Ordonnancement d'ateliers de type job shop avec opérateurs

Imène Benkalai[1], Djamal Rebaine[1], Pierre Baptiste[2]

[1] Université du Québec à Chicoutimi, Saguenay QC ,G7H 2B1 Canada
{imene.benkalai1,djamal.rebaine}@uqac.ca
[2] École Polytechnique de Montréal, Montréal QC, H3T 1J4 Canada
{pbaptiste}@polymtl.ca

Mots-clés : *Ordonnancement, job shop, opérateurs, mode de changement d'affectation libre, retard algébrique maximum.*

1 Introduction

La plus grande partie de travaux dédiés à l'ordonnancement a porté sur des problèmes qui ignorent les contraintes des ressources humaines. Seulement, avec l'augmentation de l'importance de ces dernières, notamment dans les environnements de production, l'ordonnancement "classique" fournissait des modèles qui n'étaient que peu réalistes. En effet, les aspects matériel et humain ne sont en général pas indépendants dans un système de production, notamment au niveau opérationnel [2]. Il est ainsi important de considérer les ressources humaines ainsi que leurs caratéristiques lors de la gestion de ressources matérielles.

Dans le présent travail, nous étudions l'ordonnancement d'ateliers de type job shop en présence d'opérateurs dont l'affectation change selon un mode libre, où l'ordre de tâches est connu au départ et l'objectif est la minimisation du retard algébrique maximum. Les opérateurs ont des niveaux différents de performance. À notre connaissance, ce problème n'a pas encore été étudié.

2 Description du problème

Dans un problème de job shop, n tâches doivent être traitées par m machines. Chaque tâche j, $j = 1, \ldots, n$ doit être traitée par chacune des machines i, $i = 1, \ldots, m$ et ce dans un ordre fixe qui lui est propre.

Dans notre modèle, nous supposons qu'une opération nécessite la présence de l'un des k opérateurs ($k < m$) pour toute la durée de son traitement. L'impact de la présence d'un nombre d'opérateurs inférieur au nombre de machines y est ainsi modélisé par des temps d'attente durant lesquels une machine restera inactive même si elle a des opérations en attente. Les paramètres p_{ij}, d_{ij} et C_{ij} représentent respectivement le temps de traitement, la date d'échéance et la date de fin de traitement de la tâche j sur la $i^{\text{ème}}$ machine.

Nous avons ℓ types d'opérateurs, k_h, $h = 1, \ldots, \ell$, opérateurs de chaque type avec $\sum_{h=1}^{\ell} k_h = k$. Un opérateur de type h a une cadence v_h. Selon le mode libre, le changement d'affectation des opérateurs peut se produire à tout instant. Un opérateur pourra alors interrompre le traitement d'une tâche pour s'occuper d'une autre tâche. Cette tâche pourra par la suite être reprise par ce même opérateur ou être confiée à un autre opérateur. L'objectif est de minimiser le retard algébrique maximum, $L_{\max} = \max_{j=1,\ldots,n} L_{mj}$ où $L_{ij} = C_{ij} - d_{ij}$.

3 Complexité et méthodes de résolution

Nous notons notre problème $J_m|res\ \ell k_h 1, S, v_h, f|L_{\max}$ conformément à la notation proposée dans [4] à laquelle nous ajoutons les symboles S pour indiquer que l'affectation se fait sous un

ordre de tâches donné S, v_h qui indique que les opérateurs ont des cadences différente et f qui indique que le mode de changement d'affectation est libre.

3.1 Cas à k opérateurs, $k \geq 3$

Théorème 1. Le problème $J_m|res\ \ell k_h 1, S, v_h, f|L_{\max}$ est NP-difficile au sens fort pour $k \geq 3$.

Démonstration. Nous utilisions une réduction du problème de flow shop avec différents types d'opérateurs et un changement d'affectation libre $F_m|res\ \ell k_h 1, S, v_h, f|L_{\max}$. La réduction se fait naturellement, le flow shop étant un cas particulier du job shop.

Le problème $F_m|res\ \ell k_h 1, S, v_h, f|L_{\max}$ est NP-difficile au sens fort [3]. □

3.2 Cas à $k = 2$ opérateurs

Théorème 2. Le problème $J_m|res\ 2k_h 1, S, v_h, f|L_{\max}$ est résoluble en temps polynomial.

Démonstration. La preuve se fera via la méthode de restriction. Les contraintes de passage des opérations sur les machines et celles générées par S, forment un graphe de précédence. Ainsi, notre problème peut être vu comme un problème de deux machines parallèles uniformes avec un graphe de précédence et la possibilité de préemption, noté par $Q_2|prec, prmp|L_{\max}$, où les machines, les tâches et le graphe de précédence représentent respectivement les 2 opérateurs, les $m * n$ opérations, et les relations de précédence entre elles. La possibilité de préemption modélise le mode de changement d'affectation libre. Ce problème est résoluble en $O(n^2)$ [5]. □

Passons à la méthode de résolution. Les algorithmes suivants sont des adaptations de ceux décrits par Lawler dans [5].

La solution est construite en trois étapes. Il s'agit de modifier d'abord les dates d'échéance (Figure 1) puis de construire des intervalles d'ordonnancement variables (Figure 2). Enfin, une solution est obtenue avec un algorithme d'ordonnancement par priorité (Figure 3) en $O(m^2n^2)$.

Soient les données suivantes :

- $d_{ij} = d_j - \sum_{q=i+1}^{m} p_{qj}$ la date d'échéance de la $i^{\text{ème}}$ opération de la tâche j.
- Sans perdre de généralité, $v_1 = 1$, $v_2 = \frac{min\{v_1,v_2\}}{max\{v_1,v_2\}}$.
- $p_{ij}(t)$ le temps de traitement minimum qui doit être effectué sur l'opération (i, j) avant la date t pour que ladite opération respecte sa date d'échéance.
- $S(i, j)$ l'ensemble des successeurs (immédiats ou pas) de l'opération (i, j).

Le paramètre $b_{ij}^{(r)} = d_{ij} - p_{ij}^{(r)}$, où $p_{ij}^{(r)}$ est le temps de traitement restant à effectuer sur l'opération (i, j) à l'instant t_r ; il sert d'indicateur de priorité. Plus $b_{ij}^{(r)}$ est petit, plus la priorité de l'opération (i, j) est grande. Les z opérations disponibles à un instant t_r seront ré-indexées selon l'ordre croissant des $b_{ij}^{(r)}$.

Soit $x_{ij}^{(r)}$ la quantité de l'opération (i, j) exécutée durant l'intervalle $[t_r, t_{r+1}]$.

$$x_{ij}^{(r)} = \begin{cases} \Delta, & q < u, \\ max\{0, T - b_q^{(r)}\}, & q \geq u, \end{cases}$$

avec T, u, v et Δ déterminés par l'algorithme en Figure 2, v étant l'indice tel que $b_v^{(r)} \leq T < b_{v+1}^{(r)}$ et q l'indice correspondant à l'opération (i, j) après ré-indexation.

3.3 Cas à $k = 1$ opérateur

Théorème 3. Le problème $J_m|res\ 111, S, f|L_{\max}$ est résoluble en temps polynomial.

Démonstration. Nous utilisons le même principe de preuve que précédemment. Nous utilisons pour la réduction $1|prec, prmp, r_j|L_{\max}$ qui est résoluble en $O(n^2)$ [1]. □

Pour (toute opération (i,j) sans successeur)
$\{d'_{ij} \leftarrow d_{ij}\,;\}$
Créer deux listes ordonnées par date d'échéance modifiées d'_{ij} et $b'_{ij} = d'_{ij} - p_{ij}$;
Tant que ($\exists$ (i,j) / d_{ij} non modifiée et $d_{i'j'}$ modifiée $\forall (i',j') \in S(i,j)$)
{ Choisir (i,j) ;
Chercher les successeurs de cette opération dans les deux listes ordonnées ;
Calculer $\sum_{(i',j')\in S(i,j)} p_{i'j'}(d'_{i''j''})\ \forall (i'',j'') \in S(i,j)$;

$$d'_{ij} \leftarrow min\left\{ d_{ij}, \min_{d'_{(i'',j'')}/(i'',j'')\in S(i,j)} \left\{ d'_{(i'',j'')} - \frac{\sum_{(i',j')\in S(i,j)} p_{i'j'}(d'_{i''j''})}{v_1+v_2} \right\} \right\} ;$$

Pour (chaque $(i',j') \in S(i,j)$)
$\{d'_{ij} \leftarrow min\{d'_{ij}, b'_{i'j'} = d'_{i'j'} - p_{i'j'}\}\,;\}$
Insérer d'_{ij} et $b'_{ij} = d'_{ij} - p_{ij}$ dans les listes ordonnées ;
}

FIG. 1 – Algorithme pour la modification des dates d'échéance [3].

$\Delta \leftarrow 0\,;\ u \leftarrow 2\,;\ v \leftarrow 2\,;\ T \leftarrow b_2\,;\ P \leftarrow 0\,;$
Tant que $(\Delta < t_{r+1} - t_r)$
{
Tant que $(T = b_{u-1} + \Delta)$
{ $u \leftarrow u-1$ (Si $u = 1$, $b_0 = -\infty$) ; $P \leftarrow P + \Delta\,;\}$
Tant que $(T = b_{v+1})$
{ $v \leftarrow v+1$ (Si $v = z$, $b_{z+1} = \infty$)$\,;\}$
$T_1 \leftarrow (\frac{(b_{u-1}+\Delta)(2-u+v_2)-(v-u+1)T)}{((2-u+v_2)-(v-u+1))}$; $T_2 \leftarrow b_{v+1}$; $T_3 \leftarrow T + \frac{(t_{r+1}-t_r-\Delta)(2-u+v_2)}{(v-u+1)}$;
$T' \leftarrow min\{T_1, T_2, T_3\}$; $P \leftarrow P + (v-u+1)(T'-T)$; $\Delta \leftarrow \frac{P}{(2-u+v_2)}$; $T \leftarrow T'$;
}

FIG. 2 – Algorithme pour la construction d'intervalles fixes.

L'algorithme présenté en Figure 4 est une adaptation de l'algorithme de Baker *et al.* [1] en $O(m^2n^2)$ pour le problème $1|prec, prmp, r_j|L_{max}$. Dans notre cas, toutes les dates de disponibilités (r_j) sont égales à 0 et ainsi toutes les opérations appartiennent à un même block B et peuvent être traitées sans temps d'arrêt. Soit $P(B) = \sum_{(i,j)\in B} p_{ij}$.

4 Conclusion et perspectives

Dans l'environnement économique actuel, fortement concurrentiel, il devient plus important de dévolopper des outils d'aide à la décision plus performants, et ce en se basant modèles plus réalistes notamment avec la considération des caractéristiques des ressources humaines. Cela permettrait aussi de réduire l'écart entre la théorie et la pratique.

Dans le présent travail, nous avons étudié le problème d'affectation d'opérateurs ayant différents niveaux de performances dans un environnement de type job shop. Le nombre d'opérateurs est inférieur à celui des machines, l'ordre des tâches est donné, le mode de changement d'affectation est libre et l'objectif est la minimisation du retard algébrique maximum. Ce problème est fortement NP-difficile pour $k \geq 3$. Des algorithmes de complexité polynomiale ont été fournis pour les cas à un et deux opérateurs. Il serait intéressant de développer des heuristiques pour les cas à trois opérateurs et plus, mais aussi de considérer d'autres objectifs.

Calculer le degré intérieur de chaque opération (i, j) ;
Créer une file Q contenant les opérations de degré intérieur égal à 0 ;
Créer une liste A contenant les opérations disponibles pour traitement* ;
$r \leftarrow 0$;
Tant que $(Q \neq \varnothing)$
{ $r \leftarrow r + 1$; $t_r \leftarrow 0$; $A \leftarrow A \cup Q$; $Q \leftarrow \varnothing$;
Tant que $(A \neq \varnothing)$
{ $t_{r+1} \leftarrow 0$;
Exécuter l'algorithme en Figure 2 sur les opérations dans A ;
Si (les valeurs trouvées (Δ, u, v, T) violent $b_j + \Delta \leq d_j$, $j = 1, \ldots, u-1$)
{ Ré-exécuter l'algorithme en Figure 2 avec $t_{r+1} = t_r + \min_y \{p_j^y / j = 1, \ldots, u-1\}$;
$A \leftarrow A \backslash \{$opérations complétées dans$[t_r, t_{r+1}]\}$;
Mettre à jour les degrés intérieurs des opérations ;
Mettre à jour Q ; $A \leftarrow A \cup Q$; $k \leftarrow k + 1$;
}
}
}

N.B : Pour la construction d'intervalles variables, on procède de manière analogue à la différence des points suivants :
— $T \leq d_j$, $j = u, u+1, \ldots, v$. Si $T = d_j$, $j \geq u$, le calcul s'arrête.
— Si u est décrémenté, on vérifie si $b_{u-1} + \Delta > d_{u-1}$, si c'est le cas, le calcul s'arrête.
* Une opération est dite disponible à la date t si le traitement de tous ses prédécesseurs est complété et s'il reste un temps de traitment non nul à effectuer sur cette opération.

FIG. 3 – Algorithme pour la résolution de $J_m|res\ 2k_h 1, S, v_h, f|L_{\max}$ [3].

Tant que $(B \neq \varnothing)$
{ Choisir (i, j) / $S(i, j) = \varnothing$ et $L_{ij} = \min_{(i,j) \in B} \{P(B) - d_{ij}\}$;
$B \leftarrow B \backslash \{(i, j)\}$;
}

FIG. 4 – Algorithme pour la résolution de $J_m|res\ 111, S, f|L_{\max}$ [3].

Références

[1] K. R. Baker, E. L. Lawler, J. K. Lenstra, and A. H. G. Rinnooy Kan. Preemptive scheduling of a single machine to minimize maximum cost subject to release dates and precedence constraints. *Operations Research*, 31(2) :381–386, 1983.

[2] P. Baptiste, V. Giard, A. Hait, and F. Soumis. *Gestion de production et ressources humaines*. Presses Internationales Polytechnique, 2005.

[3] I. Benkalai, P. Baptiste, and D. Rebaine. Ordonnancement d'ateliers de type flow shop avec opérateurs en mode d'affectation libre. In *17 ème conférence ROADEF*, 2016.

[4] J. Blazewicz, K. Ecker, E. Pesch, G. Schmidt, and J. Weglarz. *Handbook on scheduling : From theory to applications*. Springer, 2007.

[5] E. L. Lawler. Preemptive scheduling of precedence-constrained jobs on parellel machines. In *Deterministic and stochastic scheduling*, pages 101–123. D. Reidel publishing Co., 1982.