

1 Rendu de monnaie

On considère le problème du rendu de monnaie : on cherche à faire une certaine somme (exprimée centimes, mettons) avec le moins de pièces possibles. Les valeurs de pièces sont imposées mais on n'a pas de limite sur le nombre de pièces disponibles. L'algorithme glouton consiste à prendre à chaque étape la pièce ayant la plus grosse valeur qui ne dépasse pas la somme restant à obtenir.

1. Montrer qu'avec les pièces usuelles de valeurs 1, 2, 5, 10, 20 et 50, l'algorithme glouton donne une solution optimale.

Indication : considérer le nombre maximum de pièces de chaque catégorie dans une solution optimale.

2. Trouver un ensemble de valeurs pour lesquelles l'algorithme glouton ne donne pas toujours de solution optimale.
3. Montrer que l'algorithme glouton donne toujours une solution optimale dans le cas où les valeurs des pièces sont les puissances successives d'une même valeur : $1, p, p^2, \dots, p^k$ pour $p \geq 2$ et $k \geq 1$ fixés.

2 Un problème d'ordonnement de tâches

On considère un ensemble $E = \{1, 2, \dots, n\}$ de tâches unitaires, c'est-à-dire que chaque tâche s'effectue en une unité de temps. Chaque tâche i a une date d'échéance d_i et une pénalité $v_i > 0$ qui doit être payée si la tâche est effectuée après la date d_i . On cherche un ordonnancement des tâches, c'est-à-dire une permutation de E , qui minimise le cumul des pénalités pour non respect des échéances.

On définit qu'un ensemble de tâches $F \subseteq E$ est *indépendant* s'il existe un ordonnancement des tâches de F tel qu'aucune ne soit en retard sur sa date d'échéance.

1. Montrer que le calcul d'un ordonnancement optimal équivaut à trouver un ensemble indépendant de tâches de E dont la somme des pénalités soit le plus grand possible.
2. Soit i une tâche de pénalité maximale. Existe-t-il un ensemble indépendant optimal qui contient i ?
3. En déduire un algorithme glouton qui fournit une solution optimale.