République Algérienne Démocratique et Populaire
Ministère de l'enseignement Supérieur et de la Recherche Scientifique
Université Benyoucef Benkhdda Alger 1

Faculté de Sciences

Département Mathématique Informatique

Projet

Module Programmation linéaire

THÈME :

Algorithme révisé du simplexe et programme linéaire a variables bornées

- **Réalisé par :** DRICI Mohamed Islam
MISSARA Razine Merouane

Mars 2021

Table des matières

Introduction Générale **3**

1 Méthode du simplexe révisé **4**
1.1 Introduction 4
1.2 Histoire 4
1.3 Les principes de la méthode 5
1.4 L'algorithme du simplexe révisé 5
1.5 Exemple d'application 6
1.6 L'intérêt de la méthode du simplexe révisé 7

2 Simplexe en variables bornées **8**
2.1 introduction 8
2.2 Algorithme du simplexe en variables bornées 9
2.3 Exemple d'application 11
2.4 L'intérêt du simplexe en variables bornée 12

3 Implémentation sur langage de programmation **13**
3.1 Introduction 13
3.2 Programme simplex révisé 13
3.3 Programme méthode du simplexe en variables bornées 14

Conclusion **15**

Bibliographie **16**

Introduction Générale

La Recherche Opérationnelle est l'ensemble des méthodes rationnelles d'analyse <mathématique,algorithmique,modélisation> et de synthèse des phénomènes utilisable pour l'élaboration d'une bonne décision.
La modélisation est l'une des méthodes de la recherche opérationnelle,elle contient pas mal de type comme :Processus stochastique,Graphes,Programmation linéaire.

Selon **William j.Baumaul** la programmation linéaire est une technique mathématique d'optimisation (Maximisation ou Minimisation) de fonction objectif linéaire sous des contraintes ayant la forme d'inéquation linéaire.Elle vise a sélectionner parmi différentes actions celle qui atteindra le plus probablement l'objectif visé.

Ce Projet représente deux algorithmes connues et particulièrement intéressants,ils s'agit de l'algorithme révisé du simplex et simplex en variables bornées.
L'algorithme révisé du simplex n'apporte aucune modification a l'algorithme du simplex,c'est sa mise en œuvre qui a été révisée après un examen attentif des opérations réellement nécessaire dans le simplex. Alors que la méthode du simplexe en variables bornées pour la résolution de problèmes linéaires est une modification de l'algorithme du simplexe« méthode principale de résolution de problèmes linéaires».

Ce manuscrit a été développé en trois chapitres :

Chapitre 1 est consacré a la définition de la méthode **révisé du simplexe**, son algorithme et l'intérêt de ce dernier, nous allons utiliser un exemple pour mieux comprendre.
Le chapitre 2 est dédier a la définition de la méthode de résolution de **Programme linéaire en variables bornées** .
Dans le troisième chapitre,on parlera de l'implémentation de ces algorithmes a travers des programmes (sous matlab) en les appliquant a un exemple.

Chapitre 1

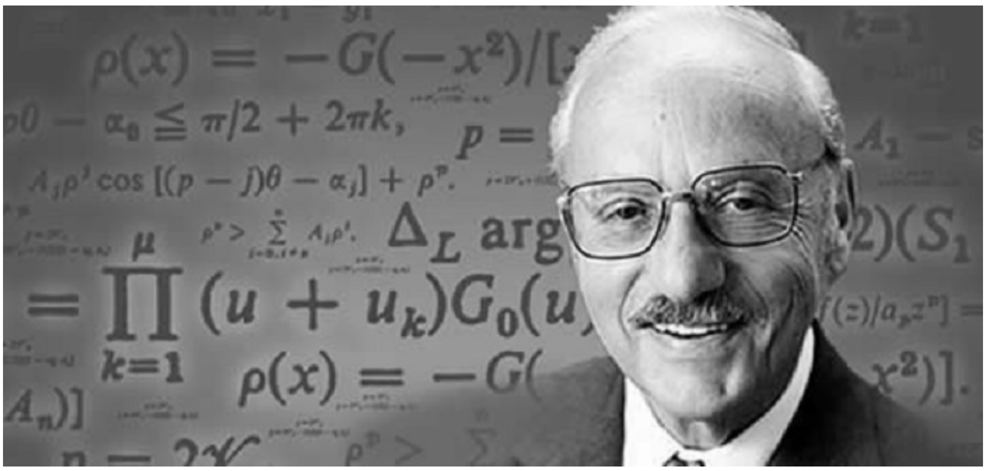
Méthode du simplexe révisé

1.1 Introduction

Plusieurs problèmes au courant de notre vie,peuvent se modélisés a des problèmes mathématique,ces derniers seront identifiés comme des programmes linéaires.
Dans la plus part des problèmes réels,on a plus que deux variables à déterminer.Une procédure algébrique pour résoudre les programmes linéaires avec plus que deux variables,c'est la méthode du simplexe.
Le concept de cette dernière consiste a effectuer un pivotage pour passer d'une base réalisable a une autre base réalisable afin d'atteindre une solution optimale (si elle existe),et d'améliorer la valeur de la fonction objective.
Plusieurs opérations de cette méthode s'avèrent inefficaces comparé a la méthode du simplexe révisé.
Ce chapitre représente cette méthode de résolution , ses principes et l'algorithme appliqué.

1.2 Histoire

DANTZIG George B.
Avec l'invention de la méthode du simplexe en 1947, George B.Dantzig, mathématicien américain, est considéré comme l'un des pères de la programmation linéaire.



La méthode du simplexe révisé a été mise au point par ***Dantzig***,***Orden*** et ***Wolfe*** a la "Rand corporation" au début des années 1950. Des techniques similaires avaient été découvertes par d'autres mathématiciens précédemment,mais sans parvenir a attirer l'attention.

1.3 Les principes de la méthode

Les principes de la méthode " Révisé du simplexe " sont les mêmes que ceux de la méthode conventionnelle du simplexe.
La nouveauté provient du fait que la procédure révisé utilise avec les données originales, alors que la procédure conventionnelle du simplexe transforme ces coefficients au fur et a mesure des itérations.
Le simplexe révisé se présente sous deux formes différentes, selon que l'on ajoute des variables d'écart ou des variables artificielles, pour la première formulation, nous supposons qu'une matrice identité est présente après l'ajout éventuel de variables d'écart.
Si des variables artificielles sont ajoutées, on utilise la second formulation qui correspond a la méthode des deux phases.
Le passage par la phase(1) n'est pas obligatoire du moment ou la solution de base de départ existe déjà (1^{ere} formulation).
Le passage a la phase(2) nous permet l'application de l'algorithme révisé du simplexe.
A chaque itération de cet algorithme on calcule :
(1) π :Vecteur multiplicatif relatif a la base J.
(2) $\widehat{C}$:Couts réduits relatifs a la base J.
(3) $\widehat{b}$:Valeurs des variables de base dans la solution de base associée a la base J.
(4) α :Valeur de Z pour la solution de base associée a J.
et comme la matrice $\widehat{A}$ n'a pas d'importance dans la méthode du simplexe révisé on vas pas la calculée, par contre on a besoin juste de la colonne $\widehat{A^s}$
(où : s est l'indice de la variable entrante dans la nouvelle base).

1.4 L'algorithme du simplexe révisé

Soit (P) un programme linéaire dans sa forme générale :

$$(P) = \begin{cases} Ax \leq b \quad , \quad x \geq 0 \\ \quad Cx = Z(max) \end{cases}$$

La résolution par la méthode du simplexe révisé se déroule selon des étapes :

1. Écrire (P) sous forme standard.

2. Calculer π par :
$$\pi = C^J.(A^J)^{-1}$$

3. Calculer $\widehat{C}$ par :
$$\widehat{C} = C - \pi.A$$
choisir s telle que $\widehat{C^s} > 0$ (Problème de maximisation)
   - Si s n'existe pas , Terminer.
     J base optimale.
     $x^*_J = (A^J)^{-1}.b$, $x^*_{\bar{J}} = 0$
     $\alpha^* = \pi.b$
   - Sinon , Aller a la 4^{eme} étape.

4. Calculer $\widehat{A^s}$ par :
$$\widehat{A^s} = (A^J)^{-1}.A^s$$
Poser $I = \{\ i\ /\ \widehat{A^s_i} > 0\}$

- Si $I = \emptyset$ Terminer.
  (P) n'admet pas de solution optimale.
- Si $I \neq \emptyset$, Aller a la 5^{eme} étape.

5. Calculer $\widehat{b}$ par :
   $$\widehat{b} = (A^J)^{-1}.b$$
   choisir $r \in I$ tel que :
   $$\frac{\widehat{b_r}}{\widehat{A_r^s}} = Min\{ \frac{\widehat{b_i}}{\widehat{A_i^s}} \}$$
   Poser $J = (J \cup \{s\})/\{col(r)\}$, $col(r) = s$
   Aller a la 2^{eme} étape.

1.5 Exemple d'application

Soit (P) un programme linéaire :

$$(P) = \begin{cases} & & -2x_3 & +2x_4 & +x_5 & = & Z(max) \\ x_1 & & -2x_3 & +x_4 & +x_5 & = & 4 \\ & x_2 & +3x_3 & -x_4 & +2x_5 & = & 2 \end{cases} \quad x_j \geq 0 \ , j = 1, ..., 5$$

.

(P) est écrit sous forme canonique par rapport a la base $J = \{1, 2\}$.

itération 1 :

1. (P) S.F.Standard , $J = \{1, 2\}$ Base réalisable , $col(1) = 1$ et $col(2) = 2$.
2. $\pi = C^J.(A^J)^{-1} = (0, 0)$ car $A^J = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$
3. $\widehat{C} = C - \pi.A = C = (0, 0, -2, 2, 1)$
   $s = 4$
   $\bullet \widehat{C^s} = 2 > 0$
4. $\widehat{A^s} = (A^J)^{-1}.A^s$, $\widehat{A^4} = A^4 = \begin{pmatrix} 1 \\ -1 \end{pmatrix}$, $I = \{1\}$
   $\bullet$I $\neq \emptyset$
5. $\widehat{b} = (A^J)^{-1}.b = \begin{pmatrix} 4 \\ 2 \end{pmatrix}$, $r = 1$, $col(r) = 1$
   $J = \{4, 2\}$, $col(1) = 4$

itération 2 :

1. (P) S.F.Standard , $J = \{4, 2\}$ Base réalisable , $col(1) = 4$ et $col(2) = 2$.
2. $\pi = C^J.(A^J)^{-1} = (2, 0)$ car $(A^J)^{-1} = \begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix}$
3. $\widehat{C} = C - \pi.A = (0, 0, -2, 2, 1) - (2, 0).\begin{pmatrix} 1 & 0 & -2 & 1 & 1 \\ 0 & 1 & 3 & -1 & -2 \end{pmatrix}$
   $= (0, 0, -2, 2, 1) - (2, 0, -4, 2, 2)$
   $= (-2, 0, 2, 0, -1)$
   $s = 3$
   $\bullet \widehat{C^s} = 2 > 0$

$\widehat{A^s} = (A^J)^{-1}.A^s$, $\quad \widehat{A^3} = \begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix} . \begin{pmatrix} -2 \\ 3 \end{pmatrix} = \begin{pmatrix} -2 \\ 1 \end{pmatrix}$, $\quad I = \{2\}$

$\bullet I \neq \emptyset$

4. $\widehat{b} = (A^J)^{-1}.b = \begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix} . \begin{pmatrix} 4 \\ 2 \end{pmatrix} = \begin{pmatrix} 4 \\ 6 \end{pmatrix}$, $r = 2$, $col(r) = 2$

   $J = \{4, 3\}$, $col(2) = 3$

   itération 3 :

   1. (P) S.F.Standard , $J = \{4, 3\}$ Base réalisable , $col(1) = 4$ et $col(2) = 3$.
   2. $\pi = C^J.(A^J)^{-1} = (4, 2)$ car $(A^J)^{-1} = \begin{pmatrix} 3 & 2 \\ 1 & 1 \end{pmatrix}$
   3. $\widehat{C} = C - \pi.A = (0, 0, -2, 2, 1) - (4, 2). \begin{pmatrix} 1 & 0 & -2 & 1 & 1 \\ 0 & 1 & 3 & -1 & -2 \end{pmatrix}$

      $= (-4, -2, 0, 0, -7) \leq 0$

      Terminer.

      $J = \{4, 3\}$ base optimale

      $x^*_J = (A^J)^{-1}.b = \begin{pmatrix} 3 & 2 \\ 1 & 1 \end{pmatrix} . \begin{pmatrix} 4 \\ 2 \end{pmatrix} = \begin{pmatrix} 16 \\ 6 \end{pmatrix}$, $\quad x^*_{\overline{J}} = 0$

   La solution optimale est $x^* = (0, 0, 6, 16, 0)$
   La valeur optimale est $Z^* = \pi.b = 20$

1.6 L'intérêt de la méthode du simplexe révisé

L'intérêt de cette méthode est de permettre de travailler a chaque itération avec les donnés originales du problème, ceci permet d'éviter la propagation des erreurs d'arrondi.
Il en résulte un gain de place lors du stockage des coefficients dans la mémoire d'un ordinateur et par conséquent un gain de temps appréciable. D'autre part le nombre total d'itérations et généralement faible avec la méthode du simplexe révisé qu'avec la méthode originale du simplexe.

Chapitre 2

Simplexe en variables bornées

2.1 introduction

Méthode du simplexe en variables bornées :
Cette méthode permet de résoudre un problème de programmation linéaire à variables bornées, sans faire apparaître explicitement les contraintes de bornes, ceci par une simple modification de la procédure de pivotage. Appliqué sur des PL de la forme :

$$(P)\begin{cases} A\,x \;=\; b \\ C\,x \;= Z(max) \\ \alpha_i \leq x_i \leq \beta_i \qquad \beta_i > \alpha_i \qquad \forall\, i \;=\; 1,2,...,n. \end{cases}$$

α_i et β_i sont respectivement les bornes inférieure et supérieure de x_i .

Remarque 1 : On peut facilement se ramener à un PL (P') où les variables sont positives et bornées supérieurement en effectuant un changement de variables comme suit : x_i'=x_i-α_i
D'où le PL suivant :

$$(P')\begin{cases} A\,(x_i'+\alpha_i) \;=\; b \\ C\,(x_i') \;= Z(max) - C.\alpha_i \\ 0 \leq x_i' \leq \beta_i - \alpha_i \qquad \beta_i > \alpha_i \qquad \forall\, i \;=\; 1,2,...,n. \end{cases}$$

Ainsi, le problème devient, avec : $b' = b - A \times \alpha_i$

$$(P')\begin{cases} A\,(x_i'+\alpha_i) \;=\; b' \\ C\,(x_i') \;= Z(max) - C.\alpha_i \\ 0 \leq x_i' \leq \beta_i - \alpha_i \qquad \beta_i > \alpha_i \qquad \forall\, i \;=\; 1,2,...,n. \end{cases}$$

Donc tout PLVB peut être ramené à la forme suivante :

$$(P)\begin{cases} A\,x \;=\; b \\ C\,x \;= Z(max) \\ 0 \leq x_i \leq \beta_i \qquad \forall\, i \;=\; 1,2,...,n. \end{cases}$$

Remarque 2 : Une façon de résoudre ce problème est de le ramener sous une forme standard

en introduisant des variables d'écart y_i, comme suit :

$$(P)\begin{cases} A\,x\ =\ b \\ x_i + y_i = \beta_i \\ C\ x\ \ = Z(max) \\ x_i, y_i \geq 0 \qquad \forall\, i\ =\ 1,2,...,n. \end{cases}$$

Ce problème est un PL sous forme standard, ayant m+n contraintes et 2n variables il peut être résolu par la méthode usuelle du simplexe mais vu la taille du problème « très grande », sa résolution s'effectue en un grand nombre d'itérations et pour cela il existe une adaptation de la méthode du simplexe à ce type de problèmes qui traite les contraintes des bornes implicitement et avec un minimum d'itérations. Comme dans l'algorithme usuelle du simplexe, nous choisirons une variable x_j hors base (c'est-à-dire valant 0 ou β_j) et nous la ferons varier (si x_j=β_j ,x_j va diminuer et si x_j =0, x_j va augmenter) jusqu'à ce que l'un des trois événements suivants se produise :

— 1) x_j varie jusqu'à sa borne opposée.
— 2) l'une des variables de base décroit jusqu'à 0.
— 3) l'une des variables de base augmente jusqu'à sa borne sup β.

Notons que dans l'algorithme du simplexe, seul le cas (2) se produisait.

2.2 Algorithme du simplexe en variables bornées

Initialisation :

L'application de cet algorithme nécessite la connaissance d'une solution réalisable de base, au départ.
Pour J une base non optimale on prend comme variable hors base X_s candidate à devenir une variable de base celle qui vérifie : C_s=max { max(C_j) tq X_j=0 , max(-C_j) tq X_j= β_j } $\forall j \in \bar{J}$.
Pour trouver une solution de base on fixe chaque variable hors base soit à sa borne inférieure (ici 0) ou à sa borne supérieure (ici β_j).
Trouver une base réalisable en appliquant la 1^{ere} phase de l'algorithme du simplexe,Si le PLVB n'a pas de base réalisable, terminer .Sinon deux cas se présentent :

1er cas : { $C_s > 0\ et\ X_s = 0$, X_s doit croitre } :

On construit les ensembles I_1 et I_2 : $\forall X_{ij}\ \in A_{ij}$
$I_1 = \{i : A_i^s > 0\}$ et calculer :

$$\theta_1 = min\left\{\ \tfrac{U_i}{A_i^s}\ \right\} = \frac{U_r}{A_r^s}$$

$I_2 = \{i : A_i^s < 0\}$ et calculer :

$$\theta_2 = min\left\{\ \tfrac{\beta_{ij} - Ui}{-A_i^s}\ \right\} = \frac{\beta_{ij} - Ur}{-Ar^s}$$

« On effectue le minimum des rapport pour rester dans le domaine réalisable des bornes des variables ».
Et on calcule $\theta = min\{\theta_1,\ \theta_2,\ \beta_s\}$ « pour augmenter le plus possible la fonction objective ».

Si $\theta = \beta_s$ (pas de changement de base seule la solution change) :

X atteint sa borne supérieure elle ne rentre pas dans la base.
Pour se mettre dans les conditions habituels du simplexe on fait le changement de variables

suivant : $X'_s = \beta_s - X_s$, et on n'effectue pas de pivotage.

$$\begin{cases} X_s = \beta_s \\ X_{ij} = U_i - A_i^s.\theta \\ -Z = -Z - \hat{C}_s.\theta \end{cases}$$

Si $\theta = \theta_1$ changement de base :
La variable rentrante prend la valeur de θ.
La variable sortante prend sa borne inférieure (ici 0).

$$\begin{cases} X_s = 0 \\ X_{ij} = U_i - A_i^s.\theta \\ -Z = -Z - \hat{C}_s.\theta \end{cases}$$

Si $\theta = \theta_2$ changement de base :
La variable rentrante prend la valeur de θ.
La variable sortante prend sa borne supérieure (iciβ_s).

$$\begin{cases} X_s = \beta_s \\ X_{ij} = U_i - A_i^s.\theta \\ -Z = -Z - \hat{C}_s.\theta \end{cases}$$

2ème cas : $\{ C_s < 0$ *et* $X_s = \beta_s$, X_s doit décroitre $\}$:
On construit les ensembles I_1 et I_2 : $\forall X_{ij} \in A_{ij}$
$I_1 = \{i : A_i^s > 0\}$ et calculer :

$$\theta_1 = max\left\{ \ \frac{U_i}{A_i^s} \ \right\} = \frac{U_r}{A_r^s}$$

$I_2 = \{i : A_i^s < 0\}$ et calculer :

$$\theta_2 = max\left\{ \ \frac{\beta_{ij} - Ui}{-A_i^s} \ \right\} = \frac{\beta_{ij} - Ur}{-Ar^s}$$

On calcule $\theta = max\{\theta_1, \ \theta_2, \ \}$.
Si $\theta = 0$ (pas de changement de base seule la solution change) :

$$\begin{cases} X_s = 0 \\ X_{ij} = U_i - A_i^s.\theta \\ -Z = -Z - \hat{C}_s.\theta \end{cases}$$

Si $\theta = \theta_1$ changement de base :
La variable rentrante prend la valeur de θ.
La variable sortante prend sa borne inférieure (ici 0).

$$\begin{cases} X_s = 0 \\ X_{ij} = U_i - A_i^s.\theta \\ -Z = -Z - \hat{C}_s.\theta \end{cases}$$

Si $\theta = \theta_2$ changement de base : La variable rentrante prend la valeur de θ.
La variable sortante prend sa borne supérieure (iciβ_s).

$$\begin{cases} X_s = \beta_s \\ X_{ij} = U_i - A_i^s.\theta \\ -Z = -Z - \hat{C}_s.\theta \end{cases}$$

Réitérer jusqu'à l'obtention d'une base optimale vérifiant les critères d'optimalités su-cité .

2.3 Exemple d'application

Enoncé :

$$(PLVB)\begin{cases} 2x_1+4x_2+x_3=Z(max) \\ 2x_1+x_2+x_3\leq 10 \\ x_1+x_2-x_3\leq 4 \\ 0\leq x_1\leq 4 \\ 0\leq x_2\leq 6 \\ 0\leq x_3\leq 3 \end{cases}$$

Résolution :

En rajoutant les variables d'écarts $x_4,\ ,x_5\ ,x_i\geq 0\ \forall\ i=4,5$ on obtiens le tableau suivant :

$$\left[\begin{array}{c|ccccc|c} x_i & x_1 & x_2 & x_3 & x_4 & x_5 & b \\ \hline x_4 & 2 & 1 & 1 & 1 & 0 & 10 \\ \hline x_5 & 1 & 1 & -1 & 0 & 1 & 4 \\ \hline Z & 2 & 4 & 1 & 0 & 0 & 0 \end{array}\right] \tag{2.1}$$

Itération 1 : Pour se ramener à la forme initial du PL de l'algorithme ,on fixe les variables hors base à leurs bornes inférieure : $x_1=x_2=x_3=0$.

On obtiens le tableau suivant :

$$\left[\begin{array}{c|ccccc|c} x_i & x_1^I & x_2^I & x_3^I & x_4 & x_5 & b \\ \hline x_4 & 2 & 1 & 1 & 1 & 0 & 10 \\ \hline x_5 & 1 & 1 & -1 & 0 & 1 & 4 \\ \hline Z & 2 & 4 & 1 & 0 & 0 & 0 \end{array}\right] \tag{2.2}$$

NB :

I=borne inférieure.

S=borne supérieure.

$x_4\ et\ x_5$ sont les variables de base et ainsi la solution de base associée est : (0,0,0,10,4)

$c^s=max\{2,4,1\}=4$, s=2

$A_i^2>0$ on calcule θ_1 :

$\theta_1=min\{\frac{b_i^2}{A_i^2}\}=min\{10,4\}$=4.

$\theta=min\{\theta_1,S\}=min\{4,6\}=4=\theta_1$.

changement de base (x_2 rentre dans la base x_5 sort) :

$x_2=\theta_1=4$, $x_5=I=0$.

$\begin{bmatrix} x_4 \\ x_5 \end{bmatrix}=\begin{bmatrix} 10 \\ 4 \end{bmatrix}-A_i^2.\,\theta=\begin{bmatrix} 10 \\ 4 \end{bmatrix}-\begin{bmatrix} 1 \\ 1 \end{bmatrix}.4=\begin{bmatrix} 6 \\ 0 \end{bmatrix}$.

$-\hat{Z}$=-Z - $\hat{C}_2$ =-0-(4.4)=-16.

Itération 2 :

$$\left[\begin{array}{c|ccccc|c} x_i & x_1^I & x_2 & x_3^I & x_4 & x_5^I & b \\ \hline x_4 & 1 & 0 & 2 & 1 & -1 & 6 \\ \hline x_2 & 1 & 1 & -1 & 0 & 1 & 4 \\ \hline Z & -2 & 0 & 5 & 0 & -4 & -16 \end{array}\right] \tag{2.3}$$

La solution de base associée : (0,4,0,6,0)

$c^s=max\{-2,5,-4\}=5$, s=3

$A_4^3 > 0$ on calcule θ_1 : θ_1=3
$A_2^3 < 0$ on calcule θ_2 : θ_2 =$\frac{6-4}{1}$=2.
$\theta = min\{\theta_1, \theta_2, S\} = min\{3, 2, 3\} = 2 = \theta_2$.
changement de base (x_3 rentre dans la base x_2 sort) :
$x_3 = \theta = 2$, $x_2 = S = 6$.

$$\begin{bmatrix} x_4 \\ x_2 \end{bmatrix} = \begin{bmatrix} 6 \\ 4 \end{bmatrix} - A_i^3 .\, \theta = \begin{bmatrix} 6 \\ 4 \end{bmatrix} - \begin{bmatrix} 2 \\ -1 \end{bmatrix} .2 = \begin{bmatrix} 2 \\ 6 \end{bmatrix}.$$

$-\hat{Z}$=-Z - $\hat{C}_3$ =-16-(5.2)=-26.

Itération 3 :

$$\left[\begin{array}{c|ccccc|c} x_i & x_1^I & x_2^S & x_3 & x_4 & x_5^I & b \\ \hline x_4 & 3 & 2 & 0 & 1 & 1 & 2 \\ \hline x_3 & -1 & -1 & 1 & 0 & -1 & 2 \\ \hline Z & 3 & 5 & 0 & 0 & 1 & -26 \end{array}\right] \tag{2.4}$$

La solution de base associée : (0,6,2,2,0)
$c^s = max\{3, -5, 1\} = 3$, s=1
$A_4^1 > 0$ on calcule θ_1 : $\theta_1 = \frac{2}{3}$.
$A_3^1 < 0$ on calcule θ_2 : θ_2 =$\frac{3-2}{1}$=1.
$\theta = min\{\theta_1, \theta_2, S\} = min\{\frac{2}{3}, 1, 4\} = \frac{2}{3} = \theta_1$.
changement de base (x_1 rentre dans la base x_4 sort) :
$x_1 = \theta = \frac{2}{3}$, $x_4 = I = 0$.

$$\begin{bmatrix} x_4 \\ x_3 \end{bmatrix} = \begin{bmatrix} 2 \\ 2 \end{bmatrix} - A_i^1 .\, \theta = \begin{bmatrix} 2 \\ 2 \end{bmatrix} - \begin{bmatrix} 3 \\ -1 \end{bmatrix} .\frac{2}{3} = \begin{bmatrix} 0 \\ \frac{8}{3} \end{bmatrix}.$$

$-\hat{Z}$=-Z - $\hat{C}_3$ =-26-3.$\frac{2}{3}$=-28.

Itération 4 :

$$\left[\begin{array}{c|ccccc|c} x_i & x_1 & x_2^S & x_3 & x_4^I & x_5^I & b \\ \hline x_1 & 1 & \frac{2}{3} & 0 & \frac{1}{3} & \frac{1}{3} & \frac{2}{3} \\ \hline x_3 & 0 & -\frac{1}{3} & 1 & \frac{1}{3} & -\frac{2}{3} & \frac{8}{3} \\ \hline Z & 0 & 3 & 0 & -1 & 0 & -28 \end{array}\right] \tag{2.5}$$

La solution de base associée :($\frac{2}{3}$,6,$\frac{8}{3}$,0,0).
J=$\{1, 3\}$ est une base optimale ,les critères d'optimalités sont vérifiés :

$$\forall j \in \bar{J} \ : (C_j \leq 0 \ et \ X_j = 0) \ et \ (C_j \geq 0 \ et \ X_j = S_j)$$

($\hat{C}_2 = 3$, $x_2 = S = 6$) et ($\hat{C}_4 = -1$, $x_4 = I = 0$).

Remarque :

Il nous faudra 6 itérations par la méthode du simplexe habituelle appliquée sur cet exemple.

2.4 L'intérêt du simplexe en variables bornée

L'algorithme du simplexe à variables bornées permet de résoudre les programmes linéaires en un temps minimal et moins de calcules qu'avec le simplexe habituel, lorsqu'il s'agit de variables bornées .

Chapitre 3

Implémentation sur langage de programmation

3.1 Introduction

On traite dans ce chapitre la version programmée (application) de la méthode su-citée à titre explicative et on trouve aussi le code source de notre application élaborée sous Matlab.

3.2 Programme simplex révisé

Soit le programme (P) à résoudre :

$$(P) = \begin{cases} \text{-2x}_1 & +\text{x}_2 & \text{-3x}_3 & \text{-x}_4 & +\text{x}_5 & = & \text{Z(min)} \\ \text{x}_1 & +\text{x}_2 & +\text{2x}_3 & +\text{3x}_4 & \text{-2x}_5 & \leq & 4 \\ \text{-x}_1 & \text{-2x}_2 & +\text{3x}_3 & \text{-2x}_4 & +\text{3x}_5 & \leq & 10 \end{cases} \quad x_j \geq 0, j = 1, ..., 5$$

Tout d'abord , on doit saisir les données du problème dans la commande window de la manière suivante :

```
Fenêtre de Commandes
>> A=[1 1 2 3 -2 ; -1 -2 3 -2 3];
>> b=[4 ; 10];
>> c=[-2 1 3 -1 1];
>> inq=[-1 -1];
>> Type='0';
```

En deuxième étape , en fait appel à la fonction qui résoud le programme linéaire en écrivant le nom de la fonction suivi par les arguments d'entrées.
Et à la fin , on obtient le résultat suivant :

```
>> Revise_simplexe(A,b,c,inq,Type)
J est la base realisable de depart
La solution de base de départ
X =

     0
     0
     0
     0
     0
     4
    10

J est une base optimale
ans =

    32
     0
     0
     0
    14
     0
     0

>>
```

et

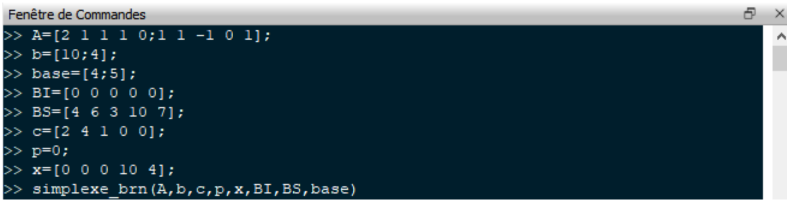
```
J =

     1     2
```

3.3 Programme méthode du simplexe en variables bornées

Ce programme résout les problèmes d'optimisation ayant une fonction objective à maximiser, si l'utilisateur souhaite introduire un PL à minimiser le vecteur coût (C) doit être multiplié par -1.
La matrice A est associée au programme linéaire sous sa forme standard.
Les bornes inférieures sont toujours égale à zéro .

$$(PLVB)\begin{cases} 2x_1 + 4x_2 + x_3 = Z(\max) \\ 2x_1 + x_2 + x_3 + x_4 = 10 \\ x_1 + x_2 - x_3 + x_5 = 4 \\ 0 \leq x_1 \leq 4 \\ 0 \leq x_2 \leq 6 \\ 0 \leq x_3 \leq 3 \\ 0 \leq x_4 \\ 0 \leq x_5 \end{cases}$$

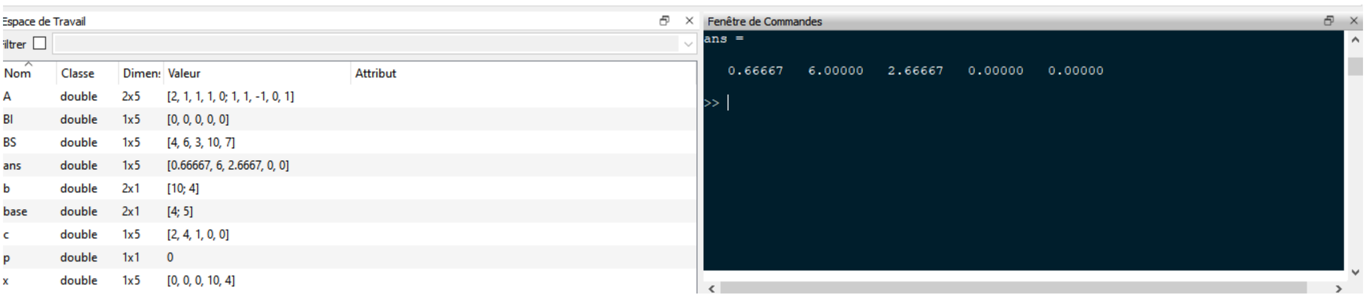
On doit saisir les données du problème dans la commande window puis en fait appel à la fonction qui résout le programme linéaire en écrivant le nom de la fonction suivi par les arguments d'entrées.



Fenêtre de Commandes

```
>> A=[2 1 1 1 0;1 1 -1 0 1];
>> b=[10;4];
>> base=[4;5];
>> BI=[0 0 0 0 0];
>> BS=[4 6 3 10 7];
>> c=[2 4 1 0 0];
>> p=0;
>> x=[0 0 0 10 4];
>> simplexe_brn(A,b,c,p,x,BI,BS,base)
```

Et à la fin , on obtient le résultat suivant :



Espace de Travail

Nom	Classe	Dimen:	Valeur	Attribut
A	double	2x5	[2, 1, 1, 1, 0; 1, 1, -1, 0, 1]	
BI	double	1x5	[0, 0, 0, 0, 0]	
BS	double	1x5	[4, 6, 3, 10, 7]	
ans	double	1x5	[0.66667, 6, 2.6667, 0, 0]	
b	double	2x1	[10; 4]	
base	double	2x1	[4; 5]	
c	double	1x5	[2, 4, 1, 0, 0]	
p	double	1x1	0	
x	double	1x5	[0, 0, 0, 10, 4]	

Fenêtre de Commandes

```
ans =

   0.66667   6.00000   2.66667   0.00000   0.00000

>>
```

Conclusion

Nous devons avouer que rétrospectivement, nous sommes satisfaits de ce projet puisque nous avons atteint de nouveaux objectifs.
En effet, ce mini projet nous a permis de comprendre et apprendre les principaux concepts de la programmation linéaire, Notamment la méthode du simplexe révisé et la méthode du simplexe à valeurs bornées, pour cela nous avons présenté à travers notre travail, les propriétés fondamentales de ces dernières ,sans oublier leur algorithmes.
Nous poursuivant notre étude par l'implémentation du révisé de simplexe sous le langage de programmation MATLAB.
Notre modeste recherche, nous à permis de voir la complexité et la richesse du sujet, tout en sachant que ce dernier est très important et efficace grâce à sa diversité dans son application dans le monde réel.

Bibliographie

[1] **M.Sakarovitch** .*Springer Texts in Electrical Engineering , Linear Programming* , Hermann. 1984.

[2] **Dantzig G.B** . *Maximization of a linear function of variables subject to linear inequalities*. In *Activity Analysis of Production and Allocation*.1951.

[3] **Dantzig G.B** , **Mukund N.Thapa** . *Linear Programming* . Peter Glynn-Stanford University. 1997.

[4] **M.Sakarovitch** .*Optimisation combinatoire-Graphes et programmation linéaire* , Hermann. 1984.

[5] **Nikoloas Ploskas , Nikolaos Samaras** . *Linear Programming User MATLAB*. Ding-Zhu Du , University of texas at Dallas.

[6] **Roy H.Kwon** . *Introduction to Linear Optimization and Extensions with MATLAB*. A.Ravi Ravindran . 2014.