UNIVERSITÉ D'ANTANANARIVO
DOMAINE SCIENCES ET TECHNOLOGIES
MENTION MATHÉMATIQUES ET INFORMATIQUE

Mémoire en vue de l'obtention du diplôme de Master
Parcours : Mathématiques Appliquées
Spécialité : Combinatoire et Optimisation

UNE AUTRE MÉTHODE DE RÉSOLUTION DES PROGRAMMES LINÉAIRES EN NOMBRES ENTIERS

présenté par : Mademoiselle RAMONJISON Domoina Mahefasoa

soutenu le 02 novembre 2017

devant le jury composé de :

Président :	Monsieur RANDRIAMBOLOLONDRANTOMALALA Princy	Professeur
Examinateur :	Monsieur RAMAMONJISOA Armand	Maître de Conférences
Encadreur :	Monsieur RANDRIANARIVONY Arthur	Professeur titulaire

UNIVERSITÉ D'ANTANANARIVO
DOMAINE SCIENCES ET TECHNOLOGIES
MENTION MATHÉMATIQUES ET INFORMATIQUE

Mémoire en vue de l'obtention du diplôme de Master
Parcours : Mathématiques Appliquées
Spécialité : Combinatoire et Optimisation

UNE AUTRE MÉTHODE DE RÉSOLUTION DES PROGRAMMES LINÉAIRES EN NOMBRES ENTIERS

présenté par : Mademoiselle RAMONJISON Domoina Mahefasoa

soutenu le 02 novembre 2017

devant le jury composé de :

Président :	Monsieur RANDRIAMBOLOLONDRANTOMALALA Princy	Professeur
Examinateur :	Monsieur RAMAMONJISOA Armand	Maître de Conférences
Encadreur :	Monsieur RANDRIANARIVONY Arthur	Professeur titulaire

Je dédie ce mémoire :

À DIEU, qui m'a donné la volonté et le courage nécessaires pour la réalisation de ce travail.

À ma famille, qui a œuvré pour ma réussite, par son amour et son soutien inconditionnel, par son réconfort et tous les sacrifices consentis, par sa présence dans ma vie.

À ceux et celles qui, par leur amour et leur dévouement, n'ont jamais cessé de m'encourager et de me soutenir.

REMERCIEMENTS

*Comme le disait Ovide : " Les petits ruisseaux font les grandes rivières ".
Je tiens donc à adresser mes vifs remerciements aux personnes qui ont contribué, de près ou de loin à l'accomplissement de mon mémoire de fin d'études.*

Ces remerciements s'adressent tout particulièrement à :

Monsieur RANDRIAMBOLOLONDRANTOMALALA Princy, Professeur, pour l'intérêt qu'il a bien voulu porter à ce travail et pour m'avoir fait l'honneur de présider le jury.

Monsieur RAMAMONJISOA Armand, Maître de Conférences, pour m'avoir fait l'honneur d'accepter la charge d'examinateur.

Monsieur RANDRIANARIVONY Arthur, Professeur titulaire, pour l'enthousiasme et le dynamisme dont il a fait preuve tout au long de cet encadrement. Il m'a permis de surmonter les moments difficiles grâce à sa disponibilité et à son très grand intérêt pour mon travail. Je tiens beaucoup à le remercier pour ses idées, ses conseils et sa détermination qui m'ont permis d'élargir le travail réalisé lors de l'élaboration de ce mémoire.

L'ensemble des enseignants et personnels au sein de la Mention Mathématiques et Informatique, qui m'ont aidée dans mon parcours universitaire ainsi que dans l'aboutissement de ce mémoire.

Tous les membres de ma famille et tous mes proches, pour leur amour et leur soutien.

Tous mes amis, pour nos bons moments passés ensemble et pour notre solidarité tout au long de cette année universitaire.

Recevez l'expression de ma profonde gratitude et reconnaissance.

Sommaire

Introduction

De nombreux problèmes rencontrés dans des situations réelles sont en fait des problèmes d'optimisation. Il n'est même pas nécessaire d'aller chercher loin, la vie quotidienne elle-même en est remplie : c'est déjà optimiser que de bien gérer son revenu mensuel, de disposer efficacement les produits dans les rayons d'un supermarché, ou encore de réaliser le maximum de bénéfices dans la gestion d'une épicerie.

Il est vrai que dans la vie de tous les jours, ces petits problèmes sont résolus instinctivement sans passer par de grandes analyses mathématiques. Mais en revanche, pour des problèmes à grande échelle nécessitant de grandes précisions dans leur résolution, la modélisation mathématique est primordiale. Une grande industrie de production, dans le but de faire le maximum de profits, ne peut pas se permettre par exemple, de décider d'une manière quelconque de la quantité idéale de production étant données toutes les contraintes et la forte somme d'argent à investir ; une analyse préalable minutieuse en tenant compte de tout ce qui entre en jeu est de rigueur. De nombreux algorithmes ont donc été élaborés afin de résoudre efficacement et dans le moins de temps possible ces types de problèmes.

Parmi tous les problèmes d'optimisation, les problèmes de programmation linéaire sont les plus faciles, puisque la fonction à optimiser et les contraintes sont toutes linéaires. Beaucoup de problèmes réels peuvent être exprimés comme un problème de programmation linéaire d'où l'importance de leur résolution par des algorithmes efficaces et diversifiés. Néanmoins, l'algorithme du simplexe est le plus célèbre et le plus efficace dans le cas général. Dans le cas où les variables de décision du programme linéaire sont astreintes à être entières, ce qui est le cas pour la majorité des problèmes dans la réalité, le programme linéaire est dit programme linéaire en nombres entiers ou encore PLNE.

Ce mémoire se focalise notamment sur la résolution de ces programmes linéaires en nombres entiers. Il comprend trois parties : la première partie comporte des rappels nécessaires sur les programmes linéaires en général, dans la deuxième partie sont exposées deux méthodes usuelles de résolution des programmes linéaires en nombres entiers et finalement, la dernière partie présente une amélioration intéressante de la méthode du simplexe pour la résolution des programmes linéaires en nombres entiers.

Première partie

Rappels nécessaires sur la programmation linéaire

1.1 Quelques définitions

1.1.1 Définition d'un programme linéaire

Un programme linéaire (PL) est un problème d'optimisation consistant à optimiser, c'est-à-dire soit à maximiser soit à minimiser, une fonction linéaire de n variables de décision appelée fonction objectif. Et ces n variables de décision sont soumises à un ensemble de contraintes exprimées sous forme d'équations ou d'inéquations linéaires.

Une solution d'un programme linéaire est l'affectation de valeurs aux variables de décision du problème. Cette solution est dite réalisable si elle vérifie toutes les contraintes du programme. Et l'ensemble des solutions réalisables est appelé domaine de réalisabilité. Ce domaine, s'il n'est pas vide, est convexe. Une solution optimale d'un programme linéaire est une solution réalisable optimisant la fonction objectif.

Puisqu'un problème de minimisation linéaire est traité de manière analogue à un problème de maximisation linéaire à la différence près qu'au lieu de faire en sorte d'augmenter la valeur de la fonction objectif, il faut faire en sorte de la diminuer, il n'est pas restrictif de ne considérer que des problèmes de maximisation dans toute la suite.

Un programme linéaire est dit sous forme canonique s'il est du type

$$\begin{cases} max\ z = c_1x_1 + \ldots + c_nx_n \\ a_{11}x_1 + \ldots + a_{1n}x_n \leqslant b_1 \\ \vdots \\ a_{m1}x_1 + \ldots + a_{mn}x_n \leqslant b_m \\ \forall i,\ x_i \geqslant 0 \end{cases}$$

Les contraintes $\forall i,\ x_i \geqslant 0$, sont appelées contraintes de non-négativité ou encore contraintes de positivité.

En posant $c = (c_1, \ldots, c_n)$, $x = {}^t(x_1, \ldots, x_n)$

$$A = \begin{pmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{m1} & \cdots & a_{mn} \end{pmatrix} \text{ où } A \in \mathscr{M}_{m,n}(\mathbb{R}) \text{ et } b = \begin{pmatrix} b_1 \\ \vdots \\ b_m \end{pmatrix}$$

La forme matricielle du (PL) sous forme canonique s'écrit :

$$\begin{cases} max\ z = cx \\ Ax \leqslant b \\ x \geqslant 0 \end{cases}$$

Un programme linéaire mis sous forme standard est du type :

$$\begin{cases} max\ z = cx \\ Ax = b \\ x \geqslant 0\ ,\ b \geqslant 0 \end{cases}$$

Un programme linéaire sous forme canonique peut être mis sous forme standard en introduisant de nouvelles variables dites variables d'écart.

Exemple : pour $n = 3$, $x_1 + 3x_2 - x_3 \geqslant 3 \iff x_1 + 3x_2 - x_3 - x_4 = 3$ avec la variable d'écart $x_4 \geqslant 0$

1.1.2 Base d'un programme linéaire mis sous forme standard

Dans toute la suite, A est une matrice rectangulaire à m lignes et n colonnes où $m \leqslant n$, m étant le rang de A. Un ensemble $\left\{x_{i_1}, ..., x_{i_m}\right\}$ est une base du programme linéaire s'il est possible d'exprimer chaque x_{i_l}, l de 1 à m, en fonction des autres variables. Dans ce cas, les x_{i_l}, pour l de 1 à m, sont appelés variables de base. Soit à noter $x_{j_1}, ..., x_{j_{n-m}}$ les autres variables, appelées variables hors base alors pour tout l de 1 à m, x_{i_l} peut s'écrire : $x_{i_l} = \bar{b}_l + \sum_{k=1}^{n-m} \bar{a}_{l,j_k} x_{j_k}$.
Si $\bar{b}_l \geqslant 0$, $\forall l$, $1 \leqslant l \leqslant m$ alors la base $\left\{x_{i_1}, ..., x_{i_m}\right\}$ est dite réalisable.

Soit B une sous-matrice de A, à m lignes et m colonnes. B est dite régulière si elle est inversible.

Posons
$$\begin{aligned}
A &= [A_1, ..., A_n] \\
B &= \left[A_{i_1}, ..., A_{i_m}\right] \\
N &= \left[A_{j_1}, ..., A_{j_{n-m}}\right] \text{ où } \{j_1, ..., j_{n-m}\} \text{ est le complémentaire de } \{i_1, ..., i_m\} \text{ dans } \{1, 2, ..., n\} \\
x_B &= {}^t(x_{i_1}, ..., x_{i_m}) \\
x_N &= {}^t(x_{j_1}, ..., x_{j_{n-m}}) \\
c_B &= (c_{i_1}, ..., c_{i_m}) \\
c_N &= (c_{j_1}, ..., c_{j_{n-m}})
\end{aligned}$$

Alors $Ax = b$ devient $Bx_B + Nx_N = b$. Et si B est régulière, alors on a : $x_B + B^{-1}Nx_N = B^{-1}b$ ou encore $x_B = B^{-1}b - B^{-1}Nx_N$.

Si $B^{-1}b \geqslant 0$, alors les variables $x_{i_1}, ..., x_{i_m}$ sont dites variables de base réalisables.
Supposons que $B^{-1}b \geqslant 0$, alors la fonction objectif $z = cx$ devient :
$z = c_B x_B + c_N x_N$ ou encore $z = c_B(B^{-1}b - B^{-1}Nx_N) + c_N x_N$
$z = c_B B^{-1}b + (c_N - c_B B^{-1}N)x_N$
En notant $\bar{z}_0 = c_B B^{-1}b$ et $\bar{c}_N = c_N - c_B B^{-1}N$,
alors on a : $z = \bar{z}_0 + \bar{c}_N x_N = \bar{z}_0 + \bar{c}_{j_1} x_{j_1} + ... + \bar{c}_{j_{n-m}} x_{j_{n-m}}$, c'est-à-dire, la fonction objectif z s'écrit en fonction des variables hors base. Les $\bar{c}_r$, $r \in \{j_1, ..., j_{n-m}\}$, sont appelés coûts réduits.

1.1.3 Forme dictionnaire d'un programme linéaire

Soit à considérer le programme linéaire noté (PL) suivant :
$$(PL)\left\{\begin{array}{l} max\ z = c_1x_1 + ... + c_nx_n \\ Ax = b \\ x \geqslant 0 \end{array}\right.$$

Supposons qu'il existe B une sous-matrice d'ordre m régulière de A alors comme vu précédemment, $x_B = B^{-1}b - B^{-1}Nx_N$.

Le système :
$$\left\{\begin{array}{c} z = \bar{z}_0 + \bar{c}_{j_1} x_{j_1} + ... + \bar{c}_{j_{n-m}} x_{j_{n-m}} \\ x_{i_1} = \bar{b}_1 + \sum_{k=1}^{n-m} \bar{a}_{1,j_k} x_{j_k} \\ \vdots \\ x_{i_m} = \bar{b}_m + \sum_{k=1}^{n-m} \bar{a}_{m,j_k} x_{j_k} \end{array}\right.$$

est un dictionnaire de (PL). Il est réalisable si $\bar{b} = \begin{pmatrix} \bar{b}_1 \\ \vdots \\ \bar{b}_m \end{pmatrix}$ est positif ou nul.

Dans le cas où le programme linéaire est sous forme canonique, donc de la forme : $\left\{\begin{array}{l} max\ z = cx \\ Ax \leqslant b \\ x \geqslant 0\ ,\ b \geqslant 0 \end{array}\right.$

Alors les variables d'écart sont directement prises comme variables de base initiales.

Et le système : $\begin{cases} z = c_1x_1 + ... + c_nx_n \\ x_{n+1} = b_1 - \sum_{k=1}^{n} a_{1k}x_k \\ \vdots \\ x_{n+m} = b_m - \sum_{k=1}^{n} a_{mk}x_k \end{cases}$

est un dictionnaire réalisable du programme linéaire ci-dessus. S'il existe un $b_i < 0$, ce dictionnaire n'est plus réalisable.

1.1.4 Forme tableau d'un programme linéaire

Soit toujours à considérer le programme linéaire (PL). Supposons qu'il existe B une sous-matrice d'ordre m régulière de A.

On avait : $x_B = B^{-1}b - B^{-1}Nx_N = \bar{b} - B^{-1}Nx_N$.

Supposons de plus que $\bar{b} \geqslant 0$, les variables de base de x_B sont alors réalisables. Sans perdre la généralité, il est aussi possible de supposer que la matrice B est formée par les m premiers vecteurs colonnes de la matrice A, c'est-à-dire, les variables $x_1, ..., x_m$ sont les variables de base initiales. On a :

$$\begin{cases} z = \bar{z}_0 + \bar{c}_1x_{m+1} + ... + \bar{c}_{n-m}x_n \\ x_1 + \sum_{k=m+1}^{n} \bar{a}_{1k}x_k = \bar{b}_1 \\ \vdots \\ x_m + \sum_{k=m+1}^{n} \bar{a}_{mk}x_k = \bar{b}_m \end{cases}$$

avec $z = \bar{z}_0 + \bar{c}_1x_{m+1} + ... + \bar{c}_{n-m}x_n \Leftrightarrow \bar{c}_1x_{m+1} + ... + \bar{c}_{n-m}x_n - z = -\bar{z}_0$

Le tableau initial réalisable de (PL) est donc :

x_1	x_2	$\cdots$	x_m	x_{m+1}	$\cdots$	x_n	z	
0	0	$\cdots$	0	$\bar{c}_1$	$\cdots$	$\bar{c}_{n-m}$	-1	$-\bar{z}_0$
1	0	$\cdots$	0	$\bar{a}_{1,m+1}$	$\cdots$	$\bar{a}_{1,n}$	0	$\bar{b}_1$
0	1	$\cdots$	0	$\bar{a}_{2,m+1}$	$\cdots$	$\bar{a}_{2,n}$	0	$\bar{b}_2$
$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$	$\cdots$	$\vdots$	$\vdots$	$\vdots$
0	0	$\cdots$	1	$\bar{a}_{m,m+1}$	$\cdots$	$\bar{a}_{m,n}$	0	$\bar{b}_m$

Remarques :

— Il est à remarquer qu'à la différence des tableaux usuels du simplexe, le tableau ci-dessus possède une colonne supplémentaire, celle de z. Cette colonne a été ajoutée pour simplifier les calculs lors de la résolution du problème. C'est effectivement afin de ne pas avoir à rendre à 1 les pivots et de ce fait, pour éviter de manipuler des fractions.

— Du fait que la forme tableau est plus pratique pour les calculs que la forme dictionnaire, alors ce sera la forme utilisée dans les résolutions pour tout ce qui va suivre.

1.2 Méthode du simplexe

La méthode du simplexe est une méthode de résolution des problèmes d'optimisation linéaire. C'est une méthode itérative parcourant les sommets du domaine de réalisabilité du problème jusqu'à ce que la fonction objectif ne puisse plus être optimisée.

Avant tout, il faut transformer les contraintes inégalitaires du programme, autres que les contraintes de positivité, en contraintes égalitaires. Si le programme linéaire est donc sous forme canonique, il faut le mettre sous forme standard.

Les transformations relatives à l'algorithme du simplexe ne s'appliquent qu'à un dictionnaire réalisable ou à un tableau réalisable. Lorsque le dictionnaire (respectivement le tableau) n'est pas réalisable, il faut passer par une première phase consistant à rechercher une base réalisable du programme linéaire. Pour cela, il faut résoudre le programme linéaire :

— formé des contraintes suivantes : les contraintes telles qu'à chacune d'elles a été ajoutée une variable artificielle positive ou nulle, notamment puisque chacune d'elles est associée à l'expression d'une variable de base non réalisable, les contraintes n'ayant subi aucun ajout mais justes reprises, les contraintes de positivité des variables artificielles,

— et dont la fonction objectif à maximiser est l'opposé de la somme des variables artificielles.

À la fin de cette phase, si la base réalisable existe alors la valeur de chaque variable artificielle est nécessairement nulle.

Dans la forme dictionnaire

Supposons qu'on ait le dictionnaire réalisable de (PL) suivant :

$$\begin{cases} z = \bar{z}_0 + \bar{c}_{j_1} x_{j_1} + \ldots + \bar{c}_{j_{n-m}} x_{j_{n-m}} \\ x_{i_1} = \bar{b}_1 + \sum_{k=1}^{n-m} \bar{a}_{1,j_k} x_{j_k} \\ \vdots \\ x_{i_m} = \bar{b}_m + \sum_{k=1}^{n-m} \bar{a}_{m,j_k} x_{j_k} \end{cases}$$

Si tous les coûts réduits $\bar{c}_{j_1}, \ldots, \bar{c}_{j_{n-m}}$ sont tous négatifs ou nuls alors $\bar{z}_0$ est la valeur maximale de z et $x^* = (x_1^*, \ldots, x_n^*)$ où $x_{i_l}^* = \bar{b}_l$ pour tout l de 1 à m et $x_{j_k}^* = 0$ pour tout k de 1 à $n-m$ est une solution optimale de (PL).

Par contre, si l'ensemble E des coûts réduits positifs stricts n'est pas vide alors voici comment procéder :

Soit $\bar{c}_r \in E$, $r \in \{j_1, \ldots, j_{n-m}\}$, alors si x_r augmente, la fonction z augmente aussi. Pour augmenter la valeur de x_r, il faut faire entrer x_r en base (x_r est alors appelée variable entrante) et faire sortir une variable de base (cette variable de base qui sort est appelée variable sortante). Pour cela, il faut considérer tous les coefficients négatifs de x_r dans le dictionnaire.

★ Supposons qu'il y ait deux coefficients négatifs de x_r, soient $\bar{a}_{pr}$ et $\bar{a}_{lr}$ ces deux coefficients. Alors voici les deux contraintes du dictionnaire à considérer :

$$x_{i_p} = \bar{b}_p + \bar{a}_{pr} x_r + \sum_{\substack{k=1 \\ j_k \neq r}}^{n-m} \bar{a}_{p,j_k} x_{j_k} \quad (1)$$

$$x_{i_l} = \bar{b}_l + \bar{a}_{lr} x_r + \sum_{\substack{k=1 \\ j_k \neq r}}^{n-m} \bar{a}_{l,j_k} x_{j_k} \quad (2)$$

De l'équation (1) s'obtient l'équation suivante :

$$x_r = \frac{x_{i_p}}{\bar{a}_{pr}} - \frac{\bar{b}_p}{\bar{a}_{pr}} - \sum_{\substack{k=1 \\ j_k \neq r}}^{n-m} \frac{\bar{a}_{p,j_k}}{\bar{a}_{pr}} x_{j_k} \quad (3)$$

En portant (3) dans (2) :

$$x_{i_l} = -\bar{a}_{lr}\left(\frac{\bar{b}_p}{\bar{a}_{pr}} - \frac{\bar{b}_l}{\bar{a}_{lr}}\right) + \frac{\bar{a}_{lr}}{\bar{a}_{pr}} x_{i_p} + \sum_{\substack{k=1 \\ j_k \neq r}}^{n-m} \left(\bar{a}_{l,j_k} - \frac{\bar{a}_{lr}\bar{a}_{p,j_k}}{\bar{a}_{pr}}\right) x_{j_k}$$

Pour que le nouveau dictionnaire soit réalisable, il faut que $\frac{\bar{b}_p}{\bar{a}_{pr}} \geqslant \frac{\bar{b}_l}{\bar{a}_{lr}}$. Ainsi, il faut choisir p tel que $\frac{\bar{b}_p}{-\bar{a}_{pr}} \leqslant \frac{\bar{b}_l}{-\bar{a}_{lr}}$ pour tout l tel que $\bar{a}_{lr} < 0$. Dans ce cas, x_{i_p} est la variable sortante. En remplaçant x_{i_p} dans la base précédente par x_r, la nouvelle base du nouveau dictionnaire est connue.

Par la suite, il faut également remplacer x_r par sa valeur (3) dans l'expression de z de sorte que la nouvelle expression de z s'exprime en fonction des variables hors base.

Si tous les coûts réduits dans la nouvelle expression de z sont négatifs ou nuls, alors la solution optimale est obtenue. Sinon, il faut poursuivre comme précédemment.

★ Si tous les coefficients de la variable entrante x_r sont positifs ou nuls, alors la solution du problème est non bornée. En effet, voici d'abord le dictionnaire réalisable en question :

$$\begin{cases} z = \bar{z}_0 + \bar{c}_{j_1} x_{j_1} + \ldots + \bar{c}_{j_{n-m}} x_{j_{n-m}} \\ x_{i_1} = \bar{b}_1 + \bar{a}_{1r} x_r + \sum_{\substack{k=1 \\ j_k \neq r}}^{n-m} \bar{a}_{1,j_k} x_{j_k} \\ \qquad \vdots \\ x_{i_m} = \bar{b}_m + \bar{a}_{mr} x_r + \sum_{\substack{k=1 \\ j_k \neq r}}^{n-m} \bar{a}_{m,j_k} x_{j_k} \end{cases}$$

où $\bar{c}_r > 0$ car $\bar{c}_r \in E$ et où tous les $\bar{a}_{ir}$ pour tout i de 1 à m sont positifs ou nuls. En supposant que tous les $\bar{b}_i$ pour tout i de 1 à m soient positifs, alors pour tout $\alpha > 0$, la solution $x^* = (x_1^*, \ldots, x_n^*)$ telle que $x_{i_l}^* = \bar{b}_l + \bar{a}_{lr}\alpha$ pour tout l de 1 à m, $x_s^* = 0$ pour tout s différent de $i_1, \ldots, i_m, r$ et $x_r^* = \alpha$, est une solution réalisable du problème. La valeur de z correspondant à cette solution est $z^* = \bar{z}_0 + \bar{c}_r\alpha$ avec $\bar{c}_r > 0$ et en faisant tendre α vers l'infini, z^* tend également vers l'infini. La solution du problème est donc non bornée.

Dans la forme tableau

Supposons qu'on ait le tableau réalisable de (PL) suivant :

x_{i_1}	x_{i_2}	$\cdots$	x_{i_m}	x_{j_1}	$\cdots$	$x_{j_{n-m}}$	z	
0	0	$\cdots$	0	$\bar{c}_{j_1}$	$\cdots$	$\bar{c}_{j_{n-m}}$	-1	$-\bar{z}_0$
1	0	$\cdots$	0	$\bar{a}_{1,j_1}$	$\cdots$	$\bar{a}_{1,j_{n-m}}$	0	$\bar{b}_1$
0	1	$\cdots$	0	$\bar{a}_{2,j_1}$	$\cdots$	$\bar{a}_{2,j_{n-m}}$	0	$\bar{b}_2$
$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$	$\cdots$	$\vdots$	$\vdots$	$\vdots$
0	0	$\cdots$	1	$\bar{a}_{m,j_1}$	$\cdots$	$\bar{a}_{m,j_{n-m}}$	0	$\bar{b}_m$

Tant que la ligne des coûts réduits contient un coût réduit positif, il faut procéder comme suit :

— Déterminer la colonne pivot j_k. Cette colonne est choisie parmi les colonnes correspondant aux coûts réduits positifs.

— Ensuite, choisir un pivot dans la colonne pivot j_k. C'est un élément du tableau noté par exemple $\bar{a}_{i,j_k}$ dans la colonne pivot j_k, choisi de façon à ce que $\frac{\bar{b}_i}{\bar{a}_{i,j_k}}$ soit positif ou nul et minimal (c'est-à-dire $\frac{\bar{b}_i}{\bar{a}_{i,j_k}} \leqslant \frac{\bar{b}_l}{\bar{a}_{l,j_k}}$ pour tout $l \neq i$, l allant de 1 à m). Si un tel pivot n'existe pas, le problème est non borné.

Remarque : Le choix de la colonne pivot est arbitraire mais dans cet ouvrage, afin de maximiser plus rapidement la fonction objectif, ce choix se fera de la manière suivante :

➢ Il faut d'abord repérer le pivot de chaque colonne correspondant à un coût réduit positif.

➢ Puis pour chacune de ces colonnes, il faut effectuer le produit du coût réduit et du rapport minimal qui a permis de déterminer le pivot de la colonne en question.

➢ La colonne pivot sera celle pour laquelle ce produit est maximal et le pivot de cette colonne sera donc le vrai pivot du tableau.

— Remplacer chaque ligne du tableau par une combinaison linéaire de la ligne en question et de la ligne pivot, de sorte à annuler tous les termes de la colonne pivot, autres que le pivot. Rappelons que la ligne pivot est la ligne où se trouve le pivot.

Deuxième partie

Méthodes classiques de résolution d’un programme linéaire en nombres entiers

2.1 Méthode des coupes de Gomory

Soit à résoudre le programme linéaire en nombres entiers (P) suivant :

$$(P)\begin{cases} max\ z = c_1x_1 + ... + c_nx_n \\ a_{11}x_1 + ... + a_{1n}x_n \leqslant b_1 \\ \quad\vdots \\ a_{k1}x_1 + ... + a_{kn}x_n \leqslant b_k \\ \quad\vdots \\ a_{m1}x_1 + ... + a_{mn}x_n \leqslant b_m \\ \forall i,\ x_i \in \mathbb{N} \end{cases}$$

Les contraintes : $\forall i,\ x_i \in \mathbb{N}$, sont appelées contraintes d'intégrité. C'est la présence en plus, de ces contraintes qui fait de (P) un programme linéaire en nombres entiers. Le programme linéaire sans ces contraintes est dit programme linéaire relaxé.

Le principe de la méthode des coupes de Gomory est d'introduire de nouvelles contraintes linéaires au problème pour réduire le domaine de réalisabilité du problème relaxé sans pour autant éliminer des points du domaine discret de réalisabilité du programme linéaire en nombres entiers (P).

2.1.1 Description de la méthode des coupes de Gomory

Après avoir résolu le programme linéaire relaxé à l'aide de l'algorithme du simplexe, une solution optimale est obtenue. Si elle est entière, c'est la solution optimale du programme linéaire en nombres entiers (P). Sinon, la résolution de (P) n'est pas encore achevée. Soit à noter x^* la solution optimale non entière, $x^* = (x_1^*, ..., x_m^*, ..., x_n^*)$.

Sans perdre la généralité, supposons que les variables de base, à l'issue de la méthode du simplexe, soient les m premières variables c'est-à-dire $x_1, x_2, ..., x_m$ et que dans le tableau final, on ait, en faisant varier k de 1 à m, les équations de la forme : $x_k + \sum_{p=m+1}^{m+n} \bar{a}_{kp}x_p = \bar{b}_k$. La solution optimale x^* est donc telle que $x_i^* = \bar{b}_i$ pour i variant de 1 à m et $x_l^* = 0$ pour $l \geqslant m+1$. Pour tout réel x, $[x]$ désignera le plus grand entier inférieur ou égal à x et $\{x\}$ désignera la partie décimale de x alors $x = \{x\} + [x]$.

Les coupes de Gomory se déduisent des équations du tableau final comme suit :
Pour k, tel que, $1 \leqslant k \leqslant m$,

$$x_k + \sum_{p=m+1}^{m+n} \bar{a}_{kp}x_p = \bar{b}_k$$

$$\Updownarrow$$

$$x_k + \sum_{p=m+1}^{m+n}[\bar{a}_{kp}]x_p + \sum_{p=m+1}^{m+n}\{\bar{a}_{kp}\}x_p = [\bar{b}_k] + \{\bar{b}_k\}$$

Comme

$$\sum_{p=m+1}^{m+n}[\bar{a}_{kp}]x_p \leqslant \sum_{p=m+1}^{m+n}\bar{a}_{kp}x_p$$

Alors

$$x_k + \sum_{p=m+1}^{m+n}[\bar{a}_{kp}]x_p \leqslant x_k + \sum_{p=m+1}^{m+n}\bar{a}_{kp}x_p$$

Ou encore

$$x_k + \sum_{p=m+1}^{m+n} [\bar{a}_{kp}]x_p \leqslant \bar{b}_k$$

En tenant compte des contraintes d'intégrité et puisque $[\bar{b}_k] \leqslant \bar{b}_k$,

on a :

$$x_k + \sum_{p=m+1}^{m+n} [\bar{a}_{kp}]x_p \leqslant [\bar{b}_k]$$

Ainsi,

$$\sum_{p=m+1}^{m+n} \{\bar{a}_{kp}\} x_p \geqslant \{\bar{b}_k\}$$

Cette dernière inégalité obtenue pour un certain k est appelée coupe de Gomory. En faisant varier k de 1 à m, il y a donc au plus m coupes de Gomory.

Il faut par la suite résoudre à l'aide de la méthode du simplexe le nouveau programme linéaire obtenu en ajoutant au programme linéaire précédent ces coupes de Gomory. Ce mécanisme de rajout des coupes de Gomory ne s'arrête que lorsque la solution optimale obtenue est entière.

2.1.2 Exemple d'application

Le problème du sac à dos fait partie des problèmes d'optimisation les plus étudiés ces longues dernières années en raison de ses nombreuses applications dans le monde réel. Il modélise toute situation analogue au remplissage d'un sac à dos, ne pouvant supporter plus d'un certain poids, avec tout ou partie d'un ensemble donné d'objets ayant chacun un poids et une valeur et tels que les objets mis dans le sac à dos doivent, sans dépasser le poids maximum, maximiser la valeur totale. C'est un problème linéaire en 0-1 ou encore un problème en variables bivalentes, c'est-à-dire, chacune des variables de décision du problème ne peut prendre comme valeur que 0 ou 1. Le problème du sac à dos intervient souvent comme sous-problème dans plusieurs domaines : le chargement d'avions ou de bateaux par exemple dans le domaine de la logistique, la gestion de portefeuilles dans le domaine de l'économie ou encore la découpe de matériaux dans le domaine de l'industrie.
Vu l'intérêt que les problèmes de sac à dos suscitent, il semble intéressant d'en prendre un comme exemple d'application. Cet exemple qui va suivre, sera de plus, traité avec toutes les autres méthodes présentées dans cet ouvrage.

Énoncé du problème :
Soit à remplir un sac ne pouvant supporter que 14kg, par un deux ou trois objets parmi les trois objets O_1, O_2, O_3 de poids respectifs 4 kg, 6 kg, 8 kg et de valeurs respectives 6, 8, 7, tout en maximisant la valeur totale des objets placés dans le sac. Soit x_i la variable égale au nombre d'objet O_i placé dans le sac, $1 \leqslant i \leqslant 3$.

Modélisation du problème :
La fonction objectif à maximiser est : $z = 6x_1 + 8x_2 + 7x_3$.

Les variables x_i doivent évidemment être des entiers, $\forall i,\ x_i \in \mathbb{N}$.
Les contraintes du problème peuvent être représentées par le système d'inéquations suivant :

$$\begin{cases} 4x_1 + 6x_2 + 8x_3 \leqslant 14 \\ \forall i,\ x_i \in \{0,1\} \end{cases}$$

Résoudre ce problème revient donc à résoudre le programme linéaire en nombres entiers (P) suivant :

$$(P) \begin{cases} max\ z = 6x_1 + 8x_2 + 7x_3 \\ 4x_1 + 6x_2 + 8x_3 \leqslant 14 \\ x_1 \leqslant 1 \\ x_2 \leqslant 1 \\ x_3 \leqslant 1 \\ \forall i,\ x_i \in \mathbb{N} \end{cases}$$

<u>Résolution de (P) :</u>
Soit d'abord à résoudre, par la méthode du simplexe, le programme linéaire relaxé :

$$\begin{cases} max\ z = 6x_1 + 8x_2 + 7x_3 \\ 4x_1 + 6x_2 + 8x_3 \leqslant 14 \\ x_1 \leqslant 1 \\ x_2 \leqslant 1 \\ x_3 \leqslant 1 \\ \forall i,\ x_i \geqslant 0 \end{cases}$$

Sa forme standard :

$$\begin{cases} max\ z = 6x_1 + 8x_2 + 7x_3 \\ 4x_1 + 6x_2 + 8x_3 + x_4 = 14 \\ x_1 + x_5 = 1 \\ x_2 + x_6 = 1 \\ x_3 + x_7 = 1 \\ \forall i,\ x_i \geqslant 0 \end{cases}$$

Dans notre cas, la première phase c'est-à-dire la recherche de base réalisable n'est pas nécessaire puisque les variables d'écart $x_4,\ x_5,\ x_6,\ x_7$ constituent déjà une base réalisable. La résolution se fera sous forme tableau et les pivots seront encadrés.

x_1	x_2	x_3	x_4	x_5	x_6	x_7	z	
6	8	7	0	0	0	0	-1	0
4	6	8	1	0	0	0	0	14
1	0	0	0	1	0	0	0	1
0	$\boxed{1}$	0	0	0	1	0	0	1
0	0	1	0	0	0	1	0	1

x_1	x_2	x_3	x_4	x_5	x_6	x_7	z	
6	0	7	0	0	-8	0	-1	-8
4	0	8	1	0	-6	0	0	8
1	0	0	0	1	0	0	0	1
0	1	0	0	0	1	0	0	1
0	0	$\boxed{1}$	0	0	0	1	0	1

x_1	x_2	x_3	x_4	x_5	x_6	x_7	z	
6	0	0	0	0	-8	-7	-1	-15
$\boxed{4}$	0	0	1	0	-6	-8	0	0
1	0	0	0	1	0	0	0	1
0	1	0	0	0	1	0	0	1
0	0	1	0	0	0	1	0	1

x_1	x_2	x_3	x_4	x_5	x_6	x_7	z	
0	0	0	-6	0	4	20	-4	-60
4	0	0	1	0	-6	-8	0	0
0	0	0	-1	4	6	$\boxed{8}$	0	4
0	4	0	0	0	4	0	0	4
0	0	4	0	0	0	4	0	4

x_1	x_2	x_3	x_4	x_5	x_6	x_7	z	
0	0	0	-28	-80	-88	0	-32	-560
32	0	0	0	32	0	0	0	32
0	0	0	-1	4	6	8	0	4
0	32	0	0	0	32	0	0	32
0	0	32	4	-16	-24	0	0	16

x_1	x_2	x_3	x_4	x_5	x_6	x_7	z	
0	0	0	-7	-20	-22	0	-8	-140
1	0	0	0	1	0	0	0	1
0	0	0	-1	4	6	8	0	4
0	1	0	0	0	1	0	0	1
0	0	8	1	-4	-6	0	0	4

D'après ce tableau final qui sera noté (T), $x_1 = 1,\ x_2 = 1,\ x_3 = \frac{4}{8} = \frac{1}{2},\ x_4 = x_5 = x_6 = 0,\ x_7 = \frac{4}{8} = \frac{1}{2}$. La solution optimale non entière du problème est donc $x^* = (1, 1, \frac{1}{2})$ et la valeur maximale que peut avoir la fonction objectif z correspondant à cette solution optimale est donc $z = \frac{140}{8} = \frac{35}{2} = 17,5$.

<u>Recherche de la solution optimale de (P) par la méthode des coupes de Gomory :</u>

De (T), on a :

$$\begin{cases} x_1 + x_5 = 1 \\ -x_4 + 4x_5 + 6x_6 + 8x_7 = 4 \\ x_2 + x_6 = 1 \\ 8x_3 + x_4 - 4x_5 - 6x_6 = 4 \end{cases} \Leftrightarrow \begin{cases} x_1 + x_5 = 1 \\ x_7 - \frac{1}{8}x_4 + \frac{4}{8}x_5 + \frac{6}{8}x_6 = \frac{4}{8} \\ x_2 + x_6 = 1 \\ x_3 + \frac{1}{8}x_4 - \frac{4}{8}x_5 - \frac{6}{8}x_6 = \frac{4}{8} \end{cases} \Leftrightarrow \begin{cases} x_1 + x_5 = 1 \\ x_7 - \frac{1}{8}x_4 + \frac{1}{2}x_5 + \frac{3}{4}x_6 = \frac{1}{2} \\ x_2 + x_6 = 1 \\ x_3 + \frac{1}{8}x_4 - \frac{1}{2}x_5 - \frac{3}{4}x_6 = \frac{1}{2} \end{cases}$$

Les coupes de Gomory sont :

$$\begin{cases} \left\{-\frac{1}{8}\right\} x_4 + \left\{\frac{1}{2}\right\} x_5 + \left\{\frac{3}{4}\right\} x_6 & \geqslant \left\{\frac{1}{2}\right\} \\ \left\{\frac{1}{8}\right\} x_4 + \left\{-\frac{1}{2}\right\} x_5 + \left\{-\frac{3}{4}\right\} x_6 & \geqslant \left\{\frac{1}{2}\right\} \end{cases} \Leftrightarrow \begin{cases} \frac{7}{8}x_4 + \frac{1}{2}x_5 + \frac{3}{4}x_6 & \geqslant \frac{1}{2} \\ \frac{1}{8}x_4 + \frac{1}{2}x_5 + \frac{1}{4}x_6 & \geqslant \frac{1}{2} \end{cases} \Leftrightarrow \begin{cases} 7x_4 + 4x_5 + 6x_6 & \geqslant 4 \\ x_4 + 4x_5 + 2x_6 & \geqslant 4 \end{cases}$$

Par l'introduction de variables d'écart, ces coupes sont transformées en égalité :

$$\begin{cases} 7x_4 + 4x_5 + 6x_6 - x_8 & = 4 \\ x_4 + 4x_5 + 2x_6 - x_9 & = 4 \end{cases}$$

Il faut ensuite ajouter ces égalités à (T). Les variables d'écart introduites dans ces égalités sont précédées du signe moins, le tableau nouvellement obtenu ne sera donc pas directement réalisable, il faut alors passer par la phase de recherche d'une base réalisable. Pour cela, il faut introduire des variables artificielles au niveau de ces égalités. Soient y_0 et y_1 ces variables artificielles, elles sont positives ou nulles.

$$\begin{cases} 7x_4 + 4x_5 + 6x_6 - x_8 + y_0 & = 4 \\ x_4 + 4x_5 + 2x_6 - x_9 + y_1 & = 4 \end{cases} \Leftrightarrow \begin{cases} y_0 = -7x_4 - 4x_5 - 6x_6 + x_8 + 4 \\ y_1 = -x_4 - 4x_5 - 2x_6 + x_9 + 4 \end{cases}$$

Puis, résoudre le programme linéaire suivant :

$$\begin{cases} max\ w = -y_0 - y_1 = 8x_4 + 8x_5 + 8x_6 - x_8 - x_9 - 8 \\ x_1 + x_5 = 1 \\ -x_4 + 4x_5 + 6x_6 + 8x_7 = 4 \\ x_2 + x_6 = 1 \\ 8x_3 + x_4 - 4x_5 - 6x_6 = 4 \\ 7x_4 + 4x_5 + 6x_6 - x_8 + y_0 = 4 \\ x_4 + 4x_5 + 2x_6 - x_9 + y_1 = 4 \end{cases}$$

x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	y_0	y_1	w	
0	0	0	8	8	8	0	-1	-1	0	0	-1	8
1	0	0	0	1	0	0	0	0	0	0	0	1
0	0	0	-1	4	6	8	0	0	0	0	0	4
0	1	0	0	0	1	0	0	0	0	0	0	1
0	0	8	1	-4	-6	0	0	0	0	0	0	4
0	0	0	7	$\boxed{4}$	6	0	-1	0	1	0	0	4
0	0	0	1	4	2	0	0	-1	0	1	0	4

x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	y_0	y_1	w	
0	0	0	-6	0	-4	0	1	-1	-2	0	-1	0
4	0	0	-7	0	-6	0	1	0	-1	0	0	0
0	0	0	-8	0	0	8	1	0	-1	0	0	0
0	1	0	0	0	1	0	0	0	0	0	0	1
0	0	8	8	0	0	0	-1	0	1	0	0	8
0	0	0	7	4	6	0	-1	0	1	0	0	4
0	0	0	-6	0	-4	0	$\boxed{1}$	-1	-1	1	0	0

x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	y_0	y_1	w	
0	0	0	0	0	0	0	0	0	-1	-1	-1	0
4	0	0	-1	0	-2	0	0	1	0	-1	0	0
0	0	0	-2	0	4	8	0	1	0	-1	0	0
0	1	0	0	0	1	0	0	0	0	0	0	1
0	0	8	2	0	-4	0	0	-1	0	1	0	8
0	0	0	1	4	2	0	0	-1	0	1	0	4
0	0	0	-6	0	-4	0	1	-1	-1	1	0	0

Les variables artificielles sortent de base et sont donc de valeur nulle. Le tableau, une fois les colonnes des variables artificielles enlevées, est réalisable et l'algorithme du simplexe pourra se poursuivre.
Il ne faut pas oublier de reprendre l'expression contenant z dans (T), et de remplacer les variables de base par leurs nouvelles expressions données par le dernier tableau ci-dessus, de sorte à n'avoir que des variables hors base dans l'expression contenant z.
L'expression contenant z dans (T) : $-7x_4 - 20x_5 - 22x_6 - 8z = -140$
Comme on a, d'après ce dernier tableau : $x_4 + 4x_5 + 2x_6 - x_9 + y_1 = x_4 + 4x_5 + 2x_6 - x_9 = 4$ alors $x_5 = 1 - \frac{1}{4}x_4 - \frac{2}{4}x_6 + \frac{1}{4}x_9$ et donc en portant cette valeur de x_5 dans l'expression contenant z, on obtient la nouvelle expression $-2x_4 - 12x_6 - 5x_9 - 8z = -120$.

x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	z	
0	0	0	-2	0	-12	0	0	-5	-8	-120
4	0	0	-1	0	-2	0	0	1	0	0
0	0	0	-2	0	4	8	0	1	0	0
0	1	0	0	0	1	0	0	0	0	1
0	0	8	2	0	-4	0	0	-1	0	8
0	0	0	1	4	2	0	0	-1	0	4
0	0	0	-6	0	-4	0	1	-1	0	0

Tous les coûts réduits étant négatifs, l'algorithme du simplexe s'arrête et puisque la solution optimale est ici entière, il n'y a plus besoin de poursuivre le mécanisme des coupes de Gomory.
Comme $x_1 = 0,\ \ x_2 = 1,\ \ x_3 = \frac{8}{8} = 1,\ \ x_5 = \frac{4}{4} = 1,\ \ x_4 = x_6 = x_7 = x_8 = x_9 = 0$
et $-2x_4 - 12x_6 - 5x_9 - 8z = -120$ alors $z = \frac{-120}{-8} = 15$, c'est la valeur totale maximale possible des objets à placer dans le sac. L'optimum discret de notre problème est $(0, 1, 1)$, il faut donc choisir de placer dans le sac les objets O_2 et O_3.

2.2 Méthode de séparation et évaluation

La méthode de séparation et évaluation repose sur le principe de diviser pour régner. Cette méthode tente d'explorer intelligemment, par une arborescence, l'ensemble des solutions admissibles en éliminant de l'espace de recherche les sous-ensembles de solutions qui ne peuvent pas fournir une solution optimale.

2.2.1 Description de la méthode de séparation et évaluation

Reprenons le programme linéaire en nombres entiers (P). Si la solution optimale après la résolution du programme linéaire relaxé par la méthode du simplexe est entière, le problème (P) est résolu. Dans le cas contraire, la résolution de (P) est encore inachevée.

Soit à noter x^* la solution optimale non entière, $x^* = (x_1^*, ..., x_n^*)$ et supposons que parmi les x_i^*, x_1^* et x_2^* soient non entiers. Il faut par la suite, résoudre séparément, toujours à l'aide de la méthode du simplexe, deux sous-problèmes. Chaque sous-problème s'obtient en ajoutant à (P) une contrainte bien déterminée. La variable dans la contrainte à ajouter à (P) est prise au choix parmi les variables qui ont des valeurs non entières (dans le cas présent, c'est soit x_1 soit x_2. Ainsi :
— Le premier sous-problème est le problème (P) auquel la contrainte $x_1 \leqslant [x_1^*]$ (respectivement $x_2 \leqslant [x_2^*]$) a été ajoutée.

— Le second sous-problème est le problème (P) auquel la contrainte $x_1 \geqslant [x_1^*]+1$ (respectivement $x_2 \geqslant [x_2^*]+1$) a été ajoutée.

Chaque sous-problème sera de nouveau scindé en deux autres sous-problèmes tant que sa solution optimale n'est pas entière. Au final, plusieurs solutions entières correspondant aux solutions optimales entières des divers sous-problèmes, seront obtenues. Puis, en comparant ces solutions par rapport à la valeur respective qu'elles donnent à la fonction à optimiser, il ne reste plus qu'à prendre la solution entière qui optimise le plus cette fonction objectif.

2.2.2 Exemple d'application

Reconsidérons le programme linéaire en nombres entiers (P) de la section précédente et résolvons-le avec la méthode de séparation et évaluation.
La solution optimale du programme linéaire relaxé trouvée un peu plus haut est $x^* = (1, 1, \frac{1}{2})$, x_3^* n'est pas un entier, ce qui amène à résoudre initialement deux sous-problèmes.

Premier sous-problème (P_1) : c'est le problème (P) avec en plus, la contrainte $x_3 \leqslant [x_3^*]$, $(P_1) = (P) + (x_3 \leqslant 0)$

$$P_1 \begin{cases} max\ z = 6x_1 + 8x_2 + 7x_3 \\ 4x_1 + 6x_2 + 8x_3 \leqslant 14 \\ x_1 \leqslant 1 \\ x_2 \leqslant 1 \\ x_3 \leqslant 1 \\ x_3 \leqslant 0 \\ \forall i,\ x_i \in \mathbb{N} \end{cases}$$

Comme x_3 doit être à la fois supérieure ou égale à 0 et inférieure ou égale à 0 alors nécessairement $x_3 = 0$. (P_1) devient :

$$\begin{cases} max\ z = 6x_1 + 8x_2 \\ 4x_1 + 6x_2 \leqslant 14 \\ x_1 \leqslant 1 \\ x_2 \leqslant 1 \\ \forall i,\ x_i \in \mathbb{N} \end{cases} \Leftrightarrow \begin{cases} max\ z = 6x_1 + 8x_2 \\ 2x_1 + 3x_2 \leqslant 7 \\ x_1 \leqslant 1 \\ x_2 \leqslant 1 \\ \forall i,\ x_i \in \mathbb{N} \end{cases}$$

Résolution de (P_1) :

La résolution graphique de (P_1) est évidemment possible et plus simple, mais pour rester dans la généralité, (P_1) sera résolu à l'aide de la méthode du simplexe.

x_1	x_2	x_4	x_5	x_6	z	
6	8	0	0	0	-1	0
2	3	1	0	0	0	7
1	0	0	1	0	0	1
0	$\boxed{1}$	0	0	1	0	1

x_1	x_2	x_4	x_5	x_6	z	
6	0	0	0	-8	-1	-8
2	0	1	0	-3	0	4
$\boxed{1}$	0	0	1	0	0	1
0	1	0	0	1	0	1

x_1	x_2	x_4	x_5	x_6	z	
0	0	0	-6	-8	-1	-14
0	0	1	-2	-3	0	2
1	0	0	1	0	0	1
0	1	0	0	1	0	1

On a : $x_3 = 0,\ \ x_1 = 1,\ \ x_2 = 1,\ \ x_4 = 2,\ \ x_5 = x_6 = 0.$
La solution optimale de (P_1) est (1,1,0), elle est déjà entière, et la valeur de z qui y est associée est $z = 14$.

Deuxième sous-problème (P_2) : c'est le problème (P) avec en plus, la contrainte $x_3 \geqslant [x_3^*] + 1$, $(P_2) = (P) + \ (x_3 \geqslant\ 1)$

$$P_2 \begin{cases} max\ z = 6x_1 + 8x_2 + 7x_3 \\ 4x_1 + 6x_2 + 8x_3 \leqslant 14 \\ x_1 \leqslant 1 \\ x_2 \leqslant 1 \\ x_3 \leqslant 1 \\ x_3 \geqslant 1 \\ \forall i,\ x_i \in \mathbb{N} \end{cases}$$

Comme x_3 doit être à la fois supérieure ou égale à 1 et inférieure ou égale à 1 alors nécessairement $x_3 = 1$. (P_2) devient :

$$\begin{cases} max\ z = 7 + 6x_1 + 8x_2 \\ 4x_1 + 6x_2 + 8 \leqslant 14 \\ x_1 \leqslant 1 \\ x_2 \leqslant 1 \\ \forall i,\ x_i \in \mathbb{N} \end{cases} \Leftrightarrow \begin{cases} max\ z = 7 + 6x_1 + 8x_2 \\ 2x_1 + 3x_2 \leqslant 3 \\ x_1 \leqslant 1 \\ x_2 \leqslant 1 \\ \forall i,\ x_i \in \mathbb{N} \end{cases}$$

Résolution de (P_2) :

x_1	x_2	x_4	x_5	x_6	z	
6	8	0	0	0	-1	-7
2	3	1	0	0	0	3
1	0	0	1	0	0	1
0	$\boxed{1}$	0	0	1	0	1

x_1	x_2	x_4	x_5	x_6	z	
6	0	0	0	-8	-1	-15
$\boxed{2}$	0	1	0	-3	0	0
1	0	0	1	0	0	1
0	1	0	0	1	0	1

x_1	x_2	x_4	x_5	x_6	z	
0	0	-3	0	1	-1	-15
2	0	1	0	-3	0	0
0	0	-1	2	$\boxed{3}$	0	2
0	1	0	0	1	0	1

x_1	x_2	x_4	x_5	x_6	z	
0	0	-8	-2	0	-3	-47
6	0	0	6	0	0	6
0	0	-1	2	3	0	2
0	3	1	-2	0	0	1

On a : $x_3 = 1$, $x_1 = \frac{6}{6} = 1$, $x_2 = \frac{1}{3}$, $x_4 = x_5 = 0$, $x_6 = \frac{2}{3}$.
La solution optimale de (P_2) est $\tilde{x} = (1, \frac{1}{3}, 1)$. Cette solution n'est pas encore entière, $\tilde{x}_2 = \frac{1}{3}$, alors (P_2) sera à son tour scindé en deux sous-problèmes $(P_{2,1})$ et $(P_{2,2})$ où :

- ✦ $(P_{2,1}) = (P_2) + (x_2 \leqslant \left[\frac{1}{3}\right])$ c'est-à-dire $(P_{2,1}) = (P_2) + (x_2 \leqslant 0)$,

- ✦ et $(P_{2,2}) = (P_2) + (x_2 \geqslant \left[\frac{1}{3}\right] + 1)$ c'est-à-dire $(P_{2,2}) = (P_2) + (x_2 \geqslant 1)$

Résolution de $(P_{2,1})$:

$$\begin{cases} max\ z = 6x_1 + 8x_2 + 7x_3 \\ 4x_1 + 6x_2 + 8x_3 \leqslant 14 \\ x_1 \leqslant 1 \\ x_2 \leqslant 1 \\ x_2 \leqslant 0 \\ x_3 = 1 \\ \forall i,\ x_i \in \mathbb{N} \end{cases} \Leftrightarrow \begin{cases} max\ z = 7 + 6x_1 + 8x_2 \\ 2x_1 + 3x_2 \leqslant 3 \\ x_1 \leqslant 1 \\ x_2 = 0 \\ x_1 \geqslant 0 \\ \forall i,\ x_i \in \mathbb{N} \end{cases} \Leftrightarrow \begin{cases} max\ z = 7 + 6x_1 \\ x_1 \leqslant \frac{3}{2} \\ x_1 \leqslant 1 \\ x_1 \geqslant 0 \\ \forall i,\ x_i \in \mathbb{N} \end{cases}$$

Par raisonnement logique, $x_1 = 1$, la solution optimale de $(P_{2,1})$ est donc : $(1, 0, 1)$ et est déjà entière. La valeur de z qui y est associée est : $z = 7 + 6 = 13$.

Résolution de $(P_{2,2})$:

$$\begin{cases} max\ z = 6x_1 + 8x_2 + 7x_3 \\ 4x_1 + 6x_2 + 8x_3 \leqslant 14 \\ x_1 \leqslant 1 \\ x_2 \leqslant 1 \\ x_2 \geqslant 1 \\ x_3 = 1 \\ \forall i,\ x_i \in \mathbb{N} \end{cases} \Leftrightarrow \begin{cases} max\ z = 7 + 6x_1 + 8x_2 \\ 2x_1 + 3x_2 \leqslant 3 \\ x_1 \leqslant 1 \\ x_2 = 1 \\ \forall i,\ x_i \in \mathbb{N} \end{cases} \Leftrightarrow \begin{cases} max\ z = 7 + 6x_1 + 8 \\ 2x_1 \leqslant 0 \\ x_1 \leqslant 1 \\ \forall i,\ x_i \in \mathbb{N} \end{cases}$$

Par raisonnement logique, $x_1 = 0$, la solution optimale de $(P_{2,2})$ est donc : $(0, 1, 1)$ et est déjà entière. La valeur de z qui y est associée est : $z = 7 + 8 = 15$.

Puisqu'il faut maximiser z, il faut prendre comme solution optimale entière de (P_2) la solution optimale de $(P_{2,2})$, c'est-à-dire $(0, 1, 1)$ où $z = 15$. Et comme c'est la solution optimale entière de (P_2) qui maximise le plus la fonction objectif z par rapport à celle de (P_1) , c'est donc la solution optimale entière, par la méthode de séparation et évaluation, du programme linéaire en nombres entiers (P).

Troisième partie

Amélioration de la méthode du simplexe pour la résolution de problèmes linéaires en nombres entiers

3.1 Principe et description de la méthode

Soit à considérer le programme linéaire en nombres entiers suivant :

$$\begin{cases} max\ z = c_1x_1 + \ldots + c_nx_n \\ a_{11}x_1 + \ldots + a_{1n}x_n \leqslant b_1 \\ \vdots \\ a_{m1}x_1 + \ldots + a_{mn}x_n \leqslant b_m \\ \forall i,\ x_i \in \mathbb{N} \end{cases}$$

Il convient de rappeler brièvement que la méthode usuelle de résolution d'un programme linéaire en nombres entiers consiste, d'après ce qui a été vu dans les parties précédentes, à utiliser l'algorithme du simplexe pour aboutir dans un premier temps à l'optimum continu, c'est-à-dire la solution optimale non entière, ou dans le meilleur des cas aboutir directement à l'optimum discret, c'est-à-dire une solution optimale déjà entière. Il est évident que, si la solution obtenue à la suite de la méthode du simplexe était toujours entière, le problème ne se poserait plus. Mais dans le cas de l'optimum continu, pour parvenir à une solution optimale entière, la résolution fait le plus souvent appel à l'utilisation des coupes telles que les coupes de Gomory, ou, passe par la méthode de séparation et évaluation.

Mais est-il possible de trouver directement la solution optimale entière sans nécessairement devoir trouver l'optimum continu ? La méthode qui va suivre s'inspire de l'idée suivante : dans la méthode sous forme tableau du simplexe, le passage d'un tableau à un autre représente le saut d'un sommet du domaine de réalisabilité du programme linéaire à un autre sommet de ce même domaine sans regarder si ces sommets correspondent à des solutions entières ou non et le tableau final traduit l'arrivée au sommet correspondant à l'optimum du programme linéaire. L'idéal est donc que, les points parcourus par l'algorithme du simplexe correspondent à des solutions réalisables entières, et qu'une fois l'algorithme arrivé au tableau final donnant la solution optimale, cette solution soit tout de suite entière.

Le principe de la méthode est donc de partir d'une solution réalisable entière et à l'aide de la méthode du simplexe, d'aller vers une autre solution réalisable entière jusqu'à atteindre l'optimum, qui sera donc un optimum discret. D'après ce principe, cette méthode n'est valable que sous certaines conditions : il faut que le point de départ de l'algorithme du simplexe corresponde à une solution réalisable entière, en d'autres termes, il faut que tous les éléments du premier tableau simplexe soient des entiers et surtout, que la matrice correspondant aux variables de base soit une matrice unité à quelques permutations près. Ces conditions sont directement satisfaites dans le cas d'un programme linéaire en nombres entiers comme susmentionné avec les $b_i \geqslant 0$.

Plus concrètement, voici comment procéder :

1. Il faut établir le premier tableau simplexe où tous les éléments sont des entiers et où la matrice des variables de base est une matrice unité à quelques permutations près.

2. Par la suite, il faut déterminer le pivot dans le tableau considéré, soit $\bar{a}_{kj}$ le pivot en question, j étant le numéro de la colonne pivot et k celui de la ligne pivot.

   - ✦ Si $\bar{a}_{kj} = 1$ alors on procède aux changements comme dans la méthode du simplexe habituelle.

✦ Sinon, on ajoute au tableau une équation obtenue comme suit :

En divisant par $\bar{a}_{kj}$ l'équation correspondant à la ligne pivot, on a :

$$\frac{\bar{a}_{k1}}{\bar{a}_{kj}}x_1 + ... + x_j + ... + \frac{\bar{a}_{kn}}{\bar{a}_{kj}}x_n + ... = \frac{\bar{b}_k}{\bar{a}_{kj}}$$

Puis, en remplaçant les coefficients de cette nouvelle équation par leur partie entière, c'est-à-dire, les $\frac{\bar{a}_{ki}}{\bar{a}_{kj}}$ par $\left[\frac{\bar{a}_{ki}}{\bar{a}_{kj}}\right]$ et $\frac{\bar{b}_k}{\bar{a}_{kj}}$ par $\left[\frac{\bar{b}_k}{\bar{a}_{kj}}\right]$, on obtient :

$$\left[\frac{\bar{a}_{k1}}{\bar{a}_{kj}}\right]x_1 + ... + x_j + ... + \left[\frac{\bar{a}_{kn}}{\bar{a}_{kj}}\right]x_n + ... \leqslant \left[\frac{\bar{b}_k}{\bar{a}_{kj}}\right] \tag{1}$$

où [.] désigne la partie entière. En transformant cette inégalité (1) en égalité par l'ajout d'une variable d'écart, l'équation à rajouter au tableau est obtenue.
Ensuite, la nouvelle variable d'écart sort de base et x_j entre en base.

3. Effectuer l'étape 2 jusqu'à ce que la condition d'arrêt comme pour l'algorithme habituel du simplexe, soit vérifiée.

Ce procédé peut être représenté par une boucle FAIRE...TANT QUE dont voici le pseudo-code :

FAIRE :
Choisir colonne pivot
Déterminer le pivot
Si pivot == 1
Continuer avec l'algorithme du simplexe
Sinon
Ajouter au tableau en cours l'ingalité de type (1) *correspondant à la ligne pivot*
Continuer avec l'algorithme du simplexe
TANT QUE : *la condition d'arrêt n'est pas vérifée*

3.2 Validation de la méthode

3.2.1 Explicitation de la procédure de la méthode

L'étape 1 permet de s'assurer que le point de départ de l'algorithme soit une solution réalisable entière. En effet, tous les éléments du tableau simplexe initial doivent être des entiers, les seconds membres dans le tableau le sont donc aussi et puisque la matrice correspondant aux variables de base est une matrice unité à quelques permutations près, alors les variables de base prendront des valeurs entières ; par conséquent, le tableau initial est associé à une solution réalisable entière.

L'étape 2 fait en sorte que chaque nouveau tableau, obtenu après les modifications par la méthode du simplexe, soit associé à une solution réalisable entière. Effectivement,

— si le pivot $\bar{a}_{kj}$ est égal à 1 alors la variable hors base qui est associée à ce pivot et qui va entrer en base aura une valeur entière, en portant cette valeur entière dans les autres équations du tableau, les autres variables de base auront toujours des valeurs entières ; de ce fait, le nouveau tableau obtenu après ces transformations relatives à la méthode du simplexe, correspondra à une autre solution réalisable entière.

— Dans le cas contraire, c'est-à-dire, si $\bar{a}_{kj} \neq 1$, alors en rajoutant au tableau l'équation $\left[\frac{\bar{a}_{k1}}{\bar{a}_{kj}}\right] x_1 + \ldots + x_j + \ldots + \left[\frac{\bar{a}_{kn}}{\bar{a}_{kj}}\right] x_n + \ldots + s_1 = \left[\frac{\bar{b}_k}{\bar{a}_{kj}}\right]$ où s_1 est la nouvelle variable d'écart introduite, cette équation constituera la nouvelle ligne pivot puisque $\left[\frac{\bar{b}_k}{\bar{a}_{kj}}\right]$ deviendra le rapport minimum car $\left[\frac{\bar{b}_k}{\bar{a}_{kj}}\right] \leqslant \frac{\bar{b}_k}{\bar{a}_{kj}}$ et il est possible de voir que le nouveau pivot, étant le coefficient de x_j dans l'équation rajoutée, est égal à 1, ce qui nous ramène au cas mentionné dans le premier point ci-dessus.

L'algorithme du simplexe présente la caractéristique que lorsqu'on passe d'un tableau à un autre, la fonction objectif est optimisée (augmente dans le cas d'un problème linéaire de maximisation et diminue dans le cas d'un problème linéaire de minimisation) ou au moins garde sa valeur jusqu'à ce que sa valeur optimale soit atteinte. Mais puisque la nouvelle méthode présentée ci-dessus n'est autre que l'algorithme du simplexe dirigé de sorte qu'au niveau de chaque tableau les contraintes d'intégrité soient respectées, alors grâce à la caractéristique de l'algorithme du simplexe, cette nouvelle méthode devrait donner au final la solution optimale du problème.

3.2.2 Vérification de la validité de la méthode

Il est nécessaire de vérifier si la solution optimale du programme linéaire en nombres entiers sans l'ajout d'inégalités de type (1) est toujours conservée malgré l'ajout de ces inégalités. Il est bien de rappeler que si aucune inégalité n'a été rajoutée, cette méthode n'est autre que la méthode du simplexe et dans la méthode du simplexe, il est connu que les contraintes d'un tableau sont équivalentes à celles de chaque autre tableau. Sans perdre la généralité, soit à considérer les contraintes (S) d'un programme linéaire en nombres entiers noté (P) :

$$(S)\begin{cases} \bar{a}_{11}x_1 + \ldots + \bar{a}_{1j}x_j + \ldots + \bar{a}_{1n}x_n = \bar{b}_1 \\ \vdots \\ \bar{a}_{m1}x_1 + \ldots + \bar{a}_{mj}x_j + \ldots + \bar{a}_{mn}x_n = \bar{b}_m \\ \\ \forall i,\ x_i \in \mathbb{N} \end{cases}$$

Soit à noter $(\bar{S})$ les contraintes (S) rajoutées d'une inégalité de type (1).

$$(\bar{S})\begin{cases} \bar{a}_{11}x_1 + \ldots + \bar{a}_{1j}x_j + \ldots + \bar{a}_{1n}x_n = \bar{b}_1 \\ \vdots \\ \bar{a}_{k1}x_1 + \ldots + \bar{a}_{kj}x_j + \ldots + \bar{a}_{kn}x_n = \bar{b}_k \\ \vdots \\ \bar{a}_{m1}x_1 + \ldots + \bar{a}_{mj}x_j + \ldots + \bar{a}_{mn}x_n = \bar{b}_m \\ \left[\frac{\bar{a}_{k1}}{\bar{a}_{kj}}\right] x_1 + \ldots + x_j + \ldots + \left[\frac{\bar{a}_{kn}}{\bar{a}_{kj}}\right] x_n \leqslant \left[\frac{\bar{b}_k}{\bar{a}_{kj}}\right] \\ \forall i,\ x_i \in \mathbb{N} \end{cases}$$

Soit à noter $(\bar{P})$ le programme linéaire en nombres entiers avec les contraintes $(\bar{S})$.

Démonstration :

Pour bien nous convaincre que la solution optimale de (P) est la même avec ou sans l'ajout d'inégalités de type (1), voici une démonstration qui consiste en premier lieu, à montrer l'équivalence suivante : toute solution réalisable de (P) est aussi solution réalisable de $(\bar{P})$ et réciproquement ; et en second lieu, à montrer que toute solution optimale pour $(\bar{P})$ est aussi optimale pour (P).

Soit donc $\tilde{x} = (\tilde{x}_1, ..., \tilde{x}_n)$ une solution réalisable de (P), soit à montrer que $\tilde{x}$ est aussi solution réalisable de $(\bar{P})$, pour cela il suffit de montrer que $\tilde{x}$ vérifie l'inégalité de type (1). Puisque $\tilde{x}$ est une solution réalisable de (P) alors $\tilde{x}$ vérifie les contraintes de (S), ainsi, pour tout i, $\tilde{x}_i \in \mathbb{N}$ et on a pour tout k variant de 1 à m :

$$\sum_{i=1}^{n} \bar{a}_{ki}\tilde{x}_i = \bar{b}_k$$
$$\sum_{i=1}^{n} \frac{\bar{a}_{ki}}{\bar{a}_{kj}}\tilde{x}_i = \frac{\bar{b}_k}{\bar{a}_{kj}}$$

D'après la définition de la partie entière, on a :

$$\sum_{i=1}^{n} \left[\frac{\bar{a}_{ki}}{\bar{a}_{kj}}\right]\tilde{x}_i \leqslant \sum_{i=1}^{n} \frac{\bar{a}_{ki}}{\bar{a}_{kj}}\tilde{x}_i$$

alors

$$\sum_{i=1}^{n} \left[\frac{\bar{a}_{ki}}{\bar{a}_{kj}}\right]\tilde{x}_i \leqslant \frac{\bar{b}_k}{\bar{a}_{kj}}$$

et par suite,

$$\sum_{i=1}^{n} \left[\frac{\bar{a}_{ki}}{\bar{a}_{kj}}\right]\tilde{x}_i \leqslant \left[\frac{\bar{b}_k}{\bar{a}_{kj}}\right]$$

En d'autres termes, la solution $\tilde{x}$ vérifie toutes les contraintes de $(\bar{S})$, $\tilde{x}$ est donc aussi une solution réalisable de $(\bar{P})$. Ainsi, toute solution réalisable de (P) est aussi solution réalisable de $(\bar{P})$.
L'implication réciproque de l'équivalence, c'est-à-dire, toute solution réalisable de $(\bar{P})$ est aussi solution réalisable de (P), est évidente puisque le domaine de réalisabilité de $(\bar{P})$ est inclus dans celui de (P).
L'équivalence a donc été bien démontrée.

Soit maintenant à montrer que toute solution optimale pour $(\bar{P})$ est aussi optimale pour (P). Soit $x^* = (x_1^*, x_2^*, ..., x_n^*)$ une solution optimale de $(\bar{P})$ et z^* la valeur optimale. Soit $\tilde{x} = (\tilde{x}_1, \tilde{x}_2, ..., \tilde{x}_n)$ une solution réalisable de (P) et $\tilde{z}$ la valeur de la fonction objectif correspondante. D'après ce qui précède, $\tilde{x}$ est une solution réalisable de $(\bar{P})$. Donc, $\tilde{z} \leqslant z^*$. Ce qui prouve que x^* est une solution optimale de (P).

D'après la démonstration ci-dessus, il est possible de conclure que la résolution du programme linéaire en nombres entiers (P) est équivalente à la résolution du programme linéaire en nombres entiers $(\bar{P})$.

3.2.3 Avantages de la méthode

Il est à rappeler que la condition d'arrêt du nouvel algorithme est la même que celle de l'algorithme du simplexe. La manière dont la colonne pivot est choisie dans cet algorithme, permet d'optimiser la valeur de la fonction objectif au passage d'un tableau au suivant. Le nouvel algorithme passe alors d'une solution réalisable entière à une autre tout en optimisant la valeur de la fonction objectif ; l'algorithme se terminera donc forcément puisqu'il arrivera un moment où cette valeur ne pourra plus être optimisée, en admettant que la solution du problème linéaire en nombres entiers existe et que de ce fait l'ensemble des solutions réalisables entières est fini. Ainsi, le nouvel algorithme converge et il permet donc de trouver l'optimum du problème en un nombre fini d'opérations.

Dans la pratique, la nouvelle méthode est plus commode que les deux autres méthodes présentées dans la partie précédente. En effet, ces deux autres méthodes ne s'arrêtent que lorsque la solution optimale obtenue est entière, ce qui peut engendrer de longs calculs fastidieux, d'autant plus que ces méthodes doivent passer au préalable par la résolution du problème linéaire relaxé, cette résolution peut aussi prendre du temps. Par contre, la nouvelle méthode résout directement le problème, elle se termine lorsque la valeur de la fonction objectif est optimale et la solution y étant associée est déjà entière.

3.3 Exemples d'application

Exemple 1

Reprenons le problème du sac à dos énoncé dans la partie précédente et résolvons-le par la méthode décrite ci-dessus. Soit donc à résoudre, par la nouvelle méthode, le programme linéaire en nombres entiers (P) :

$$\begin{cases} max\ z = 6x_1 + 8x_2 + 7x_3 \\ \quad 4x_1 + 6x_2 + 8x_3 \leqslant 14 \\ \qquad\qquad x_1 \leqslant 1 \\ \qquad\qquad x_2 \leqslant 1 \\ \qquad\qquad x_3 \leqslant 1 \\ \qquad \forall i,\ x_i \in \mathbb{N} \end{cases}$$

Dans toute la suite, puisque la colonne de z reste inchangée, il n'est pas nécessaire de la mettre et au lieu d'être mise à droite comme pour les tableaux dans les parties précédentes, la colonne de $\bar{b}$ sera placée à l'extrémité gauche, ce qui est plus pratique lors de l'ajout des variables d'écart à l'extrémité droite. Cela ne modifie en rien les procédés de calcul.

	x_1	x_2	x_3	x_4	x_5	x_6	x_7
0	6	8	7	0	0	0	0
14	4	6	8	1	0	0	0
1	1	0	0	0	1	0	0
1	0	$\boxed{1}$	0	0	0	1	0
1	0	0	1	0	0	0	1

	x_1	x_2	x_3	x_4	x_5	x_6	x_7
-8	6	0	7	0	0	-8	0
8	4	0	8	1	0	-6	0
1	1	0	0	0	1	0	0
1	0	1	0	0	0	1	0
1	0	0	$\boxed{1}$	0	0	0	1

Dans ces deux tableaux précédents, le pivot est égal à 1, il n'y a donc pas de changements particuliers à faire, l'algorithme du simplexe se poursuit directement.

	x_1	x_2	x_3	x_4	x_5	x_6	x_7
-15	6	0	0	0	0	-8	-7
0	$\boxed{4}$	0	0	1	0	-6	-8
1	1	0	0	0	1	0	0
1	0	1	0	0	0	1	0
1	0	0	1	0	0	0	1

Dans ce tableau-ci, le pivot est différent de 1, il est égal à 4. Ainsi, avant de poursuivre l'algorithme du simplexe, il faut établir l'inégalité de type (1) correspondant à la contrainte de la ligne pivot, puis transformer cette inégalité en égalité par l'ajout de variable d'écart et enfin ajouter cette égalité obtenue à ce tableau.
Contrainte de la ligne pivot : $4x_1 + x_4 - 6x_6 - 8x_7 = 0 \iff x_1 + \frac{1}{4}x_4 - \frac{3}{2}x_6 - 2x_7 = 0$
Inégalité de type (1) déduite de cette contrainte : $x_1 + \left[\frac{1}{4}\right]x_4 + \left[-\frac{3}{2}\right]x_6 - 2x_7 \leqslant 0 \iff x_1 - 2x_6 - 2x_7 \leqslant 0$
Égalité à rajouter au tableau : $x_1 - 2x_6 - 2x_7 + x_8 = 0$

	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8
-15	6	0	0	0	0	-8	-7	0
0	4	0	0	1	0	-6	-8	0
1	1	0	0	0	1	0	0	0
1	0	1	0	0	0	1	0	0
1	0	0	1	0	0	0	1	0
0	$\boxed{1}$	0	0	0	0	-2	-2	1

	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8
-15	0	0	0	0	0	4	5	-6
0	0	0	0	1	0	2	0	-4
1	0	0	0	0	1	2	$\boxed{2}$	-1
1	0	1	0	0	0	1	0	0
1	0	0	1	0	0	0	1	0
0	1	0	0	0	0	-2	-2	1

Ici, le pivot est de nouveau différent de 1, le pivot est égal à 2. Ainsi, comme pour le troisième tableau, il faut rajouter à ce tableau-ci l'égalité obtenue à partir de l'inégalité de type (1) correspondant à la contrainte de la ligne pivot, avant de continuer par la méthode du simplexe.
Contrainte de la ligne pivot : $x_5 + 2x_6 + 2x_7 - x_8 = 1 \iff \frac{1}{2}x_5 + x_6 + x_7 - \frac{1}{2}x_8 = \frac{1}{2}$
Inégalité de type (1) déduite de cette contrainte : $\left[\frac{1}{2}\right]x_5 + x_6 + x_7 + \left[-\frac{1}{2}\right]x_8 \leqslant \left[\frac{1}{2}\right] \iff x_6 + x_7 - x_8 \leqslant 0$
Égalité à rajouter au tableau : $x_6 + x_7 - x_8 + x_9 = 0$

	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9
-15	0	0	0	0	0	4	5	-6	0
0	0	0	0	1	0	2	0	-4	0
1	0	0	0	0	1	2	2	-1	0
1	0	1	0	0	0	1	0	0	0
1	0	0	1	0	0	0	1	0	0
0	1	0	0	0	0	-2	-2	1	0
0	0	0	0	0	0	1	$\boxed{1}$	-1	1

	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9
-15	0	0	0	0	0	-1	0	-1	-5
0	0	0	0	1	0	2	0	-4	0
1	0	0	0	0	1	0	0	1	-2
1	0	1	0	0	0	1	0	0	0
1	0	0	1	0	0	-1	0	1	-1
0	1	0	0	0	0	0	0	-1	2
0	0	0	0	0	0	1	1	-1	1

L'algorithme du simplexe s'arrête vu que tous les coûts réduits sont négatifs.
On a : $x_1 = 0, \;\; x_2 = 1, \;\; x_3 = 1, \;\; x_5 = 1, \;\; x_4 = x_6 = x_7 = x_8 = x_9 = 0.$
La solution optimale de (P) est directement obtenue : $x^* = (0, 1, 1)$ et la valeur de z qui y correspond est $z = 15$. Il faut donc placer les objets O_2 et O_3 dans le sac.

Exemple 2

Soit à résoudre à l'aide de la nouvelle méthode, le programme linéaire en nombres entiers suivant :

$$\begin{cases} max\ z = x_1 - 2x_2 + x_3 \\ 2x_1 - x_2 + x_3 \leqslant 10 \\ -x_1 + 2x_2 + x_3 \leqslant 6 \\ 4x_1 + 4x_2 - 2x_3 \leqslant 2 \\ \forall i,\ x_i \in \mathbb{N} \end{cases}$$

Forme standard du programme :

$$\begin{cases} max\ z = x_1 - 2x_2 + x_3 \\ 2x_1 - x_2 + x_3 + x_4 = 10 \\ -x_1 + 2x_2 + x_3 + x_5 = 6 \\ 4x_1 + 4x_2 - 2x_3 + x_6 = 2 \\ \forall i,\ x_i \in \mathbb{N} \end{cases}$$

Résolution du programme :

	x_1	x_2	x_3	x_4	x_5	x_6
0	1	-2	1	0	0	0
10	2	-1	1	1	0	0
6	-1	2	$\boxed{1}$	0	1	0
2	4	4	-2	0	0	1

	x_1	x_2	x_3	x_4	x_5	x_6
-6	2	-4	0	0	-1	0
4	$\boxed{3}$	-3	0	1	-1	0
6	-1	2	1	0	1	0
14	2	8	0	0	2	1

Pivot égal à 3 donc différent de 1.
Contrainte de la ligne pivot : $3x_1 - 3x_2 + x_4 - x_5 = 4 \iff x_1 - x_2 + \frac{1}{3}x_4 - \frac{1}{3}x_5 = \frac{4}{3}$
Inégalité de type (1) déduite de cette contrainte : $x_1 - x_2 + \left[\frac{1}{3}\right]x_4 + \left[-\frac{1}{3}\right]x_5 \leqslant \left[\frac{4}{3}\right] \iff x_1 - x_2 - x_5 \leqslant 1$
Égalité à rajouter au tableau : $x_1 - x_2 - x_5 + x_7 = 1$

	x_1	x_2	x_3	x_4	x_5	x_6	x_7
-6	2	-4	0	0	-1	0	0
4	3	-3	0	1	-1	0	0
6	-1	2	1	0	1	0	0
14	2	8	0	0	2	1	0
1	$\boxed{1}$	-1	0	0	-1	0	1

	x_1	x_2	x_3	x_4	x_5	x_6	x_7
-8	0	-2	0	0	1	0	-2
1	0	0	0	1	$\boxed{2}$	0	-3
7	0	1	1	0	0	0	1
12	0	10	0	0	4	1	-2
1	1	-1	0	0	-1	0	1

Pivot égal à 2 donc différent de 1.
Contrainte de la ligne pivot : $x_4 + 2x_5 - 3x_7 = 1 \iff \frac{1}{2}x_4 + x_5 - \frac{3}{2}x_7 = \frac{1}{2}$
Inégalité de type (1) déduite de cette contrainte : $\left[\frac{1}{2}\right]x_4 + x_5 + \left[-\frac{3}{2}\right]x_7 \leqslant \left[\frac{1}{2}\right] \iff x_5 - 2x_7 \leqslant 0$
Égalité à rajouter au tableau : $x_5 - 2x_7 + x_8 = 0$

	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8
-8	0	-2	0	0	1	0	-2	0
1	0	0	0	1	2	0	-3	0
7	0	1	1	0	0	0	1	0
12	0	10	0	0	4	1	-2	0
1	1	-1	0	0	-1	0	1	0
0	0	0	0	0	$\boxed{1}$	0	-2	1

	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8
-8	0	-2	0	0	0	0	0	-1
1	0	0	0	1	0	0	1	-2
7	0	1	1	0	0	0	1	0
12	0	10	0	0	0	1	6	-4
1	1	-1	0	0	0	0	-1	1
0	0	0	0	0	1	0	-2	1

On a donc : $x_1 = 1,\ x_2 = 0,\ x_3 = 7,\ x_4 = 1,\ x_5 = 0,\ x_6 = 12,\ x_7 = x_8 = 0$.
La solution optimale du programme linéaire en nombres entiers est donc $(1, 0, 7)$, et la valeur de z qui y correspond est $z = 8$.

Exemple 3

Il est possible de remarquer que la nouvelle méthode s'applique bien à la résolution de problèmes de sac à dos.
Soit à remplir une boîte ne pouvant supporter que 15kg, par un, deux, trois, quatre ou cinq objets parmi les cinq objets O_1, O_2, O_3, O_4, O_5 de poids respectifs 6 kg, 3 kg, 4 kg, 8kg, 5kg et de valeurs respectives 1, 4, 2, 1, 3 tout en maximisant la valeur totale des objets placés dans la boîte. Soit x_i la variable égale au nombre d'objet O_i placé dans la boîte, $1 \leqslant i \leqslant 5$.

Formulation du programme linéaire en nombres entiers correspondant au problème :

$$\begin{cases} max\ z = x_1 + 4x_2 + 2x_3 + x_4 + 3x_5 \\ 6x_1 + 3x_2 + 4x_3 + 8x_4 + 5x_5 \leqslant 15 \\ \forall i,\ x_i \in \{0,1\} \end{cases} \Leftrightarrow \begin{cases} max\ z = x_1 + 4x_2 + 2x_3 + x_4 + 3x_5 \\ 6x_1 + 3x_2 + 4x_3 + 8x_4 + 5x_5 \leqslant 15 \\ x_1 \leqslant 1 \\ x_2 \leqslant 1 \\ x_3 \leqslant 1 \\ x_4 \leqslant 1 \\ x_5 \leqslant 1 \\ \forall i,\ x_i \in \mathbb{N} \end{cases}$$

Forme standard du programme :

$$\begin{cases} max\ z = x_1 + 4x_2 + 2x_3 + x_4 + 3x_5 \\ 6x_1 + 3x_2 + 4x_3 + 8x_4 + 5x_5 + x_6 = 15 \\ x_1 + x_7 = 1 \\ x_2 + x_8 = 1 \\ x_3 + x_9 = 1 \\ x_4 + x_{10} = 1 \\ x_5 + x_{11} = 1 \\ \forall i,\ x_i \in \mathbb{N} \end{cases}$$

Résolution :

	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_{10}	x_{11}
0	1	4	2	1	3	0	0	0	0	0	0
15	6	3	4	8	5	1	0	0	0	0	0
1	1	0	0	0	0	0	1	0	0	0	0
1	0	$\boxed{1}$	0	0	0	0	0	1	0	0	0
1	0	0	1	0	0	0	0	0	1	0	0
1	0	0	0	1	0	0	0	0	0	1	0
1	0	0	0	0	1	0	0	0	0	0	1

	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_{10}	x_{11}
-4	1	0	2	1	3	0	0	-4	0	0	0
12	6	0	4	8	5	1	0	-3	0	0	0
1	1	0	0	0	0	0	1	0	0	0	0
1	0	1	0	0	0	0	0	1	0	0	0
1	0	0	1	0	0	0	0	0	1	0	0
1	0	0	0	1	0	0	0	0	0	1	0
1	0	0	0	0	$\boxed{1}$	0	0	0	0	0	1

	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_{10}	x_{11}
-7	1	0	2	1	0	0	0	-4	0	0	-3
7	6	0	4	8	0	1	0	-3	0	0	-5
1	1	0	0	0	0	0	1	0	0	0	0
1	0	1	0	0	0	0	0	1	0	0	0
1	0	0	$\boxed{1}$	0	0	0	0	0	1	0	0
1	0	0	0	1	0	0	0	0	0	1	0
1	0	0	0	0	1	0	0	0	0	0	1

	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_{10}	x_{11}
-9	1	0	0	1	0	0	0	-4	-2	0	-3
3	$\boxed{6}$	0	4	8	0	1	0	-3	-4	0	-5
1	1	0	0	0	0	0	1	0	0	0	0
1	0	1	0	0	0	0	0	1	0	0	0
1	0	0	1	0	0	0	0	0	1	0	0
1	0	0	0	1	0	0	0	0	0	1	0
1	0	0	0	0	1	0	0	0	0	0	1

Pivot égal à 6 donc différent de 1.
Contrainte de la ligne pivot : $6x_1 + 8x_4 + x_6 - 3x_8 - 4x_9 - 5x_{11} = 3 \iff x_1 + \frac{4}{3}x_4 + \frac{1}{6}x_6 - \frac{1}{2}x_8 - \frac{2}{3}x_9 - \frac{5}{6}x_{11} = \frac{1}{2}$
Inégalité de type (1) déduite de cette contrainte :
$x_1 + \left\lfloor \frac{4}{3} \right\rfloor x_4 + \left\lfloor \frac{1}{6} \right\rfloor x_6 + \left\lfloor -\frac{1}{2} \right\rfloor x_8 + \left\lfloor -\frac{2}{3} \right\rfloor x_9 + \left\lfloor -\frac{5}{6} \right\rfloor x_{11} \leqslant \left\lfloor \frac{1}{2} \right\rfloor \iff x_1 + x_4 - x_8 - x_9 - x_{11} \leqslant 0$
Égalité à rajouter au tableau : $x_1 + x_4 - x_8 - x_9 - x_{11} + x_{12} = 0$

	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_{10}	x_{11}	x_{12}
-9	1	0	0	1	0	0	0	-4	-2	0	-3	0
3	6	0	4	8	0	1	0	-3	-4	0	-5	0
1	1	0	0	0	0	0	1	0	0	0	0	0
1	0	1	0	0	0	0	0	1	0	0	0	0
1	0	0	1	0	0	0	0	0	1	0	0	0
1	0	0	0	1	0	0	0	0	0	1	0	0
1	0	0	0	0	1	0	0	0	0	0	1	0
0	$\boxed{1}$	0	0	1	0	0	0	-1	-1	0	-1	1

	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_{10}	x_{11}	x_{12}
-9	0	0	0	0	0	0	0	-3	-1	0	-2	-1
3	0	0	0	2	0	1	0	3	2	0	1	-6
1	0	0	0	-1	0	0	1	1	1	0	1	-1
1	0	1	0	0	0	0	0	1	0	0	0	0
1	0	0	1	0	0	0	0	0	1	0	0	0
1	0	0	0	1	0	0	0	0	0	1	0	0
1	0	0	0	0	1	0	0	0	0	0	1	0
0	1	0	0	1	0	0	0	-1	-1	0	-1	1

On a donc : $x_1 = 0$, $x_2 = 1$, $x_3 = 1$, $x_4 = 0$, $x_5 = 1$, $x_6 = 3$, $x_7 = 1$, $x_8 = x_9 = 0$, $x_{10} = 1$, $x_{11} = x_{12} = 0$.
La solution optimale du problème est donc $(0, 1, 1, 0, 1)$, et la valeur de z qui y correspond est $z = 9$.
Il faut donc choisir de placer les objets O_2, O_3 et O_5 dans la boîte.

Conclusion

Pour conclure, les programmes linéaires en nombres entiers ou PLNE sont sensiblement plus complexes à résoudre qu'un programme linéaire tout court. Pourtant, la plupart des problèmes dans les situations réelles correspondent plutôt à des PLNE qu'à de simples programmes linéaires, d'où la nécessité de savoir les résoudre.

La résolution des PLNE passe habituellement par la résolution du programme linéaire relaxé c'est-à-dire du programme linéaire sans les contraintes d'intégrité. Ce programme linéaire relaxé se résout le plus souvent par la méthode du simplexe. La solution finale obtenue est optimale mais n'est pas nécessairement entière, dans ce cas la résolution du PLNE se poursuit, et cela, toujours à l'aide de la méthode du simplexe mais combinée soit à des mécanismes de coupe, les coupes de Gomory sont les plus fréquemment utilisées, soit à la méthode de séparation et évaluation.

Les coupes de Gomory permettent de couper le domaine de réalisabilité du programme linéaire de sorte à se rapprocher de la solution optimale entière d'itération en itération. La méthode de séparation et évaluation, quant à elle, est une méthode arborescente d'énumération implicite.

Les deux méthodes, celle des coupes de Gomory ainsi que celle de séparation et évaluation, passent parfois par des calculs longs, la solution optimale entière n'est pas toujours directement obtenue. La méthode décrite dans la dernière partie de cet ouvrage, présente l'intérêt d'aboutir assurément à l'optimum discret une fois la méthode arrivée à son terme, et cela, sans passer par la résolution du programme linéaire relaxé. Toutefois, il n'est pas possible d'affirmer de manière générale que telle méthode est préférable à telle autre méthode, car même si théoriquement une méthode peut s'avérer plus efficace qu'une autre, elle peut l'être moins dans le cas pratique.

Plusieurs autres méthodes de résolution des PLNE existent, certaines sont des combinaisons des méthodes déjà existantes. Toutes ces méthodes sont incessamment améliorées dans un souci d'efficacité et de gain de temps pour avoir sous contrôle le maximum possible de problèmes.

Bibliographie

✍ Christophe DUHAMEL, Fatiha BENDALI et Loïc YON . *Cours de recherche opérationnelle, programmation linéaire*. Institut Supérieur d'Informatique, de Modélisation et de leurs Applications Campus des Cézeaux. B.P.1025-63173 Aubière Cedex. `http://www.isima.fr`. 2006-2007.

✍ Alain FAYE. *Inégalités valides*. ENSIIE-Master MPRO. `http://www.ensiie.fr`. 2011-2012.

✍ Paul FEAUTRIER. *Recherche Opérationnelle, Deuxième Partie*. École Normale Supérieure de Lyon. `http://perso.ens-lyon.fr`. 29 novembre 2005.

✍ P. FOUGÈRES. *Compléments de cours sur l'algorithme du simplexe* . Université Paris Nanterre. `https://ufr-segmi.parisnanterre.fr`. 2006-2007.

✍ Pierre FOUILHOUX. *Optimisation Combinatoire : Programmation Linéaire et Algorithmes*. Université Pierre et Marie Curie. 29 septembre 2015.

✍ P. PESNEAU. *Introduction à la programmation en variables entières - Cours 4*. Université Bordeaux 1. Bât A33 - Bur 265. 2013-2014.

✍ Hugues TALBOT. *Programmation linéaire en nombres entiers, Résolution : coupes et séparation - évaluation*. Laboratoire A2SI. 26 mars 2009.

Table des matières

Étudiante : RAMONJISON Domoina Mahefasoa
033 14 663 65
domoina.ramonjison@gmail.com
Encadreur : Monsieur RANDRIANARIVONY Arthur, Professeur titulaire

UNE AUTRE MÉTHODE DE RÉSOLUTION DES PROGRAMMES LINÉAIRES EN NOMBRES ENTIERS

Résumé

Ce mémoire porte principalement sur des procédures de résolution des programmes linéaires en nombres entiers. Dans la première partie, sont donnés des définitions et des rappels utiles, l'algorithme du simplexe y est notamment rappelé, étant donné que cet algorithme est nécessaire à toutes les méthodes considérées dans cet ouvrage. La deuxième partie est consacrée à la présentation de deux méthodes de résolution les plus usitées des programmes linéaires en nombres entiers, la méthode par les coupes de Gomory et la méthode de séparation et évaluation. Ces méthodes présentent l'inconvénient de devoir passer par la résolution du programme linéaire relaxé pour déterminer l'optimum continu du problème avant de commencer la recherche de l'optimum discret, ce qui a donné l'idée d'établir une méthode donnant directement cet optimum discret. Ainsi, la troisième et dernière partie de ce mémoire, à titre de contribution personnelle, expose une petite modification de l'algorithme du simplexe afin d'obtenir, de manière directe, la solution optimale d'un programme linéaire en nombres entiers donné.

Abstract

This book is mainly focused on procedures of resolution of linear programs in integer numbers. In the first part, useful definitions and reminders are given, the algorithm of the simplex is in particular pointed out there, since this algorithm is necessary to all the methods considered in this work. The second part is devoted to the presentation of two most used methods of resolution of linear programs in interger numbers, the method by the cuts of Gomory and the method of separation and evaluation. These methods present the disadvantage of having to go through the resolution of the released linear program to determine the continuous optimum of the problem before beginning the research of the discrete optimum, which gave the idea to establish a method giving directly this discrete optimum. So, the third and last part of this book, as a personal contribution, exposes a small modification of the algorithm of the simplex in order to obtain, in a direct way, the optimal solution of a given linear program in integer numbers.