

# Ordonnancement d'ateliers de type flow shop avec contrainte d'opérateurs

IMENE BENKALAI<sup>1</sup>, PIERRE BAPTISTE<sup>2</sup>, DJAMAL REBAINE<sup>3</sup>

<sup>1</sup>Département d'informatique et de mathématique  
Université du Québec à Chicoutimi  
555, boulevard de l'Université, Chicoutimi, QC, G7H 2B1, Canada  
[imene.benkalai@uqac.ca](mailto:imene.benkalai@uqac.ca)

<sup>2</sup>Département MAGI  
École Polytechnique de Montréal  
2900, Boulevard Edouard-Monpetit, Montréal, QC, H3T 1J4, Canada  
[pbaptiste@polymtl.ca](mailto:pbaptiste@polymtl.ca)

<sup>3</sup>Département d'informatique et de mathématique  
Université du Québec à Chicoutimi  
555, boulevard de l'Université, Chicoutimi, QC, G7H 2B1, Canada  
[drebaine@uqac.ca](mailto:drebaine@uqac.ca)

---

**Résumé** - Cet article traite du problème d'affectation d'opérateurs dans un atelier de type flow shop où le nombre d'opérateurs est inférieur au nombre de machines. Nous comparons une approche séquentielle à une approche simultanée. Dans la première, nous résolvons d'abord le problème du flow shop sans contrainte d'opérateurs puis utilisons plusieurs heuristiques pour l'affectation des opérateurs. Dans la deuxième, nous proposons des heuristiques qui permettent le traitement simultané de l'affectation des opérateurs avec l'ordonnancement de tâches. Ces approches sont testées sur une version du benchmark de Taillard pour le flow shop de base qui a été adaptée à notre problématique. Les résultats montrent que l'approche la plus adaptée pour la résolution de notre problème varie selon les paramètres de ce dernier, notamment le nombre d'opérateurs.

**Abstract**- This paper addresses the problem of assigning a number of operators less than the number of machines, in a flow shop environment. We compare a sequential approach with a simultaneous approach. In the former, we first solve the scheduling problem, and then we assign the operators by using several heuristic algorithms. In the latter, we design heuristic solutions that take into account simultaneously the assignment of operators and the scheduling of tasks. The two approaches are tested on a version of Taillard's benchmark for basic flow shop that was adapted to our problem. The results show that the appropriate approach depends on the parameters of the problem, including the number of operators.

**Mots clés**- Ordonnancement, flow shop, opérateurs, makespan.

**Keywords**- Scheduling, flow shop, operators, makespan.

---

## 1 INTRODUCTION

Lors de la gestion d'un système de production, les ressources additionnelles nécessaires varient en allant d'un opérateur humain à un robot ou AGV (Automated Guided Vehicle), en passant par un outil ou encore un programme informatique [Hall et al., 2000; Blazewicz et al., 2007]. Seulement, l'homme a depuis toujours un rôle décisif en économie, étant, d'un côté, à la fois consommateur et producteur, et, de l'autre, le seul à pouvoir faire face à l'imprévu, et ce, malgré les progrès qu'ont connus les équipements de production [Baptiste et al., 2005].

De nombreux travaux soulignent la nécessité de l'intégration de la composante humaine dans les décisions d'ordonnancement de production et ce, au regard des liens étroits existants entre la gestion des ressources humaines et l'ordonnancement de production, notamment au niveau opérationnel [Zouba, 2009]. Seulement, la prise en compte de la composante humaine et de ses nombreux paramètres tels que l'expérience d'un opérateur, ses qualifications, mais aussi les charges de travail variables qui lui sont affectées, etc., rend les problèmes d'ordonnancement dans ce milieu de plus en plus complexes. Ceci a rendu le développement de méthodes de résolution plus ardu. Par ailleurs, les modèles d'ordonnancement classiques ne sont que peu

réalistes. Ceci nous conduit logiquement à l'intégration des ressources humaines et leurs caractéristiques et ce, dans le but de réduire l'écart en la théorie et la pratique. En plus de l'ordonnement classique des tâches, il s'agira dans ce cadre d'attribuer non seulement des horaires aux opérateurs, mais aussi de les affecter à des postes de travail.

Une approche intégrée prenant en compte ces deux aspects trouverait son intérêt non seulement dans les cas où les deux types de ressources sont imbriqués et interdépendants, mais aussi dans les cas où leur importance est équivalente.

En ordonnancement "classique", il est supposé implicitement que le nombre de machines est égal à celui des opérateurs ; ceci n'est que très rarement vérifié dans les systèmes réels. Dans le présent papier, notre étude porte sur l'ordonnement d'ateliers de type flow shop en présence d'opérateurs avec un nombre d'opérateurs inférieur au nombre de machines et pour objectif, la minimisation du makespan, dénotant le temps total d'accomplissement d'un ordonnancement donné. Rappelons que dans le modèle de flow shop, l'ordre de passage des opérations sur les machines est identique. Si, en plus, la séquence de passage de ces opérations est la même sur toutes les machines, on parle alors du modèle de permutation. L'étude de ce modèle est justifiée par le fait que, en pratique, souvent dans des unités de production, des opérations doivent être exécutées dans le même ordre sur l'ensemble des machines.

Dans notre modèle, nous supposons que la durée des tâches n'est pas directement affectée par l'intervention des opérateurs car le traitement d'une opération nécessitera la présence d'un opérateur pour toute sa durée. Ainsi, l'impact du partage d'opérateurs sera modélisé par des temps d'attente engendrés par le fait que le nombre d'opérateurs est inférieur au nombre de machines. Il s'agit donc d'étudier l'impact de la considération simultanée des ressources humaines et matérielles sur la qualité des solutions produites. Dans un premier temps, nous comparons plusieurs approches pour l'affectation d'opérateurs pour un ordonnancement produit par diverses heuristiques pour le problème de flow shop de base. Leurs résultats seront comparés à une borne que nous avons développée pour le problème étudié. Ceci est dans le but d'évaluer les importances respectives des deux aspects du problème, à savoir la gestion des ressources humaines et la gestion des ressources matérielles.

Ensuite, dans un second temps, nous comparons l'approche de traitement simultané d'affectation d'opérateurs et d'ordonnement de tâches avec celle qui consiste à affecter les opérateurs à un ordonnancement donné. Ceci nous indiquera s'il est plus intéressant de prendre le problème dans son ensemble ou alors de procéder par phases.

Le reste de ce papier est organisé comme suit. Une revue de littérature dédiée au problème de flow shop en présence d'opérateurs fera l'objet de la Section 2. La Section 3 présente, quant à elle, des résultats concernant le statut de la complexité du problème étudié. À la Section 4, nous détaillons les méthodes de résolution et les bornes que nous avons développées. Une étude expérimentale est présentée à la Section 5 et l'analyse de ces résultats à la Section 6. Enfin, nous présenterons à la Section 7 une conclusion ainsi que quelques perspectives pour de futurs travaux.

## 2 REVUE DE LA LITTÉRATURE

Auparavant appelé "systèmes de production à  $k$  niveaux", l'atelier de type "flow shop" ne fut appelé ainsi que depuis les travaux de Heller en 1960 [Gupta et Stafford Jr., 2006]. Dans le problème du flow shop de base, tel qu'étudié dans [Johnson, 1954], il s'agit d'ordonner  $n$  tâches sur  $m$  machines de façon à minimiser un certain objectif. Les tâches doivent passer par toutes les machines exactement une fois, et ce, dans le même ordre (flow shop de permutation). Par la suite, de nombreuses hypothèses furent ajoutées et/ou relaxées par rapport au modèle de base, ce qui engendra une multitude de modèles, notamment à partir des années 1980 [Gupta et Stafford Jr., 2006].

Le problème du flow shop standard, noté  $F_m||C_{max}$ , a été prouvé NP-difficile au sens fort pour  $m \geq 3$  [Garey et al., 1976].

Comparativement au flow shop de permutation pour lequel on comptait plus de 1200 publications en 2006 [Gupta et Stafford Jr., 2006], le flow shop standard a quant à lui été peu étudié. Ceci est principalement dû à la relative simplicité du flow shop de permutation par rapport à la version standard de ce dernier. Il a été montré dans [Potts et al., 1991] que pour certaines familles d'instances, le rapport entre la meilleure solution avec contrainte de permutation pour la minimisation du makespan et la solution optimale du flow shop est supérieur à  $\frac{1}{2}\sqrt{m}$ . Rappelons toutefois que pour  $m \leq 3$ , ces deux modèles sont identiques. Pour plus de détails et un plus large panorama des méthodes de résolution appliquées au problème du flow shop standard, voir [Färber et al., 2007 ; Koulamas, 1998 ; Liao et al., 2007 ; Noorulhaq et al., 2007 ; Ruiz et al., 2004 ; Ying et Lin, 2007 ; Ying et al., 2010 ; Vahedi-Nouri et al., 2013 ; Ziaee et Sadjadi, 2007].

Dans ce qui suit, nous présentons une brève revue de la littérature consacrée au problème du flow shop avec opérateurs.

Nous avons noté que relativement peu de travaux ont porté sur les problèmes de flow shop avec opérateurs. Ceci est principalement dû à l'apparition assez récente des problèmes avec opérateurs ainsi qu'à la grande difficulté de ces derniers, notamment en ce qui concerne la modélisation des caractéristiques des ressources humaines, telles que l'expérience d'un opérateur, ses qualifications, les charges de travail variables qui lui sont affectées, etc.

Les modèles d'ordonnement "classiques" supposent implicitement que le nombre  $k$  d'opérateurs est égal au nombre  $m$  de machines. Ainsi, un opérateur pourra s'occuper exclusivement de la machine qui lui est affectée. Seulement, cette hypothèse n'est que très rarement vérifiée en pratique. En effet, le nombre d'opérateurs est généralement bien réduit en comparaison avec le nombre de machines.

Un large pan de la littérature a été dédié à l'étude des cellules de production robotisées, particulièrement en environnement de flow shop cyclique. Il y est reconnu que l'ajout de robots permet d'améliorer considérablement l'efficacité de la production [Aneja et Kamoun, 1999]. Une revue de la littérature concernant l'ordonnement de cellules de production robotisées, mais aussi des défis qu'il présente actuellement, peut être trouvé

dans [Geismar et al., 2008, Hall et al., 2000, Sethi et al., 1992]. Un état de l'art est présenté dans [Dawande et al., 2005].

Dans chacune des études dédiées aux problèmes d'ordonnancement avec opérateurs, les auteurs définissent différents modèles basés sur diverses hypothèses. Depuis les travaux de Vickson en 1980 [Vickson, 1980], la majorité des études entreprises dans le domaine de l'ordonnancement avec opérateurs a porté sur des modèles avec des temps opératoires contrôlables par la quantité de ressources. Il s'agissait, pour résoudre ces problèmes, d'affecter un surnombre d'opérateurs aux tâches pour en réduire la durée opératoire. À la différence des précédents travaux, le modèle proposé par Cheurfa [Cheurfa, 2005] étudie un atelier de type flow shop, cyclique, avec un nombre d'opérateurs inférieur au nombre de machines. L'auteur y introduit la notion de "partage" d'opérateurs et y étudie un problème de décision plutôt qu'un problème d'optimisation, avec des opérateurs de compétences équivalentes. L'intervention de ces derniers se limite à des opérations de montage, contrôle et démontage des opérations.

Une analyse de la littérature nous a permis de constater qu'une partie des travaux concernant l'ordonnancement en présence d'opérateurs est dédiée aux environnements à machines parallèles. Nous citons pour l'exemple le travail de Zouba [Zouba, 2009]. L'objet en est l'ordonnancement de tâches avec une affectation simultanée d'opérateurs dans un environnement à machines parallèles avec un nombre d'opérateurs inférieur au nombre de machines et la possibilité pour un opérateur de s'occuper de plusieurs machines en même temps. Dans ce modèle, il y est notamment question de l'impact du partage d'opérateurs sur les durées opératoires des différentes tâches. Plusieurs variantes des modes d'intervention des opérateurs y sont introduites. Dans [Baptiste et Munier, 2013], ce modèle a été un plus raffiné en supposant que le ratio entre temps de traitement réels et théoriques est une fonction linéaire des taux d'occupation des tâches par l'opérateur. Même s'il ne permet pas de prévoir les durées exactes des tâches, ce modèle permet de prévoir l'augmentation globale de la durée d'un ordonnancement.

Étant donnée la popularité du concept *OWMM (One-Worker-Multiple-Machine)* vers la fin des années 1990 et au début des années 2000, notamment dans les systèmes de production "*juste-à-temps*", plusieurs travaux ont porté sur ce sujet. Nous citons pour l'exemple l'article de Cheng et al. [Cheng et al., 1999] sur un flow shop à deux machines pour la minimisation du makespan où l'intervention de l'opérateur se limite à des opérations de réglages en début et fin de tâche. Citons aussi la thèse de doctorat de Baki [Baki, 1999]. Il y est question des modèles de flow shop et d'open shop, avec un seul opérateur. Ce dernier aura à s'occuper uniquement d'opérations de réglage au début d'une séquence de tâches, d'où l'utilisation de la notion de "batches" qui consiste à grouper ces tâches, et ensuite à les ordonnancer. La durée de ces réglages peut être soit dépendante, soit indépendante, de la séquence de tâches. Plusieurs objectifs ont été considérés. Ils sont basés sur les temps de fin de traitement et/ou des dates d'échéance des tâches. L'auteur a prouvé que la majorité de ces problèmes était *NP-difficile*. Cependant, il parvint à établir des relations de dominance pour

un certain nombre de ces problèmes. Nous citons pour l'exemple le fait que les solutions semi-actives dominent toutes les autres solutions. Il restreint ainsi sa recherche de l'optimum à ce sous-ensemble de solutions. Pour plus de détails concernant l'ordonnancement avec des ressources humaines [Baki, 1999 ; Baptiste et al., 2005].

Comme nous l'avons déjà précisé, les problèmes d'ordonnancement étant pour la plupart *NP-difficiles* ne sont que plus complexes avec l'ajout de contraintes de ressources. Ainsi, nous verrons plusieurs études porter sur des cas de taille réduite avec  $m=2$ . Pour plus de détails, voir [Baki, 1999 ; Baptiste et Munier, 2013; Oulamara et al., 2013].

### 3 COMPLEXITE DU PROBLEME

Dans un premier temps, comme pour tout problème d'optimisation combinatoire, nous avons été amenés à étudier la complexité du problème considéré. Nous avons d'abord étudié le problème d'affectation d'opérateurs étant donné un ordonnancement  $S$ , noté  $F_m | \text{res } 1k1, S | C_{\max}$ , conformément à la notation proposée dans [Blazewicz et al., 2007] à laquelle nous ajoutons le symbole  $S$  pour indiquer que l'affectation se fait étant donné un ordonnancement  $S$ . Ensuite, nous avons étudié le problème de traitement simultanée d'affectation d'opérateurs et d'ordonnancement de tâches, noté  $F_m | \text{res } 1k1 | C_{\max}$ . Les résultats sont comme suit.

#### *Théorème 3.1*

Le problème  $F_m | \text{res } 1k1, S | C_{\max}$  est *NP-difficile* au sens fort pour  $k \geq 2$ .

#### *Preuve :*

La preuve de la *NP-complétude* se fera via la méthode de restriction. Notons d'abord que les contraintes de passage des opérations sur les machines et celles générées par  $S$ , une solution donnée d'ordonnancement, forment un graphe de précédence. Ainsi notre problème de flow shop peut être vu comme un problème de machines parallèles identiques avec un graphe de précédence, noté par  $P_k | \text{prec} | C_{\max}$ , où les machines, les tâches et le graphe de précédence représentent respectivement les  $k$  opérateurs, les  $m \times n$  opérations, et les relations de précédence entre ces opérations. Ce problème est connu pour être *NP-difficile* au sens fort pour  $k \geq 2$  [Pinedo, 2002]. Le résultat du théorème est donc établi.  $\square$

#### *Théorème 3.2*

Le problème  $F_m | \text{res } 1k1 | C_{\max}$  est *NP-difficile* au sens fort pour  $k \geq 2$ .

#### *Preuve :*

Là encore, nous utilisons la méthode de restriction. Ici, le graphe de précédence est seulement formé par les contraintes de passage des opérations sur les machines générant ainsi un ensemble de  $n$  chaînes indépendantes les unes des autres. Notre problème de flow shop peut donc être vu comme un problème de machines parallèles identiques avec un graphe de précédence formé par un ensemble de  $n$  chaînes, noté par  $P_k | \text{chains} | C_{\max}$ , où les machines, les tâches et le graphe de précédence représentent respectivement les  $k$  opérateurs, les  $m \times n$  opérations, et les relations de précédence entre ces opérations. Notons qu'ici il y a possibilité que des opérations appartenant à des tâches différentes, mais

exécutées par une même machine dans le problème de flow shop soient exécutées en même temps dans  $P_k | \text{chains} | C_{\max}$ , alors que cela n'est pas possible dans  $F_m | \text{res } 1k1, S | C_{\max}$ . Si tel est le cas, il suffit de les décaler en passant de la solution de  $P_k | \text{chains} | C_{\max}$  à la solution correspondante sur  $F_m | \text{res } 1k1, S | C_{\max}$ . Une fois ce point pris en compte, nous pouvons conclure que le théorème est établi car  $P_k | \text{chains} | C_{\max}$  est connu pour être NP-difficile au sens fort pour  $k \geq 2$  [Du et al., 1991]. □

Étant donnée la NP-difficulté au sens fort de ces deux problèmes, l'utilisation d'une approche heuristique pour leur résolution se trouve ainsi justifiée.

#### 4 HEURISTIQUES ET BORNE INFÉRIEURE

Dans cette section, nous présentons les différentes heuristiques utilisées pour la résolution de notre problème. Pour l'évaluation de ces dernières, nous avons développé une borne inférieure que nous présentons juste après. Pour plus de clarté, nous terminons par un exemple numérique d'application.

##### 4.1 Heuristiques

Quatre heuristiques furent utilisées pour la résolution du problème d'affectation d'opérateurs étant donné un ordonnancement  $S$ . Trois d'entre elles furent utilisées pour la résolution du problème de traitement simultané de l'affectation d'opérateurs ainsi que de l'ordonnancement de tâches. Toutes ces heuristiques ont été adaptées pour la prise en considération des contraintes de précédence. Elles sont comme suit :

###### 4.1.1 Heuristique $H_1$

Il s'agit de l'heuristique *LPT* (Longest Processing Time) qui ordonne les tâches selon l'ordre décroissant de leurs temps de traitement, puis les affecte l'une après l'autre au premier opérateur disponible.

###### 4.1.2 Heuristique $H_2$

Il s'agit de l'heuristique *SPT* (Shortest Processing Time) qui ordonne les tâches selon l'ordre croissant de leurs temps de traitement, puis les affecte l'une après l'autre au premier opérateur disponible.

###### 4.1.3 Heuristique $H_3$

Il s'agit d'une heuristique qui fait qu'un opérateur disponible accordera la priorité aux tâches les plus en amont (i.e. qui sont traitées par les machines de plus petit indice).

###### 4.1.4 Heuristique $H_4$

Dans le cas d'un ordonnancement donné, il s'agit d'une heuristique qui fait qu'un opérateur disponible accordera la priorité aux opérations appartenant aux tâches ayant les plus petits indices dans ledit ordonnancement.

Dans le premier cas, qui consiste à étudier l'affectation d'opérateurs étant donné un ordonnancement  $S$ , ces différentes heuristiques sont adaptées pour la prise en considération des contraintes de précédence imposées par ce dernier. Dans le cas du traitement simultané des deux aspects du problème, les seules contraintes de précédence à respecter sont celles imposées par l'ordre que les opérations auront à suivre sur les machines du flow shop.

##### 4.2 Borne inférieure $B_1(k)$

Pour nous permettre d'évaluer nos méthodes, nous avons développé une borne inférieure que l'on note  $B_1(k)$  pour  $k$  opérateurs. Développée au départ pour le cas à trois machines et deux opérateurs, elle se base sur le fait qu'au début et à la fin de l'ordonnancement, un seul opérateur est actif. Le reste du temps, nous aurions idéalement les deux opérateurs qui seraient actifs à plein temps. Ceci nous donne la borne suivante, avec  $p_{ij}$  étant le temps de traitement de la tâche  $j$  sur la machine  $i$  :

$$B_1(2) = \min_{j=1,n} \{p_{1j}\} + \min_{j=1,n} \{p_{mj}\} + \frac{1}{2} \left[ \sum_{j=1}^n \sum_{i=1}^m p_{ij} - \left( \min_{j=1,n} \{p_{1j}\} + \min_{j=1,n} \{p_{mj}\} \right) \right]$$

En nous basant sur la même remarque, nous avons par la suite généralisé cette borne pour  $k$  opérateurs, ce qui a donné ce qui suit :

$$B_1(k) = \sum_{i=1}^{k-1} \min_{j=1,n} \{p_{ij}\} + \sum_{i=m-k+2}^m \min_{j=1,n} \{p_{ij}\} + \frac{1}{k} \left[ \sum_{j=1}^n \sum_{i=1}^m p_{ij} - \left( \sum_{j=1}^{k-1} \min_{j=1,n} \{p_{ij}\} + \sum_{i=m-k+2}^m \min_{j=1,n} \{p_{ij}\} \right) \right]$$

Pour plus de clarté, considérons l'exemple suivant. Soit un flow shop avec 4 tâches ( $j, j=1, \dots, 4$ ) dont les temps de traitement  $p_{ij}$  sont résumés dans le Tableau 1, 4 machines ( $M_i, i=1, \dots, 4$ ) et 2 opérateurs.

Soit donc l'ordonnancement de départ  $S = \{1, 2, 3, 4\}$ , avec  $C_{\max}(S) = 18$  (Figure 1). L'affectation des opérateurs selon l'heuristique  $H_1$  génère un  $C_{\max}(S) = 22$ . Le traitement simultané de l'affectation et de l'ordonnancement avec l'heuristique *LPT* produit un  $C_{\max}(S) = 23$ . Lors de l'application de ces heuristiques, il s'agit de choisir les  $k$  tâches à traiter selon le critère de chacune des méthodes et de différer l'exécution des autres tâches. Les ordonnancements obtenus sont illustrés en Figure 2 et 4. Nous remarquons que la solution optimale  $S^*$  avec  $C_{\max}(S^*) = B_1(2) = 21$  est produite par l'heuristique  $H_2$  (Figure 3).

Tableau 1. Temps de traitement  $p_{ij}$

	1	2	3	4
$M_1$	1	2	3	2
$M_2$	2	3	2	3
$M_3$	2	1	4	4
$M_4$	2	4	2	2

#### 5 ÉTUDE EXPERIMENTALE

En l'absence de benchmarks de référence, nous avons utilisé pour l'évaluation de nos méthodes le benchmark de Taillard pour le problème de flow shop de permutation disponible sur le site suivant : <http://mistic.heigvd.ch/taillard/problemes.dir/ordonnancement.dir/ordonnancement.html>.

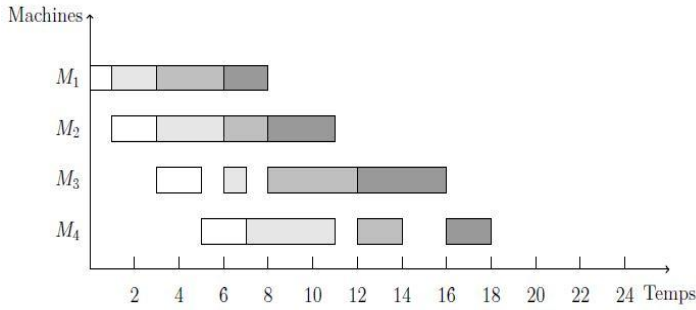


Figure 1. Solution S sans contrainte d'opérateurs

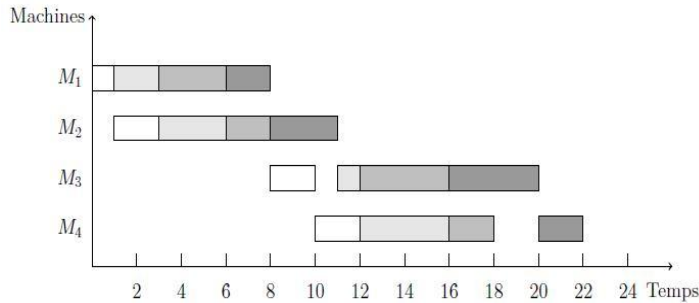


Figure 2. Solution de  $H_1$  étant donné S

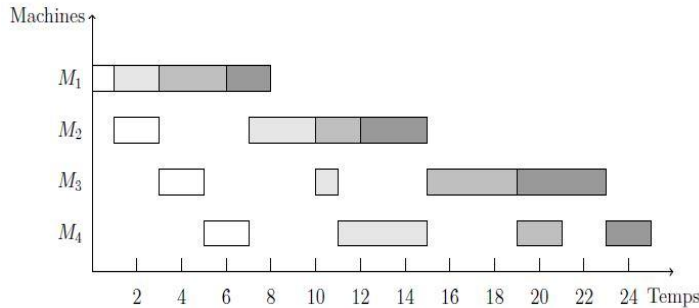


Figure 3. Solution de  $H_2$  étant donné S

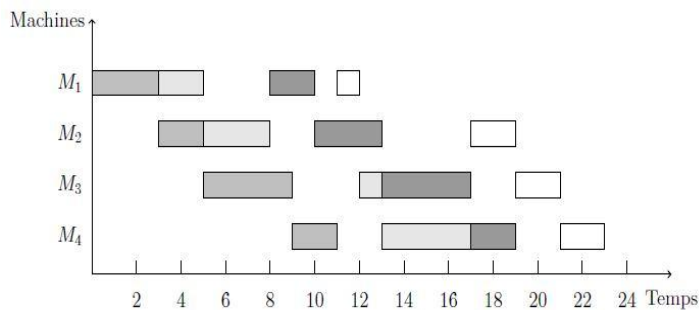


Figure 4. Affectation simultanée d'opérateurs et de tâches avec  $H_1$

Les instances utilisées ont des nombres de tâches de 20, 50 et 100, les nombres de machines étant de 5, 10 et 20. Dans un premier temps, nous avons utilisé trois approches pour la génération d'ordonnement de tâches, à savoir, NEH, HFC et la génération aléatoire (nous avons généré cinq solutions par instance). Rappelons que l'algorithme NEH [Nawaz et al., 1983] consiste à ordonner les tâches selon l'ordre décroissant de la

somme des temps de traitement de leurs opérations puis de trouver, une tâche à la fois, la meilleure position d'insertion. L'algorithme HFC [Koulamas, 1998] est basé sur une idée de Johnson pour le  $F_3||C_{max}$  (qui s'est avérée fautive par la suite [Burns et Rooker, 1976]) qui stipule que si la séquence optimale pour les deux premières machines est la même que celle qui est optimale pour les deux dernières machines, alors cette séquence est optimale pour le problème en entier [Johnson, 1954]. Cette conjecture est étendue heuristiquement aux paires de tâches. Enfin, dans la génération aléatoire, la solution est produite en générant une permutation aléatoirement, puis en échangeant, sur chaque machine, l'ordre de deux tâches choisies aléatoirement. Les résultats sont présentés dans le Tableau 1. Les moyennes des solutions obtenues par chacune des approches sont comparées à la borne inférieure et supérieure de chacune des instances. Les pourcentages de déviation par rapport à la borne supérieure (Sup) et à la borne inférieure (Inf) sont calculés comme suit :

$$d^+ = \frac{(C_{max}^H(S) - Sup) * 100}{Sup}, H \in \{NEH, HFC, Alea\},$$

$$d^- = \frac{(C_{max}^H(S) - Inf) * 100}{Sup}, H \in \{NEH, HFC, Alea\}.$$

où  $C_{max}^H(S)$  est le makespan de la solution  $S$  générée par l'heuristique  $H$ .

Ensuite, nous avons testé différentes possibilités pour l'affectation des opérateurs. D'abord, l'affectation d'opérateurs étant donné des ordonnancements fournis, respectivement, par NEH, HFC et la génération aléatoire et ce, en utilisant les heuristiques présentées à la Section 4. Puis, d'un autre côté, l'affectation simultanée d'opérateurs avec l'ordonnement de tâches suivant trois règles d'ordonnement basées sur les trois premières heuristiques.

Les résultats fournis par ces méthodes, présentés dans le Tableau 2, sont comparés à la borne inférieure  $B_1(k)$ , présentée à la Section 4 suivant le pourcentage de déviation  $d$  calculé selon la formule suivante :

$$d = \frac{(C_{max}^{H_i}(S) - B_1(k)) * 100}{B_1(k)}, i \in \{1, 2, 3, 4\},$$

où  $C_{max}^{H_i}(S)$  est le makespan de la solution  $S$  générée par l'heuristique  $H_i$ ,  $i=1, \dots, 4$ . Les instances sont notées  $n \times m$ , où  $n$  et  $m$  représentent respectivement le nombre de tâches et le nombre de machines du problème de flow shop. Dans les Figures 5-10, l'axe y représente les pourcentages de déviation, et l'axe des x représente le nombre d'opérateurs pour les différentes instances.

## 6 ANALYSE DES RESULTATS

Nous constatons d'abord que c'est l'heuristique  $H_1$ , priorisant les opérations de plus longues durées, qui génère globalement les meilleurs résultats pour les ordonnancements fournis par NEH, HFC et les solutions aléatoires (Figure 5, 6 et 7). En effet, nous avons remarqué que la performance de cette heuristique dépasse celle des autres, notamment pour un nombre élevé d'opérateurs. Ceci est dû au fait qu'après le traitement des plus longues tâches, les opérateurs auront la latitude d'assurer un roulement plus rapide des opérations, ce qui aurait tendance à minimiser le



makespan. Un autre point, que nous avons observé, est que les résultats produits s'approchent de la borne théorique pour des nombres réduits d'opérateurs. Ceci laisse penser que la résolution du problème est plus "facile" dans cette zone de travail.

Remarquons aussi que l'écart de performance entre les heuristiques appliquées aux solutions aléatoires et aux ordonnancements fournis par NEH et HFC se réduit avec la réduction du nombre d'opérateurs (Figure 8). Ceci implique, au vu des résultats précédents, que la partie du problème la plus importante est celle de l'affectation d'opérateurs. C'est sur cette partie-là que devraient se concentrer les efforts visant à la conception et/ou l'amélioration de méthodes.

En comparant les différentes heuristiques d'affectation simultanée (Figure 9), nous remarquons que c'est H<sub>3</sub> qui produit les meilleurs résultats avec l'augmentation du nombre d'opérateurs. En effet, nous avons remarqué que cette heuristique assure une utilisation plus uniforme des différentes machines sur la durée de l'ordonnancement, ce qui représente en pratique l'objectif de la minimisation du makespan, d'où sa performance.

Ainsi, bien qu'en termes de complexité et de structure notre problème se ramène à un problème à machines parallèles avec

contraintes de précedence, les heuristiques "classiques" pour la résolution de ce dernier, à savoir H<sub>1</sub> et H<sub>2</sub> sont moins bien adaptées dans le cas de la résolution simultanée que l'heuristique H<sub>3</sub> qui elle inclue de l'information spécifique au problème de départ.

Enfin, si nous comparons les approches d'affectation simultanée et celle avec un ordonnancement donné (Figure 10), nous remarquons qu'il est globalement plus intéressant, du moins en ce qui concerne les méthodes testées, d'affecter les opérateurs avec un ordonnancement donné. Ceci est probablement dû au fait que les méthodes utilisées qui sont reconnues efficaces pour le problème du flow shop de base fournissent un point de départ qui se trouverait dans une zone prometteuse de l'espace de recherche. Néanmoins, pour un nombre élevé d'opérateurs et un plus grand nombre de tâches, nous constatons que l'heuristique d'affectation simultanée H<sub>3</sub> approche, voire surpasse, les méthodes susmentionnées. Ainsi, dans le cas de la disponibilité d'un grand nombre d'opérateurs -ce qui peut être un scénario intéressant dépendamment de la politique de l'entreprise en termes de coûts- il pourrait être plus intéressant d'utiliser l'heuristique d'affectation simultanée H<sub>3</sub> qui, de plus, présente l'avantage d'un temps d'exécution qui est bien moindre à celui de NEH-H<sub>1</sub> ou encore HFC-H<sub>1</sub>.

**Tableau 2. Évaluation des solutions pour  $F_m || C_{max}$**

		Bornes		NEH		HFC		Aléatoire					
		Sup	Inf	d <sup>+</sup>	d <sup>-</sup>	d <sup>+</sup>	d <sup>-</sup>	d <sup>+</sup>	d <sup>+max</sup>	d <sup>+min</sup>	d <sup>-</sup>	d <sup>+max</sup>	d <sup>+min</sup>
Instances	20x5	1278	1232	0.62	4.38	8.45	12.50	18.64	21.67	15.96	23.07	26.22	20.29
	20x10	1582	1448	6.19	16.02	5.82	15.60	37.45	41.28	31.92	50.17	54.35	44.13
	20x20	2297	1911	4.91	26.11	14.62	37.78	32.97	38.31	24.42	59.83	66.25	49.56
	50x5	2724	2712	0.33	0.77	1.83	2.28	15.35	17.44	13.62	15.86	17.96	14.12
	50x10	3025	2907	3.64	7.84	13.45	18.12	28.24	29.52	26.91	33.44	34.78	32.06
	100x5	5493	5437	0.47	1.51	5.27	6.36	8.51	9.05	8.19	9.62	10.17	9.31

**Tableau 3. Pourcentages de déviation moyens des solutions par rapport à la borne inférieure B<sub>1</sub>(k)**

		Méthodes															
		NEH				HFC				Aléatoire				Simultané			
		k	H <sub>1</sub>	H <sub>2</sub>	H <sub>3</sub>	H <sub>4</sub>	H <sub>1</sub>	H <sub>2</sub>	H <sub>3</sub>	H <sub>4</sub>	H <sub>1</sub>	H <sub>2</sub>	H <sub>3</sub>	H <sub>4</sub>	H <sub>1</sub>	H <sub>2</sub>	H <sub>3</sub>
Instances	20x5	2	41.0	39.1	93.0	63.2	48.8	39.8	89.9	61.5	57.2	54.8	89.9	62.4	193.1	206.1	136.3
		3	73.4	64.2	173.1	84.3	94.9	75.1	165.6	86.6	104.0	95.6	168.7	95.5	289.0	322.7	153.6
		4	100.8	79.3	244.4	100.3	122.8	93.3	226.9	103.6	136.8	127.6	233.8	125.2	326.9	318.6	215.2
	20x10	2	50.1	48.2	93.1	78.9	37.4	45.1	93.1	78.8	44.5	53.5	89.8	75.3	222.9	243.5	119.8
		3	89.0	72.7	176.1	133.8	72.3	72.8	176.1	133.8	81.5	91.6	167.9	133.2	310.6	373.4	158.5
		4	121.6	95.7	245.5	178.1	92.7	94.1	249.5	176.6	117.5	127.2	242.1	180.1	469.9	457.9	167.7
		5	147.7	116.3	307.7	199.7	115.6	122.6	306.7	204.5	146.4	156.1	307.7	214.2	448.7	371.8	212.6
		6	171.1	130.1	356.6	213.8	135.8	140.3	353.9	221.8	173.8	177.3	366.3	238.7	547.6	503.8	194.9
		7	189.6	147.1	392.7	233.8	155.2	152.3	393.3	234.0	200.5	195.2	406.8	264.9	581.0	628.8	211.4
	20x20	8	208.9	152.2	420.4	252.3	163.7	166.4	423.0	245.5	223.2	211.3	435.5	283.8	504.8	794.4	251.0
		9	221.2	165.4	432.6	261.7	174.2	172.7	436.1	258.9	236.4	230.2	447.0	302.1	591.7	825.7	279.6
		2	38.9	58.6	91.8	91.3	39.1	55.2	89.7	88.5	44.2	53.3	89.8	84.6	248.0	256.6	117.3
		3	74.5	100.4	175.7	172.3	75.9	93.2	171.5	166.1	82.2	95.8	173.1	156.7	344.0	368.5	135.8
		4	95.7	135.4	251.7	243.5	100.6	124.4	240.0	233.3	115.5	133.7	245.5	220.6	475.6	419.7	166.3
		5	117.0	163.8	314.1	304.4	122.0	149.0	300.9	292.5	143.1	164.5	308.8	277.1	600.4	621.9	166.8
		6	139.4	188.3	367.0	357.5	142.2	175.6	354.9	342.7	164.8	191.0	360.2	329.4	615.6	652.6	183.9
	7	156.2	209.6	409.8	390.6	159.5	194.0	396.7	373.9	178.4	209.4	400.9	369.9	772.2	680.5	199.8	

50x5	8	173.1	222.0	436.2	425.8	174.4	211.1	431.4	409.7	193.3	228.6	430.9	411.6	843.5	729.2	233.1	
	9	183.8	238.1	460.1	447.6	185.7	228.4	448.2	430.6	208.1	247.7	445.3	445.4	768.8	734.8	255.2	
	10	194.1	252.9	473.1	459.5	203.7	243.8	466.5	441.1	221.5	265.0	456.1	473.6	924.1	882.7	245.6	
	11	206.5	266.4	489.4	477.4	213.8	251.6	483.2	452.8	236.7	279.2	468.9	501.8	1067.9	873.5	300.7	
	12	214.5	281.4	506.0	502.7	225.3	269.3	494.6	485.0	251.0	292.4	486.5	522.1	1123.0	965.7	300.9	
	13	224.1	291.0	524.6	516.1	233.8	275.5	504.5	501.4	261.9	305.7	510.7	543.6	1052.6	983.9	301.6	
	14	233.6	296.8	540.6	532.8	245.9	283.7	516.2	513.8	271.6	315.6	523.7	559.6	1142.4	1075.0	296.0	
	15	243.3	308.8	556.5	547.7	255.3	292.2	534.7	526.2	280.6	327.8	541.7	575.2	1123.3	1170.3	318.1	
	16	245.5	310.2	561.5	551.5	256.6	300.0	539.6	536.5	284.2	330.4	545.4	582.2	1177.3	1167.9	324.7	
	17	252.1	319.5	573.0	561.2	262.6	307.5	547.9	538.6	290.9	336.1	544.7	590.1	1240.2	1176.5	336.5	
	18	258.8	324.7	584.1	568.4	270.7	310.8	558.0	542.6	296.2	342.6	549.2	599.9	1298.0	1174.8	353.7	
	19	262.8	331.3	584.4	565.9	274.8	313.1	566.5	547.9	301.3	346.8	549.1	603.8	1351.4	1394.6	340.0	
	50x5	2	66.1	62.4	97.7	63.4	52.2	63.1	95.6	63.5	61.3	67.9	97.0	63.9	200.8	196.3	143.4
		3	126.8	116.5	192.4	92.2	99.6	117.8	187.2	94.7	118.6	116.1	190.5	98.3	311.4	320.5	139.6
		4	184.1	157.0	283.8	107.4	142.0	157.9	275.1	108.9	165.6	156.6	280.3	118.3	368.3	455.1	200.9
	50x10	2	50.4	55.4	96.7	81.0	55.1	60.8	96.7	79.8	46.5	48.8	77.6	79.4	233.2	237.0	106.5
		3	95.1	96.1	190.1	142.9	101.3	109.8	189.9	140.1	85.8	82.6	188.2	140.8	330.4	347.4	140.3
		4	129.5	139.8	282.0	180.8	140.0	148.2	277.1	177.6	117.9	111.4	277.1	180.6	419.0	466.9	147.2
		5	163.8	177.5	367.9	201.0	177.1	185.1	363.3	199.6	148.4	141.1	361.4	207.6	537.6	505.2	165.3
6		195.2	211.2	450.5	219.2	208.0	219.4	444.8	217.0	176.2	166.0	441.3	232.6	624.2	668.3	166.7	
7		213.6	244.2	526.0	239.1	234.9	253.6	521.5	238.3	202.1	188.8	515.3	252.9	683.7	765.0	200.9	
8		239.4	271.9	599.1	257.1	258.8	275.9	592.3	262.0	226.2	209.9	590.6	274.7	737.6	941.8	227.7	
9		261.5	296.6	671.4	273.2	278.1	306.9	655.5	277.5	248.9	231.8	659.9	290.5	737.5	1031.5	248.3	
100x5		2	80.3	64.5	98.9	59.2	64.7	78.3	98.1	59.6	61.6	74.3	97.6	60.5	179.9	210.0	145.2
	3	152.8	121.0	196.2	81.1	122.6	147.4	195.0	87.3	117.2	137.7	193.9	89.1	308.6	323.5	151.4	
	4	216.4	171.5	291.9	103.0	175.8	212.6	288.9	104.0	166.8	194.1	288.2	109.1	401.5	464.0	232.4	

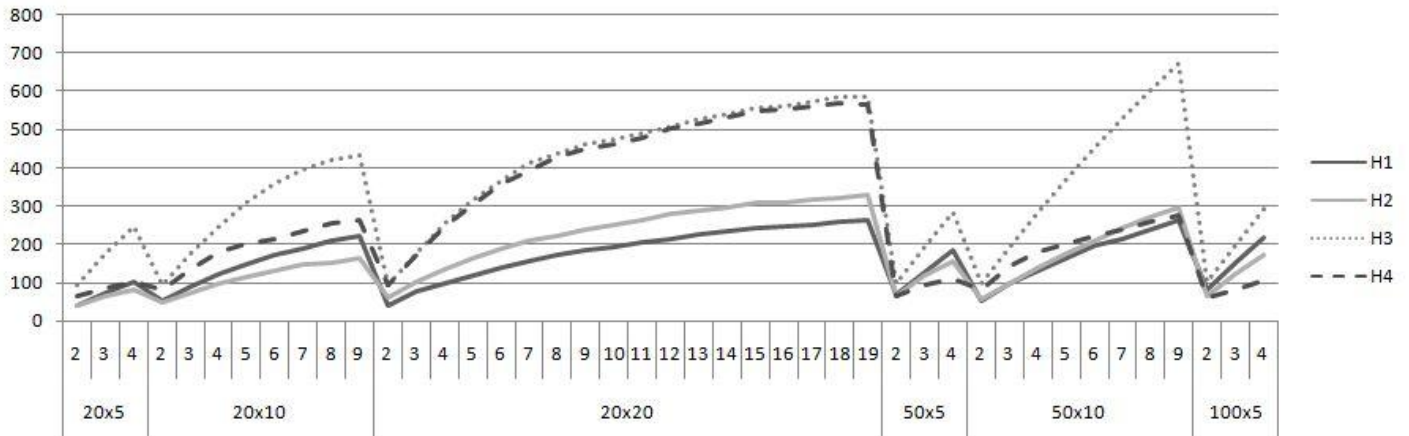
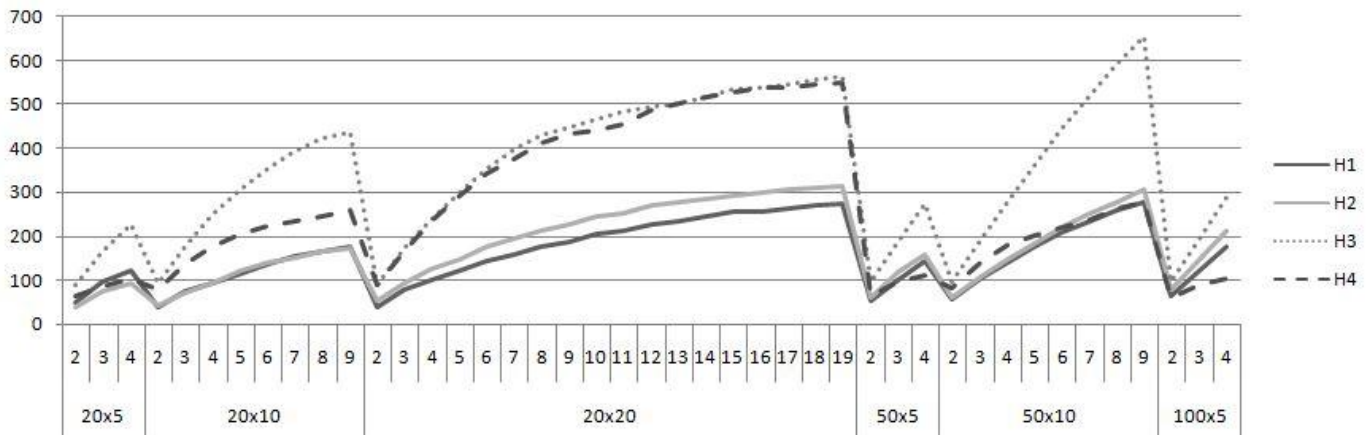
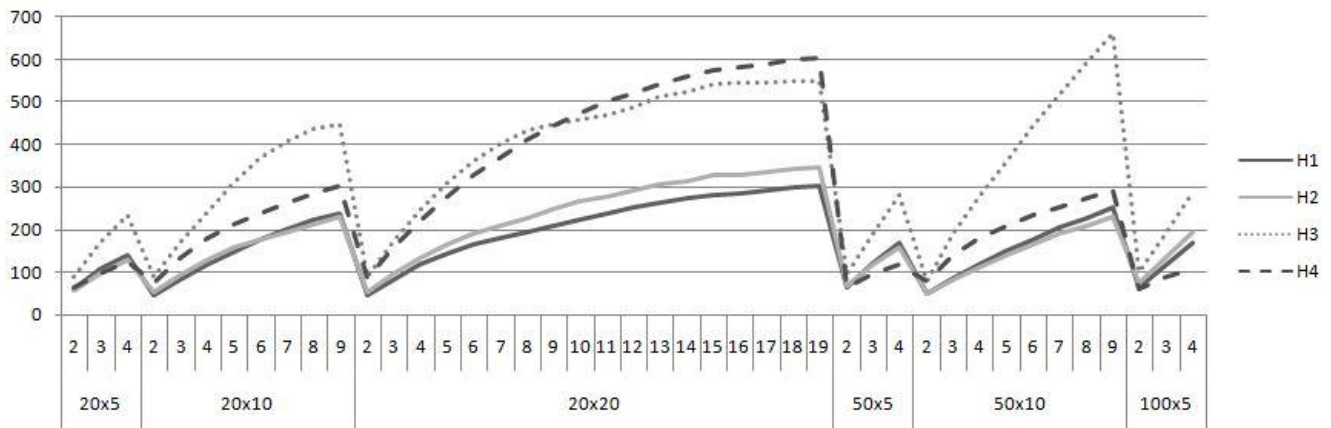


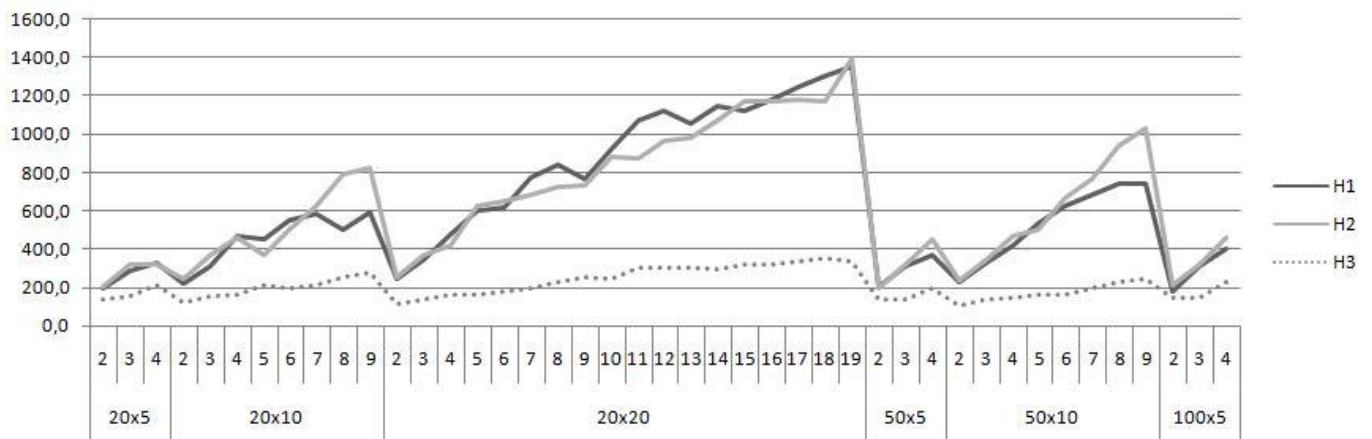
Figure 5. Comparaison des heuristiques d'affectation d'opérateurs avec NEH comme solution de départ



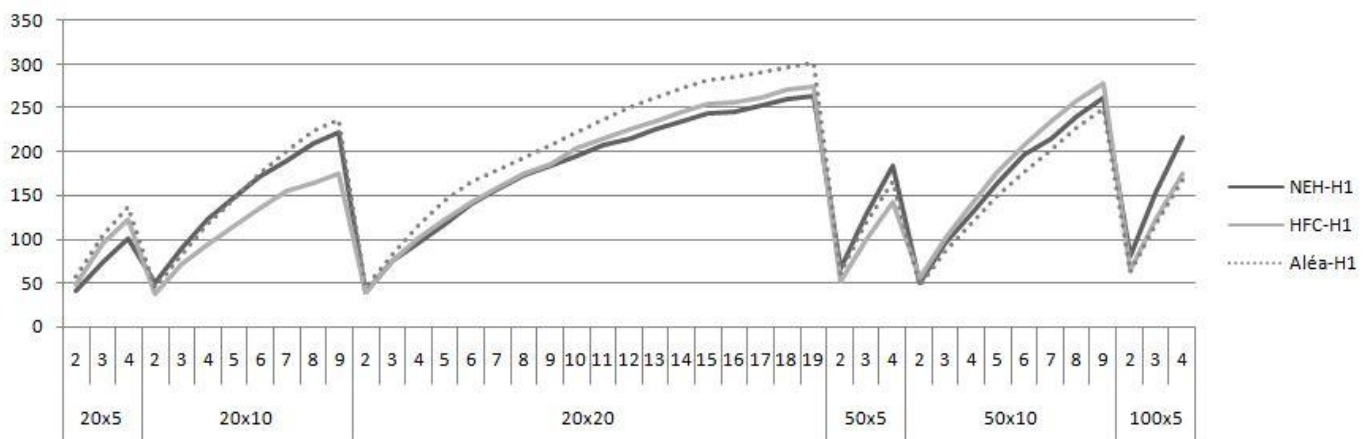
**Figure 6. Comparaison des heuristiques d'affectation d'opérateurs avec HFC comme solution de départ**



**Figure 7. Comparaison des heuristiques d'affectation d'opérateurs avec des solutions de départ aléatoires**



**Figure 8. Comparaison des heuristiques d'affectation simultanée d'opérateurs avec l'ordonnancement de tâches**



**Figure 9. Comparaison des heuristiques d'affectation d'opérateurs pour NEH, HFC et les solutions aléatoires**



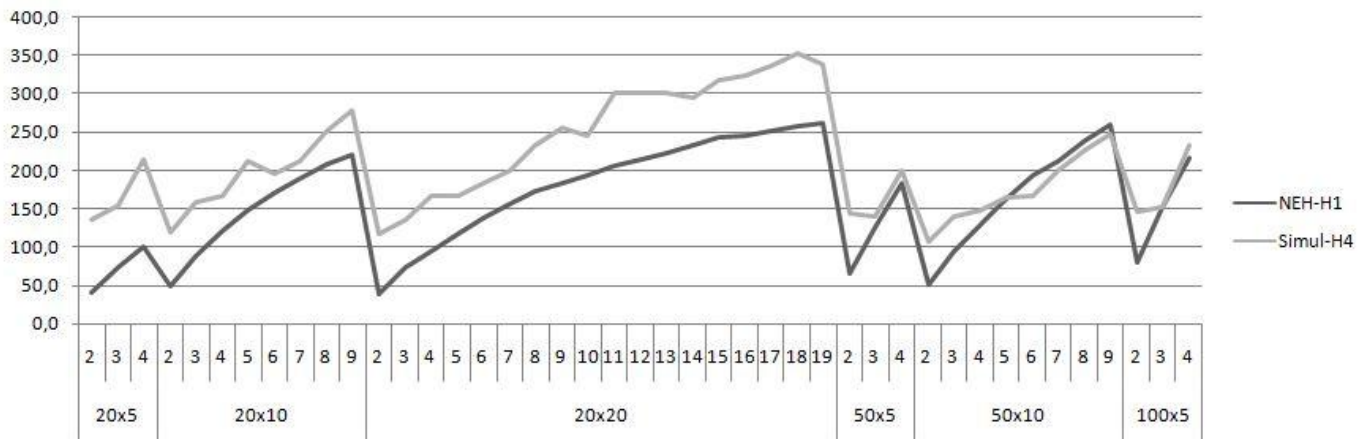


Figure 10. Comparaison des meilleures heuristiques pour NEH avec l'heuristique H<sub>4</sub> d'affectation simultanée

## 7 CONCLUSION

L'importance théorique ainsi que les applications pratiques du flow shop lui ont valu un grand intérêt de la part de la communauté scientifique. Dans l'environnement économique actuel qui est fortement concurrentiel, il devient important pour le bon fonctionnement des entreprises d'avoir une gestion efficace de leurs ressources ainsi que des systèmes d'aide à la décision adéquats. Dans cette optique, il est nécessaire de réduire l'écart entre la recherche académique et ses applications dans le monde réel, et ce, en développant des modèles qui soient de plus en plus réalistes.

Ainsi, la prise en compte des ressources humaines dans les systèmes productifs devient un domaine qu'il est utile d'étudier. Elle permet d'améliorer la flexibilité des dits systèmes tout en éliminant certaines dépenses non justifiées. Ces nouveaux problèmes remettent en cause l'ordonnancement "classique" et présentent de nouvelles contraintes (par exemple, la réglementation de travail) et de nouveaux objectifs (par exemple les coûts). L'intégration des ressources humaines dans les systèmes productifs est un domaine de recherche relativement récent, très peu étudié et qui reste largement ouvert. C'est avec l'augmentation de l'importance de la composante humaine dans les systèmes de gestion en général, et dans les systèmes industriels en particulier, que ce pan de la théorie de l'ordonnancement a vu le jour. Il a pour objectif, notamment le développement d'approches intégrées reposant sur une modélisation solide de l'acteur humain dans l'environnement de travail, en particulier l'aspect traitant de l'incidence qu'a le partage d'opérateurs sur les durées opératoires des tâches qu'ils traitent.

Dans le cadre de ce travail, nous avons étudié l'ordonnancement de tâches dans un environnement de type flow shop en présence d'opérateurs, dont le nombre est inférieur à celui des machines, et, pour objectif, la minimisation du makespan. Après la démonstration de la NP-difficulté au sens fort de notre problème, nous avons comparé deux manières de le résoudre, à savoir, l'affectation d'opérateurs étant donné un ordonnancement et le traitement simultané des aspects humains et matériels du problème. Les meilleures heuristiques ont donné de bons

résultats, proches des bornes théoriques. D'autre part, il s'est avéré que, pour les méthodes étudiées, la meilleure approche de traitement simultanée était moins performante à l'exception des cas avec un grand nombre d'opérateurs et de tâches où elle donnait non seulement de meilleurs résultats, mais aussi avec un temps d'exécution réduit. Il serait intéressant de tester l'efficacité de cette heuristique sur des problèmes de plus grande taille.

Concernant les perspectives pour de futurs travaux, il serait intéressant de considérer d'autres objectifs tels que le temps moyen d'accomplissement des tâches ( $\sum C_j$ ) et le maximum des retards des tâches ( $L_{max}$ ), qui ont aussi une grande importance en pratique. Il serait également intéressant d'étudier des configurations d'ateliers avec des opérateurs ayant des niveaux différents de compétences ou encore en tenant compte de leurs temps de déplacement.

## 8 REFERENCES

- Aneja, Y.P., Kamoun, H., (1999) Scheduling of parts and robot activities in a two machine robotic cell. *Computers & Operations Research*, 26 :297–312.
- Baptiste, P., Giard, V., Hait, A., Soumis, F., (2005) *Gestion de production et ressources humaines*. Presses Internationales Polytechnique, Québec, Canada.
- Baptiste, P., Munier, A., (2013) Ordonnancement sur machines parallèles avec partage d'opérateurs. *10<sup>ème</sup> Congrès International de Génie Industriel. La Rochelle, 12-14 juin*.
- Baki, M.F., (1999) *Some Problems in One-Operator Scheduling*, Thèse de doctorat, University of Waterloo.
- Blazewicz, J., Ecker, K., Pesch, E., Schmidt, G., Weglarz, J., (2007) *Handbook on scheduling: From theory to application*, Springer.
- Burns, F., Rooker, J., (1976) Johnson's three-machine flow-shop conjecture, *Operations Research*, 24 : 578-580.
- Cheng, T.C.E., Wang, C., Sriskandarajah, C., (1999) One-operator-two-machine flowshop scheduling with setup and dismantling times. *Computers & Operations Research*, 26:715–730.
- Cheurfa, M., (2005) *Gestion des ressources humaines en production cyclique*, Thèse de doctorat, ENSM Saint-Étienne, France.

- Dawande, M., Geismar, H.N., Sethi, S.P., Sriskandarajah, C., (2005) Sequencing and scheduling in robotic cells: Recent developments. *Journal of Scheduling*, 8: 387–426.
- Du, J., Leung, J.Y-T., Young, G.H., (1991) Scheduling chain-structured tasks to minimize makespan and flow time, *Information and Computation*, 92:219–236.
- Färber, G., Salhi, S., Coves Moreno, A.M., (2007) Sequencing in a non-permutation flowshop with constrained buffers: Applicability of genetic algorithm versus constraint logic programming. *XI Congreso de Ingeniería de Organización International Conference on Industrial Engineering and Industrial Management, Madrid*, 5-7 September.
- Garey, M.R., Johnson, D.S., Sethi, R., (1976) The complexity of flowshop and jobshop scheduling. *Mathematics of Operations Research*, 1:117–129.
- Geismar, H.N., Pinedo, M., Sriskandarajah, C., (2008) Robotic cells with parallel machines and multiple dual gripper robots: a comparative overview. *IIE Transactions*, 40(12):1211–1227.
- Graham, R.L., Lawler, E.L., Lenstra, J.K., RinnooyKan, A.H.G., (1979) Optimization and approximation in deterministic sequencing and scheduling theory: a survey. *Annals of Discrete Mathematics*, 5:287–326.
- Gupta, J.N.D., Stafford Jr., F., (2006) Flowshop scheduling research after five decades. *European Journal of Operational Research*, 169: 699–711.
- Hall, N.G., Potts, C.N., Sriskandarajah, C., (2000) Parallel machine scheduling with a common server. *Discrete Applied Mathematics*, 102:223–243.
- Johnson, S.M., (1954) Optimal two and three-stage production schedules with setup times included. *Naval research logistics quarterly*, 1(1):61–68.
- Koulamas, C., (1998) A new constructive heuristic for the flow shop scheduling problem. *European Journal of Operational Research*, 105:66–71.
- Liao, C.J., Liao, L.M., Tseng, C.T., (2007) A performance evaluation of permutation vs. non-permutation schedules in a flowshop. *International Journal of Production Research*, 44 (20): 4297–4309.
- Nawaz, M., Enscore Jr, E., Ham, I., (1983) A heuristic algorithm for the m-machine n-job flow-shop sequencing problem. *Omega, The International Journal of Management Science*, 11 :91–95.
- Noorulhaq, A., Saravanan, M., Vivekraj, A.R., Prasad, T., (2007) A scatter search approach for general flowshop scheduling problem. *International Journal of Advanced Manufacturing Technology*, 31:731–736.
- Oulamara, A., Rebaine, D., Serairi, M., (2013) Scheduling the two-machine open shop problem under resource constraints for setting the jobs. *Annals of Operations Research*, 211 :333–356.
- Pinedo, M., (2002) *Scheduling: Theory, algorithms and systems*. Prentice Hall.
- Potts, C.N., Shmoys, D.B., Williamson, D.P., (1991) Permutation vs. non permutation flowshop schedules. *Operations Research Letters*, 10:281–284.
- Ruiz, R., Maroto, C., Alcaraz, J., (2004) Solving the flowshop scheduling problem with sequence dependent setup times using advanced metaheuristics. *European Journal of Operational Research*, 217:34–54.
- Sethi, S.P., Sriskandarajah, C., Sorger, G., Blazewicz, J., Kubiak, W., (1992) Sequencing of parts and robot moves in a robotic cell. *The International Journal of Flexible Manufacturing Systems*, 4:331–358.
- Vahedi-Nouri, B., Fattahi, P., Ramezani, R., (2013) Minimizing total flow time for the non-permutation flow shop scheduling problem with learning effects and availability constraints. *Journal of Manufacturing Systems*, 32 :167– 173.
- Vickson, R.G., (1980) Choosing the job sequence and processing times to minimize total processing plus flow cost on a single machine, *Operations Research*, 28(5) :1155–1167.
- Vickson, R.G., (1980) Two single machine sequencing problems involving controllable job processing times. *AIIE transactions*, 12(3) :258–262.
- Ying, K-C., Gupta, J.N.D., Lin S-W., Lee, Z-J., (2010) Permutation and non permutation schedules for the flowline manufacturing cell with sequence dependent family setups. *International Journal of Production Research*, 48(8) :2169–2184.
- Ying, K-C., Lin, S-W., (2007) Multi-heuristic desirability ant colony system heuristic for non-permutation flow shop scheduling problems. *International Journal of Advanced Manufacturing Technology*, 33 :793–802.
- Ziaee, M., Sadjadi, S.J., (2007) Mixed binary integer programming formulations for the flow shop scheduling problems. a case study: ISD projects scheduling. *Applied Mathematics and Computation*, 185 :218–228.
- Zouba, M., (2009) *Ordonnement de machines parallèles identiques avec des contraintes de ressources humaines*. Thèse de doctorat, École Polytechnique de Montréal, Québec, Canada.