TP n°7 : Méthode du simplexe pour les problèmes de deuxième espèce

TP noté

Objectif : Dans cette séance, on s'intéressera à la détermination d'une base réalisable pour la résolution des problèmes de deuxième espèce.

1 Rappel : représentation en tableau et pivot

On s'intéresse au problème d'optimisation linéaire ($\mathscr{P}$) suivant, écrit sous forme standard :

$$\begin{array}{lrl}
\text{Maximiser} & z = & x_1 + 2\,x_2 \\
\text{sous les contraintes} & & \begin{array}{rcrcrcrcrcr}
x_1 & - & x_2 & + & x_3 & & & & & = & -2 \\
-3\,x_1 & + & x_2 & & & + & x_4 & & & = & 2 \\
x_1 & + & x_2 & & & & & + & x_5 & = & 6 \\
x_1 & , & x_2 & , & x_3 & , & x_4 & , & x_5 & \geqslant & 0
\end{array}
\end{array}$$

On définit la matrice et les vecteurs suivants :

$$A = \begin{pmatrix} 1 & -1 & 1 & 0 & 0 \\ -3 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 & 1 \end{pmatrix}, \qquad b = \begin{pmatrix} -2 \\ 2 \\ 6 \end{pmatrix} \qquad \text{et} \qquad C = \begin{pmatrix} 1 \\ 2 \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

On rappelle que le problème ($\mathscr{P}$) peut être représenté par la matrice augmentée suivante

$$M = \begin{bmatrix} A & b \\ {}^{t}C & 0 \end{bmatrix} \tag{1}$$

On dit que le problème est de *deuxième espèce* si le vecteur b possède au moins une composante strictement négative.

On donne également ci-dessous une implémentation possible de la fonction `pivot` :

```python
# la fonction pivot
def pivot(M,i,j):
    # M est une np.array (2 dimensions)
    # i indice de la ligne
    # j indice de la colonne
    N = M.copy()
    l,c = np.shape(N)
    for k in np.arange(l):
        if k==i : # on est sur la ligne de i
            N[k,:] = M[k,:]/M[k,j] # on normalise la ligne pour avoir 1 en (i,j)
        else:
            # on soustrait aux lignes pour annuler au dessus et en dessous du pivot
            N[k,:]=M[k,:] - M[k,j]/M[i,j]*M[i,:]
    return N
```

La méthode du simplexe utilise récursivement cette fonction sur la matrice M. À l'itération courante, elle est de la forme

$$M = \begin{bmatrix} A' & b' \\ {}^{t}d & -z \end{bmatrix}$$

avec A' de la même taille que A, b' de la même taille que b, d de la même taille que C et z un réel.

2 Détermination d'une base réalisable

Contrairement aux problèmes de première espèce, où la base pour laquelle les variables d'écart sont en base donne directement une base réalisable pour le problème, pour les problèmes de deuxième espèce, cette base n'est pas réalisable. Il faut donc passer par la résolution d'un problème auxiliaire pour obtenir une première base réalisable qui permettra d'initialiser la méthode du simplexe.

Plus précisément, on va s'intéresser au problème d'optimisation linéaire auxiliaire $(\mathscr{A})$ suivant :

$$
\begin{array}{lrcl}
\text{Maximiser} & -x_6 & & \\
\text{sous les contraintes} & x_1 - x_2 + x_3 - x_6 & = & -2 \\
 & -3\,x_1 + x_2 + x_4 - x_6 & = & 2 \\
 & x_1 + x_2 + x_5 - x_6 & = & 6 \\
 & x_1\ ,\ x_2\ ,\ x_3\ ,\ x_4\ ,\ x_5\ ,\ x_6 & \geqslant & 0
\end{array}
$$

dans lequel on a ajouté une variable supplémentaire.

Exercice 1 Représenter le problème $(\mathscr{A})$ à l'aide d'une matrice M'.

On rappelle qu'une base réalisable pour le problème auxiliaire s'obtient en considérant les variables d'écart, à l'exception de celle apparaissant sur la ligne où la constante b_i est la plus petite. Ainsi, pour écrire le problème auxiliaire sous forme réduite relativement à une base réalisable, il suffit de faire sortir de la base (non réalisable) constituée des variables d'écart, la variable associée à la constante la plus faible, et faire entrer dans la base la variable supplémentaire ajoutée dans le problème auxiliaire.

Exercice 2 Écrire une fonction `choix_auxilaire` qui prend en entrée une matrice M' et qui renvoie la colonne j associée à la variable sortante et la ligne i associée à la variable entrante. Tester cette fonction sur la matrice M' construite à l'exercice précédent.

Exercice 3 Écrire une fonction `pivot_auxilaire` qui prend en entrée une matrice M' et qui renvoie la matrice représentant le problème sous forme réduite relativement à la base réalisable décrite dans l'encart ci-dessus. Tester cette fonction sur la matrice M' construite à l'exercice 1. Vérifier que la base est bien réalisable.

Une fois la méthode du simplexe initialisée pour le problème auxiliaire, on peut l'itérer pour le résoudre.

1. Si la valeur optimale atteinte pour ce problème est nulle
2. et que la base correspondante ne comprend pas la variable supplémentaire,

alors cette base est réalisable pour le problème initial.

Exercice 4 Écrire une fonction `resolution_auxilaire` qui prend en entrée une matrice M' et qui résout par la méthode du simplexe le problème auxiliaire, puis renvoie 1 si la base obtenue est une base réalisable pour le problème initial, et 0 sinon (suivant les deux critères rappelés dans l'encart précédent). Tester cette fonction sur la matrice M' construite à l'exercice 1. *Pour vérifier que la variable supplémentaire est hors base, on pourra vérifier si le prix marginal associé est non nul. Attention, ce critère n'est pas nécessaire, mais on admettra qu'il suffit pour les cas considérés dans ce TP.*

On considère la fonction suivante :

```python
# la fonction mystère
def mystere(M):
    L,c = np.shape(M)
    gamma = -np.ones(L-1)
    for j in np.arange(c-1):
        if abs(M[L-1,j]) < 10**(-8):
            test = -1
            for i in np.arange(L-1):
                if abs(M[i,j]) > 10**(-8):
```

```
                    if test==-1:
                        test = i
                    else:
                        test = L
            if test!=L and test != -1 and abs(M[test,j]-1) < 10**(-8):
                gamma[test]=j
    return gamma
```

Exercice 5 Que fait cette fonction lorsqu'elle est appliquée à la représentation d'un problème sous forme réduite ? Tester sur la matrice M' et celles obtenues respectivement dans les exercices 3 et 4.

3 Résolution complète d'un problème de deuxième espèce

On revient au problème initial ($\mathscr{P}$).

Exercice 6 Représenter le problème ($\mathscr{P}$) à l'aide d'une matrice M.

Exercice 7 Écrire une fonction `representation_auxilaire` qui prend en entrée une matrice M représentant un problème de deuxième espèce sous forme standard et qui renvoie une matrice représentant le problème auxiliaire associé. Tester cette fonction sur la matrice M construite à l'exercice précédent.

On rappelle que, pour obtenir le problème initial sous forme réduite relativement à la base réalisable obtenue en résolvant le problème auxiliaire, il suffit

1. de conserver les contraintes du problème auxiliaire dans leur état final (c'est-à-dire, lorsque la méthode du simplexe a terminé), mais en supprimant la colonne correspondant à la variable supplémentaire ;
2. de calculer le vecteur des prix marginaux pour le problème initial, relativement à cette même base. Pour cela, on rappelle qu'il suffit de réaliser des pivots pour annuler les variables en base.

Exercice 8 Utiliser la fonction `mystere` pour écrire une fonction `initialisation_simplexe` qui prend en entrée une matrice M représentant un problème de deuxième espèce sous forme standard et qui renvoie la représentation de ce problème sous forme réduite relativement à la base réalisable obtenue à l'issue de la résolution du problème auxiliaire. Tester cette fonction sur la matrice M construite à l'exercice 6.

Exercice 9 Écrire une fonction `methode_simplexe_deuxieme_espece` qui prend en entrée une matrice M représentant un problème de deuxième espèce sous forme standard et qui renvoie une solution de base optimale obtenue en appliquant la méthode du simplexe sur le problème initial, initialisée à l'aide de la fonction `initialisation_simplexe`. Tester cette fonction sur la matrice M construite à l'exercice 6.

4 La méthode du simplexe dans le cas général

Dans cette section, on cherche à écrire une fonction permettant de résoudre tout type de problème d'optimisation linéaire (que ce soit un problème de première ou de deuxième espèce). On considère le problème sous forme canonique suivant :

$$\begin{array}{llrcrcr}
\text{Maximiser} & z = & x_1 & + & 2\,x_2 & & \\
\text{sous les contraintes} & & x_1 & - & x_2 & \leqslant & -2 \\
 & & -3\,x_1 & + & x_2 & \leqslant & 2 \\
 & & x_1 & + & x_2 & \leqslant & 6 \\
 & & x_1 & , & x_2 & \geqslant & 0
\end{array}$$

On définit la matrice et les vecteurs suivants :

$$a = \begin{pmatrix} 1 & -1 \\ -3 & 1 \\ 1 & 1 \end{pmatrix}, \qquad b = \begin{pmatrix} -2 \\ 2 \\ 6 \end{pmatrix} \qquad \text{et} \qquad c = \begin{pmatrix} 1 \\ 2 \end{pmatrix}$$

Exercice 10 Écrire une fonction `test_espece` qui prend en entrée un vecteur b et qui renvoie 1 si le problème sous forme canonique est de première espèce, 2 sinon. Tester cette fonction sur le vecteur b défini ci-dessus.

Exercice 11 Écrire une fonction `representation_probleme` qui prend en entrée une matrice a et deux vecteurs b et c, et qui renvoie la représentation du problème $(\mathscr{P})$ sous forme standard. Tester cette fonction sur la matrice a et les vecteurs b et c définis ci-dessus.

Exercice 12 Écrire une fonction `methode_simplexe` qui prend en entrée une matrice a et deux vecteurs b et c, et qui renvoie une solution optimale du problème $(\mathscr{P})$ **sous forme canonique** en appliquant la méthode du simplexe, soit directement sur la matrice augmentée si le problème est de première espèce, soit en commençant par chercher une première base réalisable.