

# Ordonnancement de tâches sur deux machines parallèles avec un seul opérateur : approche géométrique

MOHAMMED ZOUBA<sup>1</sup>, PIERRE BAPTISTE<sup>1</sup>, DJAMAL REBAINE<sup>2</sup>, FRANÇOIS SOUMIS<sup>1</sup>

<sup>1</sup> ÉCOLE POLYTECHNIQUE DE MONTRÉAL  
C.P. 6079, succ. Centre-ville, Montréal, Canada, H3C 3A7

[mohammed.zouba@polymtl.ca](mailto:mohammed.zouba@polymtl.ca), [pbaptiste@polymtl.ca](mailto:pbaptiste@polymtl.ca), [francois.soumis@gerad.ca](mailto:francois.soumis@gerad.ca)

<sup>2</sup> UNIVERSITÉ DU QUÉBEC À CHICOUTIMI  
555, boul. de l'Université, Saguenay, Canada G7H 2B1  
[djamal\\_rebaine@uqac.ca](mailto:djamal_rebaine@uqac.ca)

---

**Résumé** – Dans cet article, nous étudions le problème d’ordonnancement de tâches non préemptives sur deux machines parallèles identiques en présence d’un opérateur, en vue de minimiser le critère du makespan. Dans notre contexte, l’opérateur agit comme un superviseur et sa présence est nécessaire pour l’exécution des tâches. Nous considérons que l’opérateur peut s’occuper des machines de plusieurs manières, en leur donnant des priorités différentes. Il s’ensuit que le temps d’exécution des tâches devient variable car dépendant de l’affectation de l’opérateur aux machines. Nous considérons le cas où les affectations de l’opérateur peuvent être changées à tout moment. Nous proposons une représentation géométrique du problème et donnons quelques propriétés caractérisant l’optimalité d’une solution. Nous développons par la suite un algorithme pour construire une solution optimale.

**Abstract** - This paper addresses the problem of scheduling non-preemptive jobs on two identical parallel machines with a single operator acting as a supervisor. The criterion to minimize is the makespan. In our context, the presence of an operator is necessary to operate the machines through a subset of a given modus operandi: the working modes. A working mode models the way the operator divides up his interventions between the machines, and consequently the ratio of productivity induced on each machine. Within this model, the processing times become variable as they depend on the working mode being utilized. A schedule is thus a partition of the jobs on the machines with the corresponding sequences, a subset of working modes along with their duration, and the order of their utilization. In this paper, we present a geometric representation of the problem, and properties of an optimal solution. Based on these results, an algorithm to generate an optimal makespan within the free changing mode is developed.

**Mots clés** - Ordonnancement, machines parallèles, makespan, opérateur, changement libre.

**Keywords** - Scheduling, parallel machines, makespan, operator, free changing mode.

---

## 1 INTRODUCTION

Jusqu’à un passé récent, les travaux de recherche en ordonnancement de production se sont concentrés sur les contraintes liées aux machines et aux opérations : disponibilité des machines, ordre d’exécution des opérations, etc. En effet, la complexité des problèmes d’ordonnancement d’atelier dans la réalité est la difficulté d’offrir un cadre théorique intégrant l’ensemble de la réalité industrielle a conduit à des hypothèses simplificatrices [Pinedo, 2002]. Ces modèles se sont donc limités au séquençement des travaux et relaxent, en général, les contraintes liées à la gestion des opérateurs dans l’atelier, en supposant que leur nombre est infini.

Peu d’articles ont pris en compte simultanément les machines et les ressources humaines dans le processus de construction d’un ordonnancement. Toutefois, quand c’est le cas, l’approche utilisée se compose de deux phases : l’ordonnancement de l’atelier est d’abord réalisé et, ensuite, les opérateurs sont affectés aux machines en modifiant localement, au besoin, la solution de la première phase [Huq *et al.*, 2004; Hu, 2004; Hu, 2005] ou vice versa comme cela est fait dans Karlof et Wang [1996].

Pinedo [2002] souligne l’importance de l’étude de modèles combinant les problèmes d’ordonnancement des machines à celui des ressources humaines (opérateurs). Ces problèmes d’ordonnancement, intégrant l’affectation des opérateurs aux machines, consistent à déterminer l’état de l’atelier au cours du temps [Cheurfa, 2005]. Bourland et Carl [1994] se sont, quant à eux, intéressés à l’ordonnancement de machines parallèles avec des ressources humaines partagées. Ils considèrent qu’une fraction seulement de la ressource est utilisée. L’affectation des opérateurs tient compte de la compétence de ces derniers ainsi que des contraintes sociales liées à la législation du travail.

Depuis les travaux de Vickson [1980a, 1980b], de nombreuses études ont porté sur des modèles d’ordonnancement avec des temps opératoires contrôlables par la quantité de ressources (humaines ou autres) allouées aux opérations [Daniels *et al.*, 1996; Grigoriev *et al.*, 2005; Shabtay et Steiner, 2007]. La majorité de ces travaux porte sur l’analyse de la complexité de ces problèmes ainsi que l’application d’heuristiques, étant donné que ces problèmes, sans la contrainte de ressources, sont déjà NP-Complets. Dans [Daniels *et al.*, 1996], les auteurs ont étudié le problème d’ordonnancement sur des machines

parallèles identiques avec pré-affectation des tâches aux machines. Ils ont développé un algorithme de type branch & bound énumérant les affectations possibles des ressources et calcule, pour une affectation donnée, des bornes inférieures pour le makespan. Grigoriev *et al.* [2005] considèrent le problème d'ordonnement de machines parallèles non dépendantes où les temps d'exécution sont fonction du nombre d'opérateurs affectés à une machine à un instant donné. Ils utilisent une technique en deux phases, basée sur la programmation linéaire, pour affecter les opérateurs aux tâches et les tâches aux machines, de façon à minimiser le makespan. Ils produisent une borne inférieure pour ce critère et des résultats d'approximation. Ils montrent également que l'approche utilisée peut être adaptée au cas de machines dédiées. Shabtay et Steiner [2007], quant à eux, présentent un état de l'art sur cette problématique.

Un modèle dans lequel un opérateur partage ses interventions sur plusieurs machines parallèles identiques a été étudié en premier par Tchommo [2004]. Dans ce travail, les temps opératoires sont variables. Sur chaque machine, le temps opératoire d'une tâche est divisé par une valeur inférieure ou égale à 1 pour devenir le temps opératoire réel. Cette valeur peut différer d'une machine à l'autre. Le même modèle est également étudié dans [Cheurfa, 2005], mais pour un problème de flow-shop cyclique. Une description générale de ce problème et d'autres considérations sont discutées dans [Baptiste *et al.*, 2005].

Dans cet article, nous présentons un modèle où un opérateur partage ses interventions sur plusieurs machines. Ceci induit donc des durées d'exécution variables pour les tâches. Nous supposons que la dégradation de la durée d'exécution est linéaire (indépendante de la tâche réalisée). Ainsi, une affectation donnée d'un opérateur sur un ensemble de machines peut être vue comme un vecteur de coefficients de pondération des durées effectuées sur chacune des machines. D'autre part, nous considérons que si un opérateur peut s'occuper de plusieurs machines en parallèle, il peut le faire de plusieurs manières, en donnant des priorités différentes aux machines. Il peut donc exister plusieurs affectations (vecteurs de coefficients de pondération) pour le même groupe de machines.

Cette présente étude fait suite aux travaux de Zouba *et al.* [2006, 2008, 2009a, 2009b]. Dans le premier article, les auteurs ont développé des propriétés d'une solution optimale avec un seul opérateur et deux machines. Ils y ont considéré des changements d'affectations libres. Dans les deux articles suivants, les auteurs ont proposé une approche heuristique pour la résolution du problème avec plusieurs machines et plusieurs opérateurs en considérant seulement le changement calendaire. Dans le dernier article, ils ont considéré le mode de changement en fin de tâche et ont proposé une approche heuristique pour la résolution du problème avec plusieurs machines et plusieurs opérateurs. Dans le présent travail, nous considérons le changement libre et proposons un algorithme exact qui génère une solution optimale pour le cas de deux machines parallèles identiques et un seul opérateur.

Cet article est organisé comme suit. La Section 2 décrit le problème étudié. En Section 3 nous présentons une approche géométrique pour représenter le problème et quelques propriétés d'une solution optimale. La Section 4 est consacrée à la présentation de quelques propriétés, caractérisant une

solution optimale, et à notre principal résultat, à savoir, un algorithme construisant une solution optimale pour le cas de deux machines parallèles identiques et un seul opérateur. En Section 5 sont exposées notre conclusion et les perspectives pour un travail futur.

## 2 DESCRIPTION DU PROBLÈME

Le problème consiste à réaliser un ensemble de  $n$  tâches  $J = \{J_i : i = 1, \dots, n\}$  sur un ensemble de  $m$  machines parallèles identiques  $M = \{m_j : j = 1, \dots, m\}$ . Nous supposons que les tâches et les machines sont disponibles continuellement dès l'instant  $t = 0$ . Pour qu'une tâche,  $J_i$ , puisse être exécutée sur une machine, il est nécessaire que celle-ci soit supervisée par un opérateur. Le temps d'exécution de la tâche  $J_i$  sur la machine  $m_j$  est noté par  $p_{ij}$ , connu à l'avance, si la machine est supervisée exclusivement par un opérateur. Le temps  $p_{ij}$  est appelé temps opératoire de base de la tâche  $J_i$ . Nous supposons que le nombre d'opérateurs en présence dans l'atelier est inférieur au nombre de machines. Ceci fait que chaque opérateur doit s'occuper de plusieurs machines à la fois. La productivité de chacune des machines peut se trouver ainsi affectée. Autrement dit, les temps opératoires varient selon le taux de productivité induit par le mode de fonctionnement (affectation) choisi. Signalons que pour une même disposition des opérateurs, plusieurs vecteurs de productivité peuvent être définis. En effet, soit  $x_{kj} \in [0, 1]$  le taux de productivité de la machine  $m_j$  pour une affectation  $AT_k$  des opérateurs. Une affectation  $AT_k$ ,  $k = 1, \dots, l$  ( $l$  étant le nombre d'affectations) peut être définie comme suit :

$$AT_k = (x_{kj} : j = 1, \dots, m).$$

Pour une affectation  $AT_k$ , le temps opératoire  $p_{ij}$ , de la tâche  $J_i$  sur la machine  $m_j$  devient donc  $p_{ijk} = p_{ij} / x_{kj}$ .

Par conséquent, l'ordonnement en présence d'opérateurs est caractérisé par la connaissance du séquençement des tâches sur chacune des machines et les différentes affectations utilisées par les opérateurs ainsi que la durée de leur utilisation.

**Définition 1 :** Le taux global de service de l'affectation  $AT_k$  est

$$\tau_k = \sum_{j=1}^m x_{kj}.$$

En général, nous avons  $\tau_k > w$ ,  $w$  étant le nombre d'opérateurs, du moment que chaque opérateur supervise plusieurs machines simultanément. Plus le niveau d'automatisation des machines est grand, plus grande est la valeur de  $\tau_k$ . Dans tous les cas, nous avons  $\tau_k \in [w, m]$ . Le premier cas extrême,  $\tau_k = w$ , a lieu quand le fonctionnement des machines est complètement manuel. Le temps d'exécution d'une tâche est exactement égal au temps que l'opérateur passe sur la machine. Le second cas extrême,  $\tau_k = m$ , a lieu quand les machines sont complètement automatisées. L'opérateur, dans ce cas, s'occupe du chargement (et déchargement) des machines et d'opérations manuelles mineures. Par conséquent, l'opérateur est disponible à tout instant pour intervenir.

Notons que l'affectation d'un opérateur peut changer au cours du temps. On distingue ainsi trois modes de changement, à savoir le mode libre, calendaire et en fin de tâche. Dans le mode libre, l'affectation des opérateurs peut changer à tout instant. Dans le mode calendaire, la durée des affectations est un multiple d'une période : journée, demi-journée, etc. En fin de tâche, l'opérateur peut changer d'affectation dès qu'il termine une des tâches dont il s'occupe.

Le problème que nous étudions consiste donc à déterminer des intervalles de temps d'utilisation des différentes affectations et ordonnancer l'ensemble des tâches sur les machines de façon à minimiser le makespan.

Dans ce travail, nous limitons notre étude au changement libre et nous proposons un algorithme exact générant une solution optimale pour le cas de deux machines parallèles identiques. Notons que ce problème est NP-Complet. En effet, dans le cas d'une seule affectation, il est facile de voir que le problème considéré est équivalent au problème classique de machines uniformes.

**Définition 2 :** Une affectation  $AT_k = (x_{k1}, x_{k2}, \dots, x_{km})$  est dite dominée s'il existe des réels positifs  $\alpha_1, \alpha_2, \dots, \alpha_l$

avec  $\sum_{h=1, h \neq k}^l \alpha_h = 1, 0 \leq \alpha_h \leq 1$ , tels que pour tout  $j = 1, \dots, m$

$$x_{kj} \leq \sum_{h=1, h \neq k}^l \alpha_h x_{hj}$$

et il existe au moins un indice  $j, j = 1, \dots, m$ , tel que

$$x_{kj} < \sum_{h=1, h \neq k}^l \alpha_h x_{hj}.$$

### 3 REPRÉSENTATION GRAPHIQUE DU PROBLÈME

Le problème peut être représenté géométriquement dans un espace à  $m$  dimensions. Chaque dimension correspond à une machine. Chaque affectation est un point de l'espace dont chaque coordonnée correspond à la productivité d'une machine. Chaque partition des tâches est également représentée par un point dont chaque coordonnée correspond au pourcentage de travail alloué à la machine correspondante.

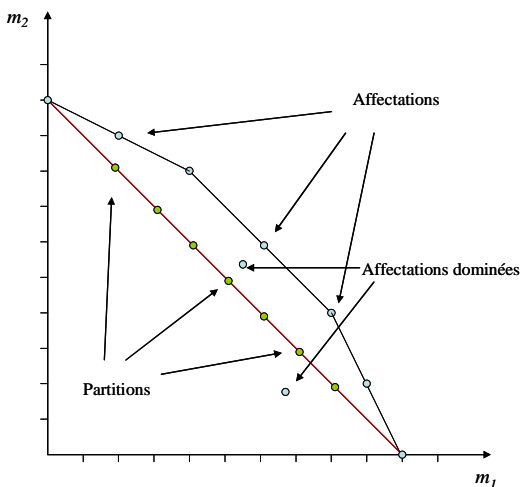


Figure 1. Représentation graphique du problème

Si le nombre de machines est 2, le problème sera représenté sur le plan. Si les affectations ont le même taux de service,  $\tau$ , elles se situent sur une droite (dont l'équation est  $x + y = \tau$ ). Les partitions (pourcentage de travail sur chaque machine) des tâches sur les deux machines seront également représentées sur une droite (dont l'équation est  $x + y = 1$ , voir Figure 1). Notons que dans le cas où les affectations ont des taux de services différents, nous pouvons avoir des affectations qui sont dominées (voir Figure 1).

### 4 CAS DE DEUX MACHINES

Si les points représentant les affectations non dominées sont reliés par des segments de droite, nous formerons une courbe concave (que nous appellerons courbe des affectations). Avant d'énoncer les propriétés mathématiques déduites de la représentation géométrique, nous donnons un exemple pour illustrer cette dernière.

**Exemple 1 :** Considérons l'instance suivante avec 2 machines, 3 tâches et 5 affectations. Les données sont comme suit :  $p_1 = 10, p_2 = 6, p_3 = 4, AT_1 = (0,1), AT_2 = (0.4,0.8), AT_3 = (0.6,0.6), AT_4 = (0.8,0.4)$  et  $AT_5 = (1,0)$ .

Cette instance peut être représentée sur un plan (Figure 2).

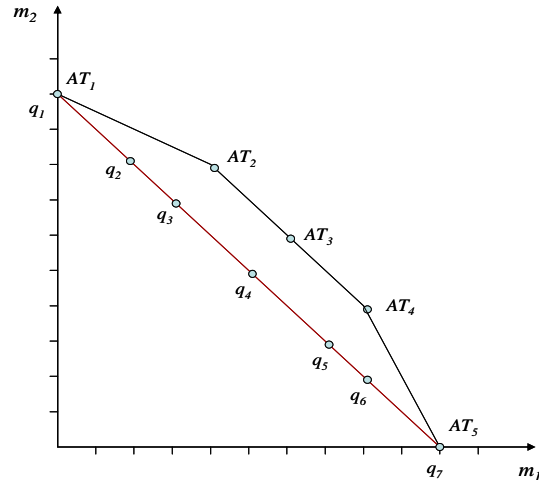


Figure 2. Exemple pour deux machines

Nous aurons sur les axes les machines  $m_1$  et  $m_2$ . La partition qui consiste à affecter la tâche  $J_1$  à  $m_1$  et les tâches  $J_2$  et  $J_3$  à  $m_2$  est représentée par le point de coordonnées  $(0.5, 0.5)$ , c'est-à-dire que nous aurons 50% du travail sur chaque machine (10 unités de temps sur  $m_1$  et 10 unités de temps sur  $m_2$ ). La partition qui consiste à affecter  $J_1$  et  $J_2$  à  $m_1$  et  $J_3$  à  $m_2$  est représentée par le point  $(0.8, 0.2)$ . Nous aurons donc 7 points pour représenter les différentes partitions possibles, comme le montre la Figure 2:

$q_1(0,1), q_2(0.8,0.2), q_3(0.3,0.7), q_4(0.5,0.5), q_5(0.7,0.3), q_6(0.8,0.2)$  et  $q_7(1,0)$ .

**Théorème 1 :** La recherche d'une solution optimale peut se limiter aux affectations non dominées.

**Preuve :** Supposons qu'une solution optimale  $S$  utilise une affectation dominée  $AT_k$ . Soient  $t_1, t_2, \dots, t_l$  les durées respectives d'utilisation des affectations  $AT_1, AT_2, \dots, AT_l$ .

Supposons également que la partition optimale soit  $(Y_1, Y_2)$ .  
Posons

$$A = \sum_{i \in Y_1} p_i \text{ et } B = \sum_{i \in Y_2} p_i.$$

Nous avons

$$\begin{aligned} A &= t_1 x_{1,1} + \dots + t_k x_{k,1} + \dots + t_l x_{l,1}, \\ B &= t_1 x_{1,2} + \dots + t_k x_{k,2} + \dots + t_l x_{l,2}. \end{aligned}$$

Comme  $AT_k$  est dominée, il existe des réels positifs

$\alpha_1, \alpha_2, \dots, \alpha_l$  avec  $\sum_{h=1, h \neq k}^l \alpha_h = 1$ ,  $0 \leq \alpha_h \leq 1$ , tels que, pour  $j = 1, 2$ , nous avons

$$x_{kj} \leq \sum_{h=1, h \neq k}^l \alpha_h x_{hj},$$

et il existe au moins un indice  $j$ ,  $j = 1, 2$ , tel que

$$x_{kj} < \sum_{h=1, h \neq k}^l \alpha_h x_{hj}.$$

Nous aurons

$$\begin{aligned} A &\leq (t_1 + t_k \alpha_1) x_{1,1} + \dots + (t_{k-1} + t_k \alpha_{k-1}) x_{k-1,1} + \\ &(t_{k+1} + t_k \alpha_{k+1}) x_{k+1,1} + \dots + (t_l + t_k \alpha_l) x_{l,1}, \end{aligned}$$

et

$$\begin{aligned} B &\leq (t_1 + t_k \alpha_1) x_{1,2} + \dots + (t_{k-1} + t_k \alpha_{k-1}) x_{k-1,2} + \\ &(t_{k+1} + t_k \alpha_{k+1}) x_{k+1,2} + \dots + (t_l + t_k \alpha_l) x_{l,2}. \end{aligned}$$

Nous aurons ainsi une solution  $S'$  n'utilisant pas l'affectation  $AT_k$  et dont le makespan est

$$\begin{aligned} C_{\max}(S') &\leq (t_1 + t_k \alpha_1) + \dots + (t_{k-1} + t_k \alpha_{k-1}) + \\ &(t_{k+1} + t_k \alpha_{k+1}) + \dots + (t_l + t_k \alpha_l) \\ &\leq \sum_{h=1, h \neq k}^l t_h + t_k \sum_{h=1, h \neq k}^l \alpha_h \\ &\leq \sum_{h=1}^l t_h \\ &\leq C_{\max}(S). \end{aligned}$$

La nouvelle solution a donc un makespan au moins égale à celui de la solution de départ  $S$ .

D'où le résultat.

#### 4.1 Propriétés d'une solution optimale

Dans cette section, nous donnons quelques propriétés caractérisant une solution optimale.

**Théorème 2 :** Une partition  $(Y_1, Y_2)$  de l'ensemble  $J$  des tâches est dans le cône de sommet  $O$  (origine du plan) et passant par les points représentant les deux affectations  $AT_1 = (x_{11}, x_{12})$  et  $AT_2 = (x_{21}, x_{22})$  si, et seulement si,

$$\min_{j=1,2} \left( \frac{x_{j1}}{\tau_j} \right) \leq \frac{\sum_{i \in Y_1} p_i}{\sum_{i \in J} p_i} \leq \max_{j=1,2} \left( \frac{x_{j1}}{\tau_j} \right).$$

**Preuve :** Supposons, sans perte de généralité, que  $AT_1$  possède la plus petite productivité sur la machine  $m_1$ . Nous aurons donc

$$\frac{x_{11}}{\tau_1} \leq \frac{x_{21}}{\tau_2}.$$

Soit  $q_1$  le point d'intersection des droites d'équations respectives

$$y = \frac{x_{12}}{x_{11}} x,$$

et

$$x + y = 1.$$

Soit également  $q_2$  le point d'intersection des droites d'équations respectives

$$y = \frac{x_{22}}{x_{21}} x,$$

et

$$x + y = 1.$$

Le point représentant la partition  $(Y_1, Y_2)$  dont les coordonnées

sont  $\left( \frac{\sum_{i \in Y_1} p_i}{\sum_{i \in J} p_i}, \frac{\sum_{i \in Y_2} p_i}{\sum_{i \in J} p_i} \right)$  est dans le cône des deux affectations

$AT_1 = (x_{11}, x_{12})$  et  $AT_2 = (x_{21}, x_{22})$  si, et seulement si, il est situé sur le segment de droite joignant les points  $q_1$  et  $q_2$ . Il est facile de vérifier que les points  $q_1$  et  $q_2$  ont pour

coordonnées  $\left( \frac{x_{11}}{\tau_1}, \frac{x_{12}}{\tau_1} \right)$  et  $\left( \frac{x_{21}}{\tau_2}, \frac{x_{22}}{\tau_2} \right)$  respectivement. Il en

résulte que le point représentant la partition  $(Y_1, Y_2)$  est dans le cône des deux affectations  $AT_1 = (x_{11}, x_{12})$  et  $AT_2 = (x_{21}, x_{22})$  si, et seulement si,

$$\frac{x_{11}}{\tau_1} \leq \frac{\sum_{i \in Y_1} p_i}{\sum_{i \in J} p_i} \leq \frac{x_{21}}{\tau_2}.$$

D'où le résultat.

**Théorème 3 :** Soient  $AT_1 = (x_{11}, x_{12})$  et  $AT_2 = (x_{21}, x_{22})$  deux affectations quelconques. S'il n'existe pas de partition  $(Y_1, Y_2)$  telle que

$$\min_{j=1,2} \left( \frac{x_{j1}}{\tau_j} \right) < \frac{\sum_{i \in Y_1} p_i}{\sum_{i \in J} p_i} < \max_{j=1,2} \left( \frac{x_{j1}}{\tau_j} \right), \quad (1)$$

alors la solution optimale utilise une seule affectation,  $AT_1$  ou  $AT_2$ .

**Preuve :** Considérons une partition  $(Y_1, Y_2)$  de l'ensemble  $J$  des tâches. Si cette partition ne vérifie pas (1) alors elle ne vérifiera pas (2) car ces deux relations sont équivalentes.

$$\min_{j=1,2} \left( \frac{x_{j2}}{\tau_j} \right) < \frac{\sum_{i \in Y_1} p_i}{\sum_{i \in J} p_i} < \max_{j=1,2} \left( \frac{x_{j2}}{\tau_j} \right). \quad (2)$$

Posons

$$A = \sum_{i \in Y_1} p_i \text{ et } B = \sum_{i \in Y_2} p_i.$$

Sans perte de généralité, nous pouvons supposer

$$\frac{x_{11}}{\tau_1} \leq \frac{x_{21}}{\tau_2}.$$

Nous aurons alors

$$\frac{x_{22}}{\tau_2} < \frac{x_{12}}{\tau_1}.$$

Les expressions (1) et (2) peuvent s'écrire alors

$$\frac{x_{11}}{\tau_1} < \frac{A}{A+B} < \frac{x_{21}}{\tau_2},$$

$$\frac{x_{22}}{\tau_2} < \frac{B}{A+B} < \frac{x_{12}}{\tau_1}.$$

Considérons les deux cas suivants :

**Cas 1 :**  $A \geq \frac{x_{21}}{\tau_2} (A+B)$  et  $B \leq \frac{x_{22}}{\tau_2} (A+B)$

Soit  $S_1$  ( $S_2$ ) la solution optimale utilisant seulement  $AT_1$  ( $AT_2$ ) et soit  $S$  la solution optimale utilisant  $AT_1$  et  $AT_2$ . Il est facile de vérifier

$$C_{\max}(S_1) = \frac{A}{x_{11}},$$

et

$$C_{\max}(S_2) = \frac{A}{x_{21}}.$$

Considérons maintenant la solution  $S$ . Soient  $t_1 > 0$  et  $t_2 > 0$  les durées d'utilisation des affectations  $AT_1$  et  $AT_2$ , respectivement. Nous aurons

$$A \leq x_{11}t_1 + x_{21}t_2,$$

$$B \leq x_{12}t_1 + x_{22}t_2.$$

Comme  $\frac{x_{21}}{\tau_2} > \frac{x_{11}}{\tau_1}$ , on déduit que  $x_{21} > x_{11}$  (autrement  $AT_1$  va dominer  $AT_2$ ).

D'où

$$C_{\max}(S_2) < C_{\max}(S_1).$$

Nous aurons aussi

$$x_{21}t_1 + x_{21}t_2 > x_{11}t_1 + x_{21}t_2 \geq A.$$

D'où

$$C_{\max}(S) = t_1 + t_2 > \frac{A}{x_{21}} = C_{\max}(S_2).$$

Il s'ensuit alors que la solution utilisant seulement l'affectation  $AT_2$  possède un meilleur makespan.

**Cas 2 :**  $A \leq \frac{x_{11}}{\tau_1} (A+B)$  et  $B \geq \frac{x_{12}}{\tau_1} (A+B)$ .

En procédant de la même manière que dans le Cas 1, on montre que la solution utilisant seulement  $AT_1$  possède le meilleur makespan. D'où le résultat du théorème.

Soit maintenant  $(Y_1, Y_2)$  une partition de l'ensemble  $J$  des tâches et  $I$  le point du plan représentant cette partition. Le point  $I$  a pour coordonnées les valeurs suivantes :

$$\left( \frac{\sum_{i \in Y_1} p_i}{\sum_{i \in J} p_i}, \frac{\sum_{i \in Y_2} p_i}{\sum_{i \in J} p_i} \right).$$

Appelons  $P$  le point d'intersection de la droite passant par l'origine du plan et le point  $I$  avec la courbe des affectations (voir Figure 3). Appelons  $AT_1 = (x_{11}, x_{12})$  et  $AT_2 = (x_{21}, x_{22})$  les affectations se situant à gauche et à droite de ce point.

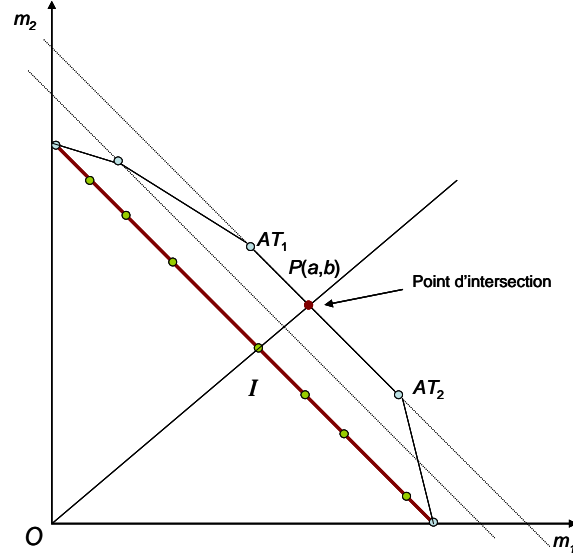


Figure 3. Le point d'intersection  $P$

Les coordonnées du point  $P$  sont

$$a = \frac{(x_{11} \cdot x_{22} - x_{12} \cdot x_{21}) \sum_{i \in Y_1} p_i}{(x_{11} - x_{21}) \cdot \sum_{i \in Y_2} p_i - (x_{12} - x_{22}) \sum_{i \in Y_1} p_i},$$

$$b = \frac{(x_{11} \cdot x_{22} - x_{12} \cdot x_{21}) \sum_{i \in Y_2} p_i}{(x_{11} - x_{21}) \cdot \sum_{i \in Y_2} p_i - (x_{12} - x_{22}) \sum_{i \in Y_1} p_i}.$$

**Théorème 4 :** La somme des coordonnées du point  $P$  est inversement proportionnelle à la valeur du makespan de la solution utilisant les deux affectations situées de part et d'autre du point  $P$ .

**Preuve :** Soient  $t_1$  et  $t_2$  les temps d'utilisation des affectations  $AT_1$  et  $AT_2$ , respectivement. Les temps  $t_1$  et  $t_2$  sont les solutions du système d'équations

$$x_{11}t_1 + x_{21}t_2 = \sum_{i \in Y_1} p_i,$$

$$x_{12}t_1 + x_{22}t_2 = \sum_{i \in Y_2} p_i.$$

Nous aurons alors

$$t_1 = \frac{x_{22} \sum_{i \in Y_1} p_i - x_{21} \sum_{i \in Y_2} p_i}{x_{11}x_{22} - x_{21}x_{12}},$$

$$t_2 = \frac{x_{11} \sum_{i \in Y_2} p_i - x_{12} \sum_{i \in Y_1} p_i}{x_{11}x_{22} - x_{21}x_{12}}.$$

D'où,

$$C_{\max} = t_1 + t_2 = \frac{(x_{22} - x_{12}) \sum_{i \in Y_1} p_i + (x_{11} - x_{21}) \sum_{i \in Y_2} p_i}{x_{11}x_{22} - x_{21}x_{12}}.$$

D'autre part, la somme des coordonnées du point  $P(a,b)$  est donnée par

$$\begin{aligned} a + b &= \frac{(x_{11}x_{22} - x_{12}x_{21}) \sum_{i \in J} p_i}{(x_{11} - x_{21}) \sum_{i \in Y_2} p_i - (x_{12} - x_{22}) \sum_{i \in Y_1} p_i} \\ &= \frac{\sum_{i=1}^n p_i}{C_{\max}}. \end{aligned}$$

D'où le résultat.

On déduit de ce résultat que pour deux affectations données, la partition optimale est celle qui est la plus proche de l'affectation ayant le plus grand taux de service. En effet, plus la partition est proche de l'affectation ayant le plus grand taux de service, plus la somme des coordonnées du point qui la représente est plus grande.

**Théorème 5 :** Il existe une solution optimale utilisant au plus deux affectations consécutives.

**Preuve :** Soit  $S$  une solution optimale et supposons qu'elle utilise  $k, k > 2$ , affectations. Soient  $AT_1, AT_2, \dots, AT_k$  ces affectations et  $t_1, t_2, \dots, t_k$  leurs durées respectives.

Nous aurons

$$A = t_1x_{1,1} + t_2x_{1,2} + \dots + t_kx_{1,k}, \quad (3)$$

$$B = t_1x_{2,1} + t_2x_{2,2} + \dots + t_kx_{2,k}. \quad (4)$$

Cette solution a pour makespan

$$C_{\max}(S) = \sum_{i=1}^k t_i.$$

De (3) et (4), on déduit

$$\frac{t_1}{C_{\max}(S)}x_{1,1} + \frac{t_2}{C_{\max}(S)}x_{1,2} + \dots + \frac{t_k}{C_{\max}(S)}x_{1,k} = \frac{A}{C_{\max}(S)},$$

$$\frac{t_1}{C_{\max}(S)}x_{2,1} + \frac{t_2}{C_{\max}(S)}x_{2,2} + \dots + \frac{t_k}{C_{\max}(S)}x_{2,k} = \frac{B}{C_{\max}(S)}.$$

Posons

$$a_1 = \frac{t_1}{C_{\max}(S)}x_{1,1} + \frac{t_2}{C_{\max}(S)}x_{1,2} + \dots + \frac{t_k}{C_{\max}(S)}x_{1,k},$$

$$b_1 = \frac{t_1}{C_{\max}(S)}x_{2,1} + \frac{t_2}{C_{\max}(S)}x_{2,2} + \dots + \frac{t_k}{C_{\max}(S)}x_{2,k},$$

et appelons  $P_1$  le point de coordonnées  $(a_1, b_1)$ .

Nous avons

$$\begin{aligned} a_1 + b_1 &= \frac{A}{C_{\max}(S)} + \frac{B}{C_{\max}(S)}, \\ &= \frac{\sum_{i=1}^n p_i}{C_{\max}(S)}. \end{aligned}$$

Supposons que la droite passant par l'origine du plan et le point,  $I$ , représentant la partition  $(Y_1, Y_2)$ , coupe le segment joignant les points représentant les affectations  $AT_h$  et  $AT_{h+1}$  en  $P(a,b)$ . Soit  $S'$  la solution optimale utilisant seulement  $AT_h$  et  $AT_{h+1}$ , alors, d'après le Théorème 4, on aura

$$a + b = \frac{\sum_{i=1}^n p_i}{C_{\max}(S')}.$$

Si les affectations ne sont pas toutes de même taux de service, alors la somme des coordonnées du point  $P_1(a_1, b_1)$  est inférieure à celle des coordonnées du point  $P(a,b)$ . Il en résulte alors

$$C_{\max}(S') < C_{\max}(S).$$

Si les affectations ont toutes le même taux de service, alors il est clair que l'utilisation de deux affectations situées de part et d'autre de la partition  $(Y_1, Y_2)$  va générer une solution optimale. D'où le résultat.

**Corollaire 1 :** Soient  $(Y_1, Y_2)$  une partition de l'ensemble des tâches et  $P$  le point d'intersection de la droite passant par l'origine du plan et le point représentant cette partition, avec la courbe des affectations. Si la somme des coordonnées de  $P$  est maximale par rapport à toutes les partitions possibles, alors la partition  $(Y_1, Y_2)$  est optimale. Les affectations à utiliser sont celles se trouvant de parts et d'autres du point  $P$ .

**Preuve :** Ce résultat découle directement des Théorèmes 4 et 5.

**Corollaire 2 :** Si  $(Y_1, Y_2)$  est une partition de l'ensemble des tâches, pour laquelle la distance du point représentant cette partition au segment des deux affectations est minimale, alors elle est optimale. La solution utilisera ces deux affectations.

**Preuve :** Ce résultat découle du résultat précédent. En effet, la distance est minimale si, et seulement si, la somme des coordonnées du point  $P$  est maximale.

#### 4.2 Résolution du problème

Dans cette section, nous présentons un algorithme qui permet de construire une solution optimale. Cet algorithme découle des propriétés précédentes.

##### Début de l'algorithme

1. Soient  $p_i, 1 \leq i \leq n$ , les temps opératoires de base de  $n$  tâches données.
2. Soient  $AT_1, AT_2, \dots, AT_l$  l'ensemble des affectations telles que  $x_{i1} < x_{j1}$  pour tout  $1 \leq i < j \leq l$ ;

3. Soient  $\tau_1, \tau_2, \dots, \tau_l$  les taux de service respectifs de l'ensemble des affectations ;
  4.  $Makespan = \sum_{i \in J} p_i$  ;
  5. Pour  $k = 1$  à  $l-1$  faire {
    - a. Si  $\tau_k > \tau_{k+1}$  alors résoudre *PL1*  
Sinon résoudre *PL2* ;
    - b. Soit  $(Y_1, Y_2)$  la partition de l'ensemble  $J$  des  $n$  tâches produite par les programmes *PL1* ou *PL2* ;
    - c. Si  $\min_{j=1,2} \left( \frac{x_{j1}}{\tau_j} \right) \leq \frac{\sum_{i \in Y_1} p_i}{\sum_{i \in J} p_i} \leq \max_{j=1,2} \left( \frac{x_{j1}}{\tau_j} \right)$  alors {
      - Posons  $A = \sum_{i=1}^n \delta_i p_i$  et  $B = \sum_{i=1}^n p_i - \sum_{i=1}^n \delta_i p_i$  ;
      - $C_{\max} = \frac{(A+B)(x_{k+1,1} - x_{k,1}) + A(\tau_k - \tau_{k+1})}{\tau_k x_{k+1,1} - \tau_{k+1} x_{k,1}}$  ;
      - Si  $Makespan > C_{\max}$  alors  
 $Makespan = C_{\max}$  ;
- } // fin du test  
} // fin de la boucle pour

### Fin de l'algorithme

Notons que *PL1* et *PL2* sont des programmes linéaires en nombres entiers. Chacun d'eux cherche la partition la plus proche de l'affectation ayant le plus grand taux de service. Avant de procéder, définissons d'abord les paramètres suivants :

$$\delta_i = \begin{cases} 1 & \text{si } i \text{ appartient à } Y_1, \\ 0 & \text{sinon.} \end{cases}$$

$$K_1 = \min_{j=1,2} \left( \frac{x_{j1}}{c_j} \right) \sum_{i=1}^n p_i,$$

et

$$K_2 = \max_{j=1,2} \left( \frac{x_{j1}}{c_j} \right) \sum_{i=1}^n p_i.$$

Les deux programmes linéaires sont donc comme suit :

$$\text{Minimiser } \left( \sum_{i=1}^n \delta_i p_i \right)$$

Sous les contraintes

$$K_1 \leq \sum_{i=1}^n \delta_i p_i \quad (PL1)$$

$$\delta_i \in [0,1]$$

$$\text{Maximiser } \left( \sum_{i=1}^n \delta_i p_i \right)$$

Sous les contraintes

$$K_2 \geq \sum_{i=1}^n \delta_i p_i \quad (PL2)$$

$$\delta_i \in [0,1]$$

Dans les deux cas, il faudrait trouver les  $\delta_i$  qui optimisent la fonction objectif.

### Justification de l'algorithme

L'algorithme considère les affectations consécutives deux à deux. Pour chaque couple d'affectations, il cherche la partition optimale dans le cône des deux affectations. Ceci est justifié par le Théorème 5. Soient  $AT_1$  et  $AT_2$  deux affectations successives, et supposons qu'il existe une partition  $(Y_1, Y_2)$  de l'ensemble  $J$  des tâches vérifiant

$$\min_{j=1,2} \left( \frac{x_{j1}}{\tau_j} \right) \sum_{i \in J} p_i \leq \sum_{i \in Y_1} p_i \leq \max_{j=1,2} \left( \frac{x_{j1}}{\tau_j} \right) \sum_{i \in J} p_i.$$

Du Théorème 4, le makespan de la solution optimale  $S$  est

$$C_{\max}(S) = \frac{(x_{22} - x_{12}) \sum_{i \in Y_1} p_i + (x_{11} - x_{21}) \sum_{i \in Y_2} p_i}{x_{11} x_{22} - x_{21} x_{12}}.$$

Cette expression peut s'écrire aussi comme suit :

$$C_{\max}(S) = \frac{(A+B)(x_{21} - x_{11}) + A(\tau_1 - \tau_2)}{\tau_1 x_{21} - \tau_2 x_{11}}.$$

Comme on peut le constater,  $C_{\max}(S)$  dépend seulement de la valeur de  $A$ , car le dénominateur ainsi que les expressions  $\tau_1 - \tau_2$  et  $(A+B)(x_{21} - x_{11})$  sont des constantes. Considérons maintenant les deux cas suivants :

#### Cas 1 : $\tau_1 \geq \tau_2$

Nous aurons

$$\frac{x_{11}}{\tau_1} < \frac{x_{21}}{\tau_2}.$$

D'où

$$\tau_1 x_{21} - \tau_2 x_{11} > 0.$$

Plus  $A$  est petit, plus petit est le makespan. D'autre part, plus  $A$  est petit, plus petit est  $\tau_1 A - x_{11}(A+B)$  et plus grand est  $(A+B)x_{21} - A\tau_2$ . Par conséquent, la partition qui minimise le makespan est celle pour laquelle  $\tau_1 A - x_{11}(A+B)$  est minimum, c'est-à-dire

$$\sum_{i \in Y_1} p_i - \min_{j=1,2} \left( \frac{x_{j1}}{\tau_j} \right) \sum_{i \in J} p_i \text{ est minimum.}$$

La partition optimale est donc donnée par *PL1*.

#### Cas 2 : $\tau_1 < \tau_2$

Nous aurons

$$\frac{x_{11}}{\tau_1} < \frac{x_{21}}{\tau_2}.$$

Sinon,  $AT_1$  va être dominée par  $AT_2$ . Par conséquent,

$$\tau_1 x_{21} - \tau_2 x_{11} > 0.$$

Plus  $A$  est grand, plus petit est le makespan. D'autre part, plus  $A$  est grand, plus grand est  $\tau_1 A - x_{11}(A+B)$  et plus petit est  $(A+B)x_{21} - A\tau_2$ . D'où la partition qui minimise le makespan est celle pour laquelle  $(A+B)x_{21} - A\tau_2$  est minimum, c'est-à-

dire  $\max_{j=1,2} \left( \frac{x_{j1}}{\tau_j} \right) \sum_{i \in J} p_i - \sum_{i \in Y_1} p_i$  est minimum.

La partition optimale est alors donnée par  $PL2$ .

La preuve de l'exactitude de l'algorithme est donc établie.

Avant de clore cette section, notons que, dans cet algorithme, on résout  $(l-1)$  fois les programmes linéaires  $PL1$  et/ou  $PL2$ . Comme le nombre  $l$  d'affectations est restreint, il est clair que la complexité de l'algorithme est dominée par la résolution de  $PL1$  et  $PL2$  qui ne sont rien d'autre que des problèmes de sac à dos 0-1. D'autre part, il a été reporté dans [Martello et Toth, 1990] une étude expérimentale détaillée sur plusieurs algorithmes résolvant le problème du sac à dos 0-1. Cette étude montre que l'algorithme de Balas et Zemel [1980] surclasse les autres algorithmes auxquels il est comparé et il permet surtout de générer une solution optimale en quelques secondes pour de très larges instances du problème.

## 5 CONCLUSION

Dans cet article, nous avons étudié un problème d'ordonnement de tâches sur deux machines parallèles identiques supervisées par un seul opérateur. Dans notre modèle, les durées d'exécution des tâches sont variables car dépendant de l'affectation de l'opérateur aux machines. Nous avons supposé que la détérioration des durées d'exécution est indépendante des tâches effectuées et que l'affectation de l'opérateur induit des taux de productivités variables pour les machines. Nous avons proposé une approche géométrique pour décrire le problème et avons présenté quelques propriétés, découlant de cette représentation, pour caractériser l'optimalité d'une solution. Ensuite, nous avons utilisé ces propriétés pour développer un algorithme exact pour construire une solution optimale. Cet algorithme peut résoudre de très grandes instances du problème. Comme travail futur, il serait intéressant d'étendre cette étude au cas de plus de deux machines et plusieurs opérateurs.

## 6 REMERCIEMENTS

Cette étude est financée par le Conseil de Recherche en Sciences Naturelles et en Génie du Canada (CRSNG) et par le Département de Mathématiques et Génie Industriel de l'École Polytechnique de Montréal.

## 7 RÉFÉRENCES

Balas, E., Zemel, E. (1980) An Algorithm for Large Zero-One Knapsack Problems. *Operations Research*, 28(5), pp. 1130-1154.

Baptiste, P., Hait, A., Soumis, F. (2005) Gestion de production et ressources humaines, Presses Internationales Polytechnique, Montréal, Canada.

Bourland, K.E., Carl, L.K. (1994) Parallel machine scheduling with fractional operator requirements. *IIE Transactions (Institute of Industrial Engineering)*, 26(5), pp. 56-65.

Cheurfa, M. (2005) Gestion des ressources humaines en production cyclique. Thèse de doctorat, École des Mines de Saint Etienne, France.

Daniels, R.L., Hoopes, B.J., Mazzola, J.B. (1996) Scheduling parallel manufacturing cells with resource flexibility. *Management Science*, 42(9) pp. 1260-1276.

Grigoriev, A., Sviridenko, M., Uetz, M. (2005) Unrelated parallel machine scheduling with resource dependent processing times, *11th International IPCO Conference on Integer Programming and Combinatorial Optimization*, Berlin, Germany, 8-10 June.

Hu, P.-C. (2005) Minimizing total flow time for the worker assignment scheduling problem in identical parallel machine models, *International Journal of Advanced Manufacturing Technology*, 25(9-10), pp. 1046-1052.

Hu, P.-C. (2004) Minimising total tardiness for the worker assignment scheduling problem in identical parallel machine models, *International Journal of Advanced Manufacturing Technology*, 23(5-6), pp. 383-388.

Huq, F., Cutright, K., Martin, C. (2004) Employee scheduling and makespan minimization in a flow shop with multi-processor work stations: a case study, *Omega*, 32(2), pp. 121-129.

Karlof, J.K., Wang, W. (1996) Bilevel programming applied to the flow shop scheduling problem, *Computers & Operations Research*, 23(5), pp. 443-451.

Kovalyov, M.Y., Cheng, T.C.E. (2000) Scheduling jobs by One Operator on Two Sequential Machines, *Seventh International Workshop on Project Management and Scheduling*, Osnabrück, Germany.

Martello, S., Toth, P. (1990) Knapsack Problems: Algorithms and Computer Implementations, John Wiley and Sons.

Pinedo, M. (2002) : Scheduling: Theory, Algorithms, and Systems, 2<sup>nd</sup> edition, Prentice-Hall.

Shabtay, D., Steiner, G. (2007) A survey of scheduling with controllable processing times. *Discrete Applied Mathematics*, 155(13), pp. 1643-1666.

Tchommo, J.L. (2004) Ordonnement simultané de production et des ressources humaines. Mémoire de maîtrise, École Polytechnique de Montréal, Canada.

Vickson, R.G. (1980a) Choosing the job sequence and processing times to minimize total processing plus flow cost on a single machine, *Operations Research*, 28(5), pp. 1155-1167.

Vickson, R.G. (1980b) Two single machine sequencing problems involving controllable job processing times, *AIIE transactions*, 12(3), pp. 258-262.

Zouba, M., Baptiste, P., Rebaine, D. (2009a) Ordonnement de machines parallèles identiques en présence d'opérateurs avec des changements en fin de tâche, *8<sup>ème</sup> Congrès International de Génie Industriel - CIGI'09*, Tarbes, France, 10-12 Juin.

Zouba, M., Baptiste, P., Rebaine, D. (2009b) Scheduling identical parallel machines and operators within a period based changing mode. *Computers & Operations Research*, 36(12), pp. 3231-3239.

Zouba, M., Baptiste, P., Rebaine, D. (2008) Ordonnement de machines parallèles identiques en présence d'opérateurs avec des changements calendaires, *7<sup>ème</sup> Conférence Francophone de Modélisation et simulation - MOSIM'08*, Paris, France, 31 Mars - 2 Avril.

Zouba, M., Baptiste, P., Rebaine, D. (2006) Ordonnement de jobs sur deux machines parallèles en présence d'un seul opérateur, *6<sup>ème</sup> Conférence Francophone de Modélisation et Simulation - MOSIM'06*, Rabat, Maroc, 3-5 Avril.