

Recherche Opérationnelles : TP n°1

Le problème du sac à dos

Le problème du sac à dos est un problème d'optimisation combinatoire. Il modélise une situation analogue au remplissage d'un sac à dos ayant une capacité maximale en poids à ne pas dépasser. Les objets mis dans le sac à dos doivent maximiser la valeur totale. Dans ce TP, nous allons implémenter les deux algorithmes naïfs vus en cours (Brute-force et Glouton).

Partie 1 : Générer les 2^n combinaisons

- Étant donné un tableau de n cases contenant des valeurs binaires, concevoir une fonction qui affiche toutes les combinaisons possibles. Vous devez compter 2^n solutions possibles.

Partie 2

Nous considérons un sac à dos de capacité maximale de 10 kg. Nous devons choisir parmi 6 objets dont les poids et les valeurs respectives sont données comme suit :

$$Poids[] = \{2, 3, 4, 3, 2, 7\}$$

$$Valeur[] = \{4, 5, 2, 1, 4, 14\}$$

Algorithme Brute-force

- Étendre vos codes de la Partie 1 afin d'implémenter l'algorithme Brute-force pour le problème du sac à dos.

Algorithme Glouton

- Implémenter l'algorithme glouton vu en cours.
- Comparer les résultats des deux algorithmes sur l'exemple considéré.