

Programmation Linéaire - Programmation en Python de l'optimisation par la Méthode de Simplexe : Cas de la production à la Brasserie SIMBA de Lubumbashi

[Linear Programming - Optimization programming in Python by the Simplex Method : Case of production at the Simba Brewery in Lubumbashi]

Mbuyi Wa Mbuyi Stephane

Département des Mathématiques Informatiques, ISP Lubumbashi, RD Congo

Copyright © 2022 ISSR Journals. This is an open access article distributed under the *Creative Commons Attribution License*, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

ABSTRACT: The purpose of this article is part of the logic of establishing a production plan in order to maximize turnover. In general, the resolution of mathematical programming problems (the production problem case of the BRASIMBA of Lubumbashi), aims to determine the best possible combination of limited resources (storage capacity for example), to achieve a certain objective. These allocations must maximize a so-called objective function, which can be a cost or a profit. To achieve our goal, we used the Python object-oriented programming language to computerize the so-called Simplex method, which starts from a basic feasible solution or acceptable basic solution that is not improved step by step. This method stems from linear programming, which is nothing more than a particularity of mathematical programming.

KEYWORDS: Algorithm, programming language, maximization, economic function, simplex array, pivot.

RESUME: L'objet de ce présent article s'inscrit dans la logique d'établir un plan de production de façon à maximiser les chiffres d'affaires. De manière générale, la résolution des problèmes de programmation mathématique (le cas de problème de production de la BRASIMBA de Lubumbashi), vise à déterminer la meilleure combinaison possible des ressources limitées (capacité de stockage par exemple), pour atteindre un certain objectif. Ces allocations doivent maximiser une fonction dite objectif, qui peut être un coût ou un profit. Pour atteindre notre objectif, nous nous sommes servis du langage de programmation orienté objet python pour informatiser la méthode dite de Simplexe, qui part d'une solution réalisable de base ou solution de base admissible que l'on n'améliore pas à pas. Cette méthode découle de la programmation linéaire qui n'est autre qu'une particularité de la programmation mathématique.

MOTS-CLEFS: Algorithme, langage de programmation, maximisation, fonction économique, tableau simplexe, pivot.

1. INTRODUCTION

En raison des enjeux décisionnels, dans un environnement où la concurrence est de plus en plus accrue, ne doivent plus agir intuitivement, ni fonder leurs décisions sur des calculs naïfs dans la gestion quotidienne de stock et de production. Une bonne résolution de ce type des problèmes, nécessite la connaissance des méthodes approuvées ainsi que la maîtrise des outils mathématiques et informatiques développés à cet effet. Toutes les méthodes proposées se résument à l'énoncé mathématique, à savoir maximiser ou minimiser une fonction numérique de variables soumises à diverses contraintes. Ces contraintes peuvent être d'origines très variées: les conditions de la production, le volume de stockage disponible, la quantité des matières en réserve, etc.

2. PROBLEMATIQUE

Nous constatons que l'optimisation d'une fonction linéaire d'un problème classique par la méthode de simplexe devient de plus en plus complexe lors du traitement manuel.

Notre travail s'organise autour d'une problématique très précise, c'est-à-dire des interrogations internes au sujet qui peuvent élever des suggestions, des points à clarifier et à développer, en essayant de donner pour ce sujet des réponses aux préoccupations suivantes:

- Comment optimiser une fonction linéaire d'un problème classique par la méthode de simplexe à l'aide du langage de programmation orienté objet python ?
- Quelle solution pourrions-nous envisager afin de rendre facile et efficace l'optimisation d'une fonction linéaire d'un problème classique par la méthode de simplexe à l'aide du langage de programmation orienté objet python ?

En ce qui nous concerne, la problématique est celle de savoir, comment informatiser l'optimisation d'une fonction linéaire pour rendre efficace l'établissement d'un plan de production à la brasserie SIMBA de façon à maximiser les chiffres d'affaires ?

3. METHODOLOGIE

Pour aborder un thème, quel que soit son domaine, sa grandeur ou son importance, il est nécessaire et important de tracer une ligne de conduite qui guiderait le thème jusqu'à son point final. D'où pour notre thème, nous avons eu à utiliser une méthode et une technique.

Pour ce qui concerne la méthode, nous avons opté pour la méthode analytique, qui nous a permis par une démarche analytique de mettre en place une application informatique pouvant répondre à notre problématique.

Pour ce qui concerne la technique, nous avons opté pour la technique de recherche documentaire, car c'est grâce à cette dernière que nous avons pu consulter un certain nombre des documents, notamment des ouvrages, des articles, des travaux de fins de cycle, des sites internet, dans le but de constituer une base documentaire pour la bonne élaboration de notre travail.

4. CONSIDERATIONS THEORIQUES

4.1. DEFINITIONS DE LA PROGRAMMATION LINEAIRE

D'après Gérald BAILLARGEON « La programmation linéaire peut se définir comme un outil mathématique qui permet d'analyser divers types de situations dans lesquelles nous retrouvons une fonction économique ou fonction linéaire d'un certain nombre de variables, appelée fonction objectif que l'on désire optimiser, c'est-à-dire maximiser ou minimiser ». Ces variables appelées variables d'activité, puis après la résolution, on les qualifie de variables de décision (dont on veut en déterminer les valeurs optimales) qui sont soumises à des contraintes imposées par des ressources limitées de la situation que l'on veut analyser, les restrictions qui sont imposées prennent forme d'équations ou d'inéquations linéaires dans la formulation mathématique d'un modèle de programmation linéaire.

DORFMAN et Samuel SON ajoutent que « La programmation linéaire est une méthode de détermination de meilleur plan d'action pour réaliser des objectifs donnés dans une situation où les ressources sont limitées ».

Dans le domaine des sciences de la gestion, on pourrait dire que la programmation linéaire est un outil scientifique qui permet d'obtenir une meilleure affectation des ressources de l'entreprise (main d'œuvre, matières premières, capitaux, espace, etc.), pour atteindre un objectif spécifique comme la maximisation des bénéfices ou la minimisation des coûts, « l'objectif de la programmation linéaire est de maximiser (ou minimiser) une fonction objectif linéaire sous contraintes, qui sont représentées par des égalités ou inégalités linéaire ».

On constate dans les définitions précédentes que la programmation linéaire est la branche des mathématiques qui étudient les variables d'activités sous la forme d'un programme linéaire et la résolution de certains problèmes d'optimisation sous contraintes linéaires; elle est utilisée, en particulier, dans la production, dans la planification et l'allocation des ressources limitées en vue d'atteindre des objectifs fixés par les décideurs.

On appelle programme linéaire, le problème mathématique qui a pour objectif d'optimiser (maximiser ou minimiser) une fonction linéaire de (n) variables réelles non négatives qui sont reliées par des (m) relations linéaires appelées contraintes sous d'égalités et ou des inégalités.

4.2. FORMULATION D'UN PROGRAMME LINEAIRE

Un programme linéaire est un problème dans lequel on est amené à maximiser (ou minimiser) une application linéaire, appelée fonction objectif ou fonction économique, sur un ensemble d'équations et ou d'inéquations linéaires, dites contraintes. Autrement dit, la programmation linéaire est une branche des mathématiques qui a pour but de résoudre des problèmes d'optimisation linéaire de type:

$$[OPT] Z = C_1X_1 + C_2X_2 + \dots + C_iX_j + \dots + C_nX_n$$

Sous les contraintes:

$$\begin{cases}
 a_{11}X_1 + a_{12}X_2 + \dots + a_{1j}X_j + \dots + a_{1n}X_n (\leq, =, \geq) b_1 \\
 a_{21}X_1 + a_{22}X_2 + \dots + a_{2j}X_j + \dots + a_{2n}X_n (\leq, =, \geq) b_2 \\
 \dots \dots \dots \dots \dots \dots \dots \dots \dots \\
 a_{i1}X_1 + a_{i2}X_2 + \dots + a_{ij}X_j + \dots + a_{in}X_n (\leq, =, \geq) b_i \\
 \dots \dots \dots \dots \dots \dots \dots \dots \dots \\
 a_{m1}X_1 + a_{m2}X_2 + \dots + a_{mj}X_j + \dots + a_{mn}X_n (\leq, =, \geq) b_m \\
 X_1, X_2, \dots, X_j, \dots, X_n \geq 0
 \end{cases}$$

Soient:

- $(X_1, X_2, \dots, X_j, \dots, X_n)$ représentent les variables d'activités; à l'hypothèse que ces variables sont positives, signifient que $X_1, X_2, \dots, X_j, \dots, X_n \geq 0$
- La fonction objectif a une forme linéaire en relation avec les variables de décision du type $[OPT] Z = C_1X_1 + C_2X_2 + \dots + C_jX_j + \dots + C_nX_n$ où les coefficients $(C_1, C_2, \dots, C_j, \dots, C_n)$ doivent avoir une valeur bien déterminée (avec certitude) et peuvent être positifs.

Par exemple le coefficient C_j peut représenter un profit unitaire lié à la quantité de la production X_j , ainsi, la valeur de la fonction Z est le profit total lié à la production $(X_2, \dots, X_j, \dots, X_n)$.

- Supposons que ces variables d'activités doivent vérifier un système d'équations linéaires définit par (m) inégalités ou (n) variables; où les coefficients sous la forme de la matrice (m x n) suivante:

$$\left(\begin{array}{cccccc}
 a_{11} & a_{12} & \dots & a_{1j} & \dots & a_{1n} \\
 a_{21} & a_{22} & \dots & a_{2j} & \dots & a_{2n} \\
 \dots & \dots & \dots & \dots & \dots & \dots \\
 a_{i1} & a_{i2} & \dots & a_{ij} & \dots & a_{in} \\
 \dots & \dots & \dots & \dots & \dots & \dots \\
 a_{m1} & a_{m2} & \dots & a_{mj} & \dots & a_{mn}
 \end{array} \right) = (a_{ij})$$

Avec: $1 \leq j \leq n$ et $1 \leq i \leq m$.

Et le vecteur colonne $(b_1, b_2, \dots, b_i, \dots, b_m)$ qui doit avoir une valeur bien déterminée et peut être positif ou nul. Le paramètre (b_i) représente la quantité des ressources disponibles de la matrice du système des contraintes dont le bien X_j utilise une quantité égale à $(a_{ij}X_j)$. « Les constantes (à savoir, les coefficients de la matrice du système des contraintes et la fonction objectif) sont appelés les paramètres du modèle ».

4.3. LES FORMES GENERALES D’UN PROGRAMME LINEAIRE

On trouve souvent deux formes de programme linéaires. La forme canonique et la forme standard. La première représente la forme initiale, elle se caractérise par des contraintes linéaires sous formes des inéquations (\leq, \geq) ou d’équations (=) comportant (j) variables et (i) contraintes comme suit:

$$[OPT] Z = C_1X_1 + C_2X_2 + \dots + C_1X_j + \dots + C_nX_n$$

Sous les contraintes:

$$\left\{ \begin{array}{l} a_{11}X_1 + a_{12}X_2 + \dots + a_{1j}X_j + \dots + a_{1n}X_n (\leq, =, \geq) b_1 \\ a_{21}X_1 + a_{22}X_2 + \dots + a_{2j}X_j + \dots + a_{2n}X_n (\leq, =, \geq) b_2 \\ \dots \dots \dots \\ a_{i1}X_1 + a_{i2}X_2 + \dots + a_{ij}X_j + \dots + a_{im}X_m (\leq, =, \geq) b_i \\ \dots \dots \dots \\ a_{m1}X_1 + a_{m2}X_2 + \dots + a_{mj}X_j + \dots + a_{mn}X_n (\leq, =, \geq) b_m \\ X_1, X_2, \dots, X_j, \dots, X_n \geq 0 \end{array} \right.$$

Avec:

X_j : Variable de décision, avec $1 \leq j \leq n$.

b_i : seconds membres (les disponibilités), avec $1 \leq i \leq m$.

C_j : Coefficients de la fonction objectif, avec $1 \leq j \leq n$.

a_{ij} : coefficients des contraintes, avec $1 \leq j \leq n$ et $1 \leq i \leq m$.

4.3.1. FORME MATRICIELLE

Un système d’équations linéaires de (m) équations à (n) inconnues admet l’écriture matricielle de la forme suivante:

$$\begin{pmatrix} a_{11} & a_{12} & \dots & a_{1j} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2j} & \dots & a_{2n} \\ \dots & \dots & \dots & \dots & \dots & \dots \\ a_{m1} & a_{m2} & \dots & a_{mj} & \dots & a_{mn} \end{pmatrix} \begin{pmatrix} X_1 \\ X_2 \\ \vdots \\ \dots \\ X_m \end{pmatrix} = \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ \dots \\ b_m \end{pmatrix}$$

On appelle fonction économique la fonction qui doit être optimisée. Elle est écrite $[Opt] (Z) = \sum_{j=1}^n C_j X_j$

- Les variables d'activité ($X_1, X_2, \dots, X_j, \dots, X_n$)

Ces variables sont appelés les variables de contrôle ou variables instrumentales ou variables de commande ou plus brièvement les commandes du programme. « On représente par X_j les variables d'activité du programme linéaire; les variables X_j ont une signification concrète (quantités produites, vendues ou transportées, durées, valeurs monétaires, etc.)

Ces variables sont toujours positives ou nulles (condition de non-négativité $X_j \geq 0$ ».

L'agent décideur à la capacité d'interpréter ces variables et son choix final sera la valeur qui optimise la fonction objective.

- Les coefficients de la fonction objectif ($C_1, C_2, \dots, C_j, \dots, C_n$):

On représente par C_j le coefficient de la variable X_j : ($X_1, X_2, \dots, X_j, \dots, X_n$) de la fonction économique. Dans beaucoup de cas, les coefficients des variables d'activité de la fonction économique représentent les profits ou les coûts associés à une unité des différentes activités. En d'autres termes, c'est la contribution unitaire de la variable correspondante à l'objectif poursuivi par cette dernière.

- Les contraintes d'activité ($a_{ij}X_j$) ($\leq = \geq$) (b_i):

Etant donné que chaque activité consomme une certaine quantité de chacune des ressources, et qu'on ne dispose que d'une quantité limitée de chaque ressource, il y aura des limitations sur les valeurs prises par les variables de décision; ces limitations seront exprimées par l'intermédiaire de relations exprimant la disponibilité limitée des ressources. On donne à ces relations le nom de contraintes

- Les coefficients techniques (a_{ij}), $1 \leq j \leq n$ et $1 \leq i \leq m$:

On représente par a_{ij} le coefficient technique associé à la ressource (i) et l'activité numéro (j), il représente une matrice de (i) lignes et de (j) colonnes. « Les coefficients techniques a_{ij} représentent la quantité du facteur (i) par production (j) ». C'est-à-dire lorsque la variable de décision associé à l'activité en question est égale à un (1).

- Les ressources disponibles b_i : ($b_1, b_2, \dots, b_i, \dots, b_m$)

On représente par b_i les quantités des ressources disponibles et qui limitent l'optimisation de l'objectif poursuivi par cette entreprise

4.3.2. LA RESOLUTION D'UN MODELE LINEAIRE

De manière générale, la résolution des problèmes de programmation linéaire vise à déterminer l'allocation optimale et la meilleure allocation possible des ressources limitées pour atteindre certains objectifs, ces allocations doivent minimiser ou maximiser une fonction dite objectif. En économie, ces fonctions sont le profit ou le coût. Pour ce faire, on fait appel aux méthodes de résolution d'un programme linéaire on cite: la méthode graphique et la méthode algébrique.

Parmi les méthodes algébriques on peut citer entre autre: méthode de simplexe, la méthode de Vogel, la méthode de Stepping Stone, ...

4.4. LA RESOLUTION DES PROGRAMMES LINEAIRES PAR LA METHODE SIMPLEXE

Résoudre un programme linéaire consiste à déterminer les valeurs des variables non négatives (X_i) qui permettent d'optimiser la fonction économique. Il existe plusieurs méthodes de résolutions, la méthode algébrique, la méthode graphique, qui porte chacune des limites. Il faut donc trouver une autre méthode: celle du simplexe, qui est la plus utilisée dans la résolution des programmes linéaires, puis qu'elle répond aux développements des techniques de résolution par ordinateur. L'algorithme du simplexe consiste à passer d'une solution à une autre jusqu'à ce qu'une solution réalisable de base optimale soit trouvée ».

En d'autres termes, la solution optimale est celle qui ne contient aucune variable positive dans la fonction objective dans le tableau final du simplexe pour un problème de maximisation et aucune variable négative dans la fonction objective dans le tableau final du simplexe pour un problème de minimisation. Pour calculer la solution optimale, la règle de sélection est appelée règle de pivotage.

4.4.3.1. LA RESOLUTION DES PROGRAMMES DE MAXIMISATION ET TYPE DE CONTRAINTES (≥):

Dans le cas d’une contrainte de signe (≥), pour la transformer en équation on doit soustraire une variable d’écart, également appelée dans ce cas, variable excédent.

Soit le programme linéaire suivant:

$$[Max] Z = C_1X_1 + C_2X_2 + \dots + C_nX_n$$

$$\begin{cases} a_{11}X_1 + a_{12}X_2 + \dots + a_{1n}X_n \geq b_1 \\ a_{21}X_1 + a_{22}X_2 + \dots + a_{2n}X_n \geq b_2 \\ \dots \dots \dots \dots \dots \dots \dots \\ a_{m1}X_1 + a_{m2}X_2 + \dots + a_{mn}X_n \geq b_m \\ X_1, X_2, \dots, X_n \geq 0 \end{cases}$$

Le passage de la forme canonique à la forme standard nécessite l’introduction des variables d’écart négatives dans chaque contrainte de type (≥):

$$[Max] Z = C_1X_1 + C_2X_2 + \dots + C_nX_n - 0X_{n+1} - 0X_{n+2} - \dots - 0X_{n+m}$$

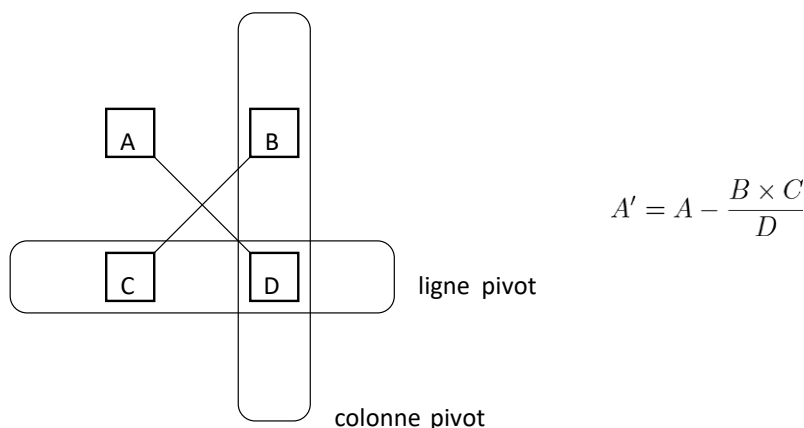
$$\begin{cases} a_{11}X_1 + a_{12}X_2 + \dots + a_{1n}X_n - X_{n+1} = b_1 \\ a_{21}X_1 + a_{22}X_2 + \dots + a_{2n}X_n - X_{n+2} = b_2 \\ \dots \dots \dots \dots \dots \dots \dots \\ a_{m1}X_1 + a_{m2}X_2 + \dots + a_{mn}X_n - X_{n+m} = b_m \\ X_1, X_2, \dots, X_n, X_{n+1}, \dots, X_{n+m} \geq 0 . \end{cases}$$

Tableau 1. Le tableau de simplexe (solution de base z=0)

H BASE BASE	X ₁	X ₂	X _nX _{n+1}	X _{n+2}	X _{n+m}	b _i
X _{n+1}	a ₁₁	a ₁₂	a _{1n}	-1	0	0	b ₁
X _{n+2}	a ₂₁	a ₂₂	a _{2n}	0	-1	0	b ₂
.....	
X _{n+m}	a _{m1}	a _{m2}	a _{mn}	0	0	-1	b _m
C _j	C ₁	C ₂	C _m	0	0	0	Z=0

De ce fait, pour appliquer la méthode de simplexe, il faut ajouter à chaque contrainte une variable artificielle (t₁) comme suit:

$$[Max] Z = C_1X_1 + C_2X_2 + \dots + C_nX_n - 0X_{n+1} - 0X_{n+2} - \dots - 0X_{n+m} - Mt_1 - Mt_2 - \dots - Mt_m$$



Formellement,

$$\alpha'_{ij} = \alpha_{ij} - \frac{\alpha_{is}\alpha_{rj}}{\alpha_{rs}}, \bar{b}'_i = \bar{b}_i - \frac{\alpha_{is}\bar{b}_r}{\alpha_{rs}}, \bar{c}'_j = \bar{c}_j - \frac{\bar{c}_s\alpha_{rj}}{\alpha_{rs}}, z' = z + \frac{\bar{c}_s\bar{b}_r}{\alpha_{rs}}$$

4.4.3.2. LE PROCESSUS DE RESOLUTION DE PROGRAMMES DE MAXIMISATION A (N) VARIABLES ET (M) CONTRAINTES

Le processus de résolution par la méthode de simplexe est identique pour les programmes linéaires à minimiser ou à maximiser. Le traitement de la fonction économique est indépendamment des contraintes:

Quel que soit la fonction économique (Max ou Min), les modifications à porter sur les contraintes pour réaliser la forme standard sont les suivantes:

- Lorsque la contrainte est de la forme (\geq), on ajoute une variable d'écart négative (-E) et une variable artificielle positive (t_i).
- Lorsque la contrainte est de la forme (\leq), on ajoute uniquement une variable d'écart positive (E_i).
- La contrainte sous forme d'équation (=), on ajoute uniquement une variable artificielle (t_i).
- Lorsque l'objectif est de maximiser une fonction économique, les variables d'écart sont associées aux coefficients (-0), les variables artificielles sont associées aux coefficients (-M).

4.4.3.3. REGLES GENERALES POUR LA RESOLUTION D'UN PROGRAMME LINEAIRE PAR LA METHODE DE SIMPLEXE (PROBLEME DE MAXIMISATION)

Selon Eric-LAGREZE: « pour passer du sommet à un sommet adjacent, Dantzig à introduit deux critères (critères de Dantzig) qui sont au cœur de la méthode du simplexe: un critère de la nouvelle variable précédemment hors base à faire entrer dans la base et un critère du choix d'une variable de la base à faire sortir ».

A. CHOIX DE LA COLONNE PIVOT (CRITERE DE SELECTION DE LA VARIABLE ENTRANTE DANS LA BASE)

La colonne pivot est définie à partir des coefficients de la fonction économique.

« Le critère de sélection de la variable entrante est donc le suivant: on choisit la variable avec le coefficient objectif le plus élevé ». On cherche à se focaliser sur la variable qui, en augmentant, augmentera le plus possible la fonction objective.

Cette variable correspond au plus grand coefficient positif, la colonne pivot est la colonne qui le contient. S'il existe plusieurs coefficients correspondant à cette valeur positive maximale, on peut choisir celui que l'on veut (arbitrairement). La variable correspondante sera la variable entrante car elle ne va plus s'annuler. On sélectionne la variable hors base ayant le plus grand coefficient positif dans la ligne.

Pour sélectionner la variable sortante de la base, il est nécessaire de rajouter une (bi) (disponibilité) et de la colonne de la variable entrante dans la base.

B. CHOIX DE LA LIGNE PIVOT (CRITERE DE SELECTION DE LA VARIABLE SORTANTE DE LA BASE)

La variable entrante va prendre la place d'une des variables de base, appelée variable sortante.

Il faut maintenant trouver quelle valeur maximiser peut prendre cette variable entrante afin de maximiser la fonction objective. Pour un programme linéaire à plusieurs variables on fait diviser chaque coefficient de la colonne pivot et on calcule les rapports (b_1/a_{ij}) .

- On sélectionne la variable dans la base ayant le plus petit coefficient positif dans la colonne ®
- « Le critère de sélection de la variable sortante est donc le suivant: prendre comme variable sortante la première variable de base à annuler de manière générale, on calcule le minimum du rapport du coefficient du membre de droite sur le coefficient de la variable entrante dans la même ligne, lorsque celui-ci est positif. La variable sortante est celle dont on lit la valeur dans la ligne ou ligne pivot « on appelle élément pivot le coefficient situé à l’intersection de la colonne pivot et de la ligne pivot ».

On appelle « élément pivot le coefficient situe à l’intersection de la colonne pivot et de la ligne pivot ».

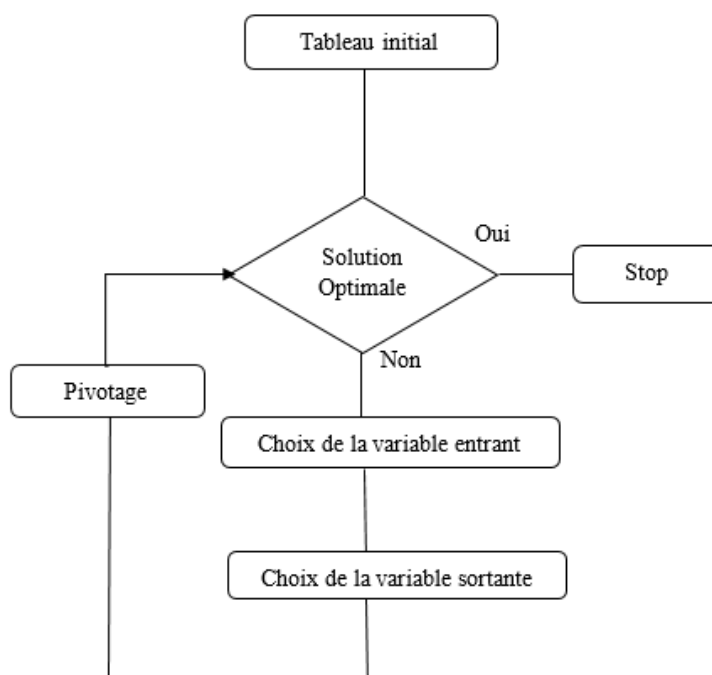
C. ITERATION

S’il existe un coefficient c_j positif dans le nouveau tableau, on retourne à la première étape (choix du pivot) puis à la deuxième (itération suivante). On réitère ce processus jusqu’à ce que tous les coefficients de la fonction économique soient négatifs.

D. CRITERE D’ARRET DES ITERATIONS

Si tous les coefficients de la ligne relatifs aux variables hors base, sont négatifs ou nuls pour la maximisation de la solution de base trouvée est optimale.

4.4.3.4. ORGANIGRAMME DE LA METHODE DE SIMPLEXE



4.4.3.5. EXEMPLE D’UN CAS

La Brasserie BRASIMBA pour fabriquer les quatre produits: CHUIN, TEMBO, SIMBA et DOPPEL pour lesquels elle utilise trois ressources maïs, houblon et malt. La fabrication de chaque produit nécessite une certaine quantité de ressources. Les ressources consommées, les stocks des ressources et les bénéfices unitaires des produits sont récapitulés dans le tableau suivant:

	CHUIN	TEMBO	SIMBA	DOPPEL	Stock
Mais	2	4	5	7	42
Houblon	1	1	2	2	17
Malt	1	2	3	3	24
Bénéfice	7	9	18	17	

Pour fabriquer 1000 centilitres de la bière CHUIN, le brasseur utilise 2g de Mais, 1 g de houblon et 1 g de malt. Et la fabrication de ce tonneau lui rapporte alors un bénéfice de 7 dollars. Le tableau se lit de manière analogue pour les bières TEMBO, SIMBA et DOPPEL

On souhaite établir un plan de production de façon à maximiser les chiffres d'affaires.

Appelant X_1, X_2, X_3, X_4 les quantités respectives de produit CHUIN, TEMBO, SIMBA, DOPPEL.

Le problème devient:

$$[Max] Z = 7X_1 + 9X_2 + 18X_3 + 17X_4$$

$$S\backslash C \left\{ \begin{array}{l} 2X_1 + 4X_2 + 5X_3 + 7X_4 \leq 42 \\ X_1 + X_2 + 2X_3 + 2X_4 \leq 17 \\ X_1 + 2X_2 + 3X_3 + 3X_4 \leq 24 \\ X_1, X_2, X_3, X_4 \geq 0 \end{array} \right.$$

Passons de la forme canonique à la forme standard;

Pour passer de la forme canonique à la forme standard, nous devons introduire les variables d'écart. ($X_5, X_6, X_7 \geq 0$).

Le problème devient:

$$[Max] Z = 7X_1 + 9X_2 + 18X_3 + 17X_4 + 0X_5 + 0X_6 + 0X_7$$

$$S\backslash C \left\{ \begin{array}{l} 2X_1 + 4X_2 + 5X_3 + 7X_4 + X_5 = 42 \\ X_1 + X_2 + 2X_3 + 2X_4 + X_6 = 17 \\ X_1 + 2X_2 + 3X_3 + 3X_4 + X_7 = 24 \\ X_1, X_2, X_3, X_4, X_5, X_6, X_7 \geq 0 \end{array} \right.$$

On reconnait là un problème de programmation linéaire sous *forme standard*.

Après introduction de trois *variables d'écart* x_5, x_6, x_7 , qui mesurent pour chaque ressource l'écart entre la quantité initialement disponible et la quantité consommée par le plan de fabrication donné par les *variables de décision* x_1, x_2, x_3, x_4 . On obtient une formulation équivalente du problème:

$$\begin{aligned} \text{Maximiser } z &= 7x_1 + 9x_2 + 18x_3 + 17x_4 \\ \text{s. c.} \quad 2x_1 + 4x_2 + 5x_3 + 7x_4 + x_5 &= 42 \\ x_1 + x_2 + 2x_3 + 2x_4 + x_6 &= 17 \\ x_1 + 2x_2 + 3x_3 + 3x_4 + x_7 &= 24 \\ x_1, x_2, x_3, x_4, x_5, x_6, x_7 &\geq 0 \end{aligned}$$

On appelle base toute sous-matrice d'ordre (m x m) extraite de A.

Base	Cr ou Cj									Observations
		P0	P1	P2	P3	P4	P5	P6	P7	
P5	0	42	2	4	5	7	1	0	0	$X_1 = X_2 = X_3 = X_4 = 0, X_5 = 42, X_6 = 17, X_7 = 24$ $X_5 = \frac{42}{5}, X_6 = \frac{17}{2}, X_7 = \frac{24}{3}$
P6	0	17	1	1	2	2	0	1	0	
P7	0	24	1	2	3	3	0	0	1	
Cr-Zr		Z=0	7	9	18	17	0	0	0	$X_1 = X_2 = X_4 = X_7 = 0$ $X_5 = 2, X_6 = 1, X_3 = 8$ On divise $\frac{X_5}{1}, \frac{X_6}{1}, \frac{X_7}{\frac{1}{3}}$
P5	0	2	1/3	2/3	0	2	1	0	-5/3	
P6	0	1	1/3	-1/3	0	0	0	1	-2/3	
P3	18	8	1/3	2/3	1	1	0	0	1/3	
Cr-Zr		Z'0=144	1	-3	0	-1	0	0	-6	
P5	0	1	0	1	0	2	1	-1	-1	$X_1 = X_2 = X_4 = X_6 = X_7 = 0$ $X_1 = 3, X_3 = 7, X_5 = 1$ ($\overbrace{X_1}^{\text{Pivot}}, \overbrace{X_3}^{\text{Pivot}}, \overbrace{X_5}^{\text{Pivot}}$)
P1	7	3	1	-1	0	0	0	3	-2	
P3	18	7	0	1	1	1	0	-1	1	
Cr-Zr		Z'0=147	0	-2	0	-1	0	-3	-4	

5. RESULTAT

La solution basique associée $x_6 = x_2 = x_7 = x_4 = 0, x_5 = 1, x_1 = 3, x_3 = 7$, correspond au sommet (3,0,7,0) du polyèdre des contraintes. Elle définit le plan de production suivant:

On fabrique 3 unités de CHUIN et 7 unités de SIMBA. Il ne nous reste qu'une unité de la ressource A.

Le bénéfice est $z = 147$. De plus, tous les coefficients de la dernière ligne sont négatifs, donc on ne peut pas se déplacer vers un sommet voisin en augmentant la fonction du profit.

Nous avons alors trouvé la solution optimale du problème.

6. DESCRIPTION DU LANGAGE PYTHON

Python est un langage de programmation interprété, multi-paradigme et multiplateformes conçu par GUIDO VAN ROSSUM en 1990. Il favorise la programmation impérative structurée, fonctionnelle et orientée objet.

Python est un langage multiplateforme, c'est-à-dire disponible sur plusieurs architectures (compatible PC, certains smartphones, etc.) et systèmes d'exploitation (Windows, Linux, Mac, Androïde pour smartphones...). C'est un des langages informatiques les plus populaires avec C, C++, C#, Objective-C, Java, PHP, JavaScript, Delphi, Visual Basic, Ruby et Perl. Python est un langage qui peut s'utiliser dans de nombreux contextes et s'adapter à tout type d'utilisation grâce à des bibliothèques spécialisées. (SWINNEN G. 2010)

Avec Python on peut faire:

- des calculs scientifiques (bibliothèque numpy)
- des graphiques (bibliothèque matplotlib)

- du traitement du son
- du traitement d'image (bibliothèque PIL)
- des applications avec interface graphique GUI (bibliothèques Tkinter, PyQt, wxPython, PyGTK...)
- des jeux vidéo en temps réel (bibliothèque Pygame)
- des applications Web (serveur Web Zope; framework Web Django, Karrigell; framework JavaScript Pyjamas)
- interfacier des systèmes de gestion de base de données (bibliothèque MySQLdb...)
- des applications réseau (framework Twisted)
- communiquer avec des ports série RS232, Bluetooth... (bibliothèque PySerial)
-

Nous allons utiliser les bibliothèques numpy (numerical python) pour manipuler les tableaux et la bibliothèque Tkinter pour concevoir des interfaces graphiques.

Par conséquent, il faut les importer dans l'en-tête du programme comme suit:

```
import numpy as np
from Tkinter import *
```

np est un alias de la bibliothèque numpy, à la deuxième ligne nous avons importé toutes les classes de la bibliothèque Tkinter

Il est important de rappeler que Python utilise le typage dynamique, il vous suffit donc d'assigner une valeur à un nom de variable pour que celle-ci soit automatiquement créée avec le type qui correspond au mieux à la valeur fournie. Ce typage permet d'écrire aisément de constructions logiques de niveau élevé, comme la métaprogrammation ou la réflexivité, dans le contexte de la programmation orientée objet.

(TAREK Z. 2010) Les différentes opérations sur le multiplicateur de lagrangien sont regroupées dans des fonctions. La syntaxe Python pour la définition d'une fonction est la suivante:

```
def nomDeLaFonction (liste de paramètres):
```

...

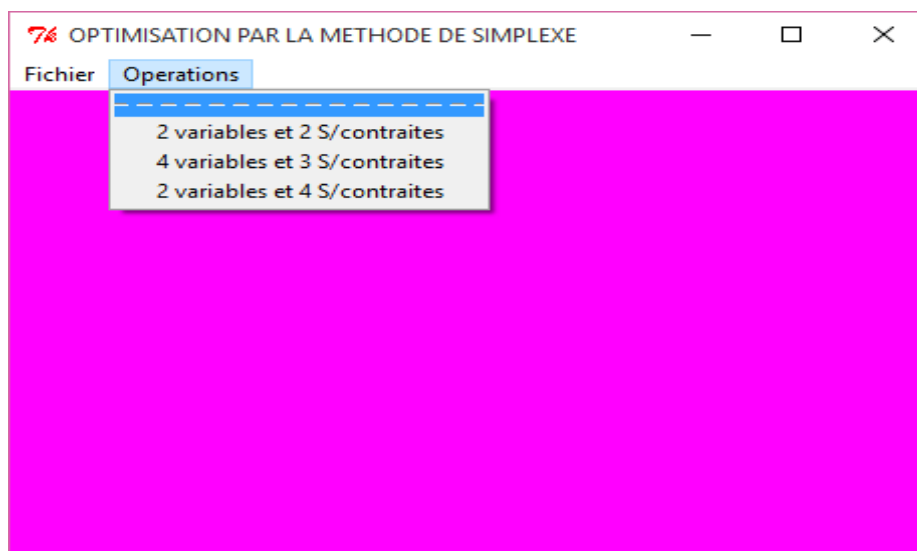
7. APPLICATION DE L'OPTIMISATION PAR LA METHODE DE SIMPLEXE AU LANGAGE PYTHON

7.1. INTERFACE GRAPHIQUE

L'interface en programmation est l'annonce de ce que l'on va faire. De nos jours, toutes les applications PC sont à base de GUI (Graphical User Interface) Interface d'Utilisateur Graphique. Le GUI permet à l'utilisateur de manipuler des icônes dans un environnement graphique à base de fenêtre (Windows) et offre une productivité beaucoup plus grande que ce qu'on peut obtenir dans un environnement à base de caractères DOS.

7.2. PRESENTATION DE L'APPLICATION

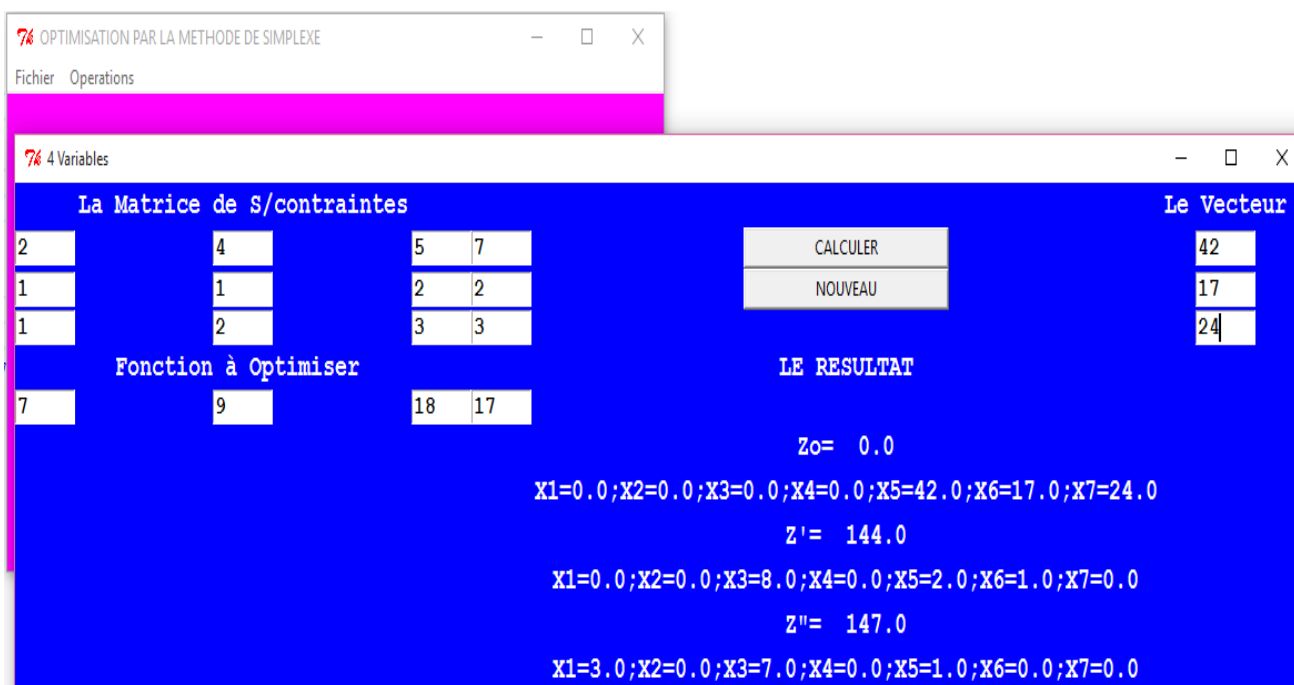
A ce point nous allons présenter quelques formulaires ou interface de notre application.



7.3. JEU D'ESSAI

	CHUIN	TEMBO	SIMBA	DOPPEL	Stock
Mais	2	4	5	7	42
Houblon	1	1	2	2	17
Malt	1	2	3	3	24
Bénéfice	7	9	18	17	

Dans la matrice de sous/contraintes est formée par les quantités des matières de 4 bières, la fonction à optimiser est formée à partir de la ligne des bénéfices et le vecteur est formé avec la colonne de stock.



7.4. CODES SOURCES

Un code source est un texte rédigé dans un langage assez compréhensible pour les humains et qui doit être traduit en code binaire pour être compris par les ordinateurs.

```
def CBase (C):
    k=0
    for j in range (n, m+n):
        CB [k] =C [j]
    return CB
def ChorsBase (C):
    for i in range (n):
        CN [i] =C [i]
    return CN
def Y (r):
    j=r
    for i in range (m):
        a [i] =A [i, j]
    return a
def identite (m):
    return np.eye (m)
P= [[A11, A12, A13, A14], [A21, A22, A23, A24], [A31, A32, A33, A34]]
#N=np.array ([[A11, A12, A13, A14], [A21, A22, A23, A24], [A31, A32, A33, A34]])
N=np.array (P)
b=np.array ([b1, b2, b3])
C=np.array ([C1, C2, C3, C4,0,0,0])
for i in range (m):
    for j in range (n):
        A [i, j] =N [i, j]
    for k in range (m):
        A [i, k+n] =B [i, k]
    CB=CBase (C)
    CN=ChorsBase (C)
    XB=np.zeros ((m), dtype='d')
    XB=np.dot (np.linalg.inv (B), b)
    z=np.dot (CB, XB)
    y=np.zeros ((m), dtype='d')
    zc=np.zeros ((n), dtype='d')
    for i in range (n):
```

```
y=np.dot (np.linalg.inv (B), Y (i))
zc [i] =np.dot (CB, y) -CN [i]
zcmin = zc.min ()
for i in range (m):
if zc [i] ==zcmin:
entree=i
k=entree
xbr=np.zeros ((m), dtype='d')
for i in range (m):
if A [i, k] >0:
xbr [i] =XB [i] /A [i, k]
x1=xbr.min ()
```

8. DISCUSSION

Cette étape d'interprétation des résultats et de mise en place de recommandation constitue la dernière étape de cette étude. Le simplexe est un algorithme d'optimisation utilisé depuis de nombreuses années afin de résoudre des problèmes de programmation linéaire. Ces problèmes sont utilisés dans la majorité des cas pour modéliser des problèmes. Le simplexe est souvent employé dans différentes industries pour des problèmes de planifications, distributions, agencements.

L'optimisation est une branche des mathématiques qui consiste à modéliser, à analyser et à résoudre analytiquement ou numériquement les problèmes de maximisation ou de minimisation d'une fonction sur un ensemble. Et l'optimisation d'une fonction linéaire d'un problème classique par la méthode de simplexe qui génère la fatigue, les erreurs et la perte de temps. Et nous avons choisi de coder nos algorithmes à l'aide du langage de programmation orienté objet PYTHON pour avoir une application informatique facilitant efficacement l'optimisation dans un laps de temps à l'aide de l'outil informatique.

9. CONCLUSION

Cette étude m'a donné l'opportunité de me familiariser au domaine de la recherche opérationnelle, ce domaine qui est la discipline des méthodes scientifiques pour aider à mieux décider et traiter les problèmes stratégiques et économiques, Le problème de production est l'un de ces problèmes classique les plus connus, Mais la complexité et la variation des contraintes de ce problème dans le domaine économique impliquent la recherche d'autres heuristiques et même des méta heuristiques plus efficaces pour la résolution. Ce qui rend difficile de tirer une conclusion définitive sur la résolution de ce type des problèmes.

Dans ce travail, on s'est intéressé davantage à l'optimisation et la résolution de problème de la production équilibré cas de la brasserie SIMBA par la méthode de simplexe qui nous permettrons d'obtenir une solution de base réalisable. Ensuite nous avons essayé de concevoir une application informatique en langage de programmation orienté objet Python de résolution d'un problème de production optimale par la méthode de simplexe.

A la fin je peux dire que cet article représente une base de départ pour résoudre les problèmes généraux de production.

REFERENCES

- [1] Karam ALLALI, recherche opérationnelle, introduction à la méthode du simplexe, fige 2009-2010, 29pages.
- [2] FAURE, R., LEMAIRE, B. et PICOULEAU, C. précis de recherche opérationnelle, éd. Dunod, Paris, 2009.
- [3] YADOLAH, D. Optimisation appliquée, éd. Springers, Paris, 2005.
- [4] Farouk BENGHEZAL, programmation linéaire, revues et augmentée, édition de l'office des publications universitaire, deuxième édition, Alger 2006.
- [5] George B. DANTZIG, Mukund TSHAPA, Linear programming, Introduction, Springer-verlag New York, LLC 1997, 435 pages.
- [6] Gerald BAILLARGEON, Programmation linéaire appliquée, outils d'optimisation et d'aide à la décision, copyright, 1996, édition SMG.
- [7] Vincent le GOFF, « Apprenez à programmer en python. Développer en python n'a jamais été facile ! » Eyrolles, 3eme Edition 2020.
- [8] Gérard Swinnem, « Apprendre à programmer avec Python 3", Copyright© 2000-2012.
- [9] Mr MAKREM Ben JEDDOU et Mme HABABOU HELLA: « Mathématiques appliquées à l'Économie et à la Gestion", université de Tunis 2008.
- [10] Stéphane MBUYI LAM, cours de programmation orientée objet PYTHON, deuxième licence en mathématiques informatiques, 2018-2019.