Modélisation et Optimisation M1 Informatique

TP1 - Knapsack - Relaxation Linéaire et Heuristique Gloutonne

F. Bendali, R. Chicoisne, J. Figueroa, D. Perdigao *2023-24*

Le problème du sac à dos unidimensionnel est défini par le programme P suivant :

$$(P) \qquad \begin{aligned} z = \max_x \quad & \sum_{j=1}^{n} c_j x_j \\ \text{s.t.} \quad & \sum_{j=1}^{n} a_j x_j \leqslant b \\ & x_j \in \{0,1\}, \qquad \forall j \in \{1,..,n\} \end{aligned}$$

où b, a_j, et c_j pour tout $j = 1, ..., n$ sont des entiers positifs.

1 Relaxation Linéaire

La relaxation Linéaire de P est obtenue en remplaçant les contraintes $x \in \{0,1\}^n$ par $x \in [0,1]^n$:

$$(LP) \qquad \begin{aligned} \bar{z} = \max_x \quad & \sum_{j=1}^{n} c_j x_j \\ \text{s.t.} \quad & \sum_{j=1}^{n} a_j x_j \leqslant b \\ & x_j \in [0,1], \qquad \forall j \in \{1,..,n\} \end{aligned}$$

L'algorithme 1 permet de résoudre LP en temps $O(n \ln n)$:

2 Heuristique Gloutonne

En utilisant la même idée, nous proposons de générer des solutions réalisables par l'heuristique gloutonne suivante décrite dans l'algorithme 2:

3 Travail demandé

1. Génération d'exemples pour $n \leqslant 100, b \leqslant 1000$.
2. Résolution de la relaxation par l'algorithme 1.

Algorithm 1: Knapsack - Linear Relaxation

Data: A problem LP

Result: A solution $\bar{x}$ that is optimal for LP

Sort the items by decreasing *utility* $c_{[1]}/a_{[1]} \geqslant c_{[2]}/a_{[2]} \geqslant \cdots \geqslant c_{[n]}/a_{[n]}$;
$\bar{x} \leftarrow (0, \cdots, 0)$; $\bar{b} \leftarrow b$;
for $j = 1, .., n$ **do**
    if $\bar{b} = 0$ **then**
        return $\bar{x}$;
    $\bar{x}_{[j]} \leftarrow \min\{\bar{b}/a_{[j]}, 1\}$;
    $\bar{b} \leftarrow \bar{b} - \bar{x}_{[j]} a_{[j]}$;
return $\bar{x}$;

Algorithm 2: Knapsack - Greedy

Data: A problem P

Result: A solution $\tilde{x}$ that is feasible for P

Sort the items by decreasing *utility* $c_{[1]}/a_{[1]} \geqslant c_{[2]}/a_{[2]} \geqslant \cdots \geqslant c_{[n]}/a_{[n]}$;
$\tilde{x} \leftarrow (0, \cdots, 0)$; $\bar{b} \leftarrow b$;
for $j = 1, .., n$ **do**
    if $\bar{b} = 0$ **then**
        return $\tilde{x}$;
    if $\bar{b} \geqslant a_{[j]}$ **then**
        $\tilde{x}_{[j]} \leftarrow 1$;
        $\bar{b} \leftarrow \bar{b} - a_{[j]}$;
return $\tilde{x}$;

3. Génération d'une solution réalisable par l'algorithme glouton 2.

4. Déterminer les courbes de complexité temps de chacun des algorithmes pour

   (a) n fixé, b variable et

   (b) b fixé, n variable.