Ordonnancement de tâches périodiques avec précédences étendues sans sémaphores

Julien Forget, Emmanuel Grolleau, Claire Pagetti

▶ **To cite this version:**

Julien Forget, Emmanuel Grolleau, Claire Pagetti. Ordonnancement de tâches périodiques avec précédences étendues sans sémaphores. ROADEF 2011, École Nationale Supérieure des Mines de Saint-Étienne, Mar 2011, SAINT ETIENNE, France. inria-00563798

HAL Id: inria-00563798

https://inria.hal.science/inria-00563798

Submitted on 7 Feb 2011

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Ordonnancement de tâches périodiques avec précédences étendues sans sémaphores

Julien Forget[1], Emmanuel Grolleau[2], Claire Pagetti[3]

[1] LIFL, Université Lille 1, France
[2] LISI/ENSMA, Université de Poitiers, France
[3] ONERA, Toulouse, France

Mots-clés : *ordonnancement, tâches périodiques, précédences*

1 Introduction

Contexte : Notre travail est motivé par la programmation sûre des systèmes embarqués critiques, comme le système de contrôle de vol d'un avion par exemple. Un tel système est d'une part *multi-périodique* (contraintes physiques différentes pour chaque équipement) et d'autre part soumis à des *contraintes d'échéances* (par exemple contrainte bout-en-bout). L'implémentation de ce type de système doit être *fonctionnellement déterministe* (les mêmes sorties produites pour les mêmes entrées). Ceci nécessite de contrôler précisément l'ordre des communications entre tâches, à l'aide de *contraintes de précédence* (afin d'assurer que le producteur d'une donnée s'exécute avant le(s) consommateur(s)).

Ordonnancement de tâches dépendantes : Cet article porte sur l'ordonnancement de tâches périodiques dépendantes (reliées par des contraintes de précédence), à l'aide de politiques basées sur les priorités. On considère un modèle de tâches classique à la Liu & Layland, où chaque tâche τ_i est caractérisée par sa période T_i, son échéance relative D_i, sa date de réveil initiale O_i et son pire temps d'exécution (WCET) C_i. Nous considérons des *précédences étendues*, c'est-à-dire qu'elles peuvent relier des tâches de périodes différentes.

Il existe deux approches dans la littérature pour ordonnancer des tâches dépendantes. Dans la première, le respect des précédences est assuré à l'aide de sémaphores. Cette approche est peu adaptée au contexte critique, principalement parce qu'elle peut mener à des anomalies d'ordonnancement : une analyse d'ordonnançabilité basée sur les WCET est insuffisante car certains systèmes sont ordonnançables si les tâches prennent toutes leur WCET mais non-ordonnançables si certaines prennent moins que leur WCET [3]. Dans la seconde approche, que nous adoptons ici, le respect des précédences est assuré par l'ajustement des attributs temps réel des tâches. Cette approche élimine tout risque d'anomalie ou d'interblocage.

2 Modèle de précédences étendues

Une précédence étendue entre deux tâches τ_i et τ_j correspond à un ensemble de précédences simples entre les instances des deux tâches. Notons $\tau_i[n] \rightarrow \tau_j[n']$ une précédence entre l'instance n de τ_i et l'instance n' de τ_j.

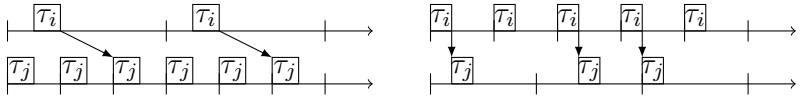
Définition 1 (Précédence étendue) *Une précédence étendue entre deux tâches τ_i et τ_j est définie par une fonction de précédence $\mathcal{P}_{i,j}$ telle que ($\perp$ signifie « indéfini ») :*

$$\mathcal{P}_{i,j} : \mathbb{N}^\star \rightarrow \mathbb{N}^\star \cup \{\perp\}$$
$$\tau_{i.k} \rightarrow \tau_{j.\mathcal{P}_{i,j}(k)} \text{ si et seulement si } \mathcal{P}_{i,j}(k) \neq \perp$$

On se restreint dans la suite au cas des *précédences étendues périodiques* qui permettent de représenter par un motif fini de l'ensemble infini des précédences entre instances. Soit $v.s$ la séquence résultant de la concaténation de la valeur v et de la séquence s. Soit $|s|$ la taille de s et $s[n]$ la $n^{ième}$ valeur de s (avec $0 \leq n < |s|$). Soit $L_{i,j}$ le ppcm de T_i et T_j. La définition ci-dessous est illustrée Figure 1.

Définition 2 (Précédence Etendue Périodique) *Une* précédence étendue périodique *entre deux tâches τ_i et τ_j est définie par une séquence de précédence $pw_{i,j}$, de taille $L_{i,j}/T_i$ et dont les valeurs sont des éléments de $[0, L_{i,j}/T_j[\cup\{\perp\}$. La fonction de précédence $\mathcal{P}_{i,j}$ correspondant à cette séquence est définie comme suit :*

$$\mathcal{P}_{i,j}(k) = pw_{i,j}(k \bmod \frac{L_{i,j}}{T_i}) + \lfloor \frac{k}{\frac{L_{i,j}}{T_i}} \rfloor \times \frac{L_{i,j}}{T_j}$$



(a) $T_i = 3T_j$, $pw_{i,j} = \perp.\perp.2$ (b) $T_i = T_j * \frac{3}{5}, pw_{i,j} = 0.\perp.1.2$

FIG. 1 – Précédence étendue périodique $\tau_i \xrightarrow{M_{i,j}} \tau_j$

3 Contribution

Nous proposons une politique d'ordonnancement pour le problème présenté ci-dessus (ordonnancement de tâches périodiques avec précédences étendues, sans sémaphores), qui est optimale dans la classe des politiques à priorité dynamique. Nous nous basons sur [1], qui propose d'encoder les précédences en ajustant les attributs temps réel des tâches de manière à ce qu'une tâche soit « plus urgente » que ses successeurs et se réveille après ses prédécesseurs :

$$D_i^* = min(D_i, \min_{\tau_j \in succs(\tau_i)}(D_j^* - C_j)) \qquad O_i^* = max(O_i, \max_{\tau_j \in preds(\tau_i)}(O_j^*))$$

L'ensemble de tâches « ajusté » est ensuite ordonnancé à l'aide de la politique EDF.

Cette technique ne porte que sur les précédences simples (entre tâches de même période). Afin de l'étendre aux précédences étendues, nous suivons l'approche proposée dans [2] qui consiste à ajuster séparément les dates de réveil et les échéances des différentes instances d'une même tâche. Par exemple, dans la Figure 1(b) il est nécessaire d'ajuster l'échéance de la première instance de τ_i, mais pas celle de la deuxième. Pour cela, nous définissons des *mots de dates de réveil* et des *mots d'échéances*, qui permettent de représenter la séquence des attributs temps réel des instances d'une tâche à l'aide de mots périodiques finis. Par exemple, si le mot d'échéance d'une tâche est $(3.5)^\omega$, alors sa première instance a pour échéance relative 3, la seconde 5, la troisième 3, la quatrième 5, etc. On ordonnance ensuite cet ensemble ajusté à l'aide d'une politique EDF très légèrement modifiée afin de supporter ces mots périodiques. Pour le test d'ordonnançabilité, comme décrit dans [4], on déplie l'ensemble ajusté sur l'hyperpériode du système et on effectue un test d'ordonnançabilité classique pour EDF sur ce dépliage.

Références

[1] H. Chetto, M. Silly, and T. Bouchentouf. Dynamic scheduling of real-time tasks under precedence constraints. *Real-Time Systems*, 2, 1990.

[2] J. Forget. *Un Langage Synchrone pour les Systèmes Embarqués Critiques Soumis à des Contraintes Temps Réel Multiples.* PhD thesis, Université de Toulouse, 2009.

[3] M. Richard, P. Richard, E. Grolleau, and F. Cottet. Contraintes de précédences et ordonnancement mono-processeur. In *Real-time and embedded systems (RTS'02)*, 2002.

[4] P. Richard, F. Cottet, and M. Richard. On-line scheduling of real-time distributed computers with complex communication constraints. In *ICECCS'2001*, Skövde (Sweden), 2001.