

**MINISTÈRE DE L'ENSEIGNEMENT SUPÉRIEUR ET DE LA RECHERCHE
SCIENTIFIQUE
UNIVERSITÉ MOULOUD MAMMERI DE TIZI-OUZOU**

**FACULTÉ DES SCIENCES
DÉPARTEMENT DE MATHÉMATIQUES**



**MÉMOIRE DE FIN D'ÉTUDES EN VUE DE L'OBTENTION
DU DIPLÔME DE MASTER EN MATHÉMATIQUES
SPECIALITE : RECHERCHE OPERATIONNELLE
OPTION : METHODES ET MODELES DE DECISION**

THÈME :

Programmation Multi-objectif et application aux ensembles flous

Présenté Par :

M^{me} DILEM KAHINA

M^{elle} HAMMAD LILA

Devant le jury composé de :

M ^r D.HAMADOUCHE	Professeur	UMMTO	Président
M ^r M.CHEBBAH	M.A.A	UMMTO	Promoteur
M ^{me} O.BOUARAB	M.C.B	UMMTO	Examinatrice

Soutenu le 04.10.2015 à 10H

REMERCIEMENTS

Nous remercions tout d'abord « DIEU » tout puissant de nous
avoir donné la santé et le courage pour effectuer ce
projet de fin d'étude, dans les meilleures conditions.

Comme nous tenons à adresser notre reconnaissance et notre
gratitude à notre promoteur : Mr. M. CHEBBAH pour ses
précieux conseils et son suivi.

Nous remercions les membres du jury d'avoir accepter
d'examiner notre travail.

Enfin, nous tenons à remercier tous ceux qui ont contribué
de près et de loin à la réalisation de ce travail.

Dilem, Hammad.

Dédicaces

J'ai l'immense plaisir de dédier ce modeste travail

À

Mes très chers parents NAFA ET FATIMA,

Mon très cher mari HAKIM,

Mon frère MOUSSA et sa femme LILIA et mes sœurs
OUARDA, FADHILA et MERIEM et leurs maris,

Mes beaux parents TAHAR et FATIMA,

Mon Beau frère KARIM et à mes belles sœurs
KARIMA, OUARDIA, DJAHIDA et RADIA,

Et mon cher binôme HAMMAD LILA,

Kahina

Dédicaces

Je dédie ce travail

À

La mémoire de ceux qui m'ont été chers.

Mes parents et ma grande mère,

Mes frères et ma sœur,

Mes oncles, mes tantes et leurs familles,

Toute la promotion (Ro) 2014/2015,

Et mes amis et mon binôme.

Lila

Table des matières

Introduction générale

Chapitre 1 : Programmation multi-objectif

1.1	Introduction.....	1
1.2	Concepts de base.....	1
1.3	Définitions.....	2
1.3.1	Dominance.....	2
1.3.1.1	Définition (dominance)	2
1.3.1.2	Dominance forte.....	2
1.3.2.3	Dominance faible.....	3
1.3.1.4	Non dominance.....	3
1.3.2	Efficacité	3
1.3.2.1	Pareto optimalité.....	3
1.3.2.2	Efficacité faible.....	3
1.3.3	Point idéal.....	4
1.3.4	Point anti- idéal.....	4
1.3.5	Vecteurs de référence.....	4
1.3.6	Front de Pareto.....	4
1.3.7	Convexité.....	4
1.4	Teste d'efficacité	5
1.5	Méthode du simplexe multicritère.....	6
	Exemple.....	11

Chapitre 2 : Méthode multi-objectif linéaire en nombres entiers

2.1	Introduction.....	15
2.2	Formulation du problème.....	15
2.4	Résolution d'un problème multi-objectif linéaire en nombres entier... 15	
2.4.1	Méthode de Abbas et Moulai	15
	Exemple.....	18

Chapitre 3 : Programmation floue

3.1	Définition d'un sous ensemble flou.....	21
3.2	Caractéristique d'un sous ensemble flou.....	22
3.2.1	Support et hauteur.....	22
3.2.2	Noyau.....	23
3.2.3	Cardinalité.....	23
3.2.4	α .Coupe.....	23
3.3	Opérations sur les sous ensembles flous.....	24
3.3.1	Egalité.....	24
3.3.2	Complémentaire.....	24
3.3.3	Inclusion.....	24
3.3.4	Union.....	25
3.3.5	Intersection.....	25
3.4	Propriétés de l'union et de l'intersection	26
3.5	Nombres flous.....	26
3.5.1	Exemple de nombres flous.....	27
3.6	Comparaison (entre un nombre flou et un nombre réel)	29
3.7	Nombre flou de type L-R.....	30
3.7.1	Définition (négatif).....	30
3.7.2	Définition (positif).....	30
3.7.3	Opérations sur les nombres flous de type L-R.....	30
3.8	Programmation linéaire floue.....	32
3.8.1	Programmation flexible.....	32
3.8.1.1	Résolution d'un programme Linéaire flexible.....	32
Exemple.....		36
3.8.2	Programmation robuste.....	40
3.8.2.1	Résolution d'un programme robuste.....	40
3.8.2.2	Cas de nombres flous.....	41
3.8.2.3	Cas des nombres flous de type L-R.....	42

Exemples.....	43
---------------	----

Chapitre 4 : Programmation sur Lingo

4.1 Introduction.....	49
4.2 Installation du logiciel.....	49
4.3 Interface du logiciel	50
4.3 Application 1.....	53
4.4 Application 2.....	54
4.5 Application 3.....	55

Conclusion générale

Introduction générale

Depuis les années 70 , les activités de recherche en Recherche Opérationnelle au niveau mondial n'ont cessé de se développer tant au niveau de ses concepts théorique et de l'amélioration technique de ses outils d'optimisations qu'au niveau applicatif où elle intervient de manière cruciale dans des secteurs de plus en plus nombreux et diversifiés comme : «la production industrielle, la planification, le transport, l'informatique , les télécommunications , l'énergie , mais aussi dans les banques et les assurances... etc ». Les modèles traditionnels développés dans le cadre des méthodes quantitatives de gestion considéraient en général un critère unique, pour lequel il existe une solution optimale. Les algorithmes mis au point consistent alors à définir un moyen d'atteindre, le plus rapidement possible, une telle solution. Cependant, dans de nombreux cas cette modélisation ne traduit pas exactement la réalité.

La plupart des problèmes réels intervenant en mathématiques de décision sont de nature qui impose la prise en compte de plusieurs critères qui sont souvent antagonistes. Tout décideur est obligé de tenir compte du maximum d'éléments en sa possession, pour aboutir à la meilleure décision possible.

Ainsi pour mieux appréhender la réalité, l'approche multicritère devient incontournable. Il est utile dans ce cas de définir un concept d'optimalité, d'étudier les propriétés et les conditions d'existence des solutions et déterminer des méthodes pratiques de recherche des décisions relatives à ce concept d'optimalité.

La première notion d'optimalité en multi-objectif a été introduite par **Edgeworth** en 1881. Elle a été utilisée de manière plus formelle par l'économiste italien **Pareto** [1]. Cette notion est appelée efficacité, optimalité selon **Pareto** ou encore non dominance.

Mais aussi dans certaines situations les connaissances dont nous disposons sur une situation quelconque sont imparfaites soit parce que nous avons un doute sur leur validité , elles sont alors incertaines , soit parce que nous éprouvons une difficulté à les exprimer clairement , elles sont alors imprécises.

Les connaissances imprécises n'ont été prises en considération qu'à partir de 1965, lorsque **L.A. ZADEH** ,professeur à l'université de Californie à Berkeley , a introduit la notion de sous-ensemble flou , à partir de l'idée d'appartenance partielle à une classe de catégorie aux limites mal définies dans une généralisation dans la théorie classique des ensembles admettant des situations intermédiaires entre le tout et le rien.

La théorie des ensembles flous offre donc une structure mathématique dont laquelle des concepts vagues peuvent être précisément et rigoureusement étudiés. Elle peut également être considérée comme un langage de modélisation convenable à des situations caractérisées de relations, critères ou phénomènes flous.

Dans ce mémoire nous avons reparti le travail comme suit :

Chapitre 1 : On s'intéressera à la programmation multi-objectif, aux différents théorèmes d'existence de solutions efficaces et présentera aussi la méthode du simplexe multicritère.

Chapitre2 : On s'intéressera à l'étude des problèmes entiers. On présentera une méthode de résolution d'un problème (MOILP) , méthode **Abbas et Moulai** , ainsi qu'un exemple d'application résolu par la méthode d' **Abbas et Moulai** .

Chapitre 3 : On abordera la programmation linéaire floue, on donnera la forme général d'un programme linéaire flou, on montrera la façon de défuzzifier les contraintes linéaires et se ramener à un programme linéaire déterministe.

Chapitre 4 : Application sur le logiciel Lingo.

Chapitre 1

Programmation multi-objectif :

1.1 Introduction :

La programmation multi-objectif cherche à optimiser plusieurs composantes d'un vecteur de fonctions coût. Contrairement à l'optimisation mono-objectif, la solution d'un problème multi objectif (MOP) n'est pas une solution unique, mais un ensemble de solutions connues comme l'ensemble des solutions Pareto optimales (PO). Toute solution de cet ensemble est optimale dans le sens qu'aucune amélioration ne peut être faite sur un composant du vecteur sans la dégradation d'au moins un autre composant du vecteur.

MOP: **M**ultiple **O**bjective Programming.

MOLP: **M**ultiple **O**bjective **L**inear Programming.

1.2 Concepts de base :

Mathématiquement, un problème d'optimisation multi-objectif (MOP) peut être défini de la manière suivante :

$$(\text{MOP}) \quad \begin{cases} \max (\min) Z(x)=(Z_1(x), Z_2(x), \dots, Z_r(x)) \\ \text{s.c } x \in S. \end{cases}$$

Où $r \geq 2$ est le nombre de fonctions objectifs, $x = (x_1, x_2, \dots, x_n)$ est le vecteur représentant les variables de décision, $S = \{ x \in \mathbb{R}^n / g_j(x) \leq 0, x \geq 0 \}$ représente l'ensemble des solutions réalisables associées à des contraintes d'égalités, d'inégalités et de bornes explicites (espace de décision). $Z(x) = (Z_1(x), Z_2(x), \dots, Z_r(x))$ est le vecteur des critères à optimiser.

Z_i et $g_j, i \leq r$ et $j \leq m$: sont des fonctions à valeurs réelles du vecteur de décision.

L'ensemble $Y = Z(S)$ représente les points réalisables dans l'espace des critères (espace objectif), et $Z(x) = (Z_1(x), Z_2(x), \dots, Z_r(x))$ avec $Y_i = Z_i(x)$ représente un point de l'espace des critères.

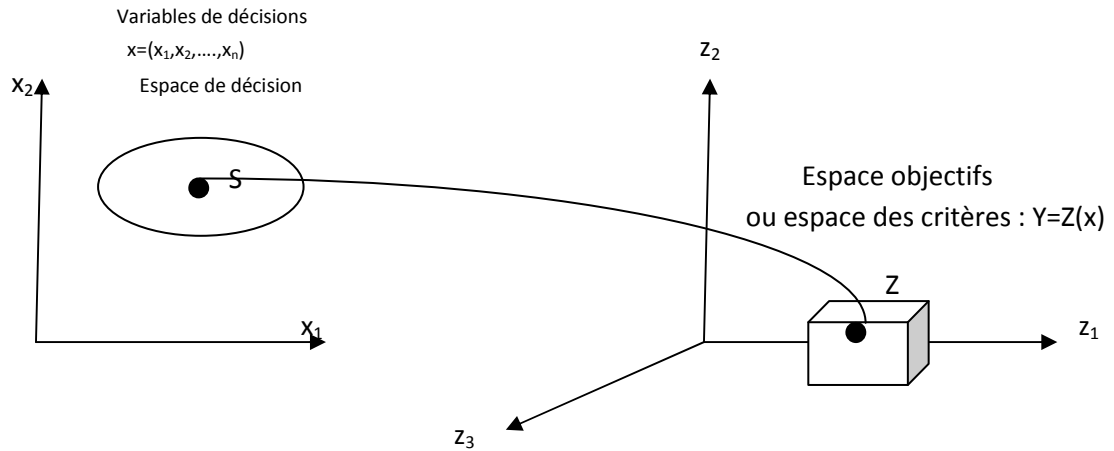


Fig1: Représentation de l'espace des décisions et l'espace des objectifs correspondant où $n=2$.

1.3 Définitions :

Remarque : Ces relations sont définies pour un problème de minimisation.

1.3.1. Dominance :

1.3.1.1. Définition : (Dominance)

Soient deux vecteurs critères $y = (y_1, y_2, \dots, y_r)$ et $z = (z_1, z_2, \dots, z_r)$.

O

On dit que y domine z si et seulement si :

$$\left\{ \begin{array}{l} \forall j \in [1, \dots, r] , y_j \leq z_j \\ \exists i \in [1, \dots, r] , y_i < z_i \end{array} \right.$$

C'est-à-dire :

Si y domine z , alors y est au moins aussi bon que z sur tous les critères et meilleur que lui sur au moins un des critères.

1.3.1.2. Définition : (Dominance forte)

Soient deux vecteurs critères $y = (y_1, y_2, \dots, y_r)$ et $z = (z_1, z_2, \dots, z_r)$.

On dit que y domine fortement z si et seulement si :

$$\forall i \in [1, \dots, r] \quad y_i < z_i$$

C'est-à-dire :

Si y domine fortement z , alors y est meilleur que z sur tous les critères.

1.3.1.3. Définition : (Dominance faible)

Soient deux vecteurs critères $y = (y_1, y_2, \dots, y_r)$ et $z = (z_1, z_2, \dots, z_r)$ on dit que y domine faiblement z si et seulement si :

$$\forall i \in [1, \dots, r] \quad y_i \leq z_i$$

1.3.1.4. Définition : (non – dominance)

Soit y^* un vecteur critère $\in Y$, on dit que y^* est non dominé si et seulement s'il n'existe aucun autre vecteur critère $y \in Y$ tel que:

$$\forall i \in [1, \dots, r] \quad y_i \leq y_i^*, \text{ et } y_i < y_i^* \text{ pour au moins un indice } i.$$

Dans le cas contraire, on dit que y^* est dominé.

1.3.2. Efficacité :

1.3.2.1. Définition : (Pareto optimale)

Une solution $x^* \in S$ est Pareto optimale si et seulement s'il n'existe pas une solution $x \in S$, telle que $Z(x)$ domine $Z(x^*)$,

$$\text{Et: } Z_i(x) \leq Z_i(x^*) \quad \forall i \in [1, 2].$$

La définition de la Pareto optimalité découle directement de la notion de dominance. Elle signifie qu'il est impossible de trouver une solution qui améliore les performances sur un critère sans que cela entraîne une dégradation des performances sur au moins un autre critère. Elles forment le front Pareto. Les solutions Pareto optimales sont aussi connues sous le nom de solutions efficaces, non-dominées ou non inférieures.

1.3.2.2. Définition : (efficacité faible)

Une solution $x^* \in S$ est dite faiblement efficace, s'il n'existe aucun vecteur $x \in S$

$$\text{Telle que } Z_i(x) < Z_i(x^*) \quad \forall i \in [1, 2].$$

Une solution est faiblement efficace si son vecteur critère est faiblement non dominé.

1.3.3. Définition : (le point idéal)

Le vecteur idéal $y^* = (y_1^*, y_2^*, \dots, y_r^*)$ est le vecteur qui optimise chacune des fonctions objectif Z_i .

i.e : $y_i^* = \min (Z_i(x), x \in S)$.

Il est clair que si le vecteur idéal est réalisable, il est la solution du problème(MOP), mais ce n'est pas en général possible à cause des conflits qui existent entre les critères

1.3.4. Définition: (point anti-idéal)

Le vecteur $y^* = (y_1^*, y_2^*, \dots, y_r^*)$ défini par :

$$y_i^* = \max ((Z_i(x), x \in S) , \text{ est le point anti-idéal.}$$

1.3.5. Définition : (vecteur de référence)

Un vecteur de référence $y^* = (y_1^*, y_2^*, \dots, y_r^*)$ est un vecteur qui définit le but à atteindre par chaque objectif Z_i .

1.3.6. Définition : (Front Pareto)

C'est l'ensemble des vecteurs de décision qui ne sont pas dominé.

1.3.7. Définition: (convexité)

Un ensemble A est convexe si et seulement si l'équivalence suivante est vérifiée :

$$\forall x \in A \text{ et } y \in A \Leftrightarrow \text{segment } (xy) \subset A$$

La convexité est le premier indicateur de la difficulté du problème. En effet, plusieurs méthodes d'optimisation sont incapables de résoudre d'une façon optimale des problèmes non convexes. Mais il existe d'autres indicateurs tout aussi important, notamment la continuité, la nature des variables de décision (entières ou réelles).

1.4. Test d'efficacité d'une solution :

Théorème 1 : Soit $x \in S$ est efficace s'il existe :

$$\lambda \in \Lambda = \{ \lambda \in \mathbb{R}^r / \sum_{j=1}^r \lambda_j = 1, \lambda_j \geq 0 \}$$

Respectivement : $\exists \lambda \in \bar{\Lambda} = \{ \lambda \in \mathbb{R}^r / \sum_{j=1}^r \lambda_j = 1, \lambda_j \geq 0 \}$ telle que x minimise le problème des sommes pondérées donné par :

$$\begin{cases} \text{Min } \lambda' C' x \\ \text{s.c} \\ x \in S \end{cases}$$

$$S = \{ x \in \mathbb{R}^n / Ax=b, x \geq 0 \}$$

Théorème 2 : Si S possède un point efficace, alors au moins un point extrême de S est efficace.

Preuve :

Soit \bar{x} un point efficace de S , d'après le théorème 1, il existe $\lambda \in \Lambda$ tel que :

$$\lambda' C' \bar{x} = \min_{x \in S} \lambda' C' x$$

Comme une fonction linéaire atteint son optimum en un point extrême, donc \bar{x} est extrême efficace.

Théorème 3 : Soit $x \in S$ un point extrême associé à une base efficace B , alors x est efficace.

Preuve :

Puisqu'il existe un $\lambda \in \Lambda$ pour lequel B est une base optimale par le théorème 1, x est efficace.

Remarque : On dit que x est un point extrême de S s'il ne peut pas s'écrire comme une combinaison convexe stricte de deux points de S .

$\nexists x_1, x_2 \in S, \alpha \in]0,1[$ [pour les quels $(1-\alpha)x_1 + \alpha x_2 = x / x_1 \neq x_2$.

Théorème 4 : Soient B et \bar{B} deux bases efficaces adjacentes obtenues à partir d'un pivot efficace, et soient x et \bar{x} les points extrêmes associés à B et à \bar{B} respectivement alors, l'arête (x, \bar{x}) est efficace.

Théorème 5 :

Soit (x, v) l'arête efficace infini de S .

Alors, x est un point extrême efficace associé à une base efficace B .

1.5. Méthode du simplexe multicritère :

La méthode du simplexe multicritère consiste à générer un premier point efficace à partir d'une solution de base réalisable, puis à recenser (énumérer) tous les autres points efficaces. Cependant cette méthode ne teste pas toutes les bases car certaines sont dominées de manière évidente.

Étant donné le problème :

$$(P) \quad \left\{ \begin{array}{l} \max Z_1 = c^1 x \\ \max Z_2 = c^2 x \\ \vdots \\ \max Z_k = c^k x \\ Ax = b, x \geq 0 \end{array} \right.$$

Le tableau du simplexe (en considérant séparément chaque objectif) est donné par le tableau (1).

Dans ce tableau, on suppose sans perte de généralités que les m premières variables sont dans la base.

Soient :

J_B : ensemble des indices basiques.

J_N : ensemble des indices hors base.

					C^1	C_1^1	C_m^1	C_{m+1}^1	C_j^1
					C^2	C_1^2	C_m^2	C_{m+1}^2	C_j^2
					\vdots							
					C^K	C_1^K	C_m^K	C_{m+1}^K	C_j^K
C_B^1	C_B^2	C_B^j	C_B^k	base	B	a_1	a_m	a_{m+1}	a_j
C_1^1	C_1^2	C_1^j	C_1^k	a_1	b_1	1		0	x_{1m+1}	x_{1j}
						0		0				
						0		0				
C_m^1	C_m^2	C_m^j	C_m^k	a_m	b_m	0		1	x_{mm+1}	x_{mj}
						0	0	0	Δ_{m+1}^1		Δ_j^i	
							0	0	Δ			
						0	0		Δ_{m+1}^i	Δ_j^1
							0					
						0	0	0	Δ_{m+1}^k	Δ_j^k

$$Z_0 = \begin{pmatrix} Z_0^1 = C^1 b \\ Z_0^2 = C^2 b \\ \dots \\ Z_0^k = C^k b \end{pmatrix}$$

De la théorie du simplexe uni-critère on a :

$$1- \Delta_j^i = \sum_{r \in J_B} C_r^i x_{rj} - C_j^i, j \in J_N, \forall i = 1, \dots, k$$

Et:

$$Z_0^i = \sum_{r \in J_B} C_r^i b_r, \forall i = 1, \dots, k$$

Si: $\Delta_j^i \geq 0, \forall j \in J_N$, alors $x^0 = t_{(b,0)}$, $(b \in \mathbb{R}^{m+}, 0 \in \mathbb{R}^{n-m})$ est une solution optimale pour le critère i.

2- Si on introduit la $j^{\text{ème}}$ variable dans la base, nous obtenons une nouvelle solution x_1 et un vecteur : $\hat{Z}_0 = Z_0 - \theta_j \Delta_j$ ou :

$$\hat{Z}_0 = \begin{pmatrix} \hat{Z}_0^1 \\ \hat{Z}_0^2 \\ \dots \\ \hat{Z}_0^k \end{pmatrix} = \begin{pmatrix} Z_0^1 \\ Z_0^2 \\ \dots \\ Z_0^k \end{pmatrix} - \theta_j \begin{pmatrix} \Delta_j^1 \\ \Delta_j^2 \\ \dots \\ \Delta_j^k \end{pmatrix}$$

$$3- \theta_j = \min \left\{ \frac{b_r}{x_{rj}} / x_{rj} > 0 \right\} \forall j \in J_N, r \in J_B.$$

Remarque 1:

1- Soit x° une solution basique réalisable.

a- S'il existe un $j \in J_N$ tel que tous les $\Delta_j^{(i)} \leq 0$, avec au moins une inégalité stricte et si $\theta_j > 0$, alors la solution courante x° est dominée.

En effet :

Si on introduit la $j^{\text{ème}}$ variable, on obtient un point extrême adjacent x^1 pour lequel $\hat{Z} \geq Z_0$ avec au moins une inégalité stricte, car :

$\theta_j \Delta_j \leq 0$, donc $\hat{Z}_0 = Z_0 - \theta_j \Delta_j \geq Z_0$.

b- S'il existe un $j \in J_N$ tel que $\Delta_j^{(i)} \geq 0$, avec au moins une inégalité stricte et si de plus $\theta_j > 0$, alors l'introduction de la jème variable a la base mène à une solution dominée.

En effet :

Si on introduit la variable j dans la base, on aura un nouveau point extrême x_1 pour lequel :

$$\hat{Z}_0 = Z_0 - \theta_j \Delta_j \leq Z_0 \text{ avec au moins une inégalité stricte.}$$

2- Soit x^0 une solution basique réalisable s'il existe $j_1, j_2 \in J_N$ tel que :

$\theta_{j_1} \Delta_{j_1} \leq \theta_{j_2} \Delta_{j_2}$ avec au moins une inégalité stricte, alors l'introduction de la variable d'indice j_2 dans la base conduit à une solution dominée par celle résultante de l'introduction de la variable d'indice j_1 :

En effet :

$\theta_{j_1} \Delta_{j_1} \leq \theta_{j_2} \Delta_{j_2}$ avec au moins une inégalité stricte, cela implique que :
 $Z_0 - \theta_{j_2} \Delta_{j_2} \leq Z_0 - \theta_{j_1} \Delta_{j_1}$ avec au moins une inégalité stricte.

Remarque 2 :

1- S'il existe un indice $i \in \{1, \dots, k\}$ tel que $\Delta_j^i \geq 0 \forall j \in J_N$, alors le ième critère est à son maximum et la solution basique correspondante est non dominée à condition qu'il n'existe pas de colonne k avec $\Delta_k^i = 0$.

2- D'après la remarque précédente, seules les colonnes (variables) non comparables à zéro et les variables $j, k \in J_N$ telle que : $\theta_k \Delta_k$ incomparable à $\theta_j \Delta_j$ sont admissibles pour une introduction dans la base.

Dans ce cas, on ne peut dire si la solution correspondante \bar{x} est dominée ou pas.

Pour cela on considère le test dit de non dominance énoncé par le théorème suivant :

Théorème 7 : (Test de dominance)

Soit le problème :

$$\max v = \sum_{i=1}^k \varepsilon_i$$

$$C'x - \xi = C'\bar{x}$$

$$x \in S = \{x \in \mathbb{R}^n \mid Ax = b, x \geq 0\}$$

$$\xi \in \mathbb{R}^n, \xi \geq 0 \text{ et } \bar{x} \in S.$$

Alors :

\bar{x} est efficace si et seulement si $\max v = 0$.

\bar{x} est dominée si et seulement si $\max v > 0$.

Exemple :

$$\text{Max } Z_1(x) = 0.6x_1 + 0.2x_2$$

$$\text{Max } Z_2(x) = -2x_1 + x_2$$

$$(1) \begin{cases} x_1 + 3x_2 \leq 100 \\ 2x_1 + x_2 \leq 200 \\ x_1 \geq 0, x_2 \geq 0 \end{cases}$$

$$(1) \Leftrightarrow \begin{cases} \text{Max } Z_1(x) = 0.6x_1 + 0.2x_2 \\ \text{Max } Z_2(x) = -2x_1 + x_2 \\ x_1 + 3x_2 + x_3 = 100 \\ 2x_1 + x_2 + x_4 = 200 \\ x_j \geq 0, j=1, \dots, 4 \end{cases}$$

$x_1 = (0, 0, 100, 200)$ solution de base réalisable

$$(a_3, a_4) = I_2, Z(x) = \begin{pmatrix} Z_1(x) \\ Z_2(x) \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$

1^{er} tableau du simplexe :

			C^1	0,6	0,2	0	0	θ
			C^2	-2	1	0	0	
C_1^B	C_2^B	Base	B	a_1	a_2	a_3	a_4	
0	0	a_3	100	1	3	1	0	100
0	0	a_4	200	2	1	0	1	100
			Δ_1	-0,6	-0,2	0	0	
			Δ_2	2	-1	0	0	

$$\Delta_1 = \begin{pmatrix} -0,6 \\ 2 \end{pmatrix}$$

$$\Delta_2 = \begin{pmatrix} -0,2 \\ -1 \end{pmatrix}$$

$$\theta_1 = \min \left(\frac{100}{1}, \frac{200}{2} \right) = 100 \text{ (r=4)}$$

$$\theta_2 = \min \left(\frac{100}{3}, \frac{200}{1} \right) = 33,33 \text{ (r=3)}$$

$$\theta_1 \Delta_1 = 100 \begin{pmatrix} -0,6 \\ 2 \end{pmatrix} = \begin{pmatrix} -60 \\ 200 \end{pmatrix}$$

$$\theta_2 \Delta_2 = 33,33 \begin{pmatrix} -0,2 \\ -1 \end{pmatrix} = \begin{pmatrix} -6,66 \\ -33,33 \end{pmatrix}$$

$\theta_1 \Delta_1$ et $\theta_2 \Delta_2$ sont incomparables.

1^{er} cas :

Introduisant en premier lieu a_2 dans la base à la place de a_3 , on aura le tableau suivant :

			C^1	0,6	0,2	0	0	θ
			C^2	-2	1	0	0	
C_1^B	C_2^B	Base	B	a_1	a_2	a_3	a_4	
0	0	a_2	100/3	1/3	1	1/3	0	100
0,6	1	a_4	500/3	5/3	0	-1/3	1	100
			Δ_1	-0,17	0	0,06	0	
			Δ_2	7/3	0	0,2/3	0	

$\exists i_2$ tel que $\Delta_j^i > 0 \forall j \in J_N$ ce qui implique que le deuxième critère est à son maximum

Donc la solution (0,100/3) est efficace.

Introduisons la base a_1 à la place de a_2

On aura le tableau suivant :

			C^1	0,6	0,2	0	0	θ
			C^2	-2	1	0	0	
C_1^B	C_2^B	Base	B	a_1	a_2	a_3	a_4	
0,2	1	a_2	0	0	1	2/5	-1/5	0
0,6	-2	a_1	100	1	0	-1/5	3/5	/
			Δ_1	0	0	-0,04	0,32	
			Δ_2	0	0	4/5	-7/5	

$$\Delta_1 = \begin{pmatrix} -0,04 \\ 4/5 \end{pmatrix}$$

$$\Delta_2 = \begin{pmatrix} -0,32 \\ -7/5 \end{pmatrix}$$

$$\theta_1 = \min(0) = (r=2)$$

$$\theta_2 = \min(500/3) = 500/3 \quad (r=1)$$

$$\theta_1 \Delta_1 = 0 \begin{pmatrix} -0,04 \\ 4/5 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$

$$\theta_2 \Delta_2 = 500/3 \begin{pmatrix} -0,32 \\ -7/5 \end{pmatrix} = \begin{pmatrix} -53,33 \\ -233,33 \end{pmatrix}$$

$\theta_1 \Delta_1$ et $\theta_2 \Delta_2$ sont incomparable

On introduit la base a_3 a la place de a_2 on aura le tableau suivant :

			C^1	0,6	0,2	0	0	θ
			C^2	-2	1	0	0	
C_1^B	C_2^B	Base	B	a_1	a_2	a_3	a_4	
0	0	a_3	0	0	5/2	1	-1/2	/
0,6	-2	a_1	100	1	1/2	0	1/2	200
			Δ_1	0	-0,7	0	0,3	
			Δ_2	0	-2	0	-1	

→

↑

$$\Delta_1 = \begin{pmatrix} -0,7 \\ -2 \end{pmatrix}$$

$$\Delta_2 = \begin{pmatrix} 0,3 \\ -1 \end{pmatrix}$$

$$\theta_3 = \min(200) = (r=3)$$

$$\theta_1 = \min(0,200) = 0 \quad (r=1)$$

$$\theta_3 \Delta_1 = 200 \begin{pmatrix} -0,7 \\ -2 \end{pmatrix} = \begin{pmatrix} -140 \\ -400 \end{pmatrix}$$

$$\theta_1 \Delta_2 = 200 \begin{pmatrix} 0,3 \\ -1 \end{pmatrix} = \begin{pmatrix} 60 \\ -200 \end{pmatrix}$$

$\theta_3 \Delta_1$ et $\theta_1 \Delta_2$ sont incomparable

- 1^{er} cas : on introduit a_4 a la place de a_1 on aura une nouvelle base (a_3, a_4) qui est déjà parcouru .
- 2^{eme} cas : on introduit a_2 à la place de a_3 et on aura la nouvelle base (a_2, a_1) qui est aussi une base déjà parcouru. Alors on s'arrête .

2^{eme} cas :

			C^1	0,6	0,2	0	0	
			C^2	-2	1	0	0	
C_1^B	C_2^B	Base	B	a_1	a_2	a_3	a_4	
0	0	a_3	100	1	3	1	0	→
0	0	a_4	200	2	1	0	1	
			Δ_1	-0,6	-0,2	0	0	
			Δ_2	2	-1	0	0	

↑

$$\theta_1 = \min(100,100) = 100$$

Si on fait sortir de la base a_4 on aura (a_3, a_1) qui est déjà parcouru.

Donc on fait sortir a_3 et on aura la base (a_1, a_4)

			C^1	0,6	0,2	0	0
			C^2	-2	1	0	0
C_1^B	C_2^B	Base	B	a_1	a_2	a_3	a_4
0.6	-2	a_1	100	1	3	-2	0
0	0	a_4	0	0	-5	-2	1
			Δ_1	0	1,6	0,6	0
			Δ_2	0	-7	-2	0



On a une solution efficace (100,0)

On aura deux possibilités de nouvelles bases :

$(a_2, a_4) \longrightarrow$ Déjà parcourue.

$(a_1, a_4) \longrightarrow$ Déjà parcourue.

Chapitre 2

Méthode multi-objectif linéaire en nombres entiers

2.1. Introduction :

Dans le problème multi-objectif, lorsque, les fonctions $Z_i(x)$ pour $i = 1, \dots, r$ sont linéaires et que l'ensemble des solutions réalisable S est défini par des contraintes linéaires, alors, on parle de problème linéaire multi-objectif (MOLP). Ainsi, si les variables x prennent des valeurs entières dans le problème (MOLP). On parle alors de problème d'optimisation multi-objectif linéaire en nombres entiers noté (MOILP).

MOILP: Multiple Objective Integer Linear Programming.

2.2. Formulation du problème :

Un problème (MOILP) est formulé comme suit :

$$(\text{MOILP}) \quad \begin{cases} \text{Max}(Z_1, Z_2, \dots, Z_r) \\ Z_i = C'_i(x) \\ x \in S \end{cases}$$

Ou : $S = \{x \in R^n / Ax \leq b, x \geq 0, x \text{ entier}\}$

r : Nombre d'objectifs à minimiser ou maximiser.

C_i : Vecteur des coefficients pour $i = 1, \dots, r$.

A : la (m, n) matrice des contraintes, avec $rg A = m \leq n$.

Z_i : Critère à maximiser ou minimiser pour $i = 1, \dots, r$.

2.4. Résolution d'un problème multi-objectif linéaire en nombre entiers :

2.4.1. Méthode de Abbas et Moulai :

C'est une méthode de coupes. Elle est appropriée à la résolution des problèmes (MOILP). La première itération consiste à déterminer une solution entière réalisable pour le problème relaxé. La seconde itération nous permet de trouver toutes les solutions entières alternatives à la solution optimale obtenue (si elle existe) et obtenir la première solution entière efficace du problème principal. Dans la prochaine étape, nous introduisons une coupe afin de calculer les autres solutions entières efficaces restantes en utilisant la méthode duale du simplexe.

L'algorithme s'arrête quand tout le domaine de décision est exploré.

Soit le problème (P₁) suivant :

$$(P_1) \quad \left\{ \begin{array}{l} \max Z_1(x) = C'x \\ \text{sc.} \\ x \in S \\ x \text{ entier} \end{array} \right.$$

Procédure : [2]

Etape 1 :

Résoudre le problème (P₁) et trouver la solution optimale entière x_1^1 sur S en choisissant un seul objectif.

Construire τ_1 .

$$\tau_k = \{j \in N_k / \Delta_{kj}^1 = 0\}$$

N_k = Ensemble des indices hors base.

Etape 2 :

- 1) Si $\tau_1 = \emptyset, x_1^1$ est l'unique solution optimale du problème sur S_1 et Z_1^1 est la valeur optimale de Z_1 . Calculer les valeurs Z_i^1 de Z_i donné par $x_1^1, i \in \{2, \dots, r\}$
 -Enregistrer le r-uplet efficace comme étant (Z_1^1, \dots, Z_r^1) pour construire l'ensemble efficace Eff_0 .
 -Tronquer le point x_1^1 par la coupe suivante : $\sum_{j \in N_1} x_j \geq 1$
 En appliquant la méthode duale du simplexe et, si nécessaire, des coupes successives de Gomory, on obtient une solution réalisable entière $x_1^1 = (x_{2,j}^1)$ dans la région tronquée S_2 .
 Lire le r-uplet correspondant (Z_1^1, \dots, Z_r^1) et l'ajouter à Eff_0 s'il n'est pas dominé par l'un des r-uplets efficaces identifiés auparavant pour obtenir Eff_1 .
- 2) Si $\tau_1 \neq \emptyset$, choisir un indice quelconque $j_1 \in \tau_1$ et calculer le nombre
 $\theta_{j_1} \leq \min \{x_{1,1}^1 / y_{1,lj_1}^1; y_{1,lj_1}^1 > 0\} \quad I \in I_1 \text{ correspondant à } x_1^1$
 a) Si $\theta_{j_1} \geq 1$, déterminer toutes les solutions entière réalisables
 $x_i^q, q \in \{2, \dots, p_K\}$ alternatives à x_1^1 le long de l'arête E_{j_1} . Chaque solution apporte une amélioration au r-uplet potentiellement non dominé à l'étape 2.
 -Tronquer l'arête E_{j_1} par la coupe suivante : $\sum_{j \in N_1 - \{j_1\}} x_j \geq 1$
 Et ceci en appliquant la méthode duale du simplexe et de successives coupes de Gomory.
 Pour obtenir une solution réalisable entière $x_1^1 = (x_{2,j}^1)$ dans la région tronquée S_2 .
 Lire le r-uplet correspondant (Z_1^1, \dots, Z_r^1) et l'ajouter à Eff_0 s'il n'est pas dominé par l'un des r-uplet efficaces précédemment identifiés pour obtenir Eff_1 .

b) Si $\theta_{j_1} < 1$ pour tout les indices $j_1 \in \tau_1$, choisir un indice $j_1 \in \tau_1$ et appliquer la coupe ;

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

- En utilisant la méthode duale du simplexe et de successives coupes de Gomory, on obtient alors une solution réalisable entière : $x_2^1 = (x_2^1, j)$ dans S_2 . Lire le r-uplet correspondant (Z_1, Z_2, \dots, Z_r) et le rajouter à l'ensemble Eff_0 s'il n'est pas dominé par l'un des r-uplets efficaces précédemment déterminés pour obtenir Eff_1 .

Etape 3 :

- Choisir un indice $j_2 \in \tau_2$ et déterminer toutes les solutions entières réalisables.

$x_2^q, q \in \{2, \dots, p_2\}$ alternatives à x_2^1 , lorsqu'elles existent sur l'arête E_{j_2} ,

Lire les nouveaux r-uplets ainsi obtenus. Augmenter l'ensemble Eff_2 par les nouveaux r-uplets non dominés pour construire Eff_2 . Eliminer de Eff_2 tous les r-uplets dominés. Donc Eff_2 est l'ensemble potentiellement efficace à l'étape 3.

Tronquer l'arête E_{j_2} en appliquant la coupe

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

Et chercher la solution optimale entière dans la région tronquée courante pour passer à l'étape 4 de la procédure.

Etape k :

- Choisir $j_{k-1} \in \tau_{k-1}$ et explorer l'arête correspondante $E_{j_{k-1}}$ pour d'éventuelles solutions entières réalisables $x_{k-1}^q, q \in \{2, \dots, p_{k-1}\}$ alternatives à x_{k-1}^1 . Augmenter Eff_{k-2} par les nouveaux r-uplets non dominés pour construire Eff_{k-1} .
- L'arête $E_{j_{k-1}}$ est tronquée par la coupe

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

en utilisant la méthode duale du simplexe et d'éventuelles successives coupes de Gomory. La solution optimale entière obtenue sur la région tronquée SK sera soit une solution réalisable de (P_1) alternatives à x_{k-1}^1 ou la prochaine meilleure solution x_k^1 ou un point non entier. Ceci marque le début de l'étape k+1.

Chapitre 2 : Méthode multi-objectif linéaire en nombres entiers

Etape finale :

La procédure se termine quand le pivot n'est pas réalisable, indiquant que la région courante contient des solutions réalisables non entières et que tous les points efficaces ont été déterminés.

Exemple (Méthode Abbas&Moulai) :

Soit le problème suivant :

$$\text{Max } Z_1(x) = C^1 x = 2x_1$$

$$\text{Max } Z_2(x) = C^2 x = -3x_1 - 2x_2$$

$$(P_1) \begin{cases} 2x_1 + x_2 \leq 3 \\ 2x_1 + 3x_2 \leq 2 \\ x_1, x_2 \in \mathbb{N} \end{cases}$$

Résolution du problème (P1) :

$$\text{Max } Z_1(x) = C^1 x = 2x_1$$

$$(P_1) \begin{cases} 2x_1 + x_2 \leq 3 \\ 2x_1 + 3x_2 \leq 2 \\ x_i \in \mathbb{N}, i=\{1,2\} \end{cases}$$

Etape 1 :

B	x_b	x_1	x_2	x_3	x_4
x_1	1	1	1.5	0	0.5
x_3	1	0	-2	1	-1
2	$z_j^1 - c_j^1$	0	3	0	1
-6	$z_j^2 - c_j^2$	0	-2.5	0	-1.5

Tableau1

$$I_1 = \{1,2,3,4\}, N_1 = \{2,4\}, \tau_1 = \{j/j \in N_1, Z_{1,j}^1 - C_j^1 = 0\} = \emptyset$$

$$Z_1^1 = Z_1(x_1^1) = 2, Z_2(x_1^1) = -3$$

L'unique solution optimale $x_1^1 = (1,0)$ produit le premier r-uplet potentiellement efficace (1,0).
Par suite : $\text{eff}_0 = (2,-3)$.

Chapitre 2 : Méthode multi-objectif linéaire en nombres entiers

Etape2 :

$\tau_1 = \emptyset, x_1^1$ est l'unique solution .Ce point entier est tronqué en utilisant la coupe $x_2 + x_4 \geq 1$
En utilisant le duale du simplexe on obtient le tableau 2 suivant :

B	x_B	x_1	x_2	x_3	x_4	x_5	x_6
x_1	1	1	1,5	0	0,5	0	
x_3	1	0	-2	1	-1	0	
x_5	-1	0	-1	0	-1	1	
2	$Z_j^1 - C_j^1$	0	3	0	1	0	
-3	$Z_j^2 - C_j^2$	0	-2.5	0	-1.5	0	
x_1	0,5	1	1	0	0	0,5	
x_3	2	0	-1	1	0	-1	
x_4	1	0	1	0	1	-1	
1	$Z_j^1 - C_j^1$	0	2	0	0	1	
-1.5	$Z_j^2 - C_j^2$						
x_1	0,5	1	1	0	0	0,5	0
x_3	2	0	-1	1	0	-1	0
x_4	1	0	1	0	1	-1	0
x_6	-0,5	0	0	0	0	-0.5	1
1	$Z_j^1 - C_j^1$	0	2	0	0	1	0
-1.5	$Z_j^2 - C_j^2$	0	-1	0	0	-1.5	0
x_1	0	1	1	0	0	0	1
x_3	3	0	-1	1	0	0	-2
x_4	2	0	1	0	1	0	-2
x_5	1	0	0	0	0	1	-2
0	$Z_j^1 - C_j^1$	0	2	0	0	0	2
0	$Z_j^2 - C_j^2$	0	-1	0	0	0	-3

Tableau2

$x_2^1 = (0,0)$, produit le deuxième r-uplet .

Le vecteur (0,0) ne domine pas le vecteur (2,-3).

Donc : $\text{Eff}_1 = \{(0,0), (2,-3)\}$.

Etape3 :

$\tau_2 \neq \emptyset$, Ce point entier est tronqué en utilisant la coupe $x_2 + x_6 \geq 1$.

B	x_B	x_1	x_2	x_3	x_4	x_5	x_6	x_7
x_1	0	1	1	0	0	0	1	0
x_3	3	0	-1	1	0	0	-2	0
x_4	2	0	1	0	1	0	-2	0
x_5	1	0	0	0	0	1	-2	0
x_7	-1	0	-1	0	0	0	-1	1
		0	2	0	0	0	2	0
		0	3	0	0	0	-3	0
B	x_B	x_1	x_2	x_3	x_4	x_5	x_6	x_7
x_1	-1	1	0	0	0	0	0	1
x_3	4	0	0	1	0	0	-1	-1
x_4	1	0	0	0	1	0	-3	1
x_5	1	0	0	0	0	1	-2	0
x_2	1	0	1	0	0	0	1	-1
		0	0	0	0	0	0	2
		0	0	0	0	0	-2	-5

En utilisant le dual du simplexe, on obtient le tableau suivant :

Tableau3

On s'arrête car on ne peut pas pivoter

Chapitre 3

Programmation floue

3.1. Définition d'un sous ensemble flou :

Dans la théorie des ensembles classiques, il n'y a que deux situations acceptables pour un élément, appartenir ou ne pas appartenir à un sous-ensemble. Le mérite de Zadeh a été de tenter de sortir de cette logique booléenne en introduisant la notion d'appartenance pondérée : permettre des graduations dans l'appartenance d'un élément à un sous-ensemble, c'est-à-dire d'autoriser un élément à appartenir plus moins fortement à ce sous-ensemble.

Soit X un ensemble de référence et soit x un élément quelconque de X . Un sous ensemble flou \tilde{A} de X est défini comme l'ensemble des couples :

$$\tilde{A} = \{(x, \mu_{\tilde{A}}(x)), x \in X\}$$

Avec : $\mu_{\tilde{A}} : X \rightarrow [0,1]$

$\mu_{\tilde{A}}(x)$: associe à chaque point x de X un réel dans l'intervalle $[0,1]$;

$\mu_{\tilde{A}}(x)$ représente le degré d'appartenance de x à \tilde{A} .

Où :

- $\mu_{\tilde{A}}(x) = 0$ si x n'appartient pas à \tilde{A} ;
- $0 < \mu_{\tilde{A}}(x) < 1$ si x appartient partiellement à \tilde{A} ;
- $\mu_{\tilde{A}}(x) = 1$ si x appartient entièrement à \tilde{A} .

La fonction d'appartenance $\mu_{\tilde{A}}(x)$ inclut ou exclut donc à ses extrémités, tout élément x au sous-ensemble \tilde{A} , mais entre les valeurs extrêmes le degré d'appartenance varie à proportion de la proximité à l'ensemble.

Remarque :

On peut faire remarquer que si A est un sous-ensemble classique, la fonction d'appartenance qui lui est associée ne peut prendre que les valeurs extrêmes 0 et 1.

On a dans ce cas :

$$\mu_{\tilde{A}}(x) = \begin{cases} 0 & \text{si } x \notin A \\ 1 & \text{si } x \in A \end{cases}$$

3.2. Caractéristiques d'un sous-ensemble flou :[3]

Un sous-ensemble flou est complètement défini par la donnée de sa fonction d'appartenance. A partir d'une telle fonction, un certain nombre de caractéristiques du sous-ensemble flou peuvent être étudiées.

3.2.1. Support et Hauteur

- **Le support** d'un sous-ensemble flou de \tilde{A} de X , noté $Supp(\tilde{A})$, est l'ensemble de tous les éléments qui lui appartiennent au moins un petit peu.

Formellement:

$$supp(\tilde{A}) = \{ (x \in X / \mu_{\tilde{A}}(x) > 0) \}$$

- **La hauteur** du sous-ensemble flou \tilde{A} de X , notée $h(\tilde{A})$, est le plus fort degré avec lequel un élément de X appartient à \tilde{A} .

Formellement:

$$h(\tilde{A}) = \sup_{x \in X} \mu_{\tilde{A}}(x)$$

3.2.2. Noyau :

Un sous-ensemble flou \tilde{A} est normalisé si sa hauteur $h(\tilde{A}) = 1$. Le noyau d'un sous ensemble flou \tilde{A} de X , noté $Noy(\tilde{A})$, est l'ensemble de tous les éléments qui lui appartiennent totalement (avec un degré 1).

Formellement :

$$\text{Noy}(\tilde{A}) = \{x \in X / \mu_{\tilde{A}}(x) = 1\}.$$

3.2.3. Cardinalité :

La cardinalité d'un sous-ensemble flou \tilde{A} de X , noté $|\tilde{A}|$, est le nombre d'éléments appartenant à \tilde{A} pondéré par leur degré d'appartenance.

Formellement, pour \tilde{A} fini :

$$|\tilde{A}| = \sum_{x \in X} \mu_{\tilde{A}}(x)$$

Si A est sous-ensemble ordinaire de X , sa cardinalité est le nombre d'éléments qui le composent, selon la définition classique.

3.2.4. α -coupe : [3]

Le sous-ensemble ordinaire A_α de X associé à \tilde{A} pour le seuil $\alpha \in [0,1]$ est l'ensemble des éléments qui appartiennent à X avec un degré au moins égal à α . On dit que α est l' α -coupe de \tilde{A} .

Formellement :

$$A_\alpha = \{x \in X / \mu_{\tilde{A}}(x) \geq \alpha\}$$

Et A_α est un sous-ensemble ordinaire de fonction caractéristique :

$$A_\alpha(x) = \begin{cases} 1 & \text{si } \mu_{\tilde{A}}(x) \geq \alpha \\ 0 & \text{sinon} \end{cases}$$

3.3. Opérations sur les sous-ensembles flous :[3]

Etant donné que le concept de sous-ensemble flou peut-être vu comme une généralisation du concept d'ensemble classique, on est conduit à introduire des opérations sur les sous-ensembles flous qui sont équivalentes aux opérations classiques de la théorie des ensembles lorsqu'on a affaire à des fonctions d'appartenance à valeurs 0 ou 1. On présente ici, les opérations les plus couramment utilisées :

3.3.1. Egalité :

Deux sous-ensembles flous \tilde{A} et \tilde{B} de X sont égaux, si leurs fonctions d'appartenance prennent la même valeur pour tous les élément x de X .

Formellement :

$$\tilde{A} = \tilde{B} \quad \text{si et seulement si : } \forall x \in X, \mu_{\tilde{A}}(x) = \mu_{\tilde{B}}(x).$$

3.3.2. Complémentaire :

Le complémentaire d'un sous-ensemble flou \tilde{A} de X noté $\overline{\tilde{A}}$ est défini par :

$$x \in X, \quad \mu_{\overline{\tilde{A}}}(x) = 1 - \mu_{\tilde{A}}(x).$$

3.3.3. Inclusion :

Soit \tilde{A} et \tilde{B} deux sous-ensembles flous de X . Si pour n'importe quel élément x de X , x appartient toujours moins à A qu'à B , alors on dit que \tilde{A} est inclus dans

$$\tilde{B} (\tilde{A} \subseteq \tilde{B}).$$

Formellement :

$$\tilde{A} \subseteq \tilde{B} \quad \text{si et seulement si : } \forall x \in X, \mu_{\tilde{A}}(x) \leq \mu_{\tilde{B}}(x).$$

3.3.4. Union:

L'union de deux sous-ensembles flous \tilde{A} et \tilde{B} de X est le sous-ensemble flou constitué des éléments de X affectés du plus grand des degrés avec lesquels ils appartiennent à \tilde{A} et \tilde{B} .

Formellement :

$$\tilde{A} \cup \tilde{B} \text{ est donné par } \mu_{\tilde{A} \cup \tilde{B}}(x) = \max(\mu_{\tilde{A}}(x), \mu_{\tilde{B}}(x))$$

3.3.5. Intersection :

L'intersection de deux sous-ensembles flous \tilde{A} et \tilde{B} de X est le sous-ensemble flou constitué des éléments de X affectés du plus petit des degrés avec lesquels ils appartiennent à \tilde{A} et \tilde{B} .

Formellement :

$$\tilde{A} \cap \tilde{B} \text{ est donné par } \mu_{\tilde{A} \cap \tilde{B}}(x) = \min(\mu_{\tilde{A}}(x), \mu_{\tilde{B}}(x))$$

3.4. Propriétés de l'union et de l'intersection :[3]

Comme pour les ensembles classiques, toutes les propriétés de treillis distributif et les relations de Morgan restent valables, ainsi que l'idempotence.

- Commutativité : $\tilde{A} \cup \tilde{B} = \tilde{B} \cup \tilde{A}$; $\tilde{A} \cap \tilde{B} = \tilde{B} \cap \tilde{A}$
- Associativité : $\tilde{A} \cup (\tilde{B} \cup \tilde{C}) = (\tilde{A} \cup \tilde{B}) \cup \tilde{C}$; $\tilde{A} \cap (\tilde{B} \cap \tilde{C}) = (\tilde{A} \cap \tilde{B}) \cap \tilde{C}$
- Idempotence : $\tilde{A} \cup \tilde{A} = \tilde{A}$; $\tilde{A} \cap \tilde{A} = \tilde{A}$
- Distributivité : $\tilde{A} \cup (\tilde{B} \cap \tilde{C}) = (\tilde{A} \cup \tilde{B}) \cap (\tilde{A} \cup \tilde{C})$; $\tilde{A} \cap (\tilde{B} \cup \tilde{C}) = (\tilde{A} \cap \tilde{B}) \cup (\tilde{A} \cap \tilde{C})$
- Les relations de Morgan : $\overline{\tilde{A} \cup \tilde{B}} = \overline{\tilde{A}} \cap \overline{\tilde{B}}$; $\overline{\tilde{A} \cap \tilde{B}} = \overline{\tilde{A}} \cup \overline{\tilde{B}}$
- Les lois d'absorption : $\tilde{A} \cup (\tilde{A} \cap \tilde{B}) = \tilde{A}$; $\tilde{A} \cap (\tilde{A} \cup \tilde{B}) = \tilde{A}$
- $\tilde{A} \cap \emptyset = \emptyset$; $\tilde{A} \cup X = X$
- Identité : $\tilde{A} \cup \emptyset = \tilde{A}$; $\tilde{A} \cap X = \tilde{A}$
- Cardinalité : $|\tilde{A}| + |\tilde{B}| = |\tilde{A} \cap \tilde{B}| + |\tilde{A} \cup \tilde{B}|$
- Formule d'équivalence : $(\overline{\tilde{A} \cup \tilde{B}}) \cap (\tilde{A} \cup \tilde{B}) = (\overline{\tilde{A}} \cap \overline{\tilde{B}}) \cup (\tilde{A} \cap \tilde{B})$
- Formule de la différence symétrique : $(\overline{\tilde{A} \cap \tilde{B}}) \cup (\tilde{A} \cap \tilde{B}) = (\overline{\tilde{A}} \cup \overline{\tilde{B}}) \cap (\tilde{A} \cup \tilde{B})$.

3.5. Nombres flous :[5]

Un nombre flou \tilde{A} est un sous ensemble flou respectant les propriétés suivantes :

1- Il est normalisé $\Rightarrow h(\tilde{A}) = 1$.

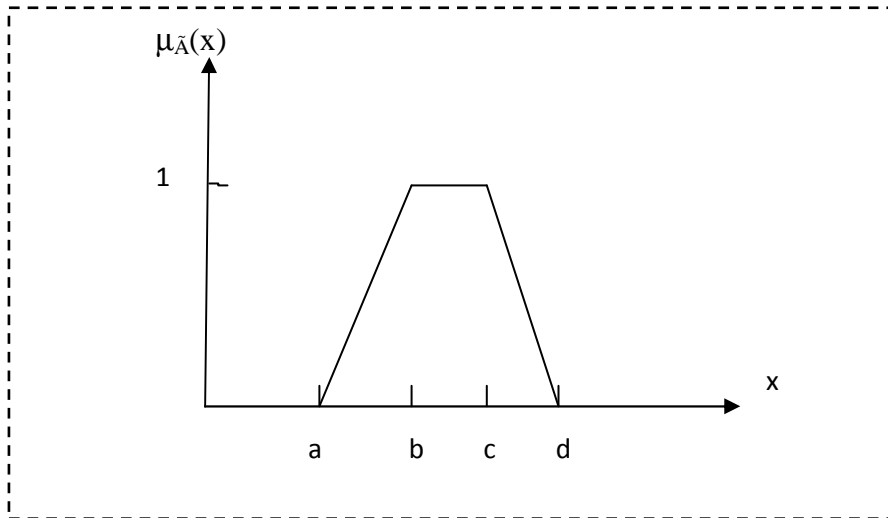
2- Son noyau est unique. C'est-à-dire il existe un seul élément x de \mathbb{R} pour lequel le degré d'appartenance est égal à 1 $\Rightarrow \{x \in \mathbb{R} / \mu_{\tilde{A}}(x) = 1\}$

3- Il est convexe c'est-à-dire :

- \tilde{A} est dit convexe si quelque soit x_1 et $x_2 \in X$ et $\lambda \in [0,1]$, on a $\mu_{\tilde{A}}(\lambda x_1 + (1-\lambda)x_2) \geq \min(\mu_{\tilde{A}}(x_1), \mu_{\tilde{A}}(x_2))$
- Autrement dit \tilde{A} est convexe si $\mu_{\tilde{A}}(x)$ est quasi concave.

3.5.1. Exemple de nombres flous :

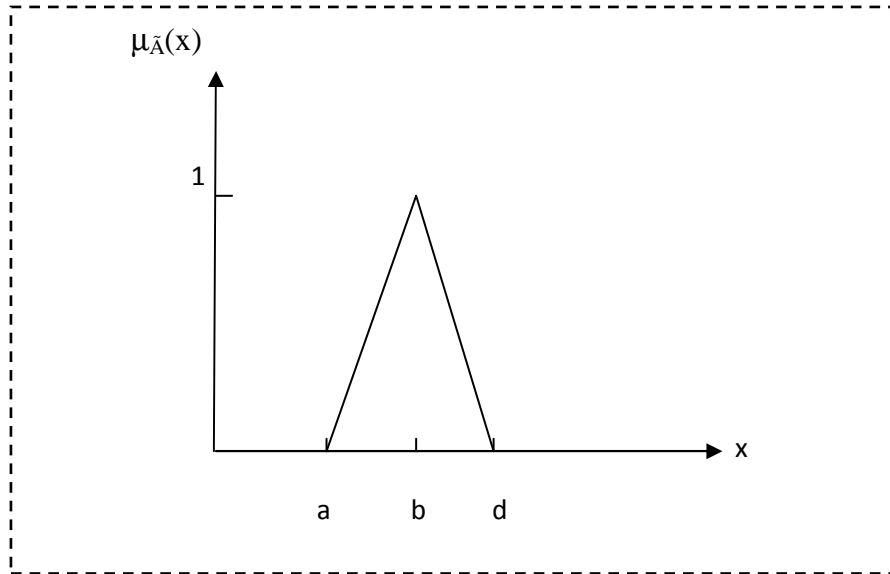
- Ensemble flou trapézoïdal : $\tilde{A} = (a,b,c,d)$ où $c,d > 0$ et $a,b \in \mathbb{R}$.



Un nombre flou est trapézoïdal si sa fonction d'appartenance est :

$$\mu_{\tilde{A}}(x) = \begin{cases} \frac{x-a}{b-a} & \text{si } a \leq x \leq b \\ \frac{x-d}{c-d} & \text{si } c \leq x \leq d \\ 1 & \text{si } b \leq x \leq c \end{cases}$$

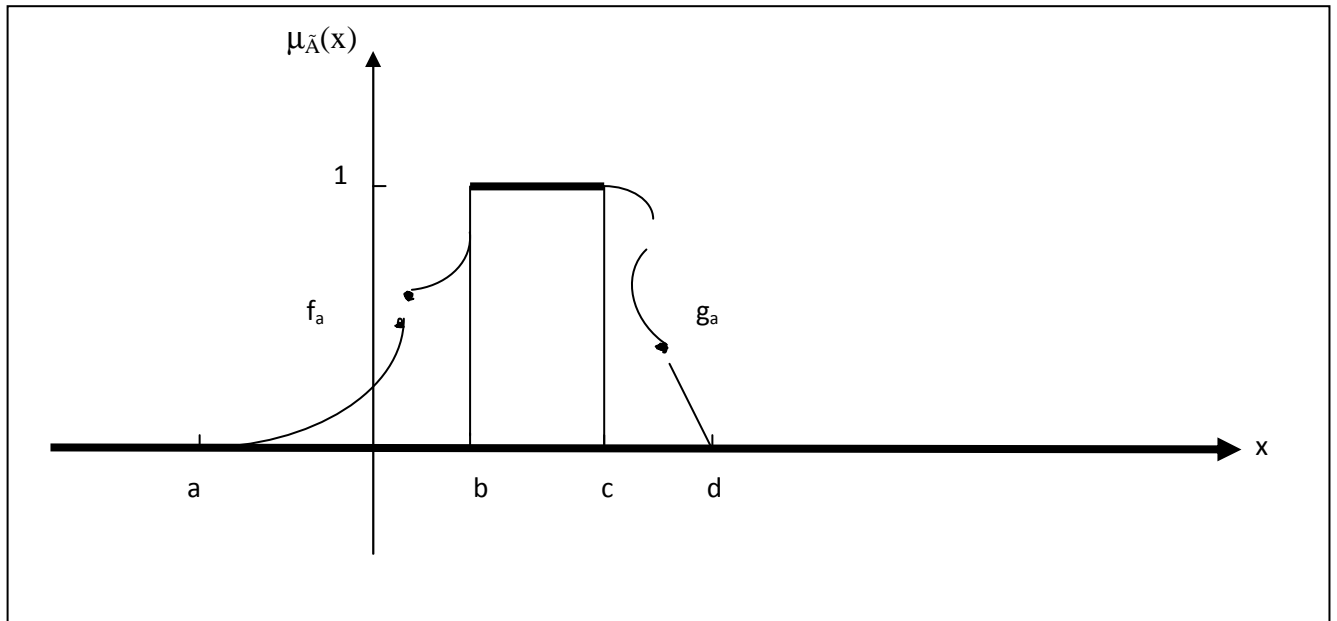
- Nombre flou triangulaire : $\tilde{A} = (a,b,d)$



Un nombre flou est triangulaire si et seulement si son noyau est réduit à un seul point :

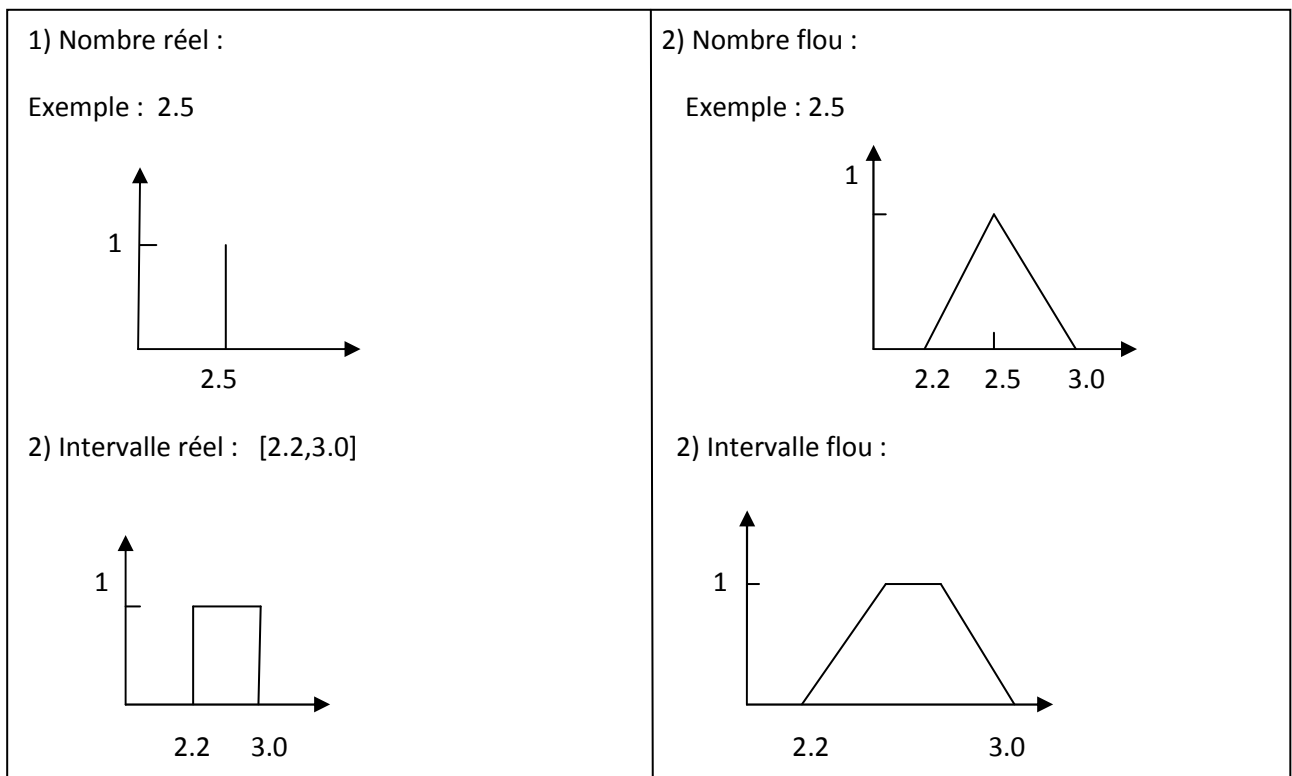
$$\mu_{\tilde{A}}(x) = \begin{cases} 0 & \text{si } x \leq a \\ x - a & \text{si } x \in]a, b] \\ d - x & \text{si } x \in]b, d] \\ 0 & \text{si } x \geq d \end{cases}$$

- Cas général :



$$\mu_{\tilde{A}}(x) = \begin{cases} f_a(x) & \text{si } x \in [a, b] \\ 1 & \text{si } x \in [b, c] \\ g_a(x) & \text{si } x \in [c, d] \\ 0 & \text{ailleurs} \end{cases}$$

3.6. Comparaisons (entre un nombre flou et un nombre réel) :



3.7 Nombres flous de type (LR) : (G\D) (Gauche\ Droite)[4]

Un nombre flou est de type (LR) s'il existe deux fonctions L et R et des réels : m, n, α, β avec $\alpha > 0$ et $\beta > 0$ tel que :

$$\mu_{\tilde{A}}(x) = \begin{cases} L\left(\frac{x-m}{\alpha}\right) & \text{si } x \leq m \\ R\left(\frac{x-n}{\beta}\right) & \text{si } x \geq n \\ 1 & \text{sinon} \end{cases}$$

L, R : fonctions de références.

Tel que : $L(x) = L(-x)$. $R(x) = R(-x)$. $R(0) = L(0) = 1$. L, R non croissantes sur $[0, +\infty[$.

m : la moyenne à gauche.

n : la moyenne à droite.

$\alpha > 0$: L'écart à gauche de m.

$\beta > 0$: L'écart à droite de n.

On note : $\tilde{A} = (m, n, \alpha, \beta)_{L,R}$

3.7.1. Définition : on dit qu'un nombre flou de type L.R est positif si et seulement si :

$$m - \alpha \geq 0$$

3.7.2. Définition : on dit qu'un nombre flou de type L.R est négatif si et seulement si :

$$n + \beta \geq 0$$

3.7.3. Operations sur les nombres flous de type (LR) :

Soient : $\tilde{A}_1 = (m_1, n_1, \alpha_1, \beta_1)_{L,R}$

$$\tilde{A}_2 = (m_2, n_2, \alpha_2, \beta_2)_{L,R}$$

$$1) \tilde{A}_1 \oplus \tilde{A}_2 = (m_1 + m_2, n_1 + n_2, \alpha_1 + \alpha_2, \beta_1 + \beta_2)$$

$$2) \tilde{A}_1 \ominus \tilde{A}_2 = (m_1 - n_2, n_1 + m_2, \alpha_1 + \beta_2, \alpha_2 + \beta_1)$$

$$3) \text{ Si : } \tilde{A}_1 \text{ et } \tilde{A}_2 \text{ sont positifs } \Rightarrow$$

$$\tilde{A}_1 \otimes \tilde{A}_2 = (m_1 m_2, n_1 n_2, m_1 \alpha_2 + \alpha_1 m_2 - \alpha_1 \alpha_2, n_1 \beta_2 + \beta_1 n_2 - \beta_1 \beta_2)$$

4) Si : \tilde{A}_1 et \tilde{A}_2 sont négatifs : \Rightarrow

$$\tilde{A}_1 \otimes \tilde{A}_2 = (n_1 n_2, m_1 m_2, -n_1 \beta_2 - \beta_1 n_2 - \beta_1 \beta_2, -m_1 \alpha_2 - \alpha_1 m_2 - \alpha_1 \alpha_2)$$

5) Si : \tilde{A}_1 est négatif et \tilde{A}_2 positif: \Rightarrow

$$\tilde{A}_1 \otimes \tilde{A}_2 = (m_1 n_2, n_1 m_2, n_2 \alpha_1 - \beta_2 m_1 + \alpha_1 \beta_2, m_2 \beta_1 - \alpha_2 n_1 - \beta_1 \alpha_2)$$

6) Si : \tilde{A}_1 est positif et \tilde{A}_2 négatif: \Rightarrow

$$\tilde{A}_1 \otimes \tilde{A}_2 = (n_1 m_2, m_1 n_2, n_1 \alpha_2 - \beta_1 m_2 + \beta_1 \alpha_2, m_1 \alpha_2 - \alpha_1 n_2 - n_2 \beta_2)$$

$$7) \lambda \tilde{A}_1 = \begin{cases} (\lambda m_1, \lambda n_1, \lambda \alpha_1, \lambda \beta_1) & \text{si } \lambda \geq 0 \\ (\lambda m_1, \lambda n_1, -\lambda \alpha_1, -\lambda \beta_1) & \text{si } \lambda < 0 \end{cases}$$

3.8. Programmation linéaire floue :

On distingue deux cas :

- 1- Le cas où les inégalités (ou égalités) sont relaxées (floues). (programmation flexible)
- 2- Le cas où les données imprécises sont représentées par des ensembles flous. (programmation robuste)

3.8.1. Programmation flexible : [6]

Un problème dont l'objectif et les contraintes sont vaguement définies.

3.8.1.1. Résolution d'un programme linéaire flexible :

Considérons le programme linéaire flexible suivant :

$$(P_{fl}) \quad \begin{cases} \overline{max} \quad c'x \\ A_i x \tilde{\theta} b_i, \quad i=1, 2, \dots, m \\ x \geq 0 \end{cases}$$

Où: $\tilde{\theta} \in \{\gtrsim, \lesssim, \cong\}$.

et : \gtrsim, \lesssim, \cong sont les versions flexibles (floues) de : $\geq, \leq, =$ respectivement.

La notation \sim désigne le fait que les objectifs et les contraintes ne sont pas précis.

Selon Zimmermann le programme (P_{fl}) devient :

$$(P'_{fl}) \quad \begin{cases} c'x \lesssim z \\ A_i x \tilde{\theta} b_i, \quad i=1, 2, \dots, m \\ x \geq 0 \end{cases}$$

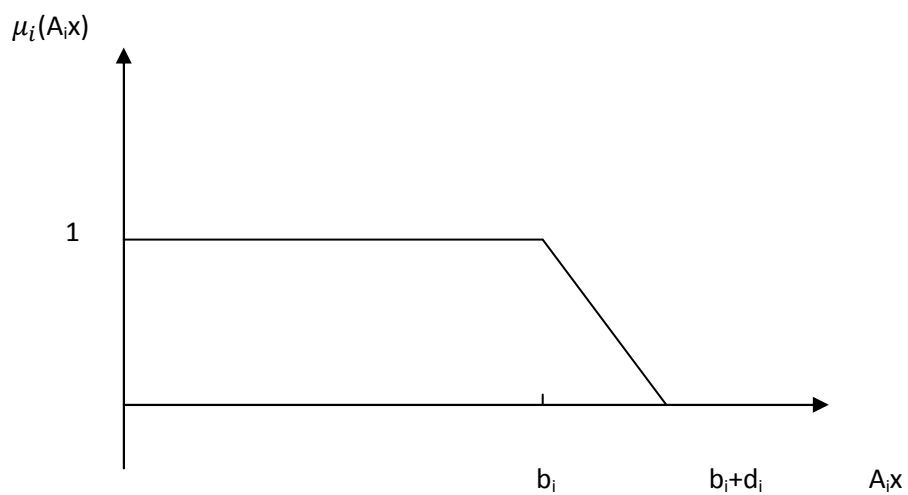
L'objectif flou et les contraintes floues sont représentées par les ensembles flous,

$U_i, i=1, 2, \dots, m$ dont les fonctions d'appartenance sont respectivement : $\mu_i : i=1, 2, \dots, m$ est définie selon que : $\tilde{\theta}$ est \gtrsim, \lesssim, \cong : comme suit :

$\tilde{\theta}$ est \lesseqgtr :

$$\mu_i(x) = \mu_i(A_i x) = \begin{cases} 1 & \text{si } A_i x \leq b_i, \quad i=1,2,\dots,m \text{ (satisfaite)} \\ \theta \in]0,1] & \text{si les contraintes sont faiblement violées} \\ 0 & \text{si } A_i x > b_i + d_i, \quad i=1,2,\dots,m \end{cases}$$

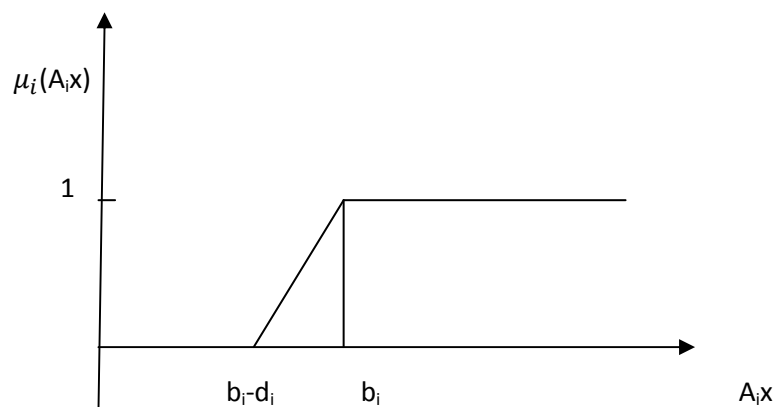
$$\mu_i(x) = \mu_i(A_i x) = \begin{cases} 1 & \text{si } A_i x \leq b_i, \quad i=1,2,\dots,m \\ 1 - \frac{A_i x - b_i}{d_i} & \text{si } b_i < A_i x \leq b_i + d_i, \quad i=1,2,\dots,m \\ 0 & \text{si } A_i x > b_i + d_i, \quad i=1,2,\dots,m \end{cases}$$



$\tilde{\theta}$ est \cong :

$$\mu_i(x) = \mu_i(A_i x) = \begin{cases} 1 & \text{si } A_i x \geq b_i, \quad i = 1, 2, \dots, m \text{ (satisfaite)} \\ \theta \in]0, 1] & \text{si les contraintes sont faiblement violées .} \\ 0 & \text{si } A_i x < b_i - d_i, \quad i = 1, 2, \dots, m \end{cases}$$

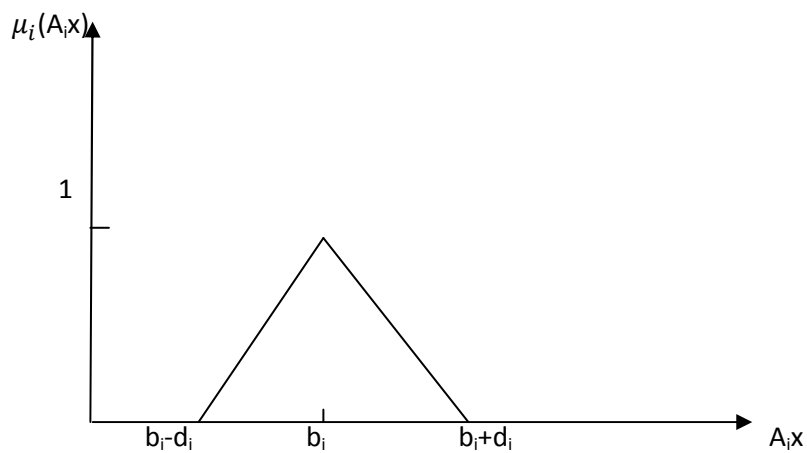
$$\mu_i(x) = \mu_i(A_i x) = \begin{cases} 1 & \text{si } A_i x \geq b_i, \quad i = 1, 2, \dots, m \\ 1 - \frac{b_i - A_i x}{d_i} & \text{si } b_i - d_i \leq A_i x < b_i, \quad i = 1, 2, \dots, m \\ 0 & \text{si } A_i x < b_i - d_i, \quad i = 1, 2, \dots, m \end{cases}$$



$\tilde{\theta}$ est \cong :

$$\mu_i(x) = \mu_i(A_i x) = \begin{cases} 1 & \text{si } A_i x = b_i, i = 1, 2, \dots, m \text{ (satisfaite)} \\ \theta \in]0, 1] & \text{si les contraintes sont faiblement violées .} \\ 0 & \text{si } A_i x > b_i + d_i, A_i x < b_i - d_i, i = 1, 2, \dots, m \end{cases}$$

$$\mu_i(x) = \mu_i(A_i x) = \begin{cases} 1 & \text{si } A_i x = b_i, i = 1, 2, \dots, m \\ 1 - \frac{A_i x - b_i}{d_i} & \text{si } b_i < A_i x \leq b_i + d_i, i = 1, 2, \dots, m \\ 1 - \frac{b_i - A_i x}{d_i} & \text{si } b_i - d_i \leq A_i x < b_i, i = 1, 2, \dots, m \\ 0 & \text{si } A_i x < b_i - d_i, A_i x > b_i + d_i, i = 1, 2, \dots, m \end{cases}$$



Le problème (P_{fl}) devient :

$$(P''_{fl}) \left\{ \begin{array}{l} \text{Max } \lambda \\ \lambda \leq \mu_i(x), i=0 \dots m \\ 0 \leq \lambda \leq 1 \\ x \geq 0 \end{array} \right.$$

Exemple 1: On considère le programme (P) flexible suivant :

$$\overline{\max} \quad Z(x) = 2x_1 - x_2$$

$$\text{s.c} \left\{ \begin{array}{l} -x_1 + 2x_2 \geq -5 \\ 3x_1 + x_2 \leq 1 \\ x_1 \geq 0, x_2 \geq 0 \end{array} \right.$$

1) On détermine les fonctions d'appartenance de l'objectif et de la 1^{ère} contraintes sachant que l'objectif excède la valeur $Z_0 = 1$ et que les écarts de tolérance de l'objectif et des contraintes sont respectivement $d_0=1, d_1=2, d_2=1$:

Transformation de l'objectif :

$$Z(x) = 2x_1 - x_2 = 1, \quad Z_0 = 1,$$

$$\mu_0(x) = \left\{ \begin{array}{ll} 1 & \text{si } 2x_1 - x_2 = 1 \\ 1 - \frac{1 - Z(x)}{1} & \text{si } 0 \leq 2x_1 - x_2 < 1 \\ 0 & \text{si } 2x_1 - x_2 < 0. \end{array} \right.$$

$$= \left\{ \begin{array}{ll} 1 & \text{si } 2x_1 - x_2 = 1 \\ \frac{Z(x)}{1} & \text{si } 0 \leq 2x_1 - x_2 < 1 \\ 0 & \text{si } 2x_1 - x_2 < 0. \end{array} \right.$$

Transformation de la 1^{ère} contrainte :

$$\mu_1(x) = \begin{cases} 1 & \text{si } -x_1 + 2x_2 \geq -5 \\ 1 - \frac{-5 + x_1 - 2x_2}{2} & \text{si } -7 \leq x_1 - 2x_2 \leq -5 \\ 0 & \text{si } -x_1 + 2x_2 < -7 \end{cases}$$

$$\mu_1(x) = \begin{cases} 1 & \text{si } x_1 - 2x_2 \geq -5 \\ \frac{7 - x_1 + 2x_2}{2} & \text{si } -7 \leq -x_1 + 2x_2 \leq -5 \\ 0 & \text{si } -x_1 + 2x_2 < -7 \end{cases}$$

Transformation de la 2^{ème} contrainte :

$$\mu_2(x) = \begin{cases} 1 & \text{si } 3x_1 + x_2 \leq 1 \\ 1 - \frac{3x_1 + x_2 - 1}{1} & \text{si } 1 \leq 3x_1 + x_2 \leq 2 \\ 0 & \text{si } 3x_1 + x_2 > 2 \end{cases}$$

$$\mu_2(x) = \begin{cases} 1 & \text{si } 3x_1 + x_2 \leq 1 \\ \frac{-3x_1 - x_2 + 2}{1} & \text{si } 1 \leq 3x_1 + x_2 \leq 2 \\ 0 & \text{si } 3x_1 + x_2 > 2 \end{cases}$$

2) Le problème déterministe (DP) associé à (P) est :

$$\left\{ \begin{array}{l} \text{Max } \lambda \\ \lambda \leq \frac{Z(x)}{1} \\ \lambda \leq \frac{7-x_1+2x_2}{2} \\ \lambda \leq \frac{-3x_1-x_2+2}{1} \\ 0 < \lambda \leq 1 \end{array} \right.$$

=>

$$\left\{ \begin{array}{l} \text{Max } \lambda \\ \lambda - 2x_1 + x_2 \leq 0 \\ 2\lambda + x_1 - 2x_2 \leq 7 \\ \lambda + 3x_1 + x_2 \leq 2 \\ x_1 \geq 0, x_2 \geq 0, 0 \leq \lambda \leq 1 \end{array} \right. \Rightarrow \left\{ \begin{array}{l} \text{Max } \lambda \\ \lambda - 2x_1 + x_2 + x_3 = 0 \\ 2\lambda + x_1 - 2x_2 + x_4 = 7 \\ \lambda + 3x_1 + x_2 + x_5 = 2 \\ \lambda + x_6 = 1 \\ x_j \geq 0, j \geq 1, 7, 0 \leq \lambda \leq 1. \end{array} \right.$$

Chapitre 3 : Programmation floue

Résolution du problème (DP) par la méthode de simplexe :

C			0	0	0	0	0	0	1	θ
	Base	b	x_1	x_2	x_3	x_4	x_5	x_6	λ	
0	x_3	0	-2	1	1	0	0	0	(1) \Rightarrow	0
0	x_4	7	1	-2	0	1	0	0	2	7/2
0	x_5	2	3	1	0	0	1	0	1	2/1
0	x_6	1	0	0	0	0	0	1	1	0
		Δ_j	0	0	0	0	0	0	-1 \uparrow	
1	λ	0	-2	1	1	0	0	0	1	/
0	x_4	7	5	-4	-2	1	0	0	0	7/5
0	x_5	2	5	0	-1	0	1	0	0	2/5 \rightarrow
0	x_6	1	2	-1	-1	0	0	1	0	1/2
		Δ_j	-2 \uparrow	1	1	0	0	0	0	
1	λ	4/5	0	1	3/5	0	2/5	0	1	
0	x_4	5	0	-4	-1	1	-1	0	0	
0	x_1	2/5	1	0	-1/5	0	1/5	0	0	
0	x_6	1/5	0	-1	-3/5	0	-2/5	1	0	
		Δ_j	0	1	3/5	0	2/5	0	0	

$(x_1, x_2, \lambda) = (2/5, 0, 4/5)$ est une solution optimale de (DP).

3.8.2. Programmation robuste : [4]

3.8.2.1. Résolution d'un programme robuste :

Un programme robuste est un programme de la forme :

$$(p_k) \quad \begin{cases} \max Cx \\ Sc \\ x_1 \odot \tilde{A}_1 \oplus x_2 \odot \tilde{A}_2 \oplus \dots \oplus \dots x_n \odot \tilde{A}_n [\subset] \tilde{b} \\ x_j \geq 0, \quad j=1,2,\dots,n \end{cases}$$

Où $\tilde{A}_j : j=1,2,\dots,n$ sous ensembles flous de \mathbb{R} .

\oplus : Addition des ensembles flous.

$[\subset]$: Inclusion entre ensembles flous.

L'ensemble de contraintes de p_k sont représenté par :

$$E = \{x \in \mathbb{R}^n / x_1 \odot \tilde{A}_1 \oplus x_2 \odot \tilde{A}_2 \oplus \dots \oplus \dots x_n \odot \tilde{A}_n [\subset] \tilde{b}, x_j \geq 0\}.$$

Théorème 1:

$x^0 \in E$ est optimal pour P_k si et seulement si : $x^0 \in E$ est optimal pour le programme suivant :

$$(P_k) \quad \begin{cases} \max Cx \\ x_1 \tilde{A}_1^\alpha + x_2 \tilde{A}_2^\alpha + \dots + x_n \tilde{A}_n^\alpha \subset \tilde{b}^\alpha \quad \forall \alpha \in [0,1] \\ x_j \geq 0, j = 1,2, \dots n. \end{cases}$$

Preuve : [4]

$$\Rightarrow \text{(CN)} \quad x_0 \in E \text{ alors } x_1^0 \tilde{A}_1 + x_2^0 \tilde{A}_2 + \dots + x_n^0 \tilde{A}_n \subseteq \tilde{b}$$

$$(x_1^0 \tilde{A}_1 + x_2^0 \tilde{A}_2 + \dots + x_n^0 \tilde{A}_n)^\alpha \subseteq \tilde{b}^\alpha \quad \forall \alpha \in [0,1]$$

$$(x_1^0 \tilde{A}_1)^\alpha + (x_2^0 \tilde{A}_2)^\alpha + \dots + (x_n^0 \tilde{A}_n)^\alpha \subseteq \tilde{b}^\alpha \quad \forall \alpha \in [0,1]$$

$$x_1^0 \tilde{A}_1^\alpha + x_2^0 \tilde{A}_2^\alpha + \dots + x_n^0 \tilde{A}_n^\alpha \subseteq \tilde{b}^\alpha \quad \forall \alpha \in [0,1]$$

$$\Leftarrow \text{(CS)} \quad \text{Soit } x_1^0 \tilde{A}_1^\alpha + x_2^0 \tilde{A}_2^\alpha + \dots + x_n^0 \tilde{A}_n^\alpha \subseteq \tilde{b}^\alpha \quad (*)$$

On pose : $x^0 = (x_1^0, x_2^0, \dots, x_n^0)$

$$\tilde{A}_i^\alpha = (\tilde{A}_1^\alpha, \tilde{A}_2^\alpha, \dots, \tilde{A}_n^\alpha)$$

$$\tilde{A} = (\tilde{A}_1, \tilde{A}_2, \dots, \tilde{A}_n)$$

$$(*) \Rightarrow x^0 \tilde{A}^\alpha \subseteq \tilde{b}^\alpha \quad \forall \alpha \in [0,1]$$

$$\alpha \in [0,1] \quad \text{alors} \quad \alpha x^0 \tilde{A}^\alpha \subseteq \tilde{b}^\alpha$$

$$\cup \alpha x^0 \tilde{A}^\alpha \subseteq \cup \alpha \tilde{b}^\alpha \quad \alpha \in [0,1]$$

$$x^0 \cup \alpha \tilde{A}^\alpha \subseteq \cup \alpha \tilde{b}^\alpha \quad \alpha \in [0,1]$$

D'après le théorème de ZADEH

$$(*) \text{ S'écrit } x^0 \tilde{A} \subseteq \tilde{b}$$

$$\text{D'où } x_1^0 \tilde{A}_1 + x_2^0 \tilde{A}_2 + \dots + x_n^0 \tilde{A}_n \subseteq \tilde{b}^\alpha$$

3.8.2.2. Cas de nombres flous :

Si les composantes \tilde{a}_{ij} de \tilde{A} et \tilde{b}_i de \tilde{b} sont des nombres flous dont les ensembles de niveau sont des intervalles compacts.

$$\text{Comme :} \quad \tilde{a}_{ij}^\alpha = [\underline{\tilde{a}}_{ij}^\alpha, \overline{\tilde{a}}_{ij}^\alpha]$$

$$\tilde{b}_i^\alpha = [\underline{\tilde{b}}_i^\alpha, \overline{\tilde{b}}_i^\alpha]$$

Avec :

$$\left\{ \begin{array}{l} \underline{\tilde{a}}_{ij}^\alpha = \inf \{x \in R / \mu_{\tilde{a}_{ij}}(x) \geq \alpha\}. \\ \overline{\tilde{a}}_{ij}^\alpha = \sup \{x \in R / \mu_{\tilde{a}_{ij}}(x) \geq \alpha\}. \\ \underline{\tilde{b}}_i^\alpha = \inf \{x \in R / \mu_{\tilde{b}_i}(x) \geq \alpha\}. \\ \overline{\tilde{b}}_i^\alpha = \sup \{x \in R / \mu_{\tilde{b}_i}(x) \geq \alpha\}. \end{array} \right.$$

En utilisant les opérations élémentaires de l'arithmétique des intervalles.

(P_k) Devient :

$$(P_k)'', \quad \left\{ \begin{array}{l} \max Cx \\ [\underline{\tilde{a}}_{i1}^\alpha, \overline{\tilde{a}}_{i1}^\alpha]x_1 + [\underline{\tilde{a}}_{i2}^\alpha, \overline{\tilde{a}}_{i2}^\alpha]x_2 + \dots + [\underline{\tilde{a}}_{in}^\alpha, \overline{\tilde{a}}_{in}^\alpha]x_n \in [\underline{\tilde{b}}_i^\alpha, \overline{\tilde{b}}_i^\alpha] \\ x_j \geq 0, j = 1, 2, \dots, n. \end{array} \right.$$

$$\Rightarrow \quad (P_k)'' \quad \left\{ \begin{array}{l} \underline{a}_{i1}^\alpha x_1 + \underline{a}_{i2}^\alpha x_2 + \dots + \underline{a}_{in}^\alpha x_n \geq \underline{b}_i^\alpha, i = 1, \dots, m. \\ \overline{a}_{i1}^\alpha x_1 + \overline{a}_{i2}^\alpha x_2 + \dots + \overline{a}_{in}^\alpha x_n \leq \overline{b}_i^\alpha, i = 1, \dots, m. \\ x_j \geq 0, j = 1, 2, \dots, n \end{array} \right.$$

3.8.2.3. Cas des nombres flous de types L-R : [4]

$$\begin{array}{l} \text{On a: } \tilde{a}_{ij} = (m_{ij}, n_{ij}, \alpha_{ij}, \beta_{ij})_{L-R} \\ \tilde{b}_i = (m_i, n_i, \alpha_i, \beta_i)_{L-R} \quad ; \quad \text{où} \end{array} \quad \left\{ \begin{array}{l} \alpha_{ij} > 0 \\ \beta_{ij} > 0 \\ \alpha_i > 0 \\ \beta_i > 0 \end{array} \right.$$

(P_k) Deviendra :

$$\left\{ \begin{array}{l} \max Z = Cx \\ \text{Sc.} \\ \sum_{j=1}^n m_{ij} x_j = m_i \\ \sum_{j=1}^n n_{ij} x_j = n_i \\ \sum_{j=1}^n \alpha_{ij} x_j = \alpha_i \\ \sum_{j=1}^n \beta_{ij} x_j = \beta_i \\ x_j \geq 0 \quad j=1, 2, \dots, n. \end{array} \right.$$

Exemple (robuste) : En posant $\alpha = 0.5$ et en utilisant l'algorithme du simplexe, on trouve la solution optimale du problème (P) :

$$(P) \begin{cases} \text{Max } Z(x) = x_1 + 2x_2 \\ \text{Sc.} \\ \tilde{a}_1 x_1 + \tilde{a}_2 x_2 \subseteq \tilde{b} \\ x_2 \geq -1 \\ x_1 > 0, x_2 > 0 \end{cases}$$

Sachant que :

$$\tilde{a}_1 = \{ (1/0.7), (2/0.5), (3/0.6), (5/0.3) \}$$

$$\tilde{a}_2 = \{ (5/0.2), (3/0.9), (1/0.7), (2/0.6) \}$$

$$\tilde{b} = \{ (6/0.7), (1/0.2), (-2/0.8), (5/0.4) \}$$

On aura :

$$(P') \begin{cases} \text{Max } Z(x) = x_1 + 2x_2 \\ \text{Sc.} \\ [1,3]x_1 + [1,3]x_2 \subseteq [-2,6] \\ x_2 \geq -1 \\ x_1 > 0, x_2 > 0 \end{cases}$$

\Rightarrow

$$\begin{cases} \text{Max } Z(x) = x_1 + 2x_2 \\ x_1 + x_2 \geq -2 \\ 3x_1 + 3x_2 \leq 6 \\ x_2 \geq -1 \\ x_1 \geq 0, x_2 \geq 0 \end{cases} \Rightarrow \begin{cases} \text{Max } Z(x) = x_1 + 2x_2 \\ x_1 + x_2 - x_3 = -2 \\ 3x_1 + 3x_2 + x_4 = 6 \\ x_2 - x_5 = -1 \\ x_j \geq 0, j=1, \dots, n \end{cases}$$

Chapitre 3 : Programmation floue

$$\Rightarrow (P)' \left\{ \begin{array}{l} \text{Max } Z(x) = x_1 + 2x_2 \\ -x_1 - x_2 + x_3 = 2 \\ 3x_1 + 3x_2 + x_4 = 6 \\ -x_2 + x_5 = 1 \\ x_j \geq 0, j=1..n \end{array} \right.$$

Résolution par simplexe :

c			1	2	0	0	0	θ
	base	b	x_1	x_2	x_3	x_4	x_5	
0	x_3	2	-1	-1	1	0	0	/
0	x_4	6	3	3	0	1	0	2
0	x_5	1	0	-1	0	0	1	/
		Δ_j	-1	-2	0	0	0	
1	x_3	4	0	0	1	1/3	0	
0	x_2	2	1	1	0	1/3	0	
0	x_5	3	1	0	0	1/3	1	
		Δ_j	1	0	0	2/3	0	

La solution optimale du problème (P) est : $(x_1, x_2) = (0, 2)$

Exemple (multi-objectif) :

Résoudre avec le duale du simplexe le problème multi-objectif suivant :

$$\left\{ \begin{array}{l} \widetilde{\max} \ 2x_1 \\ \widetilde{\max} \ 3x_1+x_2 \\ \text{SC} \\ x_1+x_2 \leq 2 \\ 3x_1+2x_2 \leq 4 \\ x_1 \geq 0, x_2 \geq 0 \end{array} \right. \quad (P_1)$$

Sachant que les objectifs ne peuvent excéder les valeurs $Z_0=1, Z_1=2$.

Et les écart de tolérance sont : $d_0=2, d_1=3$.

$(P_1) \Rightarrow$

$$\left\{ \begin{array}{l} 2x_1 \leq 1 \\ 3x_1+x_2 \leq 2 \\ x_1+x_2 \leq 2 \\ 3x_1+2x_2 \leq 4 \\ x_1 \geq 0, x_2 \geq 0 \end{array} \right.$$

Transformation du 1^{er} objectif :

$$\left\{ \begin{array}{ll} 1 & \text{si } 2x_1 \leq 1 \\ \mu_0(x) = \frac{3}{2} - x_1 & \text{si } 1 \leq 2x_1 \leq 3 \\ 0 & \text{si } 2x_1 > 3 \end{array} \right.$$

Transformation du 2^{eme} objectif :

$$\mu_1(x) = \begin{cases} 1 & \text{si } 3X_1 + x_2 \leq 2 \\ \frac{1-3x_1-x_2}{3} & \text{si } 2 \leq 3X_1 + x_2 \leq 5 \\ 0 & \text{si } 3X_1 + x_2 > 5 \end{cases}$$

Le problème (DP) associé à (P1) devient

$$\begin{aligned} & \text{Max } \lambda \\ & \left\{ \begin{array}{ll} \lambda \leq \frac{3}{2} - x_1 & \text{si } 1 \leq 2x_1 \leq 3 \\ \lambda \leq \frac{1-3x_1-x_2}{3} & \text{si } 2 \leq 3X_1 + x_2 \leq 5 \\ x_1 + x_2 \leq 2 \\ 3x_1 + 2x_2 \leq 4 \\ x_1 \geq 0, x_2 \geq 0 \\ 0 \leq \lambda \leq 1 \end{array} \right. \end{aligned}$$

Résolution du problème (DP) :

$$\left\{ \begin{array}{l} \text{Max } \lambda \\ 2\lambda + 2x_1 + x_3 = 3 \\ 3\lambda + 3x_1 + x_2 + x_4 = 1 \\ x_1 + x_2 + x_5 = 2 \\ 3x_1 + 2x_2 + x_6 = 4 \end{array} \right.$$

La résolution avec le simplexe se fera dans les tableaux suivant :

C'		0	0	0	0	0	0	1
base	b	X ₁	X ₂	X ₃	X ₄	X ₅	X ₆	λ
X ₃	3	2	0	1	0	0	0	2
X ₄	1	3	1	0	1	0	0	3
X ₅	2	1	1	0	0	1	0	0
X ₆	4	3	2	0	0	0	1	0
Δ _j		0	0	0	0	0	0	-1
base	b	X ₁	X ₂	X ₃	X ₄	X ₅	X ₆	λ
X ₃	7/3	0	-2/3	1	-2/3	0	0	0
λ	1/3	1	1/3	0	1/3	0	0	1
X ₅	2	1	1	0	0	1	0	0
X ₆	4	3	2/3	0	0	0	1	0
Δ _j		1	1/3	0	1/3	0	0	0

La solution du problème (DP) est : $X=(0,0,7/3,0,2,4,1/3)$.

Chapitre 4

Programmation sur Lingo

Lingo : Logiciel pour la résolution des programmes linéaires et non linéaires

1 : Introduction :

Lingo est un logiciel utilisé pour résoudre les modèles d'optimisation linéaire, entier et quadratique, il est aussi utilisé pour résoudre les modèles d'optimisation globale non linéaire. Une des caractéristique de lingo c'est qu'il offre des outille qui peuvent aider à l'analyse des modèles en utilisant la méthode du simplexe.

2 : Installation du logiciel :

Pour utiliser cette version de lingo il est conseillé d'avoir au moins un processeur 486 et 8Mo de mémoire RAM. Il faut aussi prévoir un espace disque dur de 2 Mo pour pouvoir l'installer.

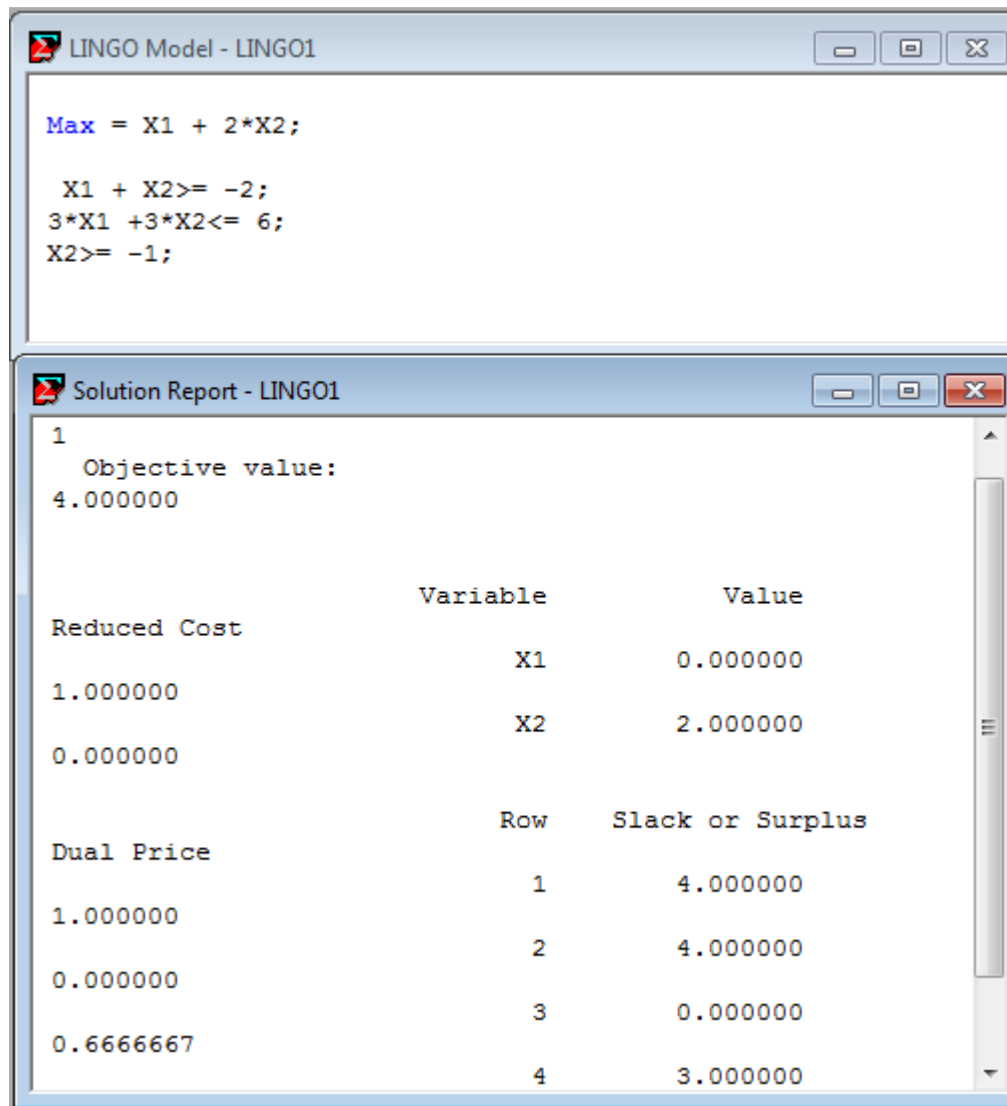
Les étapes de l'installation sont :

- 1 . Démarrer windows
- 2 . Insérer CD-ROM
- 3 . Cliquer sur l'icone setup (Install)dans votre explorateur de Windows
- 4 . Suivre les instructions de l'écran.

Pour plus d'information sur ce logiciel visiter l'adresse web www.lingo.com.

Application 1 :

Programmation du problème robuste.



The screenshot shows two windows from the LINGO software. The top window, titled "LINGO Model - LINGO1", contains the following model code:

```
Max = X1 + 2*X2;  
  
X1 + X2 >= -2;  
3*X1 + 3*X2 <= 6;  
X2 >= -1;
```

The bottom window, titled "Solution Report - LINGO1", displays the solution results. It shows the objective value and the values for the variables X1 and X2, along with the reduced costs and dual prices for the constraints.

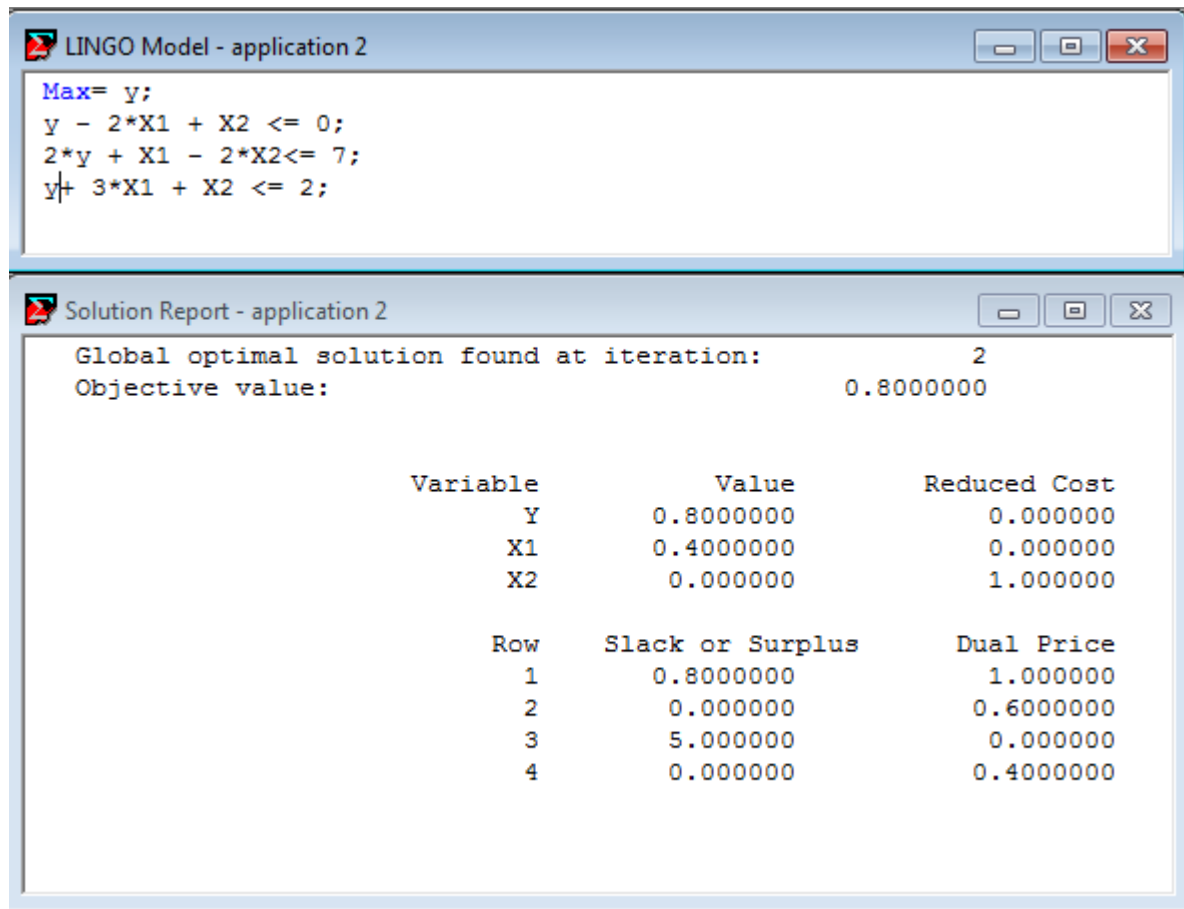
1
Objective value:
4.000000

	Variable	Value
Reduced Cost	X1	0.000000
1.000000	X2	2.000000
0.000000		

	Row	Slack or Surplus
Dual Price	1	4.000000
1.000000	2	4.000000
0.000000	3	0.000000
0.6666667	4	3.000000

Application 2 :

Programmation du problème flexible.



The screenshot shows two windows from the LINGO software. The top window, titled "LINGO Model - application 2", contains the following model code:

```
Max= y;  
y - 2*X1 + X2 <= 0;  
2*y + X1 - 2*X2 <= 7;  
y + 3*X1 + X2 <= 2;
```

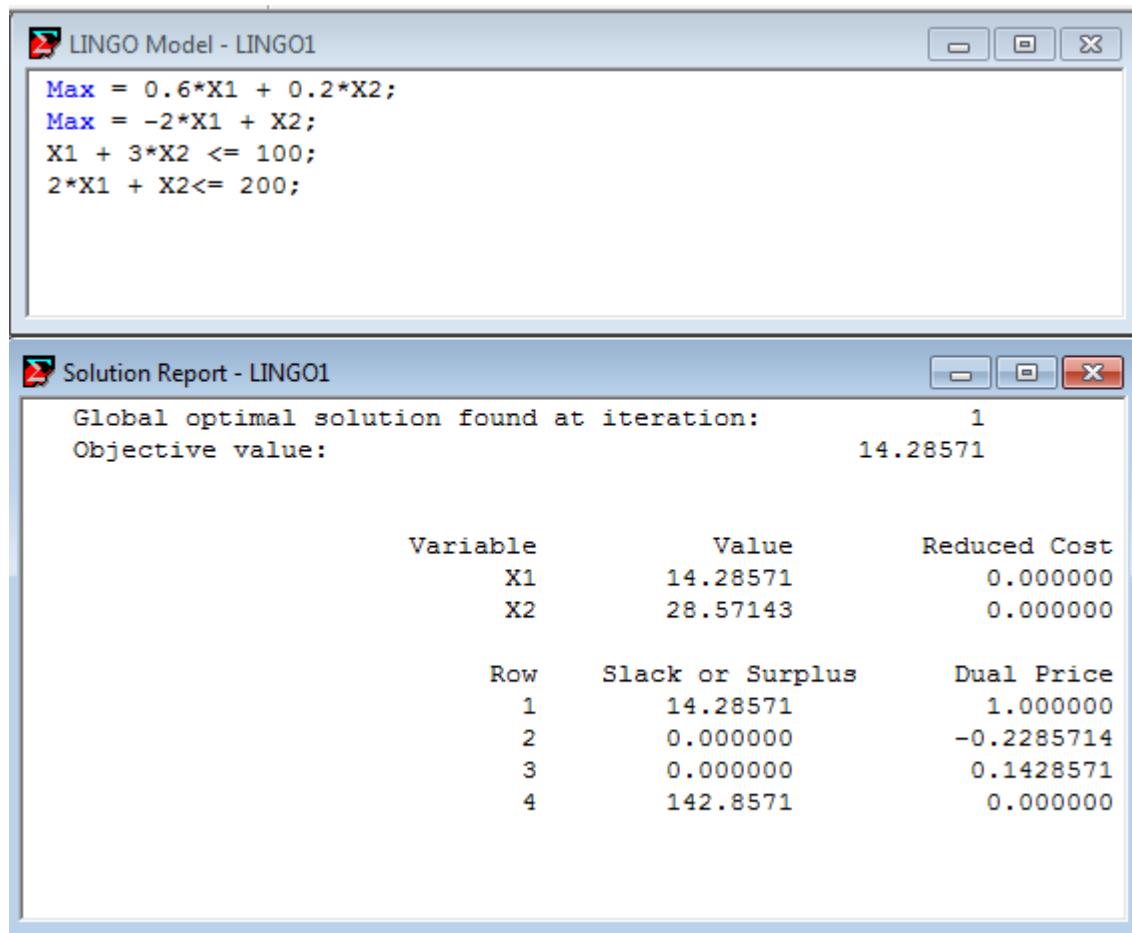
The bottom window, titled "Solution Report - application 2", displays the results of the optimization. It indicates that a global optimal solution was found at iteration 2, with an objective value of 0.8000000. Below this, two tables provide detailed data for variables and constraints.

Variable	Value	Reduced Cost
Y	0.8000000	0.000000
X1	0.4000000	0.000000
X2	0.000000	1.000000

Row	Slack or Surplus	Dual Price
1	0.8000000	1.000000
2	0.000000	0.600000
3	5.000000	0.000000
4	0.000000	0.400000

Application 3 :

Programmation du problème multicritère (chapitre1).



The image shows two windows from the LINGO software. The top window, titled "LINGO Model - LINGO1", contains the following model code:

```
Max = 0.6*X1 + 0.2*X2;  
Max = -2*X1 + X2;  
X1 + 3*X2 <= 100;  
2*X1 + X2 <= 200;
```

The bottom window, titled "Solution Report - LINGO1", displays the results of the optimization. It states that a global optimal solution was found at iteration 1, with an objective value of 14.28571. Below this, two tables are provided.

Variable	Value	Reduced Cost
X1	14.28571	0.000000
X2	28.57143	0.000000

Row	Slack or Surplus	Dual Price
1	14.28571	1.000000
2	0.000000	-0.2285714
3	0.000000	0.1428571
4	142.8571	0.000000

Conclusion Générale

Avant l'apparition de l'analyse multicritère, les problèmes de décision se ramenaient plus souvent à l'optimisation d'une seule fonction économique. Cette approche avait le mérite de déboucher sur des problèmes mathématiques bien posés mais qui ne représentaient pas sur la réalité.

Dans ce modeste travail, nous avons donné une vue panoramique des méthodes de résolution de problèmes multicritères. On peut conclure que ces méthodes sont basées sur l'optimisation continue, vu l'utilisation successives des algorithmes de la méthode du simplexe multi critère et la méthode de Abbas et Moulai.

Egalement, nous avons proposé pour résoudre un programme linéaire flou deux méthodes : flexibles (fonctions caractéristiques) et robuste et cela en transformant ce programme en un programme linéaire déterministe qui sera résolu par la méthode du simplexe.

Les deux méthodes exactes (simplexe, Abbas et moulai) concernant l'optimisation linéaire en nombres entiers à objectifs multiples représentent un résultat important dans la littérature, compte tenu des propriétés spécifiques du problème (MOLIP). Donc le problème reste ouvert aux autres chercheurs pour développer de meilleures méthodes de résolution de ce type de problèmes.

Références

- [1] Pareto A , cours d'économie politique, vol.1 et 2, F. Rouge, Lausanne , (1896).
- [2] Abbas, M, and Moulai, M., “ Solving multiple objective integer linear programming“, journal of Italian operations research society (Ricerca operative), Vol, 29 N°89 (1999), pp.15-38.
- [3] Samuel Ambapour, « Théorie des ensembles flous : application à la mesure de la pauvreté au Congo ».Bureau d'application des méthodes statistiques et informatiques, 2009, pages 5-8.
- [4] Mme Rabia, cours deuxième année master « Programmation floue » , 2015.
- [5] Internet, www.google.fr.
- [6] Aiche Farid, « sur la programmation linéaire multi-objectif floue et stochastique ». Université Mouloud Mammeri Tizi-ouzou. Pages 5-8. 2013.