REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE

Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

Université Mouloud Mammeri de Tizi Ouzou

Faculté des Sciences

Département de Mathématiques

Projet de fin d'étude

En vue de l'obtention du Diplôme de Master

En Recherche Opérationnelle

THEME

Optimisation Discrète

Cas uni-critère et multicritère

Présenté par : M^{elle} BOUDERSA Lynda

Encadré par : M^{me} RABIA

Membres du jury :

Président : M. AIDENE Professeur, U.M.M.T.O

Rapporteur : F. RABIA, M C B, U.M.M.T.O

Examinateur : B. SADI, M. C, A U.M.M.T.O

Examinateur : F. ACHEMINE, M. A, A , U.M.M.T.O

Promotion : 2010/2011

Remerciements

Nous tenions tout d'abord à remercier Dieu tout puissant d'avoir guidé nos pas vers les portes du savoir tout en illuminant notre chemin, nous avoir donné suffisamment du courage, la force et la volonté pour la réalisation de ce modeste travail.

Je tiens à exprimer mes profondes gratitudes à mes parents pour leur protection, leur engagement incessant à me prendre en charge durant toutes les étapes de ma vie, leur apport affectif et matériel inconsiderable (Que Dieu les protège et veille sur eux).

Ma reconnaissance va à ma promotrice Mme RABIA, qui a accepté de m'encadrer pour la réalisation de ce travaille, j'exprime mes profonds remerciements pour son aide, ses conseils, sa compréhension tout au long de ce travail.

Mes remerciements vont aux membres du jury, d'avoir accepté d'examiner ce travail.

Mes remerciements vont également à tous ceux et toutes celles qui ont, de prés ou de loin, contribué à la réalisation de ce travail.

A touts, merci.

Dédicaces

Je dédie mon modeste travail :

- ***A mes très chers parents***
- ***A mes frères Massinisa et sofian***
- ***A ma sœur Dehboucha***
- ***A ma grande mère***
- ***A ma tente Nora***
- ***A mes tentes maternelles et paternelle***
- ***A mes oncles maternels et paternels***
- ***A tous mes amis surtout Aghiles, Mouloud, Saida, ghania, Nawal et Ouardia***
- ***A tous ceux qui m'aime***
- ***A toute la promotion de recherche operationnelle***

Lynda

Introduction générale

Première partie :

Chapitre I : programmation discrète uni-objectif

Chapitre II : programmation discrète multi-objectifs

Deuxième parie :

Programmation de la méthode de Branch and Bound sur MATLAB

La recherche opérationnelle ou bien ce qu'on appelle l'aide à la décision est un outil puissant qui nous permet d'apparaitre dans des secteurs très divers.

Les problèmes à résoudre peuvent fréquemment être exprimé sous la forme générale d'un problème d'optimisation, dans le quel on défini une fonction objectif (économique) qu'on cherche à maximiser (minimiser) par rapport à tous les paramètres concernés (variables de décision), sous un ensemble de contraintes, la solution proposée doit respecter toutes ces contraintes.

L'aide à la décision, c'est de donner les éléments de réponse au décideur, donc lui fournir un ensemble d'alternatifs ou solutions qu'il sera capable d'envisager sans risque. Souvent dans la vie actuelle, ces problèmes exigent la considération simultanée de plusieurs objectifs (critères) qui sont généralement conflictuels, dans ce cas on se trouve face à l'aide de décision multicritères (multi-objectifs) qui fait partie du domaine de l'aide à la décision. Ces problèmes ont intéressé plusieurs auteurs dont J. Teghem Jr.et P.L. Kunsch (1986), R.E. Steuer (1986), D. Klein, E. Hannan (1982), S.Zionts (1977), B.Roy (1971).

L'étude consiste à fournir au décideur des outils lui permettant de progresser dans la résolution d'un problème de décision, en tenant compte de tous ces critères souvent conflictuels. Les méthodes d'aide à la décision n'ont pas pour objectif la détermination de solutions optimales satisfaisant tous les critères simultanément, mais elles permettent d'apporter une piste pour trouver la meilleure solution possible.

Dans les problèmes d'optimisation on distingue trois cas :

- Optimisation continue où les variables de décision sont continues,
- Optimisation discrète où les variables de décision sont discrets,
- Optimisation avec des variables mixtes,

La problématique de l'optimisation uni-objectif et multi-objectifs en variables binaires (0,1) et en variables mixtes ne seront pas discutées dans ce mémoire, nous renvoyons le lecteur intéressé à (Bitran,1979), (Zionts,1979), (Deckro et Winkofsky, 1983), (Teghem et Ulungu, 1994), (Ulungu et Teghem, 1995), (Gandibleux et al., 1996), (Puaro 1999), (Teghem , 2000), (Tuyttens 2000),(R.J. Dakin ,1965), (R. Breu, C.A. Burdet (1974).

Le but de ce travail est d'essayé de faire une synthèse sur l'optimisation discrète uni-objectif et multi-objectifs, de voir quelques méthodes de résolution et de programmer vers la fin l'un de ces algorithmes traités.

Pour cela on a partitionné le travail en deux parties importantes :

La première partie est constitué de deux chapitres :

- Le premier chapitre : traite l'optimisation uni-critère discrète où on a donné des algorithmes de résolutions pour le cas linéaire et le cas non linéaire.
- Le deuxième chapitre : traite l'optimisation discrète multicritère, et dans ce concept nous exposons deux approches < Sylva et Crema > et < M.Moulai et M.Abbas >

La deuxième partie : on a programmé la méthode de Branch and Bound qu'on a décrétée au premier chapitre, cette méthode qui nous sert aussi à trouver la première solution efficace à la première étape des algorithmes évoqués au chapitre II.

Nous terminons par une conclusion générale.

1. Introduction

Dans la plupart des problèmes d'optimisation linéaire ou non linaire, les variables considérées sont supposées à valeurs réelles or dans certaines applications pratiques on peut avoir le cas ou toutes les variables sont entières, dans ce cas les variables présentent un nombre d'unité non fractionnables.

2. Cas linéaire

2.1 Formulation du problème

Un programme linéaire en nombres entiers (PLNE) se présente comme suit

$$(P)\quad \begin{array}{c} (min)\, Max\, Z = CX \\ \leq, \geq, = ¿ \\ ¿ \\ ¿ \\ b \\ A\, X\, (¿ x_j \in N, j = 1 \ldots n ¿ \\ ¿ \end{array}$$

Où $C \in R^n$, $A \in R^{n*m}$, $b \in R^m$.

Le programme relaxé de (PLNE) est donné comme suit :

$$(PL)\quad \begin{array}{c} (min)\, Max\, Z = CX \\ \leq, \geq, = ¿ \\ ¿ \\ ¿ \\ b \\ A\, X\, (¿ x_j \geq 0, j = 1 \ldots n ¿ \\ ¿ \end{array}$$

2. 2 Méthodes de résolution

2.2.1 Méthode des troncatures (algorithme de GOMORY) [3]

La méthode des troncatures appelée également « algorithme du plan sécant » ou « méthode des coupes» a été développée par Ralph. GOMORY en 1958.

Algorithme de GOMORY :

1. Etant donné un (PLNE), résoudre le PL correspondant à l'aide d'un algorithme du simplexe. Si la solution optimale du PL est entière, elle est également une solution optimale au PL NE. La résolution est terminée.
2. Si une ou plusieurs variables de base dans la solution optimale du PL ne sont pas entières, on doit alors générer à partir d'une des lignes du tableau (celle dont la partie fractionnaire pour la variable de base correspondante est plus élevée) une contrainte supplémentaire dite coupe de GOMORY. Cette contrainte est ajoutée au tableau optimal du PL et on détermine le nouveau tableau optimal à l'aide de la méthode duale du simplexe.
3. Si les variables de bases dans le nouveau tableau optimal sont entières, nous avons obtenu la solution optimale du PLNE. La résolution est terminée.
4. Sinon, on doit générer à partir du dernier tableau optimal une nouvelle coupe de GOMORY, l'ajouter au dernier tableau optimal et trouver la solution optimale à l'aide de la méthode duale du simplexe.
   Si la solution optimale obtenue est à valeurs entières, la résolution est terminée.
   Sinon on répète la procédure jusqu'à l'obtention d'une solution optimale à valeurs entières,

Règle pour générer la coupe de GOMORY

On choisira, dans le tableau optimal, la variable de base ayant la plus grande partie fractionnaire et on déduit l'équation de la coupe en exprimant toutes les variables du tableau en fonction de la valeur x_{Bi} de la variable de base.

Formellement la coupe de Gomory est générée comme suit :

$$x_{Bi} + \sum_{j \in J} a_{ij} x_j = b_i \qquad (1.1)$$

Où $J = \{ j : j \, indice \, de \, variable \, hors \, base \} \; j = 1, n$

Notons par $\lfloor a \rfloor$ la partie entière inferieur à a . Dans ce cas,

$$x_{Bi}+\sum_{j\in J}\lfloor a_{ij}\rfloor x_j+\sum_{j\in J}f(a_{ij})x_j=\lfloor b_i\rfloor+f(b_i) \qquad (1.2)$$

Où $f(a)=a-\lfloor a\rfloor$. L'équation (1.2) est également égale à :

$$\lfloor a_{ij}\rfloor x_j=\lfloor b_i\rfloor+¿f(b_i)-\sum_{j\in J}f(a_{ij})x_j \qquad (1.3)$$
$$x_{Bi}+\sum_{j\in J}¿$$

Nous pouvons écrire (1.3) sous forme d'une inéquation :

$$\lfloor a_{ij}\rfloor x_j\leq\lfloor b_i\rfloor+¿f(b_i) \qquad (1.4)$$
$$x_{Bi}+\sum_{j\in J}¿$$

Cette inégalité satisfait aussi :

$$x_{Bi}+\sum_{j\in J}\lfloor a_{ij}\rfloor x_j\leq\lfloor b_i\rfloor \qquad (1.5)$$

De (1.1) nous avons :

$$x_{Bi}=b_i-\sum_{j\in J}a_{ij}x_j \qquad (1.6)$$

Nous remplaçons (1.6) dans (1.5) , nous aurons l'expression la coupe de Gomory

$$\sum_{j=1}^{n}f(a_{ij})x_j\geq f(b_i) \qquad (1.7)$$

Exemple numérique :

Soit le programme linéaire suivant :

$$max\,Z=100\,x_1+120\,x_2$$

$$3x_1+4x_2\leq 4100$$

$$x_1+3x_2\leq 2400$$

$$2x_1+2x_2\leq 2625$$

$$x_1, x_2\in N$$

En ajoutant les variables d'écart , nous obtenons le programme:

$$max\, Z=100x_1+120x_2$$

$$3x_1+4x_2+x_3=4100$$

$$x_1+3x_2+x_4=2400$$

$$2x_1+2x_2+x_5=2625$$

$$x_j\in N\,,\ j=\overline{1,5}$$

En appliquant l'algorithme du simplexe, on obtient le tableau optimal suivant :

c_j		100 120 0 0 0	
c_B	x_j de base	$x_1 x_2 x_3 x_4 x_5$	b
100 120 0	x_1 x_2 x_4	1 0 -1 0 2 0 1 1 0 $-3/2$ 0 0 -2 1 $5/2$	1150 162.5 762.5
z_j c_j-z_j		100 120 20 0 20 0 0 -20 0 -20	$Z=134\,500$

La solution n'est pas entière $\Rightarrow$ construction de la coupe de GOMORY

Ici, comme la parie fractionnaire est identique, nous choisissons arbitrairement la 2^{eme} ligne du tableau.

La variable de base qu'on veut rendre entière est donc x_2 .

La coupe de Gomory est :

$$\frac{1}{2}x_5 \geq \frac{1}{2}$$

Pour appliquer l'algorithme duale du simplexe, il faut multiplier la contrainte de Gomory par (-1) .

$$\frac{-1}{2}x_5 \leq -\frac{1}{2}$$

En introduisant la variable d'écart x_6 , on obtient :

$$\frac{-1}{2}x_5 + x_6 = \frac{-1}{2}$$

Ajoutant cette équation au tableau optimal :

c_j		100 120 0 0 0 0	
c_B	x_j de base	$x_1 x_2 x_3 x_4 x_5 x_6$	b

100 120 0 0	x_1 x_2 x_4 x_6	1 0 0 0 0	0 1 0 0 0	-1 1 -2 0	0 0 1 0	2 $-3/2$ $5/2$ $-1/2$	0 1	1150 162.5 762.5 $-1/2$
z_j $c_j - z_j$		100 0	120 0	20 -20	0 0	20 -20	0 0	$Z = 134\,500$
σ_j		–	–	–	–	40	-	

x_6 est la variable sortante. $\sigma_j = \frac{c_j - z_j}{a_{ij_0}} \{ a_{ij_0} < 0$

Où j_0 est l'indice de la variable sortante.

Le nouveau tableau après le pivotage est :

c_j		100	120	0	0	0	0	
c_B	x_j de base	x_1	x_2	x_3	x_4	x_5	x_6	b
100 120 0 0	x_1 x_2 x_4 x_5	1 0 0 0	0 1 0 0	-1 1 -2 0	0 0 1 0	1 0 0 1	4 -3 5 -2	1148 164 760 1
z_j $c_j - z_j$		100 0	120 0	20 -20	0 0	0 0	40 -40	$Z = 134\,480$

La solution optimale est : $x_1 = 1148, x_2 = 164 \, et \, Z = 134\,480$

La solution est entière donc la procédure est terminée.

2.2.2 Méthode de séparation et d'évaluation successives [3]

(Branch and Bound methods) (B&B)

Les méthodes arborescentes sont des méthodes exactes d'optimisation qui pratiquent une énumération de l'ensemble des solutions. Il partage l'espace des solutions en sous ensembles de plus en plus petits, la plupart étant éliminée par les calculs des bornes.

Algorithme de B&B : (dans le cas de maximisation)

1. **Résolution du PL avec un algorithme du simplexe**
   Résoudre le PL correspondant sans la contrainte de variables entières. La valeur de Z du tableau optimal donne une borne supérieure au (PLNE).
2. **Séparation**

- Choisir une variable non entière pour générer deux sous -problèmes. Soit x_k cette variable. Représentons par [x_k] la partie entière de x_k .
- Pour éliminer la solution non entière x_k , on crée deux branches (deux sous-problèmes). On obtient une branche avec $x_k \leq$ ¿] et l'autre avec $x_k \geq [x_k] + 1$.

  Les deux sous-problèmes sont obtenus en ajoutant au programme linéaire (PL) précédent la contrainte $x_k \leq [x_k]$ pour un des sous-problèmes et la contrainte $x_k \geq [x_k] + 1$ pour l'autre sous-problème.

  Toutes les solutions entières réalisables sont maintenant contenues dans l'un ou l'autre des sous problèmes.

3. **Résolution des sous-problèmes et détermination de nouvelles bornes pour Z s'il ya lieu**

- Il s'agit de résoudre chaque sous-problème par l'algorithme du simplexe.

- Notons par MBSD, la meilleure borne supérieure disponible de Z. Elle correspond à la valeur maximale de la fonction objectif des deux sous-problèmes que nous venons de résoudre.
- Notons par MBD, la meilleure borne disponible. Elle correspond à la valeur maximale de la fonction objectif de toutes les solutions entières obtenues jusqu'à présent.

4. Critère d'arrêt de l'exploitation d'une branche

L'exploitation d'une branche est terminée lorsque l'une des conditions suivantes est satisfaite :

- Le sous-problème de la branche considérée n'admet pas de solution réalisable.
- Une branche est explorée lorsque la solution associée est entière.
- Le sous-problème de la branche considérée admet une solution optimale non entière mais sa valeur Z^{i} est inférieure ou égale à celle d'un autre sous-problème.

5. Critère d'optimalité

La solution optimale au PLNE est la solution entière qui donne la meilleure borne pour Z (MBD)

6. pour continuer le processus de séparation (aller à 2)

Si la solution optimale de l'un ou l'autre des sous problèmes contient au moins une variable non entière, on poursuit l'étape de séparation à partir du sous-problème ayant la valeur la plus élevée de $Z^{i}(MBSD)$.

Exemple numérique : voir la deuxième partie.

3. Cas non linéaire

3.1 Formulation du problème

La forme générale d'un programme non linéaire en nombres entiers se présente comme suit :

$$(PN) \quad \begin{cases} Max\, f(x) \\ g_i(x) \le b_i \; i = 1 \ldots m \\ x \in X = \{ x \in Z^n / L_j \le x_j \le U_j \; j = 1 \ldots n \} \end{cases}$$

Où f et les g_i sont des fonctions à valeurs réelles, L_j et U_j sont des nombres entiers avec $L_j < U_j$ pour $j = 1 \dots n$.

Dans ce paragraphe nous décrirons deux méthodes de résolution essentielles de (PN) : la méthode « Poly-blocs » et la méthode de transformation d'un système non linéaire à un système linéaire.

3.2 La méthode « Poly-blocs » pour (PN) [2]

Cette méthode suppose que f et les g_i sont des fonctions croissantes sur $[L_j, U_j]$ pour $j = 1 \dots n$,

Considérons la fonction

$$G(x) = \max_{i=\overline{1,m}} \{ g_i(x) - b_i \} \qquad (1.8)$$

La frontière de la région réalisable peut être exprimée par $\Gamma = \{ x \in X / G(x) = 0 \}$

Soit $S = \{ x \in X \subset Z^n / g_i(x) \leq b_i \, i = 1 \dots m \}$

Soit (α, β) un box dans X, avec $\alpha \in S$ et $\beta \notin S$.

Supposons $G(\alpha) < 0$.

Soit x_b un point d'intersection de la droite $x = \lambda \alpha + (1 - \lambda)\beta$, $0 \leq \lambda \leq 1$ avec Γ.

Puisque $G(\alpha)<0$ et $G(\beta)>0$, il existe un point x_b dans X qui satisfait $G(x_b)=0$.

Notons par $\lfloor x \rfloor$ le vecteur dont chaque composante x_i est égale à la parie entière inferieure à x_i $i=1,n$.

Notons par $\lceil x \rceil$ le vecteur dont chaque composante x_i est égale à la parie entière supérieure à x_i $i=1,n$.

Soit $x^F=\lfloor x_b \rfloor$ et $x^I=\lceil x_b \rceil$.

On peut remarquer que $x^F \in S$ et $x^I \notin S$.

Considérons deux boxes entiers $\langle \alpha, x^F \rangle$ et $\langle x^I, \beta \rangle$.

Par la monotonie de f et des g_i, il n'y a aucun point réalisable meilleur que x^F dans $\langle \alpha, x^F \rangle$ et il n'y a aucun point réalisable dans $\langle x^I, \beta \rangle$.

Par conséquent, lorsqu'on cherche une solution optimale pour (PN), on peut supprimer les boxes $\langle \alpha, x^F \rangle$ et $\langle x^I, \beta \rangle$ de $\langle \alpha, \beta \rangle$ après avoir comparé x^F avec la solution courante.

Pour construire les boxes des nombres entiers nous utilisons les formules suivantes :

Corollaire 1.1: soit $A=\langle \alpha, \beta \rangle$, $B=\langle \alpha, \gamma \rangle$ et $C=\langle \gamma, \beta \rangle$ où $\alpha \le \gamma \le \beta$ alors $\overline{A}$ et $\overline{A}$ peut être partitionner au n nouveau subboxes

$$A = ¿ i=1 ¿ n \left(\prod_{j=1}^{i-1} \langle \alpha_j, \gamma_j \rangle * \langle \gamma_i + 1, \beta_i \rangle * \prod_{j=i+1}^{n} \langle \alpha_j, \beta_j \rangle \right) (1.9)$$

$$A = ¿ i=1 ¿ n \left(\prod_{j=1}^{i-1} \langle \gamma_j, \beta_j \rangle * \langle \alpha_i, \gamma_i - 1 \rangle * \prod_{j=i+1}^{n} \langle \alpha_j, \beta_j \rangle \right) (1.10)$$

Algorithme « Poly-blocs »

Etape 0 : Initialisation

Soient $l_1, l_2 \ldots l_n$ ¿ ¿ $L = ¿$ et u $(¿ ¿ 1, u_2 \ldots u_n)^T$ $U = ¿$.

Si L est non réalisable, alors le problème (PN) n'a pas de solution réalisable, si U est realisable, alors U est la solution optimale pour (PN), stop

Sinon poser $x_{opt} = L$, $f_{opt} = f(x_{opt})$, $x^1 = \{\langle L, U \rangle\}$ et poser $K = 1.$

Etape 1 : sélection de box et recherche d'un point de la frontière

Sélectionner un box $\langle \alpha, \beta \rangle \in X^K$

Soit $X^K = X^K \langle \alpha, \beta ¿ ¿$ ¿ , trouvons la racine $\lambda^¿$ de l'équation suivante :

$$G(\lambda \alpha + (1-\lambda)\beta) = 0, \lambda \in [0,1] (1.11)$$

Où G est définie en (1.8).

Poser $x_b = \lambda^¿ \alpha + (1 - \lambda^¿) \beta$

Poser $x^F = \lfloor x_b \rfloor$, $x^I = \lceil x_b \rceil$

Remarque : si $x^F = x_b$ poser $x^I = x_b + e_j$ avec $x_b + e_j \leq \beta$. (e_j est le vecteur de base canonique)

Si $f(x^F) > f_{opt}$, poser $x_{opt} = x^F$ et $f_{opt} = f(x^F)$.

Etape 2 : Partition et suppression

1. Appliquer la formule (1.10) pour partitionner l'ensemble $\Omega_1 = \langle \alpha, \beta \rangle \setminus x^I, \beta$ en une union de boxes de nombres entiers.
   Soit $x^F \in \langle \widetilde{\alpha}, \widetilde{\beta} \rangle \in \Omega_1$. Posons $\Omega_1 := \Omega_1 \setminus \langle \widetilde{\alpha}, \widetilde{\beta}$.
2. Appliquer la formule (1.9) pour partitionner l'ensemble $\Omega_2 = \langle \widetilde{\alpha}, \widetilde{\beta} \rangle \setminus \langle \widetilde{\alpha}, x^F$.
3. Soit $y^K = \Omega_1 \cup \Omega_2$.
4. Pour chaque box obtenu dans le processus de partition au dessus, effectuer les tests suivants :
   a. Si β est réalisable, supprimer $\langle \alpha, \beta \rangle$ de y^K .
      De plus si $f(\beta) > f_{opt}$, poser $x_{opt} = \beta$ et $f_{opt} = f(\beta)$.
   b. Si α est non réalisable, supprimer $\langle \alpha, \beta \rangle$ de y^K .
   c. Si $f(\beta) \leq f_{opt}$, supprimer $\langle \alpha, \beta \rangle$ de y^K .
   d. Si α est réalisable, β est non réalisable et $f(\alpha) > f_{opt}$, poser $x_{opt} = \alpha$ et $f_{opt} = f(\alpha)$.
      Noter par z^K l'ensemble de boxes restants à la fin du processus de suppression ci-dessus.

Etape 3 : Mise à jour des boxes

Supprimer tous les boxes $\langle \alpha, \beta \rangle$ dans X^K avec $f(\beta) \leq f_{opt}$.

Poser $X^{K+1} = X^K z^K$.

Si $X^{K+1}=\varnothing$. Stop

Sinon poser $K:=K+1$ et aller a l'étape 1.

Remarque : on peut utiliser la méthode de bissection ou bien la méthode de Newton pour trouver la racine de l'équation (1.11) .

3.3 Méthode pour transformer un système non linéaire en un système linéaire (l'algorithme de KCG (Kelley et Cheney et Goldstein [8]) :

La méthode s'applique aux programmes non linéaires convexes. Un programme non linéaire peut être facilement modifié pour avoir un programme linéaire équivalent.

Le principe fondamental de cette méthode est d'approximer le domaine réalisable d'un programme non linéaire par un ensemble fini de demi-espaces fermés puis de résoudre la séquence des programmes linéaires approximés.

Considérons le problème de programmation convexe suivant

$$(PNC)\begin{cases} Min\, f(x)=c^t x \\ g_i(x)\geq 0 \; i=1\ldots m \end{cases}$$

Où $x=(x_1,x_2\ldots x_n)^t\in R^n$ et $g_1,g_2\ldots g_m$ sont des fonctions concaves dans R^n ,

Soit $G=\{x: g_i(x)\geq 0, i=1,\acute{}m\}$ et supposons que G est continu dans un ensemble compact $T\subset R^n$, définie par un ensemble fini d'inégalités linéaires

$T=\{x: x\in R^n, Ax\geq b\}$

Pour $x\in T$, soit $\partial g_i(x)\subset R^n$ l'ensemble des sous gradients, ou sous différentiel de g_i

Prenant $\xi\in\partial g_i(x)$ alors

$g_i(y)\leq g_i(x)+\xi^t(y-x)$ Pour chaque $y\in R^n$

Si g_i est différentiable en x , alors $\xi=\nabla g_i(x)$, nous supposons que $\partial g_i(x)\neq 0\ i=\overline{1,m}$

On va donner les étapes de l'algorithme de découpage (Kelley et Cheney et Goldstein) pour résoudre le problème (PNC) :

1. Résoudre le programme linéaire $Min f(x)=c^t x$, $x\in T$ et soit x^0 sa solution optimale.
   Si x^0 est continue dans l'ensemble $G(\epsilon)=\{x: x\in T, g_i(x)\geq -\epsilon, i=1,\overline{m}\}$
   Où $\epsilon>0$, stop ; l'optimum de (PNC) est atteint.
   Sinon poser $k=0$ et aller a l'étape 2.
2. Soit $x^k\in T$, tel que $x^K\notin G(\epsilon)$, trouver l'indice s_k par

$g_{s_k}(x^k)=min\{g_i(x^k), i=\overline{1,m}\}<0$ et sélectionner $\xi^k\in\partial g_{s_k}(x^k)$.

Résoudre le programme linéaire suivant

$$\begin{cases} min\, c^t x \\ \tilde{g}_{s_h}(x,x^h)=g_{s_h}(x^h)+(\xi^h)^t(x-x^h)\geq 0, h=\overline{0,k} \\ x\in T \end{cases}$$

3. Soit x^{k+1} la solution optimale du programme linéaire précédent.
   Si $x^{k+1}\in G(\varepsilon)$, stop, sinon poser $k=k+1$ et aller a 2.

Exemple numérique : pour illustrer l'application de l'algorithme Considérons le problème convexe suivant :

$$(i)\begin{cases} \min f(x)=4x_1+5x_2 \\ sc \\ g_1(x)=-2(x_2)^2-2x_1x_2-2(x_2)^2+4\geq 0 \\ g_2(x)=-(x_1)^2-(x_2)^2+4x_1-3\geq 0 \\ T=\{x: x\in R^2, 0\leq x_1\leq 4, -4\leq x_2\leq 4\} \end{cases}$$

La première itération il s'agit de résoudre le problème suivant :

$$(1)\begin{cases} \min f(x)=4x_1+5x_2 \\ sc \\ x\in T \end{cases}$$

La solution optimale pour (1) est $x^0=(0,-4)$ où $f(x^0)=-20$

$g_1(x^0)=-28$, $g_2(x_0)=-19$ et $s_0=1$.

On voix que la contrainte $g_1(x)$ est la plus violée au point x_0 , alors on construit la contrainte linéaire suivante : $g_1(x^0)+(x-x^0)\nabla g_1(x^0)\geq 0$ (¿)

Après le calcul on trouve (¿)=¿ $2x_1+4x_2+9\geq 0$

On ajoute cette contrainte au problème (1) :

$$(2)\begin{cases} \min f(x)=4x_1+5x_2 \\ sc \\ 2x_1+4x_2+9\geq 0 \\ 0\leq x_1\leq 4 \\ -4\leq x_2\leq 4 \end{cases}$$

Nous trouvons la solution optimale $x^1=\left(\frac{7}{2},-4\right)$, $f(x^1)=-16$,

$g_1(x^1)=\frac{-49}{4}$, $g_2(x^1)=\frac{-69}{4}$ et $s_1=2.$ donc on ajoute la contrainte

$g_2(x^1)+(x-x^1)\overset{¿*¿}{\nabla} g_2(x^1)\geq 0$ ¿

Après le calcul on trouve $\begin{matrix}¿*¿\\¿\\¿\end{matrix}$ $-2x_1+32x_2+101\geq 0$

On ajoute cette contrainte au problème (2) on obtient :

$$(3)\begin{cases} min\, f(x)=4x_1+5x_2 \\ sc \\ 2x_1+4x_2+9\geq 0 \\ -2x_1+32x_2+101\geq 0 \\ 0\leq x_1\leq 4 \\ -4\leq x_2\leq 4 \end{cases}$$

La solution optimale correspondante au problème (3) est $x_2=\left(\frac{-29}{28},\frac{-155}{56}\right)$ avec

$f(x^0)=-9.69$, cette dernière solution est la solution du problème de départ (i)

Remarque : l'objectif de l'algorithme consiste à transformer un système d'équations non linéaires en un système d'équations linéaires. La contrainte non linéaire est remplacée par son hyperplan d'appui (la coupe) qui est une contrainte linéaire. Les variables de décision peuvent être continues, discrètes ou bien mixtes. Dans le cas où les variables de décision sont discrètes, on résout le problème par l'une des méthodes décrites ci- dessus.

1. Introduction

Dans la pratique, modéliser un problème avec une seule fonction objectif s'avère insuffisant pour exprimer tous les besoins et toutes les considérations d'un problème donné, alors on fait appelle à l'optimisation multicritère. Dans ce cas le décideur prend plusieurs critères à optimiser sous un ensemble de contraintes.

Soulignons par ailleurs que le point crucial dans les problèmes multi-objectifs est bien le concept d'optimalité. Du fait que les objectifs sont souvent conflictuels, il n'est pas possible de les optimiser simultanément alors on parle de notion d'efficacité.

Plusieurs méthodes on été développées pour résoudre ces problèmes.

2. Formulation du problème :

Un programme linéaire multicritère discret $(PLMD)$ peut être formulé comme suit :

Soient $r \geq 2$ un nombre entier et $c^1, c^2, .. c^r \in R^n$ des vecteurs lignes,

C une matrice $r \times n$ dont la i^{eme} ligne est donnée par $c^i, i = 1,2 \ldots r$ et

$S = \{x \in R^n\, b, A \in R^{m*n}, b \in R^m, x \geq 0\}$ un polyèdre non vide et compact dans R^n . Alors le problème linéaire en nombre entiers multi- objectifs est donné par :

$$(P)\begin{cases} \text{Max } Z(x) = C(x) \\ x \in S \\ x \in N^n \end{cases}$$

Notons par $F(P)$ l'ensemble des solutions réalisables de (P) .

La notion d'optimalité n'existe pas car les objectifs sont conflictuels, on parle de la solution non dominée ou bien de la solution efficace.

3. Définitions et théorèmes

Définition 2.1 : une solution $x^0 \in S$ est dite efficace si et seulement si il n'existe pas une autre solution $x^1 \in S$ tel que $Z_i(x^1) \geq Z_i(x^0), i \in \{1,\dots r\}$ et $Z_i(x^1) > Z_i(x^0)$ pour au moins un $i \in \{1,\dots r\}$.

S'il existe une telle solution x^1 alors le point x^0 est appelé solution dominée.

Une solution efficace est appelé également solution non dominée ou bien solution Pareto optimale.

Un résultat bien connu reliant la programmation multi-objectifs et la programmation paramétrique est le suivant [11] :

Théorème 2.1 : si $x^¿$ est une solution optimale pour un problème uni-objectif :

(P') $max\{\lambda^t Cx : x \in S\}$ $\lambda \in R^r, \lambda > 0$ alors $x^¿$ est efficace pour (P) .

Les solutions efficaces qui sont optimales pour le problème paramétrique (P') du théorème 1 sont appelées solutions efficaces de support (supported solutions).

A la différence de la programmation linéaire multi-objectifs, le réciproque de ce théorème ne tient pas pour $PLMD$, car quelques solutions efficaces (connu sous le nom de solutions efficaces non de support (unsupported solutions) peuvent ne pas être optimales pour tout $\lambda > 0$.

Proposition 2. 1 : soient $x^1, x^2, \dots x^l$ des solutions efficaces du problème (P) et

$D_s = \{x \in N^n : C\,x \leq C\,x^s\}$

Soit $x^¿$ une solution efficace pour le problème multi- objectifs discret suivant

$(PLMD_l)$: $\max\{Cx : x \in F(P) - \cup_{s=1}^{l} D_s\}$

alors $x^{¿}$ est une solution efficace pour (P) .

Corollaire 2. 1 : soient $x^1, x^2, \ldots x^l$ des solutions efficaces du problème (P) et

$$D_s = \{x \in N^n : C x \leq C x^s\}$$

Si $x^{¿}$ est une solution optimale pour $\begin{matrix} \lambda^t Cx : x \in F(P) - ¿s = 1 \\ (PL_l) : max\,(¿l D_s) \end{matrix}$ pour un $\lambda \in R^r, \lambda > 0$,

alors $x^{¿}$ est une solution efficace pour le problème (P) .

4. Méthodes de résolution :

Ils existent plusieurs méthodes pour résoudre le problème (P) , parmi ces méthodes : la méthode de Laumanns (2006) [6] , la méthode de M.Özlen, M.Azizoglu (2008) [9], et autres.., Et dans ce chapitre on a choisi d'exposer deux méthodes : méthode de Sylva et Crema (2004) Et la méthode de M. Abbas et M. Moulaï (1999).

4.1. Méthode 1 : Algorithme de Sylva et Crema [5]

Les propriétés précédentes peuvent être utilisées pour implémenter un algorithme pour $(PLMD)$.

Après le choix d'un vecteur poids λ , la première étape de l'algorithme consiste à résoudre le problème uni-critère paramétrique $(P_0) : max\{\lambda^t Cx : Ax = b, x \in N^n\}$ en utilisant l'un des algorithmes cité au chapitre I.

Si ce problème est irréalisable, alors le problème (P) est irréalisable ; sinon la solution x^1 trouvée est selon le théorème (2.1) une solution efficace pour le problème (P) .

Après l étapes du processus, si le problème (P_{l-1}) est irréalisable alors c'est la fin de l'algorithme ; sinon une nouvelle solution efficace x^l est trouvé et un nouveau problème (P_l) est défini en supprimant toutes les solutions réalisable de (P_{l-1}) tel que $Cx \leq C\,x^l$. Ceci met en application l'ajout des contraintes linéaires au (P_{l-1}) :

$$(Cx)_k \geq \left((Cx^l)_k + 1\right) y_k^l - M_k\left(1 - y_k^l\right) \; pour\; k = 1,\ldots r \qquad (2.1)$$

$$\sum_{k=1}^{r} y_k^l \geq 1 \; y_k^l \in \{0,1\} \; pour\; k = 1,..r. \qquad (2.2)$$

Où $-M_k$ est une limite inférieure pour n'importe quelle valeur réalisable de la k^{eme} fonction objectif.

L'ajout de ces contraintes est équivalent à supprimer l'ensemble D_l de la région réalisable, le problème P_l est linéairement équivalent à PL_l . N'importe qu'elle solution à ce problème sera efficace pour le problème P_l .

Le problème P_l est donné comme suit

$$(P_l)\begin{cases} max\; \lambda^t Cx \\ Ax = b \\ (Cx)_k \geq \left((Cx^s)_k + 1\right) y_k^s - M_k\left(1 - y_k^s\right),\; pour\; s = \overline{1,l}\,, k = \overline{1,r} \\ \sum_{k=1}^{r} y_k^s \geq 1,\; y_k^s \in \{0,1\}\; pour\; s = \overline{1,l}\; k = \overline{1,r} \\ x \in N^n \end{cases}$$

Répétons la procédure jusqu'à l'irréalisation, cette séquence de problèmes produira un ensemble de vecteurs non dominés.

Exemple numérique 2.1:

Considérons un $PLMD$ suivant :

$$(P)\begin{cases}\max\ x_1-2x_2\\ \quad -x_1+3x_2\\ sc\ x_1-2x_2\le 0\\ x_1,x_2\in\{0,1,2\}\end{cases}$$

Pour cet exemple $M_1=4$ et $M_2=2$.

Nous choisissons $\lambda=(4,3)^t$.

Avec ces valeurs on obtient le problème P_0 suivant

$$(P_0)\begin{cases}max\ x_1+x_2\\ sc\ x_1-2x_2\le 0\\ x_1,x_2\in\{0,1,2\}\end{cases}$$

La solution optimale à ce problème linéaire discret (PLD) est $x^1=(2,2)^t$ avec la valeur de la fonction objectif $V(P_0)=4$.

Alors $x^1=(2,2)^t$ est une solution efficace pour le problème (P) avec le vecteur valeurs des fonctions objectifs égal à $(-2,4)^t$.

Le prochain problème (PLD) à résoudre est :

$$(P_1)\begin{cases} max\, x_1+x_2 \\ sc\, x_1-x_2\le 0 \\ x_1-2x_2\ge -y_1^1-4\left(1-y_1^1\right) \\ -x_1+3x_2\ge 5\, y_2^1-2\left(1-y_2^1\right) \\ y_1^1+y_2^1\ge 1 \\ x_1, x_2\in\{0,1,2\} \\ y_1^1, y_2^1\in\{0,1\} \end{cases}$$

Ce nouveau problème a plus de contraintes que (P_0) .

Dans (P_1) tous les points qui sont dominés par $(-2,4)^t$ sont supprimés de la région réalisable

La solution optimale pour (P_1) est $x_1=2$, $x_2=1$, $y_1^1=1$, $y_2^1=0$ avec la valeur de la fonction objectif est $V(P_1)=3$.

Alors $x^2=(2,1)^t$ est une solution efficace pour (P) avec le vecteur valeurs des fonctions objectifs égal à $(0,1)^t$.

Le prochain problème à résoudre est :

$$(P_2)\begin{cases} max\, x_1+x_2 \\ sc\, x_1-2x_2\le 0 \\ x_1-2x_2\ge -y_1^1-4\left(1-y_1^1\right) \\ -x_1+3x_2\ge 5\, y_2^1-2\left(1-y_2^1\right) \\ x_1-2x_2\ge y_1^2-4\left(1-y_1^2\right) \\ -x_1+3x_2\ge 2\, y_2^2-2\left(1-y_2^2\right) \\ y_1^1+y_2^1\ge 1 \\ y_1^2\ge y_2^2\ge 1 \\ x_1, x_2\in\{0,1,2\} \\ y_1^1, y_1^2, y_2^1, y_2^2\in\{0,1\} \end{cases}$$

La solution optimale pour (P_2) est $x_1=1$, $x_2=2$, $y_1^1=y_1^2=0$, $y_2^1=y_2^2=1$ avec $V(P_2)=3$. Ceci correspond à un nouveau point efficace $x^3=(1,2)^t$ avec le vecteur valeurs des fonctions objectifs égal à $(-3,5)^t$.

Le nouveau problème (P_3) est défini :

$$
\begin{aligned}
&¿\\
&max\, x_1+x_2\\
&sc\, x_1-2x_2\le 0\\
&x_1-2x_2\ge -y_1^1-4(1-y_1^1)\\
&-x_1+3x_2\ge 5\,y_2^1-2(1-y_2^1)\\
&x_1-2x_2\ge y_1^2-4(1-y_1^2)\\
&-x_1+3x_2\ge 2\,y_2^2-2(1-y_2^2)\\
&x_1-2x_2\ge -2\,y_1^3-4(1-y_1^3)\\
&-x_1+3x_2\ge 6\,y_2^3-2(1-y_2^3)\\
&y_1^s+y_2^s\ge 1\; s=1,2,3\\
&x_1,x_2\in\{0,1,2\}\\
&y_k^s\in\{0,1\}\; k=1,2\, s=1,2,3\\
&¿\\
&(P_3)¿
\end{aligned}
$$

La solution optimale pour (P_3) est $x_1=0$, $x_2=2$, $y_1^1=y_1^2=y_1^3=0$, $y_2^1=y_2^2=y_2^3=1$, avec $V(P_3)=2.$

Ceci correspond à une nouvelle solution efficace $x_4=(0,2)^t$ avec le vecteur valeurs des fonctions objectifs égal à $(-4,6)^t$.

Le nouveau problème (P_4) est défini :

$$
\begin{aligned}
&max\, x_1+x_2\\
&sc\, x_1-2x_2\le 0\\
&x_1-2x_2\ge -y_1^1-4\left(1-y_1^1\right)\\
&-x_1+3x_2\ge 5\,y_2^1-2\left(1-y_2^1\right)\\
&x_1-2x_2\ge y_1^2-4\left(1-y_1^2\right)\\
&-x_1+3x_2\ge 2\,y_2^2-2\left(1-y_2^2\right)\\
(P_4)\quad &x_1-2x_2\ge -2\,y_1^3-4\left(1-y_1^3\right)\\
&-x_1+3x_2\ge 6\,y_2^3-2\left(1-y_2^3\right)\\
&x_1-2x_2\ge -3\,y_1^4-4\left(1-y_1^4\right)\\
&-x_1+3x_2\ge 7\,y_2^4-2\left(1-y_2^4\right)\\
&y_1^s+y_2^s\ge 1\, s=1,2,3,4\\
&x_1, x_2\in\{0,1,2\}\\
&y_k^s\in\{0,1\}\, k=1,2\, s=1,2,3,4
\end{aligned}
$$

La solution optimale est $x_1=x_2=1$, $y_1^1=y_1^3=y_1^4=1\; y_2^1=0,$ $y_2^1=y_2^3=y_2^4=0,$

$y_2^2=1$ avec $V(P_4)=2.$

Ceci indique un nouveau point efficace $x_4=(1,1)^t$ avec le vecteur valeurs des fonctions objectifs égal à $(-1,2)^t$.

Le prochain problème est

$$
\begin{cases}
max\, x_1 + x_2 \\
sc\, x_1 - 2x_2 \le 0 \\
x_1 - 2x_2 \ge -y_1^1 - 4\left(1 - y_1^1\right) \\
-x_1 + 3x_2 \ge 5\, y_2^1 - 2\left(1 - y_2^1\right) \\
x_1 - 2x_2 \ge y_1^2 - 4\left(1 - y_1^2\right) \\
-x_1 + 3x_2 \ge 2\, y_2^2 - 2\left(1 - y_2^2\right) \\
x_1 - 2x_2 \ge -2\, y_1^3 - 4\left(1 - y_1^3\right) \\
-x_1 + 3x_2 \ge 6\, y_2^3 - 2\left(1 - y_2^3\right) \\
x_1 - 2x_2 \ge -3\, y_1^4 - 4\left(1 - y_1^4\right) \\
-x_1 + 3x_2 \ge 7\, y_2^4 - 2\left(1 - y_2^4\right) \\
x_1 - 2x_2 \ge -4\left(1 - y_1^5\right) \\
-x_1 + 3x_2 \ge 3\, y_2^5 - 2\left(1 - y_2^5\right) \\
y_1^s + y_2^s \ge 1 \; s = 1,2,3,4,5 \\
x_1, x_2 \in \{0,1,2\} \\
y_k^s \in \{0,1\} \; k = 1,2 \; s = 1,2,3,4,5
\end{cases}
\leftarrow (P_5)
$$

(P_5) est irréalisable, la procédure est terminée. Toutes les solutions efficaces pour le problème (P) sont déterminées.

Conclusion:

L'inconvénient principal de cette méthode est qu'à chaque itération on ajoute plusieurs contraintes et de variables en même temps, ce qui implique que la taille du problème augmente d'une itération à une autre.

4.2. Méthode 2 : technique de la découpe (Abbas et Moulai [7])

Avant de décrire la deuxième procédure qui nous permet de résoudre $(PLMD)$ on va donner quelques notations et théorèmes :

Considérons le problème relaxé $P(¿¿1)¿$ suivant :

$$(P_1)\begin{cases} max\, Z_1(x)=c^1 x \\ sc\, x \in S \\ x \in N^n \end{cases}$$

Nous fixons maintenant les notations suivantes :

$$S_1=\{¿¿1\, x \le b_1, A_1 \in R^{m_1 * n_1}, b_1 \in R^{m_1}\}^{x \in N^{n_1}\{A}$$

S_1 est la région tronquée de S après la résolution de (P_1) .

$x_1^1 = (x_{1,j}^1)$ est la solution entière optimale de (P_1) qui donne Z_1^1

$Z_i^1=¿$ Valeur de Z_i , $i \in \{2,\dots r\}$ correspond à la solution x_1^1 .

A_{B1}^1 est la matrice de base de S_1 .

$a_{1j}^1 \in R^{m_1 * 1}$ est le j^{eme} vecteur de la matrice hors base pour x_1^1 .

$y_{1j}^1=(A_{B1}^1)^{-1} a_{1j}^1$ où $y_{1,j}^1 \in R^{m_1 * 1}$. (2.3)

$I_1=\{¿¿1\, j^1 \in A_{B1}^1\}^{i\{a}$ (Indices de base)

$N_1=\{j \;\; a_{1j}^1 \notin A_{B1}^1\}$ (Indices hors base)

$$z_{1,j}^1=\sum_{i \in I_1} c_i^1 y_{1ij}^1 \quad .$$ (2.4)

$$Z_i(x_1^1)=c^i x_1^1 \quad .$$ (2.5)

$$\Gamma_1=\{j \in N_1 \text{ et } z_{1j}^1-c_j^1=0\}$$

Pour $k\geq 2$

$$\}_k x\leq b_k, A_k\in R^{m_k*n_k}, b_k\in R^{m_k}$$
$$x\in N^{n_k}$$
$$S_k=$$

S_k est la région tronquée obtenu après l'application de la coupe $\sum_{j\in N_{k-1}\{j_{k-1}} x_j\geq 1$ où

$j_{k-1}\in\Gamma_{k-1}$.

$x_k^1=(x_{k,j}^1)$ est la k^{eme} solution optimale de problème (P_1) sur S_k à l'étape k .

x_k^q est une solution entière alternée a x_k^1 s'il existe , $q\in N$.

A_{Bk}^1 est la matrice de base de S_k .

$a_{kj}^1\in R^{m_k*1}$ j^{eme} vecteur de la matrice hors base pour x_k^1 à la région S_k .

$$y_{kj}^1=(A_{Bk}^1)^{-1}*a_{kj}^1 \quad \text{où} \quad y_{k,j}^1\in R^{m_k*1} \qquad (2.6)$$

$$I_k=\{i \in a \mid j^1\in A_{Bk}^1\} \quad \text{(Indices de base)}$$

$$N_k=\{j \in a \mid j^1\notin A_{Bk}^1\} \quad \text{(Indices hors base)}$$

$$z_k^1=\sum_{i\in I_k} c_i^1 y_{k,ij}^1 \qquad (2.7)$$

$$Z_i\left(x_k^1\right)=c^i x_k^1 \qquad (2.8)$$

$$\Gamma_k=\left\{j \in N_k \text{ et } z_{k,j}^1-c_j^1=0\right\} \quad .$$

Définition 2.2 :

Une solution x_k^q alternée a x_k^1 est définie comme suit :

$$x_{k,i}^q=x_{k,i}^1-\Phi_{j_k} y_{k,ij_k}^1, i \in I_k$$

$$x_{k,j_k}^q=\Phi_{j_k}$$

$$x_{k,v}^q=0 \text{ pour tout } v \in N_k\{j_k$$

$$x_k^q=[$$

Où $\frac{x_{k,i}^1}{y_{k,ij_k}^1}\{y$

$$0 \le \Phi_{j_k} \le \min_{i \in I_k}[k, i\, j_k{}^1>0]$$

Φ_{j_k} est le pas à l'itération k .

Théorème 2.2 [10] : toutes les solutions entières réalisables x_k^q , $q>2 et\, q \in N$ du problème (P_1) alternée à x_k^1 de la région S (ou bien la région tronquée S_k) dans la direction d'un vecteur $a_{k,j_k}^1, j_k \in \Gamma_k$, existent dans un demi espace ouvert

$$\sum_{j \in N_k\{j_k} x_j<1. \qquad (2.9)$$

Preuve :

Puisque x_k^1 est une solution réalisable du problème (P_1), alors $A\,x_k^1 \le b$ où $x_k^1=(x_{k,j}^1)$.

On a : $A\,x_k^1=\sum_{i\in I_k} a_{k,i}^1 x_{k,i}^1 \le b$

Pour un $j_k \in \Gamma_k$

$$\sum_{i\in I_k} a_{k,i}^1 x_{k,i}^1 - \Phi_{j_k} a_{k,j_k}^1 + \Phi_{j_k} a_{k,j_k}^1 \le b \quad (¿)$$

Puisque $a_{k,j_k}^1=\sum_{i\in I_k} a_{k,i}^1 y_{k,ij_k}^1$ on remplace dans $(¿)$ on aura

$$\sum_{i\in I_k} a_{k,i}^1 x_{k,i}^1 - \Phi_{j_k}\left(\sum_{i\in I_k} a_{k,i}^1 y_{k,ij_k}^1\right) + \Phi_{j_k} a_{k,j_k}^1 \le b$$

$$\Rightarrow \sum_{i\in I_k} a_{k,i}^1 \left(x_{k,i}^1 - \Phi_{j_k} y_{k,ij_k}^1\right) + \Phi_{j_k} a_{k,j_k}^1 \le b$$

Calculons $Z_1(x_k^q)$ en utilisant la définition de x_k^q

$$Z_1(x_k^q)=c^1 x_k^q=\sum_{i\in I_k} c_i^1 x_{k,i}^q + c_{j_k}^1 x_{k,j_k}^q + \sum_{\substack{v\in N_k\{j_k \\ ¿}} ¿\, c_v^1 x_{k,v}^q\, ¿$$

$$c_i^1\left(x_{k,i}^1 - \Phi_{j_k} y_{k,ij_k}^1\right) + ¿\, c_{j_k}^1 x_{k,j_k}^q$$
$$¿ \sum_{i\in I_k} ¿$$

$$c_i^1 x_{k,i}^1 - ¿\,\Phi_{j_k} \sum_{i\in I_k} c_i^1 y_{k,ij_k}^1 + c_{j_k}^1 \Phi_{j_k}$$
$$¿ \sum_{i\in I_k} ¿$$

$$c_i^1 x_{k,i}^1 - ¿\, \Phi_{j_k}\left(\sum_{i \in I_k} c_i^1 y_{k,i j_k}^1 - c_{j_k}^1\right)$$

$$¿ \sum_{i \in I_k} ¿$$

$$c_i^1 x_{k,i}^1 - ¿\, \Phi_{j_k}\left(z_{k,j_k}^1 - c_{j_k}^1\right)$$

$$¿ \sum_{i \in I_k} ¿$$

$$¿\, Z_1\left(x_k^1\right) - \Phi_{j_k}\left(z_{k,j_k}^1 - c_{j_k}^1\right)$$

Puisque $j_k \in \Gamma_k,$ alors $z_{k,j_k}^1 - c_{j_k}^1 = 0,$ ce qui implique $Z_1\left(x_k^q\right) = Z_1\left(x_k^1\right)$

Donc x_k^q est une solution réalisable de problème $\left(P_1\right)$ alternée à x_k^1 .

Puisque $x_{k,j}^q = 0$ pour tout $j \in N_k\{j_k ¿¿$, on peut voir facilement que $\sum_{¿}^{j \in N_k\{j_k} ¿¿ x_{k,j}^q < 1.$

Alors le point x_k^q existe dans un demi-espace ouvert $\sum_{¿}^{j \in N_k\{j_k} ¿¿ x_j < 1.$

Remarque : toutes les solutions entières réalisables x_k^q alternée à x_k^1 , sont obtenues en prenant en compte toutes les valeurs entières possibles de Φ_{j_k} .

Théorème 2.3 [10] : une solution entière réalisable du problème $\left(P_1\right)$ qui n'est pas sur l'arête E_{j_k} $j_k \in \Gamma_k$ de la région tronquée S_k (ou la région S) à travers un point entier réalisable x_k^1 du problème $\left(P_1\right)$ existe dans un demi-espace fermé

$$\sum_{j \in N_k \setminus \{j_k\}} x_j \geq 1. \quad (2.10)$$

Développement de la procédure :

Cette procédure est basée sur la technique de découpage.

Etape 1 : résoudre le problème (P_1) et trouver la solution entière optimale x_1^1 dans S_1 .

Construire Γ_1 .

Etape 2 : tests

1) Si $\Gamma_1 = \varnothing$, x_1^1 est l'unique solution optimale du problème (P_1) et Z_1^1 est la valeur optimale de Z_1 , calculer les valeurs Z_i^1 de Z_i donnée par x_1^1 , $i \in \{2,\dots r\}$

Garder le r-uplet $(Z_1^1, Z_2^1, \dots Z_r^1)$ pour créer un ensemble de point efficace, on le note Eff_0 .

Tronquer le point x_1^1 par la coupe suivante :

$$\sum_{j \in N_1} x_j \geq 1$$

Par l'application de la méthode duale du simplexe et les coupes successives de Gomory, si c'est nécessaire, nous obtenons une solution entière réalisable $x_2^1 = (x_{2,j}^1)$ dans la région tronquée S_2 .

Calculer le r-uplet $\left(Z_1^1, Z_2^1, \ldots Z_r^1\right)$ correspondant et l'ajouter à l'ensemble Eff_0 s'il n'est pas dominé par le r-uplet identifié précédemment, alors Eff_0 devient Eff_1 .

2) Si $\Gamma_1 \neq \varnothing$, choisissez un $j_1 \in \Gamma_1$ et calculer le rapport minimum

$$\Phi = \min_{i \in I_1} \left\{ \frac{x_{1,i}^1}{y_{1,i\,j_1}^1} \,/\, y_{1,i\,j_1}^1 > 0 \right\}$$ correspondant à la solution x_1^1 .

Calculer $\left(Z_1^1, Z_2^1, \ldots Z_r^1\right)$.

a. Si $\Phi \geq 1$, calculer toutes les solutions réalisables x_1^l , $l \geq 2 \; et \; l \in N$, alternée à x_1^1 sur l'arête E_{j_1} . Chaque solution donne naissance à un nouveau r-uplet de la forme $\left(Z_1, \ldots Z_r\right)$, actualiser l'ensemble des solutions efficaces. Tronquer l'arête E_{j_1} par la coupe suivante :

$$\sum_{j \in N_1 \setminus \{j_1\}} x_j \geq 1$$

Par l'application de la méthode duale du simplexe et les coupes successives de Gomory si c'est nécessaire, nous obtenons une solution réalisable $x_2^1 = \left(x_{2,j}^1\right)$ dans la région tronquée S_2.

Calculer $(Z_1, \dots Z_r)$ et l'ajouter à Eff_0 s'il n'est pas dominé par le r-uplet identifié précédemment, alors Eff_0 devient Eff_1 .

b. Si $\Phi<1$ pour tous $j_1 \in \Gamma_1$, choisissez un $j_1 \in \Gamma_1$ et appliquer la coupe suivante :

$$\sum_{j \in N_1 \setminus j_1} x_j \geq 1$$

Par l'application de la méthode duale du simplexe et les coupes successives de Gomory si c'est nécessaire, nous obtenons une solution réalisable $x_2^1 = (x_{2,j}^1)$ dans la région tronquée S_2.

Calculer $(Z_1, \dots Z_r)$ et l'ajouter à Eff_0 s'il n'est pas dominé par le r-uplet identifié précédemment, alors Eff_0 devient Eff_1 .

Etape 3 : choisissez $j_2 \in \Gamma_2$ et calculer toutes les solutions réalisables $x_2^l, \quad l \geq 2$ $et\ l \in N$ alternée à x_2^1 s'ils existent sur l'arête E_{j_2} , calculer $(Z_1, \dots Z_r)$, et l'ajouter à Eff_1 s'il n'est pas dominé par les r-uplets identifiés précédemment, alors Eff_1 devient Eff_2 .

Tronquer l'arête E_{j_2} par la coupe

$$\sum_{j \in N_2 \setminus j_2} x_j \geq 1$$

et chercher la solution entière optimale dans la région tronquée pour appliquer l'étape 4 de la procédure.

Etape k : choisissez $j_{k-1} \in \Gamma_{k-1}$ et chercher les solutions réalisables alternées à x_{k-1}^{l},

$l \geq 2 et l \in N$ sur l'arête $E_{j_{k-1}}$. Calculer de nouveau les r-uplets correspondants.

Incrémenter l'ensemble Eff_{k-2} avec les r-uplets non dominés pour obtenir l'ensemble Eff_{k-1}.

L'arête $E_{j_{k-1}}$ est tronquée par la coupe

$$\sum_{j \in N_{k-1} \setminus \{ j_{k-1} \}} x_j \geq 1$$

Après l'application de l'algorithme duale du simplexe et les coupes successives de Gomory, nous obtenons une solution réalisable optimale x_k^1 sur la région tronquée S_k.

Ceci marque le commencement de l'étape k+1.

Etape terminal : la procédure se termine lorsqu'on peut plus pivoter, ce qui implique que la région réalisable courante ne contient aucun point entier réalisable et tous les points efficaces ont été trouvés.

Théorème 2.4 [7]: la procédure trouve toutes les solutions efficaces du problème (P) et converge en un nombre fini d'étapes.

Preuve : puisque la région réalisable est bornée et tronquée à chaque étape par les applications répétées de la coupe (2.10) suivi de l'application des coupes de Gomory, l'ensemble S est totalement balayée de telle sorte que les points entiers déjà déterminés ne puissent pas réapparaitre, menant ainsi à la convergence de la procédure en un nombre fini d'étapes.

Exemple numérique : pour illustrer l'application de l'algorithme, considérons le problème suivant :

$$\begin{cases} max\, Z_1(x) = c^1 x = x_1 + 2x_2 \\ max\, Z_2(x) = c^2 x = 3x_1 - 2x_2 \\ max\, Z_3(x) = c^3 x = -x_1 + 2x_2 \\ sc \\ x_1 + x_2 \le 7 \\ 2x_1 \le 11 \\ 2x_2 \le 7 \\ x_1, x_2 \in N \end{cases}$$

Etape 1 : résoudre le problème (P_1) , la solution optimale de la région tronquée S_1 est donnée par le tableau 1 :

$c_B^3 c_B^2 c_B^1$			B	x_B	x_1	x_2	x_3	x_4	x_5	x_6
-1	3	1	x_1	4	1	0	1	0	0	-1
0	0	0			0	0	-2	1	0	2
2	-2	2	x_4		0	1	0	0	0	1
0	0	0	3		0	0	0	0	1	-2
			x_2							

	3 x_5 1						
$Z_1=10$	$z_j^1-c_j^1$	0	0	1	0	0	1
$Z_2=6$	$z_j^2-c_j^2$	0	0	3	0	0	-5
$Z_3=2$	$z_j^3-c_j^3$	0	0	-1	0	0	3

Tableau 1

Où x_3, x_4, x_5 sont des variables d'écart et x_6 est la variable de la coupe de Gomory.

$I_1=\{1,2,4,5\}$, $N_1=\{3,6\}$

¿

$\Gamma_1=\{j\ N_1 et\ Z_{1,j}^1-c_j^1=0\}=\varnothing$

$(Z_1,Z_2,Z_3)=(10,6,2)$, $x_1^1=(4,3)$

Le premier 3-uplet efficace est $(10,6,2)$ et $Eff_0=\{(10,6,2)\}$.

Etape 2 :

$\Gamma_1=\varnothing$, x_1^1 est unique solution optimale. Cette solution est tronquée par la coupe (2.10)

c.-à-d $x_3+x_6\geq 1$ ou $-x_3-x_6+x_7=-1$.

L'ajout de cette coupe au tableau 1, nous donne le tableau 1.1.

$c_B^3 c_B^2 c_B^1$			$B\ x_B$	$x_1 x_2 x_3 x_4 x_5 x_6$					x_7	
-1	3	1	x_1 4	1	0	1	0	0	-1	0
0	0	0		0	0	-2	1	0	2	0
2	-2	2	x_4	0	1	0	0	0	1	0
0	0	0	3	0	0	0	0	1	-2	0
0	0	0		0	0	-1	0	0	-1	1

	x_2 3 x_5 1 x_7 -1	
$Z_1=10$	$z_j^1-c_j^1$	001001 0

Tableau1.1

Appliquons l'algorithme dual du simplexe sur le tableau 1.1, nous obtenons une solution $x_2^1=(3,3)$ donnée par le tableau 2.

$c_B^3 c_B^2 c_B^1$	$B\, x_B$	$x_1 x_2 x_3 x_4 x_5 x_6 \quad x_7$
-1 3 1	x_1 3	1 0 0 0 0 -2 1
0 0 0		0 0 0 1 0 4 -2
2 -2 2	x_4	0 1 0 0 0 1 0
0 0 0	5	0 0 0 0 1 -2 0
0 0 0	x_2	0 0 1 0 0 1 -1
	3	
	x_5 1	
	x_3 1	
$Z_1=9$	$z_j^1-c_j^1$	0000001
$Z_2=3$	$z_j^2-c_j^2$	00000−83
$Z_3=3$	$z_j^3-c_j^3$	000004−1

Tableau 2

Le 3-uplet correspondant $(9,3,3)$ est non dominé par le 3-uplet trouvé précédemment.

Alors en ajoutant $(9,3,3)$ à Eff_0 , nous obtenons $Eff_1=\{(10,6,2),(9,3,3)\}$.

$I_2=\{1,2,3,4,5\}$

$N_2=\{6,7\}$

$\Gamma_2=\{6\}$

Etape 3 :

$\Gamma_2\neq 0$, prenons $j_2=6$

$\Phi_{j_2}\leq min\left\{\frac{5}{4},\frac{3}{1},\frac{1}{1}\right\}=\frac{5}{4}$ Correspond à la variable x_3 . Il existe une seule solution $x_2^2=\left(x_{2,j}^2\right)$ alternée à x_2^1 .

Pour $\Phi_{j_2}=1$, $x_2^2=(5,2)$ et le 3-uplet efficace correspondant est $(9,11,-1)$.

On aura $Eff_2=Eff_1\{(9,11,-1)\}$

L'arête E_6 est tronquée par la coupe $x_7\geq 1$ ou $-x_7+x_8=-1$, en ajoutant cette contrainte au dernier tableau et en appliquant le dual du simplexe, on obtient le tableau ci dessous:

$c_B^3 c_B^2 c_B^1$			B	x_B	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8
-1	3	1	x_1	2	1	0	0	0	0	-2	0	1
0	0	0			0	0	0	1	0	4	0	-2
2	-2	2	x_4		0	1	0	0	0	1	0	0
0	0	0	7		0	0	0	0	1	-2	0	0
0	0	0	x_2		0	0	1	0	0	1	0	-1
0	0	0			0	0	0	0	0	0	1	-1
			3									
			x_5	1								
			x_3	2								

	x_7 1	
$Z_1=8$	$z_j^1-c_j^1$	00000001
$Z_2=0$	$z_j^2-c_j^2$	00000−803
$Z_3=4$	$z_j^3-c_j^3$	0000040−1

Tableau 3

$x_3^1=(2,3)$, le 3-uplet correspondant est $(8,0,4)$ qui n'est pas dominé avec les 3-uplets trouvés précédemment.

$Eff_2=\{(10,6,2),(9,3,3),(9,11,-1),(8,0,4)\}$

$I_3=\{1,2,3,4,5,7\}$

$N_3=\{6,8\}$

$\Gamma_3=\{6\}$

Etape 4 :

$\Gamma_3\neq 0$, prenons $j_3=6$

$\Phi_{j_3}\leq min\left\{\frac{7}{4},\frac{3}{1},\frac{2}{1}\right\}=\frac{7}{4}$, il existe une seule solution $x_3^2=x_{3,j}^2$ alternée à x_3^1 .

Pour $\Phi_{j_3}=1$, $x_3^2=(4,2)$, le 3-uplet efficace correspondent est $(8,8,0)$.

$Eff_3=Eff_2\{(8,8,0)\}$

L'arête E_6 est tronquée par la coupe $x_8\geq 1$ ou $-x_8+x_9=-1$

On ajoute cette contrainte au dernier tableau, et en appliquant le dual du simplexe, on obtient le tableau suivant :

c_B^3 c_B^2 c_B^1	B x_B	x_1 x_2 x_3 x_4 x_5 x_6 x_7 x_8 x_9								
-1 3 1	x_1	1	0	0	0	0	-2	0	0	1
0 0 0	1	0	0	0	1	0	4	0	0	-2
2 -2 2	x_4	0	1	0	0	0	1	0	0	0
0 0 0	9	0	0	0	0	1	-2	0	0	0
0 0 0	x_2	0	0	1	0	0	1	0	0	-1
0 0 0	3	0	0	0	0	0	0	1	0	-1
	x_5	0	0	0	0	0	0	0	1	-1
	1									
	x_3									
	3									
	x_7									
	2									
	x_8									
	1									
$Z_1=7$	$z_j^1-c_j^1$	0	0	0	0	0	0	0	0	1
$Z_2=-3$	$z_j^2-c_j^2$	0	0	0	0	0	-8	0	0	3
$Z_1=5$	$z_j^3-c_j^3$	0	0	0	0	0	4	0	0	-1

Tableau 4

La solution correspondante est $x_4^1=(1,3)$ et le 3-uplet correspondant est $(7,-3,5).$ Il n'est pas dominé par les 3-uplets trouvés précédemment.

$Eff_3=\{(10,6,2),(9,3,3),(9,11,-1),(8,0,4),(8,8,0),(7,-3,5)\}$.

$I_4=\{1,2,3,4,5,7,8\}$

$N_4=\{6,9\}$

$\Gamma_4=\{6\}$

Etape 5 :

$\Gamma_4\neq\varnothing$, prenons $j_4=6.$

$\Phi_{j_4}\leq min\left\{\frac{9}{4},\frac{3}{1},\frac{3}{1}\right\}=\frac{9}{4}$, il existe deux solutions réalisables x_4^2 et x_4^3 alternées à x_4^1 .

Pour $\Phi_{j_4}=1$, $x_4^2=(3,2)$ et le 3-uplet correspondant est $(7,5,1)$. Ce 3-uplet est dominé.

Pour $\Phi_{j_2}=2$, $x_4^3=2$, et le 3-uplet correspondant est $(7,13,-3)$. Ce 3-uplet n'est pas dominé, il est ajouté 0 Eff_3 pour avoir Eff_4 .

$Eff_4=\{(10,6,2),(9,3,3),(9,11,-1),(8,0,4),(8,8,0),(7,-3,5),(7,13,-3)\}$.

L'arête E_6 est tronquée par la coupe $x_9\geq 1$ ou $-x_9+x_{10}=-1$.

On ajoute cette coupe au dernier tableau et en appliquant le dual du simplexe, on obtient le tableau 5 :

c_B^3		c_B^2	B	x_1		x_2	x_3		x_4		x_5		x_6
c_B^1			x_B	x_7		x_8	x_9		x_{10}				
-1	3	1	x_1	1	0	0	0	0	-2	0	0	0	1
0	0	0		0	0	0	1	0	4	0	0	0	-2
2	-2	2	0	0	1	0	0	0	1	0	0	0	0
0	0	0	x_4	0	0	0	0	1	-2	0	0	0	0
0	0	0	11	0	0	1	0	0	1	0	0	0	-1

0 0 0	0 0 0	0 0 0	x_2 3 x_5 1 x_3 4 x_7 3 x_8 1 x_9 1	0 0 0	0 0 0	0 0 0	0 0 0	0 0 0	0 0 0	1 0 0	0 1 0	0 0 1	-1 -1 -1
$Z_1=6$			$z_j^1-c_j^1$	0	0	0	0	0	0	0	0	0	1
$Z_2=-6$			$z_j^2-c_j^2$	0	0	0	0	0	-8	0	0	0	3
$Z_1=6$			$z_j^3-c_j^3$	0	0	0	0	0	4	0	0	0	-1

Tableau 5

Le tableau nous donne la solution $x_5^1=(0,3)$ et le 3-uplet correspondant $(6,-6,6)$ est non dominée par les 3-uplet précédent.

$Eff_4=\{(10,6,2),(9,3,3),(9,11,-1),(8,0,4),(8,8,0),(7,-3,5),(7,13,-3),(6,-6,6)\}$.

$I_5=\{1,2,3,4,5,7,8,9\}$

$N_5=\{6,10\}$

$\Gamma_5=\{6\}$

Etape 6 :

$\Gamma_5 \neq \varnothing$, prenons $j_5 = 6$.

$\Phi_{j_5} \leq min\left\{\frac{11}{4}, \frac{3}{1}, \frac{4}{1}\right\} = \frac{11}{4}$, il existe deux solutions réalisable x_5^2 et x_5^3 alternées à x_5^1 .

Pour $\Phi_{j_5} = 1$, $x_5^2 = (2,2)$ et le 3-uplet correspondant $(6,2,2)$ est dominé.

Pour $\Phi_{j_5} = 2$, $x_5^3 = (4,1)$ et le 3-uplet correspondant $(6,-10,-2)$ est aussi dominé.

L'arête E_6 est tronquée par la coupe $x_{10} \geq 1$ ou $-x_{10} + x_{11} = -1$, on ajoute cette contrainte au dernier tableau et en appliquant la méthode dual du simplexe on obtient le tableau 6 suivant:

c_B^3		c_B^2	B	x_1		x_2		x_3		x_4	x_5
c_B^1			x_B	x_6			x_{11}	x_{12}			
-1	3	1	x_1	1	0	0	0	0	0	1	-2
0	0	0		0	0	0	1	0	0	-2	4
2	-2	2	0	0	1	0	0	0	0	0	0
0	0	0	x_4	0	0	0	0	1	0	0	-2
0	0	0	11	0	0	1	0	0	0	-1	1
0	0	0	x_2	0	0	0	0	0	1	0	-1
			3								
			x_5								
			1								
			x_3								
			4								
			x_6								
			3								

$Z_1=6$	$z_j^1-c_j^1$	0	0	0	0	0	0	0	1	0
$Z_2=-6$	$z_j^2-c_j^2$	0	0	0	0	0	0	0	3	- 8
$Z_1=6$	$z_j^3-c_j^3$	0	0	0	0	0	0	0	-1	4

Tableau 6

x_{12} est la variable de la coupe de Gomory, car le dual du simplexe nous a donné une solution réalisable qui n'est pas entière.

Dans le tableau 6 la solution correspondante est $x_6^1=(1,2)$, le 3-uplet correspondant $(5,-1,3)$ est dominé par les 3-uplets déjà trouvés.

Donc $Eff_5=\{(10,6,2),(9,3,3),(9,11,-1),(8,0,4),(8,8,0),(7,-3,5),(7,13,-3),(6,-6,6)\}$

$I_6=\{1,\dots 10\}$

$N_6=\{11,12\}$

$\Gamma_6=\{12\}$

Etape 7 :

$\Gamma_6\neq 0$, prenons $j_6=12$, $\Phi_{j_6}\leq min\left\{\frac{9}{4},\frac{2}{1},\frac{4}{1}\right\}=2$, il existe deux solutions réalisables x_6^2 et x_6^3 alternées à x_6^1 .

Pour $\Phi_{j_6}=1$, $x_6^2=(3,1)$, le 3-uplet correspondant est $(5,7,-1)$, ce 3-uplet est dominé.

Pour $\Phi_{j_6}=2$ $x_6^3=(5,3)$, le 3-uplet correspondant est $(5,15,-5)$, ce 3-uplet est non dominé.

$$Eff_5=\left\{\begin{matrix}(10,6,2),(9,3,3),(9,11,-1),(8,0,4),(8,8,0),(7,-3,5),(7,13,-3),(6,-6,6)\\ ,(5,15,-5)\end{matrix}\right\} .$$

L'arête E_{12} est tronquée par la coupe $x_{11}\geq 1$ ou $-x_{11}+x_{13}=-1$

On ajoute cette contrainte au dernier tableau, et on appliquant le dual du simplexe, on obtient le tableau 7 suivant :

c_B^3		c_B^2	B	x_1		x_2		x_3		x_4		x_5
c_B^1			x_B	x_6			x_{12}		x_{12}			
-1	3	1	x_1	1	0	0	0	0	0	-2	1
0	0	0		0	0	0	1	0	0	4	-2
2	-2	2	0	0	1	0	0	0	0	1	0
0	0	0	x_4	0	0	0	0	1	0	-2	0
0	0	0	11	0	0	1	0	0	0	1	-1
0	0	0	x_2	0	0	0	0	0	1	-1	0
			2								
			x_5								
			3								
			x_3								
			5								
			x_6								
			1								
$Z_1=4$			$z_j^1-c_j^1$	0	0	0	0	0	0	0	0	1
$Z_2=-4$			$z_j^2-c_j^2$	0	0	0	0	0	0	0	-8	3
$Z_1=4$			$z_j^3-c_j^3$	0	0	0	0	0	0	0	4	-1

Tableau 7

La solution réalisable est $x_7^1=(0,2)$, et le 3-uplet correspondant est $(4,-4,4)$ est dominé par les 3-uplets trouvés précédemment.

Donc $Eff_6=\left\{\begin{matrix}(10,6,2),(9,3,3),(9,11,-1),(8,0,4),(8,8,0),(7,-3,5),(7,13,-3),(6,-6,6)\\ ,(5,15,-5)\end{matrix}\right\}$

$I_7=\{1,\dots 11\}$, $N_7=\{12,13\}$, $\Gamma_7=\{12\}$.

Etape 8 :

$\Gamma_7\neq 0$, prenons $j_7=12.$

$\Phi_{j_7}\leq min\left\{\frac{11}{4},\frac{2}{1},\frac{5}{1}\right\}=2$, il existe deux solutions réalisables x_7^1 et x_7^2 alternées à x_7^1 .

Pour $\Phi_{j_7}=1$, $x_7^2=(2,1)$ et le 3-uplet correspondant est $(4,4,0)$ est dominé par les 3-uplets efficaces trouvés précédemment.

Pour $\Phi_{j_7}=2$, $x_7^2=(4,0)$ et le 3-uplet correspondant est $(4,12,-4)$ est aussi dominé.

l'arête E_{12} est tronquée par la coupe $x_{13}\geq 1$ ou $-x_{13}+x_{14}=-1$

On ajoute cette contrainte au dernier tableau et on applique le dual du simplexe. On aura le tableau suivant :

c_B^3		c_B^2	B	x_1		x_2		x_3		x_4	x_5
c_B^1			x_B	x_6		x_{14}		x_{15}			
-1	3	1	x_1	1	0	0	0	0	0	1	-2
0	0	0		0	0	0	1	0	0	-2	4
2	-2	2	1	0	1	0	0	0	0	0	1

0 0 0	0 0 0	0 0 0	x_4 9 x_2 1 x_5 5 x_3 5 x_6 2	0 0 0	0 0 0	0 1 0	0 0 0	1 0 0	0 0 1		0 -1 0	-2 1 -1
$Z_1=3$			$z_j^1-c_j^1$	0	0	0	0	0	0	0	1	0
$Z_2=1$			$z_j^2-c_j^2$	0	0	0	0	0	0	0	3	-8
$Z_1=1$			$z_j^3-c_j^3$	0	0	0	0	0	0	0	-1	4

Tableau 8

x_{15} est la variable de la coupe de Gomory, car le dual du simplexe nous a donné une solution réalisable qui n'est pas entière.

Dans le tableau 7 la solution correspondante est $x_8^1=(1,2)$, le 3-uplet correspondant est $(3,1,1)$ est dominé par les 3-uplets trouvés précédemment.

Donc $Eff_7=Eff_6$.

$I_8=(1,\dots 13)$, $N_8=(14,15)$, $\Gamma_8=(15)$.

Etape 9 :

$\Gamma_8\neq\varnothing$, prenons $j_8=15$,

$\Phi_{j_9} \leq min\left\{\frac{9}{4}, \frac{1}{1}, \frac{5}{1}\right\} = 1$, il existe une seule solution réalisable x_8^2 alterné à x_8^1 .

Pour $\Phi_{j8} = 1$, $x_8^2 = (3,0)$ et le 3-uplet correspondant est $(3,9,-3)$ est dominé.

L'arête E_{15} est tronquée par la coupe $x_{14} \geq 1$ ou $-x_{14} + x_{16} = 1$.

On ajoute cette contrainte au dernier tableau et on applique le dual du simplexe.

c_B^3 c_B^2 c_B^1	B x_B	x_1 x_6	x_2 x_{15}	x_3 x_{16}	x_4	x_5
-1 3 1	x_1	1 0 0 0 0 0			-2	1
0 0 0	0	0 0 0 1 0 0			4	-2
2 -2 2	x_4	0 1 0 0 0 0			1	0
0 0 0	11	0 0 0 0 1 0			-2	0
0 0 0	x_2	0 0 1 0 0 0			1	-1
0 0 0	1	0 0 0 0 0 1			-1	0
	x_5					
	5					
	x_3					
	6					
	x_6					
	2					
$Z_1 = 2$	$z_j^1 - c_j^1$	0 0 0 0 0 0		0	0	1
$Z_2 = -2$	$z_j^2 - c_j^2$	0 0 0 0 0 0		0	-8	3
$Z_1 = 2$	$z_j^3 - c_j^3$	0 0 0 0 0 0		0	4	-1

Tableau 9

La solution réalisable est $x_9^1=(0,1)$, et le 3-uplet correspondant est $(2,-2,2)$ est dominé.

Donc $Eff_8=Eff_7=Eff_6$.

$I_9=\{1,\dots 14\}$, $N_9=\{15,16\}$, $\Gamma_9=\{15\}$.

Etape 10 :

$\Gamma_{10}\neq 0$, prenons $j_9=15$.

$\Phi_{j_9}\leq min\left\{\frac{11}{4},\frac{1}{1},\frac{6}{1}\right\}=1$, il ya une seule solution réalisable x_9^2 alterné à x_9^1 .

Pour $\Phi_{J_9}=1$, $x_9^2=(2,0)$, le 3-uplet correspondant est $(2,6,-2)$ qui dominé.

L'arête E_{15} est tronquée par la coupe $x_{16}\geq 1$ ou $-x_{16}+x_{17}=-1$

On ajoute cette contrainte au dernier tableau, et en appliquant le dual du simplexe on aura le tableau suivant :

c_B^3		c_B^2	B	x_1		x_2		x_3		x_4	x_5
c_B^1			x_B	x_6			x_{17}	x_{18}			
-1	3	1	x_1	1	0	0	0	0	0	1	-2
0	0	0		0	0	0	1	0	0	-2	4
2	-2	2	0	0	1	0	0	0	0	0	1
0	0	0	x_4	0	0	0	0	1	0	0	-2
0	0	0	11	0	0	1	0	0	0	-1	1
0	0	0	x_2	0	0	0	0	0	1	0	-1
			1								
			x_5								
			5								

	x_3 6 x_6 2								
$Z_1=1$	$z_j^1-c_j^1$	0	0	0	0	0	0	0	1	0
$Z_2=3$	$z_j^2-c_j^2$	0	0	0	0	0	0	0	3	-8
$Z_1=-1$	$z_j^3-c_j^3$	0	0	0	0	0	0	0	-1	4

Tableau 10

x_{18} est la variable de la coupe de Gomory, qu'on a ajouté après l'application du dual simplexe.

La solution donnée par le tableau 10 est $x_{10}^1=(1,0)$, et le 3-uplet correspondant est $(1,3,-1)$ est dominé avec les 3-uplets trouvés précédemment.

Donc $Eff_9=Eff_8=Eff_6$.

$I_{10}=\{1,\dots 16\}$, $N_{10}=\{17,18\}$, $\Gamma_{10}=\{18\}$.

Etape 11 :

$\Gamma_{10}\neq 0$, prenons $j_{10}=18$

$\Phi_{j_{10}}\leq min\left\{\frac{9}{4},\frac{0}{1},\frac{6}{1}\right\}=0$, alors il n'existe aucune solution réalisable alterné à x_{10}^1 .

L'arête E_{18} est tronquée par la coupe $x_{17}\geq 1$ ou $-x_{17}+x_{19}=-1$

En ajoutant cette contrainte au dernier tableau et en appliquant le dual du simplexe on aura le tableau suivant :

c_B^3		c_B^2	B	x_1		x_2		x_3		x_4		x_5
c_B^1			x_B	x_6		x_{18}		x_{19}				
-1	3	1	x_1	1	0	0	0	0	0	-2	1
0	0	0		0	0	0	1	0	0	4	-2
2	-2	2	0	0	1	0	0	0	0	1	0
0	0	0	x_4	0	0	0	0	1	0	-2	0
0	0	0	11	0	0	1	0	0	0	1	-1
0	0	0	x_2	0	0	0	0	0	1	-1	0
			0								
			x_5								
			7								
			x_3								
			7								
			x_6								
			3								
$Z_1=0$			$z_j^1-c_j^1$	0	0	0	0	0	0	0	0	1
$Z_2=0$			$z_j^2-c_j^2$	0	0	0	0	0	0	0	-8	3
$Z_1=0$			$z_j^3-c_j^3$	0	0	0	0	0	0	0	4	-1

Tableau 11

$x_{11}^1=(0,0)$, le 3-uplet correspondant est $(0,0,0)$ est dominé par les 3-uplets déjà trouvés .

Donc $Eff_{10}=Eff_9=Eff_6$.

$I_{11}=(1,\dots 17)$, $N_{11}=(18,19)$, $\Gamma_{11}=(18)$.

Etape 12 :

$\Gamma_{11} \neq 0$, prenons $j_{11} = 18$

$\Phi_{j_{11}} \leq min\left\{\frac{11}{4}, \frac{0}{1}, \frac{7}{1}\right\} = 0$, alors il n'existe aucune solution réalisable alterné à x_{11}^1 .

L'arête E_{18} est tronquée par la coupe $x_{19} \geq 1$ ou $-x_{19} + x_{20} = -1$,

En ajoutant cette contrainte au dernier tableau, et en appliquant le dual du simplexe et la coupe de Gomory on aura le tableau 12

c_B^3 c_B^2 c_B^1	B x_B	x_1 x_2 x_3 x_4 x_5 x_6 x_{20} x_{21}							
0 0 0	x_6	-0.5	0	0	0	0	1	-0.5	0
0 0 0		2	0	0	1	0	0	0	0
2 -2 2	3.5	0.5	1	0	0	0	0	0.5	0
0 0 0	x_4	-1	0	0	0	1	0	-1	0
0 0 0	11	0.5	0	1	0	0	0	-0.5	0
0 0 0	x_2	-0.5	0	0	0	0	1	-0.5	1
	-0.5								
	x_5								
	8								
	x_3								

	7.5 x_{21} -0.5									
$Z_1=-1$	$z_j^1-c_j^1$	0	0	0	0	0	0	0	1	0
$Z_2=1$	$z_j^2-c_j^2$	0	0	0	0	0	0	0	-1	0
$Z_1=-1$	$z_j^3-c_j^3$	0	0	0	0	0	0	0	1	0

Tableau 12

Sur ce tableau la variable x_2 est sélectionnée pour quitter la base et la variable x_1 ou bien x_{20} sont candidates à enter dans la base mais le vecteur ligne a_{3j} correspondant est positif ce qui implique qu'on ne peut pas pivoter, donc la procédure s'arrête. Ceci montre que la région courante ne contient aucun point entier réalisable et que toutes les solutions efficaces ont été obtenues.

Conclusion :

L'inconvénient principal de cette méthode est qu'il faut passer par tous les points entiers de la région réalisable. A chaque fois, on effectue un test pour former l'ensemble des r-uplets efficaces correspondants. Même si toutes les solutions efficaces ont été trouvées, tant qu'il reste des points entiers non visités dans la région réalisable, la procédure continue son déroulement. L'exemple précédent met en évidence cet inconvénient. On vient bien qu'à partir de l'étape 7 jusqu'à l'étape 12 nous n'avons trouvé aucun autre point efficace.

1. Introduction

Dans ce chapitre nous donnerons une petite introduction sur le logiciel MATLAB, puis nous présenterons notre application qui est basée sur la programmation de la méthode Branch and Bound sur MATLAB.

2. Qu'est ce que MATLAB ?

MATLAB est une abréviation de Matrix Laboratory, écrit à l'origine, en Forton par C.
MATLAB et destiné à faciliter l'accès au logiciel matriciel développé dans les projets LINPACK (LIN pour « linear » = linéaire) et EISPACK (EIS pour « eigenvalues » = valeurs propres). Aujourd'hui MATLAB incorpore les bibliothèques LAPACK (linear Algebra) et BLAS (Basic Linear Algebra Subroutines).
MATLAB est un environnement puissant, complet et facile à utiliser. Il est destiné au calcul scientifique. Il apporte aux ingénieurs, chercheurs et à tout scientifique un système interactif intégrant calcul numérique et visualisation.
MATLAB s'opère depuis une session de commandes en ligne (voir figure 1). Celle-ci peuvent être exécutées une à une dans l'espace de travail (work space) .

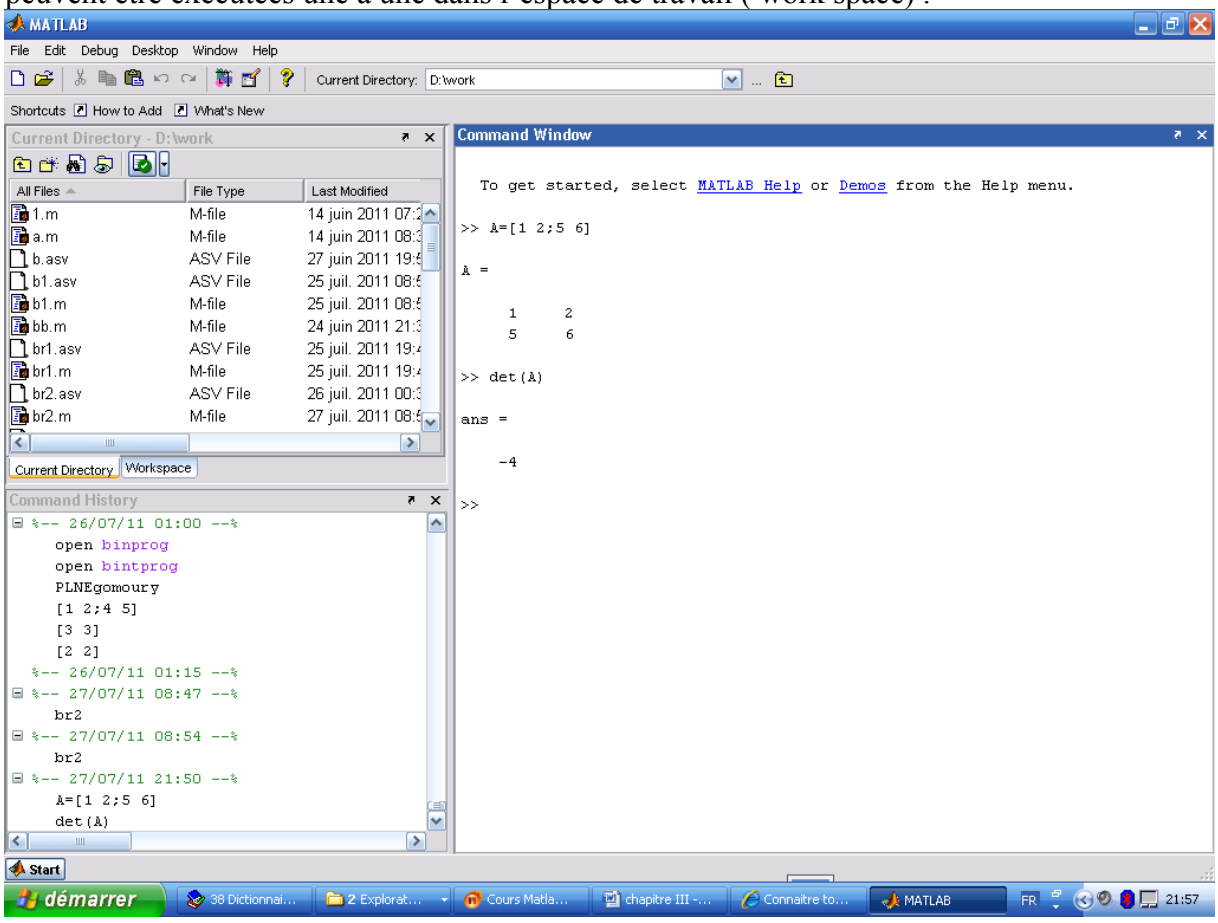


Figure 1

Deuxième partie : Implémentation de la méthode « Branch and Bound »

Il existe un grand nombre de fonctions et de commandes MATLAB qui nous permettent de réaliser :

- Des opérations vectorielles ou matricielles
- Des calculs statistiques
- La visualisation des données et images

. . . . . .

Dans MATLAB on peut programmer et créer les fonctions et fichiers nous même sur M-file, puis les sauvegarder et les exécuter sur l'espace de travail. Il suffit juste d'introduire le nom du fichier ou de la fonction.

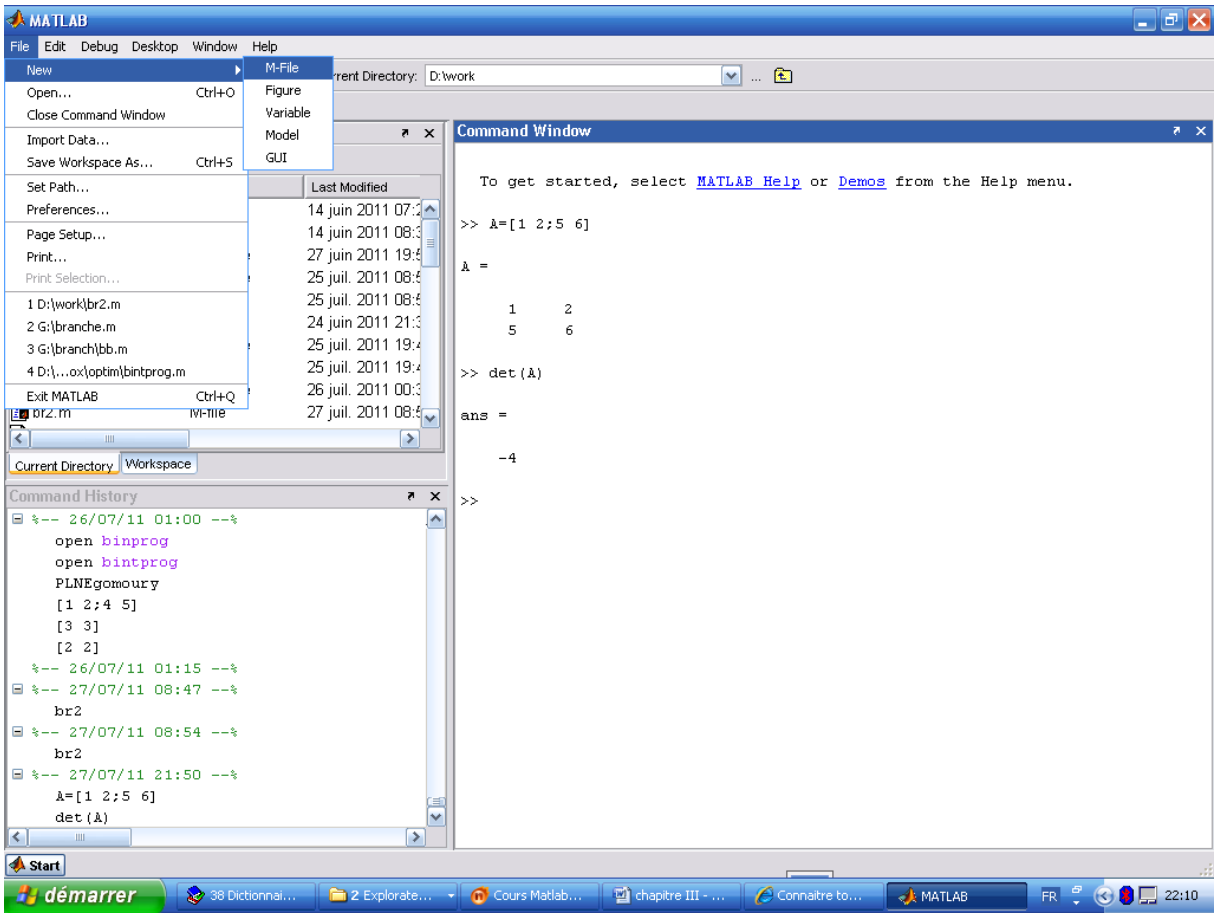


Figure 2

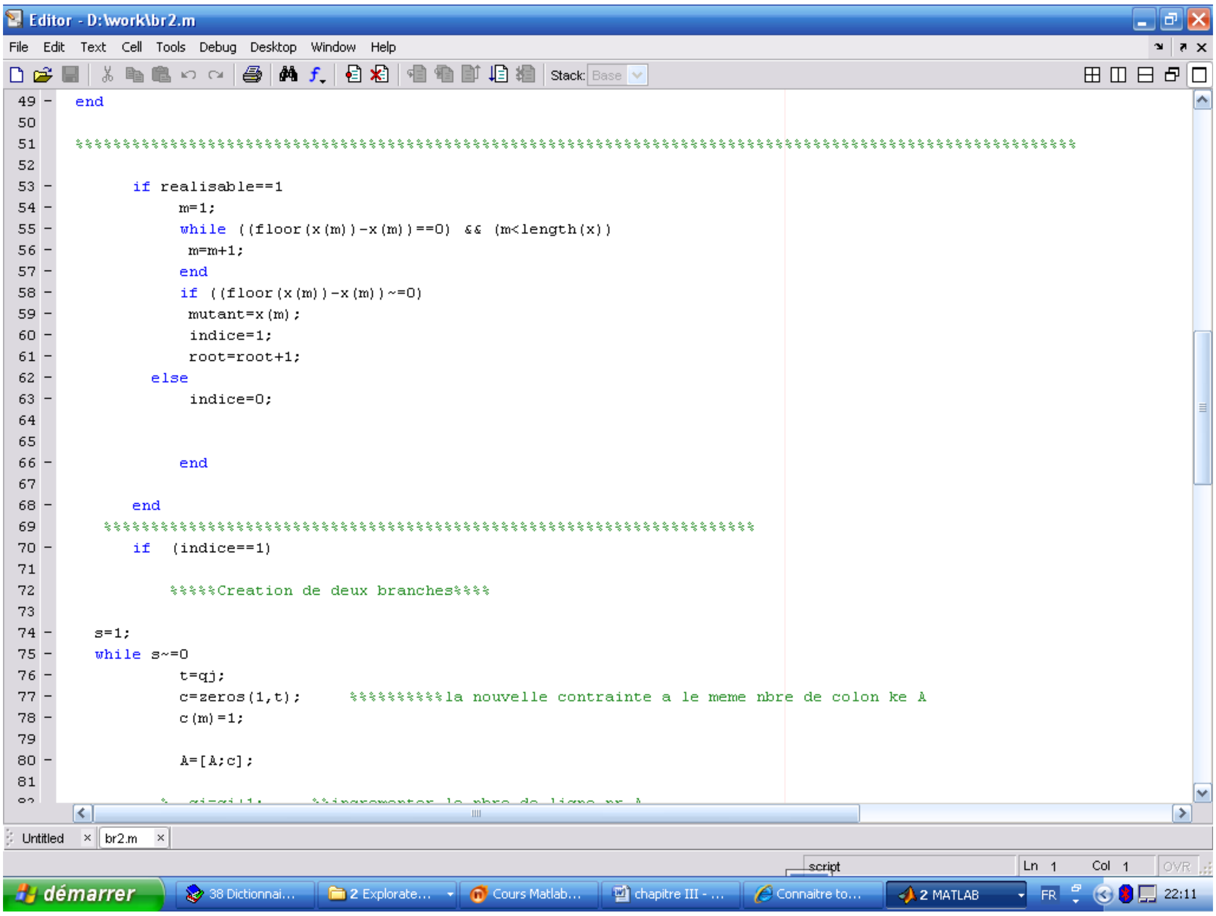


Figure 3

3. Implémentation de la méthode « Branch and Bound »

Avant de lancer notre programme, on va donner d'abord un exemple résolu manuellement :

$$max\, Z = 2x_1 + x_2$$

$$sc$$

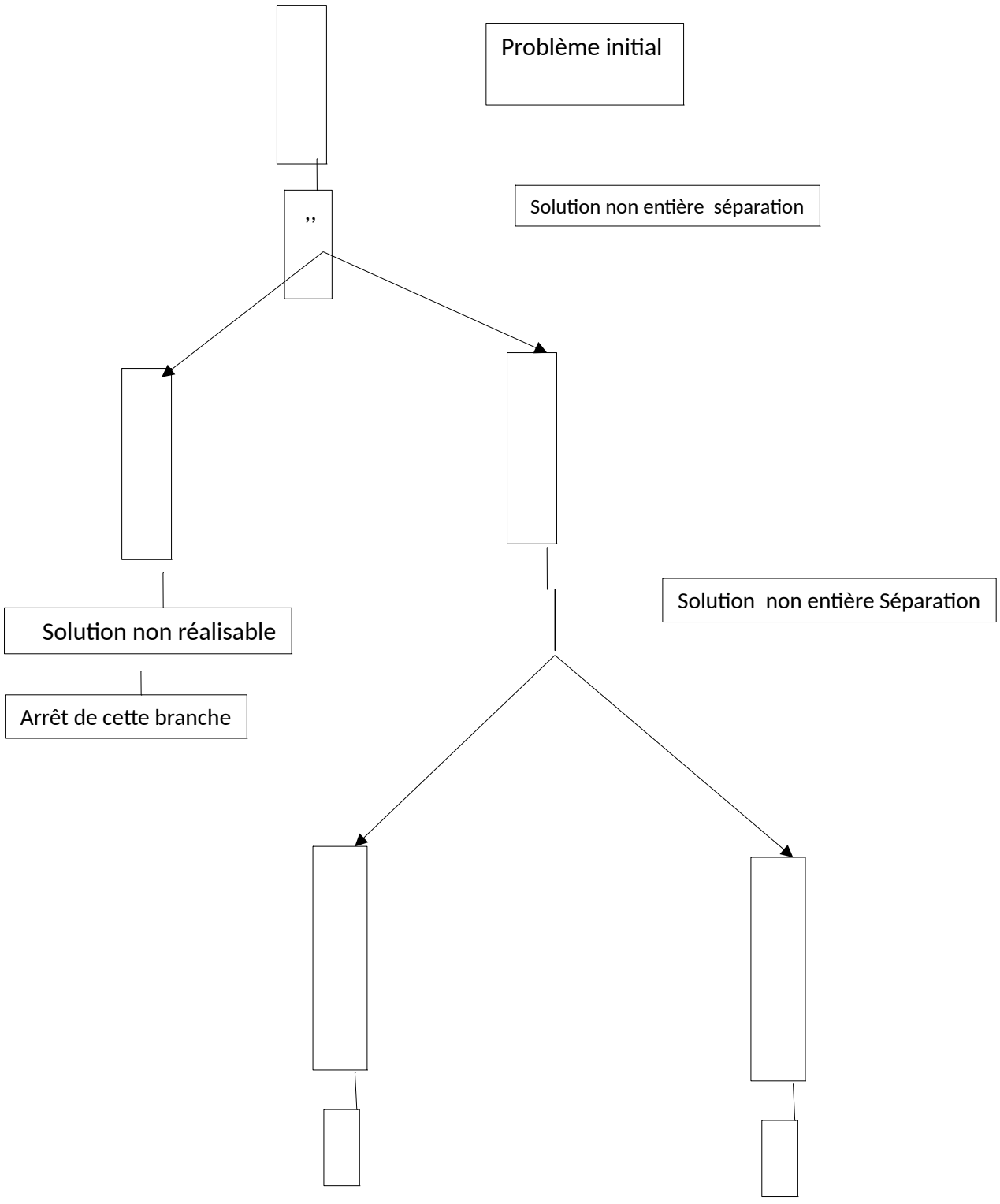
$$x_1 + 2x_2 \leq 12$$

$$2x_1 - x_2 \leq 8$$

$$4x_1 - 3x_2 \geq 4$$

$$x_1 \geq 0, x_2 \geq 0$$

$$x_1, x_2 entiers$$



La solution entière optimale est :

On peut avoir la solution de cette exemple sur notre application, il suffit d'introduire le non de notre programme sur MATLAB (espace de travail) qu'on a nommé « ma », d'introduire les données (la matrice, la fonction objectif, le vecteur b), puis cliquer sur la touche entrée. Nous obtenons la solution optimale entière accompagnée d'une figure sur laquelle apparaît l'arborescence correspondante.

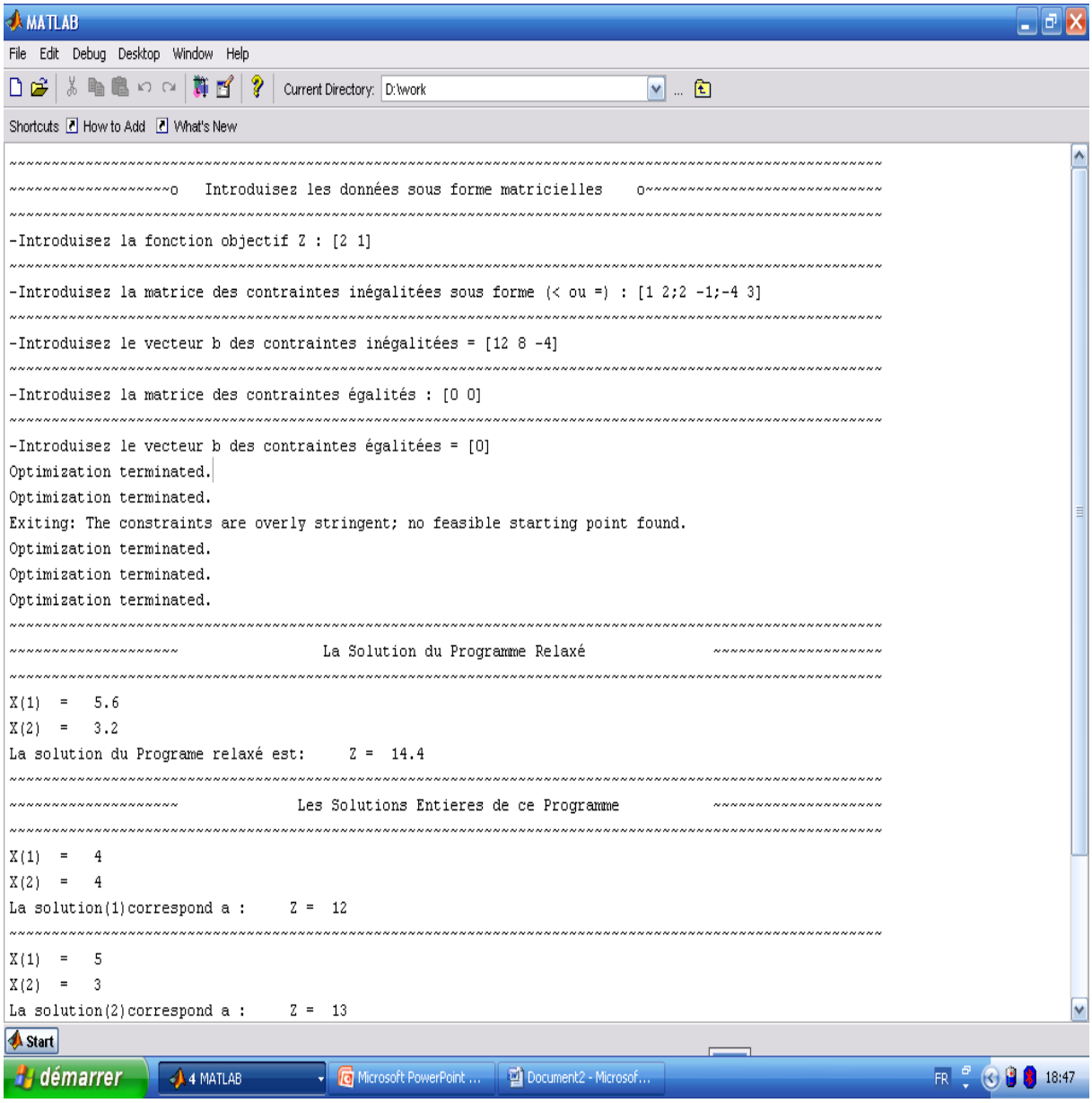


Figure 4

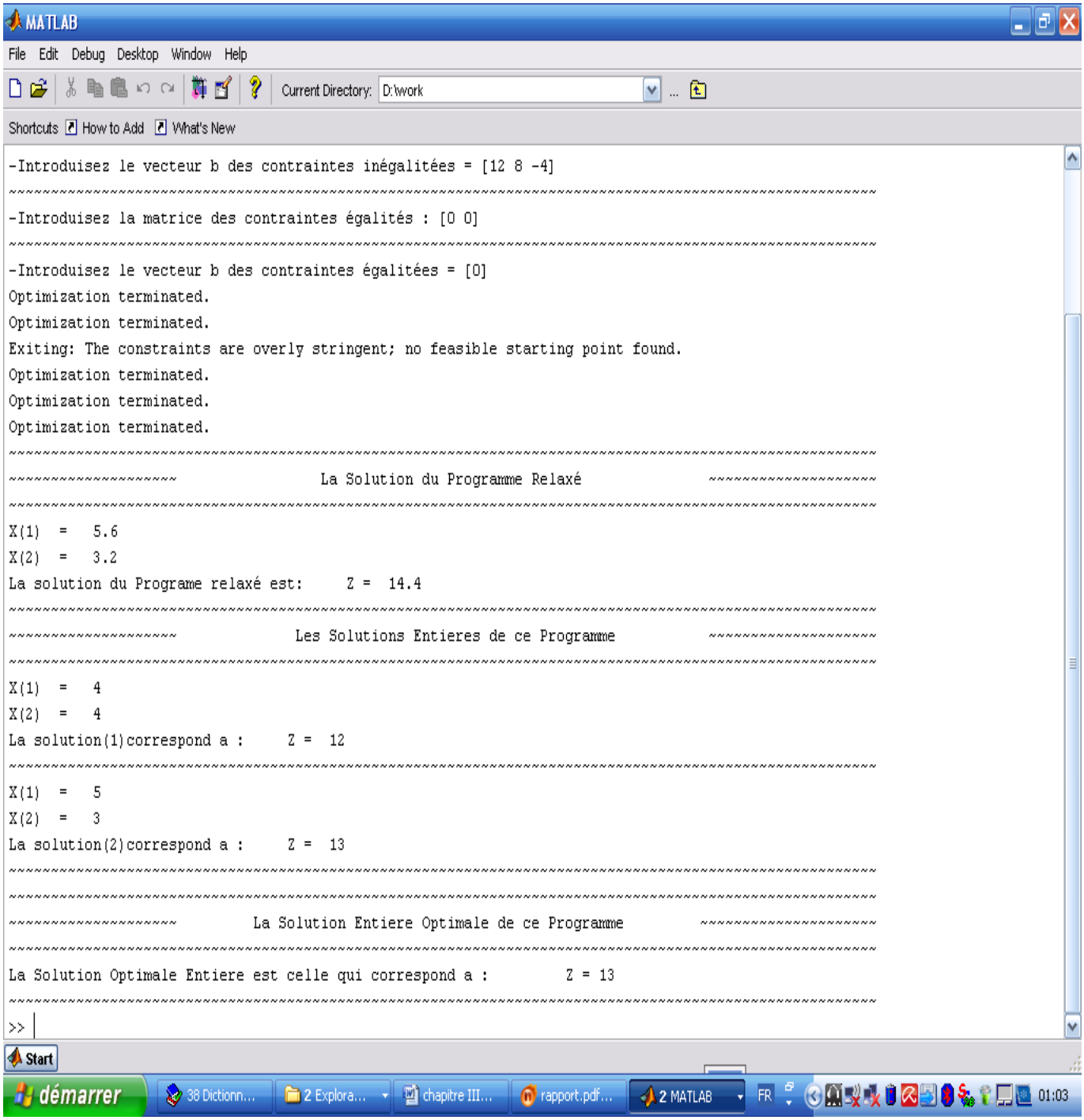


Figure 5

Sur les figures 4 et 5 on a la solution optimale du problème relaxé ainsi que les solutions entières correspondantes durant le processus de séparation.

Et vers la fin on a un message qui montre la solution entière optimale du problème donné.

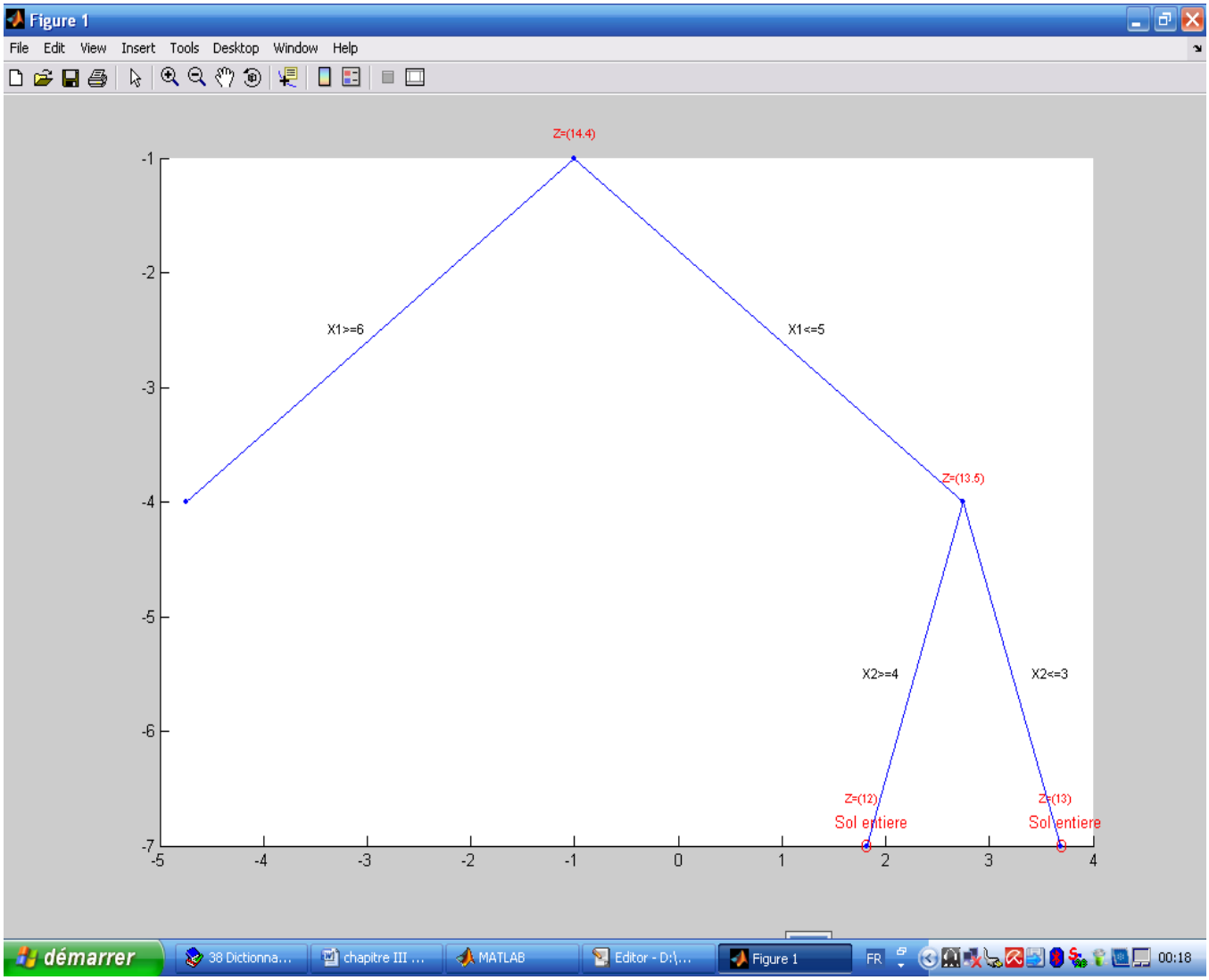


Figure 6

La figure 6 montre l'arborescence parcourue par la méthode Branch and bound sur l'exemple. Au premier nœud on a la solution du problème relaxé. Aucune variable n'est entière, on sépare sur x_1 . On construit deux nœuds, l'un avec la contrainte supplémentaire $x_1 \leq 5$ et l'autre avec la contrainte $x_1 \geq 6$. On résout les PL associés aux deux nœuds, l'un de ces nœud n'admet pas de solution réalisable, et l'autre a encore une solution non

entière et une évaluation de 13.5. On sépare donc ce nœud en deux nœuds, un avec la contrainte supplémentaire $x_2 \leq 3$ et l'autre avec la contrainte supplémentaire $x_2 \geq 4$.

Chacun de ces nœuds fournit une solution entière, un avec une évaluation $Z = 12$ et l'autre $Z = 13$, et la solution optimale de notre problème est celle qui donne une bonne évaluation pour Z c.à.d. $Z = 13$ et la solution optimale est $x_1 = 5, x_2 = 3$.

On peut donner un autre exemple :

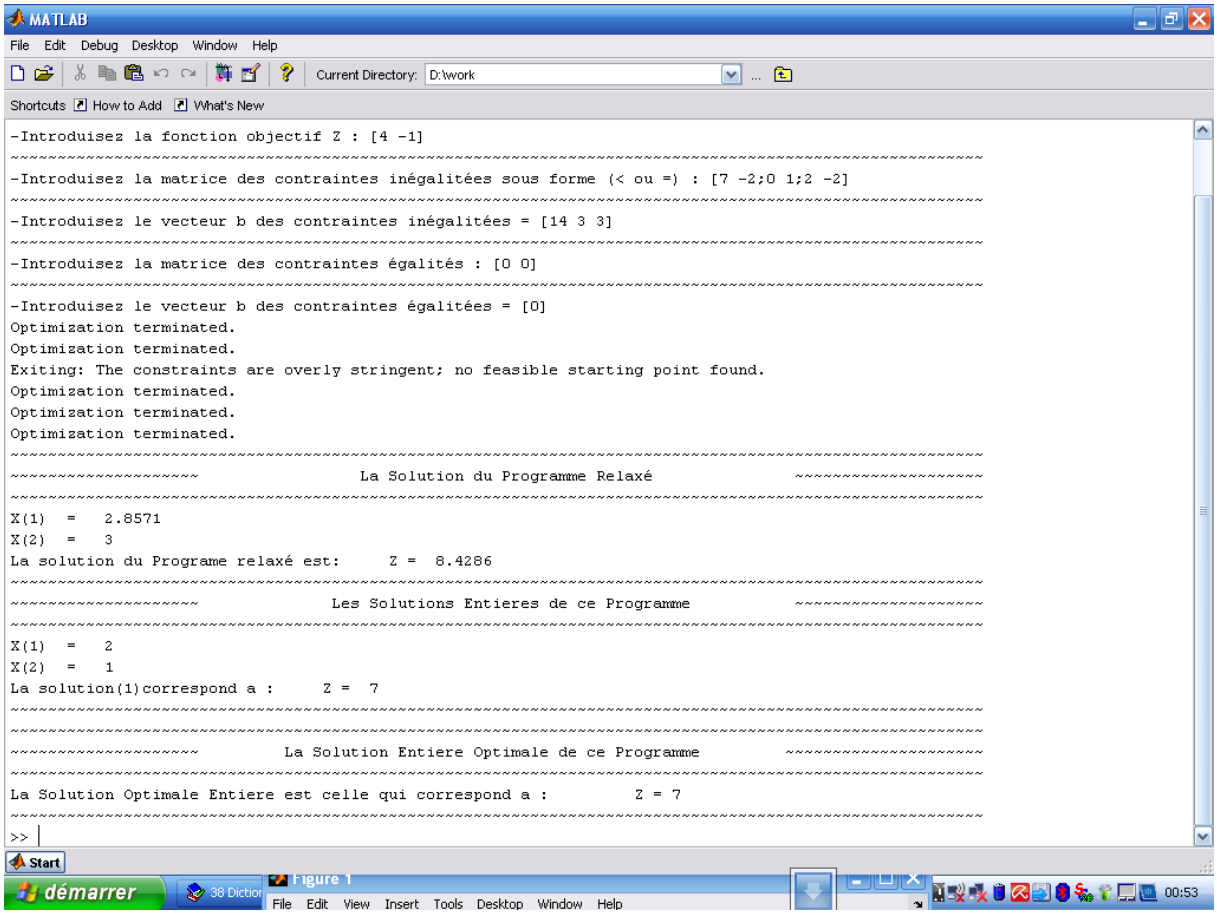
$$max\ 4x_1 - x_2$$

$$\text{sc} \begin{cases} 7x_1 - 2x_2 \leq 14 \\ x_2 \leq 3 \\ 2x_1 - 2x_2 \leq 3 \\ x_1, x_2 \in N \end{cases}$$

La solution optimale de problème relaxé est : $x_1 = 2.8571, x_2 = 3 \; et \; Z = 8.4286$

La solution optimale entière est : $x_1 = 2, x_2 = 1 \; et \; Z = 7$

On peut voir ces résultats spontanément sur notre application comme montre les figures ci-dessous :


MATLAB
File Edit Debug Desktop Window Help
Current Directory: D:\work
Shortcuts How to Add What's New
-Introduisez la fonction objectif Z : [4 -1]
-Introduisez la matrice des contraintes inégalitées sous forme (< ou =) : [7 -2;0 1;2 -2]
-Introduisez le vecteur b des contraintes inégalitées = [14 3 3]
-Introduisez la matrice des contraintes égalités : [0 0]
-Introduisez le vecteur b des contraintes égalitées = [0]
Optimization terminated.
Optimization terminated.
Exiting: The constraints are overly stringent; no feasible starting point found.
Optimization terminated.
Optimization terminated.
Optimization terminated.
La Solution du Programme Relaxé
X(1) = 2.8571
X(2) = 3
La solution du Programe relaxé est: Z = 8.4286
Les Solutions Entieres de ce Programme
X(1) = 2
X(2) = 1
La solution(1)correspond a : Z = 7
La Solution Entiere Optimale de ce Programme
La Solution Optimale Entiere est celle qui correspond a : Z = 7
>>
Start
démarrer
Figure 1
File Edit View Insert Tools Desktop Window Help
00:53

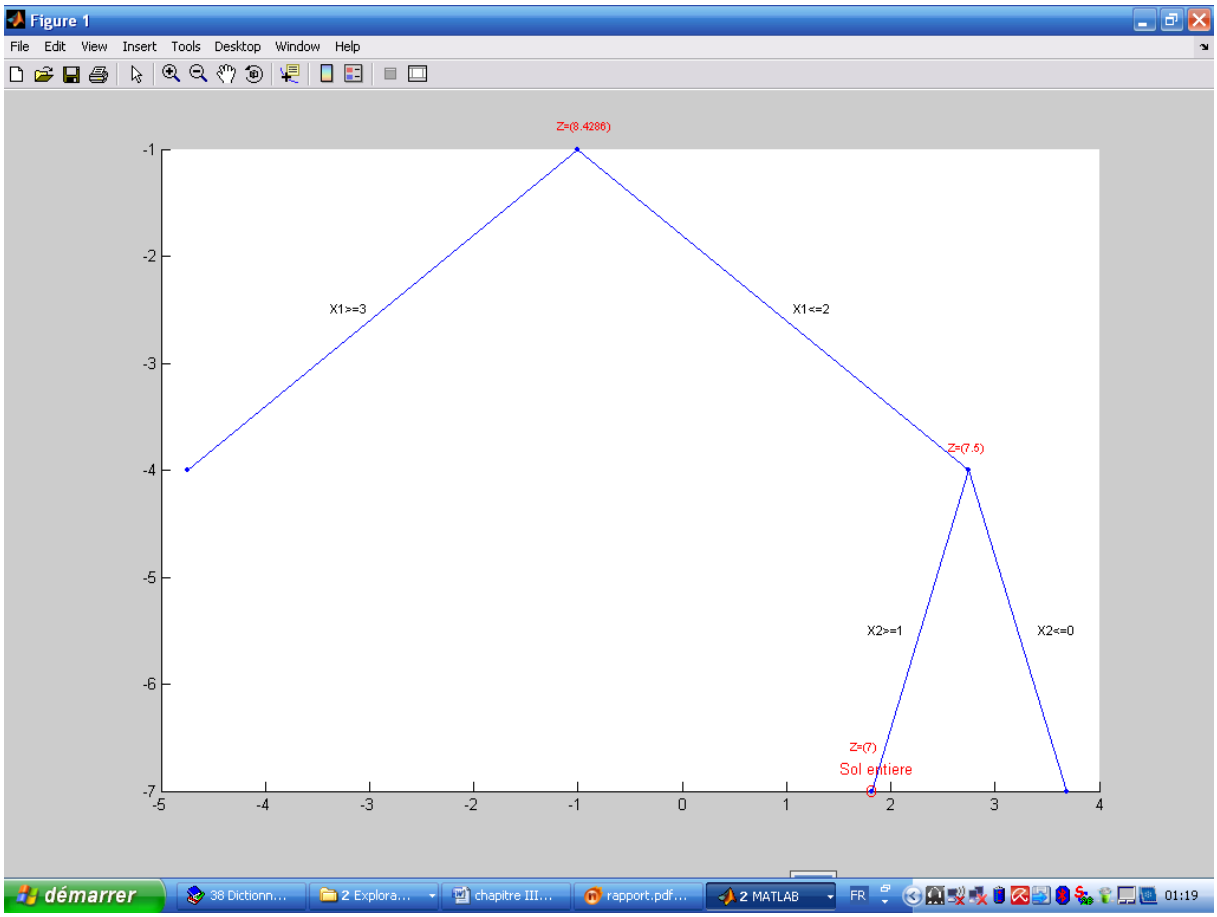


Figure 1
File Edit View Insert Tools Desktop Window Help
Z=(8.4286)
X1>=3
X1<=2
Z=(7.5)
X2>=1
X2<=0
Z=(7)
Sol entiere
démarrer
38 Dictionn...
2 Explora...
chapitre III...
rapport.pdf...
2 MATLAB
FR
01:19

[1] D. Klein, E. Hannan, An algorithm for the multiple objective integer linear programming problem. European Journal of Operational Research 9 (1982) 378_385.

[2] FREDERICK S, HILLIER, Nonlinear Integer Programming, Series Editor, stanford University (2006).

[3] G. BAILLARGEON, Programmation Linéaire Appliquée, les éditions SMG

[4] JOHN SYLVA & ALEJANDRO CREMA, A method for finding the set of non-dominated vectors for multiple objective integer linear programs, European Journal of Operational Research (2003).

[5] J. Sylva and A. Crema (2004), A method for finding the set of non-dominated vectors for multiple objective integer linear programs. European Journal of Operational Research 158 (1) 46_55.

[6] Laumanns, M.Thiele, L.Zitzler,E,2006. An efficient adaptive parameter variation scheme for metaheuristics based on the epsilon-constraint method. European journal of operational research 169, 932-942.

[7] M. Abbas, M. Moulaï, Solving multiple objective integer linear programming problem. Ricerca Operativa 29 (89) (1999) 15_39.

[8] Mordecai Avriel, Nonlinear programming, Analysis and methods (1976), Series in Automatic compulation

[9] M.Özlen, M.Azizoglu, multi-objective integer programming: A general approach for generating all non-dominated solutions, European journal of operational research (2008)

[10] R. Gupta, R. Malhotra, Multi-criteria integer linear programming problem. Cahiers du CERO 34 (1992) 51_68.

[11] R.E. Steuer, Multiple Criteria Optimization—Theory, Computation and Application, John Wiley and Sons, 1986.

Conclusion générale

Dans ce modeste travail, on a essayé de récapituler un peut sur l'optimisation discrète, tel que on a vu quelques algorithmes de résolution, soit dans les cas uni- critère ou bien multicritère, on peut conclure que l'optimisation discrète de manière générale est basée sur l'optimisation continue, vu l'utilisation successives des algorithmes de la méthode du simplexe et dual du simplexe, ce qui revient à augmenter les complexités des algorithmes applicables dans le cas discret.

Pour les algorithmes d'optimisation uni-critère dans le cas linaire, on a programmé la méthode de Branch and Bound dans la deuxième partie, comme on a donné l'interprétation graphique de cette méthode, c'est-à-dire à chaque fois qu'on exécute un problème sur ce programme, on n'aura pas seulement la solution entière optimale, mais on aura aussi l'arborescence de la méthode de branch and bound correspondante.

Pour résoudre un problème discret multi-objectif par l'un des algorithmes cité au chapitre II, on emploie toujours les algorithmes de résolution des problèmes discrets uni-objectif. Comme on a programmé la méthode de Branch and Bound, ce programme est aussi utile pour l'intégré dans ces méthodes étudiées dans ce chapitre, car la programmation de ces algorithmes ça nécessite la programmation de l'algorithme de Branch and Bound ou bien l'algorithme de Gomory, ce qu'est implique qu'on a facilité dans ce travail qu'on a réalisé la programmation de ces méthodes.

La perspective envisagée pour la suite de ce travail est d'essayé de programmer l'une des méthodes citée au chapitre II, on exploitons le programme de branch and bound pour effectuer la première étape de ces algorithmes (pour trouver la première solution efficace).

Pour les problèmes d'optimisation discrète uni-objectif, lorsqu'on cherche à obtenir une solution optimale, on trouve souvent une seule solution, par contre dans les problèmes d'optimisation multi-objectifs, lorsqu'on cherche une meilleure solution possible, ce cas est rarement rencontré, dans la plupart du temps, on trouve une multitude de solutions, du fait que certains des objectifs sont contradictoires.

Pour finir, on peut dire que l'optimisation discrète d'une manière générale n'est pas une tache facile, surtout l'optimisation multi-objectifs et la difficulté revient à les variables qui sont discret et aussi à la contestation qu'il ya sur certains objectifs, les méthodes

existantes ne résout pas ce problème à un seul coup, mais ils arrivent à déterminer l'ensemble de solutions efficaces, le choix de la solution reviendra au décideur.