Minimisation des encours sur une machine pour des tâches sujettes à des phénomènes d'usure ou d'apprentissage

Julien Moncel[1]

Laboratoire G-SCOP ; Grenoble INP ; 46 Av. Félix Viallet, F-38031 Grenoble Cédex 1, France
julien.moncel@g-scop.inpg.fr

Mots-Clés : *ordonnancement avec phénomènes d'usure ou d'apprentissage, ordonnancement avec contraintes de précédence, largeur d'arborescence.*

1 Problèmes d'ordonnancement avec phénomènes d'usure ou d'apprentissage

Nous nous intéressons ici à des problèmes de minimisation des encours sur une machine où le temps de process d'une tâche i dépend de la position r dans laquelle elle est exécutée dans l'ordonnancement. Chaque tâche i possède un temps de process de base p_i, mais le temps de process réel de i dépendra de l'ordre dans lequel les tâches seront effectuées. Formellement, nous considérons des problèmes du type $1/p_i^{[r]} = f(r)p_i/\sum C_i$, où le temps de process réel de la tâche i lorsqu'elle est exécutée en r-ème position dans l'ordonnancement, noté $p_i^{[r]}$, est calculé comme $f(r) \times p_i$, avec f fonction de $\mathbb{N}^*$ dans $\mathbb{R}^+$.

Par exemple, si l'on considère 3 tâches telles que $p_1 = 1, p_2 = p_3 = 2$, et comme fonction $f(r) = r$, alors l'ordonnancement 1 2 3 est tel que $C_1 = 1 \times p_1 = 1$, $C_2 = C_1 + 2 \times p_2 = 5$, et $C_3 = C_2 + 3 \times p_3 = 11$ (voir le cas a) dans la Figure 1). Ainsi on a $\sum C_i = 17$. Si l'on considère l'ordonnancement 2 1 3, alors il est tel que $\sum C_i = 2 + 4 + 10 = 16$ (voir le cas b) dans la Figure 1).

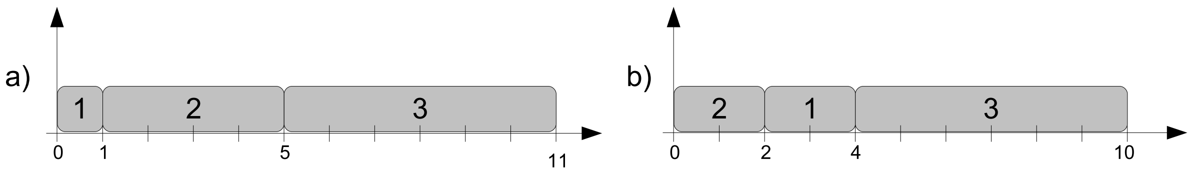

Fig. 1 – a) La durée réelle de la tâche 1 est sa durée de base car elle est réalisée en premier. La durée réelle de la tâche 2 est $p_2^{[2]} = f(2) \times p_2 = 2 \times 1$, celle de la tâche 3 est $p_3^{[3]} = f(3) \times p_3 = 3 \times 2$. b) Par rapport au cas a), la durée réelle de la tâche 2 a diminué, mais celle de la tâche 1 a augmenté. L'un dans l'autre, on y gagne : les encours ont diminué d'une unité.

La fonction f de cet exemple est telle que plus une tâche est exécutée tard, plus sa durée réelle est grande : ceci modélise un phénomène d'usure de la machine (ou de lassitude de l'opérateur). Cette fonction est étudiée par Mosheiov [6], qui montre que $1/p_i^{[r]} = rp_i/\sum C_i$ est polynomial. Si l'on prend une fonction f telle que $f(r) = \gamma^{r-1}$, avec $\gamma \in]0,1[$, alors on obtient une fonction telle que plus une tâche est exécutée tard, plus sa durée réelle est petite : ceci modélise un phénomène d'apprentissage de l'opérateur qui exécute les tâches. Cette fonction est étudiée par Gordon *et al* [5], qui montrent que le problème est polynomial. D'autres types de fonctions et de modèles pour l'apprentissage ou l'usure sont décrits dans la littérature [1, 3, 4].

2 Résultats

Nous unifions un certain nombre de résultats de la littérature concernant les problèmes d'ordonnancement avec phénomènes d'usure ou d'apprentissage, y compris les résultats de Mosheiov et de Gordon *et al* mentionnés ci-dessus. L'argument que nous utilisons est le suivant : si nous nous donnons 4 nombres $a \leq b$ et $x \leq y$, alors, des deux quantités $ax + by$ et $ay + bx$, c'est $ax + by$ la plus grande. Cet argument nous permet de plus de répondre à des questions ouvertes de la littérature.

Nous considérons de plus de tels problèmes d'ordonnancement soumis à des contraintes de précédence. En utilisant le concept de décomposition arborescente, nous unifions de même un certain nombre de résultats de la littérature. La décomposition arborescente est un concept de théorie des graphes introduit dans les années 80 par Robertson et Seymour dans une fameuse série d'articles intitulée “Graph Minors” [7]. La largeur d'arborescence (en anglais la *tree-width*) d'un graphe G est un paramètre mesurant la “distance” entre G et la classe des arbres (et des forêts). La largeur d'arborescence de tout arbre est 1, et plus cette largeur d'arborescence d'un graphe est élevée, plus le graphe est structurellement proche d'être un arbre. Cette notion permet de démontrer des théorèmes très généraux du type “si un problème $\mathcal{P}$ est exprimable dans la logique $\mathcal{L}$, alors le problème $\mathcal{P}$ est polynomial sur toute classe de graphes de largeur d'arborescence bornée” [2]. Ce type de théorèmes est intéressant lorsque l'on considère des problèmes d'ordonnancement impliquant des graphes de précédence.

Références

[1] B. Alidaee, N. .K Womer, *Scheduling with time dependent processing times : Review and extensions*, Journal of the Operational Research Society **50** (1999), 711–720.

[2] S. Arnborg, J. Lagergren, D. Seese, *Easy problems for tree-decomposable graphs*, Journal of Algorithms **12(2)** (1991), 308–340.

[3] D. Biskup, *A state-of-the-art review on scheduling with learning effects*, European Journal of Operational Research **188**(2008), 315–329.

[4] T. C. E. Cheng, Q. Ding, B. M. T. Lin, *A concise survey of scheduling with time-dependent processing times*, European Journal of Operational Research **152** (2004), 1–13.

[5] V. S. Gordon, C. N. Potts, V. A. Strusevich, J. D. Whitehead, *Single machine scheduling models with deterioration and learning : Handling precedence constraints via priority generation*, to appear in Journal of Scheduling (doi : `10.1007/s10951-008-0064-x`).

[6] G. Mosheiov, *A note on scheduling deteriorating jobs*, Mathematical and Computer Modelling **41** (2005), 883–886.

[7] N. Robertson, P. D. Seymour, *Graph minors III : Planar tree-width*, Journal of Combinatorial Theory Series B **36** (1984), 49–64.