Rapport du Projet d'ORO

Bouton Nicolas

Avril 2021

Contents

1 But du projet

Le but du projet d'Optimisation et de Recherche Opérationnelle est d'implémenter en **C** un algorithme de séparation et d'évaluation (Branch and Bound) de notre choix.

1.1 Choix du problème à résoudre

J'ai personnellement choisis le problème du sac à dos pour sa simplicité et du fait que j'ai tout de suite pensé à faire un lecteur de format csv pour ce problème, que je décrirais plus tard.

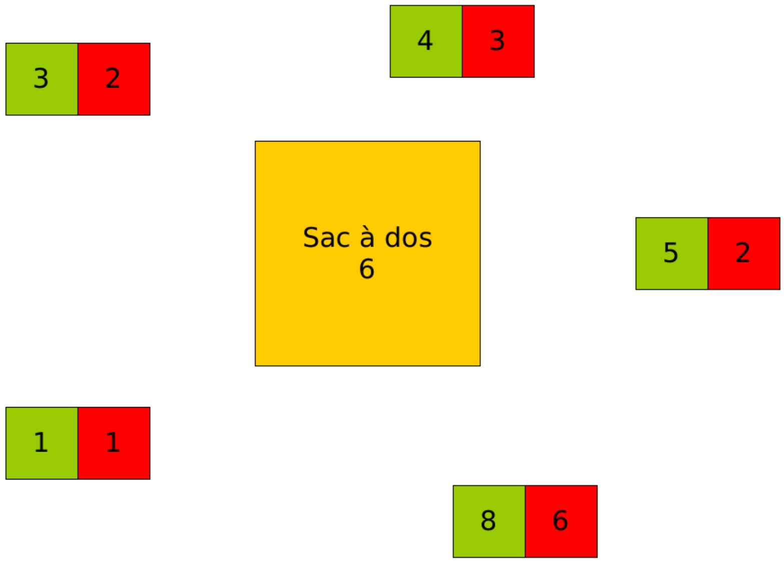

Figure 1: Problème du sac à dos

2 Problème du sac à dos

Le problème du sac à dos est de le remplir afin d'avoir une valeur maximum sans exéder son poids maximum (ici 6). Pour ce faire il y a des pairs d'objets avec des valeurs (ici en vertes) et des poids (ici en rouges).

2.1 Démonstration de l'algorithme implémenté

2.1.1 Tri de valeur par poid

Mon algorithme tri d'abord les paires par valeur par poid. C'est à dire qu'il tri en ordre décroissant les paires suivant leurs taux valeur/poids.

1. Exemple

   Imaginons que nous ayons l'entrée suivante:

   ```
   "value", "weight"
   3, 2
   4, 3
   1, 1
   8, 6
   5, 2
   ```

Dans ce cas nous aurions des taux:

```
"rate"
1.5
1.333333
1.0
1.333333
2.5
```

L'algorithme de tri mettra la pair (5, 2) en premier, et déplacera la pair (8, 6) d'un cran (car on ne change de place que si c'est strictement supérieur):

```
"value", "weight"
5, 2
3, 2
4, 3
8, 6
1, 1
```

2.1.2 Résolution par récursion

Le système de résolution par récursion est simple, à **gauche** de l'arbre on prend la pair, à **droite** on ne prend pas la pair. **Si on prend la pair**, alors il ne faut pas que le sac déborde sinon on met la valeur à -1 pour dire que le sous arbre n'est pas réalisable (c'est à dire que pour toute les noeuds suivant la solution reste iréalisable). Si le sac ne déborde pas on continue. Si on arrive à la fin alors on calcul la valeur du sac et on le stock dans un tableau suivant la position de la feuille dans l'arbre.

Sur les arrêtes on retrouve la valeur actuelle des éléments pris. A la fin on vois un tableau qui contient les valeurs de chaque chemin ce qui nous permettra ensuite de savoir quelle est la valeur maximal réalisable.

2.1.3 Trouve le chemin de la valeur maximale possible

Maintenant qu'on a le tableaux contenant la valeur maximale de tout les chemins. On peut trouver la valeur maximale réalisable qui est en fait la valeur maximale car toutes les valeurs non réalisables sont à -1. Ensuite pour savoir quel chemin à été parcourus on se base sur le fait que c'est un arbre binaire et donc on a la propriété suivante:

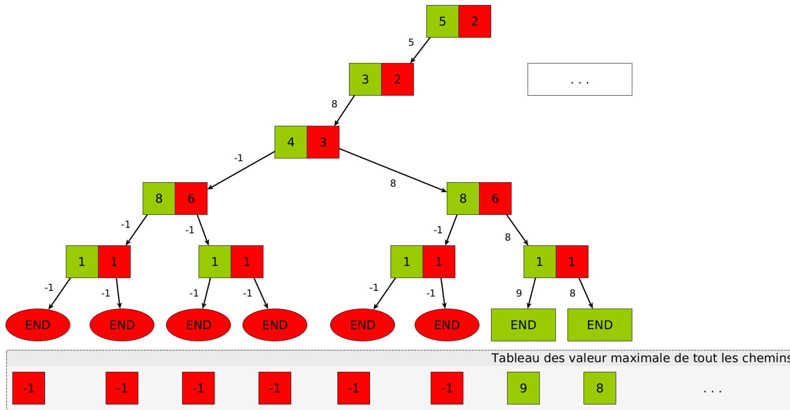

Figure 2: Démonstration de l'arbre

- gauche (on prends) égale 0 modulo 2
- droite (on ne prends pas) égale 1 modulo 2

Dans ce cas on pars des feuilles et on remonte l'arbre pour savoir quelle pair on a pris.

3 Choix d'implémentaion

4 Conclusion

Pour conclure, le travail fait pour résoudre ce problème peut être améliorer étant donné que l'algorithme pour le résoudre parcours tout les chemins, contrairement à l'algorithme que l'on a vu en cours qui parcourrait changeait le chenmin à parsourir à chaque étape.