Feuille de TP n°4
Introduction à la méthode du simplexe

Objectif : Préparer l'implémentation de la méthode du simplexe sous Python.

1 Écriture en tableau

On considère le problème sous forme **canonique** suivant :

$$\begin{array}{lrcl} \text{Maximiser} & z = & x_1 + 2\,x_2 & \\ \text{sous les contraintes} & & 3\,x_1 + 2\,x_2 & \leq 3 \\ & & 5\,x_1 + x_2 & \leq 1 \\ & & x_1 + 2\,x_2 & \leq 4 \\ & & x_1 + 4\,x_2 & \leq 5 \\ & & x_1 \;,\quad x_2 & \geq 0 \end{array}$$

dont la forme **standard** est donnée par

$$\begin{array}{lrl} \text{Maximiser} & z = & x_1 + 2\,x_2 \\ \text{sous les contraintes} & & 3\,x_1 + 2\,x_2 + x_3 = 3 \\ & & 5\,x_1 + x_2 + x_4 + = 1 \\ & & x_1 + 2\,x_2 + x_5 = 4 \\ & & x_1 + 4\,x_2 + x_6 = 5 \\ & & x_1 \;,\; x_2 \;,\; x_3 \;,\; x_4 \;,\; x_5 \;,\; x_6 \geq 0 \end{array}$$

On pose

$$M = \begin{pmatrix} 3 & 2 \\ 5 & 1 \\ 1 & 2 \\ 1 & 4 \end{pmatrix}, \qquad A = \begin{pmatrix} M & I_4 \end{pmatrix}, \qquad b = \begin{pmatrix} 3 \\ 1 \\ 4 \\ 5 \end{pmatrix}, \qquad c = \begin{pmatrix} 1 \\ 2 \end{pmatrix} \qquad \text{et} \qquad \tilde{c} = \begin{pmatrix} {}^{\mathrm{t}}c & 0_4 \end{pmatrix}$$

où I_4 désigne l'identité de $\mathbb{R}^4$ et 0_4 l'élément nul de $\mathbb{R}^4$. On représente le problème d'optimisation linéaire introduit plus haut à l'aide de la matrice augmentée suivante :

$$T = \begin{bmatrix} A & b \\ \tilde{c} & 0 \end{bmatrix}$$

Exercice 1 – Écriture en tableau

(a) Définir en Python les matrices M et A et les vecteurs b, c et $\tilde{c}$.

(b) Écrire une fonction `tableau()` qui prend en argument une matrice M de taille $m \times p$, un vecteur b de taille m et un vecteur c de taille p, et qui renvoie T la matrice augmentée définie plus haut. De quelle taille est T ?

On rappelle qu'un problème sous forme **standard** est dit **sous forme réduite** par rapport à une base γ si les vecteurs $\{A_{\gamma(i)}\}_{1\leq i\leq m}$ forment la base canonique de $\mathbb{R}^m$ et que les coefficients $\tilde{c}_{\gamma(i)}$ sont nuls. Dans ce cas, les éléments de $\tilde{c}$ sont les prix marginaux associés à la base γ.

Exercice 2 – Lecture d'un tableau On considère la matrice augmentée T suivante :

$$T = \begin{bmatrix} 2 & 2 & 1 & 0 & 8 & 0 & 8 \\ -1 & 1 & 0 & 1 & 2 & 0 & 2 \\ 0 & 4 & 0 & 0 & -1 & 1 & 3 \\ 0 & 8 & 0 & 0 & 2 & 0 & 0 \end{bmatrix}$$

(a) Vérifier que T représente bien un problème sous forme réduite. Quelle est la base associée ? Définir le vecteur $\gamma \in \mathbb{R}^3$ tel que

$$\forall\, i = 1, \dots, 3, \qquad A_{\gamma(i)} = e_i$$

Remarque : de cette manière, γ n'est plus nécessairement croissant !

(b) Définir en Python la matrice T.

(c) Écrire une fonction `tableauA()` qui prend en argument une matrice augmentée T de taille $(m+1)\times(n+1)$ et qui renvoie la matrice A associée constituée des m premières lignes des n premières colonnes de T.

(d) Écrire une fonction `tableaub()` qui prend en argument une matrice augmentée T de taille $(m+1)\times(n+1)$ et qui renvoie la matrice b associée constituée des m premières lignes de la dernière colonne de T.

(e) Écrire une fonction `tableauc()` qui prend en argument une matrice augmentée T de taille $(m+1)\times(n+1)$ et qui renvoie la matrice $\tilde{c}$ associée constituée des n premières colonnes de la dernière ligne de T.

2 Critère de Dantzig, naturel et de Bland

On rappelle le critère de Dantzig pour le choix d'un pivot :

- variable entrant dans la base / colonne dans la matrice augmentée T : une variable x_j / colonne j correspondant à un prix marginal d_j strictement positif ;
- variable sortant de la base / ligne dans T : une variable x_i / ligne i tel que le coefficient $a_{i,j}$ soit strictement positif et que le rapport $b_i/a_{i,j}$ soit minimal.

On rappelle également que ce critère n'est valable que pour des matrices T représentant un problème sous forme réduite par rapport à une base **réalisable**.

Exercice 3 – Test d'optimalité Écrire une fonction `testOptimalite()` qui prend en argument une matrice T de taille $(m+1)\times(n+1)$ **sous forme réduite** et qui renvoie 0 s'il existe au moins un prix marginal strictement positif et 1 sinon.

Exercice 4 – Critère de Dantzig

(a) Écrire une fonction `variableEntrante()` qui prend en argument une matrice T de taille $(m+1)\times(n+1)$ **sous forme réduite** et qui renvoie les indices des colonnes qui satisfont le critère de Dantzig. Tester cette fonction sur les deux matrices T des exercices précédents.

(b) Écrire une fonction `variableSortante()` qui prend en argument une matrice T de taille $(m+1) \times (n+1)$ et un indice de colonne j et qui renvoie les indices des lignes qui satisfont le critère de DANTZIG. Tester cette fonction sur les deux matrices T des exercices précédents et les indices obtenus à la question 1.

(c) Combiner les deux fonctions précédentes pour écrire une fonction `critereDantzig()` qui prend en argument une matrice T de taille $(m+1) \times (n+1)$ **sous forme réduite** et qui renvoie les indices (i, j) des pivots qui satisfont le critère de DANTZIG. Tester cette fonction sur les deux matrices T des exercices précédents.

On voit que le critère de DANTZIG fournit en général plusieurs pivots (qui sont tous acceptables). Cependant, dans le but de faciliter l'implémentation de la méthode du simplexe, on va utiliser des critères plus restrictifs qui permettent de réduire le choix du pivot à une seule possibilité. Dans le cours, on a vu deux tels critères :

- critère naturel : parmi les colonnes retenues par le critère de DANTZIG, on choisit la première associée à un prix marginal maximal ; parmi les lignes retenues par le critère de DANTZIG, on choisit la première ligne ;
- critère de BLAND : parmi les colonnes retenues par le critère de DANTZIG, on choisit la première colonne ; parmi les lignes retenues par le critère de DANTZIG, on choisit la première ligne.

Exercice 5 – Critère naturel et critère de BLAND

(a) Écrire une fonction `critereNaturel()` qui prend en argument une matrice T de taille $(m+1) \times (n+1)$ et qui renvoie l'indice (i, j) du pivot qui satisfait le critère naturel. Tester cette fonction sur les deux matrices T des exercices précédents.

(b) Écrire une fonction `critereBland()` qui prend en argument une matrice T de taille $(m+1) \times (n+1)$ et qui renvoie l'indice (i, j) du pivot qui satisfait le critère de BLAND. Tester cette fonction sur les deux matrices T des exercices précédents.

Exercice 6 – Mise-à-jour de la base

On reprend la matrice T de l'**Exercice 1**.

(a) Quelle est la base réalisable associée à T ? La définir sous Python à l'aide d'un vecteur γ.

(b) Appliquer la fonction `critereNaturel()` à la matrice T. Quelle est la nouvelle base que l'on obtient ?

(c) Écrire une fonction `majBase()` qui prend une base γ définie à l'aide du vecteur $(\gamma(1), \dots, \gamma(m))$ et un indice (i, j) et qui remplace dans la base γ la valeur $\gamma(i)$ par j. Tester cette fonction la base de la première question et pour (i, j) donné par le critère naturel, puis comparer avec la nouvelle base. Qu'en déduisez-vous ?