MINISTERE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE SCIENTIFIQUE

UNIVERSITE MOULOUD MAMMERI, TIZI-OUZOU.

FACULTE DES SCIENCES

DEPARTEMENT de MATHEMATIQUES

MEMOIRE DE FIN D'ETUDES EN VUE DE L'OBTENTION DU DIPLOME de MASTER EN MATHEMATIQUES

OPTION: RECHERCHE OPERATIONNELLE

Présenté par

M^{elle} HAMMAR Sonia et M^{elle} SEGGARI Karima

Sujet:

Problème de transport multiobjectif à coefficients intervalles

Devant le jury d'examen composé de :

M^{r} SADI Bachir	Maître de conférences A	UMMTO	Président
M^{me} RABIA Fatima	Maître de conférences A	UMMTO	promotrice
M^{me} BELAHCENE Saliha	Maître assistante A	UMMTO	Examinatrice

Soutenu le : 02 /10 /2019

Remerciements

– Nous remercions, avant tout, Dieu pour nous avoir donné la force de mener ce travail à terme.

– Nous tenons également à remercier nos parents pour leurs encouragements.

– Nos remerciements vont également à Madame RABIA qui a dirigé notre travail.

– Enfin, nous remercions les membres du jury d'avoir accepté d'examiner notre modeste travail.

Dédicaces

Je dédie ce modeste travail à :

- La mémoire de mes grands-parents, que Dieu Les accueille dans son vaste paradis.
- Mes très chers parents que nulle dédicace ne puisse exprimer ce que je leurs dois, pour leur bienveillance, leur amour et leur soutien ... " Que Dieu vous garde ".
- Mes chers frères Merzouk, Ferhat et M'hamed que je remercie pour leurs encouragements, leur aide et leur soutien.
- mes soeurs adorables Khadidja et Souhila que j'aime beaucoup.
- mes neveux Ahcene, Yani et Aylane ainsi ma nièce Mila.
- Mon binôme (Karima) ainsi qu'à toute sa famille.
- Ma chère copine Lilia .
- Tous ceux qui me sont chers.

Sonia

Dédicaces

Je dédie ce modeste travail à :

- Mes très chers parents qui m'ont beaucoup soutenu et aidé tout le long de mon parcours scolaires " Que Dieu vous garde ".
- Mon cher mari Makhlouf ainsi toute sa famille,
- Mes chères soeurs: Djidji, Aldjia et Lilia ainsi mes neveux et nièces.
- mon chèr Samir et son fils Racim.
- a toute ma famille, mes oncles, mes tantes, cousins et cousines.
- mes chères amies:Sekoura, Karima, Fatima et Samia.
- Mon binôme (sonia) ainsi qu'à toute sa famille.
- mes chères copines Tina et Katia
- tous ceux qui m'ont aidé.

Karima

Table des matières

Introduction

La recherche opérationnelle, dite science du management, ou science de la décision est une discipline dont l'objet est d'aider les gestionnaires à prendre des décisions en utilisant des modèles et des méthodes scientifiques adaptées qui servent à résoudre les problèmes complexes rencontrés dans plusieurs domaines.

C'est pour cela que beaucoup d'entreprises s'intéressent à la recherche opérationnelle afin d'optimiser leurs productions et/ou leurs services. Le but de cette activité est la gestion efficace, rationnelle et logique. La recherche opérationnelle fait appel à d'autres domaines qui sont l'informatique, les mathématiques ainsi l'économie de l'entreprise.

Le transport est l'un des problèmes auquel se sont intéressé les chercheurs en recherche opérationnelle. Il représente un nerf très sensible pour chaque entreprise que se soit de production ou de distribution. L'objectif est d'acheminer des produits depuis le lieu de production jusqu'aux installations de stockage ou de consommation afin de garantir la disponibilité permanente des produits ainsi la satisfaction des clients [3, 4, 15, 17]. Toutefois, les coûts de transport dépendent généralement de paramètres extérieurs non contrôlés, donc la modélisation du problème doit tenir compte de cette incertitude [20]. L'approche la plus classique est celle qui suppose que cette incertitude est de nature floue ou stochastique, les données sont considérées comme des nombres flous ou des variables aléatoires. Dans ce mémoire nous ne considérons que le cas de l'incertitude de nature floue.

L'objectif dans la gestion des entreprises de transport n'est pas uniquement la minimisation des coûts de transport, mais aussi l'assurance de la poursuite des activités, maximiser le profit et la part du marché et minimiser le recours au financement externe. La modélisation des problèmes de transport doit prendre en considération tous les critères de décision qui sont nombreux et diversifiés, ceux relatifs aux clients aux industriels, aux lois et règlements,. . . etc. De telles inquiétudes reflètent des intérêts qui sont généralement divergents sinon contradictoires dans la mesure ou l'optimisation d'un objectif peut entrainer la dégradation d'un ou plusieurs autres objectifs. Ceci nous amène à des problèmes décisionnels multicritères ou multiobjectifs.

Dans notre travail nous proposons un programme qui peut donner une estimation minimale du coût de transport. Nous considérons les critères flous et multiobjectif du problème de transport à coefficient intervalles. Si ce point de vue permet de mieux entourer la réalité, il complique plus les méthodes de résolution en combinant les difficultés de la programmation floue avec celles de la program-

mation multiobjectif.

Nous avons structuré notre travail sous cinq chapitres :

Le premier chapitre traite le problème de transport classique. Nous donnons une decription détaillée de sa méthode de résolution dite méthode des potentiels. Nous exposons aussi les différentes méthodes de recherche de la solution initiale.

Dans le deuxième chapitre, nous nous sommes intéressées à l'optimisation multiobjectif en rappelant les définitions de base relatives à la dominance et l'efficacité d'une solution de base réalisable. Nous y avons inclu quelques approches utilisées pour la résolution des problèmes multiobjectifs.

Nous avons consacré le troisième chapitre aux concepts de base de la théorie des ensembles flous ainsi qu'aux deux approches de résolution des programmes linéaires flous qui sont la programmation fléxible et la programmation robuste.

Le quatrième chapitre constitue la partie essentielle de notre travail. Nous avons mis au point une méthode de résolution d'un problème de transport où les coefficients des objectifs sont des intervalles. Notre méthode est basée essentiellement sur la théorie des ensembles exposé dans le chapitre précédent.

Le cinquième chapitre est réservé l'implémentation de l'algorithme développé ainsi qu'aux résultats obtenus en le déroulant sur un exemple pratique posé par Naftal de Tizi-Ouzou.

Nous clôturons notre travail par une conclusion et la liste des références utilisées.

Chapitre 1

Problème de transport

1.1 Introduction

Les problèmes de transport sont des programmes linéaires dont la principale caractéristique est d'avoir une structure spéciale. Les problèmes de transport les plus connus sont: le problème de transport classique à deux indices [21, 25], le problème de répartition équilibrée [5], le problème d'affectation [4], le problème de tournées de véhicule [11, 15] ainsi que les problèmes de transport à indices multiples [6]. Dans ce chapitre, nous nous intéresserons qu'au problème de transport classique.

1.2 Description du problème de transport classique

On dispose de m dépôts de marchandise et n centres de demandes (magasins). On note par :
$a_\imath$: La disponibilité au dépôt $(\imath)$
$b_\jmath$: La demande du magasin $(\jmath)$
$x_{\imath\jmath}$: La quantité de marchandise à acheminer du dépôt $(\imath)$ vers le magasin $(\jmath)$
$c_{\imath\jmath}$: Le coût de transport d'une unité de marchandise du dépôt $(\imath)$ au magasin $(\jmath)$

Le problème consiste à minimiser le coût de transport global de tous les dépôts vers tous les magasins. Ce coût est représenté par la fontion:

$$z = \sum_{\imath=1}^{m} \sum_{\jmath=1}^{n} c_{\imath\jmath} x_{\imath\jmath}$$

Pour que le problème de transport admette une solution, il faut que l'offre soit égale à la demande c'est à dire, $\sum_{\imath=1}^{m} a_\imath = \sum_{\jmath=1}^{n} b_\jmath$. Dans ce cas le problème s'écrit sous la forme :

$$(T) \begin{cases} z = \sum_{\imath=1}^{m} \sum_{\jmath=1}^{n} c_{\imath\jmath} x_{\imath\jmath} \rightarrow \min \\ \textit{sous les contraintes} : \\ \sum_{\jmath=1}^{n} x_{\imath\jmath} = a_\imath \quad ; \quad \imath = 1, \ldots, m \\ \sum_{\jmath=1}^{m} x_{\imath\jmath} = b_\jmath \quad ; \quad \jmath = 1, \ldots, n \\ x_{\imath\jmath} \geq 0; \imath = 1, \ldots, m, \jmath = 1, \ldots, n \end{cases} \tag{1.1}$$

En effet, $\sum_{\imath=1}^{m} a_\imath = \sum_{\imath=1}^{m} \sum_{\jmath=1}^{n} x_{\imath\jmath} = \sum_{\jmath=1}^{n} \sum_{\imath=1}^{m} x_{\imath\jmath} = \sum_{\jmath=1}^{n} b_\jmath$

1.2.1 Cas où la l'offre est supérieure à la demande

Dans cette situation ($\sum_{\imath=1}^{m} a_\imath > \sum_{\jmath=1}^{n} b_\jmath$). Le modèle mathématique s'écrit alors de la manière suivante :

$$(T)\begin{cases} Z = \sum_{\imath=1}^{m}\sum_{\jmath=1}^{n} c_{\imath\jmath}x_{\imath\jmath} \rightarrow \min \\ \textit{sous les contraintes} : \\ \sum_{\jmath=1}^{n} x_{\imath\jmath} \leq a_\imath \quad ; \quad \imath = 1,..,m. \\ \sum_{\imath=1}^{m} x_{\imath\jmath} = b_\jmath \quad ; \quad \jmath = 1,..,n. \\ x_{\imath\jmath} \geq 0; \imath = 1,\ldots,m, \jmath = 1,\ldots,n \end{cases} \tag{1.2}$$

conséquence 1.1. *Pour avoir l'équilibre (offre = demande) il faut ajouter une ligne* $(m+1)$ *(dépôt fictif). Cela a pour conséquence l'ajout de (m variables d'écart)* $x_{m+1,\jmath}$, $1 \leq \jmath \leq n$ *dont les coefficients* $c_{m+1,\jmath}$ *sont nuls* $(c_{m+1,\jmath} = 0)$. *La disponibilité au niveau du dépôt* $(m+1)$ *est* $a_{m+1} = \sum_{\imath=1}^{m} a_\imath - \sum_{\jmath=1}^{n} b_\jmath$.

1.2.2 Cas où la demande est supérieure à l'offre

Dans ce cas nous avons ($\sum_{\jmath=1}^{n} b_\jmath > \sum_{\imath=1}^{m} a_\imath$). Le modèle mathématique s'écrit alors de la manière suivante :

$$(T)\begin{cases} Z = \sum_{\imath=1}^{m}\sum_{\jmath=1}^{n} c_{\imath\jmath}x_{\imath\jmath} \rightarrow \min \\ \textit{sous les contraintes} : \\ \sum_{\jmath=1}^{n} x_{\imath\jmath} = a_\imath \quad ; \quad \imath = 1,..,m. \\ \sum_{\imath=1}^{m} x_{\imath\jmath} \leq b_\jmath \quad ; \quad \jmath = 1,..,n. \\ x_{\imath\jmath} \geq 0; \imath = 1,\ldots,m, \jmath = 1,\ldots,n \end{cases} \tag{1.3}$$

conséquence 1.2. *Pour avoir l'équilibre (offre = demande) il faut ajouter une colonne* $(n+1)$ *(destination fictive). Cela a pour conséquence l'ajout de (n variables d'écart)* $x_{\imath,n+1}$, $1 \leq \imath \leq m$ *dont les coefficients* $c_{\imath,n+1}$ *sont nuls* $(c_{\imath,n+1} = 0)$. *La demande au niveau de la destination* $(n+1)$ *est* $b_{n+1} = \sum_{\jmath=1}^{n} b_\jmath - \sum_{\imath=1}^{m} a_\imath$.

1.2.3 Ecriture matricielle du problème de transport

Les données du problème de transport sont représentées sous forme d'un tableau appelé tableau de transport:

A \ B	B_1	B_2	$\dots$	B_n	a_i
A_1	c_{11} / x_{11}	c_{12} / x_{12}	$\dots$	c_{1n} / x_{1n}	a_1
A_2	c_{21} / x_{21}	c_{22} / x_{22}	$\dots$	c_{2n} / x_{2n}	a_2
$\vdots$	$\vdots$	$\vdots$		$\vdots$	$\vdots$
A_m	c_{m1} / x_m	c_{m2} / x_{m2}	$\dots$	c_{mn} / x_{mn}	a_m
b_j	b_1	b_2	$\dots$	b_n	$\sum_{i=1}^{m} a_i = \sum_{j=1}^{n} b_j$

Tableau de transport

Le problème de transport peut aussi se présenter sous forme matricielle:

$$\begin{cases} Z = CX \rightarrow \min \\ \textit{sous les contraintes} : \\ AX = B \\ X \geq 0 \end{cases}$$

où A est une matrice à $(m+n)$ lignes et mn colonnes, B est un $(m+n)$ vecteur colonne, C est un mn-veteur ligne et X est un mn-veteur colonne avec $X = (x_{11}, x_{12}, \dots, x_{ij}, \dots, x_{mn})$.

Pour $m = 2$ et $n = 3$ par exemple, la matrice A est donné par:

$$A = \begin{pmatrix} 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 \end{pmatrix}$$

La matrice de transport possède trois propriétés essentielles:

- *A est une matrice unimodulaire : elle ne contient que des 0 et des 1*
- *Chaque vecteur colonne de A contient seulement deux valeurs positives et égales à 1*
- *Pour un problème de transport avec m dépôts et n centres de consommation, le rang de la matrice A est égal à $m+n-1$ ($Rang(A) = m+n-1$). Par conséquent, une solution de base réalisable du problème de transport possède*

donc $(m+n-1)$ composantes positives (de base) et $mn-(m+n-1) = mn-m-n+1 = (m-1)(n-1)$ composantes hors bases (nulles). La matrice A contient donc une ligne redondante.

Bien que le problème de transport soit un problème linéaire, on ne peut, en général, le résoudre avec le simplexe car, non seulement le nombre de variables et de contraintes est important mais il faut aussi trouver la ligne redondante et la supprimer avant de résoudre. C'est pour cette raison que la méthode des potentiels (ou variables duales) a été désignée pour la recherche de la solution optimale du problème de transport.

1.3 Méthode des potentiels

Cette méthode est basée sur la méthode révisée du simplexe qui utilise les variables duales associées aux contraintes du problème et sur les propriétes de la matrice de transport citées ci-dessus (voir par exemple [21, 25]).

1.3.1 Recherche de solution de base initiale

Il existe plusieurs méthodes de calcul de la solution initiale pour un problème de transport. Nous ne citerons ici que celles qui sont le plus souvent utilisées à savoir:

- ✓ *Coin nord ouest*
- ✓ *Coût minimal*
- ✓ *Méthode de Vogel.*

Méthode du coin nord-ouest

On choisit la case (1,1), située au coin nord-ouest du tableau de transport, et on lui affecte la quantité $x_{11} = \min\{a_1, b_1\}$. On peut avoir deux cas :

1. Si $x_{11} = a_1$, alors la quantité de A_1 est entièrement transportée et ceci sature la première ligne du tableau. Dans le tableau réduit, on remplacera b_1 par $(b_1 - x_{11})$ et on répète la même procédure.

2. Si $x_{11} = b_1$, alors la demande du point de distribution B_1 est entièrement satisfaite par A_1 et ceci sature la première colonne. Dans le tableau réduit, on remplacera a_1 par $(a_1 - x_{11})$ et on répètera la même procédure.

De cette manière, après $(m+n-1)$ opérations, on trouve $(m+n-1)$ quantités positives x_{ij} affectées à $(m+n-1)$ cases. Les cases restantes auront des quantités nulles $x_{ij}=0$. On obtiendra ainsi la solution de base rélisable du problème de transport.

Exemple

A \ B	B_1	B_2	B_3	B_4	$offre$
A_1	10 **5**	2 **10**	20	11	15
A_2	12	7 **5**	9 **15**	20 **5**	25
A_3	4	14	16	18 **10**	10
demande	5	15	15	15	50

$x_{11}=\min\{15,5\}=5; x_{12}=\min\{15-5,15\}=10;$
$x_{22}=\min\{25,15-10\}=5; x_{23}=\min\{25,15\}=15;$
$x_{24}=\min\{25-5-15,15\}=5$
$x_{34}=\min\{10,15\}=10$

Le coût de transport associé à cette solution est : 520

Méthode du coût minimal

Le principe consiste à choisir une case (i_1,j_1) qui correspond à l'élément c_{i_1,j_1} tel que : $c_{i_1,j_1}=min\{c_{ij},\ 1\leq i\leq m,\ 1\leq j\leq n\}$, puis on posera $x_{i_1j_1}=min\{a_{i_1},b_{j_1}\}$ dans la case (i_1,j_1). Si $x_{i_1j_1}=a_{i_1}$, on exclut la ligne i_1 et on remplace le nombre b_{j_1} par $(b_{j_1}-x_{i_1j_1})$.
Dans le cas où $x_{i_1j_1}=b_{j_1}$, on exclut la colonne j_1 et on remplace le nombre a_{i_1} par le nombre $(a_{i_1}-x_{i_1j_1})$. Ensuite nous répétons la même procédure avec le tableau réduit.
Ce processus sera répété $(m+n-1)$ fois pour trouver enfin les $(m+n-1)$ variables de base de la solution initiale recherchée.

Exemple

Reprenons le tableau du premier exemple

B / A	B_1	B_2	B_3	B_4	$offre$
A_1	10	2 **15**	20	11	15
A_2	12	7	9 **15**	20 **10**	25
A_3	4 **5**	14	16	18 **5**	10
demande	5	15	15	15	50

$c_{12} = \min_{i,j} c_{ij} = 2 = \min\{15,15\} = 15$
$c_{31} = \min_{\{i\neq 1, j\neq 2\}} c_{ij} = 4 = \min\{10,5\} = 5$
$c_{23} = \min_{\{i\neq 1, j\neq 1,2\}} c_{ij} = 9 = \min\{25,15\} = 15$
$c_{34} = \min_{\{i\neq 1, j\neq 1,2,3\}} c_{ij} = 18 = \min\{10-5,15\} = 5$
$c_{24} = \min_{\{i\neq 1,3 j\neq 1,2,3\}} c_{ij} = 20 = \min\{25-15,15-5\} = 10$

Le coût de transport associé à cette solution est : **475**

Méthode de Vogel

La méthode de Vogel peut avoir certains avantages sur la règle du coin nord-ouest pour déterminer une solution initiale de base. Il paraît, en effet, qu'elle permet d'atteindre la solution optimale plus rapidement. La démarche est la suivante:

1. Construire la matrice des coûts en incluant les disponibilités et demandes. Ajouter une destination fictive ou une origine fictive pour que l'offre soit égale à la demande $\sum_{i=1}^{m} a_i = \sum_{j=1}^{n} b_j$

2. Evaluer la différence entre les deux coûts les plus petits pour chaque ligne et chaque colonne. Nous obtenons ainsi m différences pour les lignes et n différences pour les colonnes

3. Choisir la ligne ou la colonne ayant le maximum de différences; faire un choix arbitraire si le maximum des différences n'est pas unique.

4. Allouer la quantité la plus grande possible (tout en respectant les contraintes) à la cellule possédant le coût le plus faible de la ligne ou de la colonne obtenue à l'étape 3.

5. Supprimer la ligne ou la colonne qui est saturée.

6. Retourner à l'étape 2 mais cette fois les calculs seront fait sur la matrice résultante. La procédure se termine lorsque toutes les lignes et les colonnes sont saturées.

Exemple

B / A	B_1	B_2	B_3	B_4	$offre$
A_1	10	2 **15**	20	11	15
A_2	12	7	9 **15**	20 **10**	25
A_3	4 **5**	14	16	18 **5**	10
demande	5	15	15	15	50

La procédure suivie est résumée dans le tableau ci-dessous :

itération	calcul des différences							
	lignes			colonnes				
	1	2	3	1	2	3	4	
1	8	2	10	6	5	7	7	$x_{31} = 15$
2	9	2	2	–	5	7	7	$x_{12} = 5$
3	–	11	2	–	–	7	2	$x_{23} = 5$
4	–	20	18	–	–	–	2	$x_{24} = 10$
5	–	–	18	–	–	–	18	$x_{34} = 5$

Le coût de transport associé à cette solution est : **475**

1.3.2 Calculs des variables duales ou potentiels

Notons par $u_\imath$ et $v_\jmath$ les variables duales associées respectivement aux premier et au second goupe de contraintes du problème. Le problème dual s'écrit alors,

$$(D)\begin{cases}\max W = \sum_{\imath=1}^{m} a_\imath u_\imath + \sum_{\jmath=1}^{n} b_\jmath v_\jmath \\ souslescontraintes \\ u_\imath + v_\jmath \leq c\imath\jmath, \quad (\imath = \overline{1,m}); \quad (\jmath = \overline{1,n}) \\ u_\imath \quad ,v_\jmath \quad quelconques\end{cases} \tag{1.4}$$

Les variables duales $u_\imath$ et $v_\jmath$ sont données par le système :

$$u_\imath + v_\jmath = c_{\imath\jmath}, \quad \forall(\imath,\jmath) \in I_B \tag{1.5}$$

où I_B est l'ensemble des cases de base.

Les potentiels $u_\imath$ et $v_\jmath$ sont au nombre de $(m+n)$ et doivent vérifier le système (1.5) qui est constitué de $(m+n-1)$ équations. Donc pour sa résolution, il faut choisir une valeur arbitraire par exemple u_1 et poser $u_1 = 0$, les autres potentiels sont alors déterminés d'une manière unique par les équations (1.5).

Calculons les potentiels pour la solution donnée par la méthode du coin nord-ouest: $I_B = \{(1,1),(1,2),(2,2),(2,3),(2,4),(3,4)\}$ dans ce cas la relation (1.5) se traduit par:

$u_1 - v_1 = c_{11}$
$u_1 - v_2 = c_{12}$
$u_2 - v_2 = c_{22}$
$u_2 - v_3 = c_{23}$
$u_2 - v_4 = c_{24}$
$u_3 - v_4 = c_{34}$

on posant $u_1 = 0$, nous trouvons:
$v_1 = 10$
$v_2 = 2$
$u_2 = 5$
$v_3 = 4$
$v_4 = 15$
$u_3 = 3$

1.3.3 Calculs des coûts marginaux et critère d'optimalité

Soient x^0 et $y^0 = (u^0, v^0)$ deux solutions optimales respectivement du problème (T) et de son dual (D), alors les relations suivantes sont vérifiées:

$$\begin{cases} u_i^o + v_j^o = c_{ij} \ , & si \ \ x_{ij}^o > 0, \\ u_i^o + v_j^o \leq c_{ij} \ , & si \ \ x_{ij}^o = 0, \\ 1 \leq i \leq m \quad , & 1 \leq j \leq n, \end{cases} \tag{1.6}$$

On construit les estimations ou coûts marginaux Δ_{ij} tels que
$\Delta_{ij} = u_i + v_j - c_{ij} \ , \quad 1 \leq i \leq m \ , \quad 1 \leq j \leq n$. Par construction on a donc:

$$u_i + v_j - c_{ij} = 0 \ , \qquad (i,j) \in I_B \tag{1.7}$$

$$\Delta_{ij} = u_i + v_j - c_{ij} \ , \qquad (i,j) \in I_H \tag{1.8}$$

Théorème. *: (critère d'optimalité)*
Les inégalités :

$$\Delta_{ij} \leq 0; \quad (i,j) \in I_H \tag{1.9}$$

Sont suffisantes pour l'optimalité d'une solution de base du problème de transport. Elles sont aussi nécessaires dans le cas où la solution de base est non dégénérée.

Calculaons les couts marginaux pour la solution obtenue par la méthode du coin nord-ouest
$I_H = \{(1,3),(1,4),(2,1),(3,1),(3,2),(3,3)\}$ dans ce cas la relation (1.8) se traduit par

$\Delta_{13} = u_1 + v_3 - c_{13} = 0 + 4 - 20 = -16$
$\Delta_{14} = u_1 + v_4 c_{14} = 0 + 15 - 11 = 4$
$\Delta_{21} = u_2 + v_1 + c_{21} = 5 + 10 - 12 = 3$
$\Delta_{31} = u_3 + v_1 + c_{31} = 3 + 10 - 4 = 9$
$\Delta_{32} = u_3 + v_2 + c_{32} = 3 + 2 - 14 = -9$
$\Delta_{33} = u_3 + v_3 + c_{33} = 3 + 4 - 16 = -9$
$\Delta_{i_0 j_0} = \Delta_{ij} = \Delta_{31} > 0$, le critère d'optimalité n'est pas vérifié donc la solution de base initial n'est pas optimale.

1.3.4 Amélioration de la solution

Soit $x = (x_i, j)$ une solution de base de départ à laquelle correspond l'ensemble des indices de base I_B. Si le critère d'optimalité (1.9) n'est pas vérifié alors, conformément à l'algorithme du simplexe, nous cherchons une case $(i_0, j_0) \in I_H$ telle que

$$\Delta_{i_0 j_0} = \max \Delta_{ij}, \qquad \Delta_{i,j} \geq 0 \tag{1.10}$$

A l'aide de la case $(\imath_0,\jmath_0)$ et des cases de I_B on construit un cycle qui est d'ailleurs unique en affectant des signes $(+)$ et $(-)$ successivement aux sommets de ce cycle en commençant par le sommet $(\imath_0,\jmath_0)$ et en se mouvant dans le sens des aiguilles d'une montre ou dans le sens contraire. Parmi les sommets du cycle affectés du signe $(-)$, on choisit celui où la variable $x_{\imath\jmath}$ est minimale et on pose

$$\Theta^0 = \min x_{\imath\jmath}. \tag{1.11}$$

Pour les sommets affectés du signe $(+)$, on ajoute aux variables $x_{\imath\jmath}$ la quantité Θ^0 et on soustrait la même quantité des variables $x_{\imath\jmath}$ correspondantes aux sommets affectés du signe $(-)$. Toutes les autres variables $x_{\imath\jmath}$ resteront inchangées.

Nous obtenons ainsi une nouvelle solution de base $\bar{x}$ avec un nouvel ensemble de base $\bar{I}_B$. Cette itération sera répétée jusqu'à ce que le critère d'optimalité soit satisfait.

B / A	B_1	B_2	B_3	B_4	$offre$	$u_\imath$
A_1	10 θ^- **5**	2 θ^+ **10**	20 (−16)	11 (4)	15	0
A_2	12 (3)	7 θ^- **5**	9 **15**	20 θ^+ **5**	25	5
A_3	4 θ^+ (9)	14 (−9)	16 (−9)	18 θ^- **10**	10	3
$demande$	5	15	15	15	50	
$v_\jmath$	10	2	4	15		

A l'aide de la case (1,3), on construit le cycle (1,3) → (1,1) → (1,2) → (2,2) → (2,4) → (3,4), on aura alors:
$\theta^o = x_{\imath_1\jmath_1} = \min\{5,10,5,5,10\} = 5$
Pour la nouvelle solution de base $\bar{x}$,on obtient:
$\bar{x}_{13} = x_{13} + \theta^o = 5;$
$\bar{x}_{11} = x_{11} - \theta^o = 0;$
$\bar{x}_{12} = x_{12} + \theta^o = 15;$
$\bar{x}_{22} = x_{22} - \theta^o = 0;$
$\bar{x}_{24} = x_{24} + \theta^o = 10;$
$\bar{x}_{34} = x_{34} - \theta^o = 5;$

les autres variables $\bar{x}_{\imath\jmath}$ resteront inchangées.

A \ B	B_1	B_2	B_3	B_4	$offre$	u_i
A_1	10 **0**	2 θ^- **15**	20 (-16)	11 θ^+ (4)	15	0
A_2	12 (-6)	7 θ^+ **0**	9 **15**	20 θ^- **10**	25	5
A_3	4 **5**	14 (-9)	16 (-9)	18 **5**	10	3
$demande$	5	15	15	15	50	
v_j	1	2	4	15		

La nouvelle base est $I_B = \{(1,2),(2,2),(2,3),(2,4),(3,1),(3,4)\}$

Dans la suite, nous poursuivons les calculs jusqu'à atteindre la solution optimale
$u_1 + v_2 = c_{12}$
$u_2 + v_2 = c_{22}$
$u_2 + v_3 = c_{23}$
$u_2 + v_4 = c_{24}$
$u_3 + v_1 = c_{31}$
$u_3 + v_4 = c_{34}$
posons que $u_1 = 0$ donc on aura
$v_2 = 2$
$u_2 = 5$
$v_3 = 4$
$v_4 = 15$
$u_3 = 3$
$v_1 = 1$

Calculaons les couts marginaux: $I_H = \{(1,1),(1,3),(1,4),(2,1),(3,2),(3,3)\}$ $\Delta_{11} = u_1 + v_1 - c_{11} = 0 + 1 - 10 = -9$
$\Delta_{13} = u_1 + v_3 - c_{13} = 0 + 4 - 20 = -16$
$\Delta_{14} = u_1 + v_4 + c_{14} = 0 + 15 - 11 = 4$
$\Delta_{21} = u_2 + v_1 + c_{21} = 5 + 1 - 12 = -6$
$\Delta_{32} = u_3 + v_2 + c_{32} = 3 + 2 - 14 = -9$
$\Delta_{33} = u_3 + v_3 + c_{33} = 3 + 4 - 16 = -9$
$\Delta_{i_0 j_0} = \max \Delta_{ij} = \Delta_{14} = 4 > 0$ le critère d'optimalité n'est pas vérifié donc la solution courante n'est pas optimale.

A l'aide de la case (1,4), on construit le cycle (1,4) $\rightarrow$ (2,4) $\rightarrow$ (1,2) $\rightarrow$ (2,2) $\rightarrow$ (1,2), on aura alors:
$\theta^o = x_{i_1 j_1} = \min\{10,15\} = 10$
pour le nouveau plan basique $\bar{x}$,on obtient:

$\bar{x}_{12} = x_{12} - \theta^o = 5$;
$\bar{x}_{14} = x_{14} + \theta^o = 10$;
$\bar{x}_{24} = x_{24} - \theta^o = 0$;
$\bar{x}_{22} = x_{22} + \theta^o = 10$;
les autres variables $\bar{x}_{ij}$ resteront inchangées.

B / A	B_1		B_2		B_3		B_4		$offre$	u_i
A_1	10	(-13)	2	**5**	20	(-16)	11	**10**	15	0
A_2	12	(-10)	7	**10**	9	**15**	20	(-4)	25	2
A_3	4	**5**	14	(-5)	16	(-5)	18	**5**	10	7
$demande$	5		15		15		15		50	
v_j	-3		2		4		11			

La nouvelle base est $I_B = \{(1,2),(2,4),(2,2),(2,3),(3,1),(3,4)\}$

$u_1 + v_2 = c_{12}$
$u_1 + v_4 = c_{14}$
$u_2 + v_2 = c_{22}$
$u_2 + v_3 = c_{23}$
$u_3 + v_1 = c_{31}$
$u_3 + v_4 = c_{34}$
posons que $u_1 = 0$ donc on aura
$v_2 = 2$
$u_2 = 5$
$v_3 = 4$
$u_3 = 7$
$u_1 = -33$
$v_4 = 11$

Calculaons les couts marginaux: $I_H = \{(1,1),(1,3),(2,1),(2,4),(3,2),(3,3)\}$ $\Delta_{11} = u_1 + v_1 - c_{11} = 0 - 3 - 10 = -13$
$\Delta_{13} = u_1 + v_3 - c_{13} = 0 + 4 - 20 = -16$
$\Delta_{21} = u_1 + v_4 - c_{14} = 2 + 3 - 12 = -7$
$\Delta_{24} = u_2 + v_4 - c_{24} = 2 + 11 - 20 = -7$
$\Delta_{32} = u_3 + v_2 - c_{32} = 7 + 2 - 14 = -5$
$\Delta_{33} = u_3 + v_3 - c_{33} = 7 + 4 - 16 = -5$
$\Delta_{i_0 j_0} = \max \Delta_{ij} = \Delta_{32} = -5 < 0$ le critère d'optimalité est vérifié donc la solution courante est optimale.
$x^o = \{x^o_{ij}, 1 \leq i \leq 3, 1 \leq j \leq 4\}$ avec
$x^o_{11} = 5; x^o_{14} = 10; x^o_{22} = 10; x^o_{23} = 15; x^o_{31} = 5; x^o_{34} = 5;$

$z^o = \min z(x) = z(x^o) = 435$

1.4 Dégénérescence

Nous avons une solution dégénérée lorsque on a une ou plusieurs variables de base nulles, c'est-à-dire toutes les contraintes sont satisfaites et qu'il a moins de $(m+n-1)$ cellules positives ($x_{ij} \geq 0$) dans la solution considérée. La dégénérescence peut apparaitre soit dans la première solution de base, soit au cours du processus des itérations menant à la solution optimale. Dans ce cas, pour pouvoir améliorer une solution, il faut considérer l'une (au moins) des variables hors base comme étant une variable de base.

Chapitre 2

Optimisation multi-objectif

2.1 Introduction

Dans la plupart des problèmes dans le monde réel on ne peut optimiser un seul objectif mais plusieurs objectifs simultanément. Ces objectifs sont généralement conflictuels et/ou présentés dans des unités différentes [1, 8]. Les problèmes de ce type sont résolus par la programmation multiobjectif.

La difficulté d'un problème multiobjectif réside dans le fait qu'il soit un problème mathématique mal posé dans le sens où il n'admet pas de solution optimale. En effet, il n'éxiste pas une solution meilleure que toutes les autres simultanément pour tous les objectifs. Le concept de solution optimale n'a donc pas de sens dans ce contexte.

Nous allons utiliser le terme solution optimale au sens de Pareto ou solution non dominée ou encore solution efficace, il nous faut donc en donner les définitions précises.

Mathématiquement le problème multiobjectif est défini comme suit:

$$\begin{cases} \text{" max "} z(x) = (z_1(x), z_2(x), \ldots, z_k(x)) \\ \textit{sous les contraintes} \\ x \in \mathbb{S} \end{cases} \tag{2.1}$$

$x = (x_1, x_2, \ldots, x_n)^t \in \mathbb{R}^n$ représente le vecteur de décision.
$z(x) = (z_1(x), z_2(x), \ldots, z_k(x))$ représente le vecteur de fonctions objectifs (ou critères de décision) et k est le nombre d'objectifs, $k \geq 2$.
S est l'ensemble non vide des solutions réalisables vérifiant un certain nombre de contraintes.
Le symbole " " signifie qu'en général on ne peut pas trouver des solutions réalisables qui maximisent les k critères simultanément et évidemment, le maximum peut être

remplacé par un minimum.
L'ensemble $\mathbb{R}^n$ qui contient S est dit espace de décision;
L'ensemble $\mathbb{R}^k$ qui contient $z(S)$ est dit espace des objectifs;
L'ensemble $z(S)$ est la projection de l'espace S sur l'espace des objectifs.

Dans ce qui suit, nous donnons quelques définitions relatives aux problèmes multiobjectifs dans le cas d'une maximisation.

2.2 Dominance et efficacité

ans ce paragraphe, nous donnons quelques définitions utiles pour la comprehension de la suite du document. Elles sont tirées de [23].

Dominance : Soient les deux vecteurs $y = (y_1,y_2,\ldots,y_k)$ et $z = (z_1,z_2,\ldots,z_k)$ de $\mathbb{R}^k$, on dit que y domine z si et seulement si $y_\imath \geq z_\imath \ \forall \imath = 1,\ldots,k$ et il existe $\imath_0 \in \{1,\ldots,k\}$ tel que $y_{\imath_0} > z_{\imath_0}$.
Cela veut dire que y est au moins aussi bon que z sur tous les objectifs et y est strictement meilleur que z sur au moins un objectif. On dit aussi que y est non dominé par z

Dominance faible ou stricte : On dit que $y = (y_1,y_2,\ldots,y_k)$ domine faiblement $z = (z_1,z_2,\ldots,z_k)$ si et seulement si $y_\imath > z_\imath \ \ \forall \imath = 1,\ldots,k$.
Cela veut dire que y est meilleur que z sur tous les objectifs.

Efficacité : Une solution $x^\star \in \mathbb{S}$ est dite efficace ou Pareto optimale si et seulement s'il n'existe pas de solution réalisable $x \in \mathbb{S}$ telle que: $z_\imath(x) \geq z_\imath(x^\star) \ \forall \imath = 1,\ldots,k$ et il existe au moins un $\imath_0 \in \{1,\ldots,k\}$ tel que $z_{\imath_0}(x) > z_{\imath_0}(x^\star)$.

Efficacité faible : Une solution $x^\star \in \mathbb{S}$ est faiblement efficace, s'il n'existe aucun vecteur $x \in \mathbb{S}$ tel que: $z_\imath(x) > z_\imath(x^\star) \ \ \forall \imath = 1,\ldots,k$

Il est clair qu'une solution efficace est faiblement efficace, mais l'inverse est faux.

2.3 Autres définitions utiles

Point idéal : Le point idéal est le vecteur $z^\star = (z_1^\star,z_2^\star,\ldots,z_k^\star) \in \mathbb{R}^k$ dont les composantes $z_\imath^\star$ représentent la valeur optimale de l'objectif $z_\imath$ c'est à dire,

$$z_\imath^\star = z_\imath(x^\star) = \max_{x \in \mathbb{S}} z_\imath(x)$$

Point anti-idéal : Le point anti-idéal $z^a = (z_1^a, z_2^a, \ldots, z_k^a) \in R^k$ est le vecteur dont chaque composante $z_\imath^a$ est telle que

$$z_\imath^a = \min z_\imath(x), \quad x \in \mathbb{S}$$

Point de référence: Le point de référence $z^r = (z_1^r, z_2^r, \ldots, z_k^r) \in \mathbb{R}^k$ est le vecteur qui définit le but à atteindre par chaque objectif $z_\imath$ (valeur souhaitable).

Front de Paréto: C'est l'ensemble des vecteurs critères qui ne sont pas dominés.

Matrice des gains: Soit $x_\imath^\star$ la solution optimale obtenue en maximisant le critère $z_\imath$ sur $\mathbb{S}$. La matrice des gains est la matrice carrée G de dimension k dont la diagonale principale constitue le point idéal $z^\star = (z_1^\star; z_2^\star, \ldots, z_k^\star)$ où $z_\imath^\star = \max_{x \in \mathbb{S}} z_\imath(x) = z_\imath(x_\imath^\star); \quad \imath = 1, \ldots, k$ et $z_{\imath\jmath} = z_\imath(x_\jmath^\star); \ \forall \imath = 1, \ldots, k; \ \forall \jmath = 1, \ldots, k \quad avec\ \imath \neq \jmath$.

Point nadir : De la matrice des gains, nous pouvons définir le point nadir $z^n = (z_1^n, z_2^n, \ldots, z_k^n) \in \mathbb{R}^k$ où $z_\imath^n = \min_{\jmath=1,\ldots,k} z_{\imath\jmath}; \quad \forall \imath = 1, \ldots, k$.

Remarque 2.1. *- Le point nadir est le vecteur des valeurs les plus mauvaises de chaque objectif $z_\imath$, $\imath = 1, \ldots, k$.*

- Si pour un objectif, il existe plusieurs solutions optimales alors, la matrice des gains n'est pas unique. Ainsi, le point nadir peut ne pas être unique.

- Le point nadir sert à restreinde l'espace de recherche R^k. Il peut être utilisé comme point de réservation (valeurs non souhaitables) utilisé dans les méthodes itératives.

Exemple: Soit la matrice des gains suivante:

$$G = \begin{pmatrix} 7 & 9 & 8 \\ 9 & 3 & 15 \\ 6 & 4 & 1 \end{pmatrix}$$

$z^\star = \begin{pmatrix} 7 & 3 & 1 \end{pmatrix}^t$ est le point idéal.
$z^n = \begin{pmatrix} 7 & 3 & 1 \end{pmatrix}^t$ pour un problème de maximisation.
$z^n = \begin{pmatrix} 9 & 15 & 6 \end{pmatrix}^t$ pour un problème de minimisation.

2.4 Fonctions scalarisantes

La recherche d'une solution efficace parmi toutes les solutions réalisables d'un problème multiobjectif exige une certaine connaissance de la structure de préférence. Cette information peut parfois se traduire en termes de paramètres de préférence (voir [8, 23]. Les plus courant sont :

1) Les poids $\lambda_\imath$, $\imath = 1,2,...,k$ qui reflètent l'importance relative de chaque objectif.

2) Le point de référence qui est défini par les valeurs souhaitables pour chaque objectif.

3) Le point de réservation qui est défini par des niveaux (valeurs non souhaitables) pour chaque objectif.

Une fonction scalarisante est définie par $s(z,\lambda) : \mathbb{R}^k \times \Lambda \to \mathbb{R}$, où Λ est l'ensemble des paramètres. C'est une fonction croissante des objectifs qui agrège les valeurs de ces derniers pour chaque solution. Les exemples les plus courants de fonctions scalarisantes sont les suivants:

Fonctions linéaires : $s_1(z,\lambda) = \sum_{\imath=1}^{k} \lambda_\imath z_\imath$ avec $\sum_{\imath=1}^{k} \lambda_\imath = 1,\ \lambda_\imath \geq 0,\ \forall \imath$

$$\text{et} \quad s_2(z,\lambda) = \sum_{\imath=1}^{k} \lambda_\imath \left| z_\imath - \bar{z}_\imath \right|$$

Norme L_p pondérée : $s_3(z,\lambda) = \left[\sum_{\imath=1}^{k} \lambda_\imath \left| z_\imath - \bar{z}_\imath \right|^q \right]^{\frac{1}{q}}$, $q \in \mathbb{Z}_+^*$

Norme L_∞de Tchebycheff pondérée : $s_4(z,\lambda) = \max_{1 \leq \imath \leq k} \{\lambda_\imath \left| z_\imath - \bar{z}_\imath \right|\}$

Norme Tchebycheff pondérée et augmentée:

$$s_5(z,\lambda) = \max_{1 \leq \imath \leq k} \{\lambda_\imath \left| z_\imath - \bar{z}_\imath \right|\} + \rho \sum_{\imath=1}^{k} \lambda_\imath \left| z_\imath - \bar{z}_\imath \right| ;\ \rho > 0.$$

Ici $\bar{z}_\imath$ désigne une valeur de référence de $z_\imath$.

Les fonctions scalarisantes ne peuvent engendrer que des vecteurs non dominés. Dans ce cas, elles caractérisent complètement l'ensemble des solutions efficaces.

2.5 Classification des approches multicritères

Dans la programmation mathématique multiobjectif, on distingue trois principales approches qui sont définies dans [18]:

2.5.1 Approche à priori (décideur→recherche)

Dans ce type de méthode, le problème multiobjectif est remplacé par un problème mono-objectif. Dans ce cas le décideur est supposé connaître à priori le poids de chaque objectif afin de les mélanger dans une fonction scalarisante unique.

2.5.2 Approche à postériori des préférences (recherche → décideur)

Dans cette approche, le décideur prend sa décision dans un ensemble de solutions efficaces calculées par un solveur. La qualité de la décision dépend du choix de la méthode de résolution car, celle-ci va devoir donner un ensemble de résultats le plus représentatif possible de l'espace des objectifs.

2.5.3 Approche intéractive (recherche$\longleftrightarrow$décideur)

C'est une approche qui consiste en une alternance de deux types d'étapes.

a) L'étape de calcul exécutée par l'analyste

b) Les compromis fournis par la première étape sont présentés au décideur qui réagit surtout en apportant des informations complémentaires sur les préférences. Cette quantité d'information est injectée dans le modèle utilisé et permet de construire de nouveaux compromis. Le processus continu jusqu'à satisfaction du décideur ou après la satisfaction d'un certain test d'arrêt indiquant l'étape finale (comme le nombre d'itérations fixé par le décideur).

2.6 Quelques méthodes d'optimisation multiobjectif

Reprenons le problème multiobjectif (2.1)

$$\begin{cases} \text{" max "} z(x) = (z_1(x), z_2(x), \ldots, z_k(x)) \\ \textit{sous les contraintes} \\ x \in \mathbb{S} \end{cases}$$

Tout algorithme déterminant l'ensemble E des points efficaces passent par les points suivants:

- Obtention d'une base efficace initiale ou terminer la méthode avec $E = \varnothing$ s'il n'existe aucune solution.
- Calcul des points efficaces extrêmes ou les arrêtes efficaces.

2.6.1 Méthode de la somme pondérée

Cette méthode qui est décrite dans [8, 23] consiste à associer un poids à chaque objectif. Ce poids représente l'importance relative que le décideur attribue à l'ob-

jectif. Cela réduit le problème multiobjectif à un problème mono-objectif

$$\begin{cases} \max \sum_{i=1}^{k} \lambda_i z_i(x) \\ sous\ les\ contraintes \\ x \in \mathbb{S} \end{cases} \tag{2.2}$$

où $\lambda \in \Lambda = \left\{ \lambda \in \mathbb{R}^k \mid \lambda_i \geq 0 \text{ et } \sum_{i=1}^{k} \lambda_i = 1 \right\}$.

Théorème. *(Geoffrion [16]) a) si $x^* \in S$ est une solution optimale pour (2.2), x^* est une solution efficace pour (2.1)*
b) si x^* est efficace pour (2.1) et $z(S)$ est un ensemble convexe, il existe $\lambda \in \Lambda$ tel que x^* est solution optimale de (2.2).

Cette procédure est d'une grande efficacité et très simple à mettre en œuvre mais se heurte à quelques problèmes tels que:
- Le choix des poids
- La compensation entre les différents objectifs .

2.6.2 Méthode lexicographique

Les objectifs sont préablement rangés par ordre d'importance par le décideur. Ensuite, l'optimum est obtenu en maximisant tout d'abord la fonction objectif la plus importante puis la deuxième et ainsi de suite (voir [8, 23]).

Soient donc les fonctions objectifs z_i avec $i = 1,...,k$. Supposons que $z_1 > z_2 > z_3 > ,..., > z_k$.

Nous résolvons le programme $\max_{x \in S} z_1(x)$. Soit x_1^* la solution optimale trouvée avec $z_1^* = z_1(x_1^*)$.

Nous résolvons le nouveau problème $\max_{x \in S} z_2(x)$ augmenté de la contrainte $z_1(x) = z_1^*$. Soit x_2^*, la solution optimale de ce problème.

A la ième itération on résout le problème $\max_{x \in S} z_i(x)$ augmenté des contraintes $z_1(x) = z_1^*, z_2(x) = z_2^*, ..., z_{(i-1)}(x) = z_{(i-1)}^*$.

La procédure est répétée jusqu'à ce que tous les objectifs soient traités. La solution obtenue à l'étape k sera la solution efficace du problème (2.1).

Le risque essentiel de cette méthode est la grande importance attribuée aux objectifs classés en premier. La meilleure solution z_i^* trouvée pour l'objectif le plus

important va faire converger l'algorithme vers une zone restreinte de l'espace d'état et enfermer les points dans cette zone.

2.6.3 Méthode Goal Programming

Le décideur fixe un but $T_\imath$ à atteindre pour chaque objectif $z_\imath$. Ces valeurs sont ensuite ajoutées au problème comme des contraintes supplémentaires. La nouvelle fonction objectif est modifiée de façon à minimiser la somme des écarts entre les résultas et les buts à atteindre [8, 23]. Le problème résultant est:

$$\min_{x \in S} \sum |z_\imath(x) - T_\imath \mid \tag{2.3}$$

$T_\imath$ représente la valeur à atteindre pour le ième objectif.

La méthode est facile à mettre en oeuvre mais la définition des poids et des objectifs à atteindre est une question délicate qui détermine l'efficacité de la méthode. Cette méthode a l'avantage de fournir un résultat même si un mauvais choix initial a conduit le décideur à donner un ou plusieurs but(s) $T_\imath$ non réalisable(s).

2.6.4 Méthode min-max

Cette méthode qui utilise la distance de Tchebycheff [8, 23] est assez proche de la Méthode Goal Programming. Elle minimise le maximum de l'écart entre un objectif et son but associé par le décideur

$$\min_{x \in S} \max_{\imath=1,\ldots,k} |z_\imath(x) - z_\imath^* \mid \tag{2.4}$$

Ou

$$\min_{x \in S} \max_{\imath=1,\ldots,k} \lambda_\imath \, |z_\imath(x) - z_\imath^* \mid \tag{2.5}$$

dans le cas de la distance pondérée de Tchebycheff. Ici $z_\imath^*$ désigne la ième composante du point idéal ou d'un point de référence.

Théorème. *(Bowman [9]) Un point $x^* \in S$ est une solution efficace pour (2.1) si et seulement si x^* est solution optimale unique du problème (2.4) ou (2.5).*

2.6.5 méthode ϵ-contrainte

Cette méthode est basée sur la maximisation d'un objectif $z_\imath$ en considérant que les autres objectifs $z_\jmath$ avec $\jmath \neq \imath$ doivent être supérieurs à une valeur $\epsilon_\jmath$ (voir

[8, 23]). En général, l'objectif choisi est celui que le décideur souhaite optimiser en priorité

$$\min_{x \in S} z_\jmath(x) z_\jmath \geq \epsilon_\jmath, \quad \forall \jmath \neq \imath \tag{2.6}$$

Le décideur peut ensuite réitérer ce processus sur un objectif différent jusqu'à ce qu'il trouve une solution satisfaisante.

Théorème. *(Soland [24]) Un point $x^* \in S$ est une solution efficace pour le problème (2.1) si et seulement si x^* est solution optimale unique du problème (2.6).*

Remarque 2.2. *Le lecteur intéréssé peut aussi jeter un coup d'oeil sur la méthode d'Ecker et Kouada [13], la méthode STEM de Benayoun [7], la méthode de Vanderpooten [26] ainsi que la théorie de l'utilité mult-attibuts [19,14].*

Chapitre 3

Optimisation linéaire floue

3.1 Introduction

Les connaissances sur lesquelles se base le raisonnement humain sont presque toujours entachées d'une quantité d'incertitude et d'imprécision. Cette caractéristique est en réalité inhérente à l'homme. Aucun de nos sens, ni instruments de mesures ne nous permet de décrire avec justesse ce que nous percevons du monde extérieure. Ces connaissances on été prises en considèration en 1965 par Lotfi Zadeh [27], qui a introduit la logique floue à partir de la généralisation de la théorie des ensembles classiques.

La théorie floue a l'avantage d'introduire la notion de degré de confiance dans la vérification d'une condition d'être dans un état autre que vrai ou faux. En effet un élément peut appartenir en partie à un sous-ensemble. Son degré d'appartenance est décrit par une valeur comprise entre 0 et 1.

Dans ce qui suit, nous décrivons rapidement les fondements de la théorie des ensembles flous tels qu'ils sont donnés dans [1].

3.2 Définitions

3.2.1 Ensemble flou:

Soit X un ensemble de référence ou univers et x un élément quelconque de X. On dit qu'une partie $\tilde{A}$de l'ensemble de référence X est un ensemble flou lorsqu'elle est définie de la manière suivante: $\widetilde{A} = \{(x,\mu_{\widetilde{A}}(x)),x \in X\}$
où la fonction $\mu_{\widetilde{A}} : X \rightarrow [0,1]$ dite fonction d'appartenance est définie par:

- $\mu_{\widetilde{A}} = 0$ Lorsque x n'appartient pas à $\widetilde{A}$ de façon certaine.

– $0 < \mu_{\widetilde{A}} < 1$ Lorsque x n'appartient partiellement à $\widetilde{A}$.

– $\mu_{\widetilde{A}} = 1$ Lorsque x appartient à $\widetilde{A}$ de façon certaine.

Remarque 3.1. *Plus la valeur $\mu_{\widetilde{A}}(x)$ est proche de* 1*, plus que x appartient à $\widetilde{A}$.*

Exemple

Si X désigne une communauté donnée et que le sous-ensemble $\widetilde{A}$ de X est défini par la caractéristique d'être vieux, on a trois possibilités selon le critère de sélection appliqué à l'univers X :

$\mu_{\widetilde{A}}(x) = 0 \ \forall x \in X$; c'est-à-dire que l'élément x ne satisfait pas du tout la propriété vague sous-entendue par $\widetilde{A}$: la communauté ne compte aucun vieux parmi ses membres.

$\mu_{\widetilde{A}}(x) = 1 \ \ \forall x \in X$; c'est-à-dire que x satisfait pleinement la propriété vague définie par $\widetilde{A}$: tous les membres de la communauté sont vieux ($\widetilde{A} = X$).

$0 < \mu_{\widetilde{A}}(x) < 1$; le degré d'appartenance $\mu_{\widetilde{A}}(x)$ est une valeur intermédiaire entre 0 et 1.

Vieux et jeune sont deux attributs mutuellement exclusifs, définissant les deux sous-ensembles complémentaires $\widetilde{A}$ et $\overline{\widetilde{A}}$ avec $\mu_{\overline{\widetilde{A}}}(x) = 1 - \mu_{\widetilde{A}}(x)$

3.2.2 Noyau

Le noyau de $\widetilde{A}$ noté $Noy(\widetilde{A})$ est l'ensemble des éléments de X pour lesquels la fonction d'appartenance de A vaut 1:

$$Noy(\widetilde{A}) = \{x \in X / \mu_{\widetilde{A}}(x) = 1\}$$

3.2.3 Support

Le support de $\widetilde{A}$ noté $supp\widetilde{A}$ est la partie de X sur laquelle la fonction d'appartenance de $\widetilde{A}$ n'est pas nulle.

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

3.2.4 Coupe de niveau α

Une coupe de niveau α d'un ensemble flou $\widetilde{A}$ notée $\widetilde{A}^{\alpha}$ est l'ensemble classique (ou ordinaire) définie par

$$A^{\alpha} = \{x \in X / \mu_{\widetilde{A}}(x) \geq \alpha\} \ \ pour \ \ \alpha \in]0,1]$$

3.2.5 Hauteur

La hauteur notée $h(A)$ d'un sous-ensemble flou $\widetilde{A}$ de X est la plus grande valeur prise par sa fonction d'appartenance

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

Remarque 3.2. *Le sous-ensemble flou $\widetilde{A}$ de X est normalisé si sa hauteur $h(\widetilde{A})$ égale à 1.*

Exemple : Soientt $X = [0.35]$ (l'ensemble des âges) et $\widetilde{A}$ le sous-ensemble de X des âges jeunes défini par

$$\mu_{\widetilde{A}}(x) = \begin{cases} 1 & si \quad x \in [20{,}30] \\ 0 & si \quad x \geq 35 \quad et \quad x \leq 15 \\ \alpha & si \quad x \in]15{,}20[\quad et \quad x \in]30{,}35[\end{cases}$$

Le noyau de $\widetilde{A}$ est: $Noy(\widetilde{A}) = [20{,}30]$
Le support de $\widetilde{A}$ est: $supp(\widetilde{A}) =]15{,}20[\bigcup]30{,}35[$
La hauteur de $\widetilde{A}$ est: $h(\widetilde{A}) = 1$
$\widetilde{A}$ est normalisé.

3.2.6 Ensemble flou convexe

Un ensemble flou $\widetilde{A}$ est convexe si et seulement si :

$$\forall\ x_1;\ x_2 \in \widetilde{A},\ \forall \lambda \in [0{,}1] : \mu_{\widetilde{A}}(\lambda x_1 + (1-\lambda)x_2) \geq \min(\mu_{\widetilde{A}}(x_1),\ \mu_{\widetilde{A}}(x_2))$$

Autrement dit $\widetilde{A}$ est convexe si $\mu_{\widetilde{A}}$ est concave.

3.2.7 Nombre flou

Un nombre flou est un ensemble flou convexe et normalisé de $\mathbb{R}$ et dont la fonction d'appartenance $\mu_{\widetilde{A}}$ est continue par morceaux. On note un nombre flou par $\widetilde{A} = (\mathbb{R}, \mu_{\widetilde{A}})$.

3.3 exemple

$\widetilde{4}$"à peu prêt égale à 4" est le nombre flou caractérisé par la fonction d'appartenance suivante:

$$\mu_{\widetilde{A}}(x) = \begin{cases} 0 & si \quad x \leq 3 \\ (x-3) & si \quad 3 \leq x \leq 4 \\ 1 & si \quad x = 4 \\ (-x+5) & si \quad 4 \leq x \leq 5 \\ 0 & si \quad x \geq 5 \end{cases}$$

3.3.1 Nombre flou de type triangulaire

Un nombre flou noté $\widetilde{A} = (a,b,c)$ est de type triangulaire si sa fonction d'appartenance est définie par:

$$\mu_{\widetilde{A}}(x) = \begin{cases} \frac{x-\alpha}{x-a} & si \quad a \leq x \leq b \\ 1 & si \quad x = b \\ \frac{d-x}{d-b} & si \quad b < x < c \\ 0 & sinon \end{cases}$$

Autrement dit, c'est un nombre flou dont l'ensemble $\{x \in \mathbb{R}/\mu_{\widetilde{A}}(x) = 1\}$ est réduit à un point.

3.3.2 Nombre flou de type trapézoïdal

Un nombre flou noté $\widetilde{A} = (a,b,c,d)$ est de type trapézoïdal si sa fonction d'appartenance est définie par:

$$\mu_{\widetilde{A}}(x) = \begin{cases} \frac{x-a}{b-a} & si \quad a \leq x \leq b \\ 1 & si \quad b \leq x \leq c \\ \frac{d-x}{d-c} & si \quad c < x < d \\ 0 & sinon \end{cases}$$

3.4 Opérations sur les nombres flous

Les opérations sur les nombres flous sont des extensions des opérations connues sur les ensembles classiques (égalité, réunion, intersections,...etc).

3.4.1 Egalité

Deux ensembles flous $\widetilde{A}$,$\widetilde{B}$ de X sont égaux si leurs fonctions d'appartenance prennent la même valeur pour tous les éléments x de X. Formellement $\widetilde{A} = \widetilde{B}$ si

et seulement si :

$$\mu_{\widetilde{A}}(x) = \mu_{\widetilde{B}}(x); \quad \forall x \in X$$

3.4.2 Complément

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

$$\mu_{\overline{\widetilde{A}}}(x) = 1 - \mu_{\widetilde{A}}(x); \quad \forall x \in X.$$

3.4.3 Inclusion

Soient $\widetilde{A}$ et $\widetilde{B}$ deux ensembles flous d'un même référentiel X.

$$\widetilde{A} \subseteq \widetilde{B} \;\; si \;\; et \;\; seulement \;\; si \;\; \mu_{\widetilde{A}}(x) \leq \mu_{\widetilde{B}}(x) \quad \forall x \in X$$

3.4.4 Union

L'union de deux ensembles flous $\widetilde{A}$ et $\widetilde{B}$ de X est le sous ensemble flou constitué des éléments de X affectés du plus grand des degrés avec lesquels ils appartiennent à $\widetilde{A}$ et $\widetilde{B}$. Formellement $\widetilde{A} \cup \widetilde{B}$ est défini par:

$$\mu_{\widetilde{A} \cup \widetilde{B}}(x) = \max(\mu_{\widetilde{A}}(x),\, \mu_{\widetilde{B}}(x))$$

3.4.5 Intersection

l'intersection de deux ensembles flous $\widetilde{A}$ et $\widetilde{B}$ de X est le sous-ensemble flou constitué des éléments de X affectés du plus petit des degrés avec lesquels ils appartiennent à $\widetilde{A}$ et $\widetilde{B}$. Formellement $\widetilde{A} \cap \widetilde{B}$ est défini par:

$$\mu_{\widetilde{A} \cap \widetilde{B}}(x) = \min(\mu_{\widetilde{A}}(x), \mu_{\widetilde{B}}(x))$$

Exemple
Soient $X = \{x_1, x_2, x_3, x_4, x_5\}$ un référentiel, $\widetilde{A}$ et $\widetilde{B}$ deux sous-ensembles flous de X donnés par:
$\widetilde{A} = \{(x_1,0.2),(x_2,0.7),(x_3,1),(x_4,0),(x_5,0.5)\}$

$\widetilde{B} = \{(x_1,0.5),(x_2,0.3),(x_3,1),(x_4,0.1),(x_5,0.5)\}$

On a:
$\overline{\widetilde{A}} = \{(x_1,0.8),(x_2,0.3),(x_3,0),(x_4,1),(x_5,0.5)\}$
$\widetilde{A} \cap \widetilde{B} = \{(x_1,0.2),(x_2,0.3),(x_3,1),(x_4,0),(x_5,0.5)\}$

$\widetilde{A} \cup \widetilde{B} = \{(x_1,0.5),(x_2,0.7),(x_3,1),(x_4,0.1),(x_5,0.5)\}$

3.5 Programmation linéaire floue

On distingue deux cas :

1. Cas où les inégalités (ou égalités) sont floues (programmation fléxible).
2. Cas où les données imprécises sont représentées par des ensembles flous (programmation robuste).

3.5.1 Programmation fléxible

Un problème fléxible est un problème linéaire dont l'objectif et les contraintes sont vaguement définies. Dans le cas d'ne maximisation, il se formule comme suit:

$$(P_{fl}) \begin{cases} \widetilde{\max} z = cx \\ A_\imath x \tilde{\theta} b_\imath \quad \imath = 1, \ldots, m. \\ x \geq 0 \end{cases}$$

avec $\tilde{\theta} \in \{\tilde{\leq}, \tilde{\geq}, \tilde{=}\}$ où $\tilde{\leq}, \tilde{\geq}, \tilde{=}$ sont les versions fléxibles (floues) de $\leq$, $\geq$, $=$ respectivement.
la notation $\sim$ désigne le fait que l'objectif et les contraintes ne sont pas précis.
Selon Zimmermann [28],le programme (P_{fl}) peut s'interpreter comme suit:

$$(P_{fl})\prime \begin{cases} cx \tilde{\leq} z_0 \\ A_\imath x \tilde{\theta} b_\imath \quad \imath = 1, \ldots, m \\ x \geq 0 \end{cases}$$

où Z_0 est un seuil fixé par le décideur. L'objectif et les contraintes sont respectivement représentés par les ensembles flous $\tilde{A}_0$ et $\tilde{A}_\imath = 1,2, \ldots m$ dont les fonctions d'appartenances $\mu_\imath = 0,1,2; \ldots m$ sont définies selon que $\tilde{\theta}$ est $\tilde{\leq}, \tilde{\geq}, \cong$.

Cas où $\tilde{\theta}$ est $\tilde{\leq}$

$$\mu_\imath(x) = \mu_\imath(A_\imath x) \begin{cases} 1 & si \quad A_\imath x \leq b_\imath \quad \imath = 1, \ldots, m \quad sontsatisfaites \\ \in]0.1] & \text{si} \quad \text{les contraintes sont faiblement violées} \\ 0 & \text{si} \quad \text{les contraintes sont fortement violées} \end{cases}$$

$$\mu_\imath(x) = \mu_\imath(A_\imath x) \begin{cases} 1 & si \quad A_\imath x \leq b_\imath & \imath = 1, \ldots, m \\ 1 - \frac{A_\imath x - b_\imath}{d_\imath} & si \quad b_\imath < A_\imath x \leq b_\imath + d_\imath & \imath = 1, \ldots, m \\ 0 & si \quad A_\imath x > b_\imath + d_\imath & \imath = 1, \ldots, m. \end{cases}$$

On définit de la même manière la fonction d'appartenance μ_0 associée à l'objectif. Où les d_i sont des constantes positives (subjectivement choisies) de violation des contraintes et de la fonction objectif.

Cas où $\tilde{\theta}$ est $\tilde{\geq}$

$$\mu_\imath(x) = \mu_\imath(A_\imath x) \begin{cases} 1 & si \quad A_\imath \geq b_\imath \quad \imath = 1,\ldots,m \quad \{satisfaite\} \\ \theta \in]0.1] & \text{si} \quad \text{les contraintes sont faiblement violées} \\ 0 & si \quad A_\imath x > b_\imath + d_\imath \quad \imath = 1,\ldots,m. \end{cases}$$

$$\mu_\imath(x) = \mu_\imath(A_\imath x) \begin{cases} 1 & si \quad A_\imath x \geq b_\imath & \imath = 1,\ldots,m \\ 1 - \frac{b_\imath - A_\imath x}{d_\imath} & si \quad b_\imath - d_\imath \leq A_\imath x < b_\imath & \imath = 1,\ldots,m \\ 0 & si \quad A_\imath x < b_\imath - d_\imath & \imath = 1,\ldots,m. \end{cases}$$

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

$$\mu_\imath(x) = \mu_\imath(A_\imath x) \begin{cases} 1 & si \quad A_\imath x = b_\imath \quad \imath = 1,\ldots,m \quad sontsatisfaites \\ \theta \in]0.1] & \text{si} \quad \text{les contraintes sont faiblement violées} \\ 0 & \text{si les contraintes sont fortement violées} \end{cases}$$

$$\mu_\imath(x) = \mu_\imath(A_\imath x) \begin{cases} 1 & si \quad A_\imath x = b_\imath & \imath = 1,\ldots,m. \\ 1 - \frac{A_\imath x - b_\imath}{d_\imath} & si \quad b_\imath < A_\imath x \leq b_\imath + d_\imath & \imath = 1,\ldots,m \\ 1 - \frac{b_\imath - A_\imath x}{d_\imath} & si \quad b_\imath + d_\imath \leq A_i x < b_\imath & \imath = 1,\ldots,m \\ 0 & si \quad A_\imath x < b_\imath + d_\imath \quad A_\imath x > b_\imath + d_\imath & \imath = 1,\ldots,m. \end{cases}$$

Résolution du programme déterministe équivalent

Par introduction d'une variable auxiliaire $\lambda = min(\mu_\imath(x), \imath = 0,1,\ldots m$, le problème $(P_{fl})\prime$ devient :

$$(P_D) \begin{cases} \max \lambda \\ \lambda \leq \mu_\imath(x) & \imath = 0,\ldots,m \\ 0 \leq \lambda \leq 1 \\ x \geq 0 \end{cases}$$

Théorème. *$x^\star$ est une solution optimale pour le programme $(P_{fl})\prime$ si et seulement si $(\lambda^\star, x^\star)$ est une solution optimale pour (P_D) avec $\lambda^\star = min(\mu_\imath(x), \imath = 0,1,\ldots m$.*

3.6 programmation robuste

Un programme robuste est un programme de la forme

$$(P_r)\begin{cases}\max cx\\ x_1\odot\tilde{A}_1\oplus x_2\odot\tilde{A}_2\oplus\ldots\oplus x_n\odot\tilde{A}_n[\subset]\tilde{b}\\ x_\jmath\geq 0 \quad \jmath=1,\ldots,n\end{cases}$$

Où $\tilde{A}_\jmath,\jmath=1,\ldots,n$ sont des sous ensembles flous de $\mathbb{R}$
$\oplus$: addition des ensembles flous
$\subset$: inclusion entre ensembles flous
L'ensemble des contraintes de P_r est représenté par

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

Théorème. *$x^0\in\mathbb{E}$ est optimal pour P_r si et seulement si $x^0\in\mathbb{E}$ est optimal pour le programme suivant:*

$$(P_r)\prime\begin{cases}\max cx\\ x_1\tilde{A_1}^{\alpha}+x_2\tilde{A_2}^{\alpha}+\ldots+x_n\tilde{A_n}^{\alpha}\subset\tilde{b}^{\alpha}, \quad \forall\alpha\in[0.1]\\ x_\jmath\geq 0 \quad ; \quad \jmath=1,\ldots,n.\end{cases}$$

3.6.1 Cas des nombres flous

Si les composantes $\tilde{a}_{\imath\jmath}$ de $\tilde{A}$ et $\tilde{b}_\imath$ de $\tilde{b}$ sont des nombres flous dont les ensembles de niveau $\tilde{a}_{\imath\jmath}^{\alpha}$ et $\tilde{b}_\imath^{\alpha}$ sont des intervalles de $\mathbb{R}$ (des compacts) alors

$$\tilde{a}_{\imath\jmath}^{\alpha}=[\underline{\tilde{a}}_{\imath\jmath}^{\alpha},\overline{\tilde{a}}_{\imath\jmath}^{\alpha}]$$

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

avec

$$\begin{cases}\underline{\tilde{a}_{\imath\jmath}}^{\alpha}=\inf\{x\in\mathbb{R}/\mu_{\tilde{a}_{\imath\jmath}}(x)\ \geq\alpha\}\\ \overline{\tilde{a}_{\imath\jmath}}^{\alpha}=\sup\{x\in\mathbb{R}/\mu_{\tilde{a}_{\imath\jmath}}(x)\geq\alpha\}\\ \\ \underline{\tilde{b}_\imath}^{\alpha}\ =\inf\{x\in\mathbb{R}/\mu_{\tilde{b}_\imath}(x)\ \ \geq\alpha\}\\ \\ \overline{\tilde{b}_\imath}^{\alpha}\ =\sup\{x\in\mathbb{R}/\mu_{\tilde{b}_{\imath\jmath}}(x)\geq\alpha\}\\ \end{cases}$$

En utilisant les opérations élémentaires de l'arithmétique des intervalles (P_r) devient $(P_r)\prime$

$$(P_r)\prime = \begin{cases} \max cx \\ [\underline{\tilde{a}}_{i1}^{\alpha}, \overline{\tilde{a}}_{i1}^{\alpha}]x_1 + [\underline{\tilde{a}}_{i2}^{\alpha}, \overline{\tilde{a}}_{i2}^{\alpha}]x_2 + \ldots + [\underline{\tilde{a}}_{in}^{\alpha}, \overline{\tilde{a}}_{in}^{\alpha}]x_n \subseteq [\underline{\tilde{b}_i}^{\alpha}, \overline{\tilde{b}_i}^{\alpha}] \\ x_j \geq 0 \quad ; \quad j = 1, \ldots, n. \end{cases}$$

$$(P_r)\prime = \begin{cases} \underline{\tilde{a}}_{i1}^{\alpha} x_1 + \underline{\tilde{a}}_{i2}^{\alpha} x_2 + \ldots + \underline{\tilde{a}}_{in}^{\alpha} x_n \geq \tilde{b}_i^{\alpha} \quad i = 1, \ldots, n. \\ \overline{\tilde{a}}_{i1}^{\alpha} x_1 + \overline{\tilde{a}}_{i2}^{\alpha} x_2 + \ldots + \overline{\tilde{a}}_{in}^{\alpha} x_n \leq \tilde{b}_i^{\alpha} \quad i = 1, \ldots, n. \\ x_j \geq 0 \quad ; \quad j = 1, \ldots, n. \end{cases}$$

$$(P_r)\prime = \begin{cases} \max cx \\ [\underline{\tilde{a}}_{i1}^{\alpha}, \overline{\tilde{a}}_{i1}^{\alpha}]x_1 + [\underline{\tilde{a}}_{i2}^{\alpha}, \overline{\tilde{a}}_{i2}^{\alpha}]x_2 + \ldots + [\underline{\tilde{a}}_{in}^{\alpha}, \overline{\tilde{a}}_{in}^{\alpha}]x_n \subseteq [\underline{\tilde{b}_i}^{\alpha}, \overline{\tilde{b}_i}^{\alpha}] \\ x_j \geq 0 \quad ; \quad j = 1, \ldots, n. \end{cases}$$

$$(P_r)\prime = \begin{cases} \max cx \\ \underline{\tilde{a}}_{i1}^{\alpha} x_1 + \underline{\tilde{a}}_{i2}^{\alpha} x_2 + \ldots + \underline{\tilde{a}}_{in}^{\alpha} x_n \geq \tilde{b}_i^{\alpha} \quad i = 1, \ldots, n. \\ \overline{\tilde{a}}_{i1}^{\alpha} x_1 + \overline{\tilde{a}}_{i2}^{\alpha} x_2 + \ldots + \overline{\tilde{a}}_{in}^{\alpha} x_n \leq \tilde{b}_i^{\alpha} \quad i = 1, \ldots, n. \\ x_j \geq 0 \quad ; \quad j = 1, \ldots, n. \end{cases}$$

Exemple : Soit le programme robuste suivant:

$$(P_r)\prime \begin{cases} \max z(x) = x_1 + \frac{1}{2}x_2 \\ \tilde{a_1}1x_1 + \tilde{a_1}2x_2 \subseteq \tilde{b_1}, \\ \tilde{a_2}1x_1 + \tilde{a_2}2x_2 \subseteq \tilde{b_2}, \\ x_1 \geq 0, x_2 \geq 0 \end{cases}$$

Avec
$a_{11} = (1|0.2),(-2|0.5),(3|0.9),(5|0.3)$
$a_{12} = (-1|0.2),(1|0.3),(4|0.8),(7|0.6)$
$a_{21} = (-2|0.2),(3|0.8),(5|0.9),(8|0.7)$
$a_{22} = (4|0.2),(3|0.5),(-6|0.9),(10|0.7)$
$b_1 = (12|0.3),(8|0.5),(5|0.8),(9|0.2)$

$b_2 = (5|0.6),(15|0.8),(4|0.1),(3|0.7)$
Supposons que $\alpha_1 = 0.5$ et $\alpha_2 = 0.7$ alors (Pr) s'écrit sous la forme :

$$(P_r)\prime = \begin{cases} maxz(x) = x_1 + \frac{1}{2}x_2 \\ [-2,3]x_1 + [4,7]x_2 \subseteq [5,8] \\ [3,5]x_1 + [-6,10]x_2 \subseteq [3,15] \\ x_1 \geq 0, x_2 \geq 0 \end{cases}$$

ou bien

$$(P_r)\prime \begin{cases} maxz(x) = x_1 + \frac{1}{2}x_2 \\ -2x_1 + 4x_2 \geq 5 \\ 3x_1 + 7x_2 \leq 8 \\ 3x_1 - 6x_2 \geq 3 \\ 5x_1 + 10x_2 \leq 15 \\ x_1 \geq 0, x_2 \geq 0 \end{cases}$$

3.6.2 Cas des nombres trapézoïdaux

On a

$$\tilde{a}_{ij} = \left(\ m_{ij} \ , \ \ n_{ij} \ , \ \ \alpha ij \ , \ \ \beta ij \ \right)$$
$$\tilde{b}_i = \left(\ m_i \ , \ \ n_i \ , \ \ \alpha i \ , \ \ \beta i \ \right)$$

Où

$$\begin{cases} \alpha ij > 0 \quad ; \quad \beta ij > 0 \\ \alpha i > 0 \quad ; \quad \beta i > 0 \end{cases}$$

(P_r) devient

$$(P_r)\prime \begin{cases} \max Z = cx \\ \sum_{j=1}^{n} m_{ij} x_j = m_i \\ \sum_{j=1}^{n} n_{ij} x_j \quad = n_i \\ \sum_{j=1}^{n} \alpha_{ij} x_j \quad = \alpha_i \\ \sum_{j=1}^{n} \beta_{ij} x_j \quad = \beta_i \\ x_j \geq 0, j = 1, \ldots, n. \end{cases}$$

Chapitre 4

méthode de résolution

4.1 Introduction

Dans ce chapitre, nous focalisons sur le problème de transport multiobjectif où les coefficients des fonctions objectifs sont des intervalles de $\mathbb{R}$. Ce problème est défini comme suit:

$$\min Z(x) = \sum_{i=1}^{m}\sum_{j=1}^{n}[C_{L_{ij}},C_{R_{ij}}]x_{ij},\ \ k = 1,\ldots,p \tag{4.1}$$

Sous les contraintes

$$\sum_{j=1}^{n} x_{ij} = a_i \quad , \qquad i = 1,\ldots,m. \tag{4.2}$$

$$\sum_{i=1}^{m} x_{ij} = b_i \quad , \qquad j = 1,\ldots,n. \tag{4.3}$$

$$x_{ij} \geq 0\ , \quad i = 1,\ldots,m,\ j = 1,\ldots,n. \tag{4.4}$$

Comme il n'existe aucune méthode qui résout directement des problèmes à coefficients intervalles sans transformation, nous commençons ce chapitre par transformer notre problème en un problème de transport multi-objectifs à coefficients réels. Les relations d'ordre qui représentent les préférences du décideur entre les c sont définies par la limite droite, la limite gauche, la limite centrale et la demi-largeur des l'intervalles. Finalement, nous résolvons notre problème en utilisant une approche floue. Avant de donner cette transformation, nous rappelons d'abord quelques opérations et relations d'ordre sur les intervalles.

4.2 Operations sur les intervalles

Toutes les opérations données dans cette section et la suite sont décrites dans [2].

On appelle intervalle $A = [a_L,a_R]$ l'ensemble

$$A = \{a \in \mathbb{R} : a_L \leq a \leq a_R\}$$

où a_L et a_R sont respectivement, la limite inférieure et la limite supérieur de A. Un intervalle peut être aussi défini par son centre et sa demi-largeur :

$$A = \langle a_C,a_W\rangle = \{a \in \mathbb{R} : a_C - a_W \leq a \leq a_C + a_W\}$$

où

$$a_C = (a_R + a_L)/2 \ \ et \ \ a_W = (a_R - a_L)/2$$

sont respectivement, le centre et la moitié de la largeur de A.

Définition 4.1. Soit $* \in (+, -, ...)$ les opérations sur les nombres réels. Ces opérations peuvent être entendues en un opérateur noté (*) entre intervalles fermés $A = [a_L,a_R], B = [b_L,b_R]$ de la manière suivante:

$$A(*)B = a * b : a \in A; b \in B$$

Proposition 4.1. *soient* $A = [a_L,a_R] et B = [b_L,b_R]$ *deux intervalles de* $\mathbb{R}$

$$A + B = [a_L,a_R] + [b_L,b_R] = [a_L + b_L,a_L + b_R]$$
$$A + B = \langle a_C,a_W\rangle + \langle b_C,b_W\rangle = \langle a_C + b_C,a_W + b_W\rangle$$
$$kA = k[a_L,a_R] = \begin{cases} [ka_L,ka_R] pour k \geq 0 \\ [ka_R,ka_L] pour k < 0 \end{cases}$$
$$kA = k\langle a_C,a_W\rangle = \langle ka_C,|k|a_W\rangle$$

où k est un nombre réel.

4.3 Relation d'ordre entre intervalles

Définition 4.2. La relation d'ordre $\leq_{LR}$ entre les intervalles $A = [a_L,a_R]$ et $B = [b_L,b_R]$ est définie par:

$$A \leq_{LR} B \quad ssi \quad a_L \leq b_L \ et \ a_R \leq b_R$$

$$A <_{LR} B \quad ssi \quad A \leq_{LR} B \ et \ A \neq B$$

Si $A \leq_{LR} B$; alors A est préféré à B

Définition 4.3. la relation d'ordre $\leq_{CW}$ entre $A = \langle a_C,a_W\rangle$ et $B = \langle b_C,b_W\rangle$ est définie par:

$$A \leq_{CW} B \ ssi \ a_C \leq b_C \ et \ a_W \leq b_W$$

If $A \leq_{CW} B$; alors A est préféré à B

$$A <_{CW} B \ ssi \ A \leq_{CW} B \ et \ A \neq B$$

Définition 4.4. La relation d'ordre $\leq_{RC}$ entre $A = \langle a_C,a_W\rangle$ et $B = \langle b_C,b_W\rangle$ est définie par:

$$A \leq_{RC} B \ ssi \ A \leq_{LR} B \ ou \ A \leq_{CW} B$$

$$A <_{RC} B \ ssi \ A <_{LR} B \ ou \ A <_{CW} B$$

Proposition 4.2.

$$A \leq_{RC} B \ ssi \ A \leq_{LR} B \ ou \ A \leq_{CW} B$$

$$A <_{RC} B \ ssi \ A <_{LR} B \ ou \ A <_{CW} B$$

4.4 Transformation du problème à coefficients intervalles en un problème à coefficients réels

La transformation utilisée dans ce document est celle décrite dans [].

Définition 4.5. $x^* \in S$est une solution efficace pour le problème (4.1)-(4.4) si et seulement s'il n'existe aucune autre solution qui satisfait
$Z(x) <_{LR} Z(x^*)$ et $Z(x) <_{CW} Z(x^*)$.

la définition (4.5) peut se simplifier comme suit :

Définition 4.6. $x^* \in S$ est une solution efficace pour le problème (4.1)-(4.4) si et seulement s'il n'existe aucune autre solution qui satisfait $Z(x) <_{RC} Z(x^*)$.

La limite supérieure $Z_R^k(x)$ de la fonction objective $Z^k(x)$ peut s'écrire comme suit :

$$Z_R^k(x) = \sum_{\imath=1}^{m}\sum_{\jmath=1}^{n} C_{C_{\imath\jmath}}^k x_{\imath\jmath} + \sum_{\imath=1}^{m}\sum_{\jmath=1}^{n} C_{W_{\imath\jmath}} x_{\imath\jmath}.$$

Le centre de la fonction objectif est défini par:

$$Z_C^k(x) = \sum_{\imath=1}^{m}\sum_{\jmath=1}^{n} C_{C_{\imath\jmath}}^k x_{\imath\jmath}$$

où $C_{C_{\imath\jmath}}^k$ est le centre de $Z^k(x)$ et $C_{W_{\imath\jmath}}^k$ est la moitié de la largeur du coefficient $C_{\imath\jmath}^k$ de $Z^k(x)$. Ainsi la recherche de la solution Pareto optimale du problème de transport multiobjectif à coefficients intervalles se ramène à la recherche de la solution Pareto optimale d'un problème de transport multi-objectifs déterministe suivant :

$$\begin{cases} \min\{Z_R^k(x), \min Z_C^k(x)\} & k = 1, \ldots, p \\ \{x_{\imath\jmath}\} \in S \end{cases} \tag{4.5}$$

Où S est l'ensemble formé par les contraintes (4.2)-(4.4)

4.5 Résolution du problème

Climaco et al. [10], Ringuest et al. [22] ont développé des méthodes pour résoudre les problèmes de transport multi-objectifs à coefficients réels. Ces dernières peuvent être appliquées au problème (4.5). Cependant, ces méthodes ne s'appliquent qu'aux problèmes de petite taille. Leur application aux problèmes de grande taille conduirait à un très grand nombre de solutions non dominées, donc le choix de la solution préférée serait très difficile.

Dans ce chapitre, nous proposons d'utiliser l'approche floue de Lushu et Lai [20] au problème (4.5). Nous obtenons une solution de compromis par une technique ordinaire de programmation linéaire, de programmation quadratique ou autre.

4.6 Evaluation marginale de chaque objectif

Afin de faciliter la compréhension de l'approche floue, nous allons numéroter les objectifs déterministes de 1 à 2p de sorte que le problème (4.5) s'écrive sous la forme :

$$(PTMO) = \begin{cases} \min Z^k(x) = \sum_{\imath=1}^{m}\sum_{\jmath=1}^{n} C_{\imath\jmath}^k x_{\imath\jmath} & , \quad k = 1, \ldots, 2p \\ \{x_{\imath\jmath}\} \in S \end{cases}$$

Etape 1: Nous considérons les objectifs Z^k un à un et nous résolvons donc 2p problèmes de transport classique en appliquant la méthode des potentiels décrite dans le chapitre 1.
soit $x^{(1)\star}$,$x^{(2)\star}$,...,$x^{(2p)\star}$ toutes les solutions optimales obtenues.
$\left(\; x^{(1)\star}, \;\; x^{(2)\star}, \;\; \ldots, \;\; x^{(2p)\star} \; \right)$ représente la solution idéale pour le problème (PTMO).

Etape 2: Nous calculons les valeurs des fonctions objectifs en tous les $x^{(k)\star}$, $\quad k = 1,\ldots,2p$ pour obtenir la matrice des gains G comme suit :

$$G = \begin{bmatrix} Z^1(x^{(1)\star}) & Z^1(x^{(2)\star}) & \ldots & Z^1(x^{(2p)\star}) \\ Z^2(x^{(1)\star}) & Z^2(x^{(2)\star}) & \ldots & Z^2(x^{(2p)\star}) \\ \vdots & \vdots & \ddots & \vdots \\ Z^{2p}(x^{(1)\star}) & Z^{2p}(x^{(2)\star}) & \ldots & Z^{2p}(x^{(2p)\star}) \end{bmatrix}$$

A partir de cette matrice, nous déterminons les nombres U_k et L_k définis par :

$$U_k = Z_k^- = \max_{1\leq j\leq 2p} \{Z^k(x^{(j)\star})\} \quad , \quad L_k = Z_k^+ = Z^k(x^{(k)\star}) \quad , \quad k = 1,\ldots,2p \;\; (4.6)$$

- L_k, $k = 1,\ldots,2p$ représentent les composantes du point idéal. Ce sont les valeurs correspondent aux éléments diagonaux de la matrice G.
- U_k représentent les composantes du point anti-idéal de l'objectif k. Ce sont les valeurs maximales sur les lignes de la matrice G.

Etape 3: On définit l'évaluation marginale $\phi_k(x^{(k)})$ qui correspond au k-ième objectif :

$$\phi_k(x^{(k)}) = \begin{cases} 1 & si \quad Z_k(x^{(k)}) \leq L_k \\ \frac{Z_k(x^{(k)}) - U_k}{L_k - U_k} & si \quad L_k < Z_k(x^{(k)}) < U_k \\ 0 & si \quad Z_k(x^{(k)}) \geq U_k \end{cases} \qquad (4.7)$$

$\phi_k(x^{(k)})$ nous indique à quel degré, la solution obtenue pour chaque objectif est proche du point idéal.

4.7 Evaluation globale de tous les objectifs

Après avoir calculé toutes les évaluations marginales $\phi_k(x^{(k)})$,$k = 1,\ldots,2p$, il faut déterminer une évaluation globale de tous les objectifs. Ainsi, nous déterminons $\mu : S \rightarrow [0,1]$ la fonction qui indique à quel degré chaque décision $x \in S$ satisfait tous les objectifs. $\mu(x) \in [0,1]$ est de degré subjectif de compatibilité de $x \in S$

avec la solution idéale $\left(\ x^{(1)\star},\ \ x^{(2)\star},\ \ \ldots,\ \ x^{(2p)\star}\ \right)$.
Lushu et Lai [20] supposent que la subjectivité du décideur apparait dans son jugement sur l'importance de chaque objectif. Elle est donnée par un ensemble de poids qui sont normalisés

$W = \left(\ w_1,\ \ x_2,\ \ \ldots,\ \ w2p\ \right)$ tels que $\sum_{k=1}^{2p} w_k = 1$ et $w_k \geq 0$ $(k = 1,\ldots,2p)$
Les poids sont fixés par le décideur lui même. Après la détermination des poids, Les auteurs cherchent l'opérateur d'agrégation approprié $\Phi_W : [0,1]^k \rightarrow [0,1]$ tel que

$$\forall x \in S, \quad \mu(x) = \Phi_W \left(\ \phi_1(x),\ \ \phi_2(x),\ \ \ldots,\ \ \phi_{2p}(x)\ \right)$$

Une famille importante des opérateurs d'agrégation que nous pouvons utiliser est représentée par :

$$M_W^{\alpha}(a_1,a_2,\ldots,a_{2p}) = (\sum_{\imath=1}^{2k} w_\imath a_\imath^{\alpha})^{\frac{1}{\alpha}} \quad avec\ 0 <\mid \alpha \mid< +\infty$$

Parce qu'il couvre un large eventail d'opérateurs d'agrégation souvent utilisés dans le domaine de prise de décision. Il inclut Par exemple :

1. la moyenne arithmétique pondérée $(\alpha = 1)$

$$M_W^1(a_1,a_2,\ldots,a_{2p}) = \sum_{\imath=1}^{2p} w_\imath a_\imath$$

2. la moyenne quadratique pondérée $(\alpha = 2)$

$$M_W^2(a_1,a_2,\ldots,a_{2p}) = (\sum_{\imath=1}^{2p} w_\imath a_\imath^2)^{\frac{1}{2}}$$

3. La moyenne géométrique pondérée $(\alpha \rightarrow 0)$:

$$M_W^0(a_1,a_2,\ldots,a_{2p}) = \prod_{\imath=1}^{2p} a_\imath^w$$

4. La moyenne disjonctive $(\alpha \rightarrow \infty)$:

$$M^{\infty} = \max_{1\leq\imath\leq 2p} a_\imath(w_1,w_2,\ldots,w_{2p} = \frac{1}{2k})$$

5. La moyenne conjonctive $(\alpha \rightarrow -\infty)$:

$$M^{-\infty}(a_1,a_2,\ldots,a_{2p}) = \min_{1\leq\imath\leq 2K} a_\imath(w_1,w_2,\ldots,w_{2k} = \frac{1}{2p})$$

Notons que $M^{(-\infty)}$ est l'opérateur d'agrégation le plus fréquemment utilisé dans les problèmes de programmation floue (voir Zimmerman [28]).

4.8 Formulation du programme de compromis flou

Après avoir choisi un opérateur d'agrégation approprié Φ_w et obtenu l'évaluation globale $\mu : S \rightarrow [0,1]$ pour tous les objectifs, nous pouvons convertir le problème de programmation multiobjectif (4.5) en un problème de programmation de compromis flou suivant :

$$\begin{cases} \max \mu(x) = (\Phi_W(\phi_1(x),\phi_2(x),\ldots,\phi_{2p}(x) \\ x \in S \end{cases} \tag{4.8}$$

Le problème de programmation de compromis flou (4.8) est un problème d'optimisation ordinaire à un seul objectif. Le théorème suivant indique que la solution optimale $x^\star$ obtenue à partir de (4.8) est une solution de compromis non dominée (efficace ou Paréto-optimale) pour le problème (PTMO).

Théorème. *Supposons que $w_k > 0$ et l'évaluation marginale $\phi_k(x)$ de l'objectif $Z_k(x)$ est strictement décroissante pour $k = 1,\ldots,2p$. Si nous utilisons l'opérateur $M_W^\alpha(0 <\mid \alpha \mid< \infty)$ pour définir l'évaluation globale, alors la solution optimale $x^\star$ du problème de programmation de compromis flou (4.8) est également une solution non dominée pour le problème (PTMO), c'est-à-dire qu'il n'existe aucune autre solution $x \in S$ telle que :*

$$Z^k(x) \leq Z^k(x^\star) \quad \textit{pour tout} \quad k = 1,2,\ldots,2p \tag{4.9}$$

et

$$Z^{k_0}(x) \leq Z^{k_0}(x^\star) \quad \textit{pour au moins} \quad k_0 \in \{1,2,\ldots,2p\} \tag{4.10}$$

Preuve : Démontrons ce théorème par contradiction. Supposons que $x^\star$ est une solution dominée du problème (PTMO) c'est-à-dire il existe une décision $x \in S$ telle que les équations (4.9) et (4.10) sont vérifiées. Puisque chaque évaluation marginale $\phi_k(x)$ est strictement décroissante alors :

$$\phi_k(x) \geq \phi_k(x^\star) \quad \textit{pour tout} \quad k = 1,2,\ldots,2p$$

$$\phi_{k_0}(x) > \phi_{k_0}(x^\star) \quad \textit{pour au moins} \quad k_0 \in \{1,2,\ldots,2p\}$$

Par conséquent, la stricte monotonie de l'opérateur $M_W^{(\alpha)}(0 <\mid \alpha \mid< \infty)$ implique que :

$$\mu(x) = M_W^{(\alpha)}((\phi_1(x),\phi_2(x),\ldots,\phi_2 p)) > M_W^{(\alpha)}((\phi_1(x^\star),\phi_2(x^\star),\ldots,\phi_2 p(x^\star)) = \mu(x^\star)$$

Cette inégalité contredit le fait que $x^\star$ est la solution optimale du problème (4.8).

4.8.1 Choix de la valeur α

D'un point de vue pratique, les plus importantes valeurs pour le paramètre α sont 1, 2 et $-\infty$. Nous allons discuter en détails ces trois cas :

cas 1: si $\alpha = 1$

$$\mu(x) = \sum_{k=1}^{2p} w_k \frac{Z^k(\{x_{ij}\}) - U_k}{L_k - U_k}$$

$$= \sum_{k=1}^{2p} w_k \frac{\sum_{i=1}^{m} \sum_{j=1}^{n} c_{ij}^k - U_k}{L_k - U_k}$$

$$= \sum_{i=1}^{m} \sum_{j=1}^{n} (\sum_{k=1}^{2} p \frac{w_k c_{ij}^k}{L_k - U_k}) x_{ij} - \sum_{k=1}^{2p} \frac{w_k U_k}{L_k - U_k}$$

Ainsi, le problème de programmation de compromis floue (4.8) est un modèle de transport classique que l'on résout avec la méthode des potentiels.

cas 2:$\alpha = 2$ on aura

$$\mu(x) = \sqrt{\sum_{k=1}^{2p} [\frac{Z^k(\{x_{ij}\}) - U_k}{L_k - U_k}]^2} = \sqrt{\frac{1}{2} y^T H y + P^T + Q}$$

Où :
$y = \left(\; x_{11}, \;\; \ldots, \;\; x_{1n}, \;\; x_{21}, \;\; \ldots, \;\; x_{1n}, \;\; \ldots, \;\; x_{m1}, \;\; \ldots, \;\; x_{mn} \; \right)$

$H = 2 \sum_{k=1}^{2p} \frac{w_k}{(L_k - U_k)^2} c_k^T c_k$

$P = -2 \sum_{k=1}^{2p} \frac{w_k U_k}{L_k - U_k} c_k^T$

$Q = \sum_{k=1}^{2p} w_k (\frac{U_k}{L_k - U_k})$

$$y = \left(\; c_{11}^{(k)}, \;\; \ldots, \;\; c_{1n}^{(k)}, \;\; c_{21}^{(k)}, \;\; \ldots, \;\; c_{1n}^{(k)}, \;\; \ldots, \;\; c_{m1}^{(k)}, \;\; \ldots, \;\; c_{mn}^{(k)} \; \right)$$

Ainsi, le problème (4.8) est équivalent à un problème de programmation quadratique.

cas 3:$\alpha = -\infty$ et $w_1 = w_2 = \ldots = w_{2p} = \frac{1}{2p}$,alors:

$$\mu(x) = M^{(-\infty)} = \left(\frac{Z^1(\{x_{ij}\})-U_1}{L_1-U_1}, \quad \frac{Z^2(\{x_{ij}\})-U_2}{L_2-U_2}, \quad \ldots, \quad \frac{Z^{2p}(\{x_{ij}\})-U_k}{L_k-U_k} \right)$$

$$= \wedge_{k=1}^{2p} \frac{Z^k(\{x_{ij}\}) - U_k}{L_k - U_k}$$

Où $\wedge$ est l'opérateur de minimisation. Si on pose

$$\lambda = \wedge_{k=1}^{2p} \frac{Z^k(\{x_{ij}\}) - U_k}{L_k - U_k}$$

Ainsi le problème de programmation floue (4.8) se ramène au problème de programmation linéaire suivant:

$$\begin{cases} \max \lambda \\ \frac{Z^k(\{x_{ij}\})-U_k}{L_k-U_k} \geq \lambda \qquad k = 1,2,\ldots,2p \\ x_{ij} \in S \end{cases}$$

4.9 Conclusion :

La procédure suivie dans ce chapitre est très efficace pour évaluer le (PTMO) à coefficients intervalles. Elle donne une solution optimale de compromis qui est aussi non dominée d'après le théorème 4.1. De plus cette solution est obtenue par des procédés très connus de la programmation mathématique.

Chapitre 5

Implémention et Application

5.1 introduction

Ce chapitre est consacré à la description de notre logiciel nommé "PPCF" développé avec le langage de programmation" MATLAB 7.12.0" pour mettre en œuvre la résolution du modèle mathématique précédent.

5.2 Présentation du language de programmation MATLAB

MATLAB est une application qui fournit un environnement de calcul matriciel simple, efficace, interactif permettant la mise en oeuvre des algorithmes développés dans le cadre des projets linpack et eispack.
MATLAB est constitué d'un noyau relativement réduit, capable d'interpréter puis d´évaluer les expressions numériques matricielles qui lui sont adressées :
Soit directement au clavier depuis une fenêtre de commande .
Soitsous forme de séquences d'expressions ou scripts enregistrées dans des fichiers texte appelés m-files et exécutées depuis la fenêtre de commandes.
Soit plus rarement sous forme de fichiers binaires appelés mex-files ou fichiers .mex générés à partir d'un compilateur C ou Fortran.

Ce noyau est complété par une bibliothèque de fonctions prédéfinies, très souvent sous forme de fichiers m-files et regroupés en paquetages ou toolboxes. A coté des toolboxes requises local et matlab, il est possible d'ajouter des toolboxes spécifiques à tel ou tel problème mathématique, Optimization Toolbox, Signal Processing Toolbox par exemple ou encore des toolboxes crées par l'utilisateur lui même. Un système de chemin d'accès ou path permet de préciser la liste des répertoires dans lesquels MATLAB trouvera les différents fichiers m-files.
MATLAB comporte un très grand nombre d'opérateurs, de commandes et de fonctions permettant de réaliser plusieurs opérations.
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.

5.2.1 Les étapes du programme

Recherche de la solution de base initiale

Le programme est le suivant

$functionx = initial(c,b,d,n,m)$

avec
c c'est la matrice des couts
b c'est le vecteur colonne des offres
d c'est le vecteur linge des demandes
n nombre de linge
m nombre de colonne

$t = 0;$

$while\ t < n * m$

$p = c(1,1);$

$k = 1;$

$l = 1;$

$fori = 1 : n$

$forj = 1 : m$

$ifp > c(i,j)$

$p = c(i,j);$

$k = i;$

$l = j;$

end

end

end

$if\ b(k) < d(l)$

$x(k,l) = b(k);$

$d(l) = d(l) - b(k);$

$b(k) = 0;$

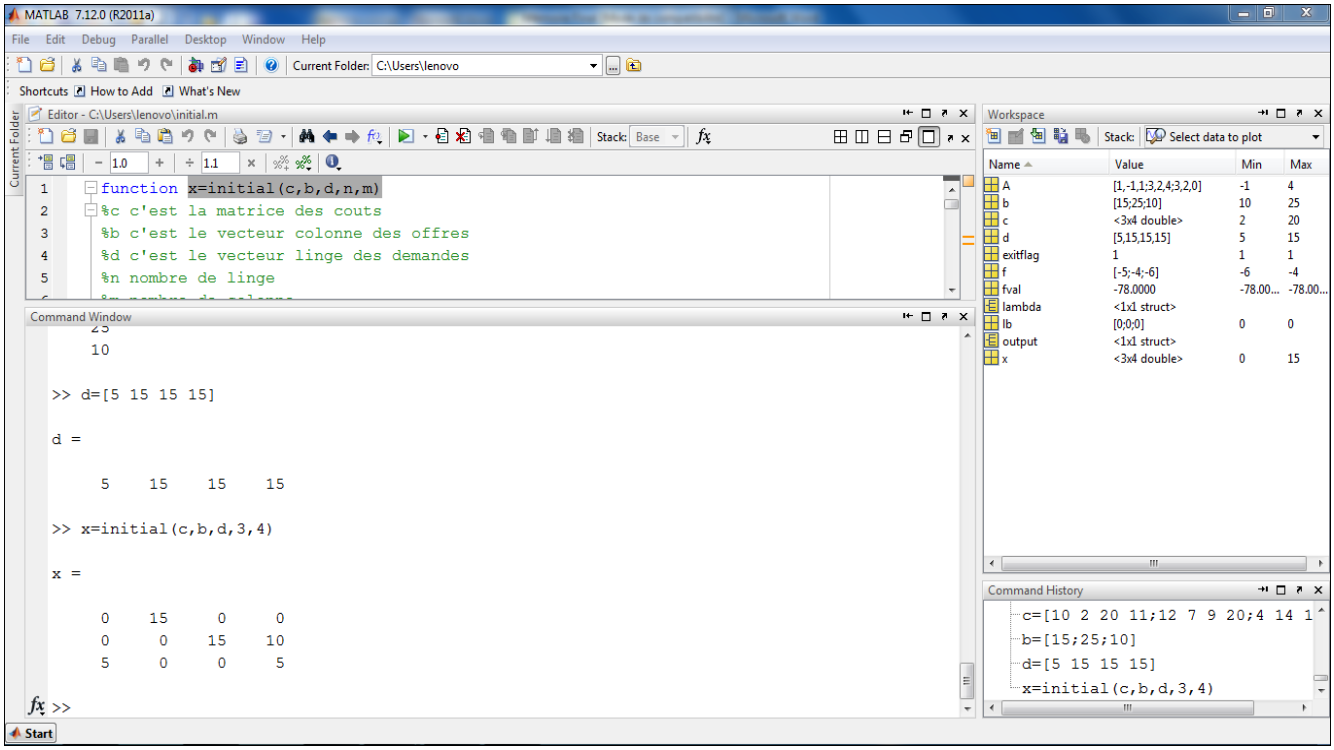
$else$

$$x(k,l) = d(l);$$
$$b(k) = b(k) - d(l);$$
$$d(l) = 0;$$

end

$$c(k,l) = 99999;$$
$$t = t + 1;$$

end



Recherche de la solution optimale

Nous avons réalisé le programme suivant : Le programme est le suivant :

function varduale (x0,c,n,m)

avec x correspond a la solution de initial trouvé (x0)

delta = *zeros*(n,m);

```
v = (9999999 : 9999999 + m);
u = (9999999 : 9999999 + n);
u(1) = 0;
t = 0;
                for ı = 1 : n
                  for ȷ = 1 : m
                    if x(ı,ȷ) = 0u(ı) ≥ 9999999v(ȷ) < 9999999
                                u(ı) = c(ı,ȷ) − v(ȷ);
                      end
                                if x(ı,ȷ) = 0u(ı) < 9999999v(ȷ) ≥ 9999999
                                  v(ȷ) = c(ı,ȷ) − u(ı);
                                end
                  end
                end
                                          t = t − 1;
        end
          for ı = 1 : n
            for ȷ = 1 : m
              delta(ı,ȷ) = u(ı) + v(ȷ) − c(ı,ȷ);
            end
          end
    ma = 0;
    t = 0;
for ı = 1 : n
  for ȷ = 1 : m
    if ma < delta(ı,ȷ)
      ma = delta(ı,ȷ);
        k = ı;
        l = ȷ;
```

$t = 1;$

end

end

end

$if t == 0$

$z1 = 0;$

$for ı = 1 : n$

$for ȷ = 1 : m$

$z1 = z1 + c(ı,ȷ) * x(ı,ȷ);$

end

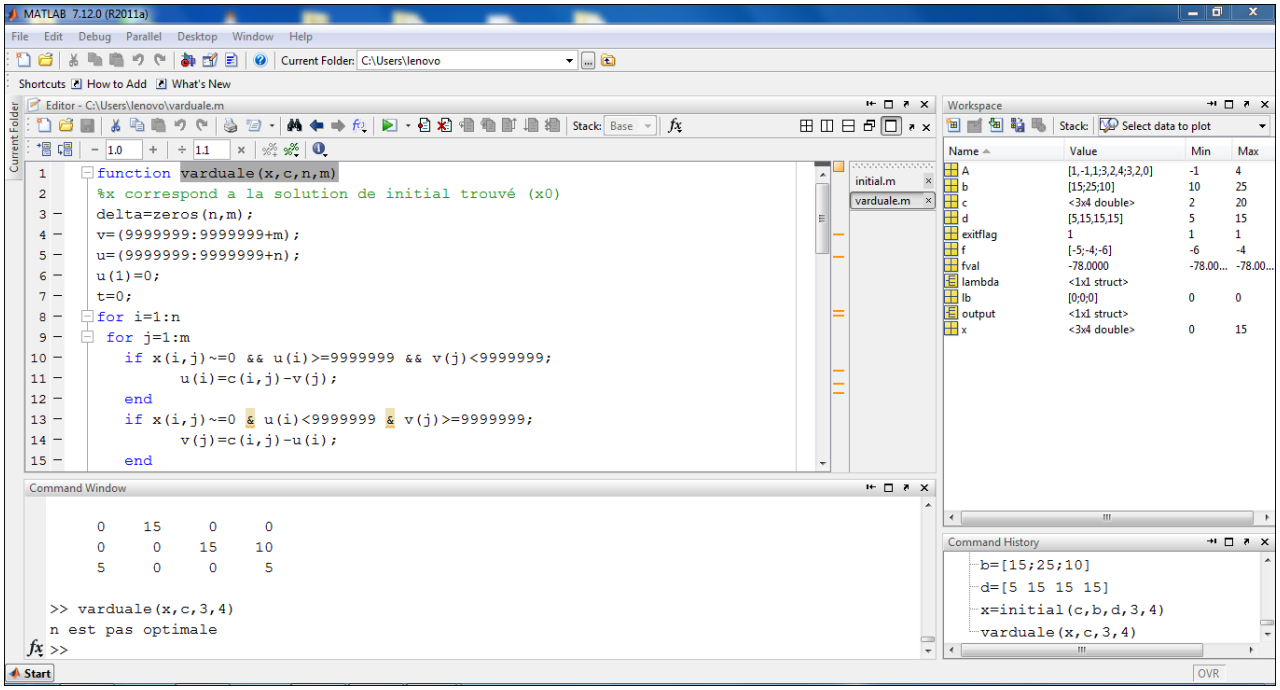
end

$x\ disp\ ('est\ une\ solution\ optimale')$

$disp('la\ valeur\ optimale\ est')z1$

$else\ x\ disp\ ('n'''est\ pas\ optimale')$

end



Changement de base

Le programme suivant nous permet de faire le changement de base et le test d'optimalité jusqu'à ce la solution optimale soit trouvée. Nous changeons quelques paramètres dans le programme précédent " varduale " devient " vardual1 " comme suit :

```
function zoptimal = vardual1(x0,c,n,m)
pas = 1;
while pas == 1
    delta = zeros(n,m);
    v = (99999 : 99999 + m);
    u = (99999 : 99999 + n);
    u(1) = 0;
          t = n + m;
       while t > 0
         for ı = 1 : n
              for ȷ = 1 : m
                 ifx(ı,ȷ) ∼= 0u(ı) >= 99999v(ȷ) < 99999
                          u(ı) = c(ı,ȷ) − v(ȷ);
                 end
                if x(ı,ȷ) ∼= 0u(ı) < 99999v(ȷ) >= 99999
                  v(ȷ) = c(ı,ȷ) − u(ı);
                  end
               end
           end
              t = t − 1;
       end
  for ı = 1 : n
    for ȷ = 1 : m
          delta(ı,ȷ) = u(ı) + v(ȷ) − c(ı,ȷ);
```

```
      end
    end
ma = 0;
 t = 0;
  for ι = 1 : n
   for ȷ = 1 : m
         if ma < delta(ι,ȷ)
           ma = delta(ι,ȷ);
                k = ι;
                l = ȷ;
                t = 1;
        end
      end
    end
    if t == 0
          z1 = 0;
     forι = 1 : n
       forȷ = 1 : m
          z1 = z1 + c(ι,ȷ) ∗ x(ι,ȷ);
      end
    end
           zoptimal = z1;
           disp('la solution optimal est')w = x
           disp('la valeur optimale est')
      pas = 0;
    else
       mi = (9999999 : 9999999 + n + m);
       x0 = x;
       z = x0;
```

```
        for ι = 1 : n
            for ȷ = 1 : m
                if x0(ι,ȷ) == 0
                z(ι,ȷ) = 9999999;
                end
            end
        end
        z(k,l) = 0;
        t = 0;
       while t < n + m
      for ι = 1 : n
            nb = 0;
         for ȷ = 1 : m
           if z(ι,ȷ) ∼= 9999999
                nb = nb + 1;
    end
end
  ifnb == 1
        for ȷ = 1 : m
            z(ι,ȷ) = 9999999;
        end
    end
end
 for ȷ = 1 : m
     nb = 0;
         forı = 1 :
          if z(ι,ȷ) ∼= 9999999
                nb = nb + 1;
           end
```

```
    end
      if  nb == 1
         for  ı = 1 : n
          z(ı,ȷ) ∼= 9999999;
         end
      end
    end
      t = t + 1;
end
       nb = 1;
     forȷ = 1 : m
      if  z(k,ȷ) ∼= 0z(k,ȷ) ∼= 9999999
           mi(nb) = z(k,ȷ);
           k2 = ȷ;
           k1 = k;
          z(k,ȷ) = z(k,ȷ) ∗ (−1);
      end
    end
     h = z(k1,k2);
     while  h ∼= 0
              for  ı = 1 : n
                 if  ı ∼= k1z(ı,k2) ∼= 9999999
                      k3 = ı;
                 end
              end
        k1 = k3;
        h = z(k1,k2);
            if  h ∼= 0
                 nb = nb + 1;
```

```
            for ȷ = 1 : m
              if ȷ ∼= k2z(k1,ȷ) ∼= 9999999
                   mi(nb) = z(k1,ȷ);
                   k4 = ȷ;
              end
            end
              k2 = k4;
              z(k1,k2) = z(k1,k2) ∗ (−1);
              h = z(k1,k2);
        end
    end
          o = min(mi);
        Forı = 1 : n
          for ȷ = 1 : m
            if z(ı,ȷ) ∼= 9999999
                 x(ı,ȷ) = z(ı,ȷ) + o;
               else
                  x(ı,ȷ) = x0(ı,ȷ);
            end
          end
        end
         for ı = 1 : n
          for ȷ = 1 : m
           if  x(ı,ȷ) < 0
             x(ı,ȷ) = x(ı,ȷ) ∗ (−1);
           end
    end
        end
              pas = 1
  end
end
```

Programme principal (PPCF)

Ce programme qui fait appel aux sous-programmes développés au départ.

$function\ ppcf(k,n,m)$

$ou\ k\ etant\ le\ nomre\ d'objectifs$

$u = zeros(1,2 * k);$

$l = zeros(1,2 * k);$

$p = zeros(1,2 * k);$

$z = 1;$

$no = 0;$

$o = 1;$

$c1 = zeros(n,m);$

$c2 = zeros(n,m);$

$b = zeros(n,1);$

$d = zeros(1,m);$

$gc = zeros(2 * k * n,m);$

$pa = 1;$

$tk = 1;$

$for\ \imath = 1 : n$

$b(\imath,1) = input\ ('donner\ la\ valeur\ de\ l\ offre :\ b =');$

end

b

$for\ \jmath = 1 : m$

$d(1,\jmath) = input('donner\ \ la\ \ valeur\ \ de\ \ la\ \ demande\ \ \ : d =');$

end

d

$rep = 1;$

$while\ \ rep == 1$

$for\ \ \imath = 1 : n$

$for\ \jmath = 1 : m$

$c1(\imath,\jmath) = input\ ('donner\ la\ borne\ superieure\ si\ il\ ya\ c'est\ non\ donner$

$directement\ la\ valeur\ de\ cout\ :\ c =');$

$gc(pa,\jmath) = c1(\imath,\jmath);$

end

$pa = pa + 1;$

end

$c1$

$p(1,z) = input\ ('donner\ le\ poids\ de\ cet\ objectif\ =\ ');$

$z = z + 1;$

$no = no + 1;$

$x = initial(c1,b,d,n,m);$

$l(1,tk) = vardual2(x,c1,n,m);$

$x = initial(-c1,b,d,n,m);$

$u(1,tk) = -vardual2(x, -c1,n,m);$

$tk = tk + 1;$

$o = input('tapez1siilyaunebornein f erieuresinontapez0 :=');$

$if\ o == 1$

$p(1,z) = p(1,z-1);$

$z = z + 1;$

$no = no + 1;$

$for\ \imath = 1 : n$

$for\ \jmath = 1 : m$

$c2(\imath,\jmath) = input\ ('donner\ la\ borne\ in f erieure\ c =');$

$gc(pa,\jmath) = (c1(\imath,\jmath) + c2(\imath,\jmath))/2;$

$c2(\imath,\jmath) = (c2(\imath,\jmath) + c1(\imath,\jmath))/2;$

end

$pa = pa + 1;$

end

```
            c2
        x = initial(c2,b,d,n,m);
        l(1,tk) = vardual2(x,c2,n,m);
        x = initial(−c2,b,d,n,m);
        u(1,tk) = −vardual2(x, − c2,n,m);
        tk = tk + 1;
    end
        rep = input('taper1silyaunautreobjectifssinontapez0 : rep =');
end
    c1 = zeros(n,m);
        p;
        u;
        l;
        gc;
        for ı = 1 : n
            tt = ı;
            for ȷ = 1 : m
              ss = 0;
                for t = 1 : no
                  ss = (p(1,t) ∗ gc(tt,ȷ))/(l(1,t) − u(1,t)) + ss;
                  tt = tt + n;
                end
                  c1(ı,ȷ) = ss;
                  tt = ı;
           end
       end
        c1;
        f = zeros(1,n ∗ m);
        t = 1;
```

```
for ı = 1 : n
    for ȷ = 1 : m
        f(1,t) = c1(ı,ȷ);
            t = t + 1;
    end
end
        a = zeros(n + m,n ∗ m);
        pas = 1;
for ı = 1 : n
    for ȷ = pas : pas + m − 1
            a(ı,ȷ) = 1;
    end
        pas = ȷ + 1;
end
        pas = 1;
for ı = n + 1 : n + m
    for ȷ = pas : m : n ∗ m
        a(ı,ȷ) = 1;
    end
    pas = pas + 1;
end
b1 = zeros(1,n + m);
t = 1;
    for ı = 1 : n
            b1(1,t) = b(ı,1);
            t = t + 1;
    end
    for ȷ = 1 : m
  b1(1,t) = d(1,ȷ);
```

```
            t = t + 1;
          end
b = b1;
lb = zeros(n * m,1);
[x,fval] = linprog(f,a,b,[],[],lb);
 xa = zeros(n,m);
 t = 1;
    for ı = 1 : n
       for ȷ = 1 : m
         xa(ı,ȷ) = x(t,1);
          t = t + 1;
      end
  end
zop = zeros(1,no);
ss = 0;
 t = 1;
 h = 0;
 s = no * n;
pa = 0;
    for  ı = 1 : s
       for  ȷ = 1 : m
            h = h + 1;
          ss = ss + x(h,1) * gc(ı,ȷ);
        end
      pa = pa + 1;
           if pa == n
           zop(1,t) = ss;
            t = t + 1;
             ss = 0;
```

$pa = 0;$

$h = 0;$

end

end

$x = xa;$

$disp\,('la\ solution\ optimal\ est\ x =')x$

$disp\,('les\ valeurs\ des\ fonctions\ objectifs\ sont\ ')zop$

5.3 Exemple d'application

Nous avons exécuté notre logiciel "PPCF" sur des données dont disposait NAFTAL dans la journée du dimanche 8 février 2009. Ces données concernent le transport de bouteilles de gaz à partir de deux dépôts vers certaines destinations (clients). Dans un premier temps, nous allons présenter les données en question puis nous donnerons les résultats trouvés par notre logiciel.

Les tableaux suivants représentent les données des deux dépôts

Le dépôt 1 (bouteilles)	Le dépôt 2 (bouteilles)
24000	15000

Tableau 5.1 : Quantités disponibles

In	Code	Q c	T1	T2	C1	C2
1	B8050	1500	[20,30]	[30,50]	[6,9]	[10,11]
2	U9743	600	[30,40]	[22,30]	[8,10]	[10,12]
3	Y7719	400	[10,20]	[18,20]	[11,12]	[13,15]
4	A1222	200	[15,25]	[17,20]	[9,10]	[12,14]
5	D3532	1500	[20,35]	[2030,]	[7,9]	[12,13]
6	U9747	1500	[52,62]	[22,36]	[6,8]	[14,16]
7	S9719	1000	[25,30]	[22,26]	[10,12]	[10,12]
8	M7707	300	[19,23]	[19,25]	[6,10]	[15,17]
9	U7940	700	[20,28]	[20,28]	[5,8]	[10,16]
10	M7998	500	[19,26]	[11,30]	[9,10]	[11,12]
11	Y7710	1000	[15,20]	[8,10]	[10,11]	[13,14]
12	M7962	2000	[20,30]	[10,15]	[13,16]	[11,12]
13	S8283	2000	[5,10]	[11,17]	[13,14]	[13,14]
14	M7966	2000	[10,14]	[40,50]	[12,13]	[10,12]
15	L8731	1000	[10,19]	[8,12]	[15,17]	[11,13]
16	D4533	1600	[12,20]	[10,18]	[14,15]	[11,14]
17	Y9708	2000	[20,25]	[11,13]	[9,10]	[10,12]
18	U9729	2000	[19,28]	[8,10]	[10,12]	[12,14]
19	L866	100	[18,26]	[10,17]	[9,11]	[11,12]
20	Y7718	1500	[20,30]	[10,16]	[9,10]	[8,10]
21	S6702	500	[30,36]	[11,19]	[13,15]	[10,11]
22	U9738	2500	[30,38]	[19,25]	[10,12]	[13,14]
23	M7948	3000	[30,39]	[20,26]	[10,11]	[8,10]
24	M7999	500	[10,20]	[22,35]	[9,10]	[10,11]
25	M7947	900	[5,10]	[30,38]	[13,14]	[9,10]
26	S6705	1600	[10,14]	[29,39]	[16,18]	[10,14]
27	B8526	1400	[12,17]	[20,34]	[10,12]	[9,10]
28	M7987	2200	[10,15]	[21,26]	[13,14]	[7,11]
29	Y9728	3000	[9,12]	[29,37]	[11,13]	[8,10]
30	Y7727	2000	[1018,]	[19,30]	[11,15]	[6,8]

Tableau 5.2 : Liste des clients

avec
Ind : indice client.
code : code client.
Q C : Quantité commandée (bouteilles).
T1 : Temps nécessaire entre le dépôt 1 et le client (en Minutes)
T2 : Temps nécessaire entre le dépôt 2 et le client (Minutes)

C1 : le coût du transport d'une bouteille du dépôt 1 vers le client en Dinars (DA)
C2 : le coût du transport d'une bouteille du dépôt 2 vers le client en Dinars (DA)

Code clients	Le nombre de bouteilles 1	Le nombre de bouteilles 2
B8050	0	1500
U9743	600	0
Y7719	400	0
A1222	200	0
D3532	1500	0
U9747	1500	0
S9719	1000	0
M7707	0	300
U7940	700	0
M7998	500	0
Y7710	1000	0
M7962	1500	0
S8283	0	500
M7966	0	2000
L8731	1000	0
D4533	1600	0
Y9708	1000	0
U9729	3000	0
L8666	100	0
Y7718	1500	0
S6702	500	0
U9738	2500	0
M7948	3000	0
M7999	0	500
M7947	0	900
S6705	0	1600
B8526	0	1400
M7987	0	2200
Y9728	0	3000
Y7727	900	1100

Tableau 5.3 Quantités transportées

avec
nombre de bouteilles 1 : est Le nombre de bouteilles transportées du dépôt 1 vers le client.

nombre de bouteilles 2 : est Le nombre de bouteilles transportées du dépôt 2 vers le client.
Les valeurs des fonctions objectif :

```
x =

  1.0e+003 *

  Columns 1 through 7

    0.0000    0.6000    0.4000    0.2000    1.5000    1.5000    1.0000
    1.5000    0.0000    0.0000    0.0000    0.0000    0.0000    0.0000

  Columns 8 through 14

    0.0000    0.7000    0.5000    1.0000    1.5000    0.0000    0.0000
    0.3000    0.0000    0.0000    0.0000    0.0000    0.5000    2.0000

  Columns 15 through 21

    1.0000    1.6000    1.0000    3.0000    0.1000    1.5000    0.5000
    0.0000    0.0000    0.0000    0.0000    0.0000    0.0000    0.0000

  Columns 22 through 28

    2.5000    3.0000    0.0000    0.0000    0.0000    0.0000    0.0000
    0.0000    0.0000    0.5000    0.9000    1.6000    1.4000    2.2000

  Columns 29 through 30

    0.0000    0.9000
    3.0000    1.1000

les valeures des fonctions objectifs sont

zop =

  1.0e+006 *

    1.3178    1.1374    0.4495    0.4117
```

Le temps optimal pour le transport varie entre :
La valeur maximale=1317800 Mn.
La valeur minimale =1137400 Mn.
Le cout de transport varie entre :
La valeur maximale=449500 DA.
La valeur minimale =411700 DA

Conclusion

Les exigences des entreprises de transport sont souvent nombreuses et diversifiées comme le respect des délais de livraison, la minimisation des couts de transport, la maximisation du profit, ainsi que la satisfaction des demandes et l'écoulement des stocks. Ces entreprises doivent donc trouver un modèle de transport qui peut alimenter le marché avec leurs produits, qui satisfait le besoin des clients et qui tient compte de facteurs extérieurs non maitrisés tels que le taux de pluviométrie, du taux de change, ...

Pour toucher cet objectif, nous avons proposé, dans ce mémoire, un modèle de transport multiobjectif dont les coefficients des objectifs sont des intervalles. Pour la résolution de notre problème, nous avons appliqué la méthode de Lushu et Lai qui est basée sur la combinaison de la théorie des ensembles flous et les techniques de la programmation multiobjectif. Cette étude s'est soldé par la conception d'un programme appelé *PPCF*: Programme de Programmation de Compromis Floue. La réalisation de ce dernier est faite à l'aide du logiciel MATLAB. Cette méthode peut aussi s'appliquer au cas où les demandes et les offres dépendent de l'incertitude. Ces cas se ramènent facilement au cas étudié.

Bibliographie

[1] Aiche F., la programmation linéaire multi-objectif floue et stochastique, Thèse de doctorat, Université Mouloud Mammeri de Tizi-Ouzou (2013).

[2] Alefeld G., Herzberger J., Introduction to Interval computations. Academic Press New York (1983).

[3] Balas E and P L Ivanescu, On the Generalized Transportation Problem. Management Science 11, 188-203 (1964).

[4] Balinski M L and R E Gomory, A primal Method for Assignement and Transportation Problem. Management Science 10 (3) 578-593, April (1964).

[5] Bellahcene F., problème de répartition équilibrée à indices multiples en environnement aléatoire. Sciences et Technologie A (22), 17-21 (2004).

[6] Bellahcene F., problème de transport à indices multiples. Thèse de Magister, Université Mouloud Mammeri de Tizi-Ouzou, (1999).

[7] Benayoun R., De Montgolfier J., Tergny J., and Larichev O., Linear Programming with Multiple Objective Functions, Step Method (STEM), Mathematical Programming 1 (2), 366-373 (1971).

[8] Berro A., Optimisation multiobjectif et stratégies d'évolution en environement dynamique, thèse de doctorat en informatique, Toulouse I,(2001).

[9] Bowman V.J., On the Relationship of the Tchebytcheff Norm an the Efficient Frontier of Multiple-Criteria Objectives, in Thiriez H., Zionts S. (eds), MCDM, Springer-Verlag, Berlin, 76-85 (1976).

[10] Climaco J.N., Antunes C.H., Alves M.J., Interactive decision support for multiopbjective transportation problems. European Journal of Operations Research 65, 58-67 (1993).

[11] Cordeau J F, Gendreau M, Laporte G, Potvin J Y and Semet F, Aguid to vehicule routing heuristics. Jornal of the Operational Researche, (2002).

[12] Deepika R., Fuzzy programming technique for solving different types of multiobjective transportation problem. Thesis of School of Mathematics and Computer Applications, Thpar University, India (2010).

[13] Ecker J.G, Kouada I. A., Finding all efficient extreme points for multiple objective programs, Mathematical Programming 14, 249-261 (1978).

[14] Farquahar P.H. Utility assesment methods, Management Sciences 30, 1283-1300 (1984).

[15] Fisher M L and Jaikumur R, A generalised assignment heuristic for vehicule routing. Netwrk 11, 109-124 (1981).

[16] Geoffrion A.M, Proper efficiency and the theory of vector maximization, Journal of Mathematical and Advances Algebra 22, 618-630 (1968).

[17] Glover F D, Karney D, Klingman and R Russell, Solving Singly Constrainted Transshipment Problems. Transportation Science 12(4), 277-297 (1988).

[18] Hwang CL and Yoon KS, Multiple attribute decision making: methods and applications. Berlin: Springer (1981).

[19] Keeeny R.L., Raiffa H., Dcisions with multiples objectives: Preferences and value tradeoff, Editions Cambridge University Press (1983).

[20] Lushu Li and K K Lai, A fuzzy approach to the transportation problem. Computers and Operations Research (27), 43-57 (2000).

[21] Michel Sakarovitch, Optimisation combinatoire. Ermann Editeurs des sciences et ses arts (1984).

[22] Miettinen K.M, Nonlinear Multiobjective Optimization. Kluwer's International Series (1999).

[23] Ringuest J L and Rinks D B, Interactive solutions for the linear multiobjective transportation problem. European Journal of Operations Research 32, 96-106 (1987).

[24] Soland R.M, Multicriteria optimization : A general characterization of efficient solutions, Decision Science 10 (1), 26-38 (1979).

[25] Teghem J., Programmation linéaire, Editions de l'université libre de Bruxelles (2003).

[26] Vanderpooten D., Vincke Ph, Description and Analysis of Some Representative Interactive Multicriteria Procedures, Mathematical and Computer Modeling 12, 1221-1238 (1989).

[27] Zadeh L.A., Fuzzy sets. Information Control 32, 9-27 (1988)

[28] Zimmermann H J, Fuzzy set theory and its applications. Boston: Kluwer Academic Publishers (1996).