

TP 6

14 mai 2012

L'objectif de ce tp est l'implémentation de fonctions permettant de traiter de manière relativement générique les problèmes d'optimisation linéaire de première ou de seconde espèce.

1 Retour sur la méthode du simplexe**Exercice 1**

On considère un problème de **première espèce** générique sous forme matricielle initiale :

$$\begin{aligned} &\text{Maximiser } Z(x) = c^T x, \\ &\text{sous les contraintes} \\ &\quad \begin{cases} Ax \leq d, \\ x \geq 0 \end{cases} \end{aligned} \tag{1}$$

1. Créer une fonction **STANDARD1** qui, à partir de la seule donnée de A, d, c construit le tableau final M associé à la forme matricielle standard (une matrice sous **scilab**) . La ligne de M associée aux coûts marginaux de Z doit figurer en dernière position. Le prototype de la fonction est le suivant :

```
function M=STANDARD1(A,d,c)
...
endfunction
```

2. Créer une fonction **DANTZIG1** qui, à partir du tableau M incluant second membre de coûts marginaux, sélectionne le pivot par application des critères de Dantzig. Le prototype de la fonction est le suivant :

```
function Index=DANTZIG1(A,d,c)
...
endfunction
```

3. Définir en utilisant les deux fonctions précédentes **STANDARD1** et **DANTZIG1** une fonction **SIMPLEXE1** qui prend pour arguments les vecteurs A, d, c associés au problème initial, et fournit la première **solution de base optimale** rencontrée par application de la méthode du simplexe. Le prototype de la fonction est le suivant :

```
function M=SIMPLEXE1(A,d,c,NmaxIter)
...
endfunction
```

La variable **NmaxIter** permet de contrôler le nombre d'itérations.

4. Faire quelques tests en générant des matrices de contraintes et des fonctions objectif à l'aide de la fonction **rand**.

Exercice 2

On considère maintenant un problème de **second espèce**.

1. Créer une fonction **STANDARD2** qui, à partir de la seule donnée de A, d, c, ineg construit le tableau final M associé à la forme matricielle standard (pour la première phase de la méthode simplexe). Le vecteur ineg code le sens des inégalités. On peut supposer que le vecteur ineg est codé de la manière suivante : ineg a la taille du nombre de contraintes, et $\text{ineg}(i) = 1$ si la contrainte est de la forme $A_i^T x \leq d_i$ et $\text{ineg}(i) = -1$ si $A_i^T x \geq d_i$. La ligne de M associée aux coûts marginaux de Z doit figurer en dernière position. En avant dernière ligne, vous écrirez le vecteur de coûts marginaux **actualisés** associés au problème auxiliaire de recherche de première solution de base réalisable. Le prototype de la fonction est le suivant :

```
function M=STANDARD2(A,d,c)
...
endfunction
```

2. Modifier la fonction **DANTZIG1** afin qu'elle puisse être utilisée pour la première phase du simplexe. On pourra par exemple demander un argument supplémentaire indiquant les vecteurs de coûts marginaux à considérer.
3. Définir en utilisant les deux fonctions précédentes **STANDARD2** et **DANTZIG2** une fonction **SIMPLEXE2** qui prend pour arguments les vecteurs A, d, c, ineg associés au problème initial, et fournit la première **solution de base optimale** rencontrée par application de la méthode du simplexe. Dans le cas, où le problème n'est pas réalisable, on pourra retourner un message à l'écran pour le signifier et arrêter la méthode du simplexe. Faire de même pour le cas où le problème n'est pas borné. Le prototype de la fonction est le suivant :

```
function M=SIMPLEXE2(A,d,c,NmaxIter)
...
endfunction
```

La variable **NmaxIter** permet de contrôler le nombre d'itérations.

4. Faire quelques tests en générant des matrices de contraintes et des fonctions objectif à l'aide de la fonction **rand**.

Exercice 3 (Facultatif)

On peut s'intéresser à des variantes des fonctions précédentes comme des fonctions implémentant le critère de Bland ou d'autres critères vus dans les TP précédents. On pourra donner directement une solution optimale sous forme de vecteurs. On peut également définir et résoudre le problème dual du problème initial.