TD - Algorithmes d'approximation

Algorithmique Avancée – ENSIMAG 2A

1 Sac à dos binaire

Soient n objets: pour $i = 1, \dots n$, l'objet i a un volume v_i et apporte un gain g_i. On veut choisir certains de ces objets pour les mettre dans un sac de capacité volumique maximale C, avec pour objectif de maximiser le gain du sac, c'est à dire la somme des gains des objets mis dans le sac.
Remarque: on verra qu'il n'existe pas d'algorithme polynomial pour résoudre ce problème, sauf si P=NP.

1. On considère l'algorithme glouton suivant. On parcourt les objets par gain volumique décroissant: si l'objet courant à un volume inférieur au volume libre restant dans le sac, alors on l'ajoute au sac. Donner un exemple avec deux objets où cet algorithme glouton n'est pas optimal.
   Montrer que, pour tout $\epsilon > 0$, il existe une instance pour laquelle l'algorithme glouton donne une solution de valeur inférieure à $\epsilon.\bar{\phi}$ où $\bar{\phi}$ est la valeur optimale.

2. On suppose toujours les objets triés par gain volumique décroissant, et tous de volume $v_i \leq C$. Soit s le plus grand entier tel que $\sum_{i=1}^{s} v_i \leq V$. On considère l'algorithme qui fournit en sortie la solution de valeur $\max\{\sum_{i=1}^{s} g_i; g_{s+1}\}$: donc soit les objets $\{1, \dots, s\}$, soit l'objet $s+1$ seul. Montrer que cet algorithme est une approximation à un facteur au plus 2 de l'optimal.

3. Majorer l'écart absolu entre la solution fournie et l'optimal: en déduire une approximation plus fine qu'un facteur 2, le facteur d'approximation $c_x < 2$ dépendant de l'instance x en entrée.

2 Bin-packing et first-fit

Soient n objets de taille s_i avec $0 < s_i < 1$. On veut ranger ces n objets dans des sacs de taille 1, en utilisant un nombre minimum de sacs. Sot $C^\star$ le nombre minimal de sacs avec un rangement optimal.
L'algorithme glouton "first-fit" consiste à prendre les objets 1 par 1 et mettre un objet dès qu'on trouve un sac dans lequel il rentre. Soit C le nombre de sacs nécessaires pour first-fit. L'ordre de parcours des sacs (par exemple si l'on parcourt le sac que l'on vient d'ajouter en premier ou en dernier) peut avoir une influence sur la performance.

1. Montrer qu'un minorant de $C^\star$ est : $C^\star \geq \lceil \sum_{i=1}^{n} s_i \rceil$.

2. Montrer que l'heuristique first-fit laisse au plus un sac à moitié vide au moins. En déduire que l'entier C vérifie $C \leq \lceil 2. \sum_{i=1} s_i \rceil$.

3. En déduire que First-Fit est à un facteur 2 de l'optimal (2-approximation relative).

4. Donner un exemple de valeurs pour les s_i pour lequel FirstFit donne la solution optimale.

5. Donner un exemple d'ordre de parcours des sacs et de valeurs pour les s_i pour lequel FirstFit atteint le ratio 2 (asymptotiquement).

6. Une analyse plus fine[1] montre qu'avec FirstFIT $C \leq \left\lceil \frac{17}{10} C^\star \right\rceil = \lceil 1.7 C^\star \rceil$ (ce majorant étant atteint). Un raffinement sur FirstFit, nommée FFD pour *First Fit Decreasing*, consiste à d'abord trier les objets par taille décroissante. Les objets sont alors placés un par un, ceux de plus grande taille d'abord, dans le premier sac qui convient. Ce raffinement de FirstFit (dont l'analyse initiale due à D.S. Johnson a été raffinée par G. Dósa) demande d'allouer seulement $C \leq \frac{11}{9} C^\star + \frac{2}{3}$ (majorant atteint), soit un ratio de performance asymptotique (pour C grand) $\frac{11}{9} \simeq 1.22$. De plus, le ratio absolu est $\frac{C}{C^\star} \leq \frac{4}{3} \simeq 1.33$, et atteint pour quasi toutes les valeurs de C. Ceci montre l'intérêt de trier les objets: sans tri, FirstFit a nécessairement un ratio absolu 1.7 (atteint).
   FFD est-il plus compliqué à implémenter que FirstFit? quel est le coût?

[1] *First Fit bin packing: A tight analysis.* G. Dósa, J. Sgall. STACS, volume 20 of LIPIcs, page 538-549. 2013.