 1629-1638.