Afrique SCIENCE 12(6) (2016) 341 - 352
ISSN 1813-548X, http://www.afriquescience.info

Sur la méthode du simplexe ordinaire

François NDAYIRAGIJE

Faculté des Sciences, Département de Mathématiques, Université du Burundi, BP 2700 Bujumbura, Burundi

* Correspondance, courriel : *ndayiragijefrancois@yahoo.fr*

Résumé

La méthode du simplexe ordinaire est venue pour résoudre des problèmes d'optimisation de fonctions économiques, en se déplaçant sur les côtés d'un polygone de contraintes ou sur les arêtes d'un polyèdre de contraintes. La résolution par la méthode du simplexe ordinaire étant algébrique, le but de cet article est de généraliser d'abord une Formule de résolution déjà existante, ensuite de donner une nouvelle formule générale dans le cas du pivotage en tableau du simplexe. Il a donc été question d'apprécier la fiabilité des résultats issus d'un calcul algébrique que nous avons vérifiés avec le logiciel TORA, un logiciel de référence en programmation linéaire.

Mots-clés : *simplexe, méthode ordinaire, logiciel TORA.*

Abstract

On the ordinary simplex method

The ordinary simplex method came to solve optimization problems of economic functions, moving to the sides of a polygon constraints or on the edges of a polyhedron constraints. As the computing by the ordinary simplex method is algebraic, the goal of this paper is first to generalize an existing computing formula and then to give a new general formula in the case of pivoting in simplex tableau. It was therefore a question of assessing the reliability of the results from an algebraic calculation which have been checked, using the TORA software, a reference software in linear programming.

Keywords : *simplex, ordinary method, TORA software.*

1. Introduction

La technique algébrique pour trouver la solution optimale d'un problème de programmation linéaire est appelée méthode du simplexe. Elle fut développée dans les années 1947 [1, 2, 3] par un jeune mathématicien américain du nom de George Bernard Dantzig. Cette méthode est applicable pour 2 variables mais peut être généralisée pour un nombre quelconque de variables. Elle peut être utilisée à la place de la méthode graphique de programmation linéaire. L'inconvénient de la méthode graphique est qu'elle n'est pas applicable pour plus de 2 variables [4] alors que ce n'est pas le cas pour la méthode du simplexe. Cette méthode du simplexe appelée méthode du simplexe ordinaire [4, p100] n'est qu'une application aveugle de la procédure algébrique. Elle n'est pas efficace au point de vue calcul sur ordinateur : elle calcule et stocke des nombres dont on a besoin à l'itération actuelle et qui ne sont peut être pas nécessaires à la prise de décision aux itérations suivantes. Les seules informations nécessaires à chaque itération sont : les coefficients des variables hors base de l'équation initiale, les coefficients de la variable de base entrant dans les ***Équations*** et les membres de droite des ***Équations***. Il existe plusieurs méthodes de résolution d'un problème de programmation linéaire. Parmi ces méthodes, il ya l'algorithme de la méthode du simplexe ordinaire en tableau. Dans leur recherche, plusieurs auteurs [1, 3, 5, 6] ont abouti à de bons résultats même si certaines méthodes ne sont pas compréhensibles aux débutants. C'est la raison pour laquelle nous avons donné des ***Formules*** générales permettant aux premiers utilisateurs de cet algorithme à comprendre directement les manipulations algébriques.

2. Méthodologie

2-1. Principe de la méthode du simplexe ordinaire

Soit,

$$\begin{cases} Max\ Z = CX \\ AX = b \\ X \geq 0 \end{cases} \tag{1}$$

un programme linéaire écrit sous sa forme standard. A est une matrice à m lignes et n colonnes, de rang m.

Soit J une base de A.

A $= [A^J,\ A^{\bar{J}}]$,

X $= \begin{bmatrix} x_J \\ x_{\bar{J}} \end{bmatrix}$,

C $= [c^J,\ c^{\bar{J}}]$,

où, A^J est la matrice relative aux variables de base, $A^{\bar{J}}$ est la matrice relative aux variables hors base, les x_J sont les variables de base et les $x_{\bar{J}}$ sont les variables hors base, les c^J sont les vecteurs coûts relatifs à la base J et les $c^{\bar{J}}$ sont les vecteurs coûts non relatifs à la base J.

Le système (1) devient

$$\begin{cases} Max\ Z = c^J x_J + c^{\bar{J}} x_{\bar{J}} \\ A^J x_J + A^{\bar{J}} x_{\bar{J}} = b \\ X \geq 0 \end{cases} \tag{2}$$

Le système (2) devient

$$\begin{cases} Max\ Z = c^J x_J + c^{\bar{J}} x_{\bar{J}} \\ x_J + (A^J)^{-1} A^{\bar{J}} x_{\bar{J}} = (A^J)^{-1} b \\ X \geq 0 \end{cases} \tag{3}$$

De simples calculs nous donnent

$$\text{Max } Z = c^J (A^J)^{-1} b + \left[c^{\bar{J}} - c^J (A^J)^{-1} A^{\bar{J}}\right] x_{\bar{J}} \tag{4}$$

Posant, $\Pi = c^J (A^J)^{-1}$ dans (4), nous avons

$$\text{Max } Z = \Pi \text{ b} + \left[c^{\bar{J}} - \Pi A^{\bar{J}}\right] \tag{5}$$

Finalement, le programme linéaire s'écrit :

$$\begin{cases} Max\ Z = \ \Pi \text{ b} + \left(c^{\bar{J}} - \Pi A^{\bar{J}}\right) \\ x_J + (A^J)^{-1} A^{\bar{J}} x_{\bar{J}} = (A^J)^{-1} b \\ \qquad X \geq 0 \end{cases} \tag{6}$$

Le programme linéaire (5) équivaut à

$$\begin{cases} Max\ Z = \ \Pi \text{ b} + \hat{\text{C}}\text{X} \\ \quad \hat{A}\, X = \hat{b} \\ \quad X \geq 0 \end{cases} \tag{7}$$

où,

$\hat{C} = \left(0, c^{\bar{J}} - \Pi A^{\bar{J}}\ \right) = C - \Pi A$ *est le vecteur des coûts réduits ; C est le vecteur des coûts ;*

$\hat{A} = \left[I_m\,, (A^J)^{-1} A^{\bar{J}}\right]$*;*

$\hat{b} = (A^J)^{-1} b$*;*

I_m *est une matrice identité d'ordre m ;*

$\Pi = \ c^J (A^J)^{-1}$ *est le vecteur multiplicatif relatif à la base J ou vecteur prix.*

Dès lors, le programme linéaire est écrit à la forme canonique par rapport à la base J. La solution de base associée est

$$\begin{cases} \quad x_{\bar{J}} = O \\ x_J = \ (A^J)^{-1} b \end{cases} \tag{8}$$

Si $\hat{C} \leq 0$ *alors J est une base optimale et Max Z*$= \Pi\ b$.

Dans le cas d'un programme linéaire

$$\begin{cases} Min\ Z = AX \\ \quad AX = b \\ \quad X \geq 0 \end{cases}$$

si $\hat{C} \geq 0$ *alors J est une base optimale et Min Z* $= \Pi\ b$.

Pour la maximisation (minimisation), la méthode du simplexe ordinaire est un algorithme de calcul d'une solution de base qui améliore (diminue) graduellement la valeur de la fonction économique tout en modifiant simultanément les variables de base de façon à toujours satisfaire le système d'équations jusqu'à ce qu'une de ces variables de base devienne nulle. Cela donne une nouvelle solution de base admissible où l'ex variable hors base remplace l'ex variable de base.

2-2. Algorithme du simplexe ordinaire

Pour le cas de la maximisation, si les coûts réduits sont négatifs ou nuls, alors x_J est optimal et la méthode du simplexe ordinaire se termine. Sinon la méthode continue. Pour le cas de minimisation, si les coûts réduits sont positifs ou nuls, alors x_J est optimal et la méthode du simplexe ordinaire se termine. Sinon la méthode continue. Pour les deux cas (maximisation et minimisation), si x_J n'est pas optimal, on choisit une nouvelle base. On choisit un indice h correspondant à une variable hors base ($x_h = 0$) pour laquelle le coût réduit est le plus positif (cas de la maximisation) ou le plus négatif (cas de la minimisation). S'il ya ex aequo, on procède au choix arbitraire. Donc, pour les deux cas, h sera l'indice de la nouvelle variable. [1] ont montré que pour le cas de la maximisation, la valeur de x_h est

$$\min\left\{\frac{x_i}{y_{ih}}, y_{ih} > 0\right\} = \frac{x_s}{y_{sh}} \tag{9}$$

où,

$$y_{ih} = (A^J)^{-1}A_h, \tag{10}$$

i prend les valeurs de la base J, A_h est la $h^{ième}$ colonne de la matrice A et x_s est la variable qui sort de la base. Dans cet article nous montrons que cette formule reste vraie aussi bien pour le cas de la maximisation que pour le cas de la minimisation. En effet, pour le cas de la maximisation (respectivement le cas de la minimisation), pour que la base J soit optimale, il faut que $\hat{C} \leq 0$ (respectivement $\hat{C} \geq 0$). Logiquement, on prendrait la valeur de x_h pour le cas de la minimisation :

$$\max\left\{\frac{x_i}{y_{ih}}, y_{ih} > 0\right\} = \frac{x_s}{y_{sh}}. \tag{11}$$

Mais, comme à la place de $\hat{C} \leq 0$ (cas de la maximisation) on prend $\hat{C} \geq 0$ (cas de la minimisation), la formule donnant la valeur de x_h pour les cas de la maximisation et de la minimisation reste la même : la ***Formule (9)***.

3. Résultats et discussion

3-1. Illustration pour le cas de la minimisation

Soit le programme linéaire suivant :

$$\begin{cases} Min\ Z = -x_1 - x_2 \\ sous\ 2x_1 + x_2 \leq 6 \\ \quad x_1 + 2x_2 \leq 6 \\ \quad\quad x_1, x_2 \geq 0 \end{cases} \tag{12}$$

A la forme standard, ce programme devient :

$$\begin{cases} Min\ Z = -x_1 - x_2 \\ sous\ 2x_1 + x_2 + x_3 = 6 \\ x_1 + 2x_2 + x_4 = 6 \\ x_1, x_2, x_3, x_4 \geq 0 \end{cases}$$

Posant J = $\{3,4\}$, A^J devient :

$A^J = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$ et donc $x_J = \begin{pmatrix} x_3 \\ x_4 \end{pmatrix} = \begin{pmatrix} 6 \\ 6 \end{pmatrix}$, $\Pi = (0,0)$ et par conséquent $\hat{C} = (-1,-1,0,0)$. Pour que la base soit optimale, on doit avoir $\hat{C} \geq 0$. Or, comme $\hat{c}_1 = -1$ et $\hat{c}_2 = -1$, on fait un choix arbitraire. Choisissons $\hat{c}_1$ et x_1 entre dans la base (h = 1).

$$y_1 = \begin{pmatrix} y_{31} \\ y_{41} \end{pmatrix} = (A^J)^{-1} A_1 = \begin{pmatrix} 2 \\ 2 \end{pmatrix}$$

La variable qui sort de la base est donnée par

$$\min \left\{ \frac{x_3}{y_{31}}, \frac{x_4}{y_{41}} \right\} = \min \left\{ \frac{6}{2}, \frac{6}{1} \right\} = \frac{x_3}{y_{31}}$$

Donc x_3 (s = 3) sort de la base.

Nous procédons de la même manière jusqu'à ce que la condition $\hat{C} \geq 0$ ait lieu. Cette condition aura lieu pour la base J = $\{1,2\}$,

où, $A^J = \begin{pmatrix} 2 & 1 \\ 1 & 2 \end{pmatrix}$, $x_J = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} 2 \\ 2 \end{pmatrix}$, $\Pi = \left(-\frac{1}{3}, -\frac{1}{3}\right)$ et $\hat{C} = \left(0, 0, \frac{1}{3}, \frac{1}{3}\right)$

En conséquence, Min Z= Π b = $\left(-\frac{1}{3}, -\frac{1}{3}\right) \begin{pmatrix} 6 \\ 6 \end{pmatrix}$ = -4 et x = $\begin{pmatrix} 2 \\ 2 \\ 0 \\ 0 \end{pmatrix}$

3-2 Vérification de la résolution du programme linéaire (12) avec le logiciel TORA

TORA Optimization System, Windows®-version 1.00
Copyright © 2000-2002 Hamdy A. Taha. All Rights Reserved
Tuesday, March 01, 2016 22:55

LINEAR PROGRAM -- ORIGINAL DATA

Title: Minimize

	x1	x2		
Minimize	-1.00	-1.00		
Subject to				
(1)	2.00	1.00	<=	6.00
(2)	1.00	2.00	<=	6.00
Lower Bound	0.00	0.00		
Upper Bound	infinity	infinity		
Unrestr'd (y/n)?	n	n		

SIMPLEX TABLEAUS -- (Starting All-Slack Method)

Title: Minimize

Iteration 1					
Basic	x1	x2	sx3	sx4	Solution
z (min)	1.00	1.00	0.00	0.00	0.00
sx3	2.00	1.00	1.00	0.00	6.00
sx4	1.00	2.00	0.00	1.00	6.00
Lower Bound	0.00	0.00			
Upper Bound	infinity	infinity			
Unrestr'd (y/n)?	n	n			
Iteration 2					
Basic	x1	x2	sx3	sx4	Solution
z (min)	0.00	0.50	-0.50	0.00	-3.00
x1	1.00	0.50	0.50	0.00	3.00
sx4	0.00	1.50	-0.50	1.00	3.00
Lower Bound	0.00	0.00			
Upper Bound	infinity	infinity			
Unrestr'd (y/n)?	n	n			
Iteration 3					
Basic	x1	x2	sx3	sx4	Solution
z (min)	0.00	0.00	-0.33	-0.33	-4.00
x1	1.00	0.00	0.67	-0.33	2.00
x2	0.00	1.00	-0.33	0.67	2.00
Lower Bound	0.00	0.00			
Upper Bound	infinity	infinity			
Unrestr'd (y/n)?	n	n			

3-3. Illustration pour le cas de la maximisation

Soit le programme linéaire suivant :

$$\begin{cases} Max\ Z = \ 5x_1 + 3x_2 + 4x_3 \\ sous\ 4x_1 \ + 2x_2 \ + 3x_3 \leq 80 \\ 2x_1 \ + 2x_2 \ + 3x_3 \leq 50 \\ x_1 \ + 3x_2 \ + 2x_3 \leq 40 \\ x_1 , x_2 , x_3 \ \geq 0 \end{cases} \tag{13}$$

A la forme standard, ce programme s'écrit :

$$\begin{cases} Max\ Z = \ 5x_1 + 3x_2 + 4x_3 \\ sous\ 4x_1 \ + 2x_2 \ + 3x_3 + x_4 = 80 \\ 2x_1 \ + 2x_2 \ + 3x_3 + x_5 = 50 \\ x_1 \ + 3x_2 \ + 2x_3 + x_6 = 40 \\ x_1 , x_2 , x_3 , x_4 , x_5 , x_6 \ \geq 0 \end{cases}$$

Posant J $= \{4,5,6\}$, A^J devient :

$A^J = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$ et donc $x_J = \begin{pmatrix} x_4 \\ x_5 \\ x_6 \end{pmatrix} = \begin{pmatrix} 80 \\ 50 \\ 40 \end{pmatrix}$, $\Pi = (0, 0{,}0)$ et par conséquent $\hat{C} = (5, 3{,}4{,}0{,}0{,}0)$.

Pour que la base soit optimale, on doit avoir $\hat{C} \leq 0$. Or, comme $\hat{c}_1$ possède la plus grande valeur ($\hat{c}_1 = 5$) que $\hat{c}_2 = 3$ et $\hat{c}_3 = 4$, x_1 entre dans la base (h = 1).

$$y_1 = \begin{pmatrix} y_{41} \\ y_{51} \\ y_{61} \end{pmatrix} = (A^J)^{-1} A_1 = \begin{pmatrix} 4 \\ 2 \\ 2 \end{pmatrix}$$

La variable qui sort de la base est donnée par

$\min\left\{\frac{x_4}{y_{41}}, \frac{x_5}{y_{51}}, \frac{x_6}{y_{61}}\right\} = \min\left\{\frac{80}{4}, \frac{50}{2}, \frac{40}{1}\right\} = \frac{x_4}{y_{41}}$. Donc x_4 (s $=$ 4) sort de la base. Nous procédons de la même manière jusqu'à ce que la condition $\hat{C} \geq 0$ ait lieu. Cette condition aura lieu pour la base $J = \{1,5,2\}$,

où $A^J = \begin{pmatrix} 4 & 0 & 2 \\ 2 & 1 & 2 \\ 1 & 0 & 3 \end{pmatrix}$, $x_J = \begin{pmatrix} x_1 \\ x_5 \\ x_2 \end{pmatrix} = \begin{pmatrix} 16 \\ 2 \\ 5 \end{pmatrix}$, $\Pi = \left(\frac{6}{5}, 0, \frac{1}{5}\right)$ et $\hat{C} = \left(0, 0,0, -\frac{6}{5}, 0, -\frac{1}{5}\right)$.

En conséquence, Max Z$= \Pi$ b $= \left(\frac{6}{5}, 0, \frac{1}{5}\right)\begin{pmatrix} 80 \\ 50 \\ 40 \end{pmatrix} = 104$ et x $= \begin{pmatrix} 16 \\ 8 \\ 0 \\ 0 \\ 2 \\ 0 \end{pmatrix}$.

3-4. Vérification de la résolution du programme linéaire (13) avec le logiciel TORA

TORA Optimization System, Windows®-version 1.00

Tuesday, March 01, 2016 23:15

LINEAR PROGRAM -- ORIGINAL DATA

Title: Maximize

	x1	x2	x3		
Maximize	5.00	3.00	4.00		
Subject to					
(1)	4.00	2.00	3.00	<=	80.00
(2)	2.00	2.00	3.00	<=	50.00
(3)	1.00	3.00	2.00	<=	40.00
Lower Bound	0.00	0.00	0.00		
Upper Bound	infinity	infinity	infinity		
Unrestr'd (y/n)?	n	n	n		

SIMPLEX TABLEAUS -- (Starting All-Slack Method)

Title: MAXIMIZE

Iteration 1

Basic	x1	x2	x3	sx4	sx5	sx6
z (max)	-5.00	-3.00	-4.00	0.00	0.00	0.00
sx4	4.00	2.00	3.00	1.00	0.00	0.00
sx5	2.00	2.00	3.00	0.00	1.00	0.00
sx6	1.00	3.00	2.00	0.00	0.00	1.00
Lower Bound	0.00	0.00	0.00			
Upper Bound	infinity	infinity	infinity			
Unrestr'd (y/n)?	n	n	n			

Basic	Solution
z (max)	0.00
sx4	80.00
sx5	50.00
sx6	40.00

Iteration 2

Basic	x1	x2	x3	sx4	sx5	sx6
z (max)	0.00	-0.50	-0.25	1.25	0.00	0.00
x1	1.00	0.50	0.75	0.25	0.00	0.00
sx5	0.00	1.00	1.50	-0.50	1.00	0.00
sx6	0.00	2.50	1.25	-0.25	0.00	1.00
Lower Bound	0.00	0.00	0.00			
Upper Bound	infinity	infinity	infinity			
Unrestr'd (y/n)?	n	n	n			

Basic	Solution
z (max)	100.00
x1	20.00
sx5	10.00
sx6	20.00

Iteration 3

Basic	x1	x2	x3	sx4	sx5	sx6
z (max)	0.00	0.00	0.00	1.20	0.00	0.20
x1	1.00	0.00	0.50	0.30	0.00	-0.20
sx5	0.00	0.00	1.00	-0.40	1.00	-0.40
x2	0.00	1.00	0.50	-0.10	0.00	0.40
Lower Bound	0.00	0.00	0.00			
Upper Bound	infinity	infinity	infinity			
Unrestr'd (y/n)?	n	n	n			

Basic	Solution
z (max)	104.00
x1	16.00
sx5	2.00
x2	8.00

3-5. Tableaux du simplexe et pivotage

Soit le programme linéaire

$$\begin{cases} Max\ Z = c^{\bar{J}} x_{\bar{J}} + c^J x_J \\ Sous\ (A^{\bar{J}}, I)\binom{x_{\bar{J}}}{x_J} \\ X \geq 0 \end{cases} \tag{14}$$

Nous mettons ce programme dans un ***Tableau***, dit du simplexe, de la forme suivante :

	$x_{\bar{J}}$	x_J	B
x_J	$A^{\bar{J}}$	I	
$\hat{C}$			

où I est une matrice identité d'ordre égal au nombre de variables d'écarts.

La résolution du *programme linéaire (14)* dans le ***Tableau*** du simplexe suit les étapes suivantes :

- Etape 1 : construire le ***Tableau*** du simplexe ;
- Etape 2 : vérifier l'optimalité ;
- Etape 3 : choisir la variable qui entre dans la base ;
- Etape 4 : choisir la variable qui sort dans la base ;
- Etape 5 : déterminer l'élément pivot : l'élément situé à l'intersection de la colonne pivot (colonne de la variable qui entre dans la base) et de la ligne pivot (ligne de la variable qui sort de la base) ;
- Etape 6 : effectuer le pivotage (diviser la ligne pivot par l'élément pivot) ;
- Etape 7 : aller à l'étape 2.

Après le pivotage, nous éliminons la variable entrante des autres équations. Dans cet article, nous donnons une formule générale pour cette élimination :

$$L_i\left(L_i \neq L_{pivot}\right) \rightarrow L_i - [\text{élément de } L_i \times \frac{\mathbf{1}}{\textbf{élément pivot}} L_{\boldsymbol{pivot}}] \tag{15}$$

où, L_{pivot} *est la ligne pivot et* L_i *la ligne qui n'est pas pivot. Cette **Formule** est utile pour les cas simples et surtout pour les cas compliqués où l'intrusion mathématique n'est pas immédiate.*

3-5-1. Illustration

Soit le programme linéaire

$$\begin{cases} Max\ Z = 20x_1 + 15x_2 + 18x_3 \\ sous\ 5x_1 + 10x_2 + 4x_3 \leq 80 \\ 15x_1 + 12x_2 + 5x_3 \leq 120 \\ 7x_1 + 21x_2 + 3x_3 \leq 84 \\ x_1, x_2, x_3 \geq 0 \end{cases} \quad (16)$$

A la forme standard, ce programme s'écrit :

$$\begin{cases} Max\ Z = 20x_1 + 15x_2 + 18x_3 \\ sous\ 5x_1 + 10x_2 + 4x_3 + x_4 = 80 \\ 15x_1 + 12x_2 + 5x_3 + x_5 = 120 \\ 7x_1 + 21x_2 + 3x_3 + x_6 = 84 \\ x_1, x_2, x_3, x_4, x_5, x_6 \geq 0 \end{cases}$$

Résolvons ce programme linéaire à l'aide du ***Tableau*** du simplexe

	x_1	x_2	x_3	x_4	x_5	x_6	B
x_4	5	10	4	1	0	0	80
x_5	15	12	5	0	1	0	120
x_6	7	21	3	0	0	1	84
$\hat{C}$	20	15	18	0	0	0	0
x_4	0	6	7/3	1	-1/3	0	40
x_1	1	4/5	1/3	0	1/15	0	8
x_6	0	77/5	2/3	0	-7/15	1	28
$\hat{C}$	0	-1	34/3	0	-4/3	0	-160
x_3	0	18/7	1	3/7	-1/7	0	120/7
x_1	1	-2/35	0	-1/7	4/35	0	16/7
x_6	0	479/35	0	-2/7	-117/315	1	116/7
$\hat{C}$	0	-211/7	0	-34/7	2/7	0	-248/7
x_3	5/4	5/2	1	1/4	0	0	20
x_5	35/4	-1/2	0	-5/4	1	0	20
x_6	13/4	189/14	0	-3/4	0	1	24
$\hat{C}$	-5/2	-30	0	-9/2	0	0	-360

A la première itération, x_1 entre dans la base et x_5 sort de la base : l'élément pivot est 15 et se trouve à la deuxième ligne (L2). Ainsi, utilisant ***la Formule (15)***, nous avons les combinaisons pour la deuxième itération :

L2 $\rightarrow \frac{1}{15}$ L2 ;

L1→L1 - [5$\times$ $\frac{1}{15}$ L2]= L1 - $\frac{1}{3}$L2 ;
L3→L3 - [7$\times$ $\frac{1}{15}$ L2]=L3 - $\frac{7}{15}$L2 ;
L4→L4 - [20$\times$ $\frac{1}{15}$ L2]= L4 - $\frac{20}{15}$L2.

A la deuxième itération, x_3 entre dans la base et x_4 sort de la base : l'élément pivot est $\frac{7}{3}$ et se trouve à la première ligne (L1). Ainsi, utilisant ***la Formule (15)***, nous avons les combinaisons pour la troisième itération :

L1→ $\frac{3}{7}$ L1 ;
L2→L2 - [$\frac{1}{3}$ $\times$ $\frac{3}{7}$ L1] = L2 - $\frac{1}{7}$L1 ;
L3→L3 - [$\frac{2}{3}$ $\times$ $\frac{3}{7}$ L1] = L3 - $\frac{2}{7}$ L1 ;
L4→L4 - [$\frac{34}{3}$ $\times$ $\frac{3}{7}$ L1] = L4 - $\times$ $\frac{34}{7}$L1.

A la troisième itération, x_5 entre dans la base et x_4 sort de la base : l'élément pivot est $\frac{4}{35}$ et se trouve à la deuxième ligne (L2). Ainsi, utilisant ***la Formule (15)***, nous avons les combinaisons pour la dernière itération (quatrième itération) :

L2→ $\frac{35}{4}$ L2 ;
L1→ L1 - [- $\frac{1}{7}\frac{35}{4}$ L2] = L1+ $\frac{5}{4}$ L2 ;
L3→ $L3$ - [- $\frac{117}{315}$ $\frac{35}{4}$ L2] = L3+ $\frac{13}{4}$ L2 ;
L4→L4 - [$\frac{2}{7}$ $\frac{35}{4}$ L2] = L4 - $\frac{5}{2}$ L2.

A la quatrième itération, les variables de base sont $x_3 = 20$, $x_5 = 20$ et $x_6 = 24$.

D'où, x=$\begin{pmatrix} 0 \\ 0 \\ 20 \\ 0 \\ 20 \\ 24 \end{pmatrix}$

Max Z $= 20x_1 + 15x_2 + 18x_3 = 20.0 + 15.0 + 18.20 = 360$.

3-5-2. Vérification de la résolution du programme linéaire (16) avec le logiciel TORA

LINEAR PROGRAM -- ORIGINAL DATA

Title: Maximize

	x1	x2	x3		
Maximize	20.00	15.00	18.00		
Subject to					
(1)	5.00	10.00	4.00	<=	80.00
(2)	15.00	12.00	5.00	<=	120.00
(3)	7.00	21.00	3.00	<=	84.00
Lower Bound	0.00	0.00	0.00		
Upper Bound	infinity	infinity	infinity		
Unrestr'd (y/n)?	n	n	n		

Iteration 1

Basic	x1	x2	x3	sx4	sx5	sx6
z (max)	-20.00	-15.00	-18.00	0.00	0.00	0.00
sx4	5.00	10.00	4.00	1.00	0.00	0.00
sx5	15.00	12.00	5.00	0.00	1.00	0.00
sx6	7.00	21.00	3.00	0.00	0.00	1.00
Lower Bound	0.00	0.00	0.00			
Upper Bound	infinity	infinity	infinity			
Unrestr'd (y/n)?	n	n	n			

Basic	Solution
z (max)	0.00
sx4	80.00
sx5	120.00
sx6	84.00

Iteration 2

Basic	x1	x2	x3	sx4	sx5	sx6
z (max)	0.00	1.00	-11.33	0.00	1.33	0.00
sx4	0.00	6.00	2.33	1.00	-0.33	0.00
x1	1.00	0.80	0.33	0.00	0.07	0.00
sx6	0.00	15.40	0.67	0.00	-0.47	1.00
Lower Bound	0.00	0.00	0.00			
Upper Bound	infinity	infinity	infinity			
Unrestr'd (y/n)?	n	n	n			

Basic	Solution
z (max)	160.00
sx4	40.00
x1	8.00
sx6	28.00

Iteration 3

Basic	x1	x2	x3	sx4	sx5	sx6
z (max)	0.00	30.14	0.00	4.86	-0.29	0.00
x3	0.00	2.57	1.00	0.43	-0.14	0.00
x1	1.00	-0.06	0.00	-0.14	0.11	0.00
sx6	0.00	13.69	0.00	-0.29	-0.37	1.00
Lower Bound	0.00	0.00	0.00			
Upper Bound	infinity	infinity	infinity			
Unrestr'd (y/n)?	n	n	n			

Basic	Solution
z (max)	354.29
x3	17.14
x1	2.29
sx6	16.57

Iteration 4

Basic	x1	x2	x3	sx4	sx5	sx6
z (max)	2.50	30.00	0.00	4.50	0.00	0.00
x3	1.25	2.50	1.00	0.25	0.00	0.00
sx5	8.75	-0.50	0.00	-1.25	1.00	0.00
sx6	3.25	13.50	0.00	-0.75	0.00	1.00
Lower Bound	0.00	0.00	0.00			
Upper Bound	infinity	infinity	infinity			
nrestr'd (y/n)?	n	n	n			

Basic	Solution
z (max)	360.00
x3	20.00
sx5	20.00
sx6	24.00

4. Conclusion

En conclusion, il est prouvé par les résultats que la ***Formule (9)*** donnant la variable qui entre dans la base, lors de la résolution d'un programme linéaire avec la méthode du simplexe ordinaire sans ***Tableau***, est valable s'il s'agit de minimiser ou de maximiser le programme linéaire. Quant à la résolution du même programme linéaire avec la méthode du simplexe ordinaire en ***Tableau***, il est vérifié que la ***Formule (15)*** utilisée dans l'élimination de la variable entrante dans les équations qui ne contiennent pas l'élément pivot, est générale. Le logiciel TORA est l'outil convenable pour toutes nos vérifications.

Références

[1] - K. C. RAO and S. L. MISHRA, "*Operations research*", Alpha Science, Harrow, U.K., (2005).
[2] - S. K. BOSE, "*Operations research*", *Alpha Science*, Harrow, U.K., (2005).
[3] - G. CHARRON et P. PARENT "*Algèbre linéaire et géométrie vectorielle*", 3 édition, *Beauchemin Chenelière Education*, Canada, (2005).
[4] - H. A. TAHA, "*Operations Research : An Introduction*", 8th Edition, Pearson Education, USA, (2007).
[5] - H. COILLAND, '' La recherche opérationnelle par l'exemple : Tome 1 : méthodes '', 1ère Edition, *bookboon.com*, ISBN 978-87-403-0738-2, (2014).
[6] - H. A. TAHA, ''*Operations Research : an Introduction*'', 7th Ed, *Prentice Hall*, New Jersey, (2003).