Université de Reims Champagne-Ardenne
Département Math., Méca. et Informatique

Pierre Delisle
Chps0742 - Optimisation
2018/2019

Travaux Pratiques 2
Résolution optimale du problème du Sac à Dos

L'objectif de ce TP est de programmer une méthode par séparation et évaluation pour résoudre de façon optimale le problème du sac à dos à l'aide du solveur GLPK. Pour ce faire, nous utiliserons le programme linéaire en nombres entiers suivant comme exemple (vu en cours) :

Maximiser $z = 10x1 + 8x2 + 5x3$
Sous les contraintes
$6x1 + 5x2 + 4x3 \leq 9$
$x1, x2, x3 \geq 0$
$x1, x2, x3 \in \mathbb{N}$

Vous pouvez télécharger le fichier de données correspondant à ce problème (**prob_sac_1.txt**) sur le site Web du professeur (http://cosy.univ-reims.fr/~pdelisle/enseignement.php).

Lecture du fichier de données

Vous noterez que la structure du fichier correspond à celle d'un programme linéaire tel que vu au TP 1. Par conséquent, la fonction de lecture de programme linéaire de ce TP devrait faire l'affaire. Vérifiez quand même le résultat à l'écran.

Calcul de la borne – Approche heuristique qualité-prix

L'approche heuristique qualité-prix permet de trouver une solution approchée au problème du sac à dos. Elle consiste à classer les variables en ordre décroissant de leur rapport utilité/volume, puis à donner à chaque variable la plus grande valeur entière possible en tenant compte des choix précédents et de la contrainte de volume total.

1) Écrivez une fonction implémentant l'approche heuristique qualité-prix. Vérifiez que vous obtenez bien le résultat suivant :

```
Heuristique qualite-prix :
x1 = 1, x2 = 0, x3 = 0
Utilite totale : 10
```

Cette solution sera utilisée comme borne initiale de la méthode par séparation et évaluation.

Fonction d'évaluation – Résolution du programme linéaire en variables réelles

Une méthode par séparation et évaluation nécessite l'utilisation d'une fonction d'évaluation qui permettra d'évaluer un sommet de l'arborescence en cours de construction. Dans la méthode proposée, cette évaluation consistera en l'utilité obtenue par la résolution du programme linéaire en variables réelles, c'est-à-dire avec relaxation de la contrainte d'intégrité des variables. Pour ce faire, nous utiliserons le solveur GLPK.

2) Écrivez une fonction permettant de résoudre le programme linéaire initial en variables réelles en utilisant GLPK. Vérifiez que vous obtenez bien le résultat suivant :

```
Resolution en variables reelles :
x1 = 1.5, x2 = 0, x3 = 0
Utilite totale : 15
```

L'utilité totale ainsi obtenue correspond à l'évaluation de la racine de l'arborescence de séparation et évaluation. Ce calcul, qui servira pour l'évaluation de chaque sommet de l'arborescence durant l'application de la méthode de séparation et évaluation, génère une solution qui peut être exacte ou non. Elle est exacte si toutes les variables sont entières, ce qu'il faut donc vérifier.

Vérification de l'obtention d'une solution à variables entières

Une solution est à variables entières si tous ses x_i sont des entiers. Comme la résolution en variables réelles, telle que son nom l'indique, attribue des valeurs réelles à des variables de type réel, il faut déterminer si ces variables représentent en fait des valeurs entières ou non. Dans l'application de la méthode par séparation et évaluation, ceci permettra de définir si l'évaluation est exacte ou non.

3) Écrivez une fonction qui, à partir d'une solution en variables réelles, détermine si les valeurs des x_i sont tous des entiers ou non. Pour ce faire, une bonne idée est d'utiliser la fonction `modf` de la bibliothèque `math.h` pour vérifier si la partie fractionnaire de la valeur donnée en paramètre est différente de 0. Après application à la solution obtenue à la question précédente, vous vous en douterez, vous devriez déterminer que la solution n'est pas à variables entières.

Méthode par séparation et évaluation

Maintenant que nous savons calculer une borne, évaluer un sommet et déterminer si une évaluation est exacte ou non, il reste à programmer la méthode par séparation et évaluation. Le pseudo-code suivant donne un aperçu de la méthode en utilisant une stratégie de développement en profondeur et en supposant que la borne initiale a été calculée au préalable :

```
Évaluer le sommet par la fonction d'évaluation
Si l'évaluation obtenue est plus grande que la borne
  Vérifier si la solution obtenue est en variables entières
  Si la solution est en variables entières (évaluation exacte)
    Mettre à jour la borne avec la valeur de la fonction objectif
  Sinon
    Sélectionner une des variables de décision non encore sélectionnées
    Pour chaque valeur possible de cette variable
      Separation-Evaluation
```

4) Écrivez une fonction implémentant une méthode par séparation et évaluation pour le problème du sac à dos (bien qu'il soit possible de la développer de façon récursive, il est fortement conseillé de le faire de façon itérative). Vous afficherez ensuite la solution optimale du problème, qui devrait être la même que celle obtenue en cours.

5) Comparez les solutions et les temps d'exécution obtenus sur les problèmes **prob_sac_x.txt** disponibles sur le site Web du professeur (http://cosy.univ-reims.fr/~pdelisle/enseignement.php).