

Ordonnancement cyclique de tâches couplées identiques

Gerd Finke, Vassilissa Lebacque, Nadia Brauner

G-SCOP UJF ; Université de Grenoble ; 46, avenue Félix Viallet, F-38031 Grenoble, France
{Gerd.Finke, Nadia.Brauner}@g-scop.grenoble-inp.fr

Mots-Clés : *Ordonnancement cyclique, high-multiplicity, tâches couplées*

Le problème des tâches couplées a été introduit par Shapiro en 1980 pour l'ordonnancement des opérations d'un radar [10]. Le radar émet un signal qui est transmis à une cible puis renvoyé au radar qui doit le réceptionner. Ainsi, le radar doit exécuter deux opérations par tâche (émission et réception) et ces opérations sont séparées par une durée fixe. Le problème d'ordonnancement des tâches couplées est alors d'exécuter des tâches sur une unique machine (le radar), chaque tâche j étant composée de deux opérations de longueurs a_j et b_j séparées par exactement L_j unités de temps. L'objectif est de minimiser le temps total pour terminer toutes les tâches dans le cas non cyclique (pour un nombre fixé de tâches couplées) ou le taux de production dans le cas cyclique (nombre de tâches infini).

Le problème des tâches couplées appartient à la classe plus large des problèmes d'ordonnancement de tâches multi-opérations où les opérations consécutives sont séparées par un certain intervalle de temps. Dans les environnements manufacturiers, le temps qui doit séparer deux opérations (délai) admet généralement un borne inférieure (voir par exemple [8]). Nous allons ici considérer uniquement des tâches couplées avec un intervalle de durée fixée séparant leurs deux opérations, comme cela s'applique dans le système des radars. Il existe une vaste littérature sur ce sujet traitant les cas offline et online et de nombreux algorithmes sont proposés.

Nous considérons ici le problème des tâches couplées offline où un ensemble de tâches couplées $\{a_j; L_j; b_j\}$ doit être ordonné sur un processeur unique avec la possibilité d'intercaler des tâches mais où des opérations ne peuvent pas se chevaucher. La complexité de la minimisation du makespan a été décrite par Orman et Potts [9]. Même le cas de tâches de durées unitaires, $a_j = 1$, $L_j, b_j = 1$ pour tout j , est NP-difficile et des algorithmes avec garantie de performances ont été développés [1, 4]. Toutefois, le status de la complexité du problème des tâches couplées identiques (où $a_j = a$, $L_j = L$, $b_j = b$ pour tout j) est ouvert à la fois dans le cas non cyclique et dans le cas cyclique (voir [2]). Il est à noter que la donnée de ce problème est composée de quatre entiers (trois dans le cas cyclique). Il s'agit donc d'un problème d'ordonnancement dit *high-multiplicity* [5, 6] pour lequel même prouver l'appartenance à NP peut s'avérer difficile. En effet, une description d'un ordonnancement (en donnant les dates de démarrage de chacune des n tâches) n'est pas polynomial dans la taille de la donnée ($\log n + \log a + \log b + \log L$ dans notre cas). Ahr *et al.* [2] proposent un algorithme linéaire en n mais exponentiel en L qui peut être adapté pour le cas cyclique [7]. Baptiste [3] a prouvé que pour a , b et L fixés, la solution optimale peut être trouvée en $O(\log n)$, ce qui améliore le temps en $O(n)$ de [2]. La constante est toujours exponentielle en L et ne conduit à aucune solution pratique.

Afin de résoudre le cas cyclique des tâches couplées identiques, nous construisons une classe particulière \mathcal{L} de solutions qui admettent une représentation polynomiale en a , b et L . Une meilleure

solution au sein de cette classe peut-être déterminée en temps polynomial. Nous montrons que la solution optimale de cette classe est également optimale pour le problème général des tâches couplées identiques qui peut donc être résolu en temps polynomial. Sans perte de généralité, on suppose que $a > b$. La classe \mathcal{L} est composée des cycles $\mathcal{C}(\alpha, \beta) = a^{\alpha+1}(ba)^\beta b^{\alpha+1}(ab)^\beta$ où α et β sont des solutions réalisables du système de contraintes suivant :

$$\begin{cases} L \geq \alpha a + \beta(a + b) \\ \alpha, \beta \geq 0 \end{cases} \quad (1)$$

Soit le cycle $\mathcal{C}^N = \mathcal{C}(\alpha^N, \beta^N, \gamma^N)$ dans \mathcal{L} définit comme suit

- (a) $\alpha^N = \psi \in \{0, 1\}$, $\beta^N = \lfloor \frac{L}{a+b} \rfloor$, $\gamma^N = L - \psi a - (a+b) \lfloor \frac{L}{a+b} \rfloor$ avec $\psi = 1$ si $L - (a+b) \lfloor \frac{L}{a+b} \rfloor \geq a$ et $\psi = 0$ sinon, avec $\gamma^N \leq (\beta^N + 1)(a - b)$; sinon
 (b) $\alpha^N = \lfloor \frac{L}{a} \rfloor$, $\beta^N = 0$, $\gamma^N = L - a \lfloor \frac{L}{a} \rfloor$ avec $b > \gamma^N \geq a - b$.

Théorème 1 *Le cycle \mathcal{C}^N est optimal pour le problème d'ordonnancement cyclique de tâches couplées identiques et ce cycle peut être déterminé en temps polynomial.*

Nous avons établi la polynomialité du cas cyclique, il reste à résoudre le problème pour un nombre fini n de tâches couplées identiques. Si n est très grand, l'ordonnancement optimal devra certainement suivre "au milieu" une configuration optimale du cas cyclique. Cependant, les parties initiales et finales de l'ordonnancement devront certainement dépendre de n .

Références

- [1] A. A. Ageev and A. E. Baburin. Approximation algorithms for UET scheduling problems with exact delays. *Operations Research Letters*, 35(4) :533 – 540, 2007.
- [2] D. Ahr, J. Békési, G. Galambos, M. Oswald, and G. Reinelt. An exact algorithm for scheduling identical coupled tasks. *Mathematical Methods of Operations Research (ZOR)*, 59 :193–203, 2004.
- [3] Ph. Baptiste. A note on scheduling identical coupled tasks in constant time. Technical report, private communication, 2009.
- [4] J. Békési, G. Galambos, M. Oswald, and G. Reinelt. Improved analysis of an algorithm for the coupled task problem with UET jobs. *Operations Research Letters*, 37(2) :93 – 96, 2009.
- [5] N. Brauner, Y. Crama, A. Grigoriev, and J. van de Klundert. A framework for the complexity of high-multiplicity scheduling problems. *Journal of Combinatorial Optimization*, 9 :313–323, 2005.
- [6] N. Brauner, Y. Crama, A. Grigoriev, and J. van de Klundert. Multiplicity and complexity issues in contemporary production scheduling. *Statistica Neerlandica*, 61(1) :75–91, 2007.
- [7] N. Brauner, G. Finke, V. Lehoux-Lebacque, C.N. Potts, and J. Whitehead. Scheduling of coupled tasks and one-machine no-wait robotic cells. *Computers and Operations Research*, 36(2) :301–307, 2009.
- [8] J. N. D. Gupta. Comparative evaluation of heuristic algorithms for the single machine scheduling problem with two operations per job and time-lags. *Journal of Global Optimization*, 9 :239–250, 1996.
- [9] A. J. Orman and C. N. Potts. On the complexity of coupled-task scheduling. *Discrete Applied Mathematics*, 72 :141–154, 1997.
- [10] R. D. Shapiro. Scheduling coupled tasks. *Naval Research Logistics Quarterly*, 27(2) :489–497, 1980.
- [11] G. Simonin, B. Darties, R. Giroudeau, and J.-C. König. Isomorphic coupled-task scheduling problem with compatibility constraints on a single processor. In *MISTA*, 2009.