

Génération de colonnes pour l'ordonnement modulo sous contraintes de ressources

Ayala Maria^{1,2}, Christian Artigues^{1,2}

¹ CNRS ; LAAS ; 7 avenue du Colonel Roche, F-31077 Toulouse

² Université de Toulouse ; UPS, INSA, INP, ISAE ; LAAS ; F-31077 Toulouse
{mayala,artigues}@laas.fr

mots clés : ordonnancement modulo, programmation linéaire, génération de colonnes.

1 Introduction et présentation du problème

Beaucoup d'études dans le domaine de la compilation pour les processeurs superscalaires modernes et architectures VLIW (Very Large Instruction Word) sont focalisées sur le problème d'ordonnement d'instructions. Ce problème est défini par un ensemble d'opérations à ordonner, un ensemble de dépendances entre ces opérations et la micro architecture du processeur cible qui définit des contraintes des ressources complexes. Une opération est considérée comme une instance d'une instruction dans un programme. L'ordonnement d'instructions par boucles internes est connu sous le nom de pipeline logiciel. Le pipeline logiciel est une méthode efficace pour l'optimisation de boucles qui permet la réalisation en parallèle des opérations des différentes itérations de la boucle. Aujourd'hui, les compilateurs commerciaux utilisent des méthodes basées sur les algorithmes d'ordonnement modulo. L'ordonnement modulo est une structure d'ordonnements cycliques 1-périodiques avec période entière. Cela rend le modèle plus simple que le modèle classique d'ordonnement cyclique et aussi plus facilement implémentable. Le modulo-ordonnement est une technique qui crée des liens entre les itérations successives d'une boucle. Le but est de trouver un ordonnancement valide pour une boucle pour qu'elle puisse se superposer plusieurs fois dans un intervalle constant, que nous appelons intervalle d'initiation ou période (λ). L'algorithme d'ordonnement modulo doit prendre en compte les contraintes de dépendance des opérations, les contraintes de ressources (et la taille des registres qui sont ignorés dans cette étude). Il doit aussi considérer les critères d'optimisation tels que la minimisation de l'intervalle d'initiation de l'ordonnement λ ou la minimisation de la durée de l'ordonnement d'une itération de la boucle. Le problème d'ordonnement modulo consiste à trouver les dates de début d'un ensemble d'opérations génériques de telle sorte que dans le contexte cyclique, les instances des opérations génériques sont répétées toutes les λ unités de temps. La date de début d'une instance d'opération est donc définie par deux valeurs : la date de début σ_i dans une période générique, variant de 0 à $\lambda - 1$ et le numéro k_i de la période. L'aspect modulo permet de limiter le nombre d'opérations à considérer aux seules opérations génériques. Les ressources sont disponibles en quantités limitées et chaque opération demande une quantité positive ou nulle de chaque ressource. Les demandes cumulées des opérations en cours d'exécution à chaque instant modulo λ sur chaque ressource ne doivent pas dépasser sa limitation. Les contraintes de dépendances sont définies relativement aux dates de début "effectives" $\sigma_i + k_i\lambda$. Le problème consiste à trouver un ordonnancement de période minimale, puis de durée totale minimale (optimisation lexicographique).

2 Génération de colonnes pour l'ordonnancement modulo.

Dans ce travail, nous cherchons à calculer des bornes inférieures de λ et du makespan pour un λ fixé. Nous prouvons l'équivalence des relaxations de deux formulations de programmation linéaire en nombres entiers proposées par Eichenberger et Davidson [1] et Dupont de Dinechin [2] basées sur des variables indicées par le temps et contraintes de précédences renforcées. Nous proposons ensuite deux extensions de ces formulations reprenant l'idée des sous-ensembles d'activités qui peuvent être exécutées simultanément, sans violer les contraintes de ressources ou de précedence, proposée par Mingozzi et al [3] pour l'ordonnancement de projet sous contraintes de ressources (RCPSP). Ces formulations nécessitent un grand nombre de variables 0 – 1 (liées aux sous-ensembles d'activités). Nous proposons un schéma de génération de colonnes pour résoudre leurs relaxations continues. Le sous-problème correspond à λ problèmes de sac-à-dos multidimensionnel ce qui permet d'identifier les conditions pour que la génération de colonnes améliore significativement la borne inférieure des formulations [1] et [2]. Lorsque les demandes des opérations en ressources sont unitaires, les bornes sont identiques. Lorsque les demandes des tâches sont non unitaires, l'amélioration moyenne de la borne inférieure est de 42% sur 34 instances. L'augmentation du temps CPU est significative mais l'algorithme de génération de colonnes pourrait être optimisé d'avantage. Les résultats expérimentaux pour comparer les différentes formulations proposées sont réalisés en utilisant des instances de RCMSPP provenant du compilateur commercial ST200 en modifiant aléatoirement les demandes de ressources.

Formulations	adpcm-st231.1			gsm-st231.1			gsm-st231.2			gsm-st231.5		
	λ_{min}	CPU(s)	Cmax	λ_{min}	CPU(s)	Cmax	λ_{min}	CPU(s)	Cmax	λ_{min}	CPU(s)	Cmax
Eichenberger et Davidson [1]	52	22.07	38.65	24	1.98	32.97	59	29.04	43.36	26	0.12	19.35
Dupont-de-dinechin et al [2]	52	25	38.65	24	0.41	32.97	59	71.01	43.36	26	2.09	17.35
Column Generation-Eichenberger	78	≥ 500	49.30	25	25.03	32	93	≥ 500	57.82	36	≥ 500	23.44
Column Generation-Dupont de Dinechin	78	≥ 500	49.30	25	39.02	32	93	≥ 500	57.82	36	≥ 500	23.44

TAB. 1 – bornes inférieure de λ et du Cmax avec demandes de ressources non unitaires

Nous montrons les résultats pour quatre instances significatives. Dans trois cas, nous constatons qu'avec les extensions de génération de colonnes la borne inférieure pour λ a été améliorée.

Références

- [1] A Eichenberger, E. S. Davidson .Efficient Formulation for Optimal Modulo Schedulers. *SIG-PLAN Conference on Programming Language Design and Implementation - PLDI'97* (1997).
- [2] B.Dupont-De-Dinechin. C. Artigues. S. Azem. Resource Constrained Modulo Scheduling. In C. Artigues, S. Demassej and E. Néron. *Resource-Constrained Project Scheduling Models, algorithms, extensions and applications*. November (2007).
- [3] A. Mingozzi and V. Maniezzo. An Exact Algorithm For The Resource Constrained Project Scheduling Problem Based on a New Mathematical Formulation. *Management Science* 44, 714-729. (1998).