MIM2 - Algorithmes d'approximation- TD 8
Emmanuelle Lebhar
elebhar@ens-lyon.fr - Bureau 317

Minimisation du temps d'exécution total
12 Novembre 2003

Minimiser le temps d'exécution total ("minimum makespan scheduling" en anglais) Étant donnés les durées d'exécution de n tâches (indépendantes), $p_1, p_2, \ldots, p_n$, et un entier m, ordonnancer les tâches sur m machines identiques de sorte à minimiser le *temps d'exécution total* ("makespan" ou "completion time" en anglais).

Exercice 1 (Une 2-approximation pour temps d'exécution minimum)

L'algorithme est très simple : nous traîtons les tâches les unes après les autres dans un ordre arbitraire, et nous les plaçons systématiquement sur la machine courante la moins chargée.

1. Comparez l'optimal à la quantité suivante :

$$\mathrm{MN} = \max\left\{\frac{1}{m}\sum_i p_i,\ \max_i\{p_i\}\right\}.$$

2. Montrez que cet algorithme est une 2-approximation.
3. Donnez une instance critique pour cet algorithme.

Exercice 2 (Empaquetage restreint)

On suppose que toutes les boîtes sont de taille 1 et que les objets ont k tailles possibles. Une instance du problème d'empaquetage est définie par le k-uplet $(i_1, i_2, \ldots, i_k)$, où i_j est le nombre d'objets de la j-ème taille possible.

Notons BOÎTES$(i_1, i_2, \ldots, i_k)$ le nombre de boîtes utilisées par un empaquetage optimal de ces objets.

1. Considérons une instance $(n_1, n_2, \ldots, n_k)$ $(n = \sum_{i=1}^k n_i)$. Soit $\mathcal{Q}$ l'ensemble de tous les k-uplets $(q_1, q_2, \ldots, q_k)$, $0 \leq q_i \leq n_i$ et $1 \leq i \leq k$, tels que BOÎTES$(q_1, q_2, \ldots, q_k) =$ 1. Remplissez la table k-dimentionnelle BOÎTES$(i_1, i_2, \ldots, i_k)$, pour $(i_1, i_2, \ldots, i_k) \in \{0, \ldots, n_1\} \times \{0, \ldots, n_2\} \times \cdots \times \{0, \ldots, n_k\}$.
2. Quel est le temps de calcul de BOÎTES$(n_1, n_2, \ldots, n_k)$ de cet algorithme ?

Exercice 3 (Un PTAS pour temps d'exécution minimum)

Le problème du temps d'exécution minimum est **NP**-difficile au sens fort, il n'admet donc pas de FPTAS, à moins que $\mathbf{P} = \mathbf{NP}$. Nous pouvons cependant obtenir un PTAS pour ce problème.

1. Réduisez ce problème au problème d'empaquetage. Peut-on utiliser directement cette réduction ?
2. Remarquez que le temps d'exécution optimal appartient à $[\mathrm{MN}, 2\,\mathrm{MN}]$. Soit $t \in [\mathrm{MN}, 2\,\mathrm{MN}]$. Soit ε le paramètre de précision, nous dirons qu'un objet est *petit* si sa taille est inférieure à $t\varepsilon$; nous omettons ces objets pour l'instant.
   Les tailles des autres objets sont arrondies vers le bas comme suit : pour chaque j, soit $i \geq 0$ tel que $p_j \in [t\varepsilon(1+\varepsilon)^i,\ t\varepsilon(1+\varepsilon)^{i+1})$, remplacer p_j par $p'_j = t\varepsilon(1+\varepsilon)^i$.
   Calculez un empaquetage optimal pour les objets arrondis dans des boîtes de taille t. Pour quelle taille de boîte peut-on utiliser cet empaquetage pour les objets réels ?
3. Plaçons les petits objets de façon gloutonne dans ce dernier empaquetage, en ouvrant de nouvelles boîtes seulement lorsque c'est nécessaire. Notons $\alpha(I, t, \varepsilon)$ le nombre de boîtes utilisées. On note $\mathrm{boites}(I, t)$ le nombre maximum de boîtes de taille t pour empaqueter l'instance I. Montrez que $\alpha(I, t, \varepsilon) \leq \mathrm{boites}(I, t)$. Déduisez une minoration de l'optimal.
4. On effectue une recherche dichotomique sur l'intervalle $[\mathrm{MN}, 2\,\mathrm{MN}]$ pour calculer

   $$\min\{t :\ \alpha(I, t, \varepsilon) \leq m\}$$

   Soit T l'extrémité droite de l'intervalle résultant. Montrez que $T \leq (1+\varepsilon) \cdot \mathrm{OPT}$. Quel est le nombre d'itérations ?
5. Estimez le temps de calcul de cette $(1+\varepsilon)$-approximation.