SOLVING MULTIOBJECTIVE MATHEMATICAL PROGRAMMING PROBLEMS WITH FIXED AND FUZZY COEFFICIENTS

by

Stanislas Sakera RUZIBIZA

submitted in accordance with the requirements
for the degree of

MASTER OF SCIENCE

in the subject

Operations Research

at the

UNIVERSITY OF SOUTH AFRICA

Supervisor: Prof M.K. LUHANDJULA

APRIL 2011

DEDICATION

To my wife Florence and my children Boso, Willy, Candide, Grace, Hope and Divine for their love;

To all my students at Lukanga Adventist Secondary School (LASS), Kanyatsi Adventist Secondary School (KASS), Rwankeri Adventist College (CAR), Adventist University of Central Africa (AUCA), Independent Institute of Lay Adventists of Kigali (INILAK) and Kigali Independent University (ULK).

ACKNOWLEDGMENTS

At the end of this program of MSc, I owe my thanks to Professor M. K. Luhandjula who kindly agreed to supervise this dissertation. His availability, goodwill and patience have fuelled my courage.

I am greatly thankful to Authorities of Independent Institute of Lay Adventists of Kigali who have funded my postgraduate studies at UNISA. May their commitment be plentifully rewarded.

I also feel so much indebted to Professor C. J. Swanepoel, Director of School of Economics, for his involvement in my scientific advancement.

My sincere gratitude is addressed to Professor W Fouché for his diligence while teaching me Combinatorial Optimization Analysis and Heuristics course.

My gratitude goes also to Rwanda Water and Sanitation Corporation (RWASCO) for all data that I used in the case study of this dissertation.

I extend special thanks to my wife and children for having encouraged me to undertake postgraduate studies.

I would like to thank all those who, in one way or another, have contributed to the completion of this scientific work.

Stanislas Sakera RUZIBIZA

DECLARATION

I declare that SOLVING MULTIOBJECTIVE MATHEMATICAL PROGRAMMING PROBLEMS WITH FIXED AND FUZZY COEFFICIENTS is my own work and that all the sources that I have used or quoted have been indicated and acknowledged by means of complete references.

Author	S. S. RUZIBIZA	………………………………………....
Degree:		MASTER OF SCIENCE
Subject:		Operations Research
Supervisor:	Prof M.K. LUHANDJULA:	………………………………………….
Date:		………………………………………….

TABLE OF CONTENTS

ABSTRACT

Many concrete problems, ranging from Portfolio selection to Water resource management, may be cast into a multiobjective programming framework. The simplistic way of superseding blindly conflictual goals by one objective function let no chance to the model but to churn out meaningless outcomes. Hence interest of discussing ways for tackling Multiobjective Programming Problems. More than this, in many real-life situations, uncertainty and imprecision are in the state of affairs.
In this dissertation we discuss ways for solving Multiobjective Programming Problems with fixed and fuzzy coefficients. No preference, a priori, a posteriori, interactive and metaheuristic methods are discussed for the deterministic case. As far as the fuzzy case is concerned, two approaches based respectively on possibility measures and on Embedding Theorem for fuzzy numbers are described. A case study is also carried out for the sake of illustration. We end up with some concluding remarks along with lines for further development, in this field.

Key words: Multiobjective Programming, Fuzzy set, Pareto optimal solution, possibility measures, Embedding Theorem.

INTRODUCTION

Optimization is a very old and classical area which is of high concern to many disciplines [3], [15].
Engineering as well as Management, Politics as well as Medicine, Artificial Intelligence as well as Operations Research and many other fields are in one way or another concerned with optimization of designs, decisions, structures, procedures or information processes.

In a deterministic environment using a single well-defined criterion for evaluating potential alternatives, the optimal decision can be obtained through user-friendly Mathematical Programming software.
Optimization procedure is, in this case, a batch-type process assuming a closed model in which all information is available and in which the Decision Maker could provide and process all information simultaneously.

There are several good books to which the reader may refer for a comprehensive treatment of Mathematical Programming [14], [30].
More detailed accounts of this subject including optimization software may be found in [50].
In a turbulent environment involving several conflicting objective functions and intrinsic or informational imprecision, the optimization is not that simple [35], [50].

In this dissertation we consider the challenging task of pondering conflicting goals in an optimization setting. We also address the issue of incorporating fuzziness in a Multiobjective Mathematical Programming framework.
From this discussion we have obtained a mini-Decision Support System for helping people facing problems that may be cast into a deterministic or fuzzy multiobjective programming framework.

The dissertation is organized as follows.
For the presentation to be somewhat self-contained, we provide basic notions of Mathematical Programming and Fuzzy set Theory in Chapter 1.
Chapter 2 is devoted to analysis of Deterministic Multiobjective Programming Problems along with methods for solving them.
In Chapter 3 we discuss Multiobjective Programming Problems under fuzziness. Ample room is allotted to two ways for solving a multiobjective program with fuzzy parameters. The first way is based on a correspondence between fuzzy numbers and their α-level sets [54]. The second relies heavily on possibility measures of uncertainty [16].
A mini Decision Support System aiming at helping a decision maker facing a problem that may be cast into a multiobjective programming setting is presented in Chapter 4 along with a case study related to the Rwandan socio-economic tissue
The dissertation ends with some concluding remarks including lines for further development in the field of Multiobjective Programming.

CHAPTER 1: PRELIMINARIES

In this dissertation we will discuss, among other things, how important ideas form Mathematical Programming and Fuzzy set Theory can be weaved synergically in order to address deterministic Multiobjective Programming Problems as well as multiobjective programming under uncertainty.
Therefore we find it relevant to introduce, in this chapter, basic concepts of Mathematical Programming and Fuzzy set Theory that are needed in the sequel.

1.1 Mathematical Programming

1.1.1 Preamble

Mathematical Programming is the field of Applied Mathematics that studies problems of optimization of a real-valued function over a domain given by mathematical relations.
Mathematical Programming has been successfully used for years in a variety of problems related to hard systems in which the structure, relations and behavior are well-defined and quantifiable.
In this section we'll briefly discuss basic ideas of Mathematical Programming that are needed in subsequent developments.

1.1.2 Problem formulation

A mathematical program is a problem of the type:

$$(\mathrm{P}_1)\begin{cases}\min\ f(x) \\ g_i(x) \le 0\ ; \qquad i \in I = \{1, \dots, m\}, \\ \end{cases}$$

where f and g_i are real-valued functions defined on $\mathbb{R}^n$.
In the sequel the set of feasible solutions of (P_1) is denoted by X. That is :

$$X = \{x \in \mathbb{R}^n /\ g_i(x) \le 0\ ;\ i \in I = \{1,..,m\}\}.$$

If there exists $x^0 \in X$ such that $g_i(x^0) = 0$, then the constraint $g_i(x) \le 0$ is said to be saturated at x^0.

A particular case of greatest interest, is when f and g_i are linear functions [30], [36].
In this case (P_1) reads:

$$(\mathrm{P}_2)\begin{cases}\min cx \\ Ax\ \le b \\ x \ge 0\end{cases}$$

where c ∈ $\mathbb{R}^n$, b∈ $\mathbb{R}^m$ and A is a nonsingular m x n matrix.
(P_2) is merely called linear program.
In what follows, we are going to discuss the main mathematical problems related to (P_1) and (P_2). Namely, the existence of a solution, the eventual unicity of this solution and the construction of such a solution.

1.1.3 Existence and unicity of a solution of a mathematical program

Theorem 1.1 (Weierstrass)

Consider (P_1) *and assume that f is continuous . Suppose also that X is compact. Then* (P_1) *has an optimal solution* x^*.

Proof

Let $\mathrm{m} = \inf_{x \in X}\{f(x)\}$. Then there exists a sequence $\{\mathrm{x}^{\mathrm{k}}\}_{k \in \mathbb{N}}$ of elements of X such that: $\mathrm{f}(\mathrm{x}^{\mathrm{k}}) \longrightarrow \mathrm{m}$. Since X is compact, there exists a sub-sequence $\{x^l\}_{l \in L}$ ($L \subset \mathbb{N}$) which converges to x^*;
Since f is continuous, we have $\mathrm{f}(x^l) \longrightarrow \mathrm{f}(x^*)$. We can therefore write:

$$m = \lim_{k \to \infty} f(x^k) = \lim_{\substack{l \to \infty \\ l \in L}} f(x^l) = \mathrm{f}(x^*)$$

Since $\mathrm{f}(x^*) > -\infty$, we have m > - ∞ and $\forall x \in X$: $\mathrm{f}(x^*) = \mathrm{m} \leq \mathrm{f}(x)$; this means that x^* is a solution of (P_1).

Remark 1.1

It is well known (see e.g [3], [9]) *that if f is strictly convex, then the solution of* (P_1) *is unique.*
For more details on these matters, the reader may consult [1], [10].

1.1.4 Necessary optimality conditions

In what flows, we assume that functions f and g_i (i=1,…,m) are differentiable. We also suppose that the feasible set X is not empty.

Definition 1.1 (Admissible curve at x^0)

If $x^0 \in X$ *is a local optimum of* (P_1)*, then f(x) cannot decrease when x describes an arc of curve* Γ *starting at* x^0 *and contained in the feasible set X.*
Such an arc of curve is called admissible at x^0.
It will be defined by a differentiable function:

$$\varphi : \mathbb{R} \longrightarrow \mathbb{R}^n$$
$$\theta \rightsquigarrow \varphi(\theta) = (\varphi_1(\theta), \varphi_2(\theta), \dots, \varphi_n(\theta))$$

verifiying the following conditions:

(a) $\varphi(0) = x^0$;
(b) for sufficiently small $\theta > 0$, $\varphi(\theta) \in X$.

Definition 1.2 (Admissible direction at x^0)

An admissible direction at x^0 *is a vector*

$$y = \frac{d\varphi}{d\theta}(0) = (\frac{d\varphi_1}{d\theta}(0), \frac{d\varphi_2}{d\theta}(0), \dots, \frac{d\varphi_n}{d\theta}(0))^T,$$

which is tangential to an arc of curve $\varphi(\theta)$ admissible at x^0.
Before proceeding further we need the following notation.

Notation

C_{ad} denotes the cone of all admissible directions at x^0; that is
$C_{ad}=\{ y \mid \nabla g_i^T(x^0).y \leq 0 \quad i = 1, \dots, m\}$
I^0 stands for the set of indices of constraints saturated at x^0 ; that is
$I^0=\{i \in I \ / \ g_i(x^0) = 0\}$
$G = \{y \mid \nabla g_i^T(x^0).y \leq 0 \ \ \forall i \epsilon I^0\}$.

Lemma 1.1
$C_{ad} \subset G$

This result derives trivially from the definition of C_{ad} and G as a matter of fact $I^0 \subset \{1,\dots, m\}$

1.1.5 Constraint qualification

We say that the domain X defined by constraints $g_i(x) \leq 0 \ \ i \in I$ satisfies the hypothesis of constraint qualification in $x^0 \in X$ if and only if

$$cl\,(C_{ad}) = G \qquad \text{(QC)}$$

where cl stands for closure.

Obviously, the direct verification of (QC) might be difficult in practice. This is why one has looked for sufficient conditions for (QC) to hold. The most important results in this regards are collected in the following lemma, the proof of these results may be found in [37].

Lemma 1.2

For (QC) to hold at every point $x \epsilon X$, it is sufficient that one of the following conditions holds:

(a) all the functions g_i are linear,
(b) all the functions g_i are convex and X has nonempty interior.

Lemma 1.3

For (QC) to hold at a point $x^0 \in X$, it is sufficient that :
(c) the gradients $\nabla g_i(x^0)$ $(i \epsilon I^0)$ are linearly independent.

1.1.6 Sufficient Optimality conditions

Consider again (P_1) and define the Lagrangian of (P_1) as follows:

$$L(x,\lambda) = f(x) + \sum_{i\in I} \lambda_i g_i(x).$$

Where $\lambda_i \in \mathbb{R}$ (λ_i i=1,…,m are called Lagrangian multiplicators)

Definition 1.3

Let $\bar{x} \in X$ and $\bar{\lambda} \geq 0$. We say that $(\bar{x},\bar{\lambda})$ is a saddle-point of $L(x,\lambda)$ if:
$L((\bar{x},\lambda) \leq L(\bar{x},\bar{\lambda}) \leq L(x,\bar{\lambda})$, $\forall \in X$, $\forall \lambda \geq 0$.

Theorem 1.2

Let $\bar{x} \in X$ and $\bar{\lambda} \geq 0$. $(\bar{x},\bar{\lambda})$ is a saddle-point for $L(x,\lambda)$ if and only if:

(a) $L(\bar{x},\bar{\lambda}) = \min_{x\in X} L(x,\bar{\lambda})$
(b) $g_i(\bar{x}) \leq 0 \quad \forall\ i\epsilon I$
(c) $\bar{\lambda}_i g_i(\bar{x}) = 0 \quad \forall\ i\epsilon I.$

Proof

(1) If $(\bar{x},\bar{\lambda})$ is a saddle-point of $L(x,\lambda)$, then (a) must be true.

On the other hand, we have

$$\forall\lambda \geq 0 \qquad L(\bar{x},\bar{\lambda}) \geq L(\bar{x},\lambda),$$

hence

$$f(\bar{x}) + \sum_{i\in I} \bar{\lambda}_i g_i(\bar{x}) \geq f(\bar{x}) + \sum_{i\in I} \lambda_i g_i(\bar{x}),$$

therefore

$$\forall\lambda \geq 0: \ \sum_{i\in I} \left(\lambda_i - \bar{\lambda}_i\right) g_i(\bar{x}) \leq 0. \qquad (*)$$

If for some subscript i, (b) does not hold, then we can always choose $\lambda_i > 0$ sufficiently large so that (*) does not hold (contradiction).
Hence (b) is should be true.
Finally, for $\lambda = 0$, (*) implies $\sum_{i\in I} \bar{\lambda}_i g_i(\bar{x}) \geq 0$, but $\bar{\lambda}_i \geq 0$ and $g_i(\bar{x}) \leq 0$ implies $\sum_{i\in I} \bar{\lambda}_i g_i(\bar{x}) \leq 0$.

Hence $\sum_{i\in I} \bar{\lambda}_i g_i(\bar{x}) = 0$.
This last relation implies $\bar{\lambda}_i g_i(\bar{x}) = 0$ ($\forall\ i \in I$) and (c) is proved.

(2) Conversely: assume the conditions (a), (b) and (c) hold true.

(a) $\Rightarrow L(\bar{x},\bar{\lambda}) \leq L(x,\bar{\lambda}) \quad \forall\ x \in X.$

On the other hand (c) $\Rightarrow L(\bar{x},\bar{\lambda}) = f(\bar{x})$.

Finally,

$$L(\bar{x},\lambda) = f(\bar{x}) + \sum_{i\in I} \lambda_i g_i(\bar{x}) \leq f(\bar{x}) = L(\bar{x},\bar{\lambda}) \qquad \forall\lambda \geq 0.$$

Hence

$$L(\bar{x},\lambda) \leq L(\bar{x},\bar{\lambda}) \leq L(x,\bar{\lambda}) \qquad \forall\ \lambda \geq 0\ and\ \forall x \in X\ ,$$

which completes the proof.

Theorem 1.3

If $(\bar{x},\bar{\lambda})$ is a saddle-point of $L(x,\lambda)$, then $\bar{x}$ is a global optimum of (P_2).

Proof
Condition (a) of the preceding theorem implies

$$f(\bar{x}) + \sum_{i\in I} \bar{\lambda}_i g_i(\bar{x}) \leq f(x) + \sum_{i\in I} \bar{\lambda}_i g_i(x) \qquad \forall\ x \in X.$$

On other hand, by (c) we have:

$$\bar{\lambda}_i g_i(\bar{x}) = 0 \quad \forall i \in I.$$

Hence

$$f(\bar{x}) \leq f(x) + \sum_{i\in I} \bar{\lambda}_i g_i(x) \quad (\forall\ x \in X)$$

and since $\bar{\lambda} \geq 0$, we have , $\sum_{i\in I} \bar{\lambda}_i g_i(x) \leq 0$ $(\forall\ x \in X)$ and therefore

$f(\bar{x}) \leq f(x) \qquad \forall\ x \in X$, as desired.

1.1.7 Algorithms for solving Mathematical Programming Problems

In this section we are going to describe the simplex method, which is a technique for solving a linear program. We also present the gradient projection method which is a scheme for solving a nonlinear program.
For other methods for linear programming we refer the readers to [25] and [26] where interior-point and ellipsoid methods are discussed.
As far as nonlinear programming problems are concerned, there is a plethora of techniques for solving them (see e.g. [39], [53]).

1.1.7.1 Simplex method

Consider the linear program (P_2) and assume that constraints are in the form of equalities after (eventually) slack variables have been added.
Then (P_2) takes the following form:

$$(\mathrm{P}_3)\begin{cases}\min cx\\ Ax = b\\ x \geq 0\end{cases}$$

where A is a m' x n' matrix of rank m', b$\in \mathbb{R}^{m'}$ and c$\in \mathbb{R}^{n'}$.

A basis B for this program is a m' x m' matrix such that det B$\neq 0$.
We assume, without loss of generality, that B is obtained by the m' first columns of the matrix A.
A basic solution corresponding to B is a solution of the form:

$$(x_B, 0, \ldots, 0) \in R^{n'}$$

where $x_B = B^{-1}b \in R^{m'}$.

In this dissertation I(B) stands for the indices of columns of B (here I(B)= {1,2,…,m"}).
Given a basis B we define Z_0 and Z_j

$$Z_0 = \sum_{i \in I(B)} c_i b_i,$$

$$Z_j = \sum_{i \in I(B)} c_i a_{ij}.$$

We are now in a position to describe the simplex method [14].

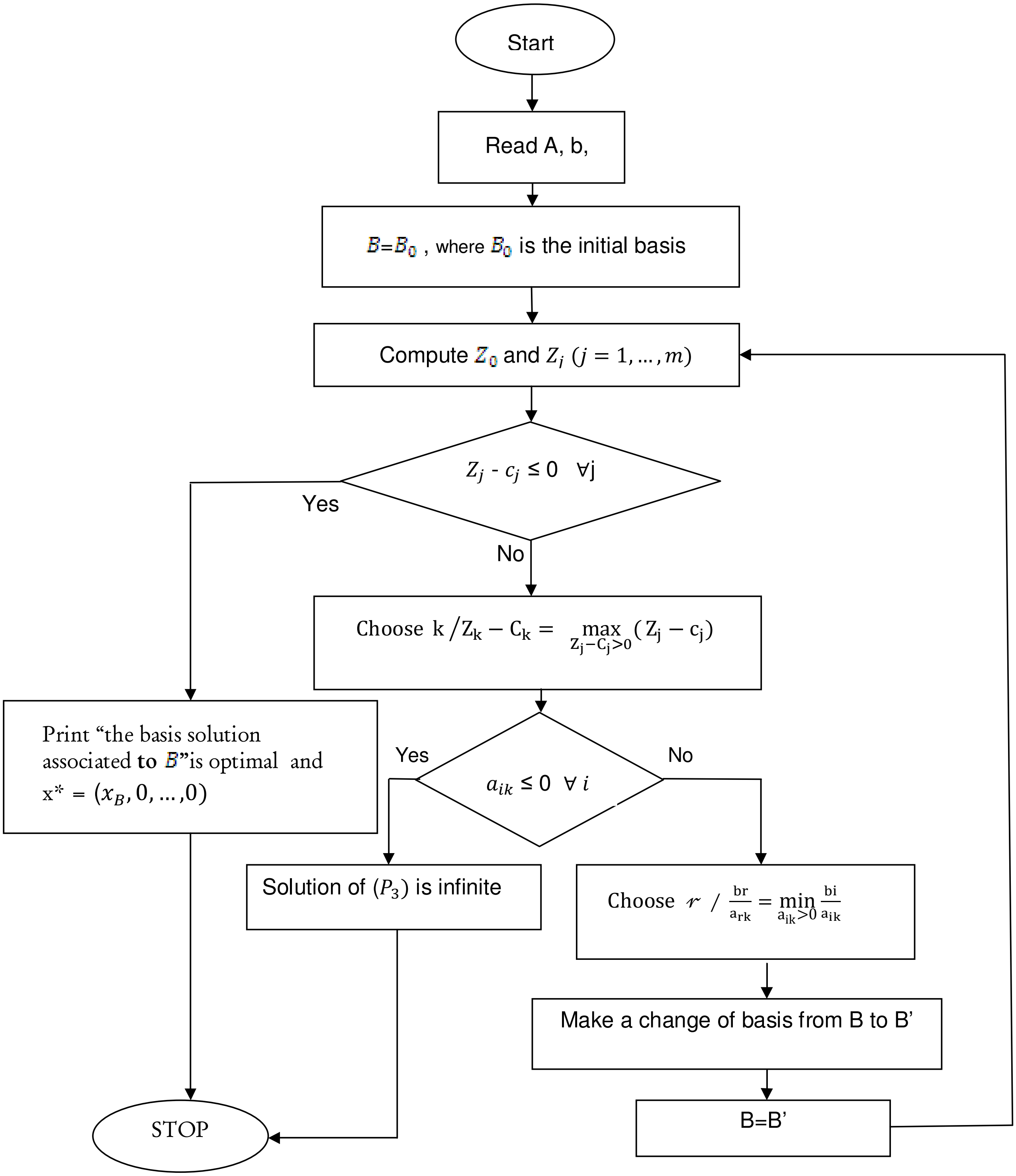


Figure 1.1: Flowchart for the simplex method for a min linear program

Remarks

- For the formula of change of basis, the reader may consult [14].
- For a max problem, the test $Z_j - c_j \leq 0 \quad \forall j$ should be replaced by

  $Z_j - c_j \geq 0 \quad \forall j$.

- k should be chosen so that $\mathrm{Z_k} - \mathrm{c_k} = \min\limits_{z_j - c_j < o} (\mathrm{Z_j} - \mathrm{c_j})$.
- Other approaches for solving linear programming problems include the Karmarkar method [25] and the Khachiyan algorithm [21].

1.1.7.2 Gradient projection method

Here we consider the mathematical program of the form:

$$(\boldsymbol{P_4}) \begin{cases} min\ f(x) & \\ a_i x \leq\ b_i & i \in\ I_1 \\ a_i x =\ b_i & i \in I_2 \\ x \in \mathbb{R}^n. & \end{cases}$$

The gradient method for this problem proceeds as follows:

(a) At step k=0, a vector x^0 satisfying the constraint of the problem is chosen.

(b) At the current step k we have x^k. Find the set $I^0(x^k)$ = {k$\in I$ / $g_k(x^0) = 0$}
Set L^0= $I^0(x^k)$.

(c) Let A^0 be the matrix whose rows correspond to the constraints i$\in L^0$.
Compute the projection matrix

P^0= I–$A^{0T}.[A^0 A^{0T}]^{-1}.A^0$ where I is identity matrix and put:
y^k = $-P^0 \nabla f(x^k)$
If $y^k = 0$ go to (e).

(d) Compute
α_{max}= max $\{\alpha / x^k + \alpha\ y^k \in X\}$ and put:

$x^{k+1}\ such\ that$ f(x^{k+1})= $\min\limits_{0 \leq \alpha \leq \alpha_{max}} \min\{\mathrm{f}(x^k + \alpha\ y^k)\}$

Set k= $k + 1$ and return to (b).

(e) Let u= $-A^{0T}.[A^0 A^{0T}]^{-1}.A^0.\nabla f(x^k)$.

If u ≥ 0, print: x^k solution of ($\mathrm{P_4}$) go to (f).
Otherwise let u_i be the most negative component of u.
Set $L^0 = L^0 - \{i\}$ and return to (c).

(f) END.
It has been proved [1] that if u≥0 then x^k is optimal for (P_4).

1.2 Fuzzy set Theory

1.2.1 Preamble

Although Probability Theory claims to cope with uncertainty, there is a qualitatively different kind of imprecision that is not covered by probabilistic apparatus. Namely: inexactness, ill-definedness, vagueness.
Situations where doubt arises about the exactness of concepts, correctness of statements and judgments, degree of credibility, have little to do with occurrence of events, the back-bone of Probability Theory.
It turns out that Fuzzy set Theory [29], [55], [56] offers a proper framework for coming to grips with the above mentioned non-stochastic imprecision.
The following subsection introduce basic notions of Fuzzy set Theory that we need in subsequent discussions.

1.2.2 Fuzzy set

The main idea behind a fuzzy set is that of gradual membership to a set without sharp boundary.
This idea is in tune with human representation of reality that is more nuanced than clear-cut.
Some philosophical related issues ranging from ontological level to application level via epistemological level may be found elsewhere [47].

In a fuzzy set, the membership degree of an element is expressed by any real number from 0 to 1 rather than the limiting extremes.
More formally, a fuzzy set of a set $A \neq \phi$ is characterized by a membership function:

$$\mu: A \rightarrow [0,1].$$

In what follows a fuzzy set will be identified with its membership function.
Moreover, for our purposes, we restrict ourselves to fuzzy sets of the real line $\mathbb{R}$.

1.2.3 Some notions related to fuzzy sets of $\mathbb{R}$

- The support of a fuzzy set μ is the crisp set denoted by supp (μ) and defined as follows:

$$\text{supp}\,(\mu) = \overline{\{x \in \mathbb{R} \mid \mu(x) > 0\}}.$$

- The kernel of a fuzzy set μ is the crisp set denoted by Ker (μ) and defined as follows:

$$\text{Ker}\,(\mu) = \{x \in \mathbb{R} \mid \mu(x) = 1\}.$$

- A fuzzy set μ is said to be normal if Ker $(\mu) \neq \phi$.

- The α-cut or α-level set of a fuzzy set μ is the crisp set μ_α defined as follows:

$$\mu_\alpha = \{x \in \mathbb{R} \mid \mu(x) \geq \alpha\}.$$

- The strict α-level set of a fuzzy set μ, in symbol $\mu_{\tilde{\alpha}}$, is defined as follows:

$$\mu_{\tilde{\alpha}} = \{x \in \mathbb{R} \mid \mu(x) > \alpha\}.$$

- A fuzzy set μ is said to be convex if $\mu(x)$is a quasi-concave function.

- A fuzzy number is a normal and convex fuzzy set of $\mathbb{R}$.

A fuzzy number is well suited for representing a vague datum [16].
For instance the vague datum: "close to five" can be represented by the fuzzy number μ as in Fig 2.

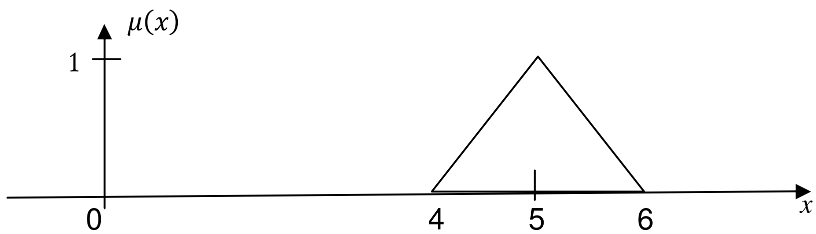


Figure 1.2: Membership function of the vague datum: "close to 5".

1.2.4 Properties of α-level sets

- For a fuzzy number μ, the following property holds:

$$\text{If } 0<\alpha \leq \beta \leq 1, \text{ then } \mu_\beta \subseteq \mu_\alpha.$$

- Let μ be a normal fuzzy set of $\mathbb{R}$. A family of subsets of $\mathbb{R}$, $\{\mu_\alpha | \alpha \in (0,1)\}$ is called a set representation of μ, if and only if:

$$\text{(i) } 0< \alpha \leq \beta < 1 \Longrightarrow \mu_\beta \subseteq \mu_\alpha$$

$$\text{(ii) } \forall t \in \mathbb{R},\ \mu(t) = \text{Sup}\{\alpha I_{\mu_\alpha} | \alpha \in (0,1)\}$$

where I_P stands for the characteristic function of P, i.e

$$I_P(x) = \begin{cases} 1 \text{ if } x \in P \\ 0 \text{ otherwise.} \end{cases}$$

Theorem 1.4

Let μ be a fuzzy number. Then $\{\mu_\alpha | \alpha \in (0,1)\}$ and $\{\mu_{\tilde{\alpha}} | \alpha \in (0,1)\}$ are set representation of μ.

Proof

Obviously μ_α and $\mu_{\tilde{\alpha}}$ are not empty for all $\alpha \in (0,1)$ and $0 < \alpha \leq \beta < 1 \Longrightarrow \mu_\beta \subseteq \mu_\alpha$ and $\mu_{\tilde{\beta}} \subseteq \mu_{\tilde{\alpha}}$
Let $t \in \mathrm{R}$ be arbitrary. We want to show that

$$\mu(t) \leq \sup\{\alpha I_{\mu_\alpha}(t)/\alpha \in (0,1)\} \text{ and}$$

$$\mu(t) \geq \sup\{\alpha I_{\mu_{\tilde{\alpha}}}(t)/\alpha \in (0,1)\} \text{ is valid.}$$

For abbreviation, define $\gamma \overset{\text{def}}{=} \mu(t)$. If $\gamma = 0$, the first assertion is obvious.
If $\gamma > 0$, then $t \in \mu_{\gamma-\varepsilon}(t)$ is valid for $\varepsilon > 0$. This implies

$$(\gamma - \varepsilon)I_{\gamma-\varepsilon}(t) = \gamma - \varepsilon$$

for all $\varepsilon > 0$, and the first assertion follows.
For all $\alpha \in (0,1)$ we have

$$I_{\mu_{\tilde{\alpha}}}(t) = \begin{cases} 1, & if\ \alpha \in (0,\gamma) \\ 0, & if\ \alpha \in (\gamma,1). \end{cases}$$

This implies

$$\alpha I_{\mu_{\tilde{\alpha}}}(t) = \begin{cases} \alpha, & if\ \alpha \in (0,\gamma) \\ 0, & if\ \alpha \in (\gamma,1) \end{cases},$$

and

$$\alpha I_{\mu_{\tilde{\alpha}}}(t) \leq \gamma I_{\gamma_{\tilde{\alpha}}}(t) = \gamma = \mu(t).$$

This demonstrates the second assertion and implies the two inequalities

$$\mu(t) \leq sup\{\alpha I_{\mu_\alpha} \mid \alpha \in (0,1)\} \leq sup\{\alpha I_{\mu_{\tilde{\alpha}}} \mid \alpha \in (0,1)\} \leq \mu(t),$$

as $\mu_\alpha \subseteq \mu_{\tilde{\alpha}}$ is valid for $\alpha \in (0,1)$.
This completes the proof.

Theorem 1.5

Let μ be a fuzzy number and $\{A_\alpha | \alpha \in (0,1)\}$ be its set representation. Then, we have for all $\alpha \in (0,1)$

$$\lim_{r\to\infty} \inf A_{\left(\alpha+\frac{1}{(2r)(1-\alpha)}\right)} = \inf \mu_\alpha$$

and

$$\lim_{r\to\infty} \sup A_{\left(\alpha+\frac{1}{(2r)(1-\alpha)}\right)} = \sup \mu_\alpha.$$

Proof

Let $\alpha \in (0,1)$.

For abbreviation define $\alpha_r \overset{\text{def}}{=} \alpha + \frac{1}{(2r)(1-\alpha)}$ for r ∈ $\mathbb{N}$.

For r,s ∈ N r ≤ s implies $\alpha_r \geq \alpha_s$, and by this $A_{\alpha_r} \subseteq A_{\alpha_s}$, holds.

Therefore $\{infA_{\alpha_r}\}_{r\in\mathbb{N}}$ is monotonously decreasing and $\{supA_{\alpha_r}\}_{r\in\mathbb{N}}$ is monotonously increasing.
We know that [38]

$$\inf\mu_\alpha \leq \inf A_{\alpha_r} \leq sup\ A_{\alpha_r} \leq \sup\mu_\alpha \text{ for } r \in \mathbb{N} \text{ holds.}$$

If inf $\mu_\alpha > -\infty$, then $\{infA_{\alpha_r}\}_{r\in\mathbb{N}}$ is convergent, let A denote the limit. A ≥ inf μ_α is obvious.
Let $\varepsilon > 0$ be arbitrary. There exists x ∈ μ_α with x ≤ inf μ_α+ε.
Then we can find [38] and r ∈ $\mathbb{N}$ with x ∈ A_{α_r} , i.e.

$$infA_{\alpha_r} \leq x \leq \inf\mu_\alpha + \varepsilon$$

Therefore

$$A = \lim_{r\to\infty} infA_{\alpha_r} \leq \inf\mu_\alpha \quad \text{follows.}$$

If inf μ_α = - ∞ and n ∈ $\mathbb{N}$ are arbitrary, there is an x ∈ μ_α with x < - n.
We can find r ∈ $\mathbb{N}$ with x ∈ A_{α_r} and can follows : $infA_{\alpha_r} \leq -n$. As $\{infA_{\alpha_r}\}_{r\in\mathbb{N}}$ is decreasing , it converges against - ∞.
The second assertion can be shown in a similar way.

1.2.5 Operations on fuzzy sets

- Consider two fuzzy sets of $\mathbb{R}$, μ_1 and μ_2.
- The complement of μ_1 is defined as $\bar{\mu}_1$ where $\bar{\mu}_1(x) = 1 - \mu_1(x)$.
- The union of μ_1 and μ_2 is defined as $\mu_1 \vee \mu_2$ where $(\mu_1 \vee \mu_2)(x) = \max(\mu_1(x), \mu_2(x))$.
- The intersection of μ_1 and μ_2 is defined as $\mu_1 \wedge \mu_2$ where $(\mu_1 \wedge \mu_2)(x) = min(\mu_1(x), \mu_2(x))$.
- Let $*$ be an arithmetic operation on real numbers a and b.
  An extension of $*$ to fuzzy numbers $\mu_{\tilde{a}}$ and $\mu_{\tilde{b}}$ is as follows:

$$(\mu_{\tilde{a}} * \mu_{\tilde{b}})(r) = \sup_{\substack{s,t \\ s*t=r}} \min(\mu_{\tilde{a}}(s), \mu_{\tilde{b}}(t)) \text{ [14].}$$

1.2.6 Possibility, Necessity and Credibility measures

Let ⊖ be a nonempty set representing the sample space. A possibility measure is a function

$$\text{Pos} : 2^{\ominus} \to [0,1]$$

satisfying the following axioms:

(i) $\text{Pos}\{\ominus\} = 1$,

(ii) $\text{Pos}\{\phi\} = 0$,

(iii) $\text{Pos}\{\cup_i A_i\} = sup_i Pos\{A_i\}$,

(iv) Let $\{\ominus_k\}_k$ be a family of sets and $\text{Pos}_k: 2^{\ominus_k} \to [0,1]$ verify (i) – (iii) and $\ominus = \ominus_1 \times \ominus_2 \ldots \times \ominus_n$.

Then for $A \subset \ominus$, $\text{Pos}\{A\} = \sup_{\ominus_1,\ldots,\ominus_n \in A} \min_{1 \le k \le n} \text{Pos}_k\{\ominus_k\}$.

Necessity and Credibility measures are obtained from Possibility measure as follows: For $A \in 2^{\ominus}$, we have:

$$\text{Nec}\{A\} = 1 - \text{Pos}(A^C)$$

and

$$\text{Cr}\{A\} = \frac{\text{Pos}\{A\} + \text{Nec}\{A\}}{2}$$

where A^c is the complement of A.

Details on Possibility, Necessity and Credibility measures may be found elsewhere [29].

These measures are used in the literature of decision making under uncertainty either to appraise the level of desirability of a decision or to transform a decision problem under uncertainty into deterministic terms [16].

CHAPTER 2: DETERMINISTIC MULTIOBJECTIVE PROGRAMMING PROBLEMS

2.1 Analysis

2.1.1 Problem setting and general notation

A general multiobjective program is a problem of the type:

$$(P_5)\begin{cases} min[f_1(x), \dots, f_k(x)],\ k \geq 2 \\ x \in X = \{x \in \mathbb{R}^n / g_j(x) \leq 0; j = 1, \dots, m\} \end{cases}$$

where $f_i(x); i = 1, \dots, k$ and $g_j(x), j = 1, \dots, m$ are real-valued functions of $\mathbb{R}^n$.
Many real-life problems may be cast in the form of (P_5). An interested reader is referred to [4], [7] for examples of concrete problems that may be formulated as Multiobjective Programming Problems.

2.1.2 Background materials

Throughout this chapter, all the vectors are assumed to be column vectors. If x and x^* are vectors of $\mathbb{R}^n$, the notation $x^T x^*$ (where T stands for the transpose) denotes the scalar product of x and x^*. Moreover, the following notation is used:

$$\mathbb{R}^n_+ = \{x = (x_1 \dots, x_n) \in \mathbb{R}^n \mid x_i \geq 0,\ i = 1, \dots, n\}$$

$$\|x\| = (\textstyle\sum_{i=1}^n x_i^2)^{½}.$$

For $X \subseteq \mathbb{R}^n$,

$$\text{dist}\,(x^*, X) = inf_{x \in X} \parallel x - x^* \parallel$$

$$B(x^*, \delta) = \{x \in \mathbb{R}^n : \parallel x - x^* \parallel < \delta\}.$$

The following definitions are needed in the sequel.

Definition 2.1

Let d $\in \mathbb{R}^n, d \neq 0$, *we say that d is a feasible direction emanating from x* $\in$ *X if there exists* $\alpha^* > 0$ *such that* $x + \alpha d \in X\ for\ all\ \alpha \in [0, \alpha^*]$.

Definition 2.2

A constraint $g_j(x) \leq 0$ *is said to be active at a point* $x^* \in X$ *if* $g_j(x^*) = 0$.
A set of active constraints at x^* *is denoted by* $J(x^*)$.

In other words,

$$\mathrm{J}(\mathrm{x}^*) = \{\mathrm{j} \in \{1, \dots, \mathrm{m}\} / \mathrm{g_j}(\mathrm{x}^*) = 0\}.$$

2.1.3 Convex multiobjective program

Definition 2.3

A function f : $\mathbb{R}^n \to \mathbb{R}$ is convex if for all x , $y \in \mathbb{R}^n$ and $\alpha \in [0,1]$ we have

$$f(\alpha x + (1-\alpha)y) \leq \alpha f(x) + (1-\alpha)f(y).$$

Definition 2.4

A set C $\subseteq \mathbb{R}^n$ is convex if for all x, $y \in$C and $\alpha \in [0,1]$, we have

$$\alpha x + (1-\alpha)y \in \text{C}.$$

Definition 2.5

Consider the multiobjective program (P_5).
If the feasible set X is convex and if the objective functions $f_i(x)$; i=1,…,k are convex, then (P_5) is said to be a convex multiobjective program.

As it will become evident in Section 2.2, most of the methods described for multiobjective programs are applicable only to convex multiobjective programs involving differentiable functions up to order 2. In this chapter, we'll also indicate what can be done if the above mentioned conditions are not satisfied.

2.1.4 Solution concepts used in multiobjective programming

In a multiobjective optimization context, unless the objective functions are not conflicting, the optimum optimorum lies outside the feasible set. It is therefore worthwhile to make explicit the meaning of optimality in this context. Several solutions concepts are discussed in the literature [8] ,[36] ,[40], [43]. We briefly discuss these solution concepts in the following subsections.

2.1.5 Pareto optimality

Definition 2.6

$x^* \in X$ is said to be a Pareto optimal (efficient) solution *for **(P_5)** if no other $x \in X$ exists, such that $f_i(x) \leq f_i(x^*)$ for all $i = 1, \dots, k$ and*
$f_\ell(x) < f_\ell(x^*)$ for som ℓ.

Remark 2.1

A Pareto optimal solution is also called a globally Pareto optimal solution.

Definition 2.7

Consider the multiobjective program *(P_5)**.** $x^* \in X$ is called locally Pareto optimal for* (P_5) *if there is $\delta > 0$ exists such that x^* is Pareto optimal for (P_5) in $X \cap B(x^*, \delta)$.*

Theorem 2.1

Assume that the multiobjective program (P_5) is convex. Then every locally Pareto optimal solution for (P_5) is also globally Pareto optimal for (P_5).

Proof

Assume (P_5) is convex and let $x^* \in X$ be locally Pareto optimal for (P_5). We can then find $\delta > 0$ and a neighborhood $B(x^*, \delta)$ of x^* such that there is no $x \in X \cap B(x^*, \delta)$ for which $f_i(x) \leq f_i(x^*)\ for\ all\ i = 1, \ldots, k$ and $f_\ell(x) < f_\ell(x^*)$ for some ℓ.
Assume that x^* is not globally Pareto optimal for (P_5). Then there exists some other point $x^0 \in X$ such that:

$$f_i(x^0) \leq f_i(x^*)\ for\ all\ i = 1, \ldots, k \text{ and } f_\ell(x^0) < f_\ell(x^*) \text{ for some } \ell. \quad (2.1)$$

Let $\hat{x} = \beta x^0 + (1-\beta)x^*$, where $\beta \in (0,1)$ is chosen such that $\hat{x} \in B(x^*, \delta)$.
Then, by the convexity of *X* and the convexity of $f_i;\ i = 1, \ldots, k,$ we have:

$$\hat{x} \in X \quad (2.2)$$

and

$$f_i(\hat{x}) \leq \beta f_i(x^0) + (1-\beta) f_i(x^*) \leq \beta f_i(x^*) + (1-\beta) f_i(x^*) = f_i(x^*) \quad (2.3)$$

for every $i = 1, \ldots, k$.

Since x^* is locally Pareto optimal for (P_5) and $\hat{x} \in B(x^*, \delta)$, we must also have

$$f_i(\hat{x}) = f_i(x^*);\ i = 1, \ldots, k. \quad (2.4)$$

Moreover, $f_i(x^*) \leq \beta f_i(x^0) + (1-\beta) f_i(x^*)$ for every $i = 1, \ldots, k$, because x* is locally Pareto optimal for (P_5). That is:

$$\beta f_i(x^*) \leq \beta f_i(x^0) \text{ for every } i=1,\ldots,k. \quad (2.5)$$

Since $\beta > 0$, we can use it to divide and obtain $f_i(x^*) \leq f_i(x^0)$ for all $i = 1, \ldots, k$.
But $f_\ell(x^*) > f_\ell(x^0)$ according to (2.1) for some ℓ. This is a contradiction. Therefore x^* is globally Pareto optimal for (P_5).

Corollary 2.1

For convex multiobjective programs, Pareto optimality and locally Pareto optimality coincide.

2.1.6 Weak and proper Pareto optimality

Definition 2.8

$x^* \in X$ is weakly Pareto optimal for (P_5) if there is no $x \in X$ such that $f_i(x) < f_i(x^*)$ for all $i = 1, \ldots, k$.

Corollary 2.2

If x is Pareto optimal for (P_5) then x* is weakly Pareto optimal for (P_5).*

Definition 2.9

$x^* \in X$ *is properly Pareto optimal for (P_5) if it is Pareto optimal for (P_5) and if there does not exist any point d* $\in \mathbb{R}^n$ *such that:*

$$\nabla f_i^T(x^*).d \le 0 \qquad \text{for all } i=1,\dots,k,$$
$$\nabla f_j^T(x^*).d < 0 \qquad \text{for some } j$$

and

$$\nabla g_\ell^T(x^*).d \le 0 \quad \text{for all } \ell \text{ satisfying } g_\ell(x^*) = 0.$$

For more details on proper efficiency the reader is referred to [36].

2.1.7 Ideal point

Definition 2.10

*Consider the multiobjective program (P_5)**.** For each of the k objective functions, there is one different optimal solution. An objective vector constructed with these individual optimal objective values constitutes the ideal objective vector denoted by* z^*

This means, $z^* = (z_1^*, \dots, z_k^*)$ where $z_i^* = f_i(x^*)$ and x^* is a solution of

$$\begin{cases} \min f_i(x) \\ x \in X \end{cases}$$

A point $x^0 \in X$ such that $f_i(x^0) = z_i^* \ \forall\, i$ is called ideal point for (P_5).

Remark 2.2

Unless the objective functions of (P_5) are not conflicting, the ideal point is outside the feasible domain X.

2.1.8 Kuhn and Tucker conditions for Pareto optimality

In this section, we discuss necessary and sufficient conditions for Pareto optimality for a multiobjective program. We assume that all the functions involved are continuously differentiable.

Theorem 2.2

A necessary condition for $x^* \in X$ *to be Pareto optimal for (P_5) is that there are vectors* $\lambda \in \mathbb{R}^k$ *and* $\mu \in \mathbb{R}^m$ *with* $\lambda \ge 0$ *and* $\mu \ge 0$ *and* $(\lambda, \mu) \ne (0,0)$*, such that the following conditions hold:*

$$\sum_{i=1}^{k} \lambda_i \nabla f_i(x^*) + \sum_{j=1}^{m} \mu_j \nabla g_j(x^*) = 0 \quad (2.6)$$

$$\mu_j g_j(x^*) = 0; j = 1, \dots, m. \quad (2.7)$$

For the proof of this result, we invite the reader to consult [13].

Equations (2.6) and (2.7) are called Kuhn and Tucker conditions for Pareto optimality. They generalize quite canonically Kuhn and Tucker conditions for optimality for single objective mathematical programs (see §1.1).

Corollary 2.3

For a convex multiobjective program, the conditions given in Theorem 2.2 are also sufficient for Pareto optimality [36].

2.2 Methods

2.2.1 Preamble

This section is devoted to methods for finding a compromise solution for a multiobjective programming problem. A look at the literature (see for example [18], [28], [50]) reveals that the most frequently used methods for solving multiobjective programs fit into five categories:

-no-preference methods,
-a priori methods,
-a posteriori methods,
-interactive methods and
-metaheuristic methods.

In the following sections we briefly discuss the general methodological principle in each of these categories.
It is worth remembering that the problem at hand is:

$$(\mathrm{P}_5)\begin{cases} min[f_1(x), \dots, f_k(x)], \quad k \geq 2 \\ x \in X = \{x \in \mathbb{R}^n | g_j(x) \leq 0; j = 1, \dots, m\} \end{cases}$$

where $f_i(x);\ i = 1, \dots, k\ and\ g_j(x);\ j = 1, \dots, m$ are real-valued functions of $\mathbb{R}^n$.

2.2.2. No-preference methods

One of the best known methods in this category is the Compromise programming (CP) method that is briefly discussed in the sequel.

Definition 2.11

We call the aspiration level z_i , the level of the i^{th} objective function of problem (P_5) that is satisfactory to the decision maker.
If z_i , $i = 1, \dots, k$ are aspiration levels, then $z = (z_1 \dots, z_k)$ is called a reference point.

The compromise programming method, also known as the global criterion method, picks a point in the feasible set X whose vector of objective values is close to some reference point for a given distance [48].
Here we take the ideal vector z^* as the reference point.
The method proceeds as follows:

Step 1
Choose a distance *d,* fix a reference point z^*, choose p and solve one of the following optimization problems according to the value of *p*.

$$(P_6)\begin{cases} \min\left(\sum_{i=1}^{k}[\,f_i(x) - z_i^*]^p\right)^{1/p}, \\ x \in X \end{cases} \qquad if\ p \in [1, \infty)$$

$$(P_7)\begin{cases} min \max\limits_{i=1,\dots,k} |f_i(x) - z_i^*|, \\ x \in X. \end{cases} \qquad if\ p = \infty$$

where z_i^* is the i^{th} component of the reference point.

Step 2
Present the solution obtained to the decision maker.

Step3
Stop.

It is clear that (P_7) is a nondifferentiable multiobjective program. Nevertheless, we can put this program in the following differentiable form:

$$(\mathrm{P}_8)\begin{cases} \min\ \alpha \\ \alpha \geq f_i(x) - z_i^* \\ x \in X. \end{cases}$$

With regard to (P_6), when all the functions involved in (P_5) are linear and *p* =1, then it is a linear program that can be solved by the simplex method (see Chapter 1).
If the functions involved in the constraints are linear, then for *p* = 2, (P_6) is a quadratic program problem that can be solved by the gradient projection method (see Chapter 1).
The next two theorems tell us something about the Pareto optimality of solutions obtained by this method.

Theorem 2.3

If x^* is a solution of (P_6), then x^* is Pareto optimal for (P_5).

Proof

Let $x^* \in X$ be a solution of (P_6) and assume that x^* is not Pareto optimal for (P_5). Then there is a point $x \in X$ such that $f_i(x) \leq f_i(x^*)$ for $i = 1, \ldots, k$ and $f_\ell(x) < f_\ell(x^*)$ for some ℓ.
Now, because $p \in [1, \infty)$, we have that:

$$(f_i(x) - z_i^*)^p \leq (f_i(x^*) - z_i^*)^p;\ i = 1, \ldots, k. \tag{2.8}$$
$$(f_\ell(x) - z_\ell^*)^p < (f_\ell(x^*) - z_\ell^*)^p \text{ for some } \ell. \tag{2.9}$$

From (2.8) and (2.9) we can write that

$$\sum_{i=1}^{k} (f_i(x) - z_i^*)^p < \sum_{i=1}^{k} (f_i(x^*) - z_i^*)^p.$$

Hence

$$\left(\sum_{i=1}^{k} (f_i(x) - z_i^*)^p\right)^{\frac{1}{p}} < \left(\sum_{i=1}^{k} (f_i(x^*) - z_i^*)^p\right)^{\frac{1}{p}}.$$

This is in contradiction to the assumption that x^* is a solution of Problem (P_6). Therefore, x^* is Pareto optimal for (P_5).

Theorem 2.4

Solutions to problem (P_7) contain at least one Pareto optimal solution.

Proof

Les us suppose that none of the optimal solutions of (P_7) is Pareto optimal. Let $x^* \in X$ be an optimal solution of (P_7). Since we assume that it is not Pareto optimal, there must exist a solution $x \in X$ which is not optimal for (P_7) but for which

$$f_i(x) \leq f_i(x^*);\ \ i = 1, \ldots, k$$

and

$$f_\ell(x) < f_\ell(x^*) \text{ for some } \ell.$$

We have now

$$f_i(x) - z_i^* \leq f_i(x^*) - z_i^*;\ i = 1, \ldots, k$$

with the strict inequality holding for one ℓ, and further

$$\max_i [f_i(x) - z_i^*] \leq \max_i\ [f_i(x^*) - z_i^*].$$

Because x^* is an optimal solution for (P_7), x has to be an optimal solution, as well. This contradiction completes the proof.

Example 2.1

Consider the multiobjective program:

$$(\mathbf{P_9})\begin{cases}\min\left[\left(\dfrac{x_1+2x_2}{x_1+x_2+1}\right),\left(\dfrac{2x_1+x_2}{2x_1+3x_2+1}\right)\right]\\ -x_1+2x_2 \leq 3\\ 2x_1 - x_2 \leq 3\\ x_1+x_2 \geq 3\\ 2 \leq x_1 \leq 25\\ 1 \leq x_2 \leq 9{,}5\end{cases}$$

Let us take the ideal vector as reference vector, that is:

$$z^* = (z_1^*; z_2^*)^T = (1; 0, 55).$$

Taking p = 2, putting the problem into the form of (P_6) and squaring the objective function yields the following program:

$$(\mathrm{P}_{10})\begin{cases}\min\ [\dfrac{x_1-1}{x_1+x_2+1}+\dfrac{0{,}9x_1-0{,}65x_2-0{,}55}{2x_1+3x_2+1}]\\ -x_1+2x_2 \leq 3\\ 2x_1 - x_2 \leq 3\\ x_1+x_2 \geq 3\\ 2 \leq x_1 \leq 25\\ 1 \leq x_2 \leq 9{,}5\end{cases}$$

Solving this program, using LINGO software, we obtain the solution $x^* = (3, 3)$ with an optimal value of 0,298214285.
By virtue of Theorem 2.3, this solution is Pareto optimal for (P_5).

The compromise programming method brings extra parameters such as *p* and the reference point into the problem under consideration. This is an inconvenience of this method.

2.2.3 A priori methods

Unlike no-preference methods, the general principle of a priori method is to first take into consideration the opinions and preferences of the decision maker before solving the multiobjective program at hand.

The most used a priori methods are goal programming and lexicographic goal programming methods [5], [22], [42], [44], [45]. These methods are briefly presented in what follows.

2.2.3.1 Goal programming method

The main idea behind the goal programming method is to find solutions that are close to predefined targets [28]. Therefore, in the goal programming method, the decision maker should fix the targets for each objective function. He then solves a single objective program aiming at minimizing the sum of deviations to the targets. Consider (P_5) and assume that b_i is the target for objective *i,* the corresponding single objective program is as follows:

$$(P_{11})\begin{cases} min\left[\sum_{i=1}^{k}(d_i^- + d_i^+)\right] \\ f_i(x) + d_i^- - d_i^+ = b_i; \quad i = 1, \dots, k \\ x \in X \\ d_i^-, d_i^+ \geq 0; \;\; i = 1, \dots, k. \end{cases}$$

The variables d_i^- and d_i^+ are underachievement and overachievement of the i^{th} goal respectively.
Using the goal programming method, the decision maker can allocate some weights to each objective so as to put some hierarchy in the objectives. To this end he can specify real numbers w_i $(i = 1, \dots k)$; w_i >0 and $\sum_{i=1}^{k} w_i$=1.
In this case, the corresponding single objective program is:

$$(P_{12})\begin{cases} min[w_1 h_1(d_1{}^-; d_1{}^+) + \cdots + w_k h_k(d_k{}^-; d_k{}^+)] \\ f_i(x) + d_i^- + d_i^+ = b_i; \quad i = 1, \dots, k \\ x \in X \\ d_i^-, d_i^+ \geq 0; \;\; i = 1, \dots, k. \end{cases}$$

Here $h_i(d_i^-; d_i^+)$; *i* = 1,…,*k* are some linear functions of the deviational variables.
Under certain conditions, the solution to program (P_{12}) is Pareto optimal for (P_5).
This is the subject matter of the following theorem, the proof for which may be found elsewhere [2]

Theorem 2.5

A solution to (P_{12}) $is\ Pareto\ optimal\ for$ (P_5) if *all the deviational variables* d_i^+ *and* d_i^- *have positive values at the optimum.*

The goal programming method can be summarized as follows.

Step 1
Set targets and weights for each objective function of (P_5).

Step 2
Formulate and solve (P_{11}) if weights are equal or formulate and solve (P_{12}) if weights are not equal.

Step 3
Present the solution obtained in Step 2 to the decision maker. If he is happy with the solution, stop. Otherwise go back to Step1 with new targets and new weights.

Example 2.2

Consider the following multiobjective program:

$$(P_{13})\begin{cases}\min[f_1(x), f_2(x), f_3(x), f_4(x), f_5(x)]\\ (x_1-3)^2+(x_2-3)^2+(x_3-3)^2 \le 4\\ x_1+x_2+x_3 \le 15\end{cases}$$

where

$$\begin{aligned} f_1(x) &= (x_1+x_2+x_3)^{\frac{5}{2}}\\ f_2(x) &= (x_1-4)^4+(x_2-3)^4+x_3^4\\ f_3(x) &= 3x_1^3+(x_2-1)^4+2(x_3-20)^4\\ f_4(x) &= (x_1+x_2+x_3)^{-1}\\ f_5(x) &= (x_1-3)^2+\ln(x_3+x_4).\end{aligned}$$

Assume that the targets for each objective have been set by the decision maker as follows:
$b_1 = 80, b_2 = 20, b_3 = 0{,}15, b_4 = 75000\ and\ \ b_5 = 60$. Assume also that the objectives have equal weights.

Then (P_{11}) reads:

$$(P_{14})\begin{cases}\min\left[\sum_{i=1}^{5}(d_i^- + d_i^+)\right]\\ (x_1+x_2+x_3)^{\frac{5}{2}} + d_1^- - d_1^+ = 80\\ (x_1-4)^4+(x_2-3)^4+x_3^4+d_2^- - d_2^+ = 20\\ 3x_1^3+(x_2-1)^4+2(x_3-20)^4+d_3^- - d_3^+ = 0{,}15\\ (x_1+x_2+x_3)^{-1}+d_4^- - d_4^+ = 75000\\ x_1+x_2+x_3 \le 15 + d_5^- - d_5^+ = 60\\ (x_1-3)^2+(x_2-3)^2+(x_3-3)^2 \le 4\\ x_1+x_2+x_3 \le 15\\ d_i^-, d_i^+ \ge 0; i = 1, \dots, k.\end{cases}$$

Solving this program, using LINGO software, yields the following solution:

$d_1^- = d_2^- = d_3^- = d_4^+ = d_5^+ = 0,$
$d_4^- = 74999.9,$
$d_5^+ = 0.278939,$
$d_1^+ = 319.3253,$
$d_2^+ = 606.0207,$
$d_3^+ = 101347.2,$
$x_1 = 2.987495,$
$x_2 = 2.990755,$
$x_3 = 4.99994$
$x_4 = 0.8640776E + 26.$

The major strength of the goal programming method is its simplicity. Although this method offers a great deal of flexibility for solving Multiobjective Programming Problems, there are some difficulties associated with it. These include the fact that it does not always produce Pareto optimal solutions for (P_5). Moreover it may be difficult to set weights or targets for objective functions.

2.2.3.2 Lexicographic goal programming method

The lexicographic programming method deals with situations where the objective functions of a multiobjective program are arranged according to their importance while a target is not given for each objective function.
Consider the multiobjective program (P_5) and assume that:

$$f_1(x) \gg f_2(x) \gg \cdots \gg f_k(x).$$

The above notation means that $f_1(x)$ is more important than $f_2(x)$, $f_2(x)$ is more important than $f_3(x)$ and so on.
The lexicographic goal programming method proceeds as follows.

Step 1
Solve the following mathematical program:

$$(P_{15})\begin{cases}\min f_1(x)\\ \quad x \in X.\end{cases}$$

Let x^* be its solution.
If x^* is unique, then x^* is considered to be the preferred solution to the entire problem.
If x^* is not unique, go to step 2.

Step 2
Solve the following mathematical program:

$$(P_{16})\begin{cases}\min f_2(x)\\ x \in X\\ f_1(x) = f_1(x^*).\end{cases}$$

This procedure is then repeated until a unique solution is obtained.
If the procedure stops at the minimum of the objective $f_i(x)$, $i < k$, then the objectives that are less important than $f_i(x)$ will be ignored.
The question that can be raised now is that of whether the solution obtained by this method is Pareto optimal. The answer to this question is given by the following theorem.

Theorem 2.6

If the lexicographic goal programming method is used to solve the multiobjective program (P_5), then the solution obtained is Pareto optimal.

Proof

Let $x^* \in X$ be a solution obtained by the lexicographic goal programming method. Assume that x^* is not Pareto optimal for (P_5). Therefore there exists a point $x \in X$ such that:

$$f_i(x) \leq f_i(x^*) \quad i = 1, \dots, k$$

and for one ℓ we have :

$$f_\ell(x) < f_\ell(x^*).$$

Let *i* = 1, then $f_1(x)$ attains its minimum at x^*. Since $f_1(x) \leq f_1(x^*)$, we should have

that:

$$f_1(x) = f_1(x^*).$$

If optimization is performed for *i= 2,* we have

$$f_2(x) = f_2(x^*)$$

for the same reason.
If optimization is performed for i= 1,…,k then

$$f_i(x) = f_i(x^*); i = 1, \dots, k.$$

This contradicts the assumption that there is at least one strict inequality.
Therefore, x^* is Pareto optimal for (P_5).

2.2.4 A posteriori methods

A posteriori methods are concerned with finding all or most of the Pareto optimal solutions for a given multiobjective program. These solutions are then presented to the decision maker who has to choose one of them. The most important a posteriori methods described in the literature include the e-constrained method [17], [35], the adaptive search method [24], the Benson's method [48] and the weighting method [5], [45]. For the sake of brevity we restrict ourselves to the weighting method.

Weighting method

Consider the program (P_5). The weighting method transforms the original multiobjective problem into a single objective one. This is done by creating a new objective from the weighted sum of the *k* objectives. By doing so, the resulting program is:

$$(P_{17})\begin{cases}\min F(x)\\ x \in X.\end{cases}$$

where

$$F(x) = \sum_{i=1}^{k} w_i\, f_i(x);$$

and w_i is the weight of the i^{th} objective;

i.e. $0 \leq w_i \leq 1,\ i = 1, \dots, k, \quad \sum_{i=1}^{k} w_i = 1.$

The following interesting result tells us that under mild conditions, a solution for (P_{17}) is Pareto optimal for (P_5).

Theorem 2.7

If, $w_i > 0$ for all i = 1,…, k, and if x^* is an optimal solution for (P_{17}) then x^* is a Pareto optimal solution for (P_5).

Proof.

Let $x^* \in X$ be a solution for the weighting problem (P_{17}) and suppose that x^* is not Pareto optimal for (P_5). Then, there exists a solution $x \in X$ such that:

$$f_i(x) \leq f_i(x^*);\ i = 1,\dots, k$$

and

$$f_\ell(x) < f_\ell(x^*) \text{ for some } \ell.$$

From the fact that $w_i > 0$; $i = 1,\dots,k$, it follows that

$$\sum_{i=1}^{k} w_i\, f_i(x) < \sum_{i-1}^{k} w_1 f_i(x^*).$$

This contradicts the assumption that x^* is a solution for the weighting problem $(\boldsymbol{P_{17}})$. Therefore, x^* *is* Pareto optimal for (P_5).

Example 2.3

Consider again problem (P_9) discussed in subsection 2.2.2:

$$(\mathrm{P}_{18})\begin{cases}\min\left[\left(\dfrac{x_1 + 2x_2}{x_1 + x_2 + 1}\right), \left(\dfrac{2x_1 + x_2}{2x_1 + 3x_2 + 1}\right)\right]\\ -x_1 + 2x_2 \leq 3\\ 2x_1 - x_2 \leq 3\\ x_1 + x_2 \geq 3\\ 2 \leq x_1 \leq 25\\ 1 \leq x_2 \leq 9{,}5\end{cases}$$

The counterpart of (P_{17}) for this multiobjective program is:

$$(\mathrm{P}_{19})\begin{cases} \min \sum_{i=1}^{2} w_i\, f_i(x) \\ -x_1 + 2x_2 \le 3 \\ 2x_1 - x_2 \le 3 \\ x_1 + x_2 \ge 3 \\ 2 \le x_1 \le 25 \\ 1 \le x_2 \le 9{,}5 \end{cases}$$

where

$$f_1(x) = \frac{x_1 + 2x_2}{x_1 + x_2 + 1}$$

$$f_2(x) = \frac{2x_1 + x_2}{2x_1 + 3x_2 + 1}$$

A set of solutions for (P_{19}) obtained by the parametric programming method is given in table 2.1.

Table 2.1: A set of solutions for (P_{19})

w_1	w_2	x_1	x_2	$F(x)$
0	1	2	1	0,62
0,1	0,9	2	1	0,66
0,2	0,8	3	3	0,70
0,3	0,7	3	3	0,78
0,4	0,6	3	3	0,83
0,5	0.5	3	3	0,92
0,6	0,4	3	3	0,99
0,7	0,3	3	3	1,07
0,8	0,2	3	3	1,14
0,9	0,1	3	3	1,21
1	0	3	3	1,29

Since the objectives are to be minimized, the most preferred alternative is given by $w_1 = 0, w_2 = 1$ and $x_1 = x_2 = 3,$ which yield the lowest value of $F(x) = 0{,}62$.
We obtain the same solution we obtained by the Compromise Programming method.
It is worth mentioning that for the weighting method, unlike in the goal programming method, the decision maker does not have to provide targets for each objective function.
Pareto optimality for (P_5) is guaranteed if the weights are strictly positive. Here both (2,1) and (3,3) are Paareto optimal for (P_{18}).

The weights are not easy for the decision makers to interpret or understand. It is also sometimes difficult for decision makers to choose a solution from a large number of generated alternatives.

2.2.5 Interactive methods

Interactive methods consist of the following steps:

Step 1
Find an initial solution.

Step 2
Discuss the solution with the decision maker. If the decision maker is satisfied, Stop. Otherwise go to the next step.

Step 3
Obtain a new solution and go back to step 2.

Many interactive methods have been developed in the literature. Here are some of them.:

-the Step method [23];
-the Sequential Proxy Optimization Technique (SPOT) [34];
-the Interactive Surrogate Worth Trade-off (ISWT) method [11];
-The Geoffrion-Dyer-Feinberg (GDF) method [27];
-the Reference point method [52].

Reference point method

Consider again the multiobjective program

$$(P_5)\begin{cases}\min\,[f_1(x),\dots,f_k(x)],\ k\geq 2\\ x\in X=\{x\in\mathbb{R}^n|g_j(x)\leq 0; j=1,\dots,m\}\end{cases}$$

where $f_i(x); i=1,\dots,k$ and $g_j(x); j=1,\dots,m$ are real-valued functions of $\mathbb{R}^n$.

Definition 2.12

An achievement function *is defined as follows*

$$x_z(z)=\max_{i=1,\dots,k}[w_i(z_i-z_i^*)]\quad i=1,\dots,k$$

where $z^*\in\mathbb{R}^k$ is an arbitrary reference point.
Consider now the following program:

$$(P_{20})\begin{cases}\min\, x_z(z)\\ z\in f(X)\end{cases}$$

where f(X)={f (x) / $x\in X$} and f(x)= $(f_1(x),\dots,f_k(x))$.

As (P_{20}) involves an achievement function, it is called an *achievement problem.*

The aim of the reference point method [47] is to minimize the related achievement function. The decision maker is requested to assist in the choice of the next reference point at each iteration.
The question that can be raised now is whether the solution obtained in this way is Pareto optimal for (P_5). An answer to this question is given by the following theorem.

Theorem 2.8

If the achievements function

$$x_z(z) = \max_{i=1,\dots,k} [w_i(z_i - z_i^*)]$$

is strictly increasing, then, if $\bar{z}$ is the solution to (P_{20}), we have that $f^{-1}(\bar{z})$ is Pareto optimal for (P_5).
The proof of this result may be found in [36].

The Reference point method consists of fixing a reference point z* defining x_z(z) and solving (P_{20}).
The reference point method is easy to understand and to implement and the decision maker is free to change his mind during the solution process.
Unlike the goal programming method, the reference point method guarantees the Pareto Optimality of the solutions of (P_5), depending on the increasing nature of the achievement function deployed.
Some shortcomings of the Reference point method include the fact that there are no criteria for the choice of aspiration levels and for elicitation of the achievement function. There is no clear strategy for producing the final solution since the reference point method does not help the decision maker to find improved solutions.

2.2.6 Metaheuristic for the multiobjective programming problem

Most of the methods described in the preceding sections apply for convex multiobjective programs. In the case of nonconvex multiobjective programs, metaheuristics may be considered [2], [26].
A metaheuristic is a method that seeks to find a good solution to an optimization problem at a reasonable computational cost. A metaheuristic often has an intuitive justification and therefore a mathematical proof cannot be constructed to guarantee the Pareto optimality of the solution obtained [2].
The most used metaheuristic are simulated annealing [41], tabu search [12] and genetic algorithm (GA) [51]. In what follows we restrict ourselves to the genetic algorithm method.

A genetic algorithm is a stochastic search method for problems based on the mechanisms of natural selection and genetics (that is, survival of the fittest).
One of the most important notions in genetic algorithms is that of chromosomes.
A chromosome is a string of numbers or symbols. A genetic algorithm starts with an initial set of randomly generated chromosomes which are called population. The population size is the number of individuals in that population. All chromosomes are evaluated by an evaluation function and the selection process is used to form a new population, which uses a sampling mechanism based on fitness values. The term "generation" is used to describe the cycle from one population to another. The

crossover and mutation operations are used to update all chromosomes and the new chromosomes are called offsprings. The new population is formed when the selection process selects new chromosomes. After a given number of generations, the best chromosome is decoded into a solution for (P_5).

A genetic algorithm usually follows the following steps.

Step 1
Initialize the chromosomes at random.

Step 2
Update the chromosomes by crossover and mutation operations.

Step 3
Calculate the objective values of all the chromosomes.

Step 4
Compute the fitness of each chromosome via the objective values.

Step 5
Select the best chromosome using the tournament selection method, ranking selection method or roulette wheel method.

Step 6
Repeat Step 2 to Step 5 for a given number of cycles.

Step 7
Report the best chromosome as the compromise solution for the multiobjective program.

Example 2.4

Consider the multiobjective program

$$(\mathrm{P}_{21})\begin{cases}\min\,[f_1(x), f_2(x), f_3(x)]\\ -x_1^2 + x_2^2 \geq 10\\ x_1^2 + x_2^2 + x_3^2 \leq 6\\ x_1, x_2, x_3 \geq 0\end{cases}$$

where

$$f_1(x) = 3 - \sqrt{x_1}$$
$$f_2(x) = 4 - \sqrt{x_1 + 2x_2}$$
$$f_3(x) = 5 - \sqrt{x_1 + x_2 x_2 3x_3}\ .$$

Using a genetic algorithm to solve this program leads to the following steps.

Step 1
Encode a solution $x = \{x_1, x_2, x_3\}$ into a chromosome $V = \{v_1, v_2, v_3\}$.

Step 2
Update the chromosomes by crossover and mutation operations.

Step 3
Compute the objective values of all chromosomes.

Step 4
Compute the fitness of each chromosome via the objective values.

Step 5
Select the best chromosome by the roulette wheel method.

Step 6 Repeat Steps 2 to 5 for 2000 generations.

The compromise solution for (P_{21}) obtained by this genetic algorithm is
$x = (x_1, x_2, x_3) = (9, 3.5, 2.597)$.

The advantage of genetic algorithm is that it can be used to solve nonconvex multiobjective programs. Unfortunately, it cannot guarantee the Pareto optimality of the solution obtained.
Another approach for solving convex or nonconvex Multiobjective Programming Problems is based on Fuzzy set Theory.
We discuss such an approach in the next section.

2.2.7 Solving a deterministic multiobjective program using fuzzy sets

2.2.7.1 *Putting a multiobjective program into a single objective*

Consider the following multiobjective program:

$$(\mathrm{P}_{22}) \begin{cases} max[f_1(x), \ldots, f_k(x)] & k \geq 2 \\ x \in X = \{x \in \mathbb{R}^n / g_j(x) \leq 0;\ j = 1, \ldots, m\} \end{cases}$$

Let:

$$L_i = \max_{x \in X} f_i(x); i = 1, \ldots, k$$

and assume that L_i is finite for all i.
Suppose further that the decision maker is able to fix $p_i < L_i$ so that he wishes to maintain $f_i(x)$ greater than p_i for all i.
Then (P_{22}) reads:

Find $x \in X$ such that $p_i \lesssim f_i(x) \lesssim L_i; i = 1, \ldots, k$. (2.10)

Here "~" means that inequalities are not strict imperatives but some leeways may be accepted.
Each flexible objective i of (2.10) may be represented by a fuzzy set U_i of X the membership function of which is defined as follows:

$$\mu_i(x) = \begin{cases} 0 & if \quad f_i(x) \le p_i \\ \frac{f_i(x)-p_i}{L_i-p_i} & if\ f_i(x) > p_i \end{cases} \tag{2.11}$$

Therefore using Bellman-Zadeh confluence principle [6], (P_{22}) can be written as follows:

$$(\mathrm{P}_{23}) \begin{cases} \max \ \bigwedge_{\mathrm{i}=1}^{\mathrm{k}} \mathrm{u_i(x)} \\ \mathrm{x} \in \mathrm{F} \end{cases}$$

where F denotes the set

$$X \cap \left(\cap_{\mathrm{i}=1}^{\mathrm{k}} \operatorname{supp} \mu_{\mathrm{i}}\right)$$

and

$$\bigwedge$$

is the operator used to translate the "and" connective.

2.2.7.2 Solving the resulting problem using the γ- operator

The γ-operator introduced by Zimmermann [57] and defined by

$$\bigwedge_{i=1}^{k} u_i(x) = \left(\prod_{i=1}^{k} u_i(x)\right)^{1-\gamma} \left[1 - \prod_{i=1}^{k}(1-u_i)\right]^{\gamma}$$

where $\gamma \in]0,1[$ yields a compensation grade between aggregated membership function. This compensatory operator seems to be more appropriate than Zadeh's min operator [55].
As a matter of fact it overcomes the ultra-pessimistic criticism addressed to the min operator.
With Zimmermann's operator, (P_{23}) reads:

$$(P_{24}) \begin{cases} \max \emptyset(x) \\ x \in F \end{cases}$$

where

$$\emptyset(x) = \left(\prod_{i=1}^{k} \frac{f_i(x)-P_i}{L_i-P_i}\right)^{1-\gamma} \left[1 - \prod_{i=1}^{k}(1 - \frac{f_i(x)-P_i}{L_i-P_i})\right]^{\gamma}. \tag{2.12}$$

The question that we raise now is that of the Pareto optimality of the solution obtained by solving (P_{24}) for (P_{22}).
The answer to this question is given in the following result.

Proposition 2.1

Suppose L_i and p_i are finite for all i and are such that $p_i < L_i$ (i=1,…,k).

If x^* is a solution of (P_{24}), then x^* is Pareto Optimal for (P_{22}).
Proof

Suppose x^* optimal for (P_{24}) and non efficient for (P_{22}). Then there exists $x^{**} \in X$ so that

$$f_i(x^{**}) \geq f_i(x^*), i = 1, \ldots, k \qquad (2.13.)$$

and

$$f_i(x^{**}) > f_i(x^*) \text{ for at least one } i \text{ .}$$

x^{**} cannot be in $X \backslash F$ otherwise there will be an i so that $f_i(x^*) > p_i \geq f_i(x^{**})$ i.e. $f_i(x^*) > f(x^{**})$ which contradicts (2.13).

So we have:

$$\frac{f_i(x^{**}) - p_i}{L_i - p_i} \geq \frac{f_i(x^*) - p_i}{L_i - p_i}, \quad i = 1, \ldots, k$$

$$\frac{f_i(x^{**}) - p_i}{L_i - p} > \frac{f_i(x^*) - p_i}{L_i - p_i} \text{ for at least one i} \in \{1, \ldots, k\}. \qquad (2.14)$$

As $L_i - p_i > 0$, $f_i(x^*) - p_i > 0 \ \ \forall\, i$ and $0 < \gamma < 1$,

we have:

$$\left(\prod_{i=1}^{k} \frac{f_i(x^{**}) - p_i}{L_i - p_i}\right)^{1-\gamma} > \left(\prod_{i=1}^{k} \frac{f_i(x^*) - p_i}{L_i - p_i})\right)^{1-\gamma}. \qquad (\S)$$

We also have:

$$1 - \frac{f_i(x^{**}) - p_i}{L_i - p_i} \leq 1 - \frac{f_i(x^*) - p_i}{L_i - p_i}, \qquad i = 1, \ldots. k$$

and

$$1 - \frac{f_i(x^{**}) - p_i}{L_i - p_i} < 1 - \frac{f_i(x^*) - p_i}{L_i - p_i}, \qquad i = 1, \ldots. k$$

for $i \in \{1, \ldots, k\}$ for which the inequality (2.14) holds.
It results that

$$\left[1 - \prod_{i=1}^{k}\left(1 - \frac{f_i(x^{**}) - p_i}{L_i - p_i}\right)\right]^{\gamma} > \left[1 - \prod_{i=1}^{k}\left(1 - \frac{f_i(x^*) - p_i}{L_i - p_i}\right)\right]^{\gamma}. \qquad (\S\S)$$

(§) and (§§) give $\emptyset(x^{**}) > \emptyset(x^*)$. This contradicts the fact that x^* is optimal for (P_{24}).

2.2.7.3 Solving the resulting problem using the min-bounded sum operator

Although the solution obtained by solving the resulting problem (P_{24})using the γ-operator is efficient for (P_{22}), it is not always easy to solve (P_{24}) by available mathematical programming software.
This leads us to use another compensatory operator, namely, the bounded min-sum defined as follows:

$$\gamma \min_{i=1,\dots,k} \mu_i + (1-\gamma)\min\left(1, \sum_{i=1}^{k} \mu_i\right)$$

where γ is a coefficient of compensation ranging on (0,1).
Empirical investigations [31] show that 0.705 is a good value for γ.
With this operator the program (P_{23}) reads:

$$(P_{25})\begin{cases} \max\,[\gamma \min_{i=1,\dots,k}\ \mu_i +\ (1-\gamma)\min\,(1, \sum_{i=1}^{k} \mu_i)] \\ \mathrm{x}\ \in F \end{cases}$$

Proposition 2.2

x^* is optimal for (P_{25}) if and only if (x^*, λ^*, μ^*) where $\lambda^* = min_i \mu_{i(}(x^*)$, $\mu^* = min\,(1, \sum_{i=1}^{k} \mu_i\,(x^*)$ is the solution of the problem

$$(P_{26})\begin{cases} max\ \gamma\lambda + (1-\gamma)\mu \\ \lambda \le \mu_i x \quad i = 1, \dots, m \\ \mu \le 1 \\ \mu \le \sum \mu_i(x) \\ x \in F \end{cases}$$

Lemma 2.1

If (x^*, λ^*, μ^*) is optimal for (P_{26}), then

$$\lambda^* = min_i \mu_{i(}(x^*), \qquad \mu^* = min\,(1, \textstyle\sum_{i=1}^{k} \mu_i\,(x^*)).$$

Proof.
Suppose (x^*, λ^*, μ^*) is optimal for (P_{26}) and $\lambda^* \ne min_i \mu_{i(}(x^*)$; let λ^{**}=$min_i \mu_{i(}(x^*)$, then $\lambda^{**} > \lambda^*$.
$(x^*, \lambda^{**}, \mu^*)$ verifies the constraints of (P_{26}) and $\gamma\lambda^*$+(1-γ)μ^* < $\gamma\lambda^{**}$+(1-γ)μ^*; this contradicts the fact that (x^*, λ^*, μ^*) is optimal for (P_{26}). The fact that $\mu^* = \min\,(1, \sum_{i=1}^{k} \mu_i\,(x^*))$ can be proved in the same way.

Proof of Proposition 2.2

(a) Necessity:
If x^* is optimal for (P_{25}), then

$$\gamma \min_{i=1,\dots,k} \mu_i\,(x^*) +\ (1-\gamma)\min\left(1, \sum_{i=1}^{k} \mu_i(x^*)\right)$$

$$\geq \gamma \min_{i=1,\dots,k} \mu_i(x) + (1-\gamma)\min\left(1, \textstyle\sum_{i=1}^{k} \mu_i(x)\right), \ \forall x \in F. \qquad (2.15)$$

Suppose that (x^*, λ^*, μ^*) with $\lambda^* = \min_i \mu_i(x^*)$ and $\mu^* = \min(1, \sum \mu_i(x^*))$ is not optimal for (P_{25}). There exists $(x^{**}, \lambda^{**}, \mu^{**}) \neq (x^*, \lambda^*, \mu^*)$ such that:

$$\begin{aligned}
&\lambda^{**} \leq \mu_i(x^{**}), i = 1, \dots, k, \\
&\mu^{**} \leq 1, \qquad \mu^{**} \leq \textstyle\sum_{i=1}^{k} \mu_i(x^{**}), \\
&x^{**} \in F, \\
&\gamma\lambda^{**} + (1-\gamma)\mu^{**} > \gamma\lambda^* + (1-\gamma)\mu^*.
\end{aligned}$$

It results that

$$\gamma \min_i \mu_i(x^{**}) + (1-\gamma) \min_i \left(1, \sum \mu_i(x^{**})\right) \geq \gamma\lambda^{**} + (1-\gamma)\mu^{**} > \gamma\lambda^* + (1-\gamma)\mu^*$$
$$= \gamma \min_i \mu_i(x^*) + (1-\gamma) \min\left(1, \sum \mu_i(x^*)\right)$$

which contradicts (2.15).

(b) Sufficiency:
If (x^*, λ^*, μ^*) is optimal for (P_{26}), by the lemma 2.1 we have

$$\lambda^* = \min_i \mu_i(x^*) \ and \ \ \mu^* = \min\left(1, \sum \mu_i(x^*)\right).$$

Suppose x^* is not optimal for (P_{25}), there exists x^{**} such that:

$$\gamma \ \min_i \mu_i(x^{**}) + (1-\gamma) \min_i(1, \textstyle\sum \mu_i(x^{**})) > \gamma \ \min_i \mu_i(x^*) + (1-\gamma) \min(1, \textstyle\sum \mu_i(x^*))$$
$$= \gamma\lambda^* + (1-\gamma)\mu^*.$$

Let

$$\lambda^{**} = \min_i \mu_i(x^{**}), \quad \mu^{**} = \min\left(1, \sum \mu_i(x^{**})\right).$$

$(x^{**}, \lambda^{**}, \mu^{**})$ verifies the constraints of (P_{26}) and

$$\gamma\lambda^{**} + (1-\gamma)\mu^{**} > \gamma\lambda^* + (1-\gamma)\mu^*.$$

This contradicts the fact that (x^*, λ^*, μ^*) is optimal for (P_{26}).

The solution obtained by (P_{25}) is not necessarily Pareto optimal.
The following result claims that it is an attractive one from the standpoint of efficiency.

Proposition 2.3

If L_i and p_i are finite with $p_i < L_I$ $(i = 1, \dots, k)$, and if x^* is optimal for (P_{25}), then each solution x^{**} which dominates strictly x^* is also optimal.

Proof
x^{**} dominates strictly x^* implies

$$f_i(x^{**}) \geq f_i(x^*) \qquad \forall i$$

and

$$f_i(x^{**}) > f_i(x^*) \text{ for at least one } i.$$

Then

$$\frac{f_i(x^{**})-p_i}{L_i-p_i} \geq \frac{f_i(x^*)-p_i}{L_i-p_i} \qquad \forall i$$

and

$$\min_i \left(\frac{f_i(x^{**})-p_i}{L_i-p_i}\right) \geq \min_i \left(\frac{f_i(x^*)-p_i}{L_i-p_i}\right).$$

We also have

$$\min\left(1, \sum \frac{f_i(x^{**})-p_i}{L_i-p_i}\right) \geq \min\left(1, \sum \frac{f_i(x^*)-p_i}{L_i-p_i}\right).$$

The last two inequalities imply that:

$$\gamma \min_i \left(\frac{f_i(x^{**})-p_i}{L_i-p_i}\right) + (1-\gamma)\min\left(1, \sum \frac{f_i(x^{**})-p_i}{L_i-p_i}\right)$$
$$\geq \gamma \min_i \left(\frac{f_i(x^*)-p_i}{L_i-p_i}\right) + (1-\gamma)\min\left(1, \sum \frac{f_i(x^*)-p_i}{L_i-p_i}\right).$$

As x^* is optimal for (P_{25}), the last relation is necessarily an equality and x^{**} is also optimal for (P_{25}).

Corollary 2.4

If the solution x^* of (P_{25}) is unique, then x^* is efficient for (P_{22}).

Proof

As x^* is unique, there is no $x^{**} \in F$ such that $f_i(x^{**}) \geq f_i(x^*)$ $\forall i$ and $f_i(x^{**}) > f_i(x^*)$ for at least one i. Otherwise, by the Proposition 2.3, x^{**} would be optimal. For each $x \in X \backslash F$ there is an ℓ so that $f_\ell(x) \leq p_\ell < f_\ell(x^*)$. We can conclude that there is no x** $\in X$ such that $f_i(x) \geq f_i(x^*)$ $\forall i$ and $f_i(x) > f_i(x^*)$ for at least one i. x^* is then efficient for (P_{22}).

2.2.7.4 Numerical example

Consider the multiobjective program:

$$(P_{27}) \begin{cases} \max\{(3x_1 + x_2), (x_1 - 2x_2)\} \\ 2x_1 - x_2 \leq 2 \\ x_1 + 2x_2 \leq 5 \\ x_1 \geq 0, x_2 \geq 0 \end{cases}$$

Put $f_1(x) = 3x_1 + x_2$ and $f_2(x) = x_1 - 2x_2$.

Let us compute L_1 and L_2

To obtain L_1 we have to solve the linear program:

$$\begin{cases} \max\ (3x_1 + x_2) \\ 2x_1 - x_2 \leq 2 \\ x_1 + 2x_2 \leq 5 \\ x_1 \geq 0, x_2 \geq 0 \end{cases}$$

Using LINGO we obtain the optimal solution: $x_1^* = 9/5$, $x_2^* = 8/5,$ and $L_1 = 7.$

To obtain L_2 we have to solve the linear program:

$$\begin{cases} \max\ (x_1 - 2x_2) \\ 2x_1 - x_2 \leq 2 \\ x_1 + 2x_2 \leq 5 \\ x_1 \geq 0, x_2 \geq 0 \end{cases}$$

Using again LINGO the optimal solution: $x_1^* = 1$, $x_2^* = 0$, and $L_2 = 1.$
Now taking decision maker opinion we fix p_1 and p_2 such that:

$$p_1 \leq L_{1,} \quad p_2 \leq L_2 \ .$$

Let p_1 = 5 and p_2= 0 and define μ_1 and μ_2 as in (2.11)

Aggregating the two membership functions using min-bounded sum operator yields:

μ_D(x) =$\gamma \min\big(\mu_1(x), \mu_2(x)\big)$ + (1-γ)$\min$ (1, $\mu_1(x) + \mu_2(x)$).

The resulting program is then

$$(\mathrm{P}_{28}) \begin{cases} \max\{\gamma \min\big(\mu_1(x), \mu_2(x)\big) + (1-\gamma) \min\big(1,\ \mu_1(x) + \mu_2(x)\big)\} \\ 2x_1 - x_2 \leq 2 \\ x_1 + 2x_2 \leq 5 \\ \mathrm{x}_1 \geq 0\ , \quad \mathrm{x}_2 \geq 0 \end{cases}$$

By virtue of Proposition 2.2, this program is equivalent to:

$$(\mathrm{P}_{29}) \begin{cases} \max\ (\gamma\lambda + (1-\lambda)\mu) \\ \lambda \leq \mu_i\,(x) \\ \mu \leq 1 \\ \mu \leq \mu_1(x) + \mu_2(x) \\ x \in F \end{cases}$$

Replacing γ by 0.705 and we obtain the program:

$$(P_{30})\begin{cases} \max\ (0.705\lambda + 0.295\mu) \\ \lambda \leq\ \ 3x_1 +\ x_2 \\ \lambda \leq\ \ x_1\ -\ 2x_2 \\ \mu \leq 1 \\ \mu \leq\ \frac{5}{2}x_1 - \frac{3}{2}\ x_2 - \frac{5}{2} \\ 2x_1 -\ x_2\ \leq 2 \\ x_1 +\ 2x_2\ \leq 5 \\ x_1\ \geq 0\ ,\ \ x_2 \geq\ 0 \end{cases}$$

Using LINGO software to solve this program we obtain the following solution:

$$\lambda^* = 1$$
$$\mu^* = 1$$
$$x_1^* = 1$$
$$x_2^* = 0.$$

CHAPTER 3: MULTIOBJECTIVE PROGRAMMING PROBLEMS UNDER FUZZINESS.

3.1 Preamble

The increased complexity and uncertainty of our social, economic and business environment are currently raising some of the greatest challenge yet faced by managers.
Old answers may no longer be appropriate. Instead, managers are required to understand, to cope and to adapt to complex and imprecise situations.
To do so, better decision making procedures must be developed to increase the ability of managers to make decisions in perspective of conflictual goals and uncertainty.
In this chapter we discuss ways for incorporating fuzziness in a multiobjective program model.

3.2 Approach for solving a fuzzy linear multiobjective programming problem using possibility measure

In this section we restrict our discussion to linear programming in a way to convey our ideas in a simpler manner.

3.2.1 Problem statement

Consider the mathematical program

$$(P_{31})\begin{cases} \max\,(\tilde{c}^1 x, \dots, \tilde{c}^k x) \\ \tilde{A}x \leq \tilde{b} \\ x \geq 0 \end{cases}$$

where $\tilde{c}^j$ *(j=1,….,k)* are n-vectors ,$\tilde{b}$ is an m-vector and $\tilde{A}$ an *m* x *n* matrix , all having components that are fuzzy numbers.
$\tilde{c}_l^j$*(l=1, ….,n)* $\tilde{b}_i$, $\tilde{a}_{ij}$ are fuzzy numbers, the membership functions of which are,$\pi_{\tilde{c}_i^j}$, $\pi_{\tilde{b}_i}, \pi_{\tilde{a}_{ij}}$ respectively.

The membership function of $\tilde{c}^j$ is defined as follows:

$$\pi_{\tilde{c}^j}(t_1, \dots, t_n) = \min\,(\pi_{\tilde{c}_1^j}(t_1), \dots, \pi_{\tilde{c}_n^j}(t_n)).$$

By the multiplicity of objectives and the imprecision of technological and objective coefficients, (P_{31}) is an ill posed problem.
We must then specify the solution concept we consider to be appropriate for this program. This is the task to which we now turn.

3.2.2 α - possibly feasible, β –possibly efficient and satisfying solutions for (P_{31})

Definition 3.1

Consider $\alpha = (\alpha_1, \ldots, \alpha_m)$, $\alpha_i \in [0.1]$ $\forall i$ (eventually $\alpha_1 = \alpha_2 = \ldots = \alpha_m$).
$x \in X = \{x \in R^n / x \geq 0\}$ is said to be α -possibly feasible for (P_{31}) if

$$Poss\ (\tilde{A}_i\ x \leq \tilde{b}_i) \geq \alpha_i \quad i = 1, \ldots, m, \tag{3.1}$$

where Poss denotes Possibility [55].

Some points relating to the above definition are in need of comments.
First using the Zadeh's extension principle [16] we have:

$$\text{Poss}\ (\tilde{A}_i\ x \leq \tilde{b}_i) = \sup_{\substack{t_i, s_i \\ \sum t_{ij} x_j \leq s_i}} \min\ (\pi_{\tilde{a}_{i1}}(t_{i1}), \ldots, \pi_{\tilde{a}_{in}}(t_{in})). \pi_{\tilde{b}_i}(s_i))$$

where $t_i = (t_{i1}, \ldots, t_{in})$, i=1,…,m.

This quantity is the possibilistic valuation of the alternative $x \in R^n$, i.e. the degree of possibility to which x satisfies the constraint $\bar{A}_i x \leq \bar{b}_i$.
Poss $(\tilde{A}_i\ x \leq \tilde{b}_i)$ may be obtained by solving the following mathematical programming problem:

$$\begin{gathered} \text{Sup } k, \\ k \leq \pi_{\tilde{a}_{i1}}(t_{i1}), \\ \vdots \\ k \leq \pi_{\tilde{a}_{in}}(t_{in}) \\ \sum_{j=1}^{n} t_{ij} x_j \leq s_i \\ t_i = (t_{i1}, \ldots, t_{in}) \in \mathbb{R}^n \\ x_j \geq 0, \ j = 1, \ldots, n, \end{gathered}$$

Second, the relative importance of different constraints may be taken into account by manipulating appropriately the target levels α_i.

Consider now the following mathematical program:

$$(P_{32}) \begin{cases} \max\ (\tilde{c}^1 x, \ldots, \tilde{c}^k x) \\ x \in D \end{cases}$$

where $D \subset X = \{x \in \mathbb{R}^n\ /\ x \geq 0\}$.

Definition 3.2

$x^0 \in D$ is β-possibly efficient for (P_{32}), if there is no x∈ D and i∈ (1,…,k) such that:
Poss ($\tilde{c}^1 x \geq \tilde{c}^1 x^0$,…,$\tilde{c}^{l-1} x \geq \tilde{c}^{l-1} x^0$, $\tilde{c}^l x > \tilde{c}^l x^0$, $\tilde{c}^{l+1} x \geq \tilde{c}^{l+1} x^0$,…, $\tilde{c}^k x \geq \tilde{c}^k x^0$)≥ β.

By the extension principle,

Poss ($\tilde{c}^1 x \geq \tilde{c}^1 x^0$,..,$\tilde{c}^{l-1} x \geq \tilde{c}^{l-1} x^0$, $\tilde{c}^l x > \tilde{c}^l x^0$, $\tilde{c}^{l+1} x \geq \tilde{c}^{l+1} x^0$,…, $\tilde{c}^k x \geq \tilde{c}^k x^0$)
= sup min($\pi_{\tilde{C}^1}(t_1)$,…, $\pi_{\tilde{C}^{l-1}}(t_{l-1})$,$\pi_{\tilde{C}^l}$ (t_l), $\pi_{\tilde{C}^{1+1}}$ (t_{l+1}),…, $\pi_{\tilde{C}^k}$ (t_k))
$(t_1,\ldots,t_k) \in T_i$

where

$$T_i = \{(t_1, \ldots, t_k) \in R^{kn} \;/\; t_1 x \geq t_1 x^0, .., t_{l-1} x \geq t_{l-1} x^0,\ t_l x > t_l x^0, t_{l+1} x \geq t_{l+1} x^0, \ldots, t_k x \geq t_k x^0\}$$

and $\pi_{\tilde{C}^i}$ (*i*= 1,.., *k*) are *n*- ary possibility distributions.

Definition 3.2 can be weakened in the following way.

Definition 3.3

x^0∈ D is β-weakly possibly efficient for (P_{32}) if there is no x∈ D such that:

Poss ($\tilde{c}^1 x > \tilde{c}^1 x^0$,..., $\tilde{c}^k x > \tilde{c}^k x^0$)≥ β

It is an easy matter to show that a β-possibly efficient solution for (P_{32}) is *β*-weakly possibly efficient for (P_{32}) but the reverse is not true.

Definition 3.4

x^0 ∈ X is an (α ,β)-satisfying solution for (P_{31}) if and only if x^0 is β-possibly efficient for the program.

$$(P_{33}) \begin{cases} max\ (\tilde{c}^1 x, \ldots, \tilde{c}^k x^0), \\ x \in X^\alpha \end{cases}$$

where X^{α} denotes the set of α -possibly feasible solutions for (P_{31}).

It is worth noticing that for α and *β* sufficiently close to 1, an (α, *β*)-satisfying solution for (P_{33}) is attractive with respect to feasibility and efficiency. As a matter of fact, such a solution achieves a great possible degree of feasibility and is not possibly dominated to a great extent.
Throughout this dissertation we suppose α, β fixed a priori. One can equally well discuss how to determine them optimally (in the Pareto optimal sense for instance). Some attempts in this direction may be found elsewhere [20]. A full parametric sensitivity analysis may also be envisaged in this context.
A natural question is that of how to single out an (α , *β*)-satisfying solution for (P_{31}).

We now move to a more precise discussion of this matter. For all to come it is assumed that all possibility distributions involved in (P_{31}) are convex.

3.2.3. Characterization of an (α, β) –satisfying solution for $(\boldsymbol{P_{31}})$

Consider the mathematical program

$$(P_{34}) \begin{cases} \max\ ((\tilde{c}^1)_\beta x, \dots, (\tilde{c}^k)_\beta x\ ), \\ x \in X^\alpha \end{cases}$$

where

$(\tilde{c}^1)_\beta = ((\tilde{c}_1^i)_\beta \dots, (\tilde{c}_n^i)_\beta\ )$ and $(c_j^i)_\beta$ denotes the β-level set of the possibilistic variable $\tilde{c}_j^i$.

By the convexity assumption on the distribution of $\tilde{c}_j^i$, $(\tilde{c}_j^i)_\beta$, *j=1, ..., n; i=1, .., k*, are real intervals that will be denoted as $[\tilde{c}_j^{iL}, \tilde{c}_j^{iU}]$.

Let now $\emptyset_\beta$ be the set of k x n matrices *C=(cij)* with $c_{ij} \in [\ \tilde{c}_j^{iL}, \tilde{c}_j^{iU}]$.

It is clear that (P_{34}) may be written

$$\max \{\ Cx/x \in X^\alpha,\ C \in \emptyset_\beta\}$$

(P_{34}) is then an infinite family of multiple objective linear programs.

Definition 3.5

$x°$ is efficient for (P_{34}) if and only if there is no $C \in \emptyset_\beta$ and $x \in X^\alpha$, such that $C\,x \geq C\,x°$ with at least one strict inequality.

To put it differently, $x°$ is efficient for (P_{34}) if and only if $x°$ is efficient for

$$\begin{cases} \max\ \mathrm{C}x, \forall\ C \in \emptyset_\beta \\ x \in X^\alpha, \end{cases}$$

From now, EF($\emptyset_\beta$) and EF(C) will denote efficient solutions for (P_{34}) and for max $\{Cx/x \in X^\alpha\}$ respectively.

By Definition 3.4, EF ($\emptyset_\beta$) = $\bigcap_{C \in \emptyset_\beta} \mathrm{EF}\,(C)$.

A slight weakening of Definition 3.5 gives:

Definition 3.6

$x°$ is weakly efficient for (P_{34}) if and only if there is no $C \in \emptyset_\beta$ and $x \in X^\alpha$, such that $C\,x > C\,x°$.

It is a remarkable fact that (α,β)-satisfying solutions for (P_{31}) are nothing but efficient solutions for (P_{34}).

This is the content of the following result.

Theorem 3.1

$x°$ is an (α,β)-satisfying solution for (P_{31}) if and only if $x°$ is efficient for (P_{34}).

Proof

Suppose $x°$ is an (α,β)-satisfying solution for (P_{31}) then by Definition 3.4, $x°$ is α-possibly feasible and β-possibly efficient for (P_{33}). Assume now that $x°$ is not efficient for (P_{34}). There is then $x^1 \in X^\alpha$ and $(q^1,\ldots, q^k)$ with $q^i \in (\tilde{c}^i)_\beta$ such that :

$q^i x^1 \geq q^i x^0 \quad \forall i \in \{1,\ldots, k\}$ and $i_0 \in \{1,\ldots,k\}$ such that:

$$q^{i_0} x^1 \geq q^{i_0} x^0 \ . \tag{3.13}$$

As $q^i \in (\tilde{C}^i)_\beta$, $i = 1\ldots, k$ we have also:

$$\min (\pi_{\tilde{C}^i} (q^1),.., \pi_{\tilde{C}^k} (q^k)) \geq \beta \tag{3.14}$$

By (3.13) and (3.14) we have:

$\sup\limits_{(t_1,\ldots,t_k) \in T^1_{i_0}} \min ((\pi_{\tilde{C}^1}(t_1),\ldots. \pi_{\tilde{c}^{i_0}}(t_{i_0-1}),\ \pi_{\tilde{c}^{i_0}}(t_{i_0}),\ \pi_{\tilde{c}^{i_0+1}}(t_{i_0+1}),\ \ldots,\pi_{\tilde{C}^k}(t_k))$

$= \text{Poss} (\tilde{c}^1 x^1 \geq \tilde{c}^1 x^0, \ldots,\tilde{c}^{i_0-1} x^1 \geq \tilde{c}^{i_0-1} x^0 ,\ \tilde{c}^{i_0} x^1 > \tilde{c}^{i_0} x^0,$
$\tilde{c}^{i_0+1} x^1 \geq \tilde{c}^{i_0-1} x^0 , \ldots, \tilde{c}^k x^1 \geq \tilde{c}^k x^0) \geq \beta$

where

$T^1_{io} = \{(t_1,\ldots,t_k) \in R^{nk} / t_1 x^1 \geq t_1 x^0, \ldots, t_{i_{0-1}} x^1 \geq t_{i_{0-1}} x^0,$

$t_{i_0} x^1 > t_{i_0} x^0, t_{i_{0+1}} x^1 \geq t_{i_{0+1}} x^0, \ldots, t_k x^1 \geq t_k x^0\}.$

This contradicts the β-possible efficiency of x^o for (P_{33}) and the if part of the theorem is established.
To show the only if part, suppose x^o is efficient for (P_{34}) and not (α,β)-satisfying for (P_{31}). Then there is $x^2 \in X^\alpha$ and $s \in \{1,\ldots,k\}$ such that:

$\text{Poss}(\tilde{c}^1 x^2 \geq \tilde{c}^1 x^o, \ldots, \tilde{c}^{s-1} x^2 \geq \tilde{c}^{s-1} x^o, \tilde{c}^s x^2 > \tilde{c}^s x^o, \tilde{c}^{s+1} x^2 \geq \tilde{c}^{s+1} x^o, \ldots,$
$\tilde{c}^k x^2 \geq \tilde{c}^k x^0) \geq \beta,$
i.e.

$$\sup_{(t_1,\ldots,t_k) \in T^2_s} \min (\pi_{\tilde{c}^1}(t_1),..,\pi_{\tilde{c}^{s-1}}) (t_{s-1}), \pi_{\tilde{c}^s}(t_s), \pi_{\tilde{c}^{s+1}}(t_{s+1}),\ldots, \pi_{\tilde{c}^k}(t_k)) \geq \beta \tag{3.15}$$

where

$T^2_s = \{(t_1,\ldots,t_k) \in R^{nk} / t_1 x^2 \geq t^1 x^0, \ldots, t_{s-1} x^2 \geq t_{s-1} x^0,$
$t_s x^2 > t_s x^0, t_{s+1} x^2 \geq t_{s+1} x^0, \ldots, t_k x^2 \geq t_k x^0\}.$

For this supremum to exist there should be a vector $(p_1,\ldots,p_k) \in R^n \mathrm{x} \ldots\ldots \mathrm{x}\, \mathrm{R}^n$ satisfying the following constraints:

$$p_1x^2 \geq p_1x^0,\ldots,p_{s-1}x^2 \geq p_{s-1}x^0, p_sx^2 > p_sx^0, p_{s+1}x^2 \geq p_{s+1}x^0, \ldots, p_kx^2 \geq p_kx^0 \quad (3.16)$$

Suppose now that for all $(p_1,\ldots,p_k)$ satisfying the system (3.16), we have
min $(\pi_{\tilde{c}^i}(p_1),\ldots,\pi_{\tilde{c}^k}(p_k))$<$\beta$.

Then
sup min $(\pi_{\tilde{c}^i}(p_1),..,\pi_{\tilde{c}^{s-1}})\ (p_{s-1}),\ \pi_{\tilde{c}^s}(p_s),\ \pi_{\tilde{c}^{s+1}}(p_{s+1}),\ldots..,\ \pi_{\tilde{c}^k}(p_k))$<$\beta$.
$(p^1,\ldots\ldots,p^k) \in T_s^2$
Contradicting (3.15).

There is then $(p_1,\ldots,p_k)$ satisfying (3.16) such that:

$$\min(\pi_{\tilde{c}^i}(p_1),\ldots.,\pi_{\tilde{c}^k}(p_k)) \geq \beta. \quad (3.17)$$

By (3.17), $\pi_{\tilde{c}^i}(p_i) \geq \beta$, $i = 1, \ldots., k$, i.e.

$$p_i \in (\tilde{c}^i)_{\beta}, \quad i=1,\ldots.,k. \quad (3.18)$$

(3.16) and (3.18) contradict the efficiency of x^0 for (P_{34}) and we are done.

Corollary 3.1

x^0 is β-weakly possibly efficient for (P_{33}) if and only if x^0 is weakly efficient for (P_{34}).

The usefulness and interest of these results will be enhanced if there is a way of finding an efficient or weakly efficient solution of the program (P_{34}). We now turn to this problem in some details.

3.2.4 Finding an efficient solution for (P_{34})

The following notations will facilitate further discussions.
M_β denotes the subset of $\emptyset_\beta$ composed of matrices C having elements of each column at the upper bound or at the lower bound, i.e. if $C \in M_\beta$ then either

$$C_{.j} = C_{.j}^L = \begin{pmatrix} \tilde{c}_j^{1L} \\ \vdots \\ \tilde{c}_j^{nL} \end{pmatrix} \text{ or } C_{.j} = C_{.j}^U = \begin{pmatrix} \tilde{c}_j^{1U} \\ \vdots \\ \tilde{c}_j^{nU} \end{pmatrix}$$

where $\tilde{c}_j^{iL}$ and $\tilde{c}_j^{iU}$ are the right and the left endpoint of $(\tilde{c}_j^i)_\beta$ respectively.
Let $K(C) = \{p \in \mathbb{R}^n \ /\ Cp \geq 0\}$.
If $S \subset \emptyset_\beta$ then $K(S) = \bigcap_{C\in S} K(C)$, and EF (S) = $\bigcap_{C\in S} EF(C)$, where, as previously, EF(C) is the set of efficient solutions for the optimization problem max (C x | $\in X^\alpha$).

Lemma 3.1

x is efficient for (P_{34}) if and only if

$$[\{x\} + K(\emptyset_\beta)] \cap X^\alpha = \{x\}. \tag{3.19}$$

Proof

Necessity: Suppose x efficient for (P_{37}) and $[\{x\} + K(\emptyset_\beta)] \cap X^\alpha \neq \{x\}$.
Then there is $y \in [\{x\} + K(\emptyset_\beta)] \cap X^\alpha$ with $y \neq x$, i.e. there is $C \in \emptyset_\beta$ such that $y = x + Cp \in X^\alpha$ with $p \in K(C)$ and $y \neq x$. Or to put it differently, there is $C \in \emptyset_\beta$ and $y \in X^\alpha$ such that $Cy \geq Cx$. As $y \neq x$, there is some row C^i of C satisfying the condition $C^i y > C^i x$,. This contradicts the efficiency of x for (P_{34}).

Sufficiency: Assume (3.19) holds and x is not efficient for (P_{34}). Then there is $y \in X^\alpha$ and $C \in \emptyset_\beta$ such that $Cy \geq Cx$ with at least one strict inequality. Let now $p = y - x$; it is clear that $p \in K(C)$.
Furthermore $y = x + p$ and $y \in [\{x\} + K(C)] \cap X^\alpha$. This is in contradiction with (3.19).

Proposition 3. 2

(a) $K(\emptyset_\beta) = \cup_{C \in M_\beta} K(C)$.
(b) $EF(\emptyset_\beta) = \cap_{C \in M_\beta} EF(C)$.

Proof

(a) Assume $p \in K(\emptyset_\beta)$, there exists $C \in \emptyset_\beta$ such that $Cp \geq 0$. Consider $C^1 = (c^1_{ij})$ defined as follows:

$$C^1_{ij} = \begin{cases} C^L_{.j} & if\ p_j \geq 0 \\ C^M_{.j} & if\ p_j < 0 \end{cases}.$$

It is clear that $C^1 p \geq C p \geq 0$ i.e. $p \in K(C^1)$. As $C^1 \in M_\beta$ it follows that $K(\emptyset_\beta) \subseteq K(M_\beta)$.
The reverse inclusion is a direct consequence of the definitions of $K(\emptyset_\beta)$ and $K(M_\beta)$.

(b) By definition of M_β we have $EF\ \emptyset_\beta) \subseteq EF(M_\beta)$.
Assume now that $x \notin EF\emptyset_\beta)$; then by Lemma 3.1 , $[\{x\} + K(\emptyset_\beta)] \cap X^\alpha \neq \{x\}$; it follows by (a) that $[\{x\} + K(M_\beta)]\ ] \cap X^\alpha \neq \{x\}$ and therefore $x \notin EF(C)$ for some $C \in M_\beta$. Hence $x \notin EF(M_\beta)$ and consequently $EF(M_\beta) \subseteq EF(\emptyset_\beta)$ as desired.

This result is insightful. It tells us that efficient solutions for (P_{34}) must be searched among elements of EF (M_β).
The following Lemma that will be used in the sequel is well known in multiple objective optimization theory [35].

Lemma 3.2

A necessary and sufficient condition for x^0 to be efficient for the multiple objective program

$$\max \; Cx, \quad x \in X.$$

is that there is λ>0 such that x^0 solves the mathematical program.

$$\max \lambda Cx, \quad x \in X.$$

A fascinating point is that the following program yields an (α ,β)-satisfying solution for (P_{31}):

$$(P_{35}) \begin{cases} \max \; q^0 x, \\ x \in X^{\alpha} \end{cases}$$

where q^0 is a solution of the system

$$V^i C^i - q = 0 \quad \forall\, i \text{ such that } C^i \in M_\beta, \qquad (3.20)$$
$$V^i \in \mathbb{R}^k, \quad V^i > 0.$$

Proposition 3.3

If x^0 is optimal for (P_{35}) then x^0 is efficient for (P_{34}).

Proof

As q^0 is a solution of (3.20), $\forall\, i$ such that $C^i \in M_\beta$, there is $V^i \in R^k$, $V^i > 0$ such that $V^i C^i = q^0$ i.e. $\forall\, i$ such that $C^i \in M_\beta$, x^0 solves max ($V^i C^i x | x \in X^\alpha$).
By Lemma 3.2, x^0 is efficient for

$$\max\, (C^i x | x \in X^\alpha) \;\forall\, C^i \in M_\beta$$

By Proposition 3.2, x^0 is efficient for

$$\max\, (C^i x | x \in X^\alpha) \;\forall C^i \in \emptyset_\beta$$

i.e. x^0 is efficient for (P_{34}) as desired.

Corollary 3.2

If x^0 is optimal for (P_{35}) then x^0 is (α β)- satisfying for (P_{31}).

This statement follows trivially from Proposition 3.3 and Theorem 3.1.

From this discussion we can derive the following method for solving.

3.2.5 Description of the method

Step 0
Start

Step 1
Read $\pi_{\tilde{c}_j^l}$ j=1,…,n , l=1,…,k, $\pi_{\tilde{a}_{ij}}$, $\pi_{\tilde{b}_i}$; i=1,…,m

Step 2
Fix α, β ;

Step 3
Put $\left(\tilde{c}_j^i\right)_\beta$ in the form $\left[\tilde{c}_j^{iL}, \tilde{c}_j^{iM}\right]$;

Step 4
Determine $\phi_\beta, M_\beta, X^\alpha$;

Step 5
Solve the system $V^i C^i - q = 0 \ \forall i$ such that:
$C^i \in M_\beta,\ V^i \in R^k, V^i > 0$,
Let q^0 be its solution;

Step 6
Solve the mathematical program:

$$\max\ q^0 x \quad x \in X^\alpha$$

Let x^0 be its solution

Step 7
Print x^0 *is* (α, β) satisfying for (P_{31});

Step 8
Stop

3.2.6 Numerical example

Consider the multiple objective linear possibilistic program:

$$(P_{36})\begin{cases} \text{Max}(\tilde{c}^1 x,\ \tilde{c}^2 x) \\ x_1 + x_2 \geq 250 \\ x_1 \leq 200 \\ 2x_2 \leq 200 \\ 2x_1 + 1.5x_2 \leq 480 \\ 3x_1 + 4x_2 \leq 900 \\ x_1 \geq 0,\ x_2 \geq 0 \end{cases}$$

where $\tilde{c}^1 = (\tilde{c}_{11}, \tilde{c}_{12})$, $\tilde{c}^2 = (\tilde{c}_{21}, \tilde{c}_{22})$ and $\tilde{c}_{ij}$ are characterized by the possibility distributions shown in figure 3.1 a-d.
Let $\beta = 1$; then $(\tilde{c}_{11})_\beta = [1,2]$, $(\tilde{c}_{12})_\beta = \{1\} = (\tilde{c}_{22})_\beta$, $(\tilde{c}_{21})_\beta = \{0\}$. The subset M_β of $\emptyset_\beta$ is composed of the matrices.

$$\begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}, \quad \begin{pmatrix} 2 & 1 \\ 0 & 1 \end{pmatrix}$$

For $q^0 = \binom{4.4}{2.9}$ the system

$$\begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} V_1^1 \\ V_2^1 \end{pmatrix} = \binom{4.4}{2.9}$$

as well as

$$\begin{pmatrix} 2 & 1 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} V_1^2 \\ V_2^2 \end{pmatrix} = \binom{4.4}{2.9}$$

have positive solutions, namely

$$\begin{pmatrix} V_1^1 \\ V_2^1 \end{pmatrix} = \binom{1.4}{2.9}, \quad \begin{pmatrix} V_1^2 \\ V_2^2 \end{pmatrix} = \binom{0.75}{5.8}$$

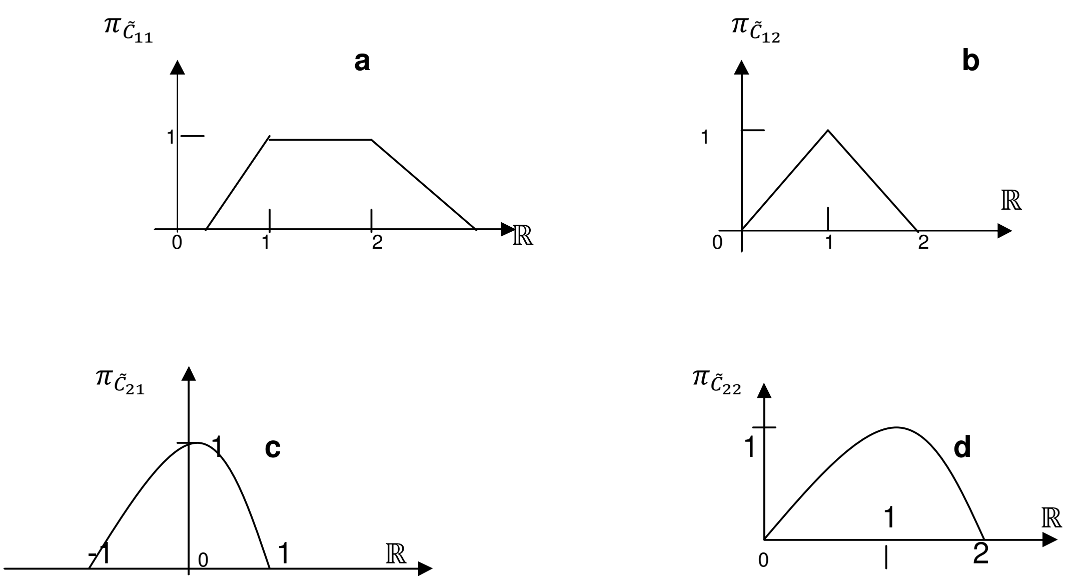


Fig.3.1: Possibility distributions of data of (P_{36}) .

respectively. As constraints are crisp, the set X of points of R^2 satisfying (P_{36}) is nothing but the set of 1-possibly feasible solutions. By the Corollary 3.2, the program max $(q^0 x \,/\, x \in X^1 = X)$ yields a (1,1)-satisfying solution for (P_{36}).
Solving this program, using LINGO software, we get the solution $x^* = (150,100)$.

3.3 Approach for solving a fuzzy multiobjective programming problem using connections between fuzzy numbers and real intervals.

3.3.1 Connection between fuzzy numbers and real intervals

Consider

$$\Pi: \mathcal{F}_{cc}(\mathbb{R}) \longrightarrow \tilde{\mathcal{C}}[0,1] \text{ x } \tilde{\mathcal{C}}[0,1]$$
$$\tilde{a} \longrightarrow (\tilde{a}^L(\alpha), \tilde{a}^U(\alpha))$$

where $\tilde{a}^L(\alpha) = \tilde{a}^L_\alpha$ and $\tilde{a}^U(\alpha) = \tilde{a}^U_\alpha$.

$\tilde{a}^L_\alpha$ and $\tilde{a}^U_\alpha$ standing for the lower and upper endpoints of the α-level of $\tilde{a}$.
Here $\mathcal{F}_{cc}(\mathbb{R})$ denotes the space of fuzzy numbers with compact support and $\tilde{\mathcal{C}}[0,1]$ is the set of real-valued bounded functions f on $[0,1]$ such that:

- f is left continuous for any $t \in [0,1]$ and right continuous at 0.
- f has a right limit for any $t \in [0,1]$.

It is shown in [54] that Π is isomorphic and isometric.
This result, called Embedding Theorem for fuzzy numbers, will be used in an essential manner in our approach for solving a multiobjective programming problem with fuzzy coefficients.

Consider the mappings: $\tilde{f}_i: \mathbb{R}^n \longrightarrow \mathbb{F}_{cc}(\mathbb{R}); i = 1, \dots, k$
We are interested in solving the following optimization problem:

$$(\mathrm{P}_{37}) \begin{cases} \max \ \{\tilde{f}_1(x), \dots, \tilde{f}_k(x)\} \\ x \in X \end{cases}$$

where $X = \{x \in \mathbb{R}^n / g_j(x) \leq 0; \ j = 1, \dots, m\}$ is a convex and bounded subset of $\mathbb{R}^n$.
We use the notation $\tilde{f}_{i\alpha}(x)$ for the α-level of $\tilde{f}_i(x)$. It is clear that

$$\tilde{f}_{i\alpha}(x) = [\tilde{f}^L_{i\alpha}(x), \tilde{f}^U_{i\alpha}(x)].$$

Consider now the following surrogate of (P_{37}):

$$(\mathrm{P}_{38}) \begin{cases} \min\{\prod(\tilde{f}_1(x)), \dots, \prod(\tilde{f}_k(x))\} \\ x \in X \end{cases}$$

The following result relates (P_{37}) and (P_{38}) in expected manner.

Proposition 3.4

x^* is efficient for (P_{37}) if and only if x^* is efficient for (P_{38})

Proof
Assume x^* is efficient for (P_{37}) and not efficient for (P_{38}).
As x^* is efficient for (P_{37}), there is no $x \in X$ such that:

$$\tilde{f}_i(x) \underset{F_{cc}(\mathbb{R})}{\leq} \tilde{f}_i(x^*) \quad \forall i \in \{1, \dots, k\} \tag{3.21}$$

and

$$\tilde{f}_\ell(x) \underset{F_{cc}(\mathbb{R})}{<} \tilde{f}_\ell(x^*) \quad \text{for some } \ell \in \{1, \dots, \text{k}\}. \tag{3.22}$$

As x^* is not efficient for (P_{38}), we may find $x \in X$ such that:

$$\Pi\tilde{f}_i(x) \leq \Pi\tilde{f}_i(x^*), \ \forall\, i \in \{1, \dots, k\} \tag{3.23}$$

and

$$\Pi\tilde{f}_\ell(x) < \Pi\tilde{f}_\ell(x^*) \quad \text{for some } \ell \in \{1,\dots,\text{k}\}. \tag{3.24}$$

This means

$$(\tilde{f}_{i\alpha}^L(x), \tilde{f}_{i\alpha}^U(x)) \leq \left(\tilde{f}_{i\alpha}^L(x^*),\ \tilde{f}_{i\alpha}^U(x^*)\right), \forall \alpha \in (0,1],\ \ \forall i\ \in \{1, \dots, k\} \tag{3.25}$$

and

$$(\tilde{f}_{\ell\alpha}^L(x), \tilde{f}_{\ell\alpha}^U(x)) < \left(\tilde{f}_{\ell\alpha}^L(x^*), \tilde{f}_{\ell\alpha}^L(x^*)\right), \forall \alpha \in (0,1] \text{ for some } \ell \in \{1,\dots,\text{k}\}. \tag{3.26}$$

(3.25) and (3.26) are equivalent to:

$$\tilde{f}_i(x) \underset{F_{cc}(\mathbb{R})}{\leq} \tilde{f}_i(x^*) \quad \forall\, i \in \{1, \dots, k\} \tag{3.27}$$

and

$$\tilde{f}_\ell(x) \underset{F_{cc}(\mathbb{R})}{<} \tilde{f}_\ell(x^*) \quad \text{for some } \ell \in \{1, \dots, \text{k}\}. \tag{3.28}$$

(3.27) and (3.28) are in contradiction with (3.21) and (3.22). This means that x^* is efficient for (P_{40}).
The other implication can be proved in similar way.

Making use of the definition of Π; (P_{38}) can be put in the following form:

$$(\mathrm{P}_{39}) \begin{cases} \min\{\left[\tilde{f}_i^L(x)(\alpha), \tilde{f}_i^U(x)(\alpha)\right] \quad i = 1, \dots, k\ \} \\ x \in X \\ \alpha \in [0,1] \end{cases}$$

Worthy to note here is the fact that (P_{39}) is a multiobjective mathematical program with infinitely many objective functions.
This is the price to pay for considering an equivalent deterministic counterpart of (P_{37}) instead of an approximate one.
To be able to carry out a fairly discussion of (P_{39}) we find it convenient to assume that minimizing an interval is tantamount to minimizing its midpoint.

This interpretation generalizes quite canonically the real case. As a matter of fact the midpoint of [a, a] is a.
Bearing in mind the above assumption and considering the fact that multiplying objective functions by a constant do not alter the localization of the optimum, (P_{39}) can be written:

$$(\mathrm{P}_{40})\begin{cases} \min \{[f_i^L(x)(\alpha)+f_i^U(x)(\alpha)];\, i=1,\dots,k\} \\ x \in X \\ \alpha \in [0,1] \end{cases}$$

From now $f_i(x)(\alpha)$ denotes $(f_i^L(x)(\alpha)+f_i^U(x)(\alpha))$ and I stands for [0,1]. Therefore (P_{40}) reads merely:

$$(\mathrm{P}_{41})\begin{cases} \min \{f_i(x)(\alpha);\, i=1,\dots,k\} \\ x \in X \\ \alpha \in I \end{cases}$$

Let's select a finite subset of I, S={ $\alpha_1,\dots,\alpha_m$} and let $w_1,\dots,w_m$ be real valued functions such that:

(i) $w_j(\alpha) \geq 0 \quad \alpha \in$ I ; j=1,…,m;

(ii) $w_j(\alpha_i) = \begin{cases} 1 & if\ \ i=j \\ 0 & if\ i \neq j \end{cases}$.

Define the operator K as follows:

$$\mathrm{K}\,[f_i(x)(\alpha)]=\textstyle\sum_{j=1}^{m} w_j(\alpha)\, f_i(x)(\alpha_j).$$

Consider now the following mathematical programs:

$$(\mathrm{P}_{42})\begin{cases} \min \{\mathrm{K}[f_i(x)(\alpha)]; \quad i=1,\dots,k\} \\ x \in X \\ \alpha \in [0,1] \end{cases}$$

$$(\mathrm{P}_{43})\begin{cases} \min \{f_i(x)(\alpha_j);\ j=1,\dots,m;\ i=1,\dots,k\} \\ x \in X \\ \alpha \in [0,1] \end{cases}$$

The following theorem tells us that (P_{42}) and (P_{43}) are equivalent in terms of efficiency.

Theorem 3.2
x^* is efficient for (P_{42}) if and only if x^* is efficient for (P_{43}).

Proof
(1) $\Longrightarrow$
Suppose that x^* is efficient for (P_{42}) and not efficient for (P_{43}).

Then there is no x ∈ X such that:

$\mathrm{K}\, f_i(x)(\alpha) \leq \mathrm{K}\, f_i(x^*)(\alpha)$ ∀ i = {1,…,k}, $\forall\alpha \in \mathrm{I}$ (3.29)

and

$\mathrm{K}\, f_\ell(x)(\bar{\alpha}) < \mathrm{K}\, f_\ell(x^*)(\bar{\alpha})$ for some ℓ ∈ {1,…,k} and for some $\bar{\alpha} \in \mathrm{I}$. (3.30)

In the same time we have, by the fact that x^* is not efficient for (P_{43})**,** that there is x∈ X such that :

$$f_i\,(x)(\alpha_j) \leq f_i(x^*)(\alpha_j) \qquad \forall\ \text{i}=\{1,\dots,\text{k}\}; \qquad \forall\ \text{j} = \{1,\dots,\text{m}\}$$

and

$$f_\ell\,(x)(\alpha_s) < f_\ell(x^*)(\alpha_s) \text{ for some } \ell \in\{1,\dots,\text{k}\} \text{ and for some s } \in\{1\ ,\dots,\text{m}\}.$$

Let now α be chosen arbitrarily in I. As $w_j(\alpha) \geq 0$ and $w_j(\alpha_j)$ =1, we have that

$$f_i\,(x)(\alpha_j) - f_i(x^*)(\alpha_j) \ \leq 0 \qquad \forall\ \text{i}=\{1,\dots,\text{k}\}; \qquad \forall\ \text{j}=\{1,\dots,\text{m}\}$$

and

$$f_\ell\,(x)(\alpha_s) - f_\ell(x^*)(\alpha_s) < 0 \text{ for some } \ell \in\{1,\dots,\text{k}\} \text{ and for some s } \in\{1\ ,\dots,\text{m}\}.$$

We can say that there is x∈ X such that:

$$\sum_{j=1}^{m} w_j(\alpha)\left[f_i\,(x)(\alpha_j) - f_i\,(x^*)(\alpha_j)\right] < 0 \qquad \forall\ \text{i} = \{1,\dots,\text{k}\}.$$

This means there is x∈ X such that:

$$\sum_{j=1}^{m} w_j(\alpha)\, f_i\,(x)(\alpha_j) < \sum_{j=1}^{m} w_j(\alpha)\, f_i\,(x^*)(\alpha_j) \qquad \forall\ \text{i} = \{1,\dots,\text{k}\}.$$

As α is chosen arbitrarily, we can say that there is x∈ X such that:

$$\mathrm{K}\, f_i(x)(\alpha) \leq \mathrm{K}\, f_i(x^*)(\alpha) \quad \forall\ \text{i}, \ \forall\alpha \in \mathrm{I}.$$

This contradicts (3.29) and (3.30 and we may conclude that x^* should be efficient for (P_{43}).

(2) ⟸
Suppose now that x^* is efficient for (P_{43}) and not efficient for (P_{42}).
Then there is no x∈ X such that:

$f_i\,(x)(\alpha_j) \leq f_i(x^*)(\alpha_j)$ ∀ j∈{1,…,m} ∀ i∈{1,…,k} (3.31) and

$f_\ell\,(x)(\alpha_s) < f_\ell(x^*)(\alpha_s)$ for some l∈{1,…,k}; and for some s∈{1 ,…,m} (3.32)

In the same time we have, by the fact that x^* is not efficient for (P_{42}), that there is x∈ X such that:

$\mathrm{K}\, f_i(x)(\alpha) \leq \mathrm{K}\, f_i(x^*)(\alpha)$ ∀ I = {1,…,k}, $\forall\alpha \in \mathrm{I}$ (3.33)

and

$\mathrm{K}\, f_\ell(x)(\bar{\alpha}) < \mathrm{K}\, f_\ell(x^*)(\bar{\alpha})$ for some $\ell \in$ {1,…,k} and for some $\bar{\alpha} \in \mathrm{I}$. (3.34)

This means by (3.33) and (3.34) that there is x∈ X such that:

$\sum_{j=1}^{m} w_j(\alpha)\left[f_i\,(x)(\alpha_j) - f_i\,(x^*)(\alpha_j)\right] \leq 0$ ∀ i={1,…,k} $\forall\alpha$ ∈I (3.35)

and

$\sum_{j=1}^{m} w_j(\bar{\alpha})\left[f_\ell\,(x)(\alpha_j) - f_\ell\,(x^*)(\alpha_j)\right] < 0$ for some ℓ ∈{1,…,k}, and some $\bar{\alpha}$ ∈I.(3.36)

By the fact that $w_j(\alpha) \geq 0$ $\forall\, j$ we have from (3.35) that:
$f_i\,(x)(\alpha_j) - f_i\,(x^*)(\alpha_j) < 0$ ∀ j∈{1,…,m} ∀ i∈{1,…,k} (3.37)

Moreover, by the fact that $w_j(\bar{\alpha}) \geq 0$ $\forall\, j$ we have from (3.36) that:

$f_l\,(x)(\alpha_s) - f_l(x^*)(\alpha_s) < 0$ for some l∈{1,…,k}; and for some s∈{1 ,…,m} (3.38)

(3.37) and (3.38) contradicts (3.31) and (3.32) and x^* should be efficient for (P_{42}).

The interpretation that goes with this theorem is as follows.
Finding an efficient solution of (P_{43}) is tantamount to find an efficient solution of (P_{41}) where $f_i\,(x)$ i = 1,…,k are replaced by $\mathrm{K}\, f_i(x)$ i = 1,…,k.

3.3.2 Description of the algorithm for solving the program (P_{37})

$h = h(\alpha_1, \dots, \alpha_m) = \max\limits_{\alpha\in\mathrm{I}} \min\limits_{1\leq i\leq m} |\alpha - \alpha_i|$ is called the roughness of the grid

S= $(\alpha_1, \dots, \alpha_m)$.

It is well known (see e.g. [19]) that the discretization error decreases when the grid is refined. That is when h is kept to a lower extend.

The foregoing discussion leads us to describe the following algorithm for solving the equivalent deterministic program (P_{37}).

Step 0
Fix an acceptable bound of error for h, ε.

Step 1
Read data of (P_{37}).

Step 2
Write the equivalent deterministic program (P_{39}).

Step 3
Put ℓ =0.

Step 4
Take a discretization S={ α_{ℓ_1} ,…, $\alpha_{\ell_{m\ell}}$} of I

Step 5
Write the discretization problem (P_{43}).

Step 6: Find an efficient solution of (P_{43}).

Step 7
Compute $h = \max_{\alpha \in I} \min_{1 \le i \le m} |\alpha - \alpha_i|$ and check whether |h|< ε.

If this is truth Go to Step 9 otherwise Go to Step 8.

Step 8
Take a finer discretization S', put S=S' and Go to step 5.

Step 9
Print the solution obtained.

Step 10
Stop.

CHAPTER 4: APPLICATIONS

4.1 Mini-Decision Support System for Multiobjective Programming Problems (MDMOPP)

4.1.1 Preamble

MDMOPP is a decision support system for Multiobjective Programming Problems. The main aim of this decision support system is to help decision makers faced with problems that may be cast into a deterministic or fuzzy multiobjective programming framework, to choose an appropriate technique for dealing with the problem at hand and to solve it.
MDMOPP assumes that the multiobjective program to be solved has been completely formulated by the decision maker and that information concerning the decision maker preferences is available.

4.1.2 Components of MDMOPP

The MDMOPP has three main components, namely a database, a modelbase and a software system. In the following subsections we describe each of these components.

4.1.2.1 Database

The MDMOPP database is able to store a collection of data files. These data files contain a combination of numerical and alphabetical data. Although some of the data are stored directly in computer, some of them may be stored on the internet. The files are protected by a security code activated by the decision maker.

4.1.2.2 Modelbase

The modelbase consists of the following methods: the compromise programming method, the genetic algorithm, the goal programming method, the lexicographic goal programming method, the reference point method, the weighting method, the fuzzy approach based on possibility measure and the fuzzy approach based on the Embedding Theorem.
All these methods have been described in detail in chapters 2 and 3.

4.1.2.3 Software subsystem

The software subsystem of MDMOPP consists of three components: database management software (DBMS), model base management software (MBMS) and dialogue generating management software (DGMS).
Through these components, the interface with the analyst is realized by a sequence of windows. Each window has distinct expression which helps considerably to facilitate the analyst's work. The analyst is able to move between the next window and the previous one, which helps him to make adjustments or corrections about information already entered into the computer. The solution process (which refers to steps that the analyst follows to solve the problem) of a given multiobjective problem

can be stopped at any step and can be continued again at a later stage. This allows the analyst to update, rearrange, retrieve and inquire about the objectives, constraints, variables or solution of the problem.
The model base management software of MDMOPP is able to manage the methods by choosing an appropriate one for a given problem. It should also be able to link the method to relevant data. The dialogue generating management system of MDMOPP provides the dialogue between the analyst and the computer. The dialogue generating management system ensures that there is interaction between the MDMOPP, the analyst and the operating system. This interaction refers to entering data from the given program (P) and information about the preferences of the decision maker.

4.2 Functioning of MDMOPP

4.2.1 Actions to perform

To solve a multiobjective program (P), the following basic actions are performed:

- Analyst answers questions about the problem (P) and about preferences of the decision maker. Here are some examples of questions that may be asked:
  - Is at least one objective or constraint of the problem at hand nonlinear?
  - Is the decision maker willing to work through the solution process together with the analyst?
  - Is the decision maker willing to have as many options as possible to choose from?
  - Does the decision maker have specific targets for each objective?
  - Does the decision maker regard some objectives as more important than others?
- Analyst enters data into MDMOPP.
- MDMOPP checks data for more details.
- MDMOPP solves (P) according to the method chosen.

4.2.2 Flowchart of MDMOPP

Here is a flowchart that summarizes the MDMOPP functioning:

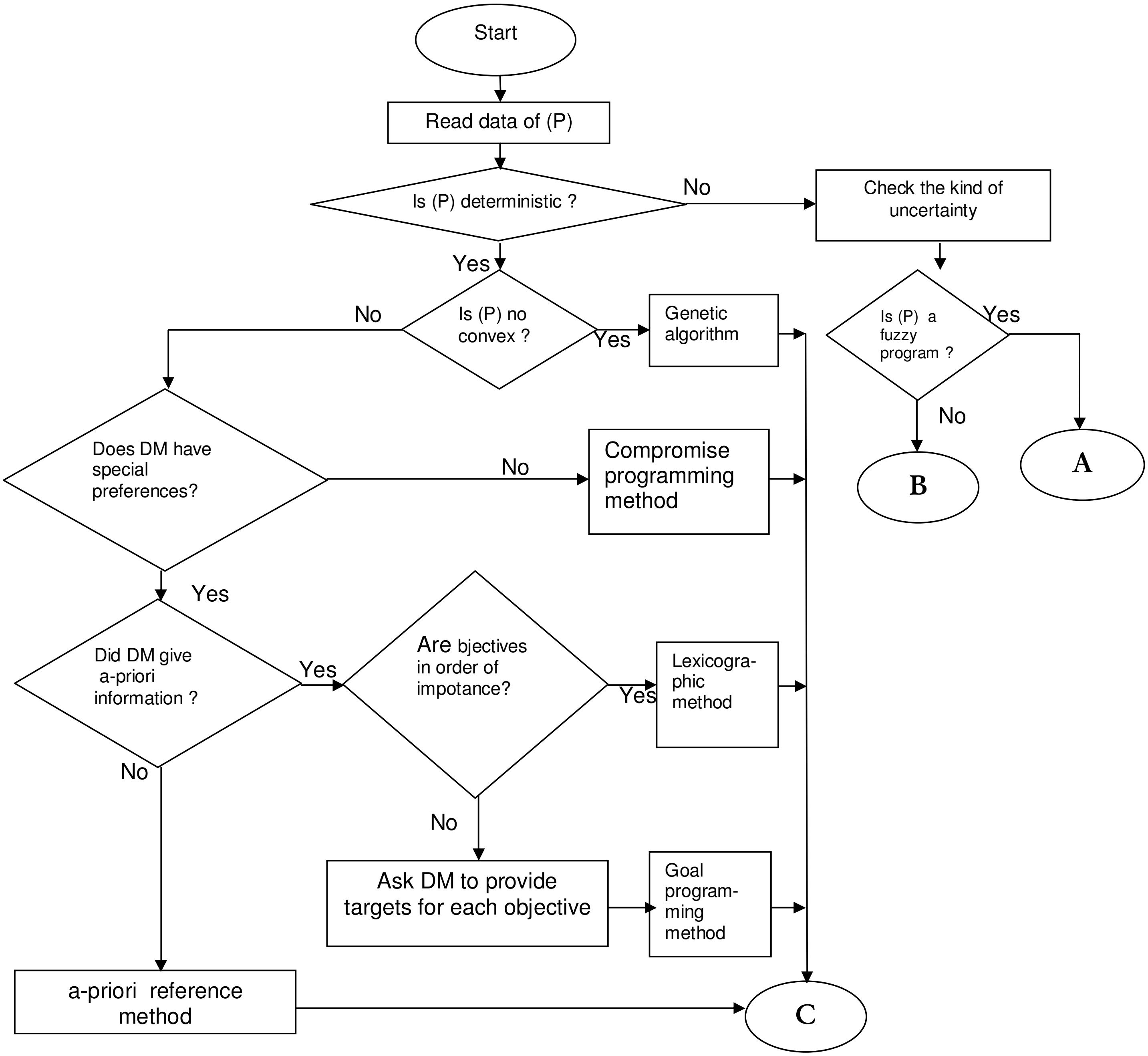
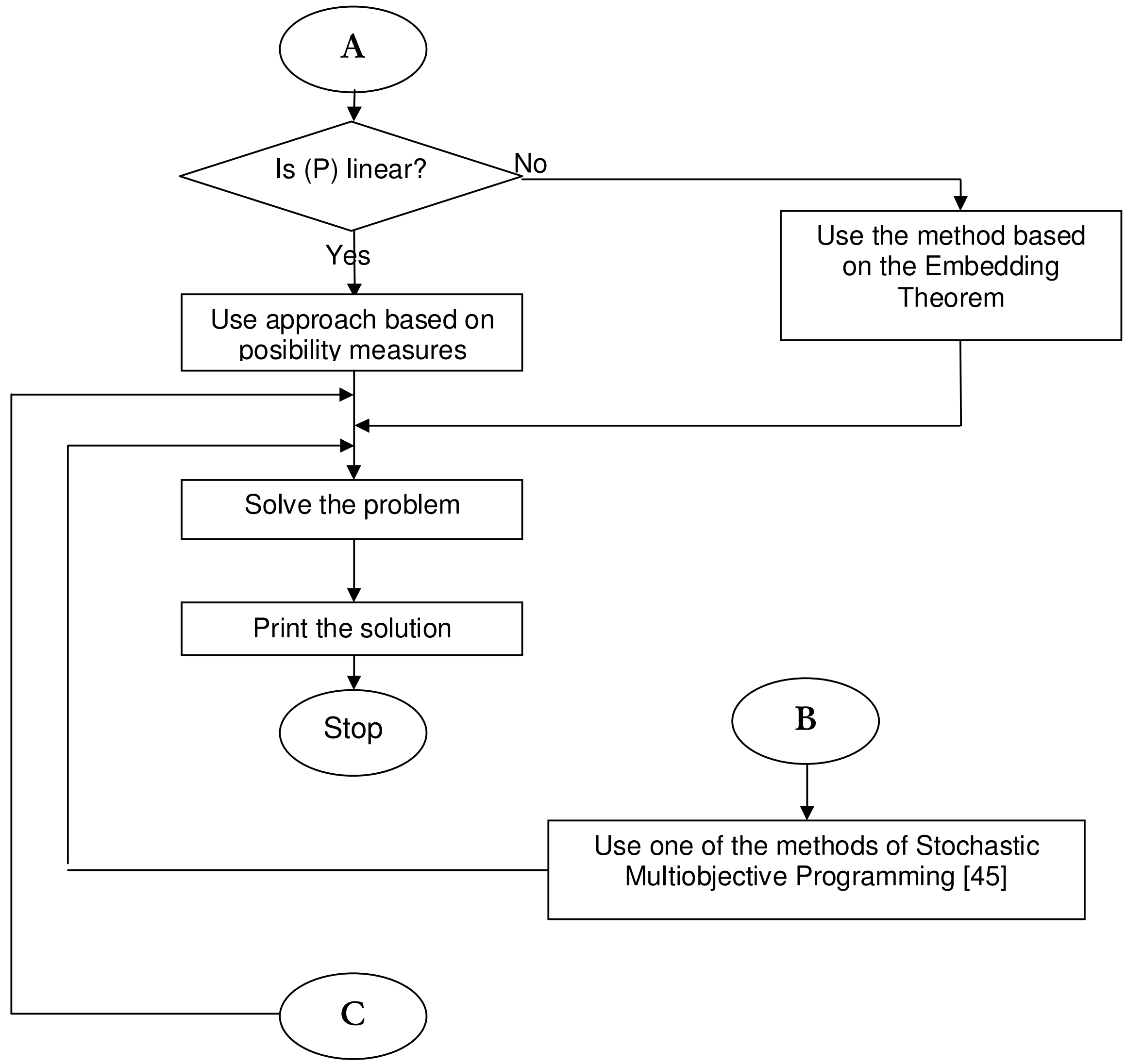


Figure 4.1: Flowchart of Mini-Decision Support System for Multiobjective Programming Problems.



The full implementation of this mini-system is a subject of further research.

4.3 Case study

4.3.1 Problem formulation

The Nyarutarama Lake, located in Kigali City, is famous for its flora which attracts migratory birds and fish. The lake is connected to many rivers. The most important ones: Karenge, Kimisagara and Nyabarongo are controlled by reservoirs. These reservoirs are managed by Rwanda Water and Sanitation Corporation (RWASCO). They provide drinking/irrigation water and fresh water to the Nyarutarama Lake. Reservoir releases for the lake contribute to flora growth.
For a given reservoir, RWASCO has decided monthly on water quantity to release for irrigation, for drinking and for the lake. These releases will be decided regarding some objectives: satisfying the drinking and irrigation water demands, controlling the drought and flood periods and contributing to the preservation of the Nyarutarama Lake.
The problem consists of determining optimal release from different reservoirs.

4.3.2 Mathematical formulation of the problem

Data related to resources and reservoirs management must be considered to build the objectives and constraints of the problem.
For this problem the decision variables are the water releases, from different reservoirs, for irrigation, $I_{i,t}$, for drinking, $D_{i,t}$ and for the Nyarutarama Lake, $L_{i,t}$; i and t are reservoir and time indices respectively.
The decision maker has to specify for each reservoir the flood, FC control storage. Moreover, the following parameters must be introduced in order to begin the solution procedure:

- maximum release from each reservoir $M_{i,t}$;
- minimum release of the lake at the end of the period t, $\mathrm{MRL_t}$.

The variables used to formulate the problem may be summarized as follows.

- *Decision variables:*

$I_{i,t}$: Releases for irrigation demand from reservoir i at the end of period t.
$D_{i,t}$: Releases for drinking demand from reservoir i at the end of period t.
$L_{i,t}$: Releases for the Nyarutarama Lake demand from reservoir i at the end of period t.

- *Parameter variables:*

$M_{i,t}$: Estimation of releases for drinking, irrigation and the lake from reservoir i.
MRLt: Minimum releases for the Lake at period t specified by the Environment Ministry.
$D_{D,i,t}$: Maximum demand for drinking water from reservoir i at the end of period t.
$D_{I,i,t}$: Maximum irrigation demand from reservoir i at the end of period t.

In our application the period t is equal to one month and the index i is equal to 1 for Karenge reservoir, to 2 for Kimisagara reservoir and 3 for Nyabarongo reservoir.

- *Objectives of the problem:*

The goals of the problem are as follows:
Goal for the control of the drinking water supply: the goal here is to satisfy the demands

$$D_{i,t} \geq D_{D,i,t} \ \forall_i.$$

Goal for the control of the irrigation water supply: the goal here is to satisfy the irrigation demands

$$I_{i,t} \geq D_{I,i,t} \ \forall_i.$$

- *Constraints of the problem:*

The constraints of the system are as follows:

Constraint on maximum water releases:

$$D_{i,t} + I_{i,t} + L_{i,t} \leq M_{i,t} \quad \forall \text{i}$$

Constraint on Lake releases:

$$L_{i,t} \geq MRL_t. \qquad \forall \text{i}$$

Non negativity constraint:

$$D_{i,t},\ I_{i,t},\ L_{i,t} \ \geq 0, \qquad \forall_i.$$

As the targets $D_{D,i,t}$ and $D_{I,i,t}$ are given for each of the objectives, we see from our mini-Decision Support System that we have to use the goal programming method. This means we have to minimize positive and negative deviations from these targets.

Let $e_{D,i,t}$ be the deviation between $D_{i,t}$ and $D_{D,i,t}$, that is $e_{D,i,t} = D_{i,t} - D_{D,i,t}$.
Let also $e_{I,i,t}$ be the deviation between $I_{i,t}$ and $D_{I,i,t}$.
The resulting mathematical program is:

$$(\text{P}_{44}) \begin{cases} \min[\sum_{i=1}^{3} (e^+_{D,i,t} + e^+_{I,i,t} + e^-_{D,i,t} + e^-_{I,i,t})] \\ D_{i,t} - e^+_{D,i,t} + e^-_{D,i,t} = D_{D,i,t} & \forall \text{i} \\ I_{i,t} - e^+_{I,i,t} + e^-_{I,i,t} = D_{I,i,t} & \forall \text{i} \\ D_{i,t} + I_{i,t} + L_{i,t} \leq M_{i,t} & \forall i \\ L_{i,t} \geq MRL_t & \forall \text{i} \\ D_{i,t},\ I_{i,t},\ L_{i,t} \ \geq 0, & \forall i \end{cases}$$

where $e^+_{D,i,t},\ e^+_{I,i,t},\ e^-_{D,i,t},\ e^-_{I,i,t}$ are positive and negative deviations respectively.

For the considered month t (September 2008), the following data, expressed in millions of liters of water, have been collected from RWASCO:

i	$D_{D,i,t}$	$D_{I,i,t}$	$M_{i,t}$
1	2.9	1.613	3.2
2	0.185	0.216	3.2
3	0.787	0.517	3.2

MRL_t= 1.8

Table 4.1: Data collected from RWASCO.

4.3.3 Solution of the problem

With above data and by denoting positive and negative deviations by EDITIV, EIITV, EDITU, and EIITU respectively, LINGO instructions read:

MIN = ED1TV + EI1TV + ED1TU + EI1TU + ED2TV + EI2TV + ED2TU + EI2TU
+ ED3TV + EI3TV + ED3TU + EI3TU ;

D1T + ED1TU - ED1TV = 2.9
D2T + ED2TU – ED2TV =0.185
D3T + ED3TU – ED3TV = 0.787
I1T + EI1TU – EI1TV = 1.613
E2T + EI2TU – EI2TV = 0.216
I3T + EI3TU – EI3TV = 0.517
D1T + I1T + L1T≤ 3.2
D2T + I2T + L2T ≤ 3.2
D3T + I3T + L3T ≤ 3.2
L1T ≥ 1.8
L2T ≥ 1.8
L2T ≥ 1.8.

Using LINGO « SOLVE » Command yields the following solution:

D1T = 1.4
D2T = 0.185
D3T = 0.787
I1T = 0.0
I2T = 0.216
I3T = 0.517
L1T = 1.8
L2T = 2.799
L3T = 1.896.

This means that the following releases (expressed in millions of liters of water) for drinking / irrigation and for the Lake, are optimal.
Releases from:

- Karenge reservoir: 1.4 for drinking, 0.0 for irrigation and 1.8 for the Lake;

- Kimisagara reservoir: 0.185 for drinking, 0.216 for irrigation and 2.799 for the Lake;

- Nyabarongo reservoir: 0.787 for drinking, 0.517 for irrigation and 1.896 for the Lake.

CONCLUDING REMARKS

Multiobjective Programming Problems are encountered in many areas of applications.

In this dissertation we have highlighted the fact that it is possible to solve these problems using adequate techniques rather than relying on the simplistic way of replacing stubbornly different conflicting objectives by a single one.
It is worth mentioning that some of ideas developed here will appear in a forthcoming issue of the journal Advanced in fuzzy sets and systems [33].

It has also been shown that Fuzzy set Theory may be of great help as both a language and a tool for dealing with Multiobjective Programming Problems with fixed and fuzzy data.

We have also described a mini-Decision Support System aiming at helping people facing these problems. This system helps both in the choice of an appropriate technique and in the solution process.

Among lines for further developments in this field, we may mention the following.

- The consideration of both possibility and necessity measures in the approach described in § 3.2.
- The extension of methods described in Chapter 3 to the case where fuzziness and randomness are in state of affairs.
- The full implementation of the mini-Decision Support System described in § 4.1.

REFERENCES

[1] Anstreicher,K.M., den Heltog,D., Roos,C., and Terlaky, T., A Long Step Barrier Method for Convex Quadratic Programming, Algorithmica, 10 (1993):365.

[2] Atkinson, D. S. and Vaidya, P. M., A cutting Plane Algorithm for Convex Programming that uses Analytic Centers, Math programming, Vol 69 (1995) 1-43.

[3] Avriel,M., Nonlinear Programming: Analysis and methods, Printice Hall, 1976.

[4] Bazaraa, M.S. and Bouzaher, A., A Linear Goal Programming Model for Developing Economies with an Illustration from the Agriculture Sector in Egypt, Mgt Science, vol. 27, (4) 1981.

[5] Belaïd, A. and Martel,J.M., Goal programming: Theory and applications, INFO JOURNAL, vol. 42 (3), 2005.

[6] Bellman, R.E., and L.A. Zadeh, Decision-making in a fuzzy environment. Mgmt Sci. 17 (1970) 141-166.

[7] Beltrami, E.J., Models for Public Systems Analysis, Academic Press, 1977.

[8] Benson, H. P., An Algorithm for Optimizing over the Weakly-Efficient Set, European J. Operational Research 25 (1986) 192-199.

[9] Bertsekas,D.P., Nonlinear programming, Second Edition, Athena Scientific, 2003.

[10 Blume, L.E., Convex Programming for The New Palgrave Dictionary of Economics, 2nd ed, Basingstone, 2008.

[11] Chankong V and Haimes, Y.Y.. The Interactive Surrogate Worth Trade-Off (ISWT) Method for Multiobjective Decision Making, Multi-criteria Problem Solving 155, (1978)42-67.

[12] Clover,F., Tabu search and adaptive memory programming-advances, applications and challenges, CiteSeerX ,1996.

[13] Da Cunha N.O. and Polak, E., Constrained minimization under vector valued-criterial in finite dimensional spaces, J. Mathematical Analysis & Applications, vol. 19 (1) (1967)103-124.

[14] Dantzig, G.B., Linear programming, Theory and Extensions, Springer-Verlag, 2003.

[15] *Dantzig, G. B. Linear Programming and Extensions. Princeton, NJ: Princeton University Press, 1963.*

[16] Dubois, D. and Prade,H., Fuzzy sets and systems: Theory and Applications. Academic Press, New York, 1980.

[17] Ehrgott M. and Wiecek, M., Multiobjective Programming, Springer- Verlag, 2005.

[18] Freimer M. and Yu, P.L., Some New Results on Compromise Solutions for Group Decision Problems, Management Science, Vol.22 (1976) 688-946.

[19] Glashoff, K., Gustafson, S-A., Linear Optimization and Approximation, Berlin, Springer-Varlag, 1983.

[20] Gomez-Skarmeta, A.F., Pareto optimality in Fuzzy Modeling, Pennsylvania State University, 1998.

[21] Gonzaga C. C., Path following methods for linear programming, SIAM Review 34 (1992) 167-227 .

[22] Ignizio, J.P., Goal programming and extensions, Lexington Books, 1976.

[23] Jaworski M., Methods of step-size distribution optimization used in S-SSFM simulation of WDM systems (Journal of Telecommunications and Information Technology) National Institute of Telecommunications Warsaw, vol. (2009) 44-50.

[24] Jong-hyun Ryu, Sujin Kim and Hong Wan, Pareto front approximation with adaptive weighted sum method in multiobjective simulation optimization, Winter Simulation Conference, (2009) 623-633.

[25] Karmarkar, N. K., A new polynomial time algorithm for linear programming, Combinatorica, vol. 4 (1984) 373-395.

[26] Khachiyan, L. G., A polynomial algorithm in linear programming. Doklady Academiia, Nauk SSSR 244 S, (1979) 1093-1096.

[27] Lee Joe-Won, Lee Sukhan and Shin Dongmok, Data fusion based state estimation of nonlinear discrete systems, Vol. 1 (2000) 310-315.

[28] Lee, S. M., Goal Programming for Decision Analysis, Auerbach, 1972.

[29] Liu B., Uncertainty theory: An Introduction to its Axiomatic foundation Springer-Verlag, 2004.

[30] Luenberger,D.G., Linear and Nonlinear Progamming (Second Edition) Springer, 2005.

[31] Luhandjula, M.K., Compensatory operators in fuzzy linear programming with multiples objectives. Fuzzy Sets and Systems vol. 8 (1982) 245-252.

[32] Luhandjula, M.K., Ichihashi H. and M.Inuigushi. Fuzzy and Semi-infinite Optimization. Information Sciences vol. 61(1992) 233-250.

[33] Luhandjula M.K, Ruzibiza S.S. and Adeyafa A.S., Solving Multiobjective Programming problems with fuzzy number coefficients (2011), to appear.

[34] Masatoshi, Sakawa, Interactive Multiobjective by Sequential Proxy Optimization Technique (SPOT), Kobe University, 2009.

[35] Mavrota,G., Generation of efficient solutions in Multiobjective Mathematical Programming Problems using GAMS
Effective implementation of the ε-constraint method, Scientific Commons, 2008.

[36] Miettinen, K. M., Nonlinear Multiobjective Optimization, Kluwer 1999.

[37] Minoux, M., Mathematical Programming, Wiley, 1986.

[38] Minoux, M., Programmation Mathématique – Theorie et algorithmes (Seconde Edition), Tec & Doc (Editions), 2008.

[39] Nash S.G. and Sofer A., Linear and NonLinear Programming, McGraw-Hill,1996.

[40] Pedregal, P., Introduction to Optimization, Springer, New York, 2004.

[41] Pham,D.T. and D. Karaboga, Intelligent Optimisation Techniques: Genetic Algorithms, Tabu Search, Simulated Annealing and Neural Networks, Springer-Verlag, 1998.

[42] Romero, C., Multiobjective and Goal-Programming Approches as a Distance Function Model, CiteSeerX, 1985.

[43] Sawaragi, Y.; Nakayama, H. and Tanino, T., Theory of Multiobjective Optimization (vol. 176 of Mathematics in Science and Engineering), Orlando, FL: Academic Press Inc, 1985.

[44] Scniederjans, M.J., Goal programming methodology and applications, Kluwer publishers, Boston, 1995.

[45] Stacey L. Faulkenberg, Quality representation in multiobjective programming, Clemson University, 2009.

[46] Taghreed A. H. and Mousa A.A., An Interactive Stochastic Approach for Solving Stochastic Multiobjective Production Planning Problem, Georgia Institute of Technology, Vol.4 (2010) 22-31.

[47] Türksen, I.B., A perspective on the Philosophical Grounding of Fuzzy Theories, Proceedings of IFSA (2003)1-15.

[48] Voorneveld, Mark, van den Nouweland, Anne and McLean, Rich,"An axiomatization of the Euclidean compromise solution," Working Paper Series in Economics and Finance 703, Stockholm School of Economics, 2008.

[49] Whinston, W. L., Introduction to Mathematical Programming, 4th Edition, Amazon.com, 2008.

[50] White, D. J., A Bibliography on the Applications of Mathematical Programming Multiple-objective Methods, Journal of the Operations Research society, Vol 41 (8) (1990) 669-691.

[51] Whitley, L.D., Foundations of Genetic Algorithms 2, Morgan Kaufmann, Los Altos, CA (*1993*) 75–92.

[52] Wierzbicki, A.P.,Reference Point Methods in Vector Optimization and Decision Support. Interim Report IR-98-017, International Institute for Applied Systems Analysis, Laxenburg, Austria, 1998.

[53] Winston W.L., Operations Research Applications and Algorithms, Thomson, 2004.

[54] Wu, H.C., Evaluate fuzzy optimization based on biobjective programming problems. Computers and mathematics with applications vol. 47(2004) 893-902.

[55] Zadeh, L. A., Fuzzy Sets as basis for a theory of possibility, Fuzzy Sets and Systems 1(1978) 3-28.

[56] Zadeh L. A., Fuzzy Sets, University of California, 1965.

[57] Zimmermann H.J. and P. Zysno, Latent connectives in human decision making, Fuzzy Sets and Systems vol. 4 (1980) 37-51.