



HAL
open science

FLOW SHOP HYBRIDE À DEUX ÉTAGES AVEC MACHINES DÉDIÉES ET DATES DE DISPONIBILITÉ DES JOBS

Asma Ouled Bedhief, Najoua Dridi

► **To cite this version:**

Asma Ouled Bedhief, Najoua Dridi. FLOW SHOP HYBRIDE À DEUX ÉTAGES AVEC MACHINES DÉDIÉES ET DATES DE DISPONIBILITÉ DES JOBS . MOSIM 2014, 10ème Conférence Francophone de Modélisation, Optimisation et Simulation, Nov 2014, Nancy, France. <hal-01166681>

HAL Id: hal-01166681

<https://hal.science/hal-01166681v1>

Submitted on 23 Jun 2015

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire HAL, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



HAL Authorization

FLOW SHOP HYBRIDE A DEUX ETAGES AVEC MACHINES DEDIEES ET DATES DE DISPONIBILITE DES JOBS

A. OULED BEDHIEF ET N.DRIDI

OASIS - Ecole Nationale d'Ingénieurs de Tunis, Université Tunis el Manar B.P. 37, Le belvédère 1002, Tunis.
Bedhief.asma@gmail.com, najoua.dridi@enit.rnu.tn

RESUME : Dans cet article, nous traitons un problème de flow shop hybride à deux étages avec machines dédiées et dates de disponibilité des jobs. Nous cherchons un ordonnancement qui minimise la date d'achèvement du dernier job ou makespan.

Nous proposons six variantes d'une métaheuristique de recherche locale: La recherche tabou. Ces variantes diffèrent par la définition du voisinage et le mode de gestion de la liste tabou. Afin de bien explorer le domaine des solutions, un travail permettant le calibrage des différents paramètres est effectué. Enfin, une étude expérimentale sur différentes familles d'instances nous permet d'évaluer ces propositions.

MOTS-CLES: Ordonnancement, Flow shop hybride, Machines dédiées, Recherche tabou

1- INTRODUCTION

Nous considérons le problème d'ordonnancement de type flow shop hybride à deux étages avec machines dédiées et dates de disponibilité des jobs. Un tel problème considère un atelier qui se compose de deux étages avec, dans le premier étage, une seule machine, M_c , commune à tous les jobs et, au deuxième étage deux machines dédiées $M_k, k \in \{1,2\}$ (figure 1). Indépendamment de son type, chaque job doit passer, obligatoirement, par la machine M_c afin d'exécuter la première opération de sa gamme de fabrication. Ensuite, il va suivre un traitement particulier en fonction de sa destination finale. Il sera exécuté sur la machine du deuxième étage correspondante à son type. Nous supposons que chaque job i n'est disponible qu'à partir de la date r_i .

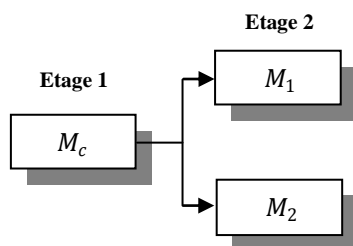


Figure 1 : Schéma du processus de production

La configuration étudiée se situe entre le flow shop hybride (FSH) et le flow shop. Elle peut être considérée comme un flow shop hybride puisque l'un des étages contient plus d'une machine. Mais, contrairement au FSH, le problème d'affectation des jobs aux machines ne se pose pas, en plus, en se limitant à un seul type de jobs, le problème devient un flow shop. Nous pouvons alors noter ce

problème en se référant à la notation utilisée dans [18] : $2FHD|r_i, 1,2|C_{\max}$, ou encore à celle utilisée dans [10] : $F_3|T = 2, r_i|C_{\max}$.

Ce type d'ateliers peut être rencontré dans plusieurs cas industriels. Citons, par exemple, les industries pharmaceutiques où un médicament peut exister sous plusieurs formes (gélules, sirop ou poudre en sachets). D'abord, la matière première subit des transformations qui ne dépendent pas de sa forme finale, par la suite, et selon la nature du produit fini, elle suivra la ligne correspondante à la forme demandée. Un autre exemple peut être trouvé dans l'industrie chimique, la manufacture des câbles électriques, l'industrie agroalimentaire, la fabrication du verre ...

Le reste de l'article est organisé comme suit. Dans la section 2, nous développons une brève revue de littérature traitant les problèmes d'ateliers avec machines dédiées. La section 3 introduit les notations et la section 4 décrit les différentes variantes de recherche tabou que nous proposons. Le paramétrage de ces variantes est présenté dans la section 5. La section 6 présente les bornes inférieures utilisées pour évaluer la qualité de la solution obtenue. Dans la section 7, les résultats expérimentaux sont présentés et analysés. Nous terminons ce travail par une conclusion et quelques perspectives.

2- ETAT DE L'ART

La configuration la plus simple de $F_{m+1}|T = m|C_{\max}$ (pour $m=2$) est NP-difficile au sens fort [11]. C'est un cas particulier du problème que nous

études (en prenant tous les $r_i = 0$), d'où $F_3|T = 2, r_i |C_{max}$ est NP-difficile au sens fort.

La plupart des travaux considérant des ateliers avec des machines dédiées ont porté sur les configurations à deux étages.

Certains travaux ont proposé des méthodes exactes. Dans [18], la programmation linéaire en variables mixtes est utilisée pour l'ordonnancement d'un atelier composé de machines dédiées dans les deux étages. Dans [12], les auteurs proposent un programme linéaire en nombres entiers qui minimise le nombre pondéré de travaux en retard dans un flow shop hybride à deux étages, avec m machines dédiées au deuxième étage. Des procédures par séparation et évaluation ont aussi été proposées pour résoudre ce problème en minimisant le makespan [6, 16]. La programmation dynamique est proposée dans [15] pour la résolution du problème $F_3|T = 2|C_{max}$ ou encore dans [12] pour la minimisation du nombre pondéré de travaux en retard.

D'autres efforts de recherche se sont dirigés vers la proposition des méthodes approchées. L'une des principales heuristiques (H_{OLC}) pour le problème $F_3|T = 2|C_{max}$, équivalent à $2FHD|1,2|C_{max}$, a été proposée dans [13], elle utilise l'algorithme de Johnson [9] pour chaque type séparément et déduit un ordre de passage des jobs sur la machine du premier étage. La généralisation de cette heuristique au cas m quelconque et l'étude de son comportement sur différentes familles de données a permis de mettre en évidence que la qualité des résultats obtenus par H_{OLC} dépend de la nature des données [2]. Les auteurs ont alors proposé une heuristique (H_{HDH}). L'étude comparative menée dans [2] a montré que pour certaines familles de données, il est très facile de trouver la solution optimale avec l'une ou l'autre des heuristiques, alors que pour d'autres familles, H_{OLC} et H_{HDH} sont complémentaires.

Trois autres heuristiques ont aussi été proposées dans [7] dans le but de résoudre le problème du flow shop hybride à deux étages avec m machines dédiées au premier étage. L'objectif est de minimiser le makespan. Dans [18], deux méthodes approchées ont été développées pour minimiser le makespan dans un flow shop hybride avec machines dédiées au premier et au deuxième étage. La première méthode est une heuristique basée sur la règle de Johnson et la deuxième est un algorithme génétique.

Quelques rares travaux ont considéré plus de deux étages. Citons par exemple l'heuristique développée dans [14], basée sur la programmation dynamique, qui minimise le makespan dans un flow shop hybride à trois étages avec deux machines dédiées au deuxième étage.

Quant aux métaheuristiques, elles sont plutôt rares dans la littérature sur les problèmes d'ordonnancement avec machines dédiées. A notre

connaissance, la méthode tabou, qui a été proposée dans [5] par exemple, pour résoudre le problème FSH où elle a donné des résultats prometteurs, n'a pas été utilisée pour le problème d'ateliers contenant des machines dédiées.. Nous avons alors choisi de tester la performance de la recherche tabou pour la minimisation du makespan dans le cas de flow shop hybride avec machines dédiées.

3- NOTATIONS

Les notations utilisées sont les suivantes:

- n Nombre de jobs.
- J Ensemble de n jobs, $J = \{J_1, J_2 \dots J_n\}$
- n_k Nombre de jobs de type k , $k \in \{1, 2\}$
- T_k Ensemble de jobs de type k , $k \in \{1, 2\}$
- M_c La machine commune du premier étage.
- M_k La machine dédiée aux jobs de type k , $k \in \{1, 2\}$.
- r_i Date de disponibilité du job i .
- a_i Temps opératoire du job i sur M_c .
- b_i Temps opératoire du job i sur M_k où $J_i \in T_k$.
- $C_{max}(\pi)$ Makespan de la solution π .
- C_{max}^* Solution optimale.
- V_π Voisinage de la solution π .

4- METHODE DE RESOLUTION

Pour résoudre ce problème NP difficile, nous développons six variantes d'une métaheuristique de recherche locale : la recherche tabou. La notion de voisinage V_π d'une solution π , définie à l'aide d'un opérateur de changement, est utilisée. Les opérateurs de changement les plus fréquemment cités dans la littérature [1,3,17] sont:

- **Opérateur de changement (APEX)**

APEX (Adjacent Pairwise Exchange) : C'est un opérateur qui autorise la permutation de deux jobs adjacents. Il a un voisinage de $n - 1$ solutions réalisables.

- **Opérateur de changement (EX)**

EX (Exchange) : C'est une généralisation d'APEX. Il autorise la permutation de deux jobs quelconques. Il a un voisinage de $n(n - 1)/2$ solutions réalisables.

- **Opérateur de changement (DSH)**

DSH (Double Shift) : C'est un opérateur qui permet d'insérer un job en une position donnée en gardant pour les autres jobs le même ordre de départ. Cet opérateur donne un voisinage de $n(n - 1)$ solutions réalisables.

Le principe de la recherche tabou est d'examiner, à chaque itération, le voisinage de la solution courante et de se déplacer sur la meilleure solution de ce voisinage. Afin d'éviter de cycler, une liste tabou est utilisée. Elle contient les dernières solutions ou leurs caractéristiques, pour les exclure pendant un nombre d'itérations égal à la longueur de cette liste.

Les méthodes les plus utilisées pour gérer une liste tabou sont [4,8,3] :

- l'enregistrement des traces des derniers mouvements effectués : une solution obtenue par un mouvement appartenant à la liste tabou ne sera pas retenue.
- l'enregistrement de la valeur de la fonction objectif : on ne se déplace pas sur une solution ayant un makespan qui figure dans la liste tabou.

La liste tabou est gérée suivant la règle FIFO (first in first out).

En fonction des opérateurs de changement et des méthodes de gestion de la liste tabou, nous avons implémenté six variantes de recherche tabou. Chacune opère avec une combinaison d'un opérateur de changement et une méthode de gestion.

MG/OP	APEX	EX	DSH
Par mouvement	Tabou1	Tabou2	Tabou3
Par makespan	Tabou4	Tabou5	Tabou6

Tableau 1: Procédures de la recherche tabou

OP* : Opérateur de changement.

MG* : Méthode de gestion.

La procédure Tabou1 consiste, par exemple, à générer l'ensemble de voisinage par l'opérateur de changement (APEX) et la liste tabou avec la méthode d'interdiction des mouvements.

La démarche générale de ces différentes procédures est présentée dans l'algorithme 1.

Algorithme 1: Recherche tabou

Etape 1: Initialisation

π_0 : solution initiale, T : Liste tabou
 $\pi \leftarrow \pi_0$; $\pi^* \leftarrow \pi_0$; $C_{max}^* \leftarrow C_{max}(\pi_0)$; $T = \emptyset$

Etape 2: Soit $\pi' \in V_\pi$ tel que $C_{max}(\pi') \leq C_{max}(\pi)$
 $\forall v \in V_\pi$ et $\pi' \notin T$, Faire : $\pi \leftarrow \pi'$ et Mettre à jour la liste tabou T .

Si $C_{max}(\pi') < C_{max}^*$ alors $\pi^* \leftarrow \pi'$ et $C_{max}^* \leftarrow C_{max}(\pi')$

Etape 3: Test d'arrêt

Si la condition d'arrêt n'est pas satisfaite retour à l'étape 2.

5- PARAMETRAGE

L'efficacité de la recherche tabou dépend fortement de l'utilisation de certaines options et du choix des différents paramètres. L'ensemble des paramètres de la recherche tabou sont : la longueur de la liste tabou et le critère d'arrêt. Les options qui peuvent agir sur la qualité de la solution et que nous allons

utiliser sont le critère d'aspiration et l'application de la diversification.

• **Longueur de la liste tabou**

Les tests effectués ont montré qu'une liste tabou de longueur $L = 10$ offre le meilleur compromis entre la nécessité d'éviter le phénomène d'allure cyclique et le temps de calcul.

• **Critères d'arrêt**

Le choix du nombre d'itérations à effectuer s'est porté sur 100 itérations, de façon à permettre à chaque procédure tabou d'appliquer la diversification.

• **Critère d'aspiration**

Nous cherchons, normalement, le meilleur mouvement non tabou. Cependant, dans le cas où un mouvement tabou permet d'obtenir une meilleure solution que toutes celles déjà parcourues, le critère d'aspiration peut lever le statut tabou.

• **La diversification**

La diversification permet d'échapper aux phénomènes cycliques, qui peuvent avoir lieu à partir d'un certain nombre d'itérations, en réinitialisant la recherche à partir d'une solution quelconque. Si la diversification est appliquée trop tôt, on pourrait perdre des solutions prometteuses voisines de la solution courante. Il serait alors intéressant de trouver le nombre d'itérations à partir duquel le phénomène cyclique apparaît. Pour cela, nous avons associé à chaque solution π une empreinte: $w_\pi = \sum_{i=1}^n \pi_i * i$. Ainsi, l'absence de phénomène cyclique sur les empreintes traduit une bonne exploration du voisinage. Dans le cas contraire, ceci peut traduire un phénomène cyclique sur les solutions visitées et donc des itérations effectuées sans aucun apport.

La figure 2 présente les empreintes des solutions examinées par Tabou1 pour une seule instance avec 100 itérations.

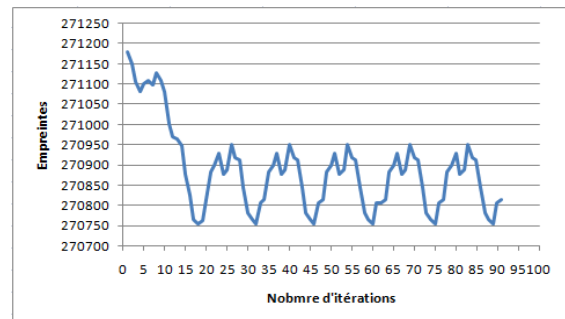


Figure 2 : Empreintes des solutions de Tabou1 pour 100 itérations

Nous remarquons que les solutions examinées suivent un cycle de 15 itérations après lesquelles la procédure revient au même point de départ. Nous

avons alors examiné le comportement du makespan pendant ces 100 itérations et nous remarquons que la valeur du makespan reste figée dès les toutes premières itérations. Par conséquent, la majorité des itérations n'étaient qu'une perte de temps de calcul (figure 3).

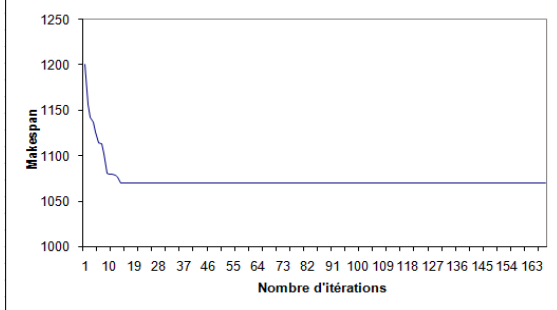


Figure 3: Evolution du Makespan du Tabou1 pour $n = 100$

Pour explorer mieux de nouvelles zones de l'espace de recherche et donner plus de chance d'éviter cette allure cyclique, nous proposons d'appliquer la diversification. Pour y parvenir, nous avons choisi de construire à partir de la permutation courante, une solution aléatoire avec laquelle nous commençons de nouveau l'exploration de l'espace de recherche.

La figure 4 présente le comportement du makespan après l'application de la diversification.

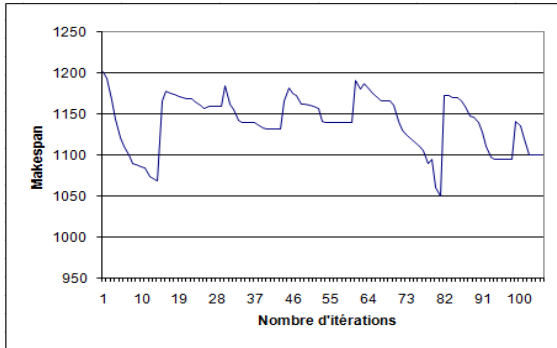


Figure 4 : Evolution du Makespan du Tabou 1 avec diversification, pour $n = 100$ et $[0,100]$.

Algorithme 2: Recherche tabou avec diversification

Etape 1: Initialisation: π_0 : solution initiale, N : nombre d'itérations avant d'appliquer la diversification, $i \leftarrow 0$; $\pi \leftarrow \pi_0$; $\pi^* \leftarrow \pi_0$; $C_{max}^* \leftarrow C_{max}(\pi_0)$; $T = \emptyset$

Etape 2: Soit $\pi' \in V_\pi$ tel que $\forall v \in V_\pi, C_{max}(\pi') \leq C_{max}(v)$ et $\pi' \notin T$; faire : $\pi \leftarrow \pi'$ et Mettre à jour la liste tabou T .

Si $C_{max}(\pi') < C_{max}^*$ alors $\pi^* \leftarrow \pi'$ et $C_{max}^* \leftarrow C_{max}(\pi')$

Etape 3: Diversification

Si $i = N$ alors construire une solution aléatoire SA à partir de π ; faire $\pi \leftarrow SA$ et $i \leftarrow 0$

Sinon $i \leftarrow i + 1$

Etape 4: Test d'arrêt

Si la condition d'arrêt n'est pas satisfaite retour à l'étape 2.

6- BORNES INERIEURES

Nous avons adapté les bornes inférieures, proposées dans la littérature [2], à notre cas en introduisant les dates de disponibilité des jobs:

$$BI_1 = \min_{J_i \in J} r_i + \sum_{J_i \in J} a_i + \min_{J_i \in J} b_i$$

$$BI_2 = \min_{J_i \in J} r_i + \left(\min_{J_i \in T_1} a_i + \min_{J_i \in T_2} a_i \right) + \min_{k \in \{1|2\}} \sum_{J_i \in T_k} b_i$$

$$BI_3 = \max_{J_i \in J} (r_i + a_i + b_i)$$

$$BI_4 = \max_{k \in \{1|2\}} \left(\min_{J_i \in T_k} r_i + \min_{J_i \in T_k} a_i + \sum_{J_i \in T_k} b_i \right)$$

BI_1 et BI_3 sont deux bornes inférieures évidentes, BI_2 tient compte du fait que sur l'une des machines dédiées l'exécution ne peut commencer qu'au plus tôt à la sortie du deuxième job de M_c . BI_4 retient la plus grande valeur obtenue lorsque chaque machine dédiée met un temps d'attente minimal.

La borne inférieure retenue est donnée par:

$$BI = \max_{1 \leq i \leq 4} BI_i$$

7- RESULTATS EXPERIMENTATUX

7.1 Données

Afin d'évaluer expérimentalement notre proposition, nous avons retenu les mêmes caractéristiques des données utilisées dans [2]. Ainsi, nous avons généré une batterie d'instances aléatoires qui appartiennent à différentes familles de F1 à F6 [2]. Pour F1, les durées opératoires sur les différentes machines sont générées dans $[1,20]$. Dans la famille F2, les temps opératoires sur M_c , sont générés dans $[1,20]$ et, ceux des deux machines dédiées dans $[1,40]$. Pour F3, les temps opératoires sur M_c , sont générés dans $[1,20]$ et, ceux des machines dédiées dans $[20,40]$. Ainsi, pour F1, les temps opératoires sont tous du même ordre de grandeur, quant à F2 et F3, les temps opératoires sur les machines dédiées sont plus importants que celles sur M_c , ce qui tend à équilibrer la charge globale sur les différentes machines.

La famille F4 traite des jobs pour lesquels la deuxième opération nécessite plus de temps que la première. De ce fait, nous avons généré des instances dans $[1,20]$ avec $a_i \leq b_i$. Enfin, pour les familles F5 et F6, nous avons introduit un entier γ comme un temps supplémentaire afin d'accentuer la

particularité des cas testés. Dans F5, a_i est générée dans $[1,20]$ et $b_i = a_i + \gamma$ où $\gamma=5$, tandis que pour F6 nous avons pris $\gamma=10$. (tableau 2).

Famille	a_i	b_i	Condition
F1	[1,20]	[1,20]	-
F2	[1,20]	[1,40]	-
F3	[1,20]	[20,40]	-
F4	[1,20]	[1,20]	$a_i \leq b_i$
F5	[1,20]	$a_i + \gamma$	$\gamma = 5$
F6	[1,20]	$a_i + \gamma$	$\gamma = 10$

Tableau 2: Familles des problèmes test

Les dates de disponibilité des jobs sont générées dans deux intervalles:

- $r_i \in [0,100]$
- $r_i \in [0, \text{nombre des jobs} * 10]$.

7.2 Analyse des résultats

Les variantes de la recherche tabou ont été programmées en langage C++. Les simulations numériques ont été effectuées sur un micro-processeur de 2.26 GHz.

Les critères d'évaluation auxquels nous avons eu recours afin d'évaluer la résolution métaheuristique sont essentiellement:

- La moyenne des écarts, E_{moy} , à la Borne inférieure: $\text{Ecart} = (C_{max} - BI) * 100/BI$
- La moyenne des temps de résolution, T_{moy} , de la procédure tabou. Elle est calculé en secondes.
- L'écart-type σ qui permet d'avoir une meilleure idée sur la dispersion des valeurs des écarts calculés dans les 20 instances testées.

Chaque famille de F1 à F6 est testée pour 10 tailles différentes ($n=50, 100, 150 \dots 500$).

20 instances sont testées pour chaque taille dans un intervalle de r_i .

Avant de faire les simulations nécessaires, nous avons effectué différents tests afin de déterminer le nombre d'itérations à partir duquel la valeur du makespan reste figée et, après lequel nous appliquons la diversification.

Vu que les procédures Tabou4, Tabou5 et Tabou6 utilisent, respectivement, les mêmes opérateurs de changement que Tabou1, Tabou2, et Tabou3, elles auront les mêmes paramétrages que ces dernières.

Nombre d'itérations choisi	Tabou1 & Tabou4	Tabou2 & Tabou5	Tabou3 & Tabou6
F1	3	3	6
F2	10	6	11
F3	5	4	7
F4	7	12	12
F5	15	18	10
F6	12	20	11

Tableau 3: Nombre d'itérations pour appliquer la diversification

Les résultats des simulations des procédures tabou sont présentés ci-dessous.

$r_i \in [0, 100]$					
F1		150	250	350	450
Tabou1	E_{moy}	0.002	0.00	0.00	0.00
	σ	0.001	0.00	0.00	0.00
	T_{moy}	0.36	0.59	0.73	0.90
Tabou2	E_{moy}	0.00	0.00	0.00	0.00
	σ	0.00	0.00	0.00	0.00
	T_{moy}	0.95	2	3	6
Tabou3	E_{moy}	0.00	0.00	0.00	0.00
	σ	0.00	0.00	0.00	0.00
	T_{moy}	1	3	6	11
Tabou4	E_{moy}	0.003	0.00	0.00	0.00
	σ	0.001	0.00	0.00	0.00
	T_{moy}	0.36	0.55	0.73	0.91
Tabou5	E_{moy}	0.00	0.00	0.00	0.00
	σ	0.00	0.00	0.00	0.00
	T_{moy}	0.95	2	3	6
Tabou6	E_{moy}	0.00	0.00	0.00	0.00
	σ	0.00	0.00	0.00	0.00
	T_{moy}	1	3	7	11
$r_i \in [0, \text{nbjobs} * 10]$					
Tabou1	E_{moy}	0.14	0.09	0.07	0.05
	σ	0.08	0.02	0.03	0.01
	T_{moy}	0.38	1	2	3
Tabou2	E_{moy}	0.09	0.05	0.03	0.02
	σ	0.04	0.02	0.01	0.008
	T_{moy}	55	159	305	528
Tabou3	E_{moy}	0.10	0.07	0.04	0.03
	σ	0.08	0.05	0.01	0.01
	T_{moy}	111	308	621	1048
Tabou4	E_{moy}	0.17	0.11	0.09	0.07
	σ	0.08	0.07	0.07	0.02
	T_{moy}	0.36	0.55	0.70	0.84
Tabou5	E_{moy}	0.08	0.06	0.04	0.03
	σ	0.06	0.03	0.01	0.01
	T_{moy}	54	149	299	504
Tabou6	E_{moy}	0.08	0.05	0.04	0.02
	σ	0.03	0.06	0.01	0.009
	T_{moy}	109	306	614	1059

Tableau 4 : Résultats des simulations, F1

$r_i \in [0, 100]$					
F2		150	250	350	450
Tabou1	E_{moy}	0.17	0.09	0.06	0.04
	σ	0.12	0.03	0.02	0.07
	T_{moy}	1	1	2	3
Tabou2	E_{moy}	0.09	0.07	0.04	0.03
	σ	0.05	0.03	0.09	0.01
	T_{moy}	55	158	317	539
Tabou3	E_{moy}	0.08	0.04	0.04	0.02
	σ	0.03	0.01	0.17	0.08
	T_{moy}	110	385	608	1055
Tabou4	E_{moy}	0.21	0.10	0.09	0.07
	σ	0.10	0.14	0.02	0.03
	T_{moy}	0.37	0.55	0.73	0.91
Tabou5	E_{moy}	0.11	0.08	0.05	0.04
	σ	0.09	0.03	0.01	0.06
	T_{moy}	55	154	315	523
Tabou6	E_{moy}	0.12	0.09	0.07	0.05
	σ	0.05	0.03	0.03	0.02
	T_{moy}	110	317	639	1074
$r_i \in [0, \text{nbjobs} * 10]$					
Tabou1	E_{moy}	0.40	0.23	0.17	0.12
	σ	0.15	0.11	0.08	0.05
	T_{moy}	1	1	2	3
Tabou2	E_{moy}	0.35	0.17	0.15	0.09
	σ	0.14	0.11	0.18	0.04
	T_{moy}	55	159	318	543

Tabou3	E_{moy}	0.26	0.21	0.19	0.10
	σ	0.12	0.10	0.07	0.04
	T_{moy}	109	319	637	1077
Tabou4	E_{moy}	0.51	0.32	0.21	0.17
	σ	0.22	0.15	0.19	0.08
	T_{moy}	0.37	0.55	0.73	0.89
Tabou5	E_{moy}	0.39	0.19	0.16	0.10
	σ	0.18	0.10	0.08	0.15
	T_{moy}	54	150	304	520
Tabou6	E_{moy}	0.32	0.21	0.19	0.13
	σ	0.12	0.09	0.07	0.10
	T_{moy}	111	316	648	1079

Tableau 5: Résultats des Simulations, F2

$r_i \in [0.100]$					
F3		150	250	350	450
Tabou1	E_{moy}	0.10	0.07	0.05	0.03
	σ	0.04	0.09	0.02	0.01
	T_{moy}	1	1	0.77	3
Tabou2	E_{moy}	0.08	0.05	0.03	0.02
	σ	0.02	0.10	0.01	0.01
	T_{moy}	55	158	318	539
Tabou3	E_{moy}	0.09	0.05	0.04	0.01
	σ	0.11	0.03	0.01	0.007
	T_{moy}	120	314	629	1068
Tabou4	E_{moy}	0.12	0.08	0.07	0.05
	σ	0.05	0.03	0.03	0.01
	T_{moy}	0.38	0.56	0.72	0.93
Tabou5	E_{moy}	0.09	0.06	0.04	0.03
	σ	0.07	0.02	0.06	0.008
	T_{moy}	55	156	316	532
Tabou6	E_{moy}	0.10	0.07	0.05	0.02
	σ	0.09	0.03	0.02	0.03
	T_{moy}	113	322	636	1101
$r_i \in [0. nbjobs * 10]$					
Tabou1	E_{moy}	0.42	0.18	0.13	0.13
	σ	0.23	0.08	0.06	0.10
	T_{moy}	1	1	2	3
Tabou2	E_{moy}	0.25	0.11	0.09	0.09
	σ	0.14	0.09	0.03	0.02
	T_{moy}	55	157	315	533
Tabou3	E_{moy}	0.37	0.17	0.16	0.16
	σ	0.29	0.10	0.11	0.08
	T_{moy}	110	304	617	1043
Tabou4	E_{moy}	0.49	0.27	0.19	0.19
	σ	0.21	0.13	0.10	0.21
	T_{moy}	0.38	0.55	0.75	0.95
Tabou5	E_{moy}	0.35	0.14	0.11	0.11
	σ	0.16	0.09	0.14	0.04
	T_{moy}	55	153	318	517
Tabou6	E_{moy}	0.22	0.17	0.15	0.15
	σ	0.09	0.08	0.23	0.11
	T_{moy}	110	320	645	1058

Tableau 6: Résultats des simulations, F3

$r_i \in [0.100]$					
F4		150	250	350	450
Tabou1	E_{moy}	0.37	0.22	0.16	0.11
	σ	0.14	0.24	0.07	0.09
	T_{moy}	1	1	2	3
Tabou2	E_{moy}	0.14	0.11	0.07	0.03
	σ	0.06	0.05	0.09	0.01
	T_{moy}	55	158	306	506
Tabou3	E_{moy}	0.32	0.18	0.12	0.08
	σ	0.26	0.09	0.05	0.02
	T_{moy}	107	307	616	1043
Tabou4	E_{moy}	0.47	0.24	0.19	0.14
	σ	0.21	0.13	0.15	0.07
	T_{moy}	0.34	0.55	0.72	0.88

Tabou5	E_{moy}	0.29	0.13	0.10	0.06
	σ	0.17	0.05	0.06	0.02
	T_{moy}	52	149	315	527
Tabou6	E_{moy}	0.42	0.08	0.05	0.03
	σ	0.23	0.04	0.01	0.06
	T_{moy}	106	309	626	1054
$r_i \in [0. nbjobs * 10]$					
Tabou1	E_{moy}	0.96	0.61	0.36	0.21
	σ	0.45	0.29	0.18	0.10
	T_{moy}	1	1	2	3
Tabou2	E_{moy}	0.54	0.38	0.24	0.18
	σ	0.23	0.16	0.11	0.07
	T_{moy}	53	155	299	523
Tabou3	E_{moy}	0.59	0.49	0.31	0.25
	σ	0.24	0.15	0.27	0.32
	T_{moy}	108	327	650	1112
Tabou4	E_{moy}	1.09	0.83	0.66	0.53
	σ	0.72	0.39	0.30	0.24
	T_{moy}	0.44	0.53	0.73	0.88
Tabou5	E_{moy}	0.57	0.42	0.29	0.24
	σ	0.62	0.19	0.14	0.30
	T_{moy}	56	159	321	527
Tabou6	E_{moy}	0.64	0.41	0.36	0.29
	σ	0.29	0.15	0.27	0.12
	T_{moy}	105	300	366	1066

Tableau 7: Résultats des simulations, F4

$r_i \in [0.100]$					
F5		150	250	350	450
Tabou1	E_{moy}	0.97	0.97	0.46	0.39
	σ	0.36	1.12	0.23	0.17
	T_{moy}	1	1	2	3
Tabou2	E_{moy}	0.55	0.55	0.31	0.27
	σ	0.57	0.26	0.10	0.12
	T_{moy}	56	158	317	535
Tabou3	E_{moy}	0.63	0.63	0.39	0.25
	σ	0.31	0.28	0.14	0.09
	T_{moy}	108	306	632	1079
Tabou4	E_{moy}	1.02	1.02	0.64	0.51
	σ	0.79	0.62	0.32	0.24
	T_{moy}	0.40	0.57	0.74	0.91
Tabou5	E_{moy}	0.59	0.59	0.35	0.29
	σ	0.25	0.46	0.61	0.14
	T_{moy}	54	156	313	522
Tabou6	E_{moy}	0.65	0.65	0.41	0.33
	σ	0.37	0.23	0.19	0.54
	T_{moy}	112	311	620	1060
$r_i \in [0. nbjobs * 10]$					
Tabou1	E_{moy}	2.26	0.70	0.62	0.57
	σ	1.85	0.29	0.31	0.24
	T_{moy}	1	1	1	3
Tabou2	E_{moy}	0.97	0.56	0.43	0.32
	σ	0.49	0.12	0.27	0.16
	T_{moy}	54	154	300	515
Tabou3	E_{moy}	1.64	0.62	0.40	0.32
	σ	0.76	0.29	0.17	0.44
	T_{moy}	113	322	623	1078
Tabou4	E_{moy}	2.85	0.98	0.73	0.68
	σ	2.03	0.38	0.19	0.31
	T_{moy}	0.39	0.50	0.73	0.91
Tabou5	E_{moy}	1.34	0.61	0.55	0.43
	σ	0.66	0.54	0.29	0.26
	T_{moy}	55	156	313	527
Tabou6	E_{moy}	2.12	0.70	0.49	0.32
	σ	1.54	0.31	0.22	0.07
	T_{moy}	107	301	608	1056

Tableau 8: Résultats des simulations, F5

		$r_i \in [0, 100]$			
F6		150	250	350	450
Tabou1	E_{moy}	0.86	0.68	0.55	0.43
	σ	0.42	0.29	0.94	0.17
	T_{moy}	1	1	2	3
Tabou2	E_{moy}	0.44	0.38	0.24	0.27
	σ	0.18	0.16	0.09	0.31
	T_{moy}	53	151	297	501
Tabou3	E_{moy}	0.59	0.37	0.30	0.24
	σ	0.26	0.19	0.12	0.09
	T_{moy}	107	307	633	1071
Tabou4	E_{moy}	0.90	0.78	0.66	0.59
	σ	0.44	0.29	0.34	0.27
	T_{moy}	0.37	0.57	0.78	0.95
Tabou5	E_{moy}	0.55	0.40	0.27	0.28
	σ	0.26	0.18	0.14	0.11
	T_{moy}	53	150	305	520
Tabou6	E_{moy}	0.60	0.41	0.29	0.29
	σ	0.29	0.17	0.32	0.13
	T_{moy}	113	319	652	1101
		$r_i \in [0, nbjobs * 10]$			
Tabou1	E_{moy}	0.85	0.69	0.37	0.29
	σ	0.91	0.31	0.28	0.11
	T_{moy}	1	1	2	3
Tabou2	E_{moy}	0.61	0.25	0.19	0.11
	σ	0.30	0.12	0.24	0.04
	T_{moy}	53	149	300	506
Tabou3	E_{moy}	0.74	0.37	0.24	0.17
	σ	0.35	0.16	0.11	0.09
	T_{moy}	110	304	619	1061
Tabou4	E_{moy}	0.96	0.43	0.40	0.33
	σ	0.47	0.21	0.18	0.12
	T_{moy}	0.37	0.56	0.72	0.92
Tabou5	E_{moy}	0.71	0.26	0.21	0.17
	σ	0.38	0.32	0.10	0.09
	T_{moy}	55	156	311	536
Tabou6	E_{moy}	0.83	0.55	0.32	0.25
	σ	0.40	0.25	0.15	0.12
	T_{moy}	110	314	643	1038

Tableau 9: Résultats des simulations, F6

Remarquons, tout d'abord, que les familles F1, F2 et F3 sont faciles à résoudre par l'une ou l'autre des variantes. Une solution optimale ou proche de celle-ci est, en effet, facile à trouver.

Par ailleurs, la résolution des familles F4, F5 et F6 est plus difficile en donnant, rarement des solutions optimales. Cependant, l'écart moyen reste relativement faible.

Nous remarquons également que l'écart moyen augmente légèrement lorsque nous augmentons l'intervalle de r_i à $[0, nbjobs * 10]$. En effet, des dates de disponibilité dispersées augmentent la probabilité d'avoir des temps d'attente imposés sur M_c .

Pour un même jeu de données, nous constatons que Tabou1 et Tabou4 sont les moins performantes en donnant le plus grand écart. Toutes les deux utilisent l'opérateur de changement (APEX). Cet opérateur offre une exploration limitée de l'espace de recherche par rapport aux autres. Par ailleurs, les procédures Tabou2 et Tabou5, utilisant l'opérateur de changement (EX), se montrent très performantes. Elles ont donné de très bons résultats pour tous types de données. Ainsi, elles n'ont pas dépassé un écart moyen de 1.5%. Les procédures

Tabou3 et Tabou6 ont, également, donné de bons résultats avec une moyenne des écarts ne dépasse pas 2.5%.

Les procédures Tabou2 et Tabou5 donnent le meilleur compromis taille du voisinage/qualité de la solution. Cependant, une légère avancée est enregistrée pour la gestion de la liste tabou utilisant les mouvements (Tabou2) plutôt que celle qui enregistre la valeur du makespan (tabou5). Ceci nous permet de confirmer la performance de tabou2 sur toutes les autres variantes.

Nous remarquons aussi que plus le nombre de jobs augmente plus la qualité de la solution devient meilleure. Le nombre important de jobs offre une grande probabilité de faire travailler les machines dédiées en parallèle et donc de mieux les occuper.

En ce qui concerne le temps d'exécution des procédures tabou, il augmente en fonction du nombre de jobs. Cependant, il reste quasiment constant en fonction de la méthode de gestion de la liste tabou pour un même opérateur de changement. En outre, le temps d'exécution relevé par les variantes tabou dont l'opérateur de changement est DSH, est plus grand. Ceci s'explique par la taille du voisinage examiné par ce dernier qui est deux fois plus grande que celle de EX qui est, elle même, plus grande que la taille donnée par APEX.

les résultats des simulations montrent, en plus, que les valeurs des écart-types (σ) sont, en moyenne, faibles. Ce qui nous permet de conclure que les valeurs sont regroupées autour de l'écart moyen visé dans les 20 instances testées.

8- CONCLUSION ET PERSPECTIVES

L'utilisation de la recherche tabou pour la minimisation du makespan dans un flow shop hybride avec machines dédiées a permis d'obtenir des solutions de très bonne qualité.

L'étude du comportement des solutions visitées a mis en évidence un comportement cyclique que nous avons pu éviter en déterminant, pour chaque famille de jobs et chaque variante tabou, la longueur des cycles observés au bout desquels la diversification est appliquée. L'étude expérimentale a montré que, contrairement à ce qui est attendu, plus le nombre de jobs augmente plus la qualité de la solution s'améliore, et ce, quelque soit la famille de jobs et la nature des r_i (éparpillés ou plutôt vers le début de la séquence).

Nous remarquons que les plus grands écarts moyens sont obtenus pour les r_i situés dans $[0, nbjobs * 10]$. Ceci devrait nous conduire à faire plus d'effort pour proposer de nouvelles bornes inférieures plus réalistes.

Le travail réalisé peut facilement s'étendre au cas de m machines dédiées, $m \geq 3$.

REFERENCES

- [1] C.R.REEVES, 1999. Landscape, Operators and Heuristic Search. *Annals of Operations Research* 86, 473-490
- [2] Dridi, N., H. Hadda and S. Hajri-Gabouj, 2001. Méthode heuristique pour le problème de flow shop hybride avec machines dédiées. *RAIRO Operations Research*.
- [3] E. Taillard, 1990. Some efficient Methods for the Flow shop sequencing problem. *European Journal of Operational Research*, Vol 47.
- [4] F. Glover, 1989. Tabu search - Part I, *ORSA J. Comput.* 1 190-206.
- [5] Haouari M., M. R., 1997. Heuristic algorithms for the two-stage hybrid flow shop problem. *Operations Research Letters* 21, 1, 43-53
- [6] Hatem Hadda, Najoua Dridi, and Sonia Hajri-Gabouj, 2014. Exact resolution of the two-stage hybrid flow shop with dedicated machines. *Optimization Letters* DOI 10.1007/s11590-014-0741-y
- [7] Jaehwan Yang, 2013. A two-stage hybrid flow shop with dedicated machines at the first stage. *Computers & Operations Research* 40, 2836–2843.
- [8] Jen-Shiang Chen, Jason Chao-Hsien Pan, Chien-Kuang Wu, 2008. Hybrid tabu search for re-entrant permutation flow-shop scheduling problem *Expert Systems with Applications*, Volume 34, Issue 3, Pages 1924-1930.
- [9] Johnson, S. M, (Mar. 1954). Optimal two- and three-stage production schedules with setup times included. *Naval Research Logistics Quarterly* 1, 1, 61–68.
- [10] Lawler, E. L., Lenstra, J. K., Rinnooy Kan, A. H. G. and Shmoys, D. B., , 1989. Sequencing and scheduling: Algorithms and complexity. In *Handbooks in Operations Research and Management Science*, Vol 4: Logistic of Production and Inventory, eds S. C. Graves, A. H. G. Rinnooy Kan, and E Zipkin. North Holland, pp. 445-522.
- [11] Lin, B. M. T, 1999. The strong NP-hardness of two-stage flow shop scheduling with a common second-stage machine. *Computers & Operations Research* 26, 695–698.
- [12] Mosheiov G, Sarig A, 2010. Minimum weighted number of tardy jobs on an m-machine flow shop with a critical machine. *European Journal of Operational Research*, 201(2):404–8.
- [13] Ogzu, C., Lin, B. M. T., and Cheng, T. C. E, 1997. Two-stage flow shop scheduling problem with a common second-stage machine. *Computers & Operations Research* 24, 1169–1174.
- [14] Riane, F., Artiba, A., Elmaghraby, S. E, 1998. A hybrid three-stage flow shop problem: Efficient heuristics to minimize makespan, *European Journal of Operational Research* 109 (2): 321–329.
- [15] Riane F, Artiba A, Elmaghraby SE, 2002. Sequencing a hybrid two-stage flow shop with dedicated machines. *International Journal of Production Research*; 40:4353–80.
- [16] Shijin, W., Ming, L, 2013. A heuristic method for two-stage hybrid flow shop with dedicated machines. *Computer & Operations Research* 40 438-450.
- [17] T. Stutzle, 1998. Applying Iterated Local Search to the Permutation Flow shop Problem. Technical Report AIDA-98-04, FG Intellektik, TU Darmstadt, Intellectics Group,
- [18] W. Besbes, T. Loukil. ET J. Teghem, 2010. A two-stage flow shop with parallel dedicated machines: in 8th International Conference of Modeling and Simulation «Evaluation and optimization of innovative production systems of goods and services”.