Ordonnancements, flots et couplages verts

Evripidis Bampis[1], Dimitrios Letsios[1,2], Giorgio Lucarelli[1]

[1] LIP6 UMR 7606, Université Pierre et Marie Curie,
Place Jussieu, 750005 Paris, France
{evripidis.bampis, giorgio.lucarelli}@lip6.fr

[2] IBISC EA 4526, Université d'Evry Val d'Essonne
Boulevard Mitterrand, 91025 Evry, France
dimitris.letsios@ibisc.fr

Mots-clés : *Ordonnancement, énergie, flot, couplage*

1 Introduction

Nous nous intéressons à l'ordonnancement de tâches sur un ensemble des machines qui sont capables de changer dynamiquement leurs vitesses d'exécution et notre objectif est de déterminer un ordonnancement qui garantit un certain niveau de Qualité de Service tout en consommant le moins d'énergie possible. Dans ce contexte, l'ordonnanceur doit décider pour chaque instant non seulement quelle tâche va être exécutée sur chaque machine, mais également la vitesse de chaque machine. Plus grandes seront les vitesses des machines, plus rapide sera l'exécution des tâches, mais plus grande sera la consommation d'énergie. Le calcul de la consommation d'énergie d'une machine est basé sur le modèle proposé par Yao, Demers et Shenker [4], connu comme le modèle de variation de vitesse, selon lequel la puissance de la machine est une fonction convexe de sa vitesse $P(s) = s^\alpha$, où $\alpha > 1$ est une constante qui dépend de la machine. La vitesse de la machine est définie comme la quantité de travail exécutée par unité de temps. L'énergie consommée peut être calculée en intégrant la puissance au fil du temps. Ainsi, si la machine fonctionne avec une vitesse constante s durant un intervalle de temps $[t, t']$, alors elle exécute une quantité de travail égale à $(t' - t) \cdot s$ et sa consommation d'énergie est égale à $(t' - t) \cdot s^\alpha$.

Il existe un grand nombre de papiers étudiant les problèmes d'ordonnancement en tenant compte de la consommation d'énergie [1]. Dans ce travail, nous proposons l'utilisation des flots convexes et des couplages de poids minimum pour la résolution de trois problèmes d'ordonnancement de tâches dans le modèle de variation de vitesse.

2 Contributions

Le premier problème qui nous intéresse est le suivant : on se donne un ensemble des tâches $J = \{J_1, J_2, \ldots, J_n\}$, chacune caractérisée par une quantité de travail w_j, une date de relâchement r_j et une date d'échéance d_j et m machines à vitesse variable. On veut déterminer un ordonnancement où toutes les tâches respectent leurs dates de relâchement et d'échéance minimisant l'énergie totale consommée. La préemption et la migration de tâches est autorisée, i.e., l'exécution d'une tâche peut être interrompue et continuée plus tard sur la même ou une autre machine. Nous notons ce problème comme $S|pmtn, r_j, d_j|E$. Ce problème a été déjà résolu de manière indépendante par Albers et al. [2] et Angel et al. [3]. Une propriété importante que l'on utilise est que dans un ordonnancement optimal, chaque tâche s'exécute avec une vitesse unique. Ces travaux proposent des algorithmes combinatoires basés sur le calcul des vitesses de tâches en utilisant des calculs répétés des flots maximums sur des réseaux de transports appropriés. L'ordonnancement optimal est ensuite calculé en résolvant le problème de faisabilité $P|pmtn, r_j, d_j|-$. Les preuves d'optimalité sont basées sur plusieurs lemmes techniques. Ici, nous montrons comment calculer les vitesses de tâches en utilisant le calcul d'un flot convexe unique. Notre approche permet de simplifier largement la preuve d'optimalité.

Préliminaires. Pour formuler notre problème nous avons besoin de notations suivantes : Nous considérons que le temps est partitionné dans des intervalles définis par les dates de relâchement et les dates d'échéance de tâches. Ainsi, nous définissons les instants $t_0, t_1, \ldots, t_k$ dans l'ordre croissant, où t_i correspond soit à une date de relâchement, soit à une date d'échéance, de telle sorte à ce que chaque date de relâchement et chaque date d'échéance, correspond à un t_i. Ainsi nous définissons les intervalles $I_i = [t_{i-1}, t_i]$, pour $1 \leq i \leq k$, et nous notons par $|I_i|$ la longueur de I_i. Nous appelons une tâche J_j disponible à un intervalle donné I_i, si $I_i \subseteq [r_j, d_j]$. Le nombre de tâches disponibles dans un intervalle I_i sera noté $A(I_i)$. Une propriété importante de tout ordonnancement optimal est que la somme des durées d'exécution de toutes les tâches est égale à $\sum_{i=1}^{k} |I_i| \min\{|A(I_i)|, m\}$.

Formulation comme un flot convexe. Le réseau N_s (voir Figure 1) construit par notre algorithme est le suivant. Nous introduisons un sommet source s, un sommet destination t, un sommet pour chaque tâche J_j, $1 \leq j \leq n$, et un sommet pour chaque intervalle I_i, $1 \leq i \leq k$. Pour chaque j, $1 \leq j \leq n$, nous rajoutons un arc (s, J_j) et, pour chaque i, $1 \leq i \leq k$, nous rajoutons un arc (I_i, t). Si la tâche J_j, $1 \leq j \leq n$, est disponible durant l'intervalle I_i, $1 \leq i \leq k$, nous introduisons un arc du sommet J_j au sommet I_i. Notons $c_{u,v}$ la capacité de l'arc (u, v) et $\kappa_{u,v}(f_{u,v})$ la fonction coût de l'arc (u, v), si une quantité de flot $f_{u,v}$ passe à travers l' (u, v). On définit que

$$c_{u,v} = \begin{cases} +\infty & \text{si } u = s \text{ et } v = J_j \\ |I_i| & \text{si } u = J_j \text{ et } v = I_i, \\ m|I_i| & \text{si } u = I_i \text{ et } v = t \end{cases} \qquad \kappa_{u,v}(f_{u,v}) = \begin{cases} f_{u,v} \cdot P(\frac{w_j}{f_{u,v}}) & \text{si } u = s \text{ et } v = J_j \\ 0 & \text{si } u = J_j \text{ et } v = I_i \\ 0 & \text{si } u = I_i \text{ et } v = t \end{cases}$$

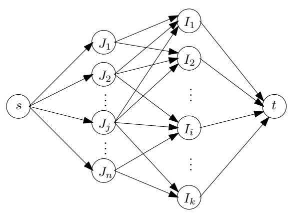

FIG. 1 – Le réseau N_s pour le problème de minimisation d'énergie considéré et le problème classique de faisabilité $P|r_j, d_j, pmtn|-$, respectivement.

En calculant un flot convexe de valeur $\sum_{i=1}^{k} |I_i| \min\{|A(I_i)|, m\}$ de coût minimum, on obtient les vitesses de tâches minimisant l'énergie totale consommée.

Extensions. Cette approche peut être utilisée pour la résolution d'autres problèmes d'ordonnancement dans le modèle à variation de vitesse. Nous avons ainsi été capables de résoudre la variante préemptive du problème "open-shop" en le formulant comme un flot convexe. Nous avons également utilisé une approche similaire, basée sur le calcul d'un couplage de coût minimum, pour résoudre le problème d'ordonnancement sur plusieurs machines où l'objectif est la minimisation de la somme pondérée de l'énergie consommé et de la somme de temps de complétude des tâches.

Références

[1] S. Albers. Energy-Efficient Algorithms. *Communications of the ACM*, 53(5) :86-96, 2010.

[2] S. Albers, A. Antoniadis, G. Greiner. On Multi-processor Speed Scaling with Migration. *ACM Symposium on Parallelism in Algorithms and Architectures (SPAA)*, p. 279-288, 2011.

[3] E. Angel, E. Bampis, F. Kacem, D. Letsios. Speed Scaling on Parallel Processors with Migration. *International European Conference on Parallel and Distributed Computing (Euro-Par)*, LNCS 7484, p. 128-140, 2012.

[4] F. F. Yao, A. J. Demers, S. Shenker. A Scheduling Model for Reduced CPU Energy. *IEEE Symposium on Foundations of Computer Science (FOCS)*, p. 374-382, 1995.