OPTIMISATION ECONOMIQUE DE LA PRODUCTION D'ETHANOL DE BLE PAR LE CHOIX DES VARIETES DE BLES, A L'AIDE DE LA PROGRAMMATION LINEAIRE FLOUE.

1. INTRODUCTION 2
2. LA PROGRAMMATION LINEAIRE: DEFINITIONS ET NOTATIONS. 3
3. PROGRAMMATION LINEAIRE FLOUE 3
3.1 Flexible Linear Programming (FLP) *4*
3.1.1 Etat de l'art 4
3.1.2 Fuzzification de ≤ 5
3.1.3 Fuzzification du max 5
3.1.4 Approche Symétrique de Zimmermann 6
3.1.5 Critiques 7
3.1.6 Application 7
3.2 Possibilistic Linear Programming (PLP) *8*
3.2.1 Etat de l'art 8
3.2.2 Choix du nombre flou: trapézoïdal. 9
3.2.3 Arithmétique des nombres flous. 10
3.2.4 Comparaison des nombres flous. 11
3.2.5 Programmation linéaire avec des nombres flous. 14
3.2.6 Optimisation 14
3.2.7 Résultats 15
4. LE MODELE: CARACTERISTIQUES TECHNIQUES 17
5. CONCLUSIONS 17
6. BIBLIOGRAPHIE 18
7. ANNEXES 19

1. Introduction

Ce rapport concerne un stage de DEA en Intelligence Artificielle à l'Université de Paris VI, qui se déroula à la station d'Economie et Sociologie Rurales de l'Institut National de Recherches Agronomiques (INRA) situé à Grignon. Une grande partie des scientifiques de cette station travaille dans la modélisation économique et utilise le plus souvent la programmation linéaire classique pour parvenir à des résultats corrects à faible coût (temps, information, machines, etc.). La préoccupation de ces scientifiques a été alors de trouver alors de nouvelles techniques de programmation mathématique, qui permettent l'amélioration de leurs modèles de manière assez simple. Une des pistes était la logique floue. Le responsable de la station Jean Claude Sourie proposa un stage sur la programmation linéaire floue, qui s'appuyait sur une application réelle que nous allons présenter.

Le gouvernement français sous la forme de l'ADEME[1] a créé le projet AGRICE[2] dont un des ses programmes, concerne les nouvelles sources d'énergie, en particulier les bio-carburants. Il se trouve que la production de ces bio-carburants est pour le moment beaucoup plus élevée que celle des carburants traditionnel. La station d'économie rurale de l'INRA a alors été appelée pour estimer jusqu'à combien on peut réduire le prix de ce nouveau type d'essence.

En ce qui concerne le stage, on a cherché à optimiser du point de vue économique la transformation des blés en éthanol. Plus précisément, il s'agissait de vérifier, par un choix judicieux des variétés et des techniques de production du blé, s'il est possible de diminuer le coût de l'éthanol. Pour mener à bien ce projet, la modélisation du processus biochimique industriel était indispensable. Il fallait modéliser la gestion dynamique de chaînes de production industrielle, en prenant en compte différents types de contraintes liées à la qualité de production, aux moyens existants (capacités, débits), au temps, etc. Pour mener à bien cette tâche, la programmation linéaire s'avère un outil tout à fait adapté.

Pour faciliter la résolution du problème, on a procédé par étapes, on a commencé tout d'abord par construire un modèle classique de programmation. Les données concernant les différentes variétés de blés prisés en compte et les explications techniques relatives au processus industriel réel de fermentation ont été fournies par Jean Philippe Leygue de L'ITCF[3]. Mais il se trouve que, d'une part on s'est appuyé sur des données provenant d'une expérimentation biochimique et donc entachées d'imprécisions et incertitudes, et d'autre part les contraintes réelles sont moins figées et rigides que dans la modélisation classique. C'est alors que l'introduction de la logique floue permet une meilleure représentation du monde réel. On a donc dans un deuxième temps réalisé une recherche sur l'état de l'art, qui nous a permis de choisir deux approches de "fuzzification". Finalement on a construit deux modèles flous, utilisant le modèle classique comme base.

---

[1] ADEME: Agence de l'Environnement et de la Maîtrise de l'Espace.

[2] AGRICE: Agriculture pour la Chimie et l'Energie.

[3] ITCF: Institut de Techniques Céréalières et Fourrages.

2. La programmation linéaire: définitions et notations.

La programmation linéaire vise à réaliser l'affectation efficace de ressources limites à des activités, tout en tenant compte de l'objectif à atteindre, comme par exemple maximiser les profits ou minimiser les coûts. Le modèle classique de programmation linéaire peut se poser de la manière suivante:

maximiser	$z = f(c,x) = c^T x$	***Objectif***
tel que	$Ax \leq b$	
et	$0 \leq x$	Contraintes
avec	$c, x \in R^n, b \in R^m, A \in R^{mxn}$,	

Ici c est le vecteur des coefficients de profit de la fonction objectif et b est le vecteur du total de ressources disponibles. x est le vecteur de variables de décision (ou alternatives), et A est la matrice des coefficients techniques.

Il est important de rappeler ici, qu'il existe des techniques extrêmement efficaces pour résoudre ce type de problèmes. Et donc toute formulation floue à intérêt à être ramenée à cet énoncé pour pouvoir être résolue de manière efficace et économique.

3. Programmation linéaire floue

Dans cette partie de ce papier, je présente une esquisse du savoir-faire actuel de la Programmation Linéaire Floue. Pour pouvoir condenser les idées d'un domaine si vaste, on est obligé de réaliser une classification. Plusieurs auteurs ont déjà traité ce problème: Zimmermann [25,26, 1985; BM64, 1987], Slowinski [18, 1986], Wierzchon [22, 1987], Luhandjula [14, 1989], Lai and Hwang [12]. D'abord Zimmermann classifia la programmation mathématique floue en modèles symétriques et non-symétriques. Luhandjula ensuite classifia la programmation mathématique floue en programmation flexible, en programmation avec paramètres flous et en programmation stochastique floue. Enfin les autres auteurs ont essayé de grouper les différents problèmes, en décrivant ce qui a été rendu flou. [2]

Deux grandes approches se dégagent de l'ensemble des travaux existants: D'une part, il y a une série des travaux qui centrent leurs efforts de fuzzification sur les contraintes et les objectifs et d'autre part on a les travaux qui s'appuient sur la théorie possibiliste pour rendre flou chacun des coefficients (des nombres). On appellera respectivement les approches Flexible Linear Programming (FLP[4]) et Possibilistic Linear Programming (PLP). On remarquera que cette classification permet aussi de séparer les modèles en deux grands groupes: ceux qui interprètent les degrés

Commentaire [G2LU1] :

[4] FLP est généralement l'abréviation de Fuzzy Linear Programming. Je préfère le terme "flexible" au terme "flou", car le deuxième prête à confusion. On remarquera que les auteurs Yazenin [23] et Buckley [3] font aussi la distinction entre FLP et PLP.

d'appartenance, comme étant des degrés de "préférence" (les FLP) et ceux qui leur donnent une interprétation possibiliste (les PLP).

3.1 Flexible Linear Programming (FLP)

Ici on s'appuie sur la notion de préférence. Préférer veut dire se déterminer en faveur d'une chose plutôt que d'une autre, aimer mieux, estimer davantage. Et donc les préférences sont des marques particulières, justifiées ou non, d'affection ou d'honneur qu'on accorde à quelqu'un ou quelque chose. Pour les représenter on va ici utiliser une fonction d'appartenance. Cette fonction va traduire de façon numérique les souhaits et préférences du décideur. Le décideur est complètement satisfait si et seulement si la fonction d'appartenance vaut 1; il n'est pas satisfait du tout si et seulement si la fonction d'appartenance vaut 0. On pourra alors utiliser toutes les valeurs entre [0,1] pour représenter des positions intermédiaires.

3.1.1 Etat de l'art

On peut, dans le cas particulier de la programmation linéaire flexible, classifier les approches, comme l'a fait Zimmermann [27], en symétriques et non symétriques.

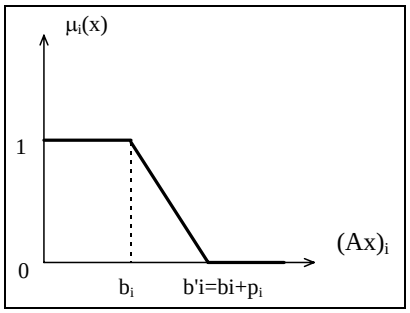
D'une part les modèles symétriques s'appuient essentiellement sur la définition de la décision proposée par Bellman et Zadeh [1]. Ils considèrent que la décision est le résultat de la confluence des objectifs et contraintes, toutes les deux considérées sur le même plan. Dans cette catégorie, on trouve les travaux de Zimmermann [24], qui applique strictement la définition de Bellman et Zadeh [1] et pour un type particulier de fonction d'appartenance retombe sur un programme linéaire classique. D'autre part, les modèles non symétriques cherchent à déterminer d'abord l'espace flou de décision et puis utilisent une fonction non floue qui maximise la décision. Ainsi Verdegay [20] ne présente qu'une façon de fuzzifier des contraintes, en effectuant l'optimisation de manière classique. Werners [21] propose une méthode qui permet, à partir des contraintes floues, de déterminer une fonction d'appartenance pour tomber à la fin sur un modèle symétrique. Chana [4] applique une approche très similaire à la précédente: premièrement il travaille seulement avec les contraintes, pour ensuite aider le décideur à estimer ses objectifs dans un deuxième temps.

En ce qui nous concerne, nous avons choisi de réaliser un modèle sur l'exemple de Zimmermann [24], car nous avions remarqué que son approche était au centre des autres travaux. Pendant la construction du modèle nous avons remarqué qu'en choisissant des fonctions d'appartenance judicieusement, on peut retomber sur les approches proposées par Verdegay [20], Werners [21] ou encore Chana [4]. Ceci nous a aidés à déterminer les objectifs dans le programme linéaire.

3.1.2 Fuzzification de ≤

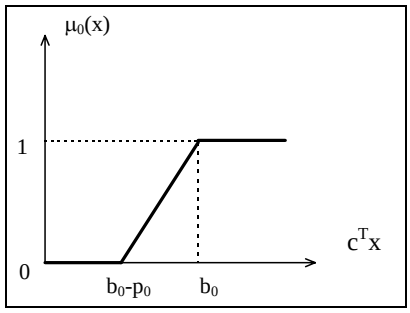
Voici comment nous avons fuzzifié notre modèle:

On peut assouplir les contraintes de la programmation linéaire, c'est-à-dire fuzzifier la contrainte d'inégalité. Par exemple si le décideur considère que $(Ax)_i \leq b_i$ est complètement satisfait alors la fonction d'appartenance de $(Ax)_i \leq b$ vaudra 1. Ensuite s'il considère que la contrainte n'est pas satisfaite, alors il donnera la valeur 0 à la fonction d'appartenance. Entre ces deux valeurs on peut prendre les valeurs intermédiaires. On appellera alors ce type de contrainte des *contraintes flexibles* .



Dans la littérature il arrive parfois qu'on dise qu'on est en train de fuzzifier les coefficients b_i, et qu'on ne touche pas la relation d'ordre "≤". En réalité les deux manières de s'exprimer sont équivalentes, ce qui malheureusement n'est jamais dit. En effet, dire que $(Ax)_i \leq b_i$ vaut 1 équivaut à choisir une valeur b_i telle que si $((Ax)_i \leq b_i)$ on puisse dire qu'on est complètement satisfait. Et dire que $(Ax)_i \leq b_i$ vaut 0 équivaut à choisir une valeur $b_i' = b_i + p_i$ telle que si $((Ax)_i \geq b_i')$ on puisse dire qu'on n'est pas du tout satisfait. Ce qui permet de décrire ainsi un nombre flou *b*.

3.1.3 Fuzzification du max.



On peut aussi fuzzifier le 'max' qui apparaît dans l'objectif de la programmation linéaire. Pour ce faire, le décideur doit choisir à quel moment il est complètement satisfait (alors le degré d'appartenance sera égal à 1) et quand il ne l'est plus du tout (alors le degré d'appartenance sera égal à 0). On dit ici qu'on a fuzzifié le but.

Comme dans le cas de la fuzzification de l'inégalité, il arrive parfois qu'on dise qu'on fait la fuzzification de z sans toucher au 'max'. Avec un raisonnement analogue à celui du paragraphe précédent, on peut montrer que ces deux approches sont équivalentes. On a donc à chaque fois deux approches semblables (fuzzifier "≤" ou b et fuzzifier max ou z). Ce qui est important ici c'est que dans les deux cas on interprète la fonction d'appartenance comme une représentation de la *satisfiabilité* de la contrainte du point de vue du décideur et donc de sa préférence.

3.1.4 Approche Symétrique de Zimmermann

Zimmermann considère que le problème de l'optimisation sous contraintes peut être traduit[5] par:

*" J'ai un objectif **ET** une contrainte C1 **ET** une contrainte C2 **ET** ... "*

Comme on a une représentation floue de l'objectif et des contraintes, il suffit alors de définir l'opérateur ET. Classiquement (Zadeh) on prend l'opérateur minimum. On remarquera au passage que dans le cas classique il se comporte exactement comme l'opérateur 'et' logique. On obtient ainsi l'ensemble flou de décisions possibles. Il est clair qu'on cherche une solution optimale, pour la trouver Zimmermann propose d'effectuer une maximisation sur cet ensemble. Ceci s'écrit mathématiquement de la manière suivante:

$$\mathbf{max\ \mu_D = max\ [\ min\ (\mu_{Obj}, \mu_{C1}, \mu_{C2}, ...)\ ].}$$

μ_D étant la fonction d'appartenance de l'ensemble de décisions.
μ_{Obj} étant la fonction d'appartenance de l'objectif
μ_{Ci} étant la fonction d'appartenance de la contrainte i.

En appliquant aux équations paramétriques des fonctions d'appartenance présentées dans le paragraphe précédent les opérateurs min et max, on obtient que: $\max\ \mu_D = \max\ [\ \min\ (\mu_{Obj}, \mu_{C1}, \mu_{C2}, ...)\ ]$ s'écrit après quelques arrangements mathématiques, en particulier en posant $\alpha = \mu_D$:

max α
$cx \geq b_0 - (1 - \alpha)* p_0$
$(Ax)_i \leq b_i + (1 - \alpha)* p_i$
$x \geq 0$ et $0 \leq \alpha \leq 1$

On obtient ainsi un problème de programmation linéaire classique. On peut donc utiliser n'importe quelle méthode (simplexe, graphique) pour trouver l'optimum.

[5] Il s'agit exactement de l'approche de Bellman et Zadeh [12, 1970]: "Decision = confluence of Goals and Constraints"

3.1.5 Critiques

D'une part, en regardant ces équations on voit qu'on cherche en réalité à augmenter la valeur du réel α, c'est-à-dire la satisfiabilité du décideur. Si $\alpha = 1$ alors pour toutes les contraintes le décideur est satisfait à 100%. On remarquera qu'un des défauts majeurs de cette approche est qu'on a le même α pour toutes les contraintes et pour le but.

D'autre part on utilise l'opérateur max-min. L'utilisation d'autres opérateurs n'est pas exclue, et le choix de Zimmermann n'est pas ici justifié. Chen et Hwang [5] donnent un tableau qui résume les différents opérateurs existants (voir annexe). Des travaux ultérieurs peuvent montrer les avantages ou désavantages dans l'utilisation d'un opérateur plutôt qu'un autre.

Finalement j'aimerais insister sur le fait que FLP est l'abréviation de Fuzzy Linear Programming. Mais on remarque que dans cette approche on s'intéresse simplement aux contraintes sans prendre en compte l'existence ou non des incertitudes sur les données. C'est pour cette raison que je préfère le terme "flexible" qui fait référence à la souplesse des contraintes au terme "flou".

3.1.6 Application

La réalisation du modèle nous a permis d'une part de nous rendre compte lors de sa construction qu'on peut se servir de résultats intermédiaires (comme on l'a dit précédemment) pour fixer les objectifs. On a remarqué que le défaut majeur de cette méthode: un α unique pour toutes les contraintes, oblige à avoir toutes les contraintes "tendues". Sinon le degré de satisfaction α après optimisation est petit. Paradoxalement, on peut se servir de cette caractéristique: en travaillant de manière interactive, on peut se rendre compte des contraintes limitantes trop fortes et de leurs sensibilités.

Un autre défaut majeur de cette approche est qu'elle ne prévoit pas l'introduction directe de l'incertain dans les coefficients. On a donc poursuivi nos recherches en nous intéressant à la programmation linéaire possibiliste.

3.2 Possibilistic Linear Programming (PLP)

La théorie des possibilités formulée par Zadeh en 1977, prolonge la théorie des ensembles flous, développée par lui-même dès les années 60. Ce dernier concept veut rendre compte du fait qu'un objet peut plus ou moins correspondre à une certaine catégorie dans laquelle on voudrait le ranger. Les degrés de possibilité peuvent prendre toutes les valeurs entre 0 et 1. La théorie des possibilités offre un modèle de quantification du jugement qui permet aussi une généralisation du calcul d'erreurs. On remarque que, quand les degrés de possibilité ne prennent que les valeurs 1 ou 0, le calcul des possibilités coïncide exactement avec le calcul d'erreurs, dans lequel l'information imprécise est représentée en termes de valeurs possibles (au lieu de valeurs précises).

3.2.1 Etat de l'art

Il existe de nombreux travaux dans le cadre de la théorie possibiliste, mais malheureusement les applications réelles sont rares. En particulier aucune application n'existe dans le domaine de l'agro-économique. Dans le paragraphe qui suit, je présente les principales approches, en expliquant en particulier où les efforts de fuzzification ont été réalisés. Pour cela on utilisera les notations habituelles du problème formel (voir définitions et notations).

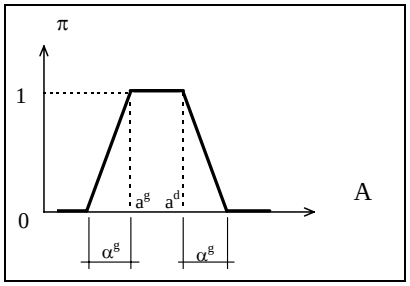
Commençons avec les travaux de Ramik et Rimanek [15] qui insistent seulement sur la fuzzification de A et b. Les coefficients qu'ils proposent sont des trapèzes et ils donnent une méthode rigide de comparaison, qui permet de traduire les contraintes. L'optimisation est celle réalisée classiquement. Tanaka [19] propose une meilleure approche en donnant au décideur le choix d'un paramètre α a priori, qui donne un certain contrôle sur la comparaison. Malheureusement, son approche est spécifique aux nombres triangulaires flous et son paramètre n'a pas de traduction intuitive. Dubois [7] présente une approche très générale, mais malheureusement il laisse de côte l'optimisation, en l'effectuant à la manière classique. Rommelfanger, Hanuschek et Wolf [16] proposent une approche où les coefficients de l'objectif sont des nombres flous. Après deux optimisations indépendantes, ils ramènent le problème à une approche du type symétrique de Zimmermann [24]. De même Lai et Hwang [13] proposent une approche, qui traduit l'objectif: diminution du risque et augmentation du profit, dans une approche de type Zimmermann [24]. Malheureusement elle est spécifique aux nombres flous triangulaires.

Pour le modèle qu'on envisageait de faire, on avait besoin d'une approche qui permettait d'avoir A, b et c floues, car notre modèle calculait les coefficients intervenant dans la fonction objectif à partir des coefficients technologiques flous intervenant dans les contraintes.

3.2.2 Choix du nombre flou: trapézoïdal.

Pour la construction de notre modèle on a réalisé des nombreux choix, voici les explications:

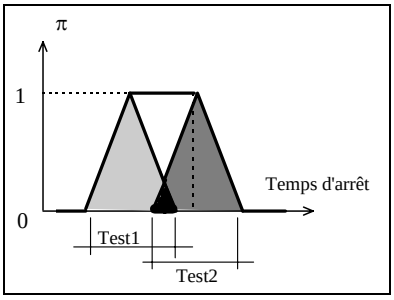
Un nombre flou trapézoïdal peut être noté comme suit: $Tp=(a^g,a^d,\alpha^g,\alpha^d)$. La fonction d'appartenance vaut 1 pour les valeurs comprises entre a et b. Ceci veut dire qu'il est tout à fait possible que la variable prenne n'importe quelle valeur entre a^g et a^d. α^g et α^d sont les "erreurs" qu'on fait sur a^g et a^d respectivement.



On a choisi parmi les divers types de nombres flous les trapézoïdaux, car ces nombres sont assez simples (linéaires par morceaux), ce qui les rend particulièrement utiles en résolution de problèmes dans la programmation linéaire possibiliste. Et en même temps cette catégorie de nombres contient les nombres triangulaires (il suffit que $a^g=a^d$) et les nombres réels classiques x=(x,x,0,0). On remarquera aussi par la suite qu'on ne donne aucune interprétation particulière à la forme des deux pentes. Ce qui est une raison de plus de rester dans ce cas simple.

Construction des ces Nombres

Une fois le modèle classique construit, la première difficulté qu'on a trouvée (en tant que décideurs) était la construction des nombres flous. Les différents travaux existants éludent le problème en disant que c'est au décideur de les déterminer. Il semblerait que cette étape de la modélisation soit évidente, or au moment d'une application réelle on s'aperçoit que ce processus est délicat. Prenons un exemple concret: le temps de fermentation dans notre modèle. La pratique expérimentale consiste à mesurer à des intervalles de temps irréguliers la concentration de glucose dans le vin; lorsque la mesure de celle-ci est inférieur à une valeur fixe (toujours la même), on décide de s'arrêter. Nous avons cherché à préciser un temps d'arrêt. Pour homogénéiser ces données nous avons décidé de réaliser tout simplement une interpolation linéaire, qui a ramené toutes les variables aux valeurs qu'elles avaient quand la concentration était égal à la valeur fixe. Les points pris en compte, pour l'interpolation ont été le dernier temps de mesure avant seuil et le premier après seuil. Nous avons réalisé une interpolation linéaire car les équations différentielles partielles qui traduisent la cinétique sont trop lourdes à résoudre et l'approximation linéaire en fin de fermentation semble tout à fait correcte.



Les expériences de fermentation se réalisent une ou deux fois pour chaque type de blé. On garde alors toute l'information en construisant deux triangles d'erreurs, avec comme valeur centrale le résultat de l'interpolation et comme étalement l'erreur[6] de mesure donnée par le laboratoire. On remarque que, à cause de la variabilité biologique, il se peut que l'intersection d'intervalles d'erreurs soit vide. Ensuite on a pris l'approche pessimiste qui consiste à dire que, si on a trouvé ces deux valeurs, il est tout à fait possible de trouver les valeurs intermédiaires et on construit ainsi notre trapèze. De cette manière on définit une plage de variation tout à fait possible nuancée sur les bords.

3.2.3 Arithmétique des nombres flous.

Dans ce paragraphe, on présente des méthodes de calcul d'expressions comportant des quantités imprécises , représentées par des distributions de possibilité. Ces méthodes sont en total accord avec ce qu'on appelle couramment le calcul d'erreurs dont elles sont une extension au cas d'intervalles pondérés. (voir Dubois et Prade [8] et Chen and Hwang [5]).

Soient $A=(a^g,a^d,\alpha^g,\alpha^d)$ et $B=(b^g,b^d,\beta^g,\beta^d)$ deux nombres flous trapézoïdaux. Soit F(A,B) une fonction a deux variables floues A et B.

Si F(x,y) est une fonction croissante par rapport à x et à y, on peut établir la formule approchée[7] dans certains cas**:**

$$F(A,B) = (F(a^g,b^g),\ F(a^d,b^d) , F(a^g,b^g) - F(a^g-\alpha^g,b^g-\beta^g) , F(a^d+\alpha^d,b^d+\beta^d) - F(a^d,b^d))$$

Si F(x,y) est une fonction croissante par rapport à x et décroissante par rapport à y, on peut établir la formule approchée dans certaines cas**:**

$$F(A,B) = (F(a^g,b^d),\ F(a^d,b^g),\ F(a^g,b^d) - F(a^g-\alpha^g,b^d+\beta^d),\ F(a^d+\alpha^d, b^g-\beta^g) - F(a^d,b^g))$$

---

[6] Les laboratoire calculent souvent leurs erreurs grâce à une analyse de variance. Il faut faire la remarque que l'erreur qu'on doit utiliser ici, ne doit pas provenir de cette l'analyse, car d'une part cette méthode prends en compte l'écart entre les résultats, ce qui se traduit dans notre cas par la justification de l'écart par l'erreur. Ceci ne se produirait pas si les données étaient statistiques ou en tout cas plus que deux.

[7] On remarquera que quand $\alpha^g,\alpha^d,\beta^g$ et β^d deviennent très petits on retombe sur le classique calcul d'erreurs (utilisé par les physiciens):

$$F(a^g,b^g) - F(a^g-\alpha^g,b^g-\beta^g) \cong F'_A(a^g,b^g)*\alpha^g + F'_B(a^g,b^g)*\beta^g = \text{ERREUR}$$

On trouve donc pour les quatre opérations arithmétiques classiques les formules suivantes dans le cas où **A,B ≥ 0** :

Opération		*Résultat*
Somme	A + B	$(a^g+b^g , a^d+b^d , \alpha^g+\beta^g , \alpha^d+\beta^d)$
Soustraction	A + B	$(a^g-b^d,\ a^d-b^g, \alpha^g+\beta^d, \alpha^d+\beta^g)$
Multiplication [8]	A * B	$(a^g.b^g , a^d.b^d , (a^g.\beta^g + b^g.\alpha^g - \alpha^g.\beta^g) , (a^d.\beta^d + b^d.\alpha^d - \alpha^d.\beta^d))$
Division [8]	A / B	$(a^g/b^d , a^d/b^g , (a^g.\beta^d+b^d.\alpha^g) / [b^d.(b^d+\beta^d)] , (a^d.\beta^g+b^g.\alpha^d) / [b^g.(b^g-\beta^g)])$

Remarque: ces opérations sont valables si et seulement si on considère que les variables A et B sont non - interactives.

On est arrivé à ces formules à partir du calcul d'erreurs classique de manière semblable à celle donné par Dubois et Prade dans [8] . On arrive aux mêmes formules si on prend comme t-norme le min et comme t-conorme le max pour le principe d'extension et ensuite on réalise deux α-coupes l'une au niveau α=1 et l'autre au niveau α=0. Ce qui précède nous permet alors de conclure, d'une part qu'on est "pessimiste" à cause du choix min-max et d'autre part que les formules précédentes sont exactes pour le noyau et pour le support du nombre flou. Comme on ne donne pas d'interprétation particulière à la forme de la pente on peut faire *l'approximation* et prendre une droite. On remarque que cette approximation par une droite est d'autant plus juste que α^g et α^d sont petits devant a^g et a^d respectivement.

Dans notre modèle on a rencontré un problème de variables interactives: On veut calculer un débit: éthanol/temps. Il semble évident que si le temps de fermentation augmente, alors l'éthanol dans le vin augmente aussi. Pour résoudre ce type de difficulté, on doit se ramener à une modélisation où les variables sont non-interactives. Si on applique les formules précises précédentes, on va diviser le résultat en éthanol d'une expérience par le temps de fermentation de l'autre, ce qui paraît absurde. Il faut donc traiter les deux expériences séparément (dissocier le trapèze en deux triangles dans notre cas), et alors les variables ne seront plus interactives.

3.2.4 Comparaison des nombres flous.

La comparaison de nombres flous est au cœur de la problématique de la programmation mathématique possibiliste. En effet la contrainte $Ax \leq b$, A et b étant des nombres flous, est difficile à traduire. Chen et Hwang [5] donnent un tableau qui résume différentes méthodes existantes (voir annexe).

En ce qui concerne la programmation linéaire floue, tout d'abord les contraintes ont été traduites en inégalités par des calculs fixes. Par exemple Ramik et Rimanek [15] présentent dans leur approche, une méthode qui traduit l'inégalité apparaissant dans la contrainte. Mais cette inégalité se traduit par une formule fixée et on ne peut pas la changer. Tanaka et al. [19] proposent pour des nombres flous triangulaires une

[8] solution approchée

comparaison qui s'appuie sur le principe d'extension, où le décideur règle un paramètre. Ceci nous donne déjà plus de souplesse dans l'inégalité (contrainte). Cette souplesse semble importante; ainsi Rommelfanger [17] présente une méthode qui, grâce à la fuzzification de la somme par le principe d'extension (utilisant la norme de Yager) permet des variations sur les contraintes, tout en laissant fixe la comparaison.

Les méthodes de comparaison sont nombreuses, mais il faut savoir que ce qu'on cherche n'est pas de comparer deux nombres flous (i.e. si tel nombre est plus petit que tel autre), mais plutôt de traduire des contraintes. On remarque que les contraintes "≤" peuvent être traduites de différentes manières. Ainsi Dubois [7] propose deux types de contraintes: en s'appuyant sur les notions de nécessité et possibilité, il déduit des contraintes faibles et fortes. En plus, pour chacune, l'utilisateur doit régler un paramètre. Slowinski [18] propose de même une méthode à deux paramètres, un dit de contrôle optimiste et l'autre pessimiste. Ces approches sont satisfaisantes, mais malheureusement elles ont des comportements qui ne correspondent pas à nos attentes. En particulier, dans ces approches, il existe la possibilité que les noyaux se chevauchent. Ceci se traduirait dans notre cas, par la possibilité que l'inégalité ne soit pas vérifiée pour un couple des valeurs qu'on a considérées comme tout a fait possibles. Pour cette raison je présente ici une approche personnelle:

Il s'agit d'une méthode simple où le décideur choisit le niveau de risque qu'il veut prendre sur la partie incertaine. La partie incertaine est par définition même la partie où il n'est pas tout à fait possible qu'on trouve la valeur mesurée. Ceci se traduit dans le cas des trapèzes par la région là où la fonction d'appartenance ne prend pas la valeur 1, c'est-à-dire les deux pentes.

Lors de la modélisation d'une contrainte, on aimerait pouvoir imposer que toutes les valeurs de possibilité supérieures à une certaine valeur γ vérifient l'inégalité. Ce qu'on veut, c'est que pour des possibilités élevées, la contrainte soit vérifiée.

Pour réaliser ceci, il suffit dans notre cas d'empêcher que l'intersection des deux pentes dépasse en hauteur (possibilité) la valeur γ, en prenant la précaution que les noyaux vérifient aussi l'inégalité. Cette deuxième condition empêche qu'on se trouve dans des cas pathologiques. On remarquera que les calculs qui donnent l'intersection des deux pentes dans le cas général ne sont pas simples. Ce que je propose de faire alors, c'est d'observer simplement l'inégalité à la hauteur. Comme les nombres utilisés sont des fonctions monotones avant le noyau d'une part et après le noyau d'autre part, alors la condition cherchée sera satisfaite.

La valeur γ, qu'on règle, est en quelque sorte un paramètre risque. Il nous dit jusqu'à quelle possibilité l'utilisateur est prêt à accepter que la contrainte ne soit pas satisfaite. En particulier quand $\gamma = 0$, on dira qu'alors on ne prend pas de risque et quand $\gamma = 1$ alors le risque est total (*pour la partie incertaine seulement*).

On rappelle que ce qu'on cherche à faire c'est à satisfaire aux mieux une contrainte. En ce qui concerne la région tout à fait possible, deux cas peuvent se produire. Ou bien on veut que les valeurs du premier noyau soient tous plus petits que ceux du second. Ou bien on veut que la plus grande valeur du premier soit plus petite que la plus grande du second. Les autres cas restants (comparer le plus petit, avec le plus grand, le plus grand avec le plus petit et enfin les plus petits entre eux) n'ont pas de sens, si on se rappelle encore une fois notre but: *ajuster* au mieux les contraintes. On parlera dans le premier cas de contrainte disjointe et dans l'autre de contrainte emboîtée. Ce qu'il est agréable de retrouver ici, c'est que le cas disjoint traduit

intuitivement l'inégalité habituelle et le cas emboîté l'inclusion en tant que relation d'ordre.

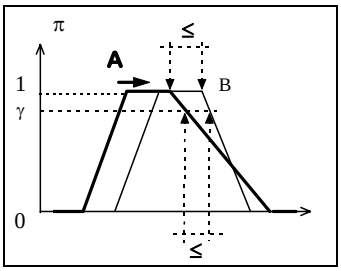
Soient $(a^g,a^d,\alpha^g,\alpha^d)$ et $(b^g,b^d,\beta^g,\beta^d)$ les deux nombres flous à comparer. La première chose à déterminer c'est le type d'inégalité: disjointe ou emboîtée. Ensuite il faut choisir le niveau de risque γ (entre 0 et 1).

La contrainte "$A \leq B$" se traduit alors par les contraintes suivantes:

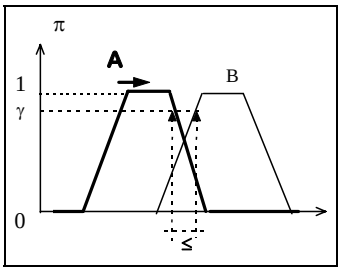
le cas disjoint :	$a^d + \alpha^d.(1-\gamma) \leq b^g + \beta^g.(\gamma-1)$	
le cas emboîté:	$a^d + \alpha^d.(1-\gamma) \leq b^d + \beta^d.(1-\gamma)$ et $a^d \leq b^d$	quand A varie et B fixe
	$a^g + \alpha^g.(\gamma-1) \leq b^g + \beta^g.(\gamma-1)$ et $a^g \leq b^g$	quand B varie et A fixe

Ce qui se traduit graphiquement par:

Emboîté (A variable)



Disjoint



Ce type de contraintes empêche d'une part que l'intersection entre deux zones de risque dépasse le niveau de possibilité γ et d'autre part que les noyaux se superposent, ce qui était notre souhait.

Remarque: Il semblerait qu'ici la forme de la pente joue un rôle. En effet, l'intersection n'est pas en général au même endroit pour des droites que pour d'autres courbes. Mais cette erreur est dans notre cas négligeable, car α^d et α^g sont négligeables respectivement devant a^d et a^g. Et même si ce n'était pas le cas, on sait que le risque estimé par l'utilisateur n'est pas une fonction linéaire, ce qui implique que le paramètre risque est un paramètre intuitif. Pour finir sur cette remarque j'insiste sur le fait qu'on n'interprète à aucun moment la forme de la courbe.

3.2.5 Programmation linéaire avec des nombres flous.

Maintenant qu'on connaît les opérateurs arithmétiques de base, traduisons le programme linéaire classique: Soient $A.x \leq b$ l'ensemble de contraintes. Alors la $i^{éme}$ ligne s'écrit: $\Sigma_j (A_{i,j} * x_j) \leq b_i$. Si $A_{i,j}$ et b_i sont des nombres flous trapézoïdaux, cette contrainte s'écrit grâce aux opérateurs définis précédemment:

$$\mathbf{(\Sigma_j (a^g_{i,j} * x_j), \Sigma_j (a^d_{i,j} * x_j), \Sigma_j (\alpha^g_{i,j} * x_j), \Sigma_j (\alpha^d_{i,j} * x_j)) \leq (b^g_i , b^d_i , \beta^g_i , \beta^d_i)}$$

Les méthodes de comparaison donnent pour un risque $\gamma_i \in [0,1]$:

cas disjoint:	$\{ \beta^g_i, + \Sigma_j (\alpha^d_{i,j} * x_j) \}. (1-\gamma_i)$	$\leq$	$b^g_i, - \Sigma_j (a^d_{i,j} * x_j)$
cas emboîté [9]:	$\{ \Sigma_j (\alpha^d_{i,j} * x_j) - \beta^d_i \} . (1-\gamma_i)$	$\leq$	$b^d_i - \Sigma_j (a^d_{i,j} * x_j)$
	et $\Sigma_j (a^d_{i,j} * x_j),$	$\leq$	b^d_i

On a donc réécrit les contraintes, il reste à traduire l'optimisation.

3.2.6 Optimisation

Le calcul des quantités floues propose essentiellement un raffinement de l'analyse de sensibilité qui peut ainsi devenir nuancée, et sans augmentation spectaculaire de la masse de calculs nécessaires. Ce qui est important ici c'est de comprendre ce qu'on entend par "optimiser". On peut d'une part considérer l'optimum comme un compromis (le meilleur) entre la prise de risque et la diminution du profit. Tous les travaux existants cherchent à trouver un (et un seul) vecteur x tel que la solution soit maximale dans un sens multicritère. Pour les besoins du modèle agro-économique qui sont plutôt du type analyse, on peut considérer que l'optimisation avec des nombres flous doit simplement répercuter les incertitudes des données sur l'optimum. On cherche donc une extension de l'optimisation classique en prenant en compte des incertitudes. Pour réaliser cette extension on s'appuie ici sur l'approximation présentée dans le paragraphe sur l'arithmétique des nombres flous. On obtient comme optimum un nombre flou:

$$\mathbf{[(c^g)^T.x] , max [(c^d)^T.x], \{max [(c^g)^T x] - max [(c^g-\delta^g)^T.x]\}, \{max [(c^d+\delta^d)^T.x] - max [(c^d)^T.x]\})}$$

(c^g, c^d, δ^g, δ^d) étant quatre vecteurs où chaque ligne correspond à un des coefficient d'un nombre flou trapézoïdal, qui représente un coefficient économique de la fonction objectif.

On sait qu'alors le profit peut se trouver n'importe où, là où la fonction d'appartenance de l'optimum vaut 1. Cette fonction nous donne en plus l'incertitude sur les bornes de cette plage. On distinguera quatre optimisations:

(Opt. Pessimiste Certaine, Opt. Optimiste Certaine, Opt. Pess. Incertaine, Opt. Opt. Incertaine)

Par exemple, quand on dit qu'on fait une optimisation pessimiste certaine, on fait une optimisation sur les valeurs pessimistes tout à fait possibles des coefficients de la fonction objectif (possiblité = 1), tout en respectant les contraintes de la manière habituelle.

---

[9] Ici seulement Ax varie et b est fixe.

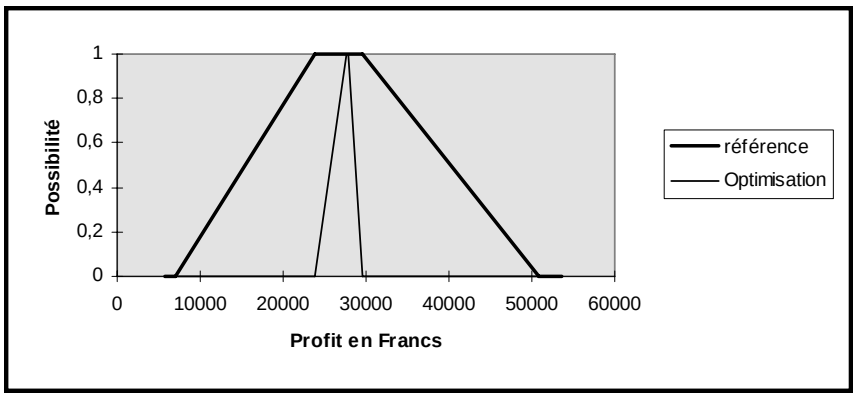
On remarque que pour chacune des optimisations précédentes, il se peut que les vecteurs x trouvés soient différents. Ce qu'on cherche c'est la variabilité de l'optimum et non un optimum compromis unique. Des travaux ultérieurs peuvent viser à définir une optimisation multiobjectif, qui donnerait un vecteur x unique.

3.2.7 Résultats

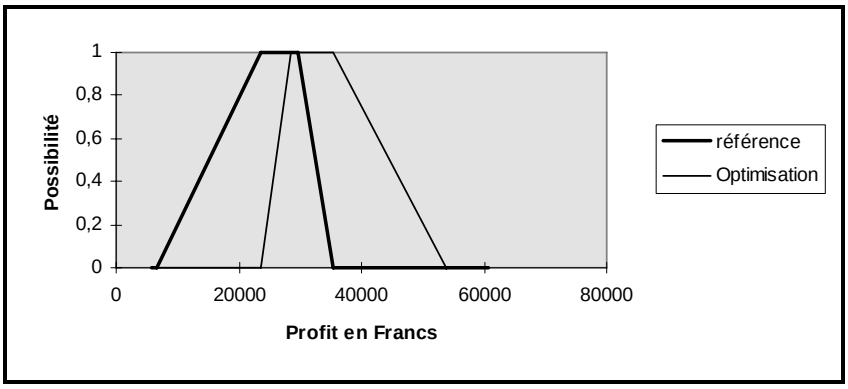
Comme on vient de le voir, l'optimisation qu'on proposée donne un nombre unique flou. On est allé plus loin dans l'interprétation des résultats, que la simple analyse de ce nombre flou. Pour chacun de ces quatre sous-optimisations, un type de blé était effectivement choisi. On avait de cette manière automatiquement comme image de la fonction objectif, le nombre flou qui représente le profit. Sur un graphique, on pouvait alors comparer le profit optimum avec celui d'un blé moyen. On a obtenu trois résultats différents. Lors de l'optimisation pessimiste incertaine, on a obtenu un profit complètement inclus dans le profit réalisé par un blé moyen; dans ce cas-là, on peut conclure que ce blé n'apporte certainement aucune amélioration. Lors de l'optimisation pessimiste certaine, on obtient un profit qui est, sur le noyau, essentiellement supérieur au profit du blé moyen. Cet blé est un bon choix. On sait qu'alors le profit peut se trouver sur la plage qui délimite le noyau. Les optimisations optimistes certaine et incertaine choisissent le même blé, dont le profit est effectivement le meilleur dans les cas les plus favorables, mais il peut aussi de manière tout à fait possible, être plus bas que celui donné par le blé moyen. Cette très grande variation est peut-être due à des erreurs expérimentales lors de mesures. Il faudrait pour confirmer ces résultats répéter les expériences pour ce type de blé.

On a trouvé donc deux solutions optimales, sur lesquelles on connaît les conséquences des imprécisions des données initiales. Le risque entre en jeu et c'est donc le décideur qui fait le choix final, cette modélisation ne l'aidant qu'à mieux choisir.

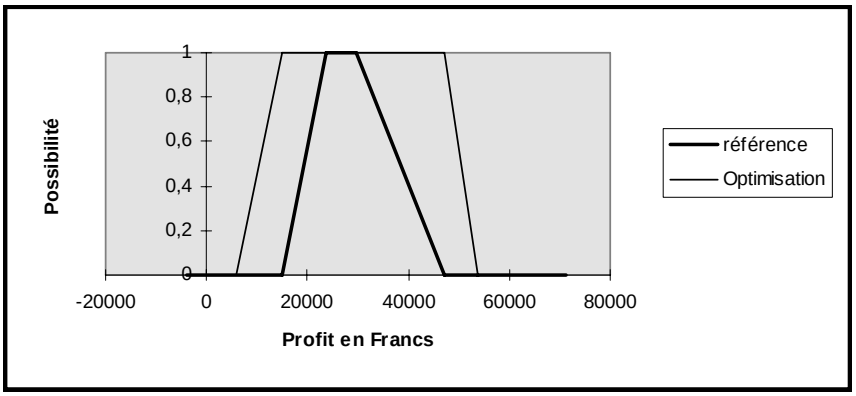
Optimisation Pessimiste Incertaine



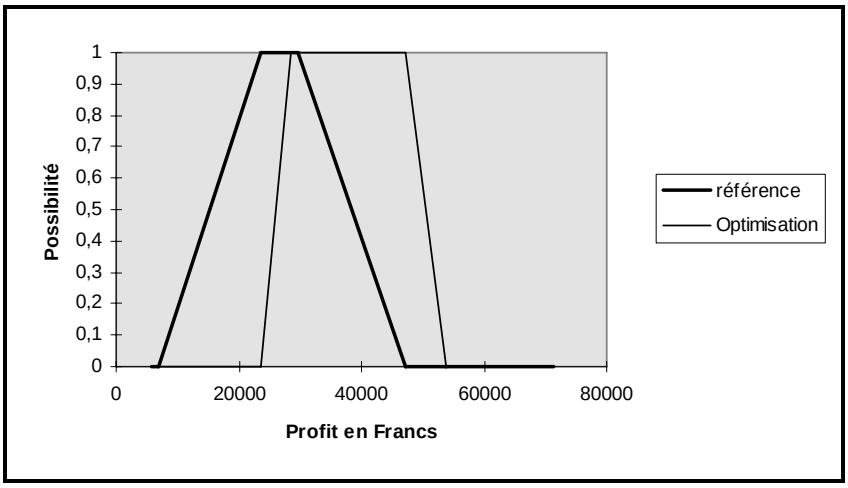
Optimisation Pessimiste Certaine



Optimisation Optimiste Certaine et Incertaine



Optimisation Générale - Analyse



4. Le Modèle: Caractéristiques techniques

Le modèle auquel il est fait référence dans ce texte est de petite taille. Il a 34 variables (34 types de blé) et quatre contraintes. Mais la complexité n'est pas moindre, car pour chaque type de blé on calcule dans un premiers temps 22 coefficients techniques et économiques à partir d'une trentaine des caractéristiques, sur lesquelles l'optimisation est faite. On remarquera d'autre part que nos quatre contraintes sont toutes différentes. La taille de ce modèle est la conséquence de la difficulté à accéder à l'information. On a rajouté dans le modèle une rétroaction visuelle pour l'utilisateur. Toutes les contraintes et les optimisations sont automatiquement représentes par des graphes, ainsi le décideur a une représentation concrète et globale de ce qui se passe.

Le modèle a été construit sur le tableur d'Excel de Microsoft. Malgré les très bonnes performances lors de l'optimisation du modèle classique, sa capacité s'est vue très vite limitée lors des premiers essais du modèle possibiliste. Il ne faut pas oublier que le nombre de valeurs est alors multiplié par 4, avec tous les calculs qui en résultent. Le temps de calcul n'a jamais dépassé les dix minutes.

Pour plus d'explications sur le modèle classique et ses résultats, nous présentons trois articles en annexe.

5. Conclusions

Ce projet est une expérience de collaboration pluridisciplinaire à l'INRA. Les changements apportés à la programmation linéaire classique, sont le fruit de la confrontation et de l'intégration du savoir et des différents points de vue provenant des économistes, des mathématiciens, des industriels, des chimistes et des informaticiens.

Si on réalise maintenant une évaluation critique du contenu du projet lui même. On s'aperçoit que l'approche de Zimmermann est limitée. En effet, on a vu ses manques et ses défauts, En particulier le α unique. Mais on a trouvé paradoxalement dans cet apparent défaut une autre application et un intérêt: tendre les contraintes. Un autre intérêt non négligeable de cette modélisation est qu'elle est très répandue et simple, ce qui a impliqué la naissance de toute une famille de travaux qui suivent cette approche de très près, et qu'il ne faut surtout pas confondre avec les approches possibilistes, même s'il existe des travaux possibilistes qui ramènent leur problème à un modèle de type Zimmermann lors de l'optimisation.

Dans l'approche possibiliste, nous avons introduit une nouvelle méthode pour traduire les contraintes. Elle a l'avantage d'être réglable par le décideur. On remarque que les paramètres ont une signification réelle, et il me semble que ceci est indispensable dans toute prise de décision. Malheureusement et heureusement en même temps, on ne peut pas tout régler. Il se peut que, dans un autre modèle, ces traductions n'aient plus de sens, par exemple à cause d'une autre interprétation des nombres flous.
Dans cette modélisation, on n'a pas vraiment traité le problème de l'optimisation, car ce qu'on cherche c'est à propager l'imprécision des données jusqu'à la solution optimale. On peut obtenir dans cette approche, jusqu'à quatre vecteurs x optimaux différents. Lequel choisir ? La solution est-elle stable (voire robuste) pour ce choix ? Ce sont des questions à développer ultérieurement.

6. Bibliographie

[1] Bellman R.E. et L.A. Zadeh , Decision-making fuzzy environmental, *Management Science* 17 (1970) B141-B164.

[2] Bezdek, J.C. *Analysis of Fuzzy Information (Vol III): Applications in Engineering and Sciences* (CRC Press, Boca Raton, 1987)

[3] Buckley J.J., Stochastic versus possibilistic programming, *Fuzzy Sets and Systems* 34 (1990) 173-177.

[4] Chana S., The use of parametric programming in FLP, *Fuzzy Sets and Systems* 11 (1983) 243-251.

[5] Chen S.J. et C.L. Hwang, *Fuzzy Multiple Attribute Decision Making: Methods and Applications* (Springer-Verlag, Heidelberg, 1991).

[6] Delgado M., S. Kacprzyk, J.-L. Verdegay, M.A. Vila (eds.), *Fuzzy Optimisation : Recent Advances* (Physica-Verlag, Heidelberg, 1994).

[7] Dubois D. Linear programming with fuzzy data, in [2] (1987) 241-263.

[8] Dubois D. et H. Prade, *Possibility Theory: An Approach to Computerised Processing of Uncertainy* (Plenum, New York, 1987).

[9] Gupta, M.M. and E. Sanchez (eds.), *Approximate Reasoning in Decision Analysis* (North-Holland, Amsterdam, 1982).

[10] Jones A., A. Kaufmann and H.-J. Zimmermann (eds.), *Fuzzy Sets Theory and Applications* (D. Reidel, Dordrecht, 1985).

[11] Kacprzyk, J. and R. Yager (eds.), *Management Decision Support Systems Using Fuzzy Sets and Possibility Theory*. (Verlag TUV Rheinland, New York, 1985).

[12] Lai Y.J. and Hwang C.L., *Fuzzy Mathematical Programming: Methods and Applications* (Springer-Verlag, Heidelberg, 1992).

[13] Lai, Y.J et C.L. Hwang, A new approach to some possibilistic linear programming problem, *Fuzzy Sets and Systems* 49 (1992).

[14] Luhandjula M.K., Fuzzy optimisation and appraisal, *Fuzzy Sets and Systems* 30 (1989) 257-282.

[15] Ramik J. et J. Rimanek, Inequality relation between fuzzy numbers and its use in fuzzy optimisation, *Fuzzy Sets and Systems* 16 (1985) 123-138.

[16] Rommelfanger H. , R. Hanuschek et J. Wolf , Linear programming with fuzzy objectives, *Fuzzy Sets and Systems* 29 (1989) 31-48.

[17] Rommelfanger H., Somme problems of fuzzy optimisation with t-norm based extended addition, in [6] 1994.

[18] Slowinski R., A Multiobjectif fuzzy linear programming method for water supply system development planning, *Fuzzy Sets and Systems* 25 (1986) 217-237.

[19] Tanaka H., H. Ichihashi and K. Asai, A formulation of fuzzy linear programming problems with trapezoid fuzzy parameters, in [11] (1985) 146-154.

[20] Verdegay J.L., Fuzzy mathematical programming, in [9] (1982) 231-236.

[21] Werners B., Interactive multiple objective programming subject to flexible constraints, *European Journal of Operational Research* 31 (1987) 342-349.

[22] Wierzchon S.T., Linear Programming with fuzzy sets: a general approach, *Mathematical Modelling* 9 (1987) 447-459

[23] Yazenin, A.V., Fuzzy and Stochastic programming, *Fuzzy Sets and Systems* 22 (1987) 171-180.

[24] Zimmermann H.-J., Fuzzy programming and linear programming with several objective functions, *Fuzzy Sets and Systems* 1 (1978) 45-55.

[25] Zimmermann, H. J., Application of fuzzy set theory to mathematical programming, *Information sciences* 36 (1985) 29-58.

[26] Zimmermann H. J., Fuzzy set theory and mathematical programming, in [10] (1985) 99-114.

[27] Zimmermann H. J., *Fuzzy Set Theory and Its Application* (Kluwer-Nijhoff, Hinghum, 1985)

7. Annexes

Les annexes sont organisées en trois parties:

1. Tableaux complémentaires:
   - arithmétique floue pour des nombres trapézoïdaux,
   - méthodes de comparaison de nombres flous,
   - opérateurs min-max compensatoires.
2. Un exemple montrant comment construire un programme linéaire flexible, selon l'approche de Zimmermann, en évitant les ennuis provoqués par le α unique.
3. Trois articles, qui présentent le modèle économique classique et ses résultats:
   - Optimisation des coûts industriels de la production d'éthanol de blé
   - Optimisation économique de la transformation industrielle des blés en éthanol.
   - Optimisation économique de la production d'éthanol de blé par le choix des variétés de blés et des itinéraires techniques.