INTERACTIVE MULTIPLE OBJECTIVE LINEAR PROGRAMMING

Kendal Clive Jordi

A thesis submitted to the Faculty of Science, University of the Witwatersrand, Johannesburg, in fulfilment of the degree of *Doctor of Philosophy*.

Johannesburg 1985

DECLARATION

I declare that this thesis is my own, unaided work.

It is being submitted for the degree of Doctor of Philosophy in the University of the Witwatersrand, Johannesburg. It has not been submitted before for any degree or examination in any other University.

Twenty second day of January, 1985.

ACKNOWLEDGEMENTS

I would like to express my gratitude to

Professor M Sears, Department of Applied Mathematics,
University of the Witwatersrand, Johannesburg

Professor T J Stewart, Department of Mathematical Statistics,
University of Cape Town, Cape Town

for their supervision and valuable guidance with this work.

ABSTRACT

Extensive research effort has been directed in the last ten to fifteen years towards finding suitable solution procedures for problems which have multiple objectives (criteria, goals). The overall aim of this thesis is to contribute to this rapidly growing class of solution procedures by considering the specific case of linear programming problems with multiple objectives.

The concept of a preference structure (utility function) for a Decision Maker who has a problem with multiple objectives is outlined briefly. Some of the more important results (and assumptions) about utility functions and their corresponding indifference contours are given.

A literature survey is then presented, where the important areas in Multiple Criteria Decision Making are reviewed briefly. The important interactive methods which have been proposed for solving linear programming problems with multiple objectives are then discussed in greater detail.

A new method of utilizing indifference trade-off information which is given by the Decision Maker, is then proposed. This method enables the Decision Maker to give conflicting information (to some extent) about these trade-offs. Provided that the Decision Maker satisfies certain restrictions regarding the pairs of objectives he chooses to compare, an estimate of the marginal utility vector can be computed. The properties of this resulting vector are then examined in some detail.

The estimated marginal utility vector is then used in two interactive algorithms which are presented. Provided that the Decision Maker's

underlying preference structure is either

a) linear in the case of the first algorithm, or

b) concave and twice continuously differentiable for the second,

and the Decision Maker's estimated marginal utility vector satisfies certain assumptions, both of the algorithms are shown to converge to the Decision Maker's most preferred solution.

The assumptions which the estimated marginal utility vector is required to satisfy (in order to prove convergence), enable ranges to be computed within which the true (unknown) marginal utility vector is assumed to lie. As a result, ranges can also be computed for the indifference trade-off ratios. The methods of obtaining these ranges are discussed and are illustrated by means of simple examples.

Finally, a practical problem is presented, where the second of the algorithms is used as the basic method of finding a satisfactory solution for the Decision Maker. The results of the interactions with the Decision Maker, and the need to adopt a somewhat pragmatic solution procedure, are reported and commented on in some detail.

CONTENTS

CHAPTER 1. INTRODUCTION

1.1 DEFINITION OF THE PROBLEM

We are concerned with a multiple objective (criteria) linear programming problem of the form

$$\text{Maximize} \quad f(x) \tag{1.1}$$

$$\text{subject to} \quad Ax \leq b; \quad x \geq 0, \tag{1.2}$$

where x is a vector in R^n, A is a $m \times n$ matrix and b is a vector in R^m. The function f is linear and maps $R^n \to R^r$. i.e. f is of the form

$$f \triangleq (f_1(x), \ldots, f_r(x))$$

where $f_i(x) = c_{i1}x_1 + c_{i2}x_2 + \ldots + c_{in}x_n$; $i=1,..,r$;

and the c_{ij}'s are scalar constants $\forall$ i, $\forall$ j.

Define the constraint set (1.2) as the set

$$X \triangleq \{ x \mid x \in R^n, \; Ax \leq b, \; x \geq 0 \}. \tag{1.3}$$

The constraint set X and the r objective functions which make up f are defined beforehand. i.e. they arise from some real problem which needs a solution.

The difficulty is that the r objectives are to be maximized simultaneously. We can expect that, in general, the optimal solutions for each of the objectives considered individually will differ. As a result, finding a solution which gives an improvement in one (or more) objective usually causes a decrease in at least one other objective.

<u>DEFINITION</u> 1.1 A solution $x \in X$ is said to be *nondominated* (efficient) if there is no point $y \in X$ such that

$$f_i(x) \leq f_i(y) \quad \forall\ i=1,\ldots,r$$

$$\text{and} \quad f_k(x) < f_k(y) \quad \text{for at least one } k;\ 1 \leq k \leq r.$$

Since we are trying to maximize each objective, it is clear that a solution to problem (1.1) (1.2) will come from the set of nondominated solutions in X. This leads us into the area of finding a compromise solution to (1.1) (1.2) - i.e. attempt to find a nondominated point $x \in X$ which, although not necessarily optimal for any single objective, is the overall 'optimal' (best compromise, most preferred) solution to the problem.

<u>DEFINITION</u> 1.2 The person who is responsible for choosing the final recommended solution which is to be implemented is called the *Decision Maker* (abbreviated to DM).

Consider the points x and y in the set X. We shall make the following assumption (Keeney and Raiffa(1976, pages 79-80)):

<u>ASSUMPTION</u> 1.3 Any two points x and y in the set X are comparable in the sense that one, and only one, of the following holds:

(i) the DM is indifferent between $f(x)$ and $f(y)$ ($f(x) \sim f(y)$)

(ii) the DM prefers $f(x)$ to $f(y)$ ($f(x) \succ f(y)$)

(iii) the DM prefers $f(y)$ to $f(x)$. ($f(x) \prec f(y)$)

<u>ASSUMPTION</u> 1.4 Writing $f(x) \succsim f(y)$ to mean " not $(f(x) \prec f(y))$ ", we assume that all the relations $\sim, \succ, \succsim$ are transitive.

A preference structure is defined on the *objective space in* R^r if any two points $f(x)$ and $f(y)$ in R^r (where $x, y \in X$) are comparable and there are no intransitivities.

<u>DEFINITION</u> 1.5 A function U, which associates a real number $U(f(x))$ to each point $f(x)$ in the objective space, is *said to be a Utility Function* representing the DM's preference structure provided that:

(i) $f(x) \sim f(y)$ iff $U(f(x)) = U(f(y))$, and

(ii) $f(x) \succ f(y)$ iff $U(f(x)) > U(f(y))$.

If U is a utility function representing the DM's preferences, then (1.1) (1.2) can be stated as the standard optimization problem:

$$\underset{x \in X}{\text{maximise}} \quad U(f(x)). \tag{1.4}$$

If the function U is known, or can be established beforehand, then (1.4) can be solved using a standard mathematical programming algorithm (see Section 1.3, for example). Keeney and Raiffa (1976) discuss ways of determining the DM's utility function explicitly. However, they refer to the function U as a value function and reserve the use of the word "utility" for the case where alternatives have probabilities associated with their outcomes. Since we are dealing with deterministic problems, where we assume that the outcome of each alternative is known, no confusion should arise if we follow the generally used practice in Interactive Programming of referring to the function U as the DM's utility function.

1.2 PROPERTIES OF THE UTILITY FUNCTION

The construction of utility (value) functions which reflect the DM's preferences, as well as their properties and underlying assumptions, are well documented (see Keeney and Raiffa (1976), for example). It is not our intention to discuss this topic in full, as we do not need to know (or obtain) the entire function U in the work which is developed in this thesis. The concepts and properties that are needed will be outlined. Define the points

$$A = (f_1(x), \ldots\ldots, f_r(x))$$
$$B = (f_1(y), \ldots\ldots, f_r(y))$$

and $A \neq B$.

DEFINITION 1.6 If the DM cannot establish a preference *between the* points A and B, then he is called *indifferent* between the two alternatives. i.e. a DM is indifferant between A and B if and only if $U(A) = U(B)$.

DEFINITION 1.7 All the points $f(x)$ *such that* $U(f(x)) = c$, where c is a constant, constitute an *indifference curve* for the DM in the objective function space.

PROPERTIES 1.8 Indifference curves and utility functions have the following *properties (see Ferguson* (1969), for example):

(i) *Indifference curves* are the "contours" of the utility function U.

(ii) U is defined over the whole domain. i.e. there is some (implicit) measure of the DM's satisfaction at all points $f(x)$).

(iii) It is assumed that we are in a "well behaved world" where U has a smooth second derivitive.

(iv) $$\nabla_i U(f(x)) \stackrel{\Delta}{=} \partial U(f(x))/\partial f_i > 0; \qquad i = 1,\dots,r.$$

i.e. an increase (decrease) in objective function i *will give the DM greater (less) satisfaction.*

(v) *Given some point* x such that $U(f(x)) = c$, then it is assumed that for any two objectives, say i and j, there exists a continuous function $s_i(f_j)$ such that

$$U(f_1(x),\dots,s_i(f_j),\dots,f_j,\dots,f_r(x)) = c.$$

(vi) Indifference curves cannot intersect.

(vii) Indifference curves are generally assumed to be concave.

Given some point $f(x)$ in the objective space, we need to establish the slope of the indifference curve at $f(x)$. Let $\Delta > 0$ and define

$$C = (f_1(x),\dots,f_i(x)-\lambda_{ij}(x,\Delta)\Delta,\dots,f_j(x)+\Delta,\dots,f_r(x)).$$

If the j^{th} *objective is* increased by Δ, by how much does the i^{th} objective need to decrease in order for the DM to remain indifferent between C and $A = f(x)$? Clearly, if both objectives are increased then the DM should prefer C to A, and if they are both decreased then he should prefer A to C. Therefore, given a value of $\Delta > 0$ in C, we are seeking a value $\lambda_{ij}(x,\Delta) > 0$ such that $U(A) = U(C)$.

<u>DEFINITION</u> 1.9 $\lambda_{ij}(x)$, which is obtained by taking the limit as $\Delta \to 0$ of $\lambda_{ij}(x,\Delta)$ above, is called the *marginal rate of substitution* (MRS) *of objective i for objective j.*

<u>PROPERTIES</u> 1.10 Given any two objectives i and j, where $1 \leq i \leq r$ and $1 \leq j \leq r$; $i \neq j$; we have

(i) $$\lambda_{ij}(x) = \frac{\partial U(f(x))}{\partial f_j} \Big/ \frac{\partial U(f(x))}{\partial f_i} = \nabla_j U(f(x))/\nabla_i U(f(x)) = 1/\lambda_{ji}(x)$$

(ii) $\lambda_{ij}(x) = \lambda_{ik}(x).\lambda_{kj}(x)$; $\quad 1 \leq k \leq r;\ k \neq i;\ k \neq j$.

Proof:

(i) Given any $\Delta > 0$ at $A = f(x)$, from Property 1.8 (v) there exists $\lambda_{ij}(x) > 0$ such that $U(A) = U(C)$.

i.e. $U(f_1,\ldots,f_r) - U(f_1,\ldots,f_i-\lambda_{ij}(x,\Delta)\Delta,\ldots,f_j+\Delta,\ldots,f_r) = 0$

where all the terms are evaluated at x. Adding and subtracting

$$U(f_1,\ldots,f_i-\lambda_{ij}(x,\Delta)\Delta,\ldots,f_j,\ldots,f_r)$$

in the above expression, dividing by Δ and taking the limit (which exists by Property 1.8 (iii) or (v)) as $\Delta \to 0$ gives the result.

(ii)
$$\begin{aligned}\lambda_{ij}(x) &= \nabla_j U(f(x))/\nabla_i U(f(x)) \\ &= \frac{\nabla_k U(f(x))}{\nabla_i U(f(x))} \times \frac{\nabla_j U(f(x))}{\nabla_k U(f(x))} \\ &= \lambda_{ik}(x).\lambda_{kj}(x).\end{aligned}$$
□

DEFINITION 1.11 *The vector* $\nabla U(f(x))$ (which has positive components by Property 1.8 (iv)) is called the *marginal utility vector*.

NOTE 1.12

(i) $$\nabla U(f(x)) = \nabla_j U(f(x))\left(\frac{\nabla_1 U(f(x))}{\nabla_j U(f(x))},\ \ldots,\ 1,\ \ldots,\ \frac{\nabla_r U(f(x))}{\nabla_j U(f(x))}\right)$$

$$= \nabla_j U(f(x))(\lambda_{j1}(x),\ \ldots,\ 1,\ \ldots,\ \lambda_{jr}(x))$$

$$= \nabla_j U(f(x))(\lambda_{1j}^{-1}(x),\ \ldots,\ 1,\ \ldots,\ \lambda_{rj}^{-1}(x))$$

by Property 1.10 (i).

The vector on the right hand side of the expression can be obtained by comparing each objective i, $i = 1,\ldots,r$ $(i \neq j)$

with the reference objective j $(1 \leq j \leq r)$, one at a time, at the point x to obtain the marginal rates of substitution $\lambda_{ij}(x)$.

(ii) From Property 1.10 (i) we have that

$$\frac{1}{\lambda_{ij}(x)} = \frac{\partial U(f(x))}{\partial f_i} \Big/ \frac{\partial U(f(x))}{\partial f_j}$$

$$\approx \frac{\Delta f_j(x)}{\Delta f_i(x)} .$$

Therefore, the DM can approximate a vector which is collinear with the marginal utility vector by answering the $r-1$ questions: "With all other objectives held constant at the point x, by how much are you prepared to decrease the value of objective i to obtain an increase Δf_j in objective j?" (Dyer (1973a)). □

The fact that the marginal utility vector is collinear with the vector containing the inverse of the marginal rates of substitution of $r-1$ of the objectives with respect to the remaining objective is important in some of the work which is presented later on. In addition, if the $\lambda_{ij}(x)$ are obtained by asking the questions posed in Note 1.12 (ii), then Property 1.10 (ii) can be used as a check on the consistency of the DM who is making these estimates by using more than one reference objective and posing further questions.

We now consider a special form of the utility function which will simplify the mathematical analysis considerably. The underlying assumptions on the form of the DM's preference structure are, however, fairly severe.

<u>ASSUMPTION</u> 1.13 At some point $x \in X$ the DM is presented with the vector $f(x)$ and chooses a reference objective j $(i \leq j \leq r)$. Assume that the DM feels that the tradeoffs between each of the remaining $r-1$ objectives and objective j, where each pairwise comparison is made individually, do not depend on the values of the remaining $r-2$ objectives.

<u>PROPERTY</u> 1.14 If Assumption 1.13 holds at some point $x \in X$, then the utility function has the additive form (Keeney and Raiffa (1976, pages 108–118))

$$U(f_1(x),\ldots,f_r(x)) = \sum_{i=1}^{r} v_i(f_i(x)), \tag{1.5}$$

where v_i is a utility function defined over the single objective function i. This function is assumed to be strictly increasing since we are trying to maximize each objective function. □

It is often true that a DM is prepared to pay less and less for a positive, fixed change of Δ units in objective i as f_i increases. If this is the case then $v_i(f_i(x))$ is strictly concave: i.e. it exhibits, in the language of classical economics, a decreasing marginal utility. If the property is true for each of the objectives, then $U(f(x))$, as defined in (1.5), is strictly concave. Keeney and Raiffa (1976, page 89) demonstrate, however, that the function $v_i(f_i(x))$ may not be concave. Despite this, we retain our assumption (Property 1.8 (vii)) that the overall utility function $U(f(x))$ is concave.

In practise, it is often convenient to approximate each utility function v_i for objective i by the linear function $u_i f_i(x)$, where u_i is a positive, constant scalar. This approximation is equivalent to making the following extreme assumption:

<u>ASSUMPTION</u> 1.15 The marginal rates of substitution between any two objectives do not depend on x : i.e. the local rate is also global and is applicable at any point and to substitution in any amount. (Therefore, we do not need to let $\Delta \to 0$ in Definition 1.9.)

PROPERTY 1.16 If Assumption 1.15 is satisfied then the utility function has the form

$$U(f_1(x),\ldots,f_r(x)) = \sum_{i=1}^{r} u_i f_i(x) \qquad (1.6)$$

where

$$\frac{\partial U(f(x))}{\partial f_i} = u_i \text{, a constant; } \quad i = 1,\ldots,r;$$

and u_i is positive for all i from Property 1.8 (iv). □

If Assumption 1.15 holds, then for some reference objective j, where $1 \leq j \leq r$, we have that

$$\frac{u_i}{u_j} = \frac{\nabla_i U(f(x))}{\nabla_j U(f(x))} = \lambda_{ji} = \frac{1}{\lambda_{ij}} \quad ; \quad i = 1,\ldots,r.$$

Using (1.16) we get

$$U(f_1(x),\ldots,f_r(x)) = u_j \sum_{i=1}^{r} w_{ij} f_i \quad ; \quad 1 \leq j \leq r;$$

where

$$w_{ij} = \lambda_{ij}^{-1} \quad ; \quad i = 1,\ldots,r.$$

Since we are attempting to maximize $U(f(x))$ over the constraint set X, and u_j is a positive constant (Property 1.8), this is equivalent to solving the problem (if Assumption 1.15 holds):

$$\underset{x \in X}{\text{maximize}} \quad \sum_{i=1}^{r} w_{ij} f_i(x) \qquad (1.7)$$

where j is some reference objective ($1 \leq j \leq r$), w_{ij} is the inverse of the marginal rates of substitution of objective i for objective j as defined in Definition 1.9, $i \neq j$, and $w_{jj} = 1$.

1.3 THE FRANK-WOLFE ALGORITHM

ASSUMPTION 1.17 Assume that the utility function U is two times continuously differentiable with respect to each of its arguments and that it is concave (Properties 1.8 (iii) and (vii)).

Suppose that at iteration k we have a solution $x^k \in X$. Expanding U about x^k to first order (neglecting higher order terms) gives

$$U(f(x)) \cong U(f(x^k)) + (f(x) - f(x^k)) \nabla U(f(x^k))$$

where

$$\nabla U(f(x^k)) = \left. \begin{pmatrix} \frac{\partial U(f(x))}{\partial f_1} \\ \cdot \\ \cdot \\ \frac{\partial U(f(x))}{\partial f_r} \end{pmatrix} \right|_{f=f(x^k)}$$

We wish to find $x^{k+1} \in X$ such that

$$U(f(x^{k+1})) > U(f(x^k)).$$

This is achieved by solving the problem

$$\underset{x \in X}{\text{maximize}} \quad (f(x) - f(x^k)) \nabla U(f(x^k))$$

or the equivalent problem

$$\underset{x \in X}{\text{maximize}} \quad \sum_{i=1}^{r} f_i(x) \nabla U_i(f(x^k)).$$

Call the solution y^k. Then the direction $y^k - x^k$ gives a direction of ascent. In order to find an improved solution, solve the 1-dimensional problem

$$\underset{0 \leq \rho \leq 1}{\text{maximize}} \quad U(f(x^k + \rho(y^k - x^k))).$$

This gives the optimal step length which is denoted ρ^k. Define

$$x^{k+1} \triangleq x^k + \rho^k(y^k - x^k).$$

THEOREM 1.18 If Assumption 1.17 is satisfied, then:

(i) $x^{k+1} \in X$;

(ii) if $(f(y^k) - f(x^k))\, \nabla U(f(x^k)) > 0$,

then $y^k - x^k$ is a direction of ascent and $U(f(x^{k+1})) > U(f(x^k))$;

(iii) if $(f(y^k) - f(x^k))\, \nabla U(f(x^k)) \leq 0$,

then x^k is optimal for problem (1.4).

Proof

(i) follows from the fact that x^k, $y^k \in X$, where X is a convex set, and $0 \leq \rho^k \leq 1$.

(ii) Using a Taylor Series expansion we get

$$U(f(x^{k+1})) - U(f(x^k)) = \rho^k(f(y^k) - f(x^k))\, \nabla U(f(x^k)) + \text{order } (\rho^k)^2 \text{ terms}.$$

For ρ^k small enough, the first term on the right hand side dominates the rest, which gives the result (since $\rho^k > 0$).

(iii) Using the property of concave functions

$$U(f(x)) \leq U(f(x^k)) + (f(x) - f(x^k))\, \nabla U(f(x^k)) \qquad \forall\, x \in X\,,$$

we have that

$$U(f(x)) \leq U(f(x^k)) + (f(y^k) - f(x^k))\, \nabla U(f(x^k)) \qquad \forall\, x \in X$$

since y^k *maximizes* $f(x)\nabla U(f(x^k))$ over the set X. Therefore,

$$U(f(x)) \leq U(f(x^k)) \qquad \forall\, x \in X. \qquad \square$$

NOTE 1.19 The point y^k is a *nondominated* point in X. However, the *point* x^{k+1} could be a dominated solution even if x^k is nondominated. This possibility arises because the direction $f(y^k - x^k)$ may point through the interior of the objective function space defined over X. □

Frank and *Wolfe (1956)* proposed the following algorithm to solve problems of the type defined in (1.4) where U is known.

ALGORITHM 1.20

Step 0. *Find an initial feasible solution* $x^0 \in X$.

Set $k = 0$.

Step 1. Calculate $\nabla U(f(x^k))$.

Step 2. Solve the programming problem

$$\underset{x \in X}{\textit{maximize}} \quad \sum_{i=1}^{r} f_i(x)\, \nabla U_i(f(x^k)).$$

Call the solution y^k.

If $(f(y^k) - f(x^k))\, \nabla U(f(x^k)) \leq 0$ then *stop*.

x^k is optimal *(from Theorem 1.18 (iii))*.

Step 3. Otherwise solve the 1-dimensional programming problem

$$\underset{0 \leq \rho \leq 1}{\text{maximize}} \quad U(f(x^k + \rho(y^k - x^k)))$$

for ρ^k. Set

$$x^{k+1} = x^k + \rho^k(y^k - x^k).$$

Then $U(f(x^{k+1})) > U(f(x^k))$ (from Theorem 1.18 (ii)).

Set $k = k+1$ and go to 1. □

The convergence proprties of the Frank-Wolfe algorithm are well understood (see Wolfe *(1970)*). *Its rapid initial rate* of convergence is an *important* advantage when only a few iterations are carried out.

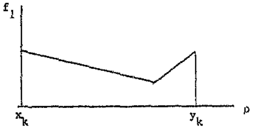
1.4 INTERACTIONS WITH THE DECISION MAKER

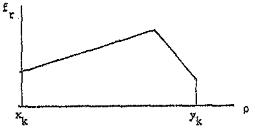
From Algorithm 1.20 we see that if the utility function U is not known, then we are required to make two interactions with the DM at each iteration. Given the nondominated solution x^k:

Interaction 1: Obtain $\nabla U(f(x))$ at the point x^k.

Interaction 2: Obtain the maximum of the utility function along the direction $y^k - x^k$, from x^k to y^k.

The second interaction is generally considered to be the easier to solve as it involves the scalar variable ρ only. Graphs can be drawn up to show the behaviour of each of the objective functions as we move from x^k to y^k. For example:





The r graphs can then be presented to the DM simultaneously for consideration and the DM would then be required to choose the "best" value for ρ.

Alternatively, a table can be computed for discrete values of ρ between 0 and 1 as follows:

Values for ρ

		0	h	2h		1
	1	$f_1(x^k)$				$f_1(y^k)$
Objectives	2 . .					
	r	$f_r(x^k)$				$f_r(y^k)$

and the DM asked to choose the most preferred column.

<u>NOTE</u> 1.21 In the special case where the utility function is assumed to be linear (1.6) or (1.7), this interaction is dispensed with since $\rho^k = 1$ will always give the maximum.

It is the first interaction where most research interest for solving problem (1.4) is taking place. Very different algorithms for solving (1.4) have been proposed which depend on how the local trade-off information (marginal rates of substitution) is obtained and how this information is then used. Some of these methods will be surveyed in the next Chapter.

CHAPTER 2. LITERATURE SURVEY

The number of publications of research into and applications of Multiple Criteria Decision Making (MCDM) has exploded since the early 1970's. Zeleny (1982, pages 60 - 61) states that "MCDM was unquestionably the fastest growing and most innovative OR/MS field of the seventies.. ... At the present time research and applications of MCDM are continuing. Interactive programming, descriptive decision models, interfaces with decision support systems and judgemental psychology, multidimensional risk analysis, and applications to strategic management policy making represent the major trends."

Starr and Zeleny (1977, page 12) state that "it is our personal opinion that.... the true foundations of serious and continuous study of MCDM were laid by Erik Johnsen in his monograph 'Studies in Multiobjective Decision Models' in 1968. MCDM was firmly on its path." Zeleny (1982, page 60) points to the first international conference on MCDM held at the University of South Carolina in 1972 as the turning point in MCDM research and applications. With many already famous and soon-to-be famous researchers in MCDM participating, the conference proceedings, edited by Cochrane and Zeleny (1973), has become a classic of sorts.

The problems of MCDM can be broadly classified into two categories (Hwang and Masud (1979, pages 6 - 7)):

a) Multiple Attribute Decision Making: characterized by a limited number of predetermined alternatives which have an associated level of achievement of the attributes (not necessarily quantifiable) based on which the final decision is to be made.

b) Multiple Objective Decision Making: alternatives are not predetermined.

The thrust of these models is to design the 'best' alternative by considering the various interactions with the design constraints which best satisfies the DM by way of attaining some acceptable levels of a set of some quantifiable objectives.

It is beyond the scope of this thesis to do a comprehensive review of MCDM and the techniques which have been developed for MCDM problems. An extensive bibliography of papers in MCDM which have been published in English is given in Zeleny (1982). We shall confine ourselves to a few of the more important areas and publications in the general field of MCDM and devote more attention to the specific topic of interactive multiple objective programming.

2.1 SURVEY PUBLICATIONS

Many of the conference proceedings on MCDM contain brief overviews of the historical development of MCDM and/or the central areas and main approaches used in MCDM. These are written by the editors of the conference proceedings as an introduction or preface to the papers which are presented. See Cochrane and Zeleny (1973), Starr and Zeleny (1977) and Bell, Keeney and Raiffa (1977) for example.

An historical survey of Multicriteria Optimization from 1776 - 1960 was done by Stadler (1979). The survey reviews the work done by Pareto which is relevent to the area and covers the development of the concepts of utility, preference and welfare theory and game theory. Efficiency, the vector maximization problem and Pareto optima are reviewed in connection with production theory, programming and economics.

Roy (1971) gave a synthesis of the main approaches to the MCDM problem. He distinguished between four different classes of approaches, giving the general philosophy of each class with their conceptual bases and discussed their practical limitations. MacCrimmon (1973) used a 'meta-model' to "relate the multiple objectives of a DM and the multiple attributes characterizing his alternatives to the model an analyst builds of these processes and choices " (page 18). A framework for examining a variety of models under four headings is given with a brief description of each model's most prominent features and underlying assumptions. More recent overviews of the general field of MCDM, or some particular aspects of the area, have been given by Cohon (1978), Hwang and Masud (1979), Zionts (1979), Hwang and Yoon (1981), Spronk (1981), Goicoechea, Hansen and Duckstein (1982), Zeleny (1982) and Haimes, Hall and Friedman (1975).

By reviewing the two main areas of MCDM (viz: Multiple Objective Decision Making and Multiple Attribute Decision Making - see the previous section) separately, Hwang and his collaborators are able to use a more refined classification of the various methods used to solve problems with miltiple criteria. Hwang and Masud (1979) give a system of classifying the major methods used in MODM which is based on four categories of the preference information given by the DM. These are:

a) a priori (before),

b) progressive (during),

c) a posteriori (after), and

d) no articulation of the preference information.

A description of the main methods under each heading is given as well as simple applications to illustrate the differences between the approaches. This work was condensed into a survey tutorial by Hwang, Paidy, Yoon and Masud (1980).

Hwang and Yoon (1981) followed up the overview of MODM with a sequel on MADM. A system of classifying the major techniques is presented. This classification is based on the three categories of information which is given by the DM:

a) no information given,

b) information given about attributes, and

c) information given about alternatives.

Simple numerical examples are given with each of the methods which is discussed under the three headings.

Zeleny (1982) gives a comprehensive overview and in depth treatment of certain areas of MCDM. Several of the chapters are concerned with his own considerable contribution (with its distinctive philosophical flavour) to the area over the past decade. An excellent treatment of MCDM is given by Goicoechea, Hansen and Duckstein (1982). The authors cover a wide range of methodologies in depth, and illustrate the different approaches with examples.

The rest of this chapter will be concerned with a survey of the main methods used for solving MODM problems. We shall use the classification of the methods suggested by Hwang and Masud (1979), and will direct most of our attention to interactive methods for solving problem (1.4).

2.2 A POSTERIORI ARTICULATION OF PREFERENCE INFORMATION

We recall that we are concerned with solving a problem of the form defined in (1.4). Algorithms have been developed to generate the set of nondominated (efficient) extreme points of the constraint set X which was defined in (1.3). (See Zeleny (1974), Philip (1972), Yu and Zeleny (1975), Steuer (1976) and Evans and Steuer (1973), for example.) Goicoechia, Hansen and Duckstein (1982) discuss four methods in detail, including those of Zeleny and Philip. Hwang and Masud (1979) also give a full discussion of several algorithms.

The set (or a subset) of nondominated extreme points is generated. Once this finite set of points has been determined and each of the objective functions has been evaluated at each of the points (alternatives), a tableau of results can be set up as follows:

OBJECTIVES

ALTERNATIVES		1	2	. . . .	r
	x^1	f_{11}	f_{12}		f_{1r}
	x^2	f_{21}	f_{22}		f_{2r}
	.				
	.				
	x^k	f_{k1}	f_{k2}		f_{kr}

$f_{ij} = f_i(x^j)$. i.e. the value of objective i at alternative j.

In these methods, any information from the DM is obtained after the set of nondominated solutions has been obtained. However, even for a relatively "small" problem, the number of alternatives which are generated can be quite large. Zeleny (1974, pages 117 - 121) gives an example with 5 objectives, 8 variables and 8 constraints which has 70 nondominated extreme points. Therefore, presenting the above tableau to the DM and asking him to choose his most preferred solution could present him with quite a formidable task.

We are essentially confronted here with a problem which falls into the discrete classification of MADM. Techniques need to be used to reduce the dimensionality of the problem which is now represented by the tableau. Methods which attempt to achieve the above include:

a) Filtering (or elimination) techniques which reduce the number of alternatives to a few (or one) by eliminating those alternatives which do not meet certain levels of performance in all the objectives.

b) Multidimensional scaling techniques which result in a "representative" one- or two-dimensional diagram of the data in the tableau. Often the data includes some "ideal" point. Certain trends that the alternatives follow with respect to the objectives can often be ascertained. The most preferred solution (or at least a few acceptable solutions) is then identified.

c) Techniques which assess (preference) weights about the relative importance of each attribute (objective). These methods require an implicit trade-off evaluation between the objectives to be made by the DM. Their evaluation principles, however, are quite diverse. Included amongst these methods is the ELECTRE method, which Hwang and Yoon (1981, page 127) consider to be one of the best methods because of its simple logic, full utilization of information (in the tableau) and refined computional procedure.

d) Techniques which utilize information about the relative importance among attributes (i.e. ordinal preferences rather than the cardinal preferences of (c)).

e) Techniques where pairwise comparisons of alternatives are made. As we noted earlier, however, the set of alternatives can be too large for this to be a practical proposition.

MADM techniques are reviewed by Hwang and Yoon (1981) and Goicoechia, Hansen and Duckstein (1982). Both references have detailed descriptions of the methods which are presented along with numerical examples.

A (possible) disadvantage of the above approach to solving (1.4) is that the methods require a finite number of discrete alternatives for analysis. If the most preferred, or 'optimal' solution to (1.4) occurs at an extreme point of the constraint set X, then this solution should be located by the DM after an analysis of the generated data has been carried out. However, as we do not know the form of the utility function beforehand (and a linear form is a severe assumption), we may in fact only be obtaining the most preferred extreme point and not the solution to (1.4). A further examination of the edges and faces of the constraint set X needs to be performed.

We note, in passing, that both Zeleny and Steuer (among others) have proposed methods which reduce the nondominated set by interacting with the DM. See Section 2.4 for further details.

2.3 PRIOR ARTICULATION OF PREFERENCES

Various methods require the DM to articulate his worth or preference structure prior to solving (1.4). The effect of this is to reduce the set of nondominated solutions to a much smaller set of solutions. The two methods which are reviewed in this section can both be used to handle discrete (as in the last section) and continuous decision variables.

2.3.1 Goal Programming

In Goal Programming (GP) each objective function is reformulated as:

$$f_i(x) + d_i^- - d_i^+ = b_i \; ; \qquad i = 1,\ldots,r;$$

where b_i is a level the DM would like objective i to achieve. The variables d_i^- and d_i^+, called deviational variables, are nonnegative measures of the under- or over-achievement respectively of the objective function f_i with respect to the desired level b_i. In GP, the deviations from the goal levels b_i are minimized. The objective function, therefore, is made up of the deviation variables.

The DM is required to provide the levels b_i for each objective. He can give information about which objectives (or goals) he is most concerned about achieving.In this case, priorities are assigned to the goals which reflect their rankings with respect to each other. In addition, within a given ranking (i.e. at the same priority level), weights can be pre-assigned by the DM to the deviation variables which indicate the DM's relative concern about achieving the goals at that priority level. If no ranking is made, the objective function is a weighted sum of the deviation variables.

The GP algorithm then solves the new formulation of (1.4) using the Simplex Algorithm to minimize the deviation variables. This is done at each priority level, in order of importance, without violating the overall levels achieved by goals at higher priorities. If a solution is obtained which achieves all the goal levels, then one or more of the goal levels b_i can usually be increased. For a full discussion of Goal Programming as well as applications to several problems consult Lee (1972) or Ignitzio (1976).

GP is an operational definition of Simon's (1957) concept of satisficing: viz. that DMs think in terms of 'bounded rationality' and look for solutions that are 'good enough'. For a re-examination of what an optimal decision is, and the process by which it can be arrived at (including the use of GP), consult Keen (1977). Harrald, Leotta, Wallace and Wendell (1978) discuss some of the shortcomings of pre-emptive goal programming and its incompatibility with utility preferences.

In order to find the DM's most preferred solution (rather than a satisfactory solution), a post-optimal sensitivity analysis needs to be performed so as to determine what effects changes in the goal levels will have on the solution. This can be performed by using the dual formulation. See Ignitzio (1976) for details. Dauer and Krueger (1977) and Isermann (1976) (1977) show how information from the dual can be employed in the search for a compromise solution. In order to perform this analysis in an efficient manner, and to avoid the problem of generating too much information, this would best be per.ormed interactively with the DM. (See Section 2.4.8.)

2.3.2 Assessing the Utility Function

If the DM conforms to certain axioms relating to choices among certain and uncertain outcomes, a utility function which represents his preference structure can be constructed. The utility function is simply a mapping of the values in the range of an objective function into a cardinal worth scale as determined by the DM. This becomes a formal mathematical representation of his preference structure. See Chapter 1.

Hwang and Masud (1979, pages 30 - 31) note that the determination of the function U for a complex problem with multiple objectives is very difficult. The advantage is that once U has been correctly assessed, (1.4) can be solved directly and it will ensure the most satisfactory solution to the DM.

Goicoechia, Hansen and Duckstein (1982) report that there are a number of theoretical and practical difficulties associated with multi-attribute utility function concepts and assessment procedures. These include:

a) The stringency of the psychological presumptions underlying the procedure.
b) The necessity of asking extreme value questions leads to problems in keeping the computational requirements for specifying a utility function at a manageable level.
c) Tedium of calculating the component utility functions and scaling constants.
d) Lack of immediate feedback to the DM of the implications of his preferences.
e) Absence of an efficient procedure to 'update' the DM's preferences and conducting a sensitivity analysis.

The approach of pre-assigning utilities appears to be better suited to problems where an alternative has an associated probabilistic outcome. i.e. if alternative x is chosen, then outcome y *will occur with a* certain probability. In (1.4), an alternative drawn from the constraint set X results in deterministic outcomes (subject to accuracy of measurement). In this context, as mentioned in Section 1.1, many authors refer to the function U, representing the DM's preference structure over the known objectives $f(x)$, as a value function. Consult Keeney and Raiffa (1976) for a reference on the construction of utility (value) functions to solve (1.4) and the underlying assumptions which the DM must satisfy.

2.4 INTERACTIVE METHODS

In this section we discuss the methods that Hwang and Masud (1979) refer to as methods for progressive articulation of information. This class of methods relies on the progressive definition of the DM's preferences along with the exploration of the objective function space defined over X. At some solution in the constraint set X, the DM is asked about some trade-off or preference information based on the current solution (or the set of current solutions). This information is used to determine a new point in X and the procedure then continues. Hwang and Masud (1979) list the following advantages and disadvantages to this approach. The advantages are:

a) There is no need for 'a priori' preference information.

b) It is a learning process for the DM to understand the behaviour of the system.

c) Only local preference information is needed.

d) Since the DM is part of the solution process, the solution obtained has a better prospect of being implemented.

e) There are less restrictive assumptions as compared to the methods described in Section 2.3.

The disadvantages include:

a) The solutions depend on the accuracy of the local preferences that the DM gives.

b) For many methods, there is no guarantee that the preferred solution can be obtained within a finite number of interactive cycles.

c) Much more effort is required of the DM than is so with methods outlined previously.

The performance of interactive algorithms, as suggested by Wallenius (1975), is judged according to:

a) The DM's confidence in the solution.

b) The ease of use of the method.

c) The ease of understanding the logic of the method on the basis of the instructions.

d) Usefulness of information provided to aid the DM.

e) Speed of convergence, measured by the number of cycles and the total time taken to solve the problem.

f) Total computer time (or CPU time).

The question of which one is the most important and which is least important is (to some extent) dependent on the DM and the decision situation. Methods for solving problem (1.4) which fall into the class of interactive algorithms should be judged against the six criteria listed above.

Since the algorithms which we propose in later chapters fall into the class of interactive algorithms, we will give a fairly detailed overview of the more important interactive methods which have been proposed at this stage.

2.4.1 The Interactive Frank-Wolfe Algorithm

Geoffrion, Dyer and Feinberg (1972) used the Frank-Wolfe Algorithm (discussed in Section 1.3) as the basis for an interactive approach to solving problem (1.4). The mathematical programming problem solved in Step 2 of Algorithm 1.20 is defined in (1.17): viz.

$$\underset{x \in X}{\text{maximize}} \quad \sum_{i=1}^{r} w_{ij}(x^k)\ f_i(x) \tag{2.1}$$

where $w_{ij}(x^k)$ is the inverse of the marginal rate of substitution of objective i ($i = 1,\ldots,r$; $i \neq j$) for some reference objective j ($1 \leq j \leq r$) at some feasible solution $x^k \in X$.

There are no underlying assumptions on the structure of the utility function (apart from the differentiability and concavity of Assumption 1.17). The weights w_{ij}, therefore, depend explicitly on the present solution x^k. These weights are approximated by posing the $r-1$ questions given in Note 1.12 (ii). Dyer (1973 b) proposed a trade-off estimation routine which creates a dialogue between man and machine by obtaining information from the DM through a series of simple, ordinal comparisons. The effects of errors in the estimation of the weights in (2.1) on the performance of Algorithm 1.20 was considered by Dyer (1974). He reports (page 173) that "if the errors are unbiased and given a stochastic interpretation, the mean initial rate of convergence of the algorithm is equal to the initial rate of convergence ignoring any error terms...(and, therfore) random errors and inconsistencies do not appear to be a significant hindrance to the use of this robust procedure".

One drawback to the method was mentioned in Note 1.19: viz. the solutions x^k, generated by the Frank-Wolfe Algorithm, could be dominated. Dyer (1974) showed that the method should converge to the efficient surface if, loosely speaking, the DM is *consistent in his* responses. Wallenius (1975), however, found in experiments that he conducted that the subjects seemed to find, on average, solutions 'nearer' to the efficient surface using an unstructured *approach on a chosen* problem than the one they found using the Geoffrion approach.

Dyer (1973 a) reports the results of experiments conducted on a group of 9 students (who had been *exposed to the basic* concepts of mathematical programming but had no background in solution techniques for problems with multiple objectives) using the Geoffrion approach to solve a linear programming problem with 3 *objectives*. He concludes that an analysis of the students' experiences indicates that the time-sharing program can be used successfully by relatively unsophisticated DMs. These findings are in *conflict with those* of Wallenius (1975) who conducted a similar experiment on a larger group of people. Wallenius found that the overall performance (measured in terms of the criteria listed at the start of this section) of the Geoffrion approach did not turn out to be as good as might be inferred from some previous experiments (viz. Dyer (1973 a)). This was mainly due to the difficulties experienced by the subjects in estimating the marginal rates of substitution which are needed in (2.1).

Rosinger (1981) attempts to overcome some of the difficulties in estimating the MRS by giving the DM the freedom to:

a) choose the way the inquiry is conducted,

b) compare the marginal utilities of certain groups of objectives only, and

c) make statements which can, to some extent, *contradict* each other.

A number $p \geqslant 1$ of subsets of at least two elements

$$J_1, \dots, J_p \subset \{ 1, \dots, r \},$$

representing groups of objectives whose marginal utilities the DM is prepared to compare, is selected. An inquiry pattern P (a $p \times r$ matrix), which represents this information, is *constructed as follows:*

$$\begin{aligned} p_{mi} &= 1 \quad \text{if} \quad i \in J_m \\ &= 0 \quad \text{if} \quad i \notin J_m, \end{aligned}$$

where $m = 1,\ldots,p$ and $1 \leqslant i \leqslant r$. P is required to satisfy certain requirements so as to ensure that adequate information is obtained from the DM.

Suppose that in (2.1) the reference objective j *is chosen to be* objective 1. Redefine

$$w_{ij} \triangleq w_{i1} \triangleq w_i,$$

where $w_1 = 1$ (since $w_{jj} = 1$) in (2.1). The DM is then asked to give the vectors

$$d_m = \{ d_{mi} : i \in J_m \} \qquad m = 1,\ldots,p \tag{2.2}$$

trying, as far as possible, to fulfill the condition that

$$d_m \text{ has the direction of } \{ w_i : i \in J_m \} \text{ for } m = 1,\ldots,p.$$

The vectors (2.2) are supposed to be proportional, by a positive factor (say a_m), to the corresponding group of weights chosen by *the Geoffrion* method. A P-answer *matrix* D *is constructed* as follows:

$$\begin{aligned} d_{mi} &= d_{mi} \quad \text{if} \quad i \in J_m \\ &= 0 \quad \text{if} \quad i \notin J_m, \end{aligned}$$

where $m = 1,\ldots,p$ and $1 \leqslant i \leqslant r$. The characteristic property of the

$p \times r$ matrix D is given by

$$p_{mi}\, d_{mi} = d_{\sigma j} \qquad m = 1,\dots,p; \quad i = 1,\dots,r.$$

The non-zero vector of weights $w = (w_1,\dots,w_r)$ is then computed by minimizing, in some sense, the matrix

$$P \begin{pmatrix} w_1 & & \\ & \ddots & \\ & & w_r \end{pmatrix} - \begin{pmatrix} a_1 & & \\ & \ddots & \\ & & a_p \end{pmatrix} D \tag{2.3}$$

for some $a_1,\dots,a_p > 0$, representing the scaling factors which were mentioned earlier.

Rosinger shows that the Geoffrion method of inquiry (choosing $j = 1$ as the reference objective) is equivalent to the DM choosing subsets

$$J_m = \{ 1, m+1 \} \qquad m = 1,\dots,r-1,$$

which leads to the inquiry pattern

$$P = \begin{pmatrix} 1 & 1 & 0 & \cdot & \cdot & 0 \\ 1 & 0 & 1 & & & 0 \\ \cdot & & & \cdot & & \\ \cdot & & & & \cdot & \\ 1 & 0 & & & & 1 \end{pmatrix}.$$

Since the DM is making the same pairwise comparisons now as in the Geoffrion method, the vector of weights w in (2.1) satisfies

$$P \begin{pmatrix} w_1 & & \\ & \ddots & \\ & & w_r \end{pmatrix} = \begin{pmatrix} a_1 & & \\ & \ddots & \\ & & a_p \end{pmatrix} D \tag{2.4}$$

for some set of scaling factors $a > 0$.

By allowing P to be more general than the above form, Rosinger has introduced more freedom (and hence flexibility) into the questioning of the DM for his MRS. In addition, by not requiring that (2.4) is satisfied, only that (2.3) is minimized in some sense, the DM can make statements which may be contradictory. Algorithm 1.20 is shown to converge under conditions which require that the estimates made by the DM in the matrix D become more accurate (with respect to their true 'implicit' values) as the algorithm progresses.

Oppenheimer (1978) combines the Geoffrion approach with a global modelling technique of the type mentioned in Section 2.3.2. He suggests two forms of utility functions as local approximations to the true utility function. These are then used in an interactive feasible directions algorithm. The author does not assume that the approximation (proxy) is the true utility function; it is only used to guide the search for an optimal decision.

Instead of throwing away past information as in Geoffrion's approach, the method uses the trade-off information at the current and previous points. This information is then used to 'fit' a proxy function of a predetermined form to the data. Since the approximation is very good locally, the algorithm converges at a faster rate than the interactive Frank-Wolfe algorithm, requiring far fewer interactions with the DM. The algorithm is stopped when the DM is unable to distinguish between successive trial solutions.

Oppenheimer reports that the algorithm has been implemented successfully on a practical problem and that the solution generated by the algorithm was adopted by a satisfied DM.

Musselman and Talavage (1980) propose a cutting plane algorithm as the basis for an interactive algorithm instead of the Frank-Wolfe algorithm. Trade-off information at some feasible point is obtained from the DM in the same manner as the Geoffrion method. Instead of using this to find a direction of movement, they construct a cutting plane (where it is assumed that the most preferred solution remains in the reduced constraint set). A point near the 'centre' of the constrained set is then found and presented to the DM for further consideration.

This method was applied to a water resources problem. The authors report that when the problem was given to a knowledable DM (a graduate student), the DM was generally able to arrive at a final compromise solution in relatively few (5 to 10) iterations using their method.

2.4.2 The Method of Zionts and Wallenius

The approach proposed by *Zionts and Wallenius* (1976) takes an initial (*arbitrary*) *set of weights* w^0 and determines some nondominated extreme point in X by using the Simplex Algorithm to solve

$$\underset{x \in X}{\text{maximize}} \sum_{i=1}^{r} w_i^k f_i(x), \tag{2.5}$$

where $k = 0$. For each nonbasic variable x_{j_k} in the simplex tableau, a vector of trade-offs z_{j_k} is computed. A linear programming problem is then solved for each nonbasic variable to determine *whether the introduction of that variable into the basis* leads to a dominated solution or not. The trade-offs which lead to nondominated extreme points are then presented to the DM for consideration. The DM is required to decide whether each set of trade-offs is desirable, undesirable or neither of the two. Wallenius and Zionts (1977, page 80) recognise that it may be preferable to compute the adjacent nondominated solutions and to ask the DM preference information about each solution when compared to the current *solution*.

At iteration k, the DM's *responses are used to constrain* the true (*unknown*) *set of weights* as follows:

(i) $$\sum_{i=1}^{r} z_{ij_m} w_i \geq \varepsilon \qquad m = 0,\ldots,k-1 \tag{2.6}$$

if the trade-offs are desirable (where z_{ij_m} represents an increase in objective function $f_i(x)$ due to some increase in the nonbasic variable x_{j_m} at iteration $m < k$).

(ii) $$\sum_{i=1}^{r} z_{ij_m} w_i \leq -\varepsilon \qquad m = 0,\ldots,k-1 \tag{2.7}$$

if the trade-offs z_{j_m} are undesirable.

$$\text{(iii)} \qquad \sum_{i=1}^{r} z_{ij_m} = 0 \qquad m = 0,\ldots,k-1 \qquad (2.8)$$

if the trade-offs are neither of the above (i.e. the DM is uncertain.)

ε is a sufficiently small positive number.

A feasible solution to the constraint set (2.6) - (2.7), with

$$\sum_{i=1}^{r} w_i = 1 \qquad (2.9)$$

$$w_i \geqslant \varepsilon \qquad i = 1,\ldots,r, \qquad (2.10)$$

is then found. This gives a new set of weights w^k and the algorithm is restarted. The procedure stops when none of the trade-offs at the current solution leads to a nondominated extreme point or when all the trade-offs presented to the DM are considered to be undesirable.

One limitation to this method is the implicit assumption that the utility function is linear since the final solution will be found at an extreme point of the constraint set X. Even if the most preferred solution does occur at an extreme point x^*, there is no guarantee that the weights $w(x^*)$ will satisfy the constraints (2.6) - (2.8) for a more general class of utility functions than the linear form. Zionts and and Wallenius (1983) suggest that 'certain' constraints be dropped if the situation arises for which no feasible weights can be found.

The major advantage of this method is that it is less demanding on the DM than most other methods. The authors report that (in practise) convergence is considerably improved if the uncertain responses in (2.8) are ignored when generating a new set of consistent weights.

de Samblanckx, Depraetere and Muller (1982) discuss some of the problems which may be encountered in the implementation of the algorithm.

In tests that they conducted on groups of students, however, they report that the students found the method easy to handle and that it converged rapidly for the problem which was being solved.

Zionts and Wallenius (1983) refined the above method to allow for an underlying pseudo-concave utility function. The presentation of trade-offs to the DM is replaced by the following possibilities:

a) Ask the DM to choose between the current solution and a distinctly different adjacent nondominated extreme point, and/or
b) ask the DM whether he likes an efficient trade-off not leading to a point asked about in (a), and/or
c) ask the DM whether he likes any trade-offs leading to adjacent efficient points which he did not prefer to the current solution in (a).

The information obtained is used to define constraints on the weights w which are similar to (2.6) - (2.10), with ε set to 1. Clearly (2.9) must be replaced with

$$\sum_{i=1}^{r} w_i = M$$

(in view of (2.10)), where the authors report that the problem of determining ε has now become one of determining M. If there is no feasible solution to the constraint set on the weights, then the set is modified by dropping the oldest of the active constraints from consideration. A new set of weights w^k is generated which is used to find a new solution.

The DM is then asked to choose between the current solution, a point found in (a) (if any) and the new solution. If the current solution is preferred, then it is locally optimal and a search procedure is implemented to find the global optimum. Otherwise the algorithm is restarted with the new solution.

The authors report considerable success in applying the method to several problems and remark that the users have generally been quite satisfied with the model.

White (1980) derives his method in part from Zionts and Wallenius. The method allows for an unspecified form of underlying utility function. An extreme point of X is found (using any prior restrictions on the weights if required). A set of weights is then used to generate further extreme points which are then compared to each other by the DM in an effort to find the best solution so far. Responses from the DM are incorporated into constraints which are used to shrink the convex cone containing the optimal weights. A new set of weights is computed from which further solutions (requiring comparison) are found.

A by-product of the procedure is an iterative method for finding the generators of the polyhedral cone containing the weights. In problems where there is evidence that $w_1 \geqslant w_2 \geqslant \ldots \geqslant w_r$ (a problem which was also considered by Pearman (1977)), the generators of the appropriate sets of parameters can be determined in a methodical manner.

As yet, no impirical testing of the method has been cited, and Zionts and Wallenius (1983) point out that the increasing number of solutions which are presented to the DM at each iteration could become a problem.

2.4.3 The Method of Belenson and Kapur

Belenson and Kapur (1973) use game theory to generate the weights for the linear programming problem:

$$\underset{x \in X}{\text{maximize}} \quad \sum_{i=1}^{r} w_i \, f_i(x) \tag{2.11}$$

$$\text{subject to} \quad \sum_{i=1}^{r} w_i = 1$$

$$w_i \geq 0 \qquad i = 1,\ldots,r.$$

Initially, r linear programming problems are solved with each w_i set to 1 in turn (the rest are set to zero). A normalised $r \times r$ matrix is then constructed which reflects how each objective function performs on each of the r nondominated extreme points which are computed. This matrix is then treated as the pay-off matrix for a two person, zero-sum game. This is then solved for the optimal mixed strategy which gives a new set of weights for use in (2.11) and a new nondominated extreme point is then found. The DM is required to identify the least preferred solution from the set of r previous solutions and this is replaced in the pay-off matrix by the new solution.

Although the algorithm has an appealing simplicity, it does not appear to have been used on any large practical problems.

2.4.4 The Method of Stewart

Stewart (1984) proposes a method which allows for the analysis of MODM problems in an interactive algorithm where the DM expresses preferences which are inconsistent with a simple utility model and/or are self-inconsistent.

Consider a simple pairwise choice between any two elements of X, say x^J and x^K. Stewart (1981) associates with this choice a probability $P[\ x^J \succ x^K\ ]$ that x^J will be stated by the DM to be preferred to x^K. If certain axioms are complied with, then it can be concluded that

$$P[\ x^J \succ x^K\ ] = (\ 1 + \exp\{\ U(x^K) - U(x^J)\ \}\ )^{-1},$$

where $U(x^J)$ and $U(x^K)$ are some 'utility indicators' of x^J and x^K. It is assumed that the utility indicator $U(x)$ is approximated by the simple linear weighted sum of the objective functions. i.e.

$$U(x) = \sum_{i=1}^{r} w_i\ f_i(x), \qquad w_i \geq 0. \tag{2.12}$$

If each element of the set S - the set of observed pairwise preference statements made by the DM - is represented by the ordered pair (J,K), indicating that the DM preferred x^J to x^K, then the maximum likelihood estimate for the weights w in (2.12) is obtained by Stewart (1984) by maximizing the log-likelihood function

$$L(\lambda/S) = - \sum_{(J,K)\ \in\ S} \log(\ 1 + \exp(-\sum_{i=1}^{r} w_i\ z_{iJK})\ ) \tag{2.13}$$

where $\quad z_{iJK} = f_i(x^J) - f_i(x^K)$

and $\quad w_i \geq 0 \qquad\qquad$ for $i = 1,\ldots,r.$

Noting that this maximum may occur at infinity along some ray in the positive orthant of R^r, Stewart justifies the inclusion of the bound

$$\sum_{i=1}^{r} w_i \leq M, \qquad (2.14),$$

where $M = 100$ was found to be convenient in practice.

The method proposed by Stewart (1984) requires the computation of the r solutions which maximize each objective function individually. The objectives are then scaled so that $f_i(x) \in [0,1]$ on the r nondominated solutions in X. A $(r+1)^{th}$ nondominated solution is generated by solving (2.12) for the scaled objectives with each $w_i = 1$. From these $r+1$ initial solutions, a set of r pairs of solutions are selected. (See Stewart for the details of how the pairs are selected.) The DM is required to state a definite preference between each pair of solutions which are presented for comparison. This information initialises the set S used in (2.13).

The maximum likelihood estimate of the weights $\hat{w}$ is computed from (2.13) (2.14) and these weights are used in (2.12) to obtain a most likely solution $\hat{x}$ and to compute a set of most likely solutions $\hat{X}^L$ (see Stewart for details). $\hat{x}$ and some point in $\hat{X}^L$ are then presented to the DM for additional preference information; the set S is updated and (2.13) (2.14) is re-solved for new weights $\hat{w}$. The algorithm continues until

a) the set $\hat{X}^L$ contains only $\hat{x}$, or

b) the DM is satisfied with the set $\hat{X}^L$ as a final short list, or

c) no further comparisons can be made between $\hat{x}$ and elements $x \in \hat{X}^L$ which have not been made previously.

Stewart (1984) notes that the linearity assumption in (2.12) is less constraining than in the previous methods discussed in the sense that some nonlinearities can be absorbed into the 'uncertainties' or *deviations* from the linear model. Also, we observe that, *although* r pairwise comparisons are made initially by the DM, only one pairwise comparison is required at each *successive* iteration. This is a general improvement on the other methods which are discussed. The only drawback is the requirement that the DM has to express a preference between two alternatives. Stewart indicates, however, that some success has been obtained in handling indifferences, but that this still needs further investigation. At this stage, no references are cited where this method has been implemented on any large practical problems.

2.4.5 The STEP Method (STEM)

In the STEP Method, originally proposed by Benayoun, de Montgolfier, Tergny and Laritchev (1971), an ideal point f^* is initially computed, where

$$f_i^* = \underset{x \in X}{\text{maximum}} \sum_{j=1}^{n} c_{ij} x_j ; \qquad i = 1,\ldots,r. \tag{2.15}$$

A solution is then computed which is nearest, in the MINIMAX sense, to f^*. i.e. solve (at iteration k)

$$\left.\begin{array}{ll} \text{minimize} & z \\ \text{subject to} & z \geq f_i^* - f_i(x)\ \pi_i \ ; \ i = 1,\ldots,r; \\ & x \in X \cap D^k, \end{array}\right\} \tag{2.16}$$

where $D^0 = R^n$. The weights π_i give the relative importance of the distance to the ideal f^* and are computed from

$$\pi_i = \alpha_i / \sum_{i=1}^{r} \alpha_i$$

where

$$\alpha_i = \frac{f_i^* - f_i^{min}}{f_i^*} \left(1 / \sum_{j=1}^{n} c_{ij}^2 \right)^{\frac{1}{2}} \qquad \text{if} \quad f_i^* > 0$$

$$= \frac{f_i^{min} - f_i^*}{f_i^{min}} \left(1 / \sum_{j=1}^{n} c_{ij}^2 \right)^{\frac{1}{2}} \qquad \text{if} \quad f_i^* \leq 0.$$

f_i^{min} is the minimum value of objective i taken over the set of r solutions used to compute f^* in (2.15).

The algorithm has 2 phases:

Phase I: Determine a solution x^k to problem (2.16) and compute the resulting performance of the r objective functions at x^k.

Phase II: (Interactive Phase) The DM compares $f(x^k)$ with f^*. He decides which objectives can be relaxed to allow an improvement in unsatisfactory objectives and the amount of the relaxation. For the next cycle, the feasible region D^k is modified

$$D^{k+1} \begin{cases} D^k & \\ f_i(x) \geqslant f_i(x^k) - \Delta f_i & \text{where satisfactory objective } i \text{ is relaxed by } \Delta f_i \\ f_i(x) \geqslant f_i(x^k) & \text{where objective } i \text{ must do no worse.} \end{cases}$$

The algorithm terminates when the DM has found a solution which has acceptable levels for all of the objectives.

Hwang and Masud (1979, page 182) report that Benayoun, in collaboration with others, has developed other techniques which are similar to STEM. Included amongst these is the Progressive Orientation Procedure (POP), which presents the DM with a subset of nondominated extreme points. If the subset contains an acceptable solution, the procedure is stopped; otherwise the DM chooses a best subset of solutions, which is used to find a new set of nondominated extreme points and the process is repeated.

2.4.5 The Surrogate Worth Trade-Off Method

The Surrogate Worth Trade-off Method (SWT) was proposed by Haimes and Hall (1974) and Haimes, Hall and Friedman (1975). Reviews of the method, as well as extensions, are given by Hall and Haimes (1975) and Haimes (1979). The method consists of three major steps. These are:

a) The identification and generation of nondominated solutions which form the trade-off functions in the objective function space.

b) The interaction with the DM to assess an indifference band (where the improvement of one objective is equivalent to the degradation of another in the mind of the DM. c.f. Definition 1.7 of an indifference curve) using what are referred to as surrogate worth functions.

c) The determination of the optimal solution set.

The trade-off functions can be determined from the values of the dual variables associated with the constraints of the problem:

$$\begin{array}{llll} \underset{x \in X}{\text{maximize}} & f_j(x) & 1 \leq j \leq r & (2.17) \\ \text{subject to} & f_i(x) \geq \varepsilon_i & i = 1,\ldots,r;\ i \neq j, & \\ \text{where} & \varepsilon_i = f_i^* - \varepsilon_i^* ; & i = 1,\ldots,r;\ i \neq j. & \end{array}$$

f_i^* are the ideal values determined in (2.15) and $\varepsilon_i^* > 0$ are deviations which are varied parametrically.

Formulating the generalised Lagrangian and applying the Kuhn-Tucker necessary conditions for an optimal solution (see Avriel (1976), for example), the conditions of interest for further analysis are

$$\rho_{ji}(\ f_i(x) - \ \varepsilon_i\ ) = 0, \quad \rho_{ji} \geq 0, \qquad (2.18)$$

where ρ_{ji} are the generalised Lagrange multipliers for (2.17) and j

is some *arbitrary* objective $(1 \leq j \leq r)$. $i = 1,\ldots,r$ $(i \neq j)$ correspond to the *additional* $r-1$ *constraints* on the remaining objective functions. Then it can be shown that (*see* Hall and Haimes (1975, pp 213 - 214), for example)

$$\rho_{ji} = -\partial f_j(\cdot)/\partial f_i(\cdot) \;; \qquad i,j = 1,\ldots,r; \;\; i \neq j;$$

which have the properties (c.f. Properties 1.10 (ii) for MRS)

$$\rho_{ji} = \rho_{jk}\,\rho_{ki} \;; \qquad 1 \leq k \leq r; \qquad (2.19)$$

$$\rho_{ji} = 1/\rho_{ij} \;; \qquad i,j = 1,\ldots,r; \;\; i \neq j. \qquad (2.20)$$

The first phase of the SWT method is to choose an arbitrary objective j and to identify and generate a set of nondominated solutions by varying the ϵ's parametrically in (2.17). Compute the trade-off functions ρ_{ji} for fixed j. Using (2.19) (2.20), the full set of trade-off functions ρ_{ji} are then computed for $i,j = 1,\ldots,r$; $i \neq j$.

The *next phase* is the interactive phase with the DM to obtain the surrogate worth *functions* s_{ji}. Given the computed values of ρ_{ji}, the value of the surrogate *worth function* is an assessment by the DM as to how much (on an ordinal scale, *say of from* -10 to 10, with 0 signifying equal preference) he prefers trading ρ_{ji} marginal units of f_j for one marginal unit of f_i (given the values of *the objectives* $f_1,\ldots,f_r$ corresponding to ρ_{ji}).

The indifference band is constructed as follows: the DM is asked whether he likes, is indifferent to, or dislikes trading ρ_{ji} units of f_j for one unit of f_i for two distinct values of ρ_{ji}. This is assessed on an ordinal scale $s_{ji}(\rho_{ji})$. ρ_{ji}^* is then approximated by finding $s_{ji}(\rho_{ji}^*) = 0$ on the line segment joining the two values of

$s_{ji}(\rho_{ji})$. The values ρ_{ji}^{*} can be checked for consistency by making use of (2.19). At the points ρ_{ji}^{*}, the surrogate worth functions are simultaneously equal to zero. i.e. marginal gain equals marginal loss for all objectives taken two at a time.

The final phase of the SWT method is to find the best compromise solution which corresponds to the values of ρ_{ji}^{*}. To each ρ_{ji}^{*} there corresponds a value $f_i^{*}(x)$, $i = 1,\ldots,r$, $i \neq j$. These $f_i^{*}(x)$ correspond to the value ε_i in constraint (2.17). The most preferred solution is then found by solving (2.17) with ε_i set to $f_i^{*}(x)$.

Hwang and Masud (1979, page 144) comment that as the number of objective functions increase, the number of trade-offs which are required from the DM may become excessive. Other disadvantages are:

a) the possible exclusion of the most preferred solution while varying the ε's parametrically in (2.17), and
b) the DM may not perceive that the ρ_{ji}'s indicate marginal trade-offs in a small region about the current solution when he is providing the surrogate worth functions.

The major advantages of the SWT method for solving (1.4) with one DM are:

a) the DM is required to consider only two objectives at a time while assessing the worth functions, and
b) there is only one major interaction with the DM to obtain the worth functions. i.e. the method is not iterative in the sense of Algorithm 1.20.

Haimes, Hall and Friedman (1975) report that the SWT method has been applied successfully to several practical problems.

Chankong and Haimes (1977) proposed an interactive version of the SWT method, imbedding it in the interactive version of the Frank-Wolfe

Algorithm. A recent approach is the Sequential Proxy Optimization Technique (SPOT), proposed by Sakawa (1982), in which the SWT method is combined with the Multiattribute Utility Function approach mentioned briefly in Section 2.3.2. SPOT parallels the approach taken by Oppenheimer (1977), where the Geoffrion method was combined with MUF (see Section 2.4.1).

In SPOT, the SWT method is used to find a nondominated solution and to compute the generalised Lagrange multipliers (trade-off functions). A proxy function is then fitted to the marginal rates of substitution at this solution which are provided by the DM. The function is used to determine a direction of improvement and the step size is then computed numerically (with DM verification) to find a new nondominated solution. The algorithm is continued until the Lagrange multipliers and the marginal rates of substitution at the current solution are sufficiently close.

As with Oppenheimer (1977), it is not claimed that the proxy function is an approximation to the full utility function; it is only a local approximation which is used to give a direction of improvement. Sakawa (1982) illustrates the algorithm with a simple numerical example and states (page 395) that applications of SPOT to environmental problems will be reported elsewhere.

Algorithm. A recent approach is the Sequential Proxy Optimization Technique (SPOT), proposed by Sakawa (1982), in which the SWT method is combined with the Multiattribute Utility Function approach mentioned briefly in Section 2.3.2. SPOT parallels the approach taken by Oppenheimer (1977), where the Geoffrion method was combined with MUF (see Section 2.4.1).

In SPOT, the SWT method is used to find a nondominated solution and to compute the generalised Lagrange multipliers (trade-off functions). A proxy function is then fitted to the marginal rates of substitution at this solution which are provided by the DM. The function is used to determine a direction of improvement and the step size is then computed numerically (with DM verification) to find a new nondominated solution. The algorithm is continued until the Lagrange multipliers and the marginal rates of substitution at the current solution are sufficiently close.

As with Oppenheimer (1977), it is not claimed that the proxy function is an approximation to the full utility function; it is only a local approximation which is used to give a direction of improvement. Sakawa (1982) illustrates the algorithm with a simple numerical example and states (page 395) that applications of SPOT to environmental problems will be reported elsewhere.

2.4.6 The Method of the Displaced Ideal

The ideal solution f^* is *initially* defined to be the vector which solves the r linear programming problems given in (2.15). f^{min}, *referred* to by Zeleny (1976, page 175) as the anti-ideal, is defined as before in Section 2.4.5.

Zeleny (1976, page 174) states the following Axiom of *Choice:*

> Alternatives that are closer to the ideal are preferred to those that are further away. To be as close as possible to the perceived ideal is the rationale of human choice.

Zeleny (1976) remarks that preference can be expressed as an 'as far as possible' concept as well, using the anti-ideal as a point of reference. He states his belief (1976, page 199) that humans try to be both as close as possible to the ideal and as far as possible from the anti-ideal, and that humans are capable of switching between these two regimes according to the given circumstances of the decision process.

The degree of closeness of objective i, $f_i(x)$, where $x \in X$, to the i^{th} component of the ideal f_i^*, is defined as:

$$d_i(x) = 1 \qquad \text{if} \quad f_i(x) = f_i^*$$

$$\text{and} \qquad 0 \leq d_i(x) \leq 1 \qquad \forall\, i = 1,\ldots,r; \quad x \in X.$$

The assignment of $d_i(x)$ can be defined by

$$d_i(x) = \frac{f_i(x) - f_i^{min}}{f_i^* - f_i^{min}} \; ; \qquad i = 1,\ldots,r; \quad x \in X.$$

The distance of a solution $x \in X$ from the ideal point f^* can then be measured using the family of L_p metrics. These are given by:

$$L_p(x,w) = \left(\sum_{i=1}^{r} w_i^p \, (1 - d_i(x))^p \right)^{1/p}, \qquad p = 1,2,\ldots , \qquad (2.21)$$

where the weights w_i are introduced to *reflect the fact* that the objectives *cannot be assumed* to have equal importance. The concept of 'as close as possible' in the Axiom of Choice is then achieved by minimizing $L_p(x,w)$ over the set X for a given set of weights and given value of p. Using a result proved by Yu (1973) which shows that the degree of closeness is interrelated with the weights, Zeleny (1976, page 188) notes that this compounding effect must be clearly understood to avoid 'double weighting'. Numerical examples of how the weights are calculated can be found in Zeleny (1974) (1982). We note in passing that Zeleny (1982, page 197) suggests forms of distance functions other than (2.21) to incorporate the weights.

The Method of the Displaced Ideal was proposed by Zeleny (1973) (1974) and (1976). For a full discussion of the method and the underlying philosophy within the context of MCDM, consult Zeleny (1982). The overall approach can be summarized as follows: Calculate the *ideal point* f^* and the anti-ideal f^{min} over the *constraint set* X; define the d_i's *for each objective* and calculate the weights w_i. Locate an approximation to the compromise set C by minimizing (2.21) over the constraint set for $p = 1,2,\infty$. Present the compromise set C and the ideal point as a reference to the DM. Using this information, the DM imposes further restrictions on the constraint set. A new ideal ('displaced') is then calculated over the constraint set and the method is restarted. The method terminates when the set C is small enough for the DM to choose a *satisfactory* solution. The anti-ideal can also be included in the optimization phase of the method to give further information about the set C.

Generally, in the literature where this methodology is explained and numerical examples are given to illustrate the approach, use is made of a discrete representation of the set X by considering the set (or a subset) of nondominated extreme points of X; or it is assumed that the objectives are equally important so as to overcome the problem of computing the weights which are needed in (2.21).

2.4.7 The Method of Steuer

At the start of Section 2.2 we mentioned that there are algorithms which can generate all the nondominated extreme points of the constraint set X. Even in relatively 'small' problems, however, the number of these solutions can be quite large. Steuer (1977) proposes an interactive method for generating a subset of the nondominated extreme points. Subinterval bounds on the weights can be prescribed (Steuer and Schuler (1978)). Using a cone which contains the optimal weights, a subset of the nondominated extreme solutions is generated from a convex combination of weighting vectors within the cone. The objective functions are then evaluated at each of the solutions and the resulting tableau is presented to the DM who is required to select his most preferred solution. The responses are used to contract and shift the cone, and a new set of weights is generated from which a new set of solutions to (2.1) is computed. The procedure is stopped when the DM finds his most preferred extreme point.

Even though the subinterval weights (on the objectives) which are used by Steuer do generate the most relevent subset of nondominated extreme points, it is still possible that this subset may be too large to realistically expect the DM to extract the most preferred solution. Steuer and Schuler (1978) incorporate a filtering method to reduce the size of the tableau which is presented to the DM still further.

2.4.8 Interactive Goal Programming

It was pointed out in Section 2.3.1 that when goal programming is used to find a solution to problem (1.4), the DM needs to pre-assign goal levels for each objective. Since these goal levels are preset by the DM without any prior information regarding

a) what can actually be achieved within the *constraint set* X, and
b) the interaction between the *goal levels* (*i.e.* as one goal is lowered, which others can be *raised* and by how much?),

a sensitivity analysis needs to be performed to evaluate the effects that changes in the goal levels have on the proposed solution. Interactive goal programming performs this analysis in consultation with the DM in an attempt to ascertain what changes in the goal levels are considered to be desirable. Spronk (1981) includes a brief overview of some of these approaches.

Dyer (1972) considers a problem originally given in the form

$$\begin{array}{lll} \text{minimize} & \sum_{i=1}^{r} (d_i^- + d_i^+) & \\ \text{subject to} & f_i(x) + d_i^- - d_i^+ = b_i ; & i = 1,\ldots,r; \\ & x \in X & \\ & d_i^-, d_i^+ \geq 0 ; & i = 1,\ldots,r. \end{array}$$

This problem is reformulated as a 'one sided' GP problem with the single objective to minimize the sum of weighted deviations from the goal levels. The goal levels are chosen so that the marginal increase in utility associated with additional units of each objective above b_i is zero. The problem is then solved using the Geoffrion approach which was described in Section 2.4.1 to determine the weightings (in the reformulated objective function) interactively for use in the interactive Frank-Wolfe algorithm.

Van Delft and Nijkamp (1977) discuss hierarchical optimization methods which are based on the assumption that the objectives (goal variables) can be ranked in an ordinal way, from the most important to the least important (c.f. concept of priorities in GP). The objectives are ranked in importance from 1 to r and a series of linear programming problems is solved. At iteration k, $k = 1,\ldots,r$, the k^{th} most important objective is maximized by solving

$$\begin{aligned} &\text{maximize} \quad f_k(x) \\ &\text{subject to} \quad f_i(x) \geq \beta_i\, f_i^*; \qquad i = 1,\ldots,k-1; \\ &\qquad\qquad\quad x \in X. \end{aligned}$$

f_i^* is the optimal solution obtained when solving problem i at the i^{th} iteration ($i < k$). $0 \leq \beta_i \leq 1$ is a tolerance parameter associated with f_i^* which is obtained from the DM.

An unambiguous ranking of the goals (or objectives) is necessary to have the procedure work properly. Spronk (1981, page 119) believes that "it is very hard, if not impossible, to obtain an appropriate ranking". Another problem is the assessment of the tolerance parameters. The above approach implicitly assumes that the DM is able to specify trade-offs between more and less important goal variables which must be valid at the optimal solution, even though this solution is unknown at the time.

The Sequential Multiobjective Problem Solving Technique (SEMOPS), proposed by Monarchi, Kisiel and Duckstein (1973), allows the DM to trade off one objective against another in an interactive manner.

A range for each objective i is denoted by $[f_{iL}, f_{iU}]$ which is not necessarily the maximum and minimum values of the i^{th} objective over the constraint set X. Nondimensionality is achieved by transforming objective $f_i(x)$ into $y_i(x)$ with a range of values in the interval $[0,1]$ by

$$y_i(x) = \frac{f_i(x) - f_{iL}}{f_{iU} - f_{iL}}\,; \qquad i = 1,\ldots,r.$$

For each objective, the DM is required to give an aspiration level AL_i which is transformed into A_i in the range $[0,1]$ by

$$A_i = \frac{AL_i - f_{iL}}{f_{iU} - f_{iL}}\,; \qquad i = 1,\ldots,r.$$

An aspiration level may be a desired upper bound, lower bound, a particular value, or bounds of intervals within which the most preferred value of an objective is (or is not) contained. For each goal defined in this way, d_i, a dimensionless indicator of attainment, is defined. d_i is, in general, nonlinear.

At each iteration k, one principle problem and m_k auxiliary problems are solved, where m_k is the number of objectives which have not been restricted by the DM in earlier iterations. At iteration k, the principle problem is

$$\text{minimize} \quad s_k = \sum_{i \in I_k} d_i$$

$$\text{subject to} \quad x \in X \cap Y_k,$$

where $I_k \subset \{1,\ldots,r\}$ is the set of objectives (or goals) which have not been restricted previously and

$$Y_k = \{ x \mid f_i(x) \text{ satisfies the imposed restriction } \forall\, i \in I - I_k \}.$$

The set of m_k (= r-k+1) auxiliary problems solved for each $j \in I_k$ is given by

$$\begin{aligned} &\textit{minimize} && s_{kj} = \sum_{\substack{i \in I_k \\ i \neq j}} d_i \\ &\text{subject to} && x \in X \cap Y_k \cap Z_j, \\ &\textit{where} && Z_j = \{ x \mid f_j(x) \text{ satisfies } AL_j \}. \end{aligned}$$

Note that at iteration 1, $I_1 = \{1,\ldots,r\}$, $Y_1 = R^n$ and $m_1 = r$.

The solutions to the principle problem and the m_k auxiliary problems are presented to the DM who, on the basis of this information, revises one of the aspiration levels. This is then included in the constraint set Y_{k+1} in the next iteration and the corresponding attainment indicator is dropped from the objective functions. During the iterations it may be found that the DM needs to revise some of the aspiration levels imposed as constraints in Y_k. The algorithm stops when the DM is satisfied with one of the solutions which are presented to him for consideration.

Advantages of SEMOPS are that it can incorporate objectives which do not have to be maximized or minimized, and that the DM can re-evaluate achievement levels at each cycle. Disadvantages are that linear problems are often translated into nonlinear problems which take a lot of time to solve, and the possibility of " ding inconsistent constraint sets when solving the auxiliary problems. This causes the determination of a systematic set of aspiration levels to become random and unsystematic (Hwang and Masud (1979, page 207)).

The remaining methods which we look at all make use of the ideal point f^*, where

$$f_i^* = \underset{x \in X}{\text{maximum}} \sum_{j=1}^{n} c_{ij} x_j ; \qquad i = 1,\ldots,r. \qquad (2.22)$$

In addition, some of the methods make use of the pessimistic point (anti-ideal) f^{min}, where f_i^{min} is the minimum value that objective function i takes on over the set of r solutions computed in (2.22).

The Goal Programming STEP Method (GP STEM), which links the STEP method and goal programming, was proposed by Fichefet (1976). This method explores the nondominated solutions which give the objectives value which are 'close' to the goal levels imposed by the DM.

The r linear programming problems of (2.22) are solved for an ideal point f^*. Using this point as a reference, the DM is asked to provide best satisfactory levels (goals) b_i for each objective i, $i = 1,\ldots,r$. The goal programming problem

$$\text{minimize} \quad \sum_{i=1}^{r} (d_i^- + d_i^+) \qquad (2.23)$$

$$\text{subject to} \quad f_i(x) + d_i^- - d_i^+ = b_i; \qquad i = 1,\ldots,r;$$

$$x \in X,$$

is solved for a solution x^g. If $d_i^- = 0$ for all i, then all the goal levels have been achieved and, generally, are then increased.

The method involves solving a set of parametric linear programming problems to explore the nondominated solutions 'nearby' the solution x^g. The parametric problems examine the effects of reducing certain goal levels, considered one at a time, on the sum of the underachievement of the goals in (2.23) (with $d_i^+ = 0$ for all i). This information is used to determine the weights of the linear programming problem (2.1), which is then solved for a solution x^c. The two solutions are then presented to the DM for consideration. If any $d_i^- > 0$, $1 \leq i \leq r$, and at least one goal level can be relaxed, this relaxation can be achieved in a systematic manner using intervals which were obtained during the parametric stages. The method terminates with solution x^c when no goal levels can be relaxed, or when the DM indicates that x^g is satisfactory.

Hwang and Masud (1979, page 225) note that GPSTEM had not been tested on large practical MODM problems at that stage. In addition, the method for obtaining the weights to solve (2.1) for x^c involves a complex procedure. However, because the DM can fix the goal levels properly, using f^* as a reference, and the relaxation of these goals can be done systematically, Hwang and Masud believe that GPSTEM may give a satisfactory solution in fewer iterations than STEM does.

Spronk (1981) proposes an Interactive Multiple Goal Programming (IMGP) method in order to overcome some of the problems and/or disadvantages encountered in the methods discussed previously. In IMGP, a potency matrix is initially constructed from the ideal solution f^* calculated in (2.22) and the corresponding pessimistic solution f^{min}. The potency matrix and some initial solution is then presented to the DM for consideration.

The DM is asked to look at the current proposed solution and to decide which objective needs to be increased first. This level is then increased in a systematic (i.e. algorithmic) manner (consult Spronk (1981) for details) and is incorporated into the problem as a constraint. (2.22) is then solved again with the added restriction and a new ideal point and pessimistic solution are calculated (c.f. Zeleny's Method of the Displaced Ideal). A trial pessimistic solution is constructed which differs from the previous solution only where the increased objective is concerned.

The new potency matrix and solution are presented to the DM for comparison with the previous ones, where the shifts in the potency matrix and solution can be viewed as a 'sacrifice' for reaching the proposed solution. If the DM is not satisfied, the level of the objective is systematically reduced until a satisfactory (to the DM) outcome is achieved. When the DM considers a sacrifice to be justified, the algorithm restarts by asking the DM which objective should be increased next. This procedure continues until the DM is satisfied with the levels achieved by all the objectives at the current solution.

The method can also be adapted to enable more than one objective to be adjusted in an iteration. The method requires a limited amount of information from the DM. The DM is only required to specify which objectives should be increased; this is then done systematically until the DM is satisfied with the outcome. No trade-off information is required and the DM is not required to specify goal levels (although information of this type can be included), or the amount of relaxation (c.f. STEM) the DM is prepared to allow. A disadvantage of this method is that it stops with a satisfactory solution. There is no guarantee that this solution is the most preferred solution and may in fact be dominated.

Spronk (1981) gives an account of IMGP when applied to several practical problems.

Hwang and Masud (1981) overcome the problem of a dominated solution with their Interactive Sequential Goal Programming (ISGP) method. Unlike the previous method, however, *ISGP requires the DM to set goal levels.*

In this method, f^* and f^{min} are computed as before and presented to the DM. The DM then sets goal levels for each objective, where

$$f_i^{min} \leq b_i \leq f_i^*; \qquad i = 1,\dots,r.$$

The following goal programming problem is then solved:

$$\text{minimize} \quad \{ \sum_{i=1}^{r} d_i^- \, , \, \sum_{i=1}^{r} -d_i^+ \} \tag{2.24}$$

$$\text{subject to} \quad x \in X$$

$$f_i(x) + w_i\, d_i^- - w_i\, d_i^+ = b_i; \quad i = 1,\dots,r; \tag{2.25}$$

$$d_i^- \leq 1; \qquad i = 1,\dots,r;$$

where $w_i = b_i - f_i^{min}$ is a normalisation factor.

The first objective is to achieve the goal levels and the second is to achieve a nondominated solution. The above problem is then solved a further r times with d_i^- set to zero in turn for $i = 1,\dots,r$ so as to generate solutions which satisfy the individual goals.

A trade-off table comprising f^*, f^{min}, the solution to (2.24) (2.25), the r additional solutions and the goal levels b is then constructed. If the DM is not satisfied with any of the $r+1$ solutions, he is asked to modify b from the information in the table. A consistency check is then performed to make sure that if the DM wants an *improvement* in one objective, that he is ready to accept a decrease in at least one other objective since the current solution is nondominated.

A new goal programming problem is then solved, where the first priority is to attain those goals which were reduced (or not changed) by the DM; the second priority is to satisfy the rest of the goals and the third priority is to ensure nondominance of the solution. A further r additional solutions are again computed for each objective considered in turn and a new trade-off table is constructed. The method continues until the DM finds a nondominated solution from the table which he considers to be satisfactory.

The authors report (page 400) that ISGP has been applied successfully to a development planning model.

2.5 CONCLUDING REMARKS

In this section we have given a brief overview of the different techniques for solving MCDM problems and then given an outline of some of the main methods which have been proposed for solving linear programming problems with more than one objective function. We have assumed throughout that only one DM is involved in the decision making process and that there is some implicit function which represents the DM's preference structure. We have seen that the methods for solving problem (1.4) are quite diverse in their philosophy and consequently in their approach.

Some of the methods which have been mentioned have been generalised to multiple objective programming problems where the objective functions are concave (rather than linear) and/or the constraint set X is a convex set (rather than (1.3)). Several of the methods have also been extended to handle the cases:

a) where more than one DM is involved and a compromise solution between DM's needs to be found, and/or

b) the models are not entirely deterministic and could have stochastic components.

CHAPTER 3 THE MARGINAL UTILITY VECTOR

Wallenius (1975) reported that the Interactive Frank-Wolfe algorithm (or the Method of Geoffrion which was discussed in Section 2.4.1) performed poorly when it was used on a test problem of the form defined in (1.4). It was found that this was primarily due to the difficulty that a Decision Maker experiences when trying to estimate the marginal rates of substitution between the objectives. Rosinger (1981) introduced a more flexible approach (than had been used previously) to the estimation of the MRS in an attempt to (partially) overcome this problem.

In this chapter we propose a method of estimating the MRS which is based on the approach developed by Rosinger. This method gives the DM the freedom to:

a) evaluate the marginal rate of substitution between any two objectives (subject to certain restrictions), and

b) make statements which *can, to some extent*, contradict each other.

Whereas Rosinger also enabled the DM to compare marginal utilities of more than two objectives simultaneously, we restrict the DM to pairwise comparisons *between objectives in order* to obtain the marginal rate of substitution between the two objectives.

We introduce an indifference trade-off matrix which contains the *information* obtained from the DM regarding his estimates of the MRS between any two objectives at some feasible solution to (1.4). The properties of this indifference matrix and its corresponding enquiry pattern are then examined. The estimated marginal utility vector (or a vector which is collinear with this vector) is then found by solving an eigenvector problem. The properties of this vector are then discussed.

3.1 THE INDIFFERENCE TRADE-OFF MATRIX

We require the DM to select a number $p \geq r-1$ *of subsets of* exactly two elements (from the r objective functions)

$$J_1,\ldots, J_p \subset \{ 1,\ldots,r \} \tag{3.1}$$

representing pairs of objectives whose marginal rates of substitution (or *indifference* trade-offs) he feels that he is able to estimate at some feasible solution. For each of these subsets

$$J_m = \{ i,j \},\ \text{say},$$
$$\text{where}\ \ 1 \leq i \leq r;\ 1 \leq j \leq r;\ i \neq j;\ 1 \leq m \leq p;$$

the DM states his indifference between a gain (or loss) of d_{mi} units in objective i for a loss (or gain) of d_{mj} units in objective j. i.e. *suppose that* we present the DM with a feasible solution x and the point

$$f(x) = (f_1(x),\ldots,f_r(x)).$$

The DM indicates that he is prepared to compare objectives i and j at the m^{th} pairwise comparison. Formulate the two points

$$A = (f_1(x),\ldots,f_i(x),\ldots,f_j(x),\ldots,f_r(x))$$

$$B = (f_1(x),\ldots,f_i(x)+d_{mi},\ldots,f_j(x)+d_{mj},\ldots,f_r(x))$$

where $d_{mi} \times d_{mj} < 0$ (i.e. one is positive and one is negative). The DM *is* then *asked* which *of* the two points he prefers. If he prefers A then, keeping d_{mi} constant, d_{mj} is increased (i.e. made more positive if it is positive and less negative if it is negative) until he prefers B to A. At that stage d_{mj} is decreased until he cannot distinguish, or is indifferent, between A and B. Then define the vector d_m to be

$$d_m = (0,\ldots,d_{mi},\ldots,d_{mj},\ldots,0) \tag{3.2}$$

where d_{mi} and d_{mj} are the indifference trade-offs between objectives i and j.

If $u^*(x)$ is the "true" vector of weights at the feasible solution x (i.e. is collinear with $\nabla U(f(x))$), then this vector is normal to the tangent (represented by the *indifference trade-offs) to the* indifference curve at the point x. In estimating the vector d_m, the DM is trying, as far as is possible, to fulfill the condition that

$$d_{mi}u_i^*(x) + d_{mj}u_j^*(x) = 0$$

$$\text{or} \quad d_m . u^*(x) = 0; \qquad \forall\, m = 1,\ldots,p. \tag{3.3}$$

The indifference trade-off matrix D, which is $p \times r$, is then constructed from the p row vectors d_m as follows:

$$D \triangleq \begin{pmatrix} d_1 \\ \cdot \\ \cdot \\ d_p \end{pmatrix} \tag{3.4}$$

where $p \geqslant r-1$ *and* r *is* the number of objective functions.

Since we are attempting to find a local estimate of the slope of the indifference curve at some point x, the values d_{mi} and d_{mj} should not be too large, but should be large enough for the DM to make a meaningful distinction between A and B.

3.2 THE INQUIRY PATTERN

A $p \times r$ matrix P, corresponding to D, can be constructed as follows:

$$\begin{aligned} p_{mi} &= 1 \quad \text{if} \quad d_{mi} > 0 \\ &= 0 \quad \text{if} \quad d_{mi} = 0 \\ &= -1 \quad \text{if} \quad d_{mi} < 0 \end{aligned} \tag{3.5}$$

where $1 \leqslant m \leqslant p;\ 1 \leqslant i \leqslant r;\ p \geqslant r-1$.

From the construction of D, given in (3.2) and (3.4), each row of P has exactly two nonzero elements: a 1 and a -1.

LEMMA 3.1 Rank $(P) \leqslant r - 1$.

Proof:

Let $P = (p_1, p_2, \ldots, p_r)$

where p_i is a column vector. Then

$$p_1 + p_2 + \ldots + p_r = 0$$

since each row of P contains a 1, -1 and $r-2$ zeros. □

The m^{th} row of the inquiry pattern P contains the information of which two objectives have been compared at the m^{th} pairwise comparison. We impose the *following restrictions on* P, and hence on D, to ensure that the vector of weights which we are trying to estimate is uniquely defined (subject to scaling) for all the objectives.

RESTRICTION 3.2 Each objective is compared to some other objective at *least once*.

i.e. $\bigcup_{m=1}^{p} J_m = \{ 1, \ldots, r \}$.

RESTRICTION 3.3 Any two objectives *should be compared to each other at* most once. i.e. if $J_m = J_n$, then $m = n$.

RESTRICTION 3.4 There are no groups of disjoint subsets of comparisons. i.e. there exists no set $I \subset \{ 1,\ldots,p \}$ such *that*

$$(\bigcup_{m \in I} J_m) \cap (\bigcup_{n \in \{1,\ldots,p\}-I} J_n) = \phi$$

unless $I = \{ 1,\ldots,p \}$.

The first two restrictions ensure that information about each objective is given and that this information is not repeated. The last restriction guarantees that the vector of weights, which is computed from the indifference matrix D, will be uniquely defined (subject only to *scaling*). This property will be proved later on. Restrictions 3.2 - 3.4 have the following consequence on the inquiry pattern.

LEMMA 3.5 If the inquiry pattern P satisfies Restrictions 3.2 - 3.4, then P *has rank* r-1.

Proof: From Lemma 3.1 we have that P has rank $\leq$ r-1.

Restriction 3.2 ensures that no column vectors of P will consist of zeros only. Assume that P has rank $s < r-1$. Reorder the columns of P so that the s linearly independent columns are numbered from 1,..,s. Let $I_1 = \{ 1,\ldots,s+1 \}$ and $I_2 = \{ s+2,\ldots,r \}$. Since the s+1 columns of P which are indexed in I_1 are linearly dependent, *there exists* $\alpha \in R^{s+1}$, $\alpha \neq 0$, such that

$$\sum_{i \in I_1} \alpha_i p_i = 0$$

where p_i is the i^{th} column of the reordered P. By Restriction 3.4, one of the objectives, say $j \in I_1$, is compared to at least one of the objectives indexed in the set I_2 at the m^{th} $(1 \leq m \leq p)$ pairwise comparison. Suppose that objective $j \in I_1$ is compared to objective $k \in I_2$. Then, in row m there is a nonzero element in the j^{th} column and zeros in the rest of the columns for all $i \in I_1$, $i \neq j$. Therefore, $\alpha_j = 0$. Delete j from I_1 and include it in I_2. Applying the above procedure repeatedly, we eventually get that $I_1 = \phi$ (by Restriction 3.4) and that $\alpha_i = 0$ for $i = 1,\ldots,s+1$. This contradiction implies that $s = r-1$. $\square$

3.3 THE CONSISTENT DECISION MAKER

Consider an indifference matrix D as defined in (3.2) (3.4) with a corresponding inquiry pattern P as defined in (3.5).

DEFINITION 3.6 A DM is called *consistent* if, for some (not any) set of scaling factors $w_m > 0$ $(1 \leq m \leq p)$, there exists a vector $a \in R^r$ such that (c.f. (2.4) with a and w interchanged)

$$P \begin{pmatrix} a_1 & & 0 \\ & \ddots & \\ 0 & & a_r \end{pmatrix} = \begin{pmatrix} w_1 & & 0 \\ & \ddots & \\ 0 & & w_p \end{pmatrix} D . \tag{3.6}$$

EXAMPLE 3.7 Let $r = 3$ and suppose that the DM is able to compare all 3 *objectives* with each other. i.e. $J_1 = \{1,3\}$, $J_2 = \{2,3\}$ and $J_3 = \{1,2\}$. Then $p = 3$. Suppose that the DM estimates the indifference trade-off matrix D as:

$$\begin{pmatrix} 2 & 0 & -3 \\ 0 & 1 & -2 \\ 4 & -3 & 0 \end{pmatrix}$$

This has the corresponding inquiry pattern P given by:

$$\begin{pmatrix} 1 & 0 & -1 \\ 0 & 1 & -1 \\ 1 & -1 & 0 \end{pmatrix}$$

which has rank 2. Scaling the rows of D so that the nonzero elements in each column are the same in magnitude gives:

$$\begin{pmatrix} 1 & 0 & -1\frac{1}{2} \\ 0 & \frac{3}{4} & -1\frac{1}{2} \\ 1 & -\frac{3}{4} & 0 \end{pmatrix} = \begin{pmatrix} \frac{1}{2} & & \\ & \frac{3}{4} & \\ & & \frac{1}{4} \end{pmatrix} D .$$

Then there exists $a = (1, \frac{3}{4}, 1\frac{1}{2})$ such that

$$P \begin{pmatrix} 1 & & \\ & \frac{3}{4} & \\ & & 1\frac{1}{2} \end{pmatrix} = \begin{pmatrix} \frac{1}{2} & & \\ & \frac{3}{4} & \\ & & \frac{1}{4} \end{pmatrix} D$$

and so the DM is consistent. □

<u>NOTE</u> 3.8

(i) The scaling which was done in the above example can only be achieved if the DM has given no conflicting information: hence the name "consistent".

(ii) If $p = r-1$, then the DM has given the minimum amount of information which is required. This is insufficient for any conflicting information to emerge. The Geoffrion Method (discussed in Section 2.4.1) requires $r-1$ pairwise comparisons and satisfies this definition, as was shown in (2.4). Further pairwise comparisons are made in order to throw up any inaccuracies which are being made by the DM. Any conflicting information is then resolved by the DM before the method proceeds. Kosinger (1981) overcomes this by incorporating and utilizing all the information given by the DM, whether this was conflicting or not.

(iii) A DM being consistent does not imply that the trade-offs (or MRS) are the correct ones. The DM could be consistently "incorrect". □

<u>LEMMA</u> 3.9 Suppose that the DM gives an indifference trade-off matrix D as defined in (3.2) (3.4) with a corresponding inquiry pattern P as defined in (3.5). If P satisfies Restrictions 3.2-3.4 and the DM is consistent, then:

(i) D has rank $r-1$

(ii) there exists $a \in R^r$ satisfying (3.6) such that $a > 0$, and

(iii) there exists $u \in R^r$, with $u_i > 0 \ \forall i$, such that

$$Du = 0. \tag{3.7}$$

Furthermore, u is unique for $\|u\| = 1$.

Proof:

(i) Follows from (3.6) and Lemma 3.5.

(ii) The existence of a is by Definition 3.6 of consistent. Consider the m^{th} row of (3.6). Equating coefficients on the left and right hand sides of the equation gives

$$p_{mi} \, a_i = w_m \, d_{mi}$$

i.e. $$a_i = w_m \, d_{mi} \, / \, p_{mi}.$$

Then a_i is greater than zero because $w_m > 0$ (Definition 3.6) and d_{mi} and p_{mi} have the same sign (from (3.5)). Restriction 3.2 ensures that the above is true for every objective i at some pairwise comparison m. Therefore, $a_i > 0 \ \ \forall i = 1,\ldots,r$.

(iii) From Definition 3.6 there exists $a \in R^r$, corresponding to some scaling vector $w \in R^p$, $w > 0$, such that

$$\begin{pmatrix} w_1 & & \\ & \ddots & \\ & & w_p \end{pmatrix} D = P \begin{pmatrix} a_1 & & \\ & \ddots & \\ & & a_r \end{pmatrix}.$$

Postmultiplying this equation by some nonzero $u \in R^r$ (i.e. $u_i \neq 0$ $\forall$ i) gives:

$$\begin{pmatrix} w_1 & & \\ & \ddots & \\ & & w_p \end{pmatrix} D\,u = P \begin{pmatrix} a_1 & & \\ & \ddots & \\ & & a_r \end{pmatrix} u\,.$$

Define

$$y = \begin{pmatrix} a_1 & & \\ & \ddots & \\ & & a_r \end{pmatrix} u\,,$$

or, $\quad y_i = a_i\, u_i\,;\quad i = 1,\ldots,r.$

Choose $y_i = 1$ for all i. Therefore, setting

$$u_i = 1/a_i > 0;\quad i = 1,\ldots,r; \tag{3.8}$$

we have

$$\begin{pmatrix} w_1 & & \\ & \ddots & \\ & & w_p \end{pmatrix} D\,u = P \begin{pmatrix} a_1u_1 \\ \\ a_ru_r \end{pmatrix}$$

$$= P \begin{pmatrix} 1 \\ \vdots \\ 1 \end{pmatrix} \quad \text{by (3.8)}$$

$$= \sum_{i=1}^{r} p_i = 0$$

where p_i are the columns of P (and each row of P has a -1. and a $+1$).

Since $w_m > 0$ for $m = 1,\ldots,p$, we have that there exists $u > 0$, defined in (3.8), such that $Du = 0$.

Furthermore, the vector $y = \alpha(1,\ldots,1)$ is the only solution to $Py = 0$, where α is any scalar (since P has rank $r-1$). Therefore,

$$u_i = \alpha/a_i > 0 \quad \text{for } i = 1,\ldots,r,$$

will be uniquely determined by the normalization requirement

$$\| u \| = 1. \qquad \square$$

If Restriction 3.4 on P is left out, then the rank of P may be less than $r-1$. In this case $u > 0$ such that $\| u \| = 1$ may not be uniquely determined.

We now show that the definition of a consistent DM, as given in Definition 3.6, is equivalent to showing that the rank of D is $r-1$. To do so, we turn our attention to the properties of D.

<u>LEMMA</u> 3.10 Suppose that the DM has given an indifference matrix D with rank $r-1$, which has a corresponding inquiry pattern P satisfying Restrictions 3.2-3.4. Then there exists $u \in R^r$ with $u > 0$ such that $Du = 0$.

<u>Proof:</u> Since D is $p \times r$ with $p \geq r-1$ and D has rank $r-1$, there exists $u \neq 0$ such that $Du = 0$. Also, $-u$ satifies $Du = 0$. Therefore, we can find at least one component of u, say u_i, which is positive. By Restriction 3.2 objective i is compared to some other objective, say j, at the m^{th} pairwise comparison where $1 \leq m \leq p$. Multiplying the m^{th} row of D into u gives

$$d_{mi}u_i + d_{mj}u_j = 0$$

i.e. $$u_j = - d_{mi}u_i/d_{mj}$$

where $_{mi} \cdot d_{mj} < 0$ and $u_i > 0$. So we have that $u_j > 0$.

By Restriction 3.4, either objective i or objective j (or both) is compared to at least one other objective, say k. Then, as above, we have that $u_k > 0$. Continuing in this way, Restriction 3.4 ensures that we are able to compare all the objectives at some stage and we will be able to construct u such that $u_i > 0$ for all $i = 1,\ldots,r$. □

<u>THEOREM</u> 3.11 Suppose that the DM gives an indifference *trade-off matrix* D with rank r-1, which has corresponding inquiry pattern P satisfying Restrictions 3.2-3.4. Then the DM is consistent. i.e. Definition 3.6 is satisfied.

<u>Proof:</u> By Lemma 3.10 there exists $u > 0$ such that $Du = 0$.

Consider the m^{th} row of D and suppose that objectives i and j have been compared. Then

$$d_{mi}u_i + d_{mj}u_j = 0$$

$$\text{i.e.} \qquad d_{mi}u_i = - d_{mj}u_j, \tag{3.9}$$

where $d_{mi}.d_{mj} < 0$ and $u_i, u_j > 0$. Without loss of generality, take that $d_{mi} > 0$ and $d_{mj} < 0$. For all w_m a scalar, we have that

$$w_m d_{mi} u_i + w_m d_{mj} u_j = 0. \tag{3.10}$$

Consider the matrix

$$\begin{pmatrix} w_1 & & \\ & \ddots & \\ & & w_p \end{pmatrix} D \begin{pmatrix} u_1 & & \\ & \ddots & \\ & & u_r \end{pmatrix}. \tag{3.11}$$

Looking at the m^{th} row we have the row vector

$$(0,\ldots , w_m d_{mi} u_i, \ldots , w_m d_{mj} u_j, \ldots ,0)$$

Let

$$w_m = \frac{1}{d_{mi}u_i} = \frac{-1}{d_{mj}u_j}\ , \quad \text{by (3.9)}, \tag{3.12}$$

which is positive. This choise of w_m still satisfies (3.10). Then, for this choice of w_m, we have that the m^{th} row of the matrix given in (3.11) is

$$(0,\ldots,1,\ldots,-1,\ldots,0),$$

which is the definition of the m^{th} row of P (from (3.5)). Since this holds for all m, $1 \leqslant m \leqslant p$, we have that for w_m given in (3.12),

$$\begin{pmatrix} w_1 & & \\ & \ddots & \\ & & w_p \end{pmatrix} D \begin{pmatrix} u_1 & & \\ & \ddots & \\ & & u_r \end{pmatrix} = P.$$

Therefore, there exists $a \in R^r$, given by

$$a_i = 1/u_i \quad \text{for} \quad i = 1,\ldots,r,$$

such that Definition 3.6 is satisfied. □

We have already seen, in Lemma 3.9 (i), that consistency implies that the rank of D is $r-1$. In the above theorem we proved that the converse is also true. Therefore, we are able to redefine consistency in a more satisfactory way since the following definition should be easier to check.

<u>DEFINITION</u> 3.12 Suppose that the DM gives an indifference matrix D which has a corresponding inquiry pattern P satisfying Restrictions 3.2-3.4. If the rank of D is $r-1$, then the DM is called consistent.

In the proof of Lemma 3.10 we saw that it is possible to construct $u > 0$ satisfying $Du = 0$ when D has rank $r-1$. This construction is formalised in the following algorithm. It should be borne in mind that we are not generating the direction of the "true" *marginal utility vector*, merely an estimate of this direction. How good this estimate is depends on how accurately the DM (who we still assume to be consistent) is giving the indifference trade-offs in the matrix D.

ALGORITHM 3.13

Step 0. Choose some i such that $1 \leq i \leq r$. Set $u_i = 1$.
Set $I = \{ i \}$, $J = \phi$.

Step 1. Select $i \in I$, $i \notin J$.
For all objectives j $(1 \leq j \leq r)$, $j \notin I$, where objective j is compared to objective i at the m^{th} $(1 \leq m \leq p)$ pairwise comparison, with

$$d_{mi}u_i + d_{mj}u_j = 0,$$

set
$$u_j = -d_{mi}u_i/d_{mj}. \tag{3.13}$$

Set $I = I \cup \{j\}$.
Repeat until there exists no $j \notin I$ which is compared to i.

Step 2. If $I = \{ 1,\ldots,r \}$ then stop.
Otherwise set $J = J \cup \{i\}$ and go to 1. □

NOTE 3.14

(i) The set I keeps a record of the u_i's which have been computed, while the set J keeps a record of the objectives (in I) which cannot be compared to any more objectives which are not already in I.

(ii) In (3.13), since $d_{mi} \cdot d_{mj} < 0$ and $u_i > 0$, we have that $u_j > 0$. □

EXAMPLE 3.15 Suppose that we have a problem with 4 objectives and that the DM gives an indifference trade-off matrix (normalised so that the magnitude of the largest element in each row is a 1):

$$D = \begin{bmatrix} 1 & 0 & -0.25 & 0 \\ 0 & 1 & -0.75 & 0 \\ 0 & 0 & 1 & -1 \\ 0 & 1 & 0 & -0.75 \end{bmatrix}$$

Applying Algorithm 3.13 we get:

Step 0. Arbitrarily choose $i = 1$. Set $u_1 = 1$.

Set $I = \{ 1 \}$, $J = \phi$.

Step 1. Compare 1 with 3.

$$u_3 = u_1/0.25 = 4.$$

$I = \{ 1,3 \}$.

No further $j \notin I$ are compared to 1.

Step 2. $J = \{ 1 \}$. Go to 1.

1. i must be set to 3.

Compare 3 with 2.

$$u_2 = 0.75\, u_3 = 3.$$

$I = \{ 1,2,3 \}$.

Compare 3 with 4.

$$u_4 = u_3 = 4$$

$I = \{ 1,2,3,4 \}$.

2. Stop.

We have generated $u = (1,3,4,4)$ which satisfies $Du = 0$. □

3.4 THE IDEAL DECISION MAKER

DEFINITION 3.16 A Decision Maker who is able to assess the indifference trade-offs (or the Marginal Rates of Substitution) (which constitute the indifference trade-off matrix D) correctly at any point $x \in X$, is called *ideal*.

Since an Ideal Decision Maker is able to assess the indifference trade-offs in the matrix D correctly, we have from (3.3) that the marginal utility vector $\nabla U(f(x))$, at the point x, satisfies

$$D \nabla U(f(x)) = 0.$$

In this case, we denote the indifference *trade-off matrix* at the point x by $D^*(x)$. If the corresponding *inquiry pattern* P satisfies Restrictions 3.2-3.4, then *$D^*(x)$ has rank $r-1$*. If $u^*(x)$ is a vector which is *collinear with* $\nabla U(f(x))$, then

$$D^*(x)\ u^*(x) = 0.$$

From the results of the previous section, there is a unique $u^*(x)$ with $u^*(x) > 0$, which satisfies $\|u^*(x)\| = 1$.

Obviously we have no way of knowing if a DM is an Ideal DM; only if he is a Consistent DM. However, the concept provides us with a point of reference.

3.5 THE INCONSISTENT DECISION MAKER

Suppose that we have obtained an indifference trade-off matrix from the DM and that it has a corresponding inquiry pattern which satisfies Restrictions 3.2-3.4. We note that at least $r-1$ pairwise comparisons are needed in order to satisfy Restriction 3.2. If the DM only makes $r-1$ comparisons, then the DM is consistent (Note 3.8 (ii)) and the rank of D is $r-1$. Any additional comparisons which are made cannot reduce the rank of D to less than $r-1$. Therefore, we use the negation of Definition 3.12 to define inconsistency.

DEFINITION 3.17 Suppose that the DM gives an indifference trade-off matrix D with p pairwise comparisons, where $p \geqslant r$, and that the corresponding inquiry pattern P satisfies Restrictions 3.2-3.4. Then the DM is called *inconsistent* if the rank of D is equal to r.

Since D has rank r when the DM is inconsistent, there does not exist a $u \neq 0$ such that $Du = 0$. Therefore, we find a vector which comes "closest" to fulfilling the condition $Du \approx 0$ by solving the problem (c.f. (2.3) for Rosinger)

$$\text{minimize} \quad u^T D^T D u \tag{3.14}$$

$$\text{subject to} \quad u^T u = 1. \tag{3.15}$$

Applying the Kuhn-Tucker Necessary Conditions of Optimality (see Avriel (1976), for example), we have that at the optimal solution to (3.14) (3.15) there exists a scalar Lagrange Multiplier λ such that

$$- D^T D u = \lambda u. \tag{3.16}$$

This is an eigenvalue/ eigenvector problem. Since the matrix $-D^TD$ is real and symmetric, we know that the r eigenvalues and the corresponding eigenvectors are real. Premultiplying (3.16) by u^T and using (3.15) give gives

$$\lambda = \lambda\, u^T u = -\, u^T D^T D\, u = -\|D\, u\|^2 \leq 0.$$

Therefore, the eigenvalúes are all nonpositive. Problem (3.14) (3.15) then becomes the equivalent problem of finding the maximum (nonpositive) eigenvalue for (3.16) and its corresponding eigenvector.

REMARK 3.18 If the DM is consistent, then $\lambda = 0$. This follows from the fact that there exists $u \neq 0$ such that $Du = 0$. In addition, $\lambda = 0$ is not a repeated eigenvalue since the rank of D is $r-1$. λ is then a "measure of the inconsistencies" in the matrix D. i.e. the larger λ is in magnitude, the "worse" are the inconsistencies. A similar measure of consistency (or inconsistency) is used by Saaty (1980). Although the problems he considers are usually hierarchical in structure, the process of evaluating weights (by collecting information from the DM regarding trade-offs and collating this in a $r \times r$ matrix) can be applied to a problem of the form (1.4).

NOTE 3.19 The indifference trade-off information (or the slope of the indifference curve) given by (d_{mi}, d_{mj}) between two objectives i and j at the m^{th} pairwise comparison is the same as $(\alpha d_{mi}, \alpha d_{mj})$, $\alpha \neq 0$. However, these two representations of the same information affects the value of u computed in (3.14) (3.15) and, therefore, (3.16). Define

$$z \triangleq D\, u.$$

Then the objective function (3.14) becomes:

$$\text{minimize} \sum_{n=1}^{p} z_n^2 .$$

At comparison m suppose that objectives i and j are compared. Then

$$d_m = (0,\ldots, d_{mi},\ldots, d_{mj},\ldots, 0)$$

and so

$$z_m = d_{mi}u_i + d_{mj}u_j .$$

The objective function is then

$$\text{minimize} \sum_{\substack{n=1 \\ n\neq m}}^{p} z_n^2 + (d_{mi}u_i + d_{mj}u_j)^2 .$$

However, if

$$d_m = (0,\ldots,\alpha d_{mi},\ldots,\alpha d_{mj},\ldots, 0)$$

then the objective function is

$$\text{minimize} \sum_{\substack{n=1 \\ n\neq m}}^{p} z_n^2 + \alpha^2(d_{mi}u_i + d_{mj}u_j)^2 .$$

As α increases, so u_i and u_j should decrease, for trade-off information which is the same. Therefore, the rows of D need to be scaled or normalised in some uniform manner. We adopt the procedure of dividing each row d_m of the matrix D by d_{mk}, where

$$d_{mk} = \max_{i,j} \{ |d_{mi}| , |d_{mj}| \}. \tag{3.17}$$

After this normalisation, each row of the matrix D will have one element which is a 1, and another nonzero element which is less than zero and greater than or equal to -1. The *inquiry pattern* P should then be adjusted to account for a possible change of signs in each row (although this is not really necessary). □

<u>LEMMA</u> 3.20 If D is an indifference matrix as constructed in (3.4), then the solution u to (3.14) (3.15), and consequently (3.16), can be chosen to satisfy $u \geq 0$.

<u>Proof:</u> Let $z = Du$. Then (3.14) becomes

$$\text{minimize} \sum_{m=1}^{p} z_m^2.$$

Now,

$$z_m = d_{mi_m} u_{i_m} + d_{mj_m} u_{j_m}$$

where i_m and j_m are the two objectives which are compared at the m^{th} pairwise comparison, and

$$d_{mi_m} \cdot d_{mj_m} < 0. \tag{3.18}$$

Therefore,

$$\sum_{m=1}^{p} z_m^2 = \sum_{m=1}^{p} d_{mi_m}^2 u_{i_m}^2 + \sum_{m=1}^{p} d_{mj_m}^2 u_{j_m}^2 + 2 \sum_{m=1}^{p} d_{mi_m} d_{mj_m} u_{i_m} u_{j_m} \tag{3.19}$$

(i) Suppose that the solution to (3.14) (3.15) is $\bar{u} \leq 0$. Then, letting $u = -\bar{u} \geq 0$, we see from (3.19) that the value of the objective function remains unaltered, and so $u \geq 0$ is also optimal.

(ii) Assume that the optimal solution to (3.14) (3.15) has at least one positive and one negative component. Let

$$u_i = \begin{cases} \bar{u}_i & \text{if } \bar{u}_i \geq 0 \\ -\bar{u}_i & \text{if } \bar{u}_i < 0 \end{cases} \qquad \text{for } i = 1,\dots,r. \tag{3.20}$$

Then,

$$\sum_{m=1}^{p} d_{mi_m} d_{mj_m} u_{i_m} u_{j_m} \leq \sum_{m=1}^{p} d_{mi_m} d_{mj_m} \bar{u}_{i_m} \bar{u}_{j_m}$$

in (3.19). This follows from (3.18) and the fact that $u_{i_m} u_{j_m} \geq 0$

$\forall\, m$ by *construction* (3.20); whereas $\bar{u}_{i_m}\bar{u}_{j_m}$ may be negative for some values of m. Therefore, given *that the first* two terms on the right hand side of (3.19) are the same for u and $\bar{u}$, a solution u can be constructed from $\bar{u}$ which is at least as good as $\bar{u}$ and has nonnegative components. □

THEOREM 3.21 If D *is an indifference* trade-off matrix as constructed in (3.4) with a corresponding inquiry pattern F which *satisfies* Restrictions 3.2 - 3.4, then the solution to the maximum eigenvalue problem (3.16) *has a corresponding* eigenvalue $u > 0$.

Proof:

From Lemma 3.20 we have that an optimal solution to (3.14) (3.15) *satisfies* $u \geq 0$. The Kuhn-Tucker necessary conditions of optimality imply that this solution *will be the eigenvector* which is associated with the maximum (nonpositive) eigenvalue of (3.16).

Assume that u has some components which are zero. (It must have *at least one positive* component otherwise it is the null vector which contradicts (3.15).) Define *the sets*

$$I_1 = \{ i \mid u_i > 0 \; ; \; 1 \leq i \leq r \}$$
$$I_2 = \{ i \mid u_i = 0 \; ; \; 1 \leq i \leq r \},$$

where $I_1 \neq \phi$ and $I_2 \neq \phi$ by assumption. Consider the i^{th} row of (3.16), where $i \in I_2$. Then

$$-\,(D^T Du)_i = \lambda u_i = 0.$$
$$\text{i.e.} \qquad D_i^T D\, u = 0,$$

where D_i^T is the i^{th} row of D^T: i.e. the i^{th} column of D. The i^{th} column of D contains all the trade-off information of *objective* i with respect to any other objectives (at least one) which were compared to it. Consider the off-diagonal elements in the i^{th} row of D^TD:

$$D^TD_{i,j} = 0 \qquad \text{if objectives } i \text{ and } j \text{ are } \textit{not compared}$$
$$= d_{mi}d_{mj} < 0 \quad \text{if objectives } i \text{ and } j \text{ are compared} \qquad (3.21)$$

at the m^{th} pairwise comparison; $j \neq i$. Consider some $i \in I_2$. Define the vector

$$y = D_i^T D.$$

Then

$$y.u = D_i^T D u = -\lambda u_i = 0. \qquad (3.22)$$

Now, by Restriction 3.4 there is some $i \in I_2$ and some $j \in I_1$ which are compared at pairwise comparison m. Consider an $i \in I_2$ for which this is true. Then

$$\begin{aligned} 0 &= y.u && \text{by (3.22)} \\ &= \sum_{j\in I_1} y_j u_j + \sum_{j \in I_2} y_j u_j \\ &= \sum_{j\in I_1} y_j u_j && \text{by definition of } I_2 \\ &< 0 && \text{by Restriction 3.4 and (3.21).} \end{aligned}$$

This contradiction implies that $u > 0$. □

<u>REMARK</u> 3.22 In the theorem above we have shown that the eigenvector corresponding to the maximum (nonpositive) eigenvalue of (3.16) has positive *components*. If this eigenvector is normalised so that $u^Tu = 1$, then it is the optimal solution to (3.14) (3.15) and is then our estimate of the direction of the marginal utility vector. □

3.6 CONCLUSION

The main results of this chapter can be summarised as follows:

a) the DM is required to make $p \geqslant r-1$ pairwise comparisons to estimate the MRS between any two objectives - (3.2);

b) this information is used to construct an *indifference trade-off* matrix D (3.4) and the rows of D *are normalised* - (3.17);

c) an inquiry pattern P, *corresponding* to D, is defined - (3.5) - and *Restriction*; 3.2 - 3.4 are checked;

d) the eigenvalue problem

$$-D^T Du = \lambda u$$

is solved for the maximum (nonpositive) eigenvalue and its corresponding eigenvector (which has positive components);

e) this eigenvector is collinear with the estimated vector of marginal utilities;

f) if $\lambda_{max} = 0$, then the DM has been consistent in his responses.

We showed in Note 3.19 that provision for scaling needed to be taken into account when the estimates of the indifference trade-offs are made. We proposed that this be handled by dividing each row of D by the largest element (in magnitude) in the row. i.e. we required $\| d_m \| = 1$, where we used the L_∞ norm. In addition, for inconsistent responses from the DM, we minimized $\| Du \|$, where the norm which was used is the L_2 norm. Further investigations need to be carried out in order to establish whether the norms which we have chosen do in fact give the 'best' estimate of the resulting vector of weights corresponding to the information which was provided by the DM. We note that when the DM is consistent, the direction of u is not affected by the choice of norms.

CHAPTER 4. THE LINEAR UTILITY FUNCTION: AN INTERACTIVE ALGORITHM

Assume that the (*explicitly*) unknown utility function $U(f(x))$, representing the Decision Maker's preference structure over the objective functions, is a linear, weighted sum of the r objective functions. The assumptions which a DM must satisfy in order for a preference structure to be defined over the objective space are given in Assumptions 1.3 and 1.4. The underlying assumption on the DM's preference structure, in order for the utility function to be linear, is given in Assumption 1.15. Problem (1.4) then becomes problem (1.6): viz

$$\begin{aligned} \text{maximize} \quad & \sum_{i=1}^{r} u_i \, f_i(x) \\ \text{subject to} \quad & A\,x = b \\ & x \geqslant 0, \end{aligned}$$

where $x \in R^n$, $b \in R^m$, A is $m \times n$, and

$$f_i(x) = \sum_{j=1}^{n} c_{ij} \, x_j \; ; \qquad i = 1,\ldots,r.$$

Since

$$U(f_1(x),\ldots,f_r(x)) = \sum_{i=1}^{r} u_i \, f_i(x),$$

we have that

$$\frac{\partial U(f(x))}{\partial f_i} = u_i \quad ; \qquad i = 1,\ldots,r \tag{4.1}$$

where u_i is a positive, constant scalar.

An Ideal DM would be able to provide us with the correct *indifference* trade-offs (or marginal rates of substitution) between pairs of objectives. These constitute the true *indifference* trade-off matrix D^* from which a vector u^*, which is collinear with $\nabla U(f(x))$ can be computed. In this case, the weights u^* can be used in the above linear programming problem and the DM's most preferred solution can be found directly. In general, however, the DM will provide estimates of the indifference trade-offs. Consequently, the resulting set of weights u will only give an estimate of the direction of u^* and so the linear programming problem needs to be incorporated into an interactive algorithm in order to find the DM's most preferred solution.

In this chapter we propose an interactive algorithm to solve problem (1.4), where the underlying utility function is assumed to be linear. We then propose assumptions which we require the DM to satisfy (some of these are implicit) in order to prove convergence of the algorithm to the DM's most preferred solution. Finally, an example is giver to illustrate the use of the algorithm and the assumptions.

4.1 THE SIMPLEX TABLEAU

Suppose that we have a basic feasible solution x_B associated with a basis B. Then the simplex tableau, associated with a multiple objective linear programming problem, looks as follows:

			c_{11}	c_{12}	$c_{1\ell}$	c_{1n}
			c_{r1}	c_{r2}	$c_{r\ell}$	c_{rn}
c_B	Basis	x_B	x_1	x_2	x_ℓ	x_n
			y_{11}	y_{12}	$y_{1\ell}$	y_{1n}
			y_{m1}	y_{m2}	$y_{m\ell}$	y_{mn}
	f_1		z_{11}	z_{12}	$z_{1\ell}$	z_{1n}
	f_r		z_{r1}	z_{r2}	$z_{r\ell}$	z_{rn}

If x_B is associated with the basis B, then

$$x_B = B^{-1}b$$

$$y_\ell = B^{-1}a_\ell , \qquad \ell = 1,\ldots,r,$$

where a_ℓ are the n columns of the constraint matrix A.
The vector of trade-offs for the objectives if x_ℓ is brought into the basis at the next iteration, is defined to be

$$z_\ell = c_B B^{-1} a_\ell - c_\ell$$

where c_B is the matrix of coefficients of the objective functions associated with vectors in the basis B. c_ℓ is the vector of coefficients of the r objective functions associated with the variable x_ℓ.

4.2 AN INTERACTIVE SIMPLEX-BASED ALGORITHM

ALGORITHM 4.1

Step 0. Choose equal weights. i.e. set $u_i^0 = 1$ for $i = 1,\ldots,r$.
Set $k = 1$.

Step 1. Using the weights u^{k-1}, solve the linear programming problem

$$\text{maximize} \quad \sum_{i=1}^{r} u_i^{k-1} f_i(x) \tag{4.2}$$

$$\text{subject to} \qquad Ax = b; \quad x \geqslant 0. \tag{4.3}$$

Call the solution x^k and the trade-offs associated with the non-basic variables x_ℓ, z_ℓ^k.

Step 2. Present x^k and $f(x^k) = (f_1(x^k),\ldots,f_r(x^k))$ to the Decision Maker. Obtain the indifference trade-off matrix D^k from the DM. Ensure that the corresponding inquiry pattern P^k satisfies Restrictions 3.2 - 3.4.

Step 3. Calculate the estimated set of weights u^k by solving the eigenvalue problem

$$-(D^k)^T D^k u = \lambda u.$$

u^k corresponds to the maximum eigenvalue (which are all nonpositive).

Step 4. Calculate $\quad u^k.z_\ell^k \qquad \forall \ \ell$ non-basic.

If $\quad u^k.z_\ell^k \geqslant 0 \quad \forall \ \ell$ non-basic, then GO TO 5.

Otherwise set $k = k+1$ and GO TO 1.

Step 5. Ask the DM if he wishes to revise D^k.

If YES then GO TO 2. Otherwise STOP. □

REMARK 4.2

(i) In Step 1 of the algorithm, the simplex tableau associated with the nondominated basic feasible solution x^k will have the property

$$u^{k-1}.z_\ell^k \geqslant 0 \qquad \forall\ \ell = 1,\ldots,r \tag{4.4}$$

otherwise the simplex algorithm would not have terminated at the "optimal" solution x^k associated with the weights u^{k-1}.

(ii) The algorithm terminates when the DM provides information which causes the algorithm to remain at the previously generated solution. □

4.3 CONVERGENCE OF THE ALGORITHM

It is a well established result of linear programming (see Hadley (1974), for example) that at least one optimal solution will occur at an *extreme point* (*basic feasible* solution) of the constraint set. If the objective function is known, then the simplex algorithm will find an optimal solution to a linear programming problem in a finite number of iterations (provided that any problems arising from the possibility of degeneracy are overcome).

To ensure that Algorithm 4.1 converges in a finite number of iterations to the DM's most preferred solution, we need to

a) assume that the DM does not cause the algorithm to cycle, and
b) assume that the DM "recognises" his most preferred solution when it is presented to him.

In practice, these present no real problems. It is easy enough to realise when cycling (used in this context to mean that the algorithm is generating new points for the DM's consideration which have been presented to him at earlier iterations) is happening. If the DM gives us information which causes *his most preferred solution* to be discarded (this could well happen if this solution is generated after only a few iterations of the algorithm), information which he gives in later iterations, as he "learns" more about the problem, should cause the algorithm to return to this solution. In order to prove theoretical convergence of the algorithm, however, we need to impose assumptions which exclude the possibilities mentioned above from arising.

At iteration k of Algorithm 4.1 we have a nondominated, extreme solution x^k with associated trade-offs z_ℓ^k; where ℓ refers to the non-basic variables x_ℓ. From (4.4), we have

$$u^{k-1}.z_\ell^k \geq 0 \qquad \forall \ell \text{ non-basic; } 1 \leq \ell \leq n.$$

Given x^k and $f(x^k)$, the DM estimates D^k, from which u^k is calculated. We assume that

$$u^k.z_\ell^k \geq 0 \qquad \forall \ell = 1,\ldots,n,$$

only if x^k is "optimal". Otherwise, there exists ℓ, $1 \leq \ell \leq n$, such that

$$u^k.z_\ell^k < 0$$

and the algorithm returns to Step 1. When this does happen, we retain the vector of trade-offs which gives the initial direction of steepest ascent with respect to the weights u^k. At iteration $k > 1$, define the set

$$Z^k \triangleq \{ z_{q_j}^j \mid u^j.z_{q_j}^j = \min_{1 \leq \ell \leq n} (u^j.z_\ell^j) \ : \ j = 1,\ldots,k-1 \}.$$

Define the following sets of "allowable" values for u at iteration k

$$S^1 \triangleq \{ u \in R^r \mid u > 0 \}, \qquad k = 1; \qquad (4.5)$$

$$S^k \triangleq \{ u \in R^r \mid u > 0 \text{ and } u.z < 0 \ \ \forall z \in Z^k \},\ k > 1. \qquad (4.6)$$

We now make the following assumptions:

ASSUMPTION 4.3 The true vector of weights u^* is contained in the set S^k for $k = 1,2,\ldots\ldots$.

This assumption implies that once the DM has given information which causes x^{k-1} to be discarded, then x^{k-1} cannot be a most preferred extreme point since there exists an ℓ, $1 \leq \ell \leq n$, such that

$$u^*.z_\ell^{k-1} < 0.$$

In view of this, we make the following assumption which implies a form of weak consistency on the responses from the DM with respect to the estimates of u^*.

<u>ASSUMPTION</u> 4.4 At iteration k, the estimated weights u^k belong to the set S^k, for $k = 1,\ldots$.

This assumption, which is checkable, is not too restrictive. Given that the (implicit) utility function is assumed to be linear, u^* is independent of the extreme points being presented to the DM for consideration. All we require, is that the DM does not give information which varies "markedly" with previous *information. The above assumption,* merely formalises the requirement that a vector of trade-offs which is perceived (indirectly) by the DM as being the most promising at some efficient solution, does not become undesirable at a later stage.

<u>REMARK</u> 4.5 At this stage, we recall that the DM is not providing estimates of u^* directly, but is estimating the indifference trade-offs in the matrix D^k. The matrix D^k is an approximation to the true indifference trade-off matrix D^*, corresponding to the inquiry pattern P^k. For the moment, however, we deal with the vector u^k which is generated by D^k. It should be pointed out, that if estimates of u^* are being obtained in some other way, then these assumptions *can be applied directly.* □

<u>LEMMA</u> 4.6 Suppose that the estimates u^k of u^* satisfy Assumption 4.4 at each iteration of *Algorithm 4.1. Then, the algorithm does not* cycle. i.e. at each iteration $k > 1$,

$$x^k \neq x^j \quad \text{for all } j = 1,\ldots,k-1.$$

<u>Proof:</u> Assume that the algorithm does cycle. i.e. that $x^k = x^j$ for some j; $1 \leq j \leq k-1$; $k > 1$. Then

$$z_\ell^k = z_\ell^j \quad \forall\ \ell = 1,\ldots,n \tag{4.7}$$

for some j; $1 \leq j \leq k-1$.

Now, $u^{k-1} \in S^{k-1}$ by Assumption 4.4. Therefore,

$$u^{k-1}.z_{q_j}^{j} < 0 \quad \forall\ j = 1,\ldots,k-1. \tag{4.8}$$

This follows from (4.6) and the fact that the algorithm did not terminate at x^{k-1}. i.e. there existed ℓ, $1 \leq \ell \leq n$, such that

$$u^{k-1}.z_\ell^{k-1} < 0$$

in Step 4. From (4.7), there exists a j, $1 \leq j \leq k-1$, such that

$$u^{k-1}.z_\ell^k = u^{k-1}.z_\ell^j \quad \forall\ \ell = 1,\ldots,n;$$

and so

$$u^{k-1}.z_\ell^k < 0$$

for some ℓ, $1 \leq \ell \leq n$, by (4.8). From Remark 4.2 (i), the Simplex Algorithm will not terminate with the solution x^k, but will find some other point satisfying (4.4). □

<u>THEOREM</u> 4.7 Suppose that the estimates u^k of u^* satisfy Assumption 4.4 at each iteration of Algorithm 4.1. Then the algorithm terminates after a finite number of iterations.

<u>Proof</u>: From Lemma 4.6, we have that we can never present the same solution to the DM at any two (or more) iterations. Since the Simplex Algorithm only gives basic feasible solutions, and there are only a finite number of these, none of which can be repeated, the result follows. □

The above result only guarantees finite termination of the algorithm; it does not ensure termination at the most preferred extreme solution. In order to ensure this, we need to make the following assumption:

ASSUMPTION 4.8   Algorithm 4.1 terminates at a nondominated, extreme point x^k only if

$$u^* . z_\ell^k \geq 0 \qquad \forall\ \ell = 1,\ldots,n.$$

In the above assumption, we are assuming that the DM "recognises" his most preferred solution as well as all other "less desirable" solutions.

THEOREM 4.9   Suppose that Assumptions 4.3 and 4.8 are satisfied and that the estimates u^k of u^* satisfy Assumption 4.4 at each iteration of Algorithm 4.1. Then the algorithm will terminate after a finite number of iterations with a solution x^k iff x^k is the DM's most preferred solution.

Proof: Suppose that the algorithm terminates at x^k. Then Assumption 4.8 requires this solution to be "optimal".
On the other hand, suppose that x^k is the most preferred solution. Finite termination follows from Theorem 4.7, and Assumption 4.8 implies that the final solution must be optimal. In addition, if the DM is presented with x^k at iteration k, Assumption 4.3 implies that the DM will not give information which causes the algorithm to leave x^k; otherwise $u^* \in S^{k+1}$, which is not true. □

In the next section we illustrate Algorithm 4.1, as well as the assumptions, by means of an example.

4.4 AN EXAMPLE

Consider the problem given by Zionts and Wallenius (1976).

$$\begin{aligned} \text{maximize} \quad f_1 &= 3x_1 + x_2 + 2x_3 + x_4 \\ f_2 &= x_1 - x_2 + 2x_3 + 4x_4 \\ f_3 &= -x_1 + 5x_2 + x_3 + 2x_4 \end{aligned}$$

subject to

$$\begin{aligned} 2x_1 + x_2 + 4x_3 + 3x_4 &\leq 60 \\ 3x_1 + 4x_2 + x_3 + 2x_4 &\leq 60 \end{aligned}$$

$$x_1, x_2, x_3, x_4 \geq 0.$$

Suppose that the utility function is linear:

i.e. $\quad U(f(x)) = u_1 f_1 + u_2 f_2 + u_3 f_3.$

Zionts and Wallenius assume that the true direction of the marginal utility vector is given by the set of weights $u^* = (\ 0{,}58;\ 0{,}21;\ 0{,}21\ )$ which has associated optimal solution $\bar{x} = (\ 12;\ 0;\ 0;\ 12\ )$.

The Algorithm

Initially choose $u^0 = (\ 1;\ 1;\ 1\ )$. This leads to the simplex tableau (optimal for these weights):

Basis	x_B	x_1	x_2	x_3	x_4	x_5	x_6
4	18	0,5		1,5	1	0,4	-0,1
2	6	0,5	1	-0,5		-0,2	0,3
f_1	24	-2		-1		0,2	0,2
f_2	66	0,5		4,5		1,8	-0,7
f_3	66	4,5		-0,5		-0,2	1,3

i.e. $x^1 = (0; 6; 0; 18)$.

The point x^1 and $f(x^1) = (24; 66; 66)$ are presented to the DM. Since there are three objectives, the DM is required to make at least 2 pairwise comparisons. Suppose that he chooses to compare objectives 1 and 3 and objectives 2 and 3.

i.e. $J_1 = \{ 1,3 \}$, $J_2 = \{ 2,3 \}$.

Then $p = 2$, and so the DM will be consistent whatever trade-off information he gives. This information must not result in an estimate u^1 of u^* which has the property that

$$u^1 . z_\ell^1 \geq 0 \quad \text{for } \ell = 1,\ldots,6.$$

This follows by Assumption 4.8, because

$$u^* . z_1^1 < 0$$

where $z_1^1 = (-2; 0,5; 4,5)$. Suppose that the DM estimates the indifference trade-off matrix as

$$D^1 = \begin{pmatrix} 1 & 0 & -4 \\ 0 & 1 & -2 \end{pmatrix}.$$

Using (3.17), this is normalised to

$$D^1 = \begin{pmatrix} -0,25 & 0 & 1 \\ 0 & -0,5 & 1 \end{pmatrix}$$

which has corresponding inquiry pattern

$$P^1 = \begin{pmatrix} -1 & 0 & 1 \\ 0 & -1 & 1 \end{pmatrix}.$$

p^1 satisfies Restrictions 3.2 – 3.4. Using Algorithm 3.13 to compute u^1 we get

$$u^1 = (1;\ 0,5;\ 0,25) \in S^1.$$

For this set of weights, only $u^1.z_1^1 < 0$, and so x_1 will enter the basis at the first iteration. Then

$$Z^2 = \{ z_1^1 \},$$

and Assumption 4.3 is satisfied. The final simplex tableau (optimal for these weights) is:

Basis	x_B	x_1	x_2	x_3	x_4	x_5	x_6
4	12		-1	2	1	0,6	-0,4
1	12	1	2	-1		-0,4	0,6
f_1	48		4	-3		-0,6	1,4
f_2	60		-1	5		2	-1
f_3	12		-9	4		1,6	-1,4

i.e. $x^2 = (12;\ 0;\ 0;\ 12)$.

Assumption 4.8 requires that we stop here, since

$$u^*.z_\ell^2 \geq 0 \qquad \text{for } \ell = 1,\ldots,6.$$

<u>NOTE</u> 4.10

(i) If the DM chooses u^1 such that

$$u^1.z_3^1 < 0 \quad \text{and} \quad u^1.z_3^1 < u^1.z_1^1 \text{ , say,}$$

then $Z^2 = \{ z_3^1 \}$, which contradicts Assumption 4.3 (since $u^*.z_3^1 > 0$).

(ii) u^2 must be chosen so that Assumptions 4.4 and 4.8 are satisfied.

i.e. $u^2.z_1^1 < 0$

so that $u^2 \in Z^2$ (Assumption 4.4), and

$$u^2.z_\ell^2 \geqslant 0 \qquad \text{for } \ell = 1,\ldots,6. \qquad \square$$

4.5 CONCLUSION

In this chapter we have proposed an interactive algorithm to solve problem (1.4) when the (explicitly) unknown *utility function is assumed* to be linear. In practical implementations of the algorithm, as the DM is 'exposed' to the ramifications of the problem, a DM should (eventually?) find his most preferred solution. There is certainly nothing new in this, as it is simply linear programming with different weightings being used on the objective functions to give a single objective function.

In order to prove convergence of the algorithm to the most preferred solution, however, we need to impose *certain (fairly mild) assumptions* on the responses which are made by the DM. Assumption 4.3 is less severe than the underlying assumptions of Zionts and Wallenius (1976), although the process of obtaining the responses is generally considered to be more difficult (for the DM). Zionts and Wallenius require the DM to indicate desirable and undesirable trade-offs at an extreme point and these responses are incorporated into constraints (2.6) (2.7). These constraints are then used to find a new set of weights *computationally*. The underlying assumption is that these responses are all accurate, and that the true set of weights u^* satisfies the constraint set. We require the DM to recognise (implicitly) a desirable trade-off which is then incorporated into constraint set (4.6), which is considerably smaller than that of (2.6) (2.7). The Zionts - Wallenius Method generates a new set of weights which satisfies a more severe form of Assumption 4.4; whereas we require the DM to give responses which satisfy *this assumption*. *Finally*, at the DM's most preferred extreme point, Zionts and Wallenius require the DM to recognise that all the trade-offs are undesirable; whereas we require this implicitly from the DM's responses.

In view of the above, Algorithm 4.1 could be modified to generate all the nondominated extreme solutions about the current solution (as Zionts and Wallenius do) and to ask the DM to find the best solution (at this stage), place the corresponding trade-off vector into (4.6) and generate a set of weights computationally. This could be weakened further by requiring any preferred solution to be identified. The rate of convergence, however, to the most preferred solution is bound to be poor when compared to the Zionts Wallenius' approach, since considerably less information is utilized when constructing the constraint set for the weights. Further research needs to be carried out in order to determine how the rate of convergence is affected as more information is obtained from the DM and to discover the trade-off (to the DM) between the amount of information which is required at an interaction and the number of interactions.

Although Wallenius (1975) found that DM's have difficulty in estimating the MRS, our assumptions require only 'rough' estimates (with a measure of consistency) of the weights in order for the method to converge. In addition, we enable the DM to give conflicting information (to a certain extent). It is expected, that as these estimates become more accurate, so the rate of convergence should improve, since an Ideal DM will provide information which leads to the most preferred extreme point in one interaction. In the ideal case, the Zionts Wallenius approach is unlikely to converge in one iteration in general, since the weights are computed numerically from (2.6) (2.7).

CHAPTER 5. ERROR BOUNDS ON THE ESTIMATED VECTOR OF WEIGHTS

In this chapter we are concerned with establishing bounds on the errors made by the Decision Maker in the estimates of the direction of the true set of weights u^* (which is collinear with the marginal utility vector $\nabla U(f(x))$). These estimates are used in the linear programming problem

$$\begin{aligned} \text{maximize} \quad & \sum_{i=1}^{r} u_i f_i(x) \\ \text{subject to} \quad & Ax = b, \\ & x \geq 0, \end{aligned}$$

in the interactive algorithm of Section 4.2.

The assumptions which were made in Chapter 4 (when the explicitly unknown utility function is assumed to be linear), in order to guarantee convergence of Algorithm 4.1, are given in terms of the estimated vector of weights u. We first establish bounds on the magnitude of the errors made in u when approximating u^* (i.e. bounds on $\| u - u^* \|$) so that Assumptions 4.4 and 4.8 are satisfied. Bounds are also established on the estimates u, made from iteration to iteration, so that Assumption 4.4 is satisfied. New assumptions are then proposed in terms of these bounds (which imply satisfaction of Assumptions 4.3, 4.4 and 4.8). These are illustrated by means of a simple example. The assumptions are then re-examined and less restrictive assumptions are then proposed.

5.1 BOUNDS ON THE APPROXIMATE VECTOR OF WEIGHTS

In order to establish convergence of Algorithm 4.1 and the algorithm which is proposed in Chapter 7 to the Decision Maker's most preferred solution, we need to determine error bounds between the true vector of weights u^* (which gives the direction of the marginal utility vector $\nabla U(f(x))$) and the approximate vector of weights u^k which is obtained from the DM at iteration k.

THEOREM 5.1 Suppose that

$$a.z \geq \Delta \geq 0$$

where $a \in R^r$, $z \in R^r$. If $b \in R^r$ satisfies

$$a_i - \frac{\Delta}{(\sum_{j=1}^{r} |z_j|)} \leq b_i \leq a_i + \frac{\Delta}{(\sum_{j=1}^{r} |z_j|)} \; ; \qquad i = 1, \ldots, r;$$

then $\quad b.z \geq 0.$

Proof:

Define

$$\phi = \frac{\Delta}{\sum_{j=1}^{r} |z_j|} .$$

Then

$$b.z = \sum_{i=1}^{r} b_i z_i$$

$$= \sum_{z_i \geq 0} b_i z_i + \sum_{z_i < 0} b_i z_i$$

$$\geq \sum_{z_i \geq 0} (a_i - \phi) z_i + \sum_{z_i < 0} (a_i + \phi) z_i$$

$$= \sum_{i=1}^{r} a_i z_i - \phi \sum_{z_i \geq 0} z_i + \phi \sum_{z_i < 0} z_i$$

$$\geq \Delta - \phi \sum_{i=1}^{r} |z_i| = 0. \qquad \square$$

<u>COROLLARY</u> 5.2 *Suppose that*

$$a.z \leq \Delta < 0$$

where $a \in R^r$, $z \in R^r$. If $b \in R^r$ satisfies

$$a_i - \phi \leq b_i \leq a_i + \phi \; ; \qquad i = 1,\ldots,r,$$

where

$$0 < \phi + \varepsilon = \frac{-\Delta}{\sum_{j=1}^{r} |z_j|} ,$$

ε an arbitrary small, positive number,

then $\qquad b.z < 0.$

<u>Proof:</u> The proof is similar to the previous proof. $\square$

We now use these two results to re-examine the assumptions which were made in Chapter 4 to give finite convergence of Algorithm 4.1 to the DM's most preferred solution (assuming a linear utility function). We first recall the following from Chapter 4. At each iteration $j = 1,\ldots,k-1$ $(k > 1)$ we retained the vector of trade-offs which gave the direction of steepest ascent at the start of the simplex algorithm with respect to the weights u^j. Define this set of vectors at iteration k, $k > 1$, by

$$Z^k \triangleq \{ z_{q_j}^j \mid u^j.z_{q_j}^j = \min_{1 \leq \ell \leq n} (u^j.z_\ell^j) : j = 1,\ldots,k-1 \}.$$

The set of permissible values for u^k at iteration k is given by:

$$S^1 \triangleq \{ u \in R^r \mid u > 0 \} \qquad \text{for } k = 1$$

$$S^k \triangleq \{ u \in R^r \mid u > 0 \; ; \; u.z < 0 \;\; \forall z \in Z^k \} \quad \text{for } k > 1. \tag{5.1}$$

We introduce the following definition:

$$\phi^j \triangleq -u^j.z_{q_j}^{\;j} / \| z_{q_j}^{\;j} \| \tag{5.2}$$

where $\| a \| \triangleq \sum_{i=1}^{r} |a_i|$.

<u>THEOREM</u> 5.3 Suppose that a vector $u \in R^r$ satisfies

$$| \, u_i - u_i^j \, | < \phi^j - \varepsilon, \qquad i = 1,\ldots,r;$$

where u^j *is the estimated vector of weights used in iteration* j of Algorithm 4.1, ϕ^j is defined in (5.2) and ε is an arbitrarily small, positive number. If $u^j.z_{q_j}^{\;j} < 0$, then $u.z_{q_j}^{\;j} < 0$.

<u>Proof:</u>

$$| \, u_i - u_i^j \, | < \phi^j - \varepsilon \qquad \forall \, i = 1,\ldots,r$$

implies that

$$\varepsilon - \phi^j + u_i^j < u_i < - \varepsilon + \phi^j + u_i^j \qquad \forall \, i = 1,\ldots,r.$$

Letting

$$\phi = - u^j.z_{q_j}^{\;j} / \| z_{q_j}^{\;j} \| - \varepsilon = \phi^j - \varepsilon$$

in Corollary 5.2 gives the result. □

We are now in a position to replace Assumptions 4.3 and 4.4 by the *following two assumptions respectively:*

ASSUMPTION 5.4 At iteration k of Algorithm 4.1, the estimated vector of weights $u^k > 0$ satisfies the inequalities

$$| u_i^* - u_i^k | \leq \phi^k - \varepsilon; \quad i = 1,\ldots,r;$$

where ϕ^k is defined in (5.2), ε is an arbitrarily small, positive number and u^* is the true vector of weights (collinear with $\nabla U(f(x))$).

ASSUMPTION 5.5 At iteration $k > 1$ of Algorithm 4.1, the estimated vector of weights $u^k > 0$ satisfies the inequalities

$$| u_i^k - u_i^j | \leq \phi^j - \varepsilon; \ i = 1,\ldots,r; \quad \forall\, j = 1,\ldots,k-1;$$

where ϕ^j is defined in (5.2), ε is an arbitrarily small, positive number and u^j are the estimates of the weights made at the previous *iterations* $j = 1,\ldots,k-1$.

Using Theorem 5.3, with k replacing j, satisfaction of Assumption 5.4 by the weights u^k at every iteration k of Algorithm 4.1 implies that $u^* \in S^k$. This assumption cannot be checked as we do not know u^*. The values for ϕ^k, however, can be computed and the range on each component of u^* can be determined. Assumption 5.5 implies that Assumption 4.4 is satisfied (by Theorem 5.3). This assumption can be checked, since all the values are known. The above conditions were imposed so as to ensure that *the true set of weights* u^* were not "cut out" of the set S^k and that the DM is weakly consistent with his estimates. The last condition also removed the possibility of the algorithm cycling. This gave finite termination of the algorithm.

In order to ensure that the algorithm terminates at the DM's most preferred solution, we use Theorem 5.1 to replace Assumption 4.8 by the following assumption. This assumption is also not checkable, since u^* is not known. *It does, however, provide us with a means of determining ranges on the components of* u^* at the final solution.

ASSUMPTION 5.6 Algorithm 4.1 terminates at the nondominated, extreme *point* x^k *only if the estimates of the weights* $u^k > 0$, made at this point, satisfy

$$|u_i^* - u_i^k| \leq \phi_k ; \qquad i = 1,\ldots,r;$$

where

$$\phi_k = \min_{\ell \text{ non basic}} u^k.z_\ell^k / \|z_\ell^k\| \geq 0$$

and u^* is the true set of weights (collinear with $\nabla U(f(x))$).

We have *from Steps* 4 *and* 5 *of* Algorithm 4.1 that the algorithm terminates at x^k when

$$u^k.z_\ell^k \geq 0 \qquad \text{for } \ell = 1,\ldots,n.$$

Since the trade-offs associated with basic variables are zero, we have that

$$u^k.z_\ell^k \geq 0 \qquad \forall\, \ell \text{ nonbasic; } 1 \leq \ell \leq n.$$

If the above assumption holds, then by Theorem 5.1,

$$u^*.z_\ell^k \geq 0 \qquad \forall\, \ell \text{ nonbasic; } 1 \leq \ell \leq n;$$

and x^k is the DM's most preferred solution.

We illustrate the use of these assumptions by means of an example which is given in the next section.

5.2 AN EXAMPLE TO ILLUSTRATE THE COMPUTATION OF THE ERROR BOUNDS ON THE ESTIMATES OF THE WEIGHTS.

NOTE 5.7 The vector of estimated weights, obtained from the DM's indifference matrix D (3.16), is normalised so that $u^T u = 1$ (3.15). In the rest of this chapter we will use the L_∞ norm instead of the L_2 norm. i.e. u is normalised so that

$$\max_{1 \leqslant i \leqslant r} u_i = 1.$$

This change of magnitude does not affect any of the previous analysis as we do not need to know the magnitude of the marginal utility vector: we are interested in its direction. □

Consider the following example which, although relatively simple, will enable us to demonstrate the use of Assumptions 5.4 - 5.6 in the computation of the error bounds.

$$\begin{array}{ll} \text{maximize} & f_1 = x_1 + 2x_2 \\ & f_2 = 2x_1 - x_2 \\ & f_3 = -x_1 + 4x_2 \\ \text{subject to} & 3x_1 + 2x_2 \leqslant 12 \\ & 2x_1 + 3x_2 \leqslant 12 \\ & x_1 \leqslant 3 \\ & x_2 \leqslant 3 \\ & x_1, x_2 \geqslant 0. \end{array}$$

Assume that the (implicit) utility function is linear and is given by

$$U(f_1,f_2,f_3) = u_1^* f_1 + u_2^* f_2 + u_3^* f_3$$

where $u^* = (\ 0{,}5;\ 1;\ 0{,}5\ )$.

Initially set $u^0 = (\ 1;\ 1;\ 1\ )$.

Then the final simplex tableau (optimal for these weights) is:

Basis	x_B	x_1	x_2	x_3	x_4	x_5	x_6
2	3		1				1
5	1,5				-0,5	1	1,5
1	1,5	1			0,5		-1,5
3	1,5			1	-1,5		2,5
f_1	7,5				0,5		0,5
f_2	0				1		-4
f_3	10,5				-0,5		5,5

Therefore, $x^1 = (\ 1{,}5;\ 3\ )$ and $f(x^1) = (\ 7{,}5;\ 0;\ 10{,}5\ )$.

From the tableau, we have

$$z_4^1 = \begin{pmatrix} 0{,}5 \\ 1{,}0 \\ -0{,}5 \end{pmatrix} \ ; \quad z_6^1 = \begin{pmatrix} 0{,}5 \\ -4{,}0 \\ 5{,}5 \end{pmatrix} .$$

We see that $u^*.z_4^1 > 0$ and $u^*.z_6^1 < 0$.

Assumption 4.8 implies that u^1, obtained from the DM, must cause us to discard x^1. Assumption 4.3 implies that $z_6^1 \in Z^2$ (and not z_4^1). This is achieved by any estimate of u^1 which has the properties that

$$u^1.z_6^1 < 0 \quad \text{and} \quad u^1.z_6^1 < u^1.z_4^1 .$$

Suppose that the DM looks at $f(x^1)$ and decides that f_2 needs a larger weight. He overestimates this in specifying the trade offs, yielding an estimated utility vector

$$u^1 = (\ 1/3;\ 1;\ 1/3\ ) \qquad \text{(see Note 5.7).}$$

Then

$$\phi^1 = -\frac{u^1.z_6^1}{\| z_6^1 \|} = \frac{2}{10} = 0{,}2\ .$$

We see that

$$|\ u_i^* - u_i^1\ | < 1/6 \ < 0{,}2 \qquad \text{for} \quad i = 1,2,3,$$

and so Assumption 5.4 is satisfied. Note that if the DM gives estimates of u_1^1 and u_3^1 which decrease with respect to u_2^1 fixed on 1, so ϕ^1 becomes larger and the margin of error can increase.

Using the weights u^1 we get the simplex tableau (optimal for these weights):

Basis	x_B	x_1	x_2	x_3	x_4	x_5	x_6
2	1,5		1	0,5		-1,5	
4	1,5			-1,5	1	2,5	
1	3	1				1	
6	1,5			-0,5		1,5	1
f_1	6			1		-2	
f_2	4,5			-0,5		3,5	
f_3	3			2		-7	

Therefore, $x^2 = (\ 3;\ 1{,}5\ )$ and $f(x^2) = (\ 6;\ 4{,}5;\ 3\ )$.

From the tableau, we have

$$z_3^2 = \begin{pmatrix} 1 \\ -0,5 \\ 2 \end{pmatrix} \; ; \quad z_5^2 = \begin{pmatrix} -2 \\ 3,5 \\ -7 \end{pmatrix} .$$

where $u^*.z_3^2 > 0$ and $u^*.z_5^2 < 0$.

Assumptions 4.3 and 4,8 require that $z_5^2 \in Z^3$ (and not z_3^2). Therefore, the DM must give an estimate of u^2 such that

$$u^2.z_5^2 < 0 \quad \text{and} \quad u^2.z_5^2 < u^2.z_3^2 .$$

In *addition, Assumption* 4.4 requires a measure of consistency from the DM. This is measured in Assumption 5.5, where u^2 must satisfy

$$|\, u_i^2 - u_i^1 \,| < \phi^1 = 0,2 \qquad \text{for} \quad i = 1,2,3, \tag{5.3}$$

where u^2 has been normalised so that $\max\,(\, u_1^2,\, u_2^2,\, u_3^3 \,) = 1$.

Assumption 5.4 requires that

$$|\, u_i^* - u_i^2 \,| < \phi^2 \triangleq - u^2.z_5^2 \,/\, \| z_5^2 \| \quad \text{for} \quad i = 1,2,3. \tag{5.4}$$

Suppose that the DM chooses

$$u^2 = (\, 0,45;\; 1;\; 0,52 \,).$$

Then both (5.3) and (5.4) are satisfied, where $\phi^2 = 0,0832$.

The final simplex tableau (optimal for the weights u^2) is given over the page. From the tableau,

$$x^3 = (\, 2,4;\; 2,4 \,) \quad \text{and} \quad f(x^3) = (\, 7,2;\; 2,4;\; 7,2 \,).$$

We see that $u^*.z_3^3 > 0$ and $u^*.z_4^3 > 0$. From Assumptions 4.3 and 4.8 we require u^3 to satisfy

$$u^3.z_3^3 \geqslant 0 \quad \text{and} \quad u^3.z_4^3 \geqslant 0 \tag{5.5}$$

Basis	x_B	x_1	x_2	x_3	x_4	x_5	x_6
2	2,4		1	-0,4	0,6		
5	0,6			-0,6	0,4	1	
1	2,4	1		0,6	-0,4		
6	0,6			0,4	-0,6		1
f_1	7,2			-0,2	0,8		
f_2	2,4			1,6	-1,4		
f_3	7,2			-2,2	2,8		

Suppose that the DM gives

$$u^3 = (0,45;\ 1;\ 0,48).$$

Then

$$\phi_3 = \min (u^3.z_3^3 /\| z_3^3 \| ,\ u^3.z_4^3/\| z_4^3 \|)$$

$$= \min (0,1135;\ 0,0608)$$

$$= 0,0608.$$

We see that

$$| u_i^* - u_i^3 | \leq 0,05 < 0,0608 \qquad \text{for } i = 1,2,3.$$

Therefore, from Theorem 5.1, we have that (5.5) is satisfied.

5.3 A RE-EVALUATION OF THE ASSUMPTIONS

In the example discussed in the last section, we saw that the requirement that

$$| u_i^* - u_i^k | < (\leqslant) \ \phi^k \qquad \text{for} \quad i = 1,2,3,$$

in order to satisfy Assumption 5.4 (5.6), is fairly restrictive when applied to vectors which have been normalised according to Note 5.7. The errors made in the estimates could be much "larger" if we allowed the maximal element in u^k to vary. A *uniform approach*, however, is desirable, since the estimates u^k are required for use in the algorithm.

If we recognise the fact that $a_q = b_q$ for some q ($1 \leqslant q \leqslant r$) in Theorem 5.1, we find that if we define

$$\phi = \delta \ / \sum_{\substack{j=1 \\ j \neq q}}^{r} |z_j|$$

in the proof, that the result still holds. The same observation applies to Corollary 5.2. We are, therefore, able to replace Assumptions 5.4 and 5.6 by the following two (less restrictive) assumptions respectively.

<u>ASSUMPTION</u> 5.8 At iteration k of Algorithm 4.1, assume that the estimated vector of weights $u^k > 0$, which has been normalised so that

$$u_q^k = \max_{1 \leqslant i \leqslant r} u_i^k = 1,$$

satisfies the inequalities

$$| u_i^* - u_i^k | \leqslant \phi^k - \varepsilon \qquad \text{for} \quad i = 1,\ldots,r;$$

where ε is an arbitrarily small positive number;

$$\phi^j = \max_{1 \leq \ell \leq n} \{ -u^j . z_\ell^j \Big/ \sum_{\substack{i=1 \\ i \neq q}}^{r} |z_{i\ell}^j| \};$$

and u^* is the true (unknown) vector of weights, collinear with $\nabla U(f(x))$, which has been scaled so that $u_q^* = 1$.

ASSUMPTION 5.9 Assume that Algorithm 4.1 terminates at a nondominated extreme point x^k only if the estimated vector of weights $u^k > 0$, which has been normalised so that

$$u_q^k = \max_{1 \leq i \leq r} u_i^k = 1,$$

satisfies the inequalities

$$| u_i^* - u_i^k | \leq \phi_k \qquad \text{for } i = 1, \ldots, r;$$

where

$$\phi_k = \min_{\ell \text{ nonbasic}} u^k . z_\ell^k \Big/ \sum_{\substack{i=1 \\ i \neq q}}^{r} |z_{i\ell}^k| \geq 0$$

and u^* is the true (unknown) vector of weights, collinear with $\nabla U(f(x))$, which has been scaled so that $u_q^* = 1$.

In both of these assumptions, the assumed scaling of u^* is not important as we are interested in the direction of the marginal utility vector. The right hand side limits in the inequalities, however, have been increased (or are as good as) the limits in the corresponding Assumptions 5.4 and 5.6 by the removal of one of the terms in the denominator (unless this was zero). Since the relaxation of the limits in the above two assumptions should generally give the DM greater flexibility in the estimates he makes of u^*, one may be tempted to apply the same relaxation to Assumption 5.5. It is our belief, however, that

Assumption 4.4 (which is implied by Assumption 5.5) should be retained in its present form as it is easily checkable and less restrictive than Assumption 5.5 is. By this we mean that any vector which satisfies Assumption 5.5 must also satisfy Assumption 4.4 (by Theorem 5.3). The converse, however, is not generally true. We now *restate Assumption 4.4*, for completeness, as:

ASSUMPTION 5.10 At iteration k of Algorithm 4.1, assume that the estimated vector of weights $u^k > 0$ is contained in the set S^k, defined *in (5.1), for* $k = 1,2,\ldots$.

If the DM is able to satisfy Assumptions 5.8 - 5.10 at each iteration of Algorithm 4.1, then the algorithm will converge in a finite number of iterations to the DM's most preferred solution (when the utility function is assumed to be linear).

We now demonstrate that Assumptions 5.8 - 5.10 give the DM greater *flexibilty when* estimating the weights, by reconsidering the example of Section 5.2.

EXAMPLE 5.11

Consider the example of Section 5.2.

At the point $x^1 = (1,5;\ 3)$, suppose that the DM gives an estimate

$$u^1 = (0,25;\ 1;\ 0,25).$$

Assume that u^* is of the form $\;\;{}_1\,;\ 1;\ u_2^*)$.

$$\phi^1 = -\,u^1.z_6^1 \,/(|z_{1,6}^1| + |z_{3,6}^1|) = 0,416$$

Confirmation that this is indeed the limit is obtained by choosing

$$u = (0{,}416^{\cdot} + u_1^1;\ 1;\ 0{,}416^{\cdot} + u_2^1)$$
$$= (0{,}6^{\cdot};\ 1;\ 0{,}6^{\cdot}).$$

For this value of u, $u.z_6^1 = 0$.

We see that

$$|\ u_i^* - u_i^1\ | < 0{,}416^{\cdot} \quad \text{for} \quad i = 1,3.$$

Using the weights u^1, *we get the solution* $x^2 = (3;\ 1{,}5)$, which is the same solution as before. At x^2 we require (or assume) that the DM gives an estimate u^2 which satisfies

$$u^2.z_6^1 < 0 \qquad \text{(Assumption 5.10)} \tag{5.6}$$

and

$$|\ u_i^* - u_i^2\ | < \phi^2 = -u^2.z_5^2 /(|z_{1,5}^2| + |z_{3,5}^2|);\ i = 1,3 \tag{5.7}$$

if $u_2^2 = u_2^* = 1$ (by Assumption 5.8).

Suppose that the DM gives

$$u^2 = (0{,}55;\ 1;\ 0{,}67).$$

Then both (5.6) and (5.7) are satisfied, where $\phi^2 = 0{,}254^{\cdot}$.

We see, that if Assumption 5.5 was assumed to be satisfied rather than (5.6), *then*

$$|\ u_3^2 - u_3^1\ | = 0{,}42 > 0{,}416^{\cdot} = \phi^1,$$

and we may conclude, wrongly, that Assumption 4.4 is not *satisfied.*

The solution for the weights u^2 is $x^3 = (2{,}4;\ 2{,}4)$, which is the same solution obtained previously. At x^2, we require the DM to recognise this as his most preferred solution by giving an estimate of u^3 which satisfies (5.5).

Assumption 5.9 requires the DM to give u^3 such that

$$| u_i^* - u_i^3 | \leqslant \min (u^3.z_3^3/2.4 \ , \ u^3.z_4^3/3.6) \tag{5.8}$$

where $u_2^3 = u_2^* = 1$.

Since $u_1^* = u_3^* = 0,5$, the ranges on u_1^3 and u_3^3 in (5.8) will be the same. Letting $u_1^3 = u_3^3 = y$ in (5.8), we see that we need y to lie in the range

$$\frac{-0,8y + 1,4 - 2,8y}{3.6} \leqslant y - 0,5 \leqslant \frac{-0,2y + 1,6 - 2,2y}{2,4}$$

$$\text{i.e.} \qquad 0,4^{\cdot} \leqslant u_i^3 \leqslant 0,583^{\cdot} \qquad \text{for } i = 1,3. \tag{5.9}$$

The DM can make estimates of u_1^3 and u_3^3 in the range given in (5.9) and (5.8) will be satisfied. □

We have shown that estimates of the vector of weights can be less accurate if Assumptions 5.8 - 5.10 are used instead of Assumptions 5.4 - 5.6. We recall, however, that the DM is not giving these weights directly, but the indifference matrix D (from which the weights are calculated). In the next chapter we consider what errors can be made by the DM when he estimates the indifference trade-offs.

5.4 CONCLUSION

In this chapter we have determined bounds on the elements of the estimated vector of weights which are used in *Step 1* of *Algorithm* 4.1, which ensure that the algorithm converges to the DM's most preferred solution. These bounds give a practical way of determining the ranges in which the true vector of weights must lie. During the implementation of the algorithm and at the termination of the algorithm, these ranges can be displayed to the DM. Although the DM is not estimating the weights directly, he may be able to pass some judgement on whether they reflect his feelings about the relative weightings of the objectives or not.

It is our opinion that these bounds are reasonable in practice. (i.e. they reasonably represent the assumptions in Chapter 4 which they have replaced.) The following points, however, still need further investigation.

a) The underlying Assumptions 4.4, 4.5 and 4.8. Other assumptions can be made which guarantee convergence of Algorithm 4.1. More restrictive (and consequently tighter) bounds should speed up convergence. This point was mentioned in Section 4.5.

b) The requirements that

$$| u_i - u_i^* | \leqslant \phi , \qquad i = 1,\ldots,r,$$

where ϕ is a function of u, need to be examined further. Is it possible to determine bounds on $\| u - u^* \|$ which are independent of the estimates u?

c) Re-examine Assumption 5.9 at the final solution x^k. This has the anomaly that if u^k is given such that

$$0 \leqslant u^k \cdot z_\ell^k < \varepsilon \qquad \text{for one nonbasic } \ell,$$

ε arbitrarily small, then we require

$$u_i^k \approx u_i^* \qquad \forall\ i = 1,\ldots,r.$$

Assumption 4.8 merely required u^* to satisfy

$$u^*.z_\ell^k \geqslant 0 \qquad \forall\ \ell = 1,\ldots,n.$$

This set of linear constraints on u^* can be constructed. In addition, constraints $u^* \in S^j$ $(j \leqslant k)$ could be incorporated to restrict the range still further. This is similar to Zionts and Wallenius' (1976) *implied requirements at an optimal solution.* Our reason for considering the constraint set, however, is to determine a uniform bound on $\|u^k - u^*\|$. To do so, suitable objective functions need to be formulated. Assumption 5.8 does not present us with the same practical difficulties since $\mathfrak{z}^k$ is not computed from all the nonbasic trade-offs, only the best one is used.

d) The objective functions mentioned in (c) could be facilitated by making an assumption that the error in the estimated weights is proportional to the (unknown) true weights. i.e. that

$$u_i^k = (1+\varepsilon_i^k)\ u_i^*, \qquad i = 1,\ldots,r, \tag{5.10}$$

where $-\varepsilon^1 \leqslant \varepsilon_i^k \leqslant \varepsilon^2$, $\varepsilon^1, \varepsilon^2 \geqslant 0$. Is this a reasonable assumption? What effects will it have on the other results in this chapter? This also ties up with the point raised in (b) since we now have

$$-\varepsilon^1\ u_i^* \leqslant u_i - u_i^* \leqslant \varepsilon^2\ u_i^* \qquad i = 1,\ldots,r.$$

e) We have chosen the L_∞ norm to examine $\|u - u^*\|$. It is possible that a different choice of norm may be more *useful.* A *different* choice of norm will, however, affect the results of the next chapter.

CHAPTER 6 ERROR BOUNDS ON THE ESTIMATED INDIFFERENCE TRADE-OFFS

In Chapter 5 we showed that if the Decision Maker gives estimates of the true (unknown) set of weights which lie in a prescribed range, then Algorithm 4.1 will converge to the DM's most preferred solution in a finite number of iterations. In Chapter 3, however, we did not require the DM to give estimates of these weights directly, but to give the indifference trade-offs between pairs of objectives. This information is incorporated into the indifference trade-off matrix D and the weights u are then computed from the eigenvalue/ eigenvector problem (3.16).

We have already seen that in Section 5, at iteration k of Algorithm 4.1, we required

$$| u_i^* - u_i^k | \leq \phi^k - \varepsilon \qquad \text{for } i = 1,\ldots,r \tag{6.1}$$

to be satisfied; where u^k is the estimate of the true set of weights u^*, ϕ^k is some prescribed non-negative scalar which is defined in (5.2) and ε is an arbitrary small, positive number. Bounds on the elements of the indifference trade-off matrix need to be established which ensure that (6.1) is satisfied.

Using a standard perturbation approach for eigenvalue problems (see Wilkinson (1965)), we were able to find bounds on the elements of the matrix D which did guarantee that the resulting eigenvector satisfied the bounds of (6.1). On computing these bounds, however, for a reasonably sized problem (7 objectives and 8 pairwise comparisons), these bounds were found to be so small (in relation to ϕ^k) as to be useless for any practical purposes. Therefore, in this Chapter, a different approach is proposed in order to obtain useful bounds on the elements of D.

NOTATION 6.1 We drop the superscript k for convenience sake in the rest of this chapter. The indifference trade-off matrix D^k, obtained from the DM at the solution x^k, and the resulting set of weights u^k are denoted by D and u respectively. D^k has corresponding inquiry pattern P^k, defined in (3.5), which is denoted by P. The true indifference trade-off matrix $D^*(x^k)$, corresponding to P^k, and the true set of weights $u^*(x^k)$ (collinear with $\nabla U(f(x^k))$), are denoted by D^* and u^* respectively. □

We now give two examples to illustrate some of the difficulties which arise when attempting to establish bounds on the errors allowed in the matrix D.

EXAMPLE 6.2 Suppose that we have a problem with 7 objectives and that the DM gives the following indifference trade-off matrix involving 8 sets of pairwise comparisons.

$$\begin{pmatrix} 10 & 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & 3 & 0 & -2 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & -1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & -2 & 1 \\ 0 & 2 & -1 & 0 & 0 & 0 & 0 \\ 5 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 1 & 0 & 0 \end{pmatrix}$$

This matrix has full rank. i.e. the responses from the DM are inconsistent. The matrix is then normalised to (see (3.17)):

$$D = \begin{pmatrix} 1 & 0 & 0 & -0,1 & 0 & 0 & 0 \\ 0 & 1 & 0 & -0,6 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & -1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -0,5 \\ 0 & 1 & -0,5 & 0 & 0 & 0 & 0 \\ 1 & -0,2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 1 & 0 & 0 \end{pmatrix}$$

Solving (3.16) gives the solution (rounded to 5 decimal places), where $u^T u = 1$:

$$u = (\ 0,03566;\ 0,19480;\ 0,34903;\ 0,32327;\ 0,33635;\ 0,35115;\ 0,70571\ ).$$

Adjusting D to be (to 5 decimal places):

$$\bar{D} = \begin{pmatrix} 1 & 0 & 0 & -0,11031 & 0 & 0 & 0 \\ 0 & 1 & 0 & -0,60259 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & -1,03770 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & -0,99396 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & -0,49758 \\ 0 & 1 & -0,55812 & 0 & 0 & 0 & 0 \\ 1 & -0,18306 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1,04046 & 1 & 0 & 0 \end{pmatrix}$$

we have that $\bar{D}u = 0$ (subject only to rounding error), which is a property that we require for a consistent set of responses. Element d_{63} has changed by approximately 12%, while other elements have changed by smaller amounts, and yet the solution to (3.16) is still the same. $\bar{D}$ has been specifically constructed (see later on in Section 6.2) to show that "reasonable" changes in an inconsistent D can result in a set of weights $\bar{u}$ which satisfy $\bar{u} = u$. i.e. that $\phi^k - \varepsilon$ can be set to zero in (6.1).

A more general approach would be expected to have the property that the errors tended to zero as $\phi^k \to 0$. However, as this example illustrates, this may not be appropriate for inconsistent responses. □

<u>EXAMPLE</u> 6.3 Suppose that D^* for a problem with 3 objectives is

$$D^* = \begin{pmatrix} -0,25 & 1 & 0 \\ 0 & 1 & -0,25 \end{pmatrix}$$

where the inquiry pattern is

$$P = \begin{pmatrix} -1 & 1 & 0 \\ 0 & 1 & -1 \end{pmatrix}.$$

This has corresponding u^* (rounded to 4 decimal places):

$$u^* = (\ 0,6963;\ 0,1741;\ 0,6963\ )$$

or $$u^* = (\ 1;\ 0,25;\ 1\ )$$

if we normalise u^* such that $\max_i u_i^* = 1$.

Suppose that we allow the DM to make errors of the order of 20% in the elements d_{11} and d_{23}.

i.e. $-0,3 \leq d_{11} \leq -0,2$

$-0,3 \leq d_{23} \leq -0,2$.

Then, if the DM gives an indifference trade-off matrix

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

this leads to

$$u = (1; 0,3; 1)$$

which may be considered to be a reasonable estimate of u^*.

However, if

$$D = \begin{pmatrix} -0,2 & 1 & 0 \\ 0 & 1 & -0,3 \end{pmatrix}$$

then

$$u = (1; 0,2; 0,6')$$

which is considerably worse. In fact, the matrix

$$D = \begin{pmatrix} -0,5 & 1 & 0 \\ 0 & 1 & -0,5 \end{pmatrix},$$

which is well outside the allowed range, leads to

$$u = (1; 0,5; 1),$$

which is considerably better than the previous case if we are using the criterion $|u_i^* - u_i|$ as a yardstick.

In this example, we see that if both d_{11} and d_{23} are estimated incorrectly in the same direction, then fairly large errors can be made. It is when they are estimated incorrectly in different directions that problems arise. In larger problems (e.g. Example 6.2), these interactions may well be impossible to analyse. This example shows that a consistent approach to any size problem, which allows errors to be made in either direction, will result in "small" bounds being imposed on the elements of D in order to ensure that (6.1) is satisfied. □

NOTE 6.4

(i) After normalisation (3.17) has been performed on each row m of the matrix D $(m = 1,\ldots,p)$, any errors made by the DM when estimating the indifference trade-offs between objectives i and j, are accumulated in $d_{mj} < 0$. All that this means is that after normalisation (3.17) has been performed, any errors made in either, or both, of the original elements at the m^{th} pairwise comparison, will now be found in the negative element.

(ii) Given $d_{mi} = 1$, $0 > d_{mj} \geqslant -1$ in row m of the normalised D, we have that $d^*_{mi} = 1$ and $0 > d^*_{mj}$. We do not assume that $d^*_{mj} \geqslant -1$. It is quite possible that the DM has wrongly estimated the

$$\max \{ |d^*_{mi}| , |d^*_{mj}| \},$$

especially if they are "close". See $\bar{d}_{35}$ of $\bar{D}$ in Example 6.2, for instance. □

6.1 ERROR BOUNDS ON CONSISTENT RESPONSES

Suppose that the DM has given an indifference trade-off matrix D with corresponding enquiry pattern P which satisfies Restrictions 3.2 - 3.4 and that D has rank $r-1$. Then, from Definition 3.12, the DM is consistent. An estimate u of the weights is obtained by solving (3.16) for the eigenvector corresponding to the zero eigenvalue (Remark 3.18), or by using Algorithm 3.13.

At the m^{th} pairwise comparison $(1 \leqslant m \leqslant p)$, objectives i and j $(1 \leqslant i,j \leqslant r;\ i \neq j)$ are compared and

$$d_{mi} = 1, \quad 0 > d_{mj} \geqslant -1$$

after normalisation (3.17). Then, we have

$$u_i + d_{mj}\, u_j = 0 \tag{6.2}$$

and $$u_i^* + d_{mj}^*\, u_j^* = 0 \tag{6.3}$$

where d_{mj}^* is the true value of d_{mj} after normalisation.

<u>NOTE</u> 6.5 Suppose that u^* is scaled so that

$$u_q^* = u_q = \max_{1 \leqslant i \leqslant r} u_i = 1.$$

This rescaling of u^* does not affect its direction. Also, we are not assuming that

$$u_q^* = \max_{1 \leqslant i \leqslant r} u_i^*. \qquad \square$$

We now use (6.3) and Note 6.5 to work out the general form for u^* in terms of the unknown d_{mj}^*'s. The following algorithm is based on

Algorithm 3.13 for generating u for a consistent DM.

ALGORITHM 6.6

Step 0. Choose i such that $u_i = \max_{1 \leq j \leq r} u_j = 1$.

Set $u_i^* = 1$.

Set $I = \{ i \}$, $J = \phi$.

Step 1. Select $i \in I$, $i \notin J$.

For all objectives $j \notin I$, $1 \leq j \leq r$, where objective j is compared to objective i at the m^{th} pairwise comparison $(1 \leq m \leq p)$, define

(i) $\quad u_j^* = - d_{mi}^* u_i^* \quad$ if $\quad u_j^* + d_{mi}^* u_i^* = 0$

(ii) $\quad u_j^* = - u_i^* / d_{mj}^* \quad$ if $\quad u_i^* + d_{mj}^* u_j^* = 0$.

Set $I = I \cup \{ j \}$.

Repeat until no further $j \notin I$ exists which is compared to i at any pairwise comparison.

Step 2. If $I = \{ 1,\ldots,r \}$ then STOP.

Otherwise, set $J = J \cup \{ i \}$ and GO TO 1. □

EXAMPLE 6.7 *Consider the matrix* $\tilde{D}$ given in Example 6.2. $\tilde{D}$ has rank 6 and is, therefore, consistent. Normalising u so that $\max_i u_i = 1$, we have (to 5 decimal places):

$$u = (0,05053;\ 0,27603;\ 0,49458;\ 0,45808;\ 0,47660;\ 0,49758;\ 1). \quad (6.4)$$

D^* is then given by:

$$\begin{pmatrix} 1 & 0 & 0 & d_{14}^* & 0 & 0 & 0 \\ 0 & 1 & 0 & d_{24}^* & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & d_{35}^* & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & d_{46}^* & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & d_{57}^* \\ 0 & 1 & d_{63}^* & 0 & 0 & 0 & 0 \\ 1 & d_{72}^* & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & d_{84}^* & 1 & 0 & 0 \end{pmatrix}$$

where $d_{mj}^* < 0$; $1 \leq m \leq 8$; $1 \leq j \leq 7$.

We now use Algorithm 6.6 to find the general form for u^*

Step 0. Set $u_7^* = 1$, $I = \{ 7 \}$, $J = \phi$.

<u>Iteration</u> 1 $i = 7$.

Only 6 is compared to 7.

$$u_6^* = - d_{57}^* u_7^* = - d_{57}^*.$$

$I = \{ 6,7 \}$, $J = \{ 7 \}$.

<u>Iteration</u> 2 $i = 6$.

Only 3 is compared to 6.

$$u_3^* = - d_{46}^* u_6^* = d_{46}^* d_{57}^* .$$

$I = \{ 3,6,7 \}$, $J = \{ 6,7 \}$.

<u>Iteration</u> 3 $i = 3$.

2 and 5 are compared to 3.

$$u_2^* = - d_{63}^* u_3^* = - d_{63}^* d_{46}^* d_{57}^*$$

$$u_5^* = -u_3^* / d_{35}^* = d_{46}^* d_{57}^* / -d_{35}^*.$$

$I = \{ 2,3,5,6,7 \}$, $J = \{ 3,6,7 \}$.

<u>Iteration</u> 4 $i = 2$.

1 and 4 are compared to 2.

$$u_1^* = -d_{72}^* u_2^* = d_{72}^* d_{63}^* d_{46}^* d_{57}^*$$

$$u_4^* = -u_2^* / d_{24}^* = d_{63}^* d_{46}^* d_{57}^* / d_{24}^* .$$

$I = \{ 1,2,3,4,5,6,7 \}$. STOP.

Then u^* is assumed to be of the form (i.e. has the direction)

$$\begin{pmatrix} d_{72}^* d_{63}^* d_{46}^* d_{57}^* \\ -d_{63}^* d_{46}^* d_{57}^* \\ d_{46}^* d_{57}^* \\ d_{63}^* d_{46}^* d_{57}^*/d_{24}^* \\ -d_{46}^* d_{57}^*/d_{35}^* \\ d_{57}^* \\ 1 \end{pmatrix} \qquad (6.5)$$

□

<u>NOTE</u> 6.8 Since $\tilde{D}$ has more than $r-1$ rows, there is not a unique construction of u^*. At iteration 4, when $I = \{ 2,3,5,6,7 \}$ and $J = \{ 3,6,7 \}$, we may have chosen 5 from I instead of 2. This would have led to

$$u_4^* = -u_5^* / d_{84}^* = d_{46}^* d_{57}^* / d_{35}^* d_{84}^*.$$

Then $I = \{ 2,3,4,5,6,7 \}$ and $J = \{ 3,5,6,7 \}$. We may now have chosen 4 from I instead of 2. This gives

$$u_1^* = - d_{14}^* u_4^* = - d_{14}^* d_{46}^* d_{57}^* / d_{35}^* d_{84}^*.$$

Then, for u_1^* we must have that

$$d_{72}^* d_{63}^* d_{46}^* d_{57}^* = - d_{14}^* d_{46}^* d_{57}^* / d_{35}^* d_{84}^*$$

i.e. $$d_{72}^* d_{63}^* = - d_{14}^* / d_{35}^* d_{84}^* \tag{6.6}$$

and for u_4^* we must have

$$d_{63}^* d_{46}^* d_{57}^* / d_{24}^* = d_{46}^* d_{57}^* / d_{35}^* d_{84}^*$$

i.e. $$d_{63}^* / d_{24}^* = 1 / d_{35}^* d_{84}^* \tag{6.7}$$

Since the DM is being consistent (in the Ideal case), (6.6) and (6.7) should both hold and, therefore, it does not matter which elements are chosen from I. It can be easily verified that (6.6) and (6.7) are true by using (6.3) and substituting u_i^*'s and u_j^*'s for the d_{mj}^*'s. □

Algorithm 6.6 gives us a way of determining the general form for u^*. We now need to impose assumptions on the estimates of the d_{mj}^*'s made by the DM in order for (6.1) to be satisfied for some value of ϕ^k.

<u>ASSUMPTION</u> 6.9 Given objectives i and j $(1 \leqslant i,j \leqslant r;\ i \neq j)$ being compared at the m^{th} pairwise comparison $(1 \leqslant m \leqslant p)$ with $d_{mj} < 0$ (after normalisation (3.17)), we assume that d_{mj} satisfies

$$d_{mj} = d_{mj}^* + \varepsilon_m \tag{6.8}$$

where $-\Delta_m^1 \leqslant \varepsilon_m \leqslant \Delta_m^2$, $\Delta_m^1, \Delta_m^2 \geqslant 0$,

and d_{mj}^* is the true value of d_{mj}.

Clearly, if Δ_m^1 and Δ_m^2 are chosen to be large enough for all pairwise comparisons m, this assumption is always true. It is when we impose further assumptions on the deviations ε_m, that we encounter the problems mentioned in Example 6.3.

<u>ASSUMPTION</u> 6.10 Assume that the ε_m in Assumption 6.9 are proportional to the trade-off ratios d^*_{mj}.

$$\text{i.e.} \qquad \varepsilon_m = \delta_m d^*_{mj}; \qquad \forall\, m = 1,\ldots,p;\ 1 \leqslant j \leqslant r; \tag{6.9}$$

where $\quad -\Delta^1 \leqslant \delta_m \leqslant \Delta^2$ and $\Delta^1, \Delta^2 \geqslant 0$.

This assumption means that we are supposing that the accumulated proportional errors made by the DM at each pairwise comparison are, at worst, the same order of magnitude. i.e. the smaller $|d^*_{mj}|$ is, the smaller the accumulated error made in each pairwise comparison is (after normalisation (3.17)).

From (6.8) and (6.9), we have

$$d_{mj} = d^*_{mj}\,(1 + \delta_m); \quad \forall\, m = 1,\ldots,p;\ 1 \leqslant j \leqslant r. \tag{6.10}$$

Since the DM is consistent, we have from (6.2) that

$$d_{mj} = -u_i / u_j$$

where d_{mj}, u_i and u_j are known. Therefore,

$$d^*_{mj} = -u_i /(u_j(1+\delta_m)); \quad \forall\, m = 1,\ldots,p;\ 1 \leqslant j \leqslant r; \tag{6.11}$$

where $\quad -\Delta^1 \leqslant \delta_m \leqslant \Delta^2$ and $\Delta^1, \Delta^2 \geqslant 0$.

In the case where $p > r-1$, it may be impossible to apply Assumption 6.10 to all the d^*_{mj}'s simultaneously, as the next example demonstrates.

EXAMPLE 6.11 Consider a problem with 3 objective functions, where a consistent DM gives the indifference trade-off matrix

$$D = \begin{pmatrix} -0,25 & 1 & 0 \\ 0 & 1 & -0,25 \\ -1 & 0 & 1 \end{pmatrix}.$$

Then $\quad u = (1; 0,25; 1)$.

D^* has the form

$$\begin{pmatrix} d_{11}^* & 1 & 0 \\ 0 & 1 & d_{23}^* \\ d_{31}^* & 0 & 1 \end{pmatrix}.$$

Applying Algorithm 6.6, where i is chosen to be 1 in Step 0,

$$u^* = (1; - d_{11}^*; -d_{31}^*).$$

Using Assumption 6.10 and (6.11), we have

$$u^* = (1; u_2/(1+\delta_1); u_3/(1+\delta_3))$$

since $u_1 = 1$. For consistency, we need row 2 of D^* to satisfy

$$u_2/(1+\delta_1) + d_{23}^* u_3/(1+\delta_3) = 0$$

i.e. $$d_{23}^* = - u_2(1+\delta_3)/(u_3(1+\delta_1)).$$

Applying Assumption 6.10 and (6.11) to d_{23}^*, we see that we require

$$u_2/(u_3(1+\delta_2)) = - u_2(1+\delta_3)/(u_3(1+\delta_1))$$

i.e. $$1/(1+\delta_2) = (1+\delta_3)/(1+\delta_1).$$

We see that if we put δ_1 at its upper limit of Δ^2 and δ_3 at its lower limit of $-\Delta^1$, that δ_2 needs to be greater than Δ^2 in order for the equation to hold. Therefore, Assumption 6.10 will be contradicted for d_{23}^*. □

As the above example has shown, we cannot generally apply Assumption 6.10 when $p > r-1$. Since we only need $r-1$ rows of D^* to generate u^*, we concern ourselves with the $r-1$ rows which are used in Algorithm 6.6, *remembering that these* $r-1$ *rows are not the only rows which can be used* to find the form of u^*.

If the DM gives values for the d_{mj}'s which satisfy Assumptions 6.9 and 6.10, then we establish the following result which will enable us to compute values for Δ^1 and Δ^2 such that (6.1) is satisfied for some value of ϕ^k.

<u>THEOREM</u> 6.12 If at each pairwise comparison m used in Algorithm 6.6 (where objectives i and j are compared and $d_{mj} < 0$ after normalisation) the DM gives values for d_{mj} such that the matrix D has rank $r-1$ and Assumptions 6.9 and 6.10 are satisfied;

$$\text{i.e.} \quad d_{mj}^* = -u_i/(u_j(1+\delta_m)) \tag{6.12}$$

$$\text{where} \qquad -\Delta^1 \leqslant \delta_m \leqslant \Delta^2 \;;\;\; \Delta^1, \Delta^2 \geqslant 0; \tag{6.13}$$

then u^* computed from Algorithm 6.6 will have the bounds

$$\frac{u_i(1-\Delta^1)^{a_i}}{(1+\Delta^2)^{b_i}} \leqslant u_i^* \leqslant \frac{u_i(1+\Delta^2)^{a_i}}{(1-\Delta^1)^{b_i}} \tag{6.14}$$

where: for fixed i, u_i^* is defined at some iteration of Algorithm 6.6, say iteration k,

a_i and b_i are nonnegative integers which satisfy

$$a_i = \text{number of } d_{mj}^*\text{'s in the denominator of } u_i^* \tag{6.15}$$

$$b_i = \text{number of } d_{mj}^*\text{'s in the numerator of } u_i^* \tag{6.16}$$

$$a_i + b_i = k, \text{ the iteration of Algorithm 6.6} \tag{6.17}$$

and
$$u_q = \max_{1 \leq i \leq r} u_i = 1 = u_q^*. \tag{6.18}$$

Proof: From Step 0 of the algorithm we have that (6.18) is satisfied.

We prove the rest of the result by induction.

At iteration 1 of the algorithm we have that for all j that can be compared to $i = q$, either

(i)
$$\begin{aligned} u_j^* &= - d_{mi}^* u_i^* && \text{from (6.3)} \\ &= - d_{mi}^* && \text{since } u_i^* = 1 \text{ for } i = q \\ &= u_j/(\, u_i(1+\delta_m)\,) && \text{from (6.12)} \\ &= u_j/(1+\delta_m) && \text{since } u_i = 1 \text{ for } i = q. \end{aligned}$$

Then, from (6.13), we have

$$\frac{u_j}{(1+\Delta^2)} \leq u_j^* \leq \frac{u_j}{(1-\Delta^1)}.$$

which means that (6.14) is satisfied with $a_j = 0$ in (6.15), $b_j = 1$ in (6.16) and $a_j + b_j = 1$ in (6.17).

or, (ii)
$$\begin{aligned} u_j^* &= - u_i^* / d_{mj}^* && \text{from (6.3)} \\ &= - 1/ d_{mj}^* && \text{since } u_i^* = 1 \text{ for } i = q \\ &= u_j(1+\delta_m) / u_i && \text{from (6.12)} \\ &= u_j(1+\delta_m) && \text{since } u_i = 1 \text{ for } i = q. \end{aligned}$$

Then, from (6.13), we have

$$u_j(1-\Delta^1) \leq u_j^* \leq u_j(1+\Delta^2)$$

which means that (6.14) is satisfied with $a_j = 1$ in (6.15), $b_j = 0$ in (6.16) and $a_j+b_j = 1$ in (6.17).

Suppose now that the result is true at iteration k of Algorithm 6.6. i.e. that at the end of iteration k (or the start of $k+1$) there is an $i \in I$, $i \notin J$ which can be compared to objective $j \notin I$ at pairwise comparison m and that u_i^* satisfies (6.14) - (6.17) with $a_i = a$, $b_i = b$ and $a_i+b_i = a+b = k$. Then, one of the following two cases is true:

(i) $$u_j^* = -\, d_{mi}^*\, u_i^*$$

$$= u_i^*\, u_j \,/(\, u_i(1+\delta_m)\,) \qquad \text{from (6.12).}$$

Dividing (6.14) through by $u_i > 0$ and multiplying through by $u_j/(1+\delta_m) > 0$ we get

$$\frac{u_j\;(1-\Delta^1)^a}{(1+\delta_m)(1+\Delta^2)^b} \leq u_j^* \leq \frac{u_j\;(1+\Delta^2)^a}{(1+\delta_m)(1-\Delta^1)^b} .$$

Using (6.13) we have

$$u_j\,\frac{(1-\Delta^1)^a}{(1+\Delta^2)^{b+1}} \leq u_j^* \leq u_j\,\frac{(1+\Delta^2)^a}{(1-\Delta^1)^{b+1}} .$$

Therefore, (6.14) is satisfied with $a_j = a$ in (6.15), $b_j = b+1$ in (6.16) and $a_j+b_j = k+1$ in (6.17).

(ii) $$u_j^* = -u_i^* \,/\, d_{mj}^* \qquad \text{from (6.3)}$$

$$= u_i^* \, u_j(1+\delta_m) \,/\, u_i \qquad \text{from (6.12)}.$$

Dividing (6.14) through by $u_i > 0$ and muliplying through by $u_j(1+\delta_m) > 0$ gives (6.14) with $a_j = a_j+1$ in (6.15), $b_j = b$ in (6.16) and $a_j+b_j = k+1$ in (6.17).

Therefore, the theorem is proved. □

<u>COROLLARY</u> 6.13 If the assumptions and conditions of Theorem 6.12 are satisfied, then u_i^*, generated at the k^{th} *iteration of Algorithm 6.6*, has the bounds

$$u_i\left(\frac{(1-\Delta^1)^{a_i}}{(1+\Delta^2)^{b_i}} - 1 \right) \leqslant u_i^* - u_i \leqslant u_i\left(\frac{(1+\Delta^2)^{a_i}}{(1-\Delta^1)^{b_i}} - 1 \right) \tag{6.19}$$

where a_i and b_i are defined in (6.15) - (6.17), and

$$u_q^* = u_q = \max_{1 \leqslant i \leqslant r} u_i = 1,$$

i.e. $$u_q^* - u_q = 0. \tag{6.20}$$

<u>Proof:</u> (6.20) follows from (6.18)

(6.19) follows from (6.14). □

<u>EXAMPLE</u> 6.14 Suppose that the DM gives the indifference trade-off matrix $\bar{D}$ of Example 6.2 for a problem with 7 objectives. $\bar{D}$ has rank 6. The general *form of* u^* *is given in (6.5)*. Suppose that we require

$$| \, u_i^* - u_i \, | \leqslant 0,1 \qquad \text{for } i = 1,..,7$$

in (6.1), where u_i is given in (6.4).

To make the calculations easier, set $\Delta^1 = \Delta^2$. Then, from (6.19), we need (working to 5 decimal places):

<u>i = 1</u>

$$1/(1-\Delta^1)^4 \leqslant 0,1/u_1 + 1 = 2,97902$$

and $$1/(1+\Delta^1)^4 \geqslant -0,1/u_1 + 1 = -0,97902$$

i.e. $\Delta^1 \leqslant 0,23883.$

<u>i = 2</u>

$$1/(1-\Delta^1)^3 \leqslant 0,1/u_2 + 1 = 1,36228$$

$$1/(1+\Delta^1) \geqslant -0,1/u_2 + 1 = 0,63772$$

i.e. $\Delta^1 \leqslant 0,09792.$

<u>i = 3</u>

$$1/(1-\Delta^1)^2 \leqslant 0,1/u_3 + 1 = 1,20219$$

$$1/(1+\Delta^1)^2 \geqslant -0,1/u_3 + 1 = 0,79781$$

i.e. $\Delta^1 \leqslant 0,08796.$

<u>i = 4</u>

$$(1+\Delta^1)/(1-\Delta^1)^3 \leqslant 0,1/u_4 + 1 = 1,21830$$

$$(1-\Delta^1)/(1+\Delta^1)^3 \geqslant -0,1/u_4 + 1 = 0,78170$$

i.e. $\Delta^1 \leqslant 0,04873$

<u>i = 5</u>

$$(1+\Delta^1)/(1-\Delta^1)^2 \leqslant 0,1/u_5 + 1 = 1,20982$$

$$(1-\Delta^1)/(1+\Delta^1)^2 \geqslant -0,1/u_5 + 1 = 0,79018$$

which is satisfied by $\Delta^1 \leqslant 0,04873.$

i = 6

$$1/(1-\Delta^1) \leqslant 0,1/u_6 + 1 = 1.20097$$
$$1/(1+\Delta^1) \geqslant -0,1/u_6 + 1 = 0,79903$$

which are satisfied by $\Delta^1 \leqslant 0,04873$.

Therefore, the percentage error in each d^*_{mj} used in Algorithm 6.6 *to generate the* u^* *given in (6.5)*, may be at worst $\pm$ 4,8% in order to ensure that (6.1) is satisfied for $\phi^k - \varepsilon = 0,1$.

Using (6.11) with $\Delta^1 = \Delta^2 = 0,048$, we assume that

(i) $$-\frac{u_6}{u_7\,(1-0,048)} \leqslant d^*_{57} \leqslant -\frac{u_6}{u_7\,(1+0,048)}$$

i.e. $$-\,0,52290 \leqslant d^*_{57} \leqslant -\,0,47479$$

(ii) $$-\frac{u_3}{u_6(1-0,048)} \leqslant d^*_{46} \leqslant -\frac{u_3}{u_6(1+0,048)}$$

i.e. $$-\,1,04408 \leqslant d^*_{46} \leqslant -\,0,94844$$

(iii) $$-\frac{u_2}{u_3(1-0,048)} \leqslant d^*_{63} \leqslant -\frac{u_2}{u_3(1+0,048)}$$

i.e. $$-\,0,58626 \leqslant d^*_{63} \leqslant -\,0,53256$$

(iv) $$-\frac{u_2}{u_4(1-0,048)} \leqslant d^*_{24} \leqslant -\frac{u_2}{u_4(1+0,048)}$$

i.e. $$-\,0,63297 \leqslant d^*_{24} \leqslant -\,0,57499$$

(v) $$-\frac{u_1}{u_2(1-0,048)} \leqslant d^*_{72} \leqslant -\frac{u_1}{u_2(1+0,048)}$$

i.e. $$-\,0,19229 \leqslant d^*_{72} \leqslant -\,0,17468$$

(vi) $$-\frac{u_3}{u_5(1-0,048)} \leqslant d_{35}^* \leqslant -\frac{u_3}{u_5(1+0,048)}$$

i.e. $$-1,09002 \leqslant d_{35}^* \leqslant -0,99017.$$

(6.7) can be used to compute bounds for d_{84}^* and then (6.6) used to compute bounds for d_{14}^*. These bounds, however, will not satisfy Assumption 6.10.

Δ^1 was restricted to the critical value of 0,04873 when we determined $|u_4^* - u_4|$. To confirm that this is indeed the critical value, look at

$$u_4^* = d_{63}^* \, d_{46}^* \, d_{57}^* \, / \, d_{24}^*.$$

Suppose that d_{63}^*, d_{46}^* and d_{57}^* are at their upper bounds, while d_{24}^* is at its lowest bound. Then

$$|\, u_4^* - u_4 \,| = 0,09857$$

which would have been closer to 0,1 if we had worked with greater precision. □

6.2 ERROR BOUNDS ON INCONSISTENT RESPONSES

Suppose that the DM has given an indifference trade-off matrix D with corresponding inquiry pattern P which satisfies Restrictions 3.2 - 3.4, and that D has rank r. In this case, the DM must make p pairwise comparisons, where $p \geq r$, the number of objective functions. When D has rank r, the DM has given inconsistent responses when estimating the indifference trade-offs between pairs of objectives. The estimated vector of weights u is obtained by solving (3.16).

At the m^{th} pairwise comparison $(1 \leq m \leq p)$, suppose that objectives i and j $(1 \leq i,j \leq r;\ i \neq j)$ are compared and that

$$d_{mi} = 1; \quad 0 > d_{mj} \geq -1$$

after normalisation. Then, we have that for u satisfying (3.16),

$$u_i + d_{mj}\, u_j \neq 0$$

for at least one m $(1 \leq m \leq p)$. Since we know that an Ideal DM gives consistent responses and that we are attempting to find bounds on the values in D with respect to D^* (unknown), we direct our attention to finding the "nearest" consistent response to D.

DEFINITION 6.15 Given an indifference trade-off matrix D with corresponding inquiry pattern P satisfying Restrictions 3.2 - 3.4, we define the indifference trade-off matrix $\bar{D}$, with inquiry pattern P, to be a consistent set of responses with respect to the vector of weights u. i.e. $\bar{D}u = 0$, where u is the solution to (3.16), corresponding to the maximum eigenvalue.

NOTE 6.16

(i) $\bar{D}$ has rank $r-1$ (see Definition 3.12 for consistent).

(ii) If the matrix D is consistent (i.e. has rank $r-1$), then $\bar{D} = D$.

(iii) $\bar{D}$ is uniquely defined, subject to normalisation of the rows, since $\bar{d}_{mj} = -u_i/u_j$ (from $u_i + \bar{d}_{mj}u_j = 0$ for consistent). □

For each pairwise comparison m, define

$$\alpha_m = u_i + d_{mj}\, u_j \ ; \quad m = 1,\ldots,p; \quad 1 \leqslant i,j \leqslant r;\ i \neq j. \qquad (6.21)$$

Then, for a consistent DM, $\alpha_m = 0 \quad \forall\ m$; whereas this is not true for an inconsistent DM. Define the matrix $\bar{D}$ as follows:

$$\bar{d}_{mi} = d_{mi} = 1 \ ; \qquad m = 1,\ldots,p; \quad 1 \leqslant i \leqslant r; \qquad (6.22)$$

$$\bar{d}_{mj} = d_{mj} - \alpha_m/u_j; \qquad m = 1,\ldots,p; \quad 1 \leqslant j \leqslant r; \qquad (6.23)$$

and the remaining elements are zero, corresponding to the zeros in D. Then for any pairwise comparison m, we have that

$$\begin{aligned} u_i + \bar{d}_{mj}\, u_j &= u_i + (\, d_{mj} - \alpha_m/u_j\,)u_j \\ &= u_i + d_{mj}\, u_j - \alpha_m \\ &= 0, \qquad\qquad \text{from (6.21)} \end{aligned}$$

and so $\qquad \bar{D}\, u = 0.$

$\bar{D}$ is therefore a matrix of consistent responses corresponding to the vector of weights u.

<u>NOTE</u> 6.17 In Example 6.2 we had inconsistent matrix D with weights u computed from (3.16). The matrix $\bar{D}$, in the example, was constructed using (6.22) and (6.23). If ϕ^k is required to be close to zero in (6.1) (i.e. $u = u^*$), it is possible to construct a consistent matrix $\bar{D}$ corresponding to u, where we will need to assume that $\bar{D}$ is "close" to D^* in some sense. □

We now need to make assumptions on the errors which are being made by the DM in the matrix D. However, we impose these assumptions on the consistent matrix $\bar{D}$ and, therefore, implicitly on the elements of D. These assumptions are similar to Assumptions 6.9 and 6.10.

<u>ASSUMPTION</u> 6.18 Given objectives i and j ($1 \leqslant i,j \leqslant r$; $i \neq j$) being compared at the m^{th} pairwise comparison ($1 \leqslant m \leqslant p$) with $d_{mj} < 0$ (after normalisation (3.17)), we assume that the corresponding $\bar{d}_{mj} < 0$, defined in (6.23), satisfies

$$\bar{d}_{mj} = d^*_{mj} + \varepsilon_m , \qquad (6.24)$$

where $-\Delta^1_m \leqslant \varepsilon_m \leqslant \Delta^2_m$ and $\Delta^1_m, \Delta^2_m \geqslant 0$.

<u>ASSUMPTION</u> 6.19 Assume that the ε_m of Assumption 6.18 are proportional to the trade-off ratios d^*_{mj}.

i.e. $$\varepsilon_m = \delta_m d^*_{mj} ; \quad m = 1,\dots,p; \quad 1 \leqslant j \leqslant r; \qquad (6.25)$$

where $-\Delta^1 \leqslant \delta_m \leqslant \Delta^2$ and $\Delta^1, \Delta^2 \geqslant 0$.

From (6.24) and (6.25) we have that

$$d^*_{mj}\,(1+\delta_m) = \bar{d}_{mj}$$

$$= d_{mj} - \alpha_m / u_j \quad \text{from (6.23)},$$

and so we are assuming that d_{mj} satisfies

$$d_{mj} = d^*_{mj}\,(1+\delta_m) + \alpha_m/u_j\,; \quad m = 1,\dots,p;\ 1 \leqslant j \leqslant r; \quad (6.26)$$

where $\quad -\Delta^1 \leqslant \delta_m \leqslant \Delta^2 \quad$ and $\quad \Delta^1, \Delta^2 \geqslant 0.$

When the DM is consistent, i.e. $\alpha_m = 0 \;\; \forall\, m$, then (6.26) becomes (6.10).

Using (6.26) we find that

$$d^*_{mj} = (\, d_{mj} - \alpha_m/u_j\,)/(1+\delta_m)$$

$$= (\, d_{mj} - (u_i + d_{mj}\, u_j)/u_j\,)/(1+\delta_m)$$

$$= -\, u_i \,/(\, u_j(1+\delta_m)\,)$$

which is the same as (6.11). As a result, the procedure for establishing the ranges in which the elements d^*_{mj} are assumed to lie, is the same as that given in the previous section.

<u>NOTE</u> 6.20 A drawback to this approach is that although the elements of $\bar{D}$ fall into the computed ranges (since the variations are made about the $\bar{d}_{mj}$'s), the elements of D may not. Consider the matrix D given in Example 6.2, where $\bar{D}$ was constructed to be consistent with respect to the weights u. In Example 6.14, bounds were computed on the elements of D^* in order to ensure that (6.1) was satisfied with $\phi^k - \varepsilon = 0{,}1$. Looking at the element $d_{24} = -0{,}6'$ of D, we see that it lies outside the interval $(-0{,}63297;\ -0{,}57499)$ defined for d^*_{24}. We can no longer

be sure that (6.1) will hold for $\phi^k - \varepsilon = 0,1$. We may not conclude, however, that the problem lies with the estimate of d^*_{24} only. All that we do know is that the contradictions given by the DM have caused the condition to be violated.

If we find that this problem does arise, it is easy enough to present the computed ranges to the DM and ask him to re-examine the estimates he has made using the ranges as an aid. Any changes that the DM makes to the estimates will cause a shift in the computed ranges as well as a change in the width of the interval. □

NOTE 6.21 In the case where the (unknown) utility function is assumed to be linear, or approximated by a linear function, it could be helpful to the DM to present him with the current set of ranges as an aid when he revises his indifference trade-offs between objectives at the current solution under consideration. This is because the true indifference trade-offs should not vary from iteration to iteration. If the DM decides, at some stage, to revise the inquiry pattern P, then the ranges are only useful for pairwise comparisons which were made previously. Also, it should be remembered that the ranges are a function of the right hand side value in (6.1), and that this varies from iteration to iteration. □

6.3 CONVERGENCE OF ALGORITHM 4.1

We are now in a position to integrate the results of this chapter with the assumptions made in Chapter 5 which ensured the convergence of Algorithm 4.1 to the DM's most preferred solution when the utility function is assumed to be linear. This would provide bounds on the actual responses made by the DM in the matrix D, rather than bounds on the resulting vector of weights. There are, however, problems which arise if we press on and establish a uniform approach for finding bounds on the trade-off ratios. These include:

a) The DM's inquiry pattern may change at some iteration. As a result, requiring some form of weak consistency on the DM's responses from iteration to iteration (Assumption 5.5 or 5.10) cannot be applied when different sets of pairwise comparisons are made.

b) When $p \geqslant r$, the general form of u^* is not uniquely defined by the elements of D^* (see Note 6.8). As a result, depending on which elements of D^* appear in u^*, different ranges will be computed for the d_{mj}^*'s. Even when the DM gives consistent responses, Assumption 6.10 may not hold for all the trade-off ratios (as was shown in Example 6.11). It becomes impossible, therefore, with the assumptions which we have made, to establish uniform bounds on the elements of D^* which are easily computed by simply combining the results of the last two chapters.

We believe, therefore, that Assumptions 5.8 - 5.10 should be retained as the working assumptions for the convergence of Algorithm 4.1. The bounds on the trade-off ratios, which ensure the satisfaction of these assumptions can be computed in an ad-hoc manner by using Algorithm

6.6 to find a general form for u^* and then using Corollary 6.13 to determine the ranges for the d_{mj}^*'s used in the construction of u^*. The bounds on the remaining trade-off ratios (when $p \geqslant r$) can then be determined from the requirements that an Ideal DM is consistent. This ad-hoc approach was illustrated in Example 6.14.

6.4 CONCLUSION

In this chapter we have demonstrated how bounds on the trade-off ratios (given by the DM in the indifference trade-off matrix) can be determined in order to ensure that the resulting vector of weights lies in a predetermined range. We concluded that, because of the assumptions which we had made and the line of reasoning which we followed, an ad-hoc approach was the most suitable way of determining the bounds.

We assumed that the magnitude of the errors which were made in the estimates of the trade-off ratios were proportional to the true values (Assumption 6.10). As soon as uniform limits were applied to the constants of proportion (*viz.* $-\Delta \leqslant \delta_m \leqslant \Delta \;\; \forall\, m$, where δ_m is the constant of proportion for pairwise comparison m), we found that this assumption could not be applied to all the responses (Example 6.11) when the number of pairwise comparisons $p \geqslant r$, the number of objective functions. It is possible that this could be overcome (partially?) by doing one of the following:

a) not requiring u^* to be scaled so that

$$u_q^* = \max_{1 \leqslant i \leqslant r} \;\; u_i " u_q = 1.$$

This can be achieved by considerer ng αu^*, where α is a positive scalar. The disadvantage of doing this is that an extra parameter needs to be calculated when the limits Δ are being determined from (6.19) (where $\Delta^1 = \Delta^2 = \Delta$).

b) by allowing Δ^1 and Δ^2 in Assumption 6.10 to take on different values. In the practical implementation of Example 6.14 we set $\Delta^1 = \Delta^2$ so as to simplify the calculations. We note that setting $\Delta^1 = \Delta^2$ in Assumption 6.10 is equivalent to assuming that the DM is giving unbiased estimates of the trade-off ratios (instead of the original trade-offs). *Is this a reasonable assumption?* In addition, the question arises as to what the effects of an assumption of the form (5.10) will be on the results of this chapter?

We leave the above questions open for further investigation.

When the DM gives inconsistent responses, we consider the consistent matrix $\bar{D}$ corresponding to the already calculated vector of weights u, where $\bar{D}u = 0$. In Note 6.20 we pointed out that elements in D may fall outside the ranges which are computed about the elements of $\bar{D}$. If the DM gives responses which are 'very' inconsistent, this may well happen for any corresponding, consistent $\bar{D}$. A possibly better approach, however, which should reduce the chances of this happening, may be to find the 'nearest' consistent matrix $\tilde{D}$ to D. This could be done by solving

$$\begin{array}{lll} \underset{\tilde{d}_{mj} \in \tilde{D}}{\text{minimize}} & \left(\sum_{m=1}^{p} \left| d_{mj} - \tilde{d}_{mj} \right|^{s} \right)^{1/s} & \text{for } s = 1,2 \text{ or } \infty, \\ \text{subject to} & \tilde{u}_i + \tilde{d}_{mj} \tilde{u}_j = 0 & m = 1,\ldots,p, \\ & \left| u_i - \tilde{u}_i \right| \leqslant \delta & i = 1,\ldots,r, \\ & \tilde{u}_i \geqslant 0, \quad \tilde{d}_{mj} < 0, & \end{array}$$

where u and $\tilde{u}$ are normalised in the same way and δ is some predermined scalar. This is a mathematical programming problem with p ($\tilde{d}_{mj}$ values) + r ($\tilde{u}_i$ values) unknowns and p (consistency equations) + $2r$ (inequalities on $\tilde{u}_i$) + 1 (normalisation) constraints. The matrix $\bar{D}$ (defined in (6.22) (6.23)), corresponding to u, is then a feasible solution to the above problem with δ set to 0.

While this approach may well give us a 'better' estimate than $\bar{D}$

to work with, the disadvantage is the obvious increase in complexity to find the matrix. Therefore, although it may be appealing to compute $\tilde{D}$ from a theoretical point of view, the question arises as to whether the results will be that much more useful to the DM than the results obtained from using the matrix $\bar{D}$, which is far easier to calculate.

CHAPTER 7. THE CONCAVE UTILITY FUNCTION

Assume that the explicitly unknown utility function U is a concave function of the r objectives which is two times continuously differentiable. Problem (1.4) is then

$$\text{maximize} \quad U(f_1(x),\ldots,f_r(x)) \tag{7.1}$$

$$\text{subject to} \quad Ax = b; \quad x \geqslant 0, \tag{7.2}$$

In this chapter we propose a cutting plane algorithm to solve (7.1) (7.2). We first consider the algorithm when the Decision Maker is giving the trade-offs correctly at each iteration. i.e. we have an Ideal DM, or U is known explicitly. We prove that the proposed algorithm will either terminate after a finite number of iterations with the DM's most preferred solution or will find this solution as the limit of some sequence. We then consider the case when the DM is making errors in his estimates of the trade-offs. Provided that the resulting vector of weights (approximations to the true weights) satisfy certain assumptions, we show that the algorithm will still find the DM's most preferred solution (again, in a finite number of steps or as the limit of some subsequence) to (7.1) (7.2). Problems which arise in the implementation of the theoretical algorithm (and the assumptions which need to be satisfied) are then discussed and practical ways of overcoming these are suggested.

7.1 AN INTERACTIVE CUTTING PLANE ALGORITHM

Define the constraint set (7.2) by

$$X \triangleq \{ x \mid Ax = b,\ x \geq 0 \}. \tag{7.3}$$

Problem (7.1) (7.2) can be reformulated as the equivalent problem:

$$\begin{aligned} &\text{maximize} && y \\ &\text{subject to} && U(f(x)) - y \geq 0 \\ & && x \in X, \end{aligned} \tag{7.4}$$

where y is a scalar variable. The constraint set (7.4) is a convex set (from the concavity of U). Suppose that $\bar{x}$ is an optimal solution to (7.1) (7.2). Then we have the following result:

<u>THEOREM</u> 7.1 If $\bar{x}$ is an optimal solution to (7.1) (7.2) then, given any point $z \in X$:

(i) $x = \bar{x}$, $y = 0$ satisfies the problem:

find $x \in X$, $y \geq 0$ such that

$$(f(x) - f(z)).\nabla U(f(z)) \geq y; \quad \text{and}$$

(ii) if $(f(x) - f(z)).\nabla U(f(z)) \leq 0 \quad \forall\, x \in X,$

then z is optimal for (7.1) (7,2).

<u>Proof</u>:

(i) Since $\bar{x}$ is optimal, we have that

$$U(f(\bar{x})) \geq U(f(z)) \quad \forall\, z \in X.$$

From the concavity of U we have that

$$(f(\bar{x}) - f(z)).\nabla U(f(z)) \geqslant U(f(\bar{x})) - U(f(z)) \geqslant 0 \quad \forall\ z \in X$$

which gives the result.

(ii) $$0 \geqslant (f(x) - f(z)).\nabla U(f(z)) \quad \forall\ x \in X$$

$$\geqslant U(f(x)) - U(f(z)) \quad \text{by concavity of } U$$

which gives the result. □

REMARK 7.2 It should be remembered that we are not estimating $\nabla U(f(z))$ *at some point* $z \in X$, *but a normalised vector of weights which is* collinear with $\nabla U(f(z))$. If the direction of the vector of weights $u(z)$ is given correctly (i.e. we have an Ideal DM), then

$$u(z) = \nabla U(f(z)) / \| \nabla U(f(z)) \|$$

and the vector is denoted by $u^*(z)$. The results in the theorem above still hold if we replace $\nabla U(f(z))$ by $u^*(z)$. □

We use the above results in the following algorithm.

ALGORITHM 7.3

Step 0. Solve the linear programming problem

$$\text{maximize} \quad \sum_{i=1}^{r} f_i(x)$$

$$\text{subject to} \quad x \in X.$$

Call the solution x^1. Set $k = 1$.

Step 1. Present x^k and $f(x^k) = (f_1(x^k),\ldots,f_r(x^k))$ to the DM. Obtain the indifference trade-off matrix D^k from the DM. *Calculate the estimated vector of weights* u^k *by solving* the eigenvalue/ eigenvector problem

$$-(D^k)^T D^k u = \lambda u$$

for the eigenvector corresponding to the maximum eigenvalue.

Step 2. Solve the linear programming problem

$$\text{maximize} \quad y \tag{7.5}$$

$$\text{subject to} \quad (f(x) - f(x^j)).u^j \geqslant y; \quad j = 1,\ldots,k \tag{7.6}$$

$$x \in X, \quad y \geqslant 0 \tag{7.7}$$

Call the solution (x^{k+1}, y^{k+1}).

Step 3. If $y^{k+1} = 0$ then STOP.
Otherwise, set $k = k+1$ *and GO TO 1.* □

<u>REMARK</u> 7.4

(i) The representation of the constraint set (7.4) by a set of linear inequalities (7.6) is similar to the method proposed by Kelley (1960). Kelly, however, used the cutting plane constraints

$$(f(x) - f(x^j)).\nabla U(f(x^j)) + U(f(x^j)) \geqslant y \; ; \; j = 1,\ldots,k \tag{7.8}$$

instead of (7.6). We have omitted the $U(f(x^j))$ term in (7.8) because we are considering problems where U is not known explicitly.

(ii) A step length procedure can be incorporated into Algorithm 7.3 as follows:

Call the solution of problem (7.5) - (7.7), solved in Step 2, (z^{k+1}, y^{k+1}) and replace Step 3 with:

If $y^{k+1} = 0$ then STOP.

Otherwise, solve the 1-dimensional problem

$$\underset{0 \leq \rho \leq 1}{\text{maximize}} \quad U(f(x^k + \rho(z^{k+1} - x^k)))$$

to find the optimal step length (see Section 1.4). Call the solution ρ^k and set

$$x^{k+1} = x^k + \rho^k(z^{k+1} - x^k).$$

Set $k = k+1$ and GO TO 1.

(iii) The advantage of including a step length procedure is that we have

$$U(f(x^{k+1})) > U(f(x^k)).$$

i.e. the algorithm gives ascent at each iteration; whereas Algorithm 7.3 does not necessarily do so. As a result, Algorithm 7.3 would be expected to exhibit a slower rate of convergence than an algorithm which includes a step length procedure. The disadvantages are that an extra interaction with the DM is required at each iteration (although this is generally considered not to present too much of a problem - see Section 1.4) and x^{k+1} may be dominated; whereas Algorithm 7.3 generates nondominated solutions only. □

We now show that the sequence $\{y^k\}$, generated by Algorithm 7.3, is a nonincreasing sequence which has nonnegative elements. We first make the following assumption which ensures that the sequence is also bounded above.

<u>ASSUMPTION</u> 7.5 Assume that the constraint set X, defined in (7.3), is bounded.

<u>LEMMA</u> 7.6 If Assumption 7.5 holds, then the sequence $\{y^k\}$, generated by Algorithm 7.3 satisfies

$$0 \leq y^{k+1} \leq y^k < \infty \qquad \text{for} \quad k = 2,3,\ldots$$

<u>Proof</u>: At iteration k of Algorithm 7.3 we have the constraints

$$(f(x) - f(x^j)).u^j \geq y \geq 0 \qquad \forall\, j = 1,\ldots,k.$$

where

$$(f(x^k) - f(x^j)).u^j \geq y^k > 0 \qquad \forall\, j = 1,\ldots,k-1,$$

otherwise the algorithm would have terminated before iteration k. Therefore,

$$(f(x^k) - f(x^j)).u^j \geq 0 \qquad \forall\, j = 1,\ldots,k;$$

and so there exists $x = x^k \in X$ with corresponding $y = 0$ which ensures that $y^{k+1} \geq 0$.

Assumption 7.5, the linearity of f and the structure of D^1 (from which u^1 is computed) ensure that

$$0 \leq y^2 = (f(x^2) - f(x^1)).u^1 < \infty.$$

Finally, the sequence is nonincreasing because an additional constraint is added to the constraint set (7.6) at each iteration. □

THEOREM 7.7 If Assumption 7.5 holds, then the sequence $\{y^k\}$, generated by Algorithm 7.3, has the properties:

(i) $y^k \to 0$ as $k \to \infty$; $k \geq 2$; and

(ii) there exists a subsequence $\{y^{k_j}\}$ of $\{y^k\}$ such that

$$y^{k_{j+1}} < y^{k_j}$$

$\forall\, j$ such that $y^{k_j} > 0$.

Proof:

(i) At iteration $k-1$ we have

$$0 \leq y^k \leq (f(x^k) - f(x^j)).u^j \qquad \forall\, j = 1,\ldots,k-1$$

$$\leq \| f(x^k) - f(x^j) \|_2 \, \| u^j \|_2 \tag{7.9}$$

Since $x^k \in X$ for all k and X is bounded, there is a subsequence $\{x^{k_j}\}$ of the sequence $\{x^k\}$ which tends to some accumulation point x^*, say, as $j \to \infty$. Since f is linear, and $\| u^j \|_2$ is bounded (from the construction of D^j), we have that

$$\| f(x^{k_{j+1}}) - f(x^{k_j}) \|_2 \, \| u^{k_j} \| \to 0 \quad \text{as} \quad j \to \infty.$$

From (7.9)

$$0 \leq y^{k_{j+1}} \leq \| f(x^{k_{j+1}}) - f(x^{k_j}) \|_2 \, \| u^{k_j} \|_2 .$$

Now the right hand side tends to zero as $j \to \infty$. This means that $y^{k_j} \to 0$ as $j \to \infty$. From Lemma 7.6, we have that the sequence $\{y^k\}$ is nonincreasing. This gives the result.

(ii) Consider the subsequence $\{x^{k_j}\}$ referred to in (i) corresponding to $\{y^{k_j}\}$. Now, it is impossible for

$$y^{k_{j+1}} - y^{k_j} > 0$$

to hold for all j, since we have that $y^{k_j} \to 0$ as $j \to \infty$. Choose a subsequence from $\{y^{k_j}\}$ with the property that the values in the subsequence are strictly decreasing. □

The results that we have proved above for the sequence $\{y^k\}$, hold for any set of weights u^k generated from the indifference trade-off matrix D^k given by the DM. In the next section we prove convergence of the algorithm in the case where we have an Ideal Decision Maker.

7.2 CONVERGENCE OF THE ALGORITHM FOR AN IDEAL DECISION MAKER

We first consider the case where we have an Ideal DM (or U is known *explicitly*). i.e. we have that the normalised weights u^k, which are computed in Step 1 of Algorithm 7.3, satisfy

$$u^k \triangleq u(x^k) = u^*(x^k) \triangleq \nabla U(f(x^k))/\| \nabla U(f(x^k)) \| . \tag{7.10}$$

<u>LEMMA</u> 7.8 If $\bar{x}$ is an optimal solution to (7.1) (7.2), then $(\bar{x},0)$ is a feasible solution to the constraints (7.6) (7.7) at iteration k of Algorithm 7.3.

<u>Proof</u>: Follows from Theorem 7.1 (i) and Remark 7.2 □

This result means that any optimal solution to (7.1) (7.2) *is not* "cut out" of the constraint set (7.6) (7.7) at any stage of Algorithm 7.3 if the direction of the marginal utility vector is given correctly.

<u>THEOREM</u> 7.9 Suppose that Assumption 7.5 *holds*. Then *(remembering that* U is assumed to be a twice continuously differentiable concave function), Algorithm 7.3 either

(i) terminates after a finite number of iterations k with a solution x^k iff x^k is an optimal solution to (7.1) (7.2), or

(ii) generates an infinite sequence $\{x^k\}$, where some subsequence of $\{x^k\} \to \bar{x}$ as $k \to \infty$, where $\bar{x}$ is an optimal solution to (7.1) (7.2).

<u>Proof</u>:

(i) Suppose that x^k is optimal. We need to show that $y^{k+1} = 0$ so that the algorithm terminates. Since x^k is optimal, we have that

$$U(f(x^k)) \geqslant U(f(x^{k+1}))$$

and $\quad U(f(x^k)) \geq U(f(z))$

where $\quad z = x^k + \Theta(x^{k+1} - x^k) \in X \qquad \Theta \in [0,1]$

Therefore,

$$0 \geq U(f(z)) - U(f(x^k))$$

$$= (f(z) - f(x^k)).\nabla U(f(x^k))$$
$$+ \tfrac{1}{2}(f(z) - f(x^k)).\nabla^2 U(f(\bar{z})).(f(z) - f(x^k))^T$$

by the Second Mean Value Theorem, where

$$\bar{z} = x^k + \bar{\Theta}(x^{k+1} - x^k) \qquad \bar{\Theta} \in [0,\Theta].$$

Since f is linear we have that

$$f(z) - f(x^k) = f(x^k + \Theta(x^{k+1} - x^k)) - f(x^k)$$
$$= \Theta(f(x^{k+1}) - f(x^k)).$$

Therefore,

$$0 \geq \Theta(f(x^{k+1}) - f(x^k)).\nabla U(f(x^k))$$
$$+ \tfrac{1}{2}\,\Theta^2(f(x^{k+1}) - f(x^k)).\nabla^2 U(f(\bar{z})).(f(x^{k+1}) - f(x^k))^T.$$

For Θ sufficiently small, the first term will dominate the second term, and so, for $\Theta > 0$, Θ arbitrarily small, we have that

$$(f(x^{k+1}) - f(x^k)).\nabla U(f(x^k)) \leq 0 \tag{7.11}$$

Dividing through by $\|\nabla U(f(x^k))\| > 0$ and using (7.10) we have

$$(f(x^{k+1}) - f(x^k)).u^k \leq 0. \tag{7.12}$$

Therefore, from (7.6) (7.7) we get that $y^{k+1} = 0$.

Suppose that the algorithm terminates at iteration k with a solution x^k. i.e. $y^{k+1} = 0$. For any optimal point $\bar{x}$ to problem (7.1) (7.2), we have from Theorem 7.1 (i) that

$$(f(\bar{x}) - f(x^j)).\nabla U(f(x^j)) \geq 0 \qquad \forall\ j = 1,\dots,k.$$

Assume that

$$(f(\bar{x}) - f(x^j)).\nabla U(f(x^j)) = 0 \quad \textit{for some} \quad j \in [1,..,k-1].$$

Then

$$\begin{aligned} 0 &= (f(\bar{x}) - f(x^j)).\nabla U.f(x^j)) \\ &\geq U(f(\bar{x})) - U(f(x^j)) \qquad \text{by concavity of } U, \end{aligned} \tag{7.13}$$

which implies that x^j is optimal. Therefore, by (7.12), $y^{j+1} = 0$ and the algorithm would have terminated earlier. This means that

$$(f(\bar{x}) - f(x^j)).\nabla U(f(x^j)) > 0 \qquad \forall\ j = 1,\dots,k-1,$$

and so

$$(f(\bar{x}) - f(x^j)).u^j > 0 \qquad \forall\ j = 1,\dots,k-1.$$

Therefore, we must have that

$$(f(\bar{x}) - f(x^k)).u^k = 0, \tag{7.14}$$

otherwise, for this $\bar{x}$, we have

$$(f(\bar{x}) - f(x^j)).u^j > 0 \qquad \forall\ j = 1,\dots,k$$

which contradicts the algorithm stopping at iteration k with $y^{k+1} = 0$. Multiplying (7.14) by $\|\nabla U(f(x^k))\|$ and using (7.10) and (7.13) we conclude that x^k is optimal.

(ii) Let $\Omega \subset X$ denote the set of optimal points for (7.1) (7.2). Since X is bounded, Ω is a compact set. All the points $\bar{x} \in \Omega$ are feasible for some $y > 0$ in (7.6) as $k \to \infty$. As $k \to \infty$, if Ω is the only set of feasible points satisfying (7.6) for $y^k > 0$, then the theorem is proved for all convergent subsequences of $\{x^k\}$. If there exists $z \notin \Omega$, $z \in X$ which is feasible for $y^k > 0$ as $k \to \infty$, we have that

$$(f(z) - f(x^k)).u^k \geqslant 0 \qquad \forall\ k = 1,\ldots,\infty. \tag{7.15}$$

We also have that

$$(f(\bar{x}) - f(x^k)).u^k > 0 \qquad \forall\ k = 1,\ldots,\infty;\ \forall\ \bar{x} \in \Omega. \tag{7.16}$$

Consider a point

$$v = \theta\bar{x} + (1-\theta)z \qquad\qquad \theta \in (0,1)$$

where $\bar{x}$ is a point in the compact set Ω such that

$$\| \bar{x} - z \| = \min_{x \in \Omega} \| x - z \| .$$

Then $v \notin \Omega$, $v \in X$, and

$$\begin{aligned}
&(f(v) - f(x^k)).u^k \\
&= (f(\theta\bar{x} + (1-\theta)z) - f(x^k)).u^k \\
&= \theta(f(\bar{x}) - f(x^k)).u^k + (1-\theta)(f(z) - f(x^k)).u^k \\
&\qquad\qquad\qquad \text{by the linearity of } f \\
&> 0 \qquad \text{by (7.15), (7.16) and } \theta \in (0,1).
\end{aligned}$$

Therefore, there must exist some subsequence $\{x^{k_j}\}$ of $\{x^k\}$ with limit point x^*, say, such that

$$(f(v) - f(x^{k_j})).u^{k_j} \to 0 \quad \text{as} \quad j \to \infty,$$

otherwise we have that there exists $v \in X$ with

$$\max \; y = \min_k \; (f(v) - f(x^k)).u^k \geqslant \varepsilon > 0$$

which contradicts $y^k \to 0$ as $k \to \infty$. Assume that x^* is not optimal.
Then

$$(f(\bar{x}) - f(x^*)).\nabla U(f(x^*)) > 0 \quad \text{since} \quad x^* \notin \Omega \tag{7.17}$$

and

$$(f(v) - f(x^*)).u(x^*) = 0 \tag{7.18}$$

where $u^{k_j} = u(x^{k_j}) \to u(x^*)$ as $j \to \infty$. Dividing (7.17) by $\| \nabla U(f(x^*)) \| > 0$, and using (7.10), gives

$$(f(\bar{x}) - f(x^*)).u(x^*) > 0. \tag{7.19}$$

Now

$$(f(v) - f(x^{k_j})).u^{k_j}$$

$$= \Theta(f(\bar{x}) - f(x^{k_j})).u^{k_j} + (1-\Theta)(f(z) - f(x^{k_j})).u^{k_j}.$$

Letting $j \to \infty$, we have that the left hand side tends to zero (by (7.18)) while the right hand side has a positive limit (by (7.15) and (7.19)). This contradiction means that $x^* \in \Omega$, which gives the result. $\square$

7.3 BOUNDS ON THE ESTIMATED WEIGHTS

In this section we are concerned with the case where a DM gives an indifference trade-off matrix D^k at iteration k of Algorithm 7.3. This leads to an approximation u^k to the true vector of weights $u^*(x^k)$ at the point x^k. During the early iterations of the algorithm we assume that the DM makes some error in the estimates. We cannot, however, assume that these estimates are too "wild". In order to prove convergence of Algorithm 7.3 to the DM's most preferred solution $\bar{x} \in X$, we need to assume that the estimates become more accurate as the number of iterations increases. The following assumption has the characteristics mentioned above.

<u>ASSUMPTION</u> 7.10 At iteration k of Algorithm 7.3, assume that the approximations u^k (obtained indirectly from the DM) to the true set of weights $u^*(x^k)$, where $u^*(x^k)$ is scaled so that

$$u_q^*(x^k) = u_q^k = \max_{1 \leq i \leq r} u_i^k = 1,$$

satisfies

$$| \; u_i^k - u_i^*(x^k) \; | \leq \phi^{k+1} \qquad \text{for } i = 1,\ldots,r \tag{7.20}$$

where

$$0 \leq \phi^{k+1} < y^{k+1} / \max_{1 \leq j \leq k} \Big(\sum_{\substack{i=1 \\ i \neq q}}^{r} |f_i(x^{k+1}) - f_i(x^j)| \Big) \qquad \text{if } y^{k+1} > 0 \tag{7.21}$$

$$\phi^{k+1} = 0 \qquad \text{if } y^{k+1} = 0; \tag{7.22}$$

y^{k+1} is the optimal solution to the linear programming problem (7.5) - (7.7).

LEMMA 7.11 Suppose that Assumptions 7.5 and 7.10 are satisfied. At iteration k of Algorithm 7.3, if

$$(f(x^{k+1}) - f(x^j)).u^j \geq y^{k+1} > 0 \qquad \forall\ j = 1,\ldots,k,$$

then

$$(f(x^{k+1}) - f(x^j)).u^*(x^j) > 0 \qquad \forall\ j = 1,\ldots,k.$$

Proof: The proof is based on the proof of Theorem 5.1. For all $j = 1,\ldots,k$, we have:

$$\begin{aligned}
&(f(x^{k+1}) - f(x^j)).u^*(x^k)\\
&= (f_q(x^{k+1}) - f_q(x^j))\ u_q^*(x^j)\\
&\quad + \sum_{\substack{f_i(x^{k+1}) \geq f_i(x^j)\\ i \neq q}} (f_i(x^{k+1}) - f_i(x^j))\ u_i^*(x^j)\\
&\quad + \sum_{\substack{f_i(x^{k+1}) < f_i(x^j)\\ i \neq q}} (f_i(x^{k+1}) - f_i(x^j))\ u_i^*(x^j)\\
&\geq (f_q(x^{k+1}) - f_q(x^j))\ u_q^j\\
&\quad + \sum_{\substack{f_i(x^{k+1}) \geq f_i(x^k)\\ i \neq q}} (f_i(x^{k+1}) - f_i(x^j))(\ u_i^j - \phi^{k+1}\ )\\
&\quad + \sum_{\substack{f_i(x^{k+1}) < f_i(x^k)\\ i \neq q}} (f_i(x^{k+1}) - f_i(x^j))(\ u_i^j + \phi^{k+1}\ )\\
&= \sum_{i=1}^{r} (f_i(x^{k+1}) - f_i(x^j))\ u_i^j\\
&\quad - \phi^{k+1} \sum_{\substack{i=1\\ i\neq q}}^{r} |f_i(x^{k+1}) - f_i(x^k)|
\end{aligned}$$

$$\geq y^{k+1} - \phi^{k+1} \max_{1 \leq j \leq r} \left(\sum_{\substack{i=1 \\ i \neq q}}^{r} |f_i(x^{k+1}) - f_i(x^j)| \right)$$

$$> 0 \quad \text{from (7.21).} \qquad \square$$

<u>THEOREM</u> 7.12 Suppose that Assumptions 7.5 and 7.10 are satisfied. Then Algorithm 7.3 will terminate after a finite number of iterations with a solution x^k iff x^k is an optimal solution to (7.1) (7.2).

<u>Proof:</u> Suppose that x^k is optimal. Assume that the algorithm does not terminate at iteration k. i.e. $y^{k+1} > 0$. Then

$$(f(x^{k+1}) - f(x^k)).u^k \geq y^{k+1} > 0.$$

From Lemma 7.11 we have that

$$(f(x^{k+1}) - f(x^k)).u^*(x^k) > 0$$

i.e. $$(f(x^{k+1}) - f(x^k)).\nabla U(f(x^k)) > 0.$$

In Theorem 7.9, equation (7.11), we showed that if x^k is optimal, then

$$(f(x^{k+1}) - f(x^k)).\nabla U(f(x^k)) \leq 0$$

which is a contradiction. Therefore, $y^{k+1} = 0$.

Now suppose that the algorithm terminates with x^k and $y^{k+1} = 0$ during iteration k. Then we must have that

$$y^j > 0 \; ; \quad j = 1,\ldots,k;$$

otherwise the algorithm terminates at an earlier iteration. Assume that x^k is not optimal. Then we need to show that there exists $v \in X$ with

$$(f(v) - f(x^j)).u^j > 0 \qquad \forall\, j = 1,\ldots,k$$

which will contradict the optimality of $y^{k+1} = 0$ in (7.5) - (7.7).

We have

$$(f(x^k) - f(x^j)).u^j \geqslant y^k > 0 \qquad \text{for } j = 1,\ldots,k-1. \qquad (7.23)$$

We also have that

$$(f(\bar{x}) - f(x^j)).u^j \geqslant M > -\infty \qquad \text{for } j = 1,\ldots,k-1 \qquad (7.24)$$

for any optimal point $\bar{x} \in X$. Define

$$v = \Theta\bar{x} + (1-\Theta)x^k \qquad \Theta \in (0,1).$$

Multiplying (7.23) by $(1-\Theta)$ and (7.24) by Θ and using the linearity of f gives

$$(f(v) - f(x^j)).u^j \geqslant \Theta M + (1-\Theta)y^k \qquad \text{for } j = 1,\ldots,k-1.$$

Now, for Θ sufficiently small, $\Theta > 0$, the right hand side is positive. i.e. there exists $\Theta \in (0,1)$ such that

$$(f(v) - f(x^j)).u^j > 0 \qquad \text{for } j = 1,\ldots,k-1 \qquad (7.25)$$

For any nonoptimal point (in this case x^k) we have that

$$(f(\bar{x}) - f(x^k)).\nabla U(f(x^k)) > 0$$

i.e. $$(f(\bar{x}) - f(x^k)).u^*(x^k) > 0$$

Since $y^{k+1} = 0$, we have from (7.22) and (7.20) that

$$(f(\bar{x}) - f(x^k)).u^k > 0.$$

Therefore, for any $\Theta > 0$, we have

$$0 < \theta(f(\bar{x}) - f(x^k)).u^k$$

$$= \theta(f(\bar{x}) - f(x^k)).u^k + (1-\theta)(f(x^k) - f(x^k)).u^k$$

$$= (\theta f(\bar{x}) + (1-\theta)f(x^k) - f(x^k)).u^k$$

$$= (f(\theta\bar{x} + (1-\theta)x^k) - f(x^k)).u^k \quad \text{from the linearity of } f$$

and so

$$(f(v) - f(x^k)).u^k > 0. \tag{7.26}$$

Combining (7.25) and (7.26) gives $v \in X$ such that

$$(f(v) - f(x^j)).u^j > 0 \qquad \forall\ j = 1,\ldots,k,$$

which gives us the required contradiction. Therefore, x^k is optimal. □

This theorem has an interesting implication. We have not shown that any optimal $\bar{x} \in X$ satisfies the constraints

$$(f(\bar{x}) - f(x^j)).u^j > 0 \qquad \forall\ j = 1,\ldots,k \tag{7.27}$$

at iteration k. In fact, as the next example will demonstrate, Assumption *7.10 is not strong enough to guarantee this. As a result,* if all the optimal solutions are cut out of the constraint set (7.6) (7.7) by inaccurate information about u^j, then the theorem implies that the algorithm will not stop in a finite number of iterations. In this case, we can conclude that any subsequences of the infinite sequence $\{x^k\}$ will have accumulation points which are not optimal for (7.1) (7.2).

We also observe that in the proof of the first part of the theorem, we did not need to restrict ϕ^{k+1} to satisfy (7.21). All that is needed is that ϕ^{k+1} satisfy

$$\phi^{k+1} < y^{k+1} / \sum_{\substack{i=1 \\ i \neq q}}^{r} | f_i(x^{k+1}) - f_i(x^k) |. \tag{7.28}$$

In this case, although the sequence $\{y^k\}$ is nondecreasing, with a limit of zero, ϕ^{k+1} may be larger than ϕ^k. Although (7.28) *is less restrictive* than (7.21) and, therefore, "allows" the DM to be less accurate at certain later iterations than he is at earlier ones, we assume that some "learning process", on behalf of the DM, is involved in the interactive procedure. i.e. that the estimates are "getting better" as the DM learns more about the problem and sees the consequences of previous estimates. This assumption is incorporated in (7.21). In addition, (7.21) is not sufficient to guarantee that any optimal *solution satisfies constraints* (7.6) (7.7). We do not wish to make matters even worse by assuming the possibility of larger errors.

<u>EXAMPLE</u> 7.13 *Consider the example given by* Zionts and Wallenius (1976).

i.e. maximize

$$f_1 = 3x_1 + x_2 + 2x_3 + x_4$$
$$f_2 = x_1 - x_2 + 2x_3 + 4x_4$$
$$f_3 = -x_1 + 5x_2 + x_3 + 2x_4$$

subject to

$$2x_1 + x_2 + 4x_3 + 3x_4 \leq 60$$
$$3x_1 + 4x_2 + x_3 + 2x_4 \leq 60$$

$$x_1, x_2, x_3, x_4 \geq 0.$$

The DM is assumed to have a linear (a *special case of concave) utility* function

$$U(f_1, f_2, f_3) = u_1^* f_1 + u_2^* f_2 + u_3^* f_3$$

where $u^* = (\,0{,}58;\ 0{,}21;\ 0{,}21\,)$. We *normalise* u^* *so that*

$$u^* = (1;\ 0,3621;\ 0,3621).$$

Suppose that we present the DM with the efficient solution

$$x^1 = (0;\ 15;\ 0;\ 0),\quad f(x^1) = (15;\ -15;\ 75).$$

This has corresponding simplex tableau:

Basis	x_B	x_1	x_2	x_3	x_4	x_5	x_6
5	45	1,25		3,75	2,5	1	-0,25
2	15	0,75	1	0,25	0,5		0,25
f_1	15	-2,25		-1,75	-0,5		0,25
f_2	-15	-1,75		-2,25	-4,5		-0,25
f_3	75	4,75		0,25	0,5		1,25

Suppose that the DM gives *an estimate of* u^1 which, after normalisation, is

$$u^1 = (1;\ 0,2;\ 0,8).$$

Then $\max\limits_i \mid u_i^* - u_i^1 \mid = 0,4379.$

The simplex tableau corresponding to these weights is given on the next page. We see that

$$x^2 = (0;\ 12;\ 12;\ 0) \text{ and } f(x^2) = (36;\ 12;\ 72).$$

$$f(x^2) - f(x^1) = (36-15;\ 12+15;\ 72-75)$$
$$= (21;\ 27;\ -3).$$

$$(f(x^2) - f(x^1)).u^1 = (21;\ 27;\ -3).(1;\ 0,2;\ 0,8)$$
$$= 24 = y^2.$$

Therefore,

$$\phi^2 < \frac{y^2}{\sum_{i=2}^{3} |f_i(x^2) - f_i(x^1)|} = \frac{24}{30} = 0{,}8$$

and we have that Assumption 7.10 is satisfied. Now, as we saw in Section 4.4, the optimal solution corresponding to the weights u^* is

$$\bar{x} = (\ 12;\ 0;\ 0;\ 12\ ) \quad \text{with} \quad f(\bar{x}) = (\ 48;\ 60;\ 12\ ).$$

However,

$$\begin{aligned} &(f(\bar{x}) - f(x^1)).u^1 \\ &= (\ 48-15;\ 60+15;\ 12-75\ ).(\ 1;\ 0{,}2;\ 0{,}8\ ) \\ &= \ -2{,}4. \end{aligned}$$

This means that $(\bar{x},0)$ has been "cut out" of the constraint set (7.27), even though u^1 satisfies Assumption 7.10. □

This example shows that the value for ϕ^{k+1} in (7.21) can be "too large". In fact, $|u_i^* - u_i^1|$ easily satisfied (7.20), and yet the cut

$$(f(x) - f(x^1)).u^1 \geqslant 0$$

still caused $(\bar{x},0)$ to be infeasible (where $\bar{x}$ is the optimal solution). We observe that Assumption 5.8 is the same as Assumption 7.10 at the first iteration for the example above. This did not cause any difficulties in Chapter 5, however, because we did not introduce any additional constraints into the problem on the basis of this information. *It is clear, therefore,* that an additional assumption needs to be made to prevent $(\bar{x},0)$ from being made infeasible by inaccurate information.

Since we do not know $\bar{x}$ at any iteration (unless $x^k = \bar{x}$ and the algorithm terminates), we simply assume the following:

<u>ASSUMPTION</u> 7.14 Given any nondominated solution x^k at iteration k of Algorithm 7.3, assume that the DM gives an estimate of the weights u^k which has the property that

$$(f(\bar{x}) - f(x^k)).u^k > 0,$$

where $\bar{x}$ is any optimal solution to (7.1) (7.2).

This assumption is obviously not checkable and it is clear that it could be violated in practise. We discuss this in the next section.

<u>THEOREM</u> 7.15 Suppose that the constraint set X is bounded and that at each iteration of Algorithm 7.3 the DM satisfies Assumptions 7.10 and 7.14. Then, if the algorithm generates an infinite sequence $\{x^k\}$, there is some subsequence of $\{x^k\}$ which converges to an optimal solution for (7.1) (7.2) as $k \to \infty$.

<u>Proof:</u> Let $\Omega \subset X$ be the set of all optimal solutions to (7.1) (7.2). By Assumption 7.14 all points $\bar{x} \in \Omega$ are feasible for some $y > 0$ in (7.6) (7.7) as $k \to \infty$. As $k \to \infty$, if Ω is the only set of points satisfying (7.6) (7.7) for $y^k > 0$, then the result holds for all convergent subsequences of $\{x^k\}$. However, if there exists $z \notin \Omega$, $z \in X$ which is feasible as $k \to \infty$, then we have

$$(f(z) - f(x^k)).u^k \geq 0 \qquad \forall\, k = 1,2,\ldots \tag{7.29}$$

From Assumption 7.14 we have that

$$(f(\bar{x}) - f(x^k)).u^k > 0 \qquad \forall\, \bar{x} \in \Omega,\ \forall\, k = 1,2,\ldots \tag{7.30}$$

Define the point

$$v = \Theta\bar{x} + (1-\Theta)z \qquad \Theta \in (0,1)$$

where $\bar{x}$ is a point in the compact set Ω which satisfies

$$\|\bar{x} - z\| = \min_{x \in \Omega} \|x - z\|.$$

Then, $v \notin \Omega$, $v \in X$ and

$$(f(v) - f(x^k)).u^k$$

$$= (f(\Theta\bar{x} + (1-\Theta)z) - f(x^k)).u^k$$

$$= \Theta(f(\bar{x}) - f(x^k)).u^k + (1-\Theta)(f(z) - f(x^k)).u^k$$

by the linearity of f

$$> 0 \qquad \text{by (7.29) (7.30) and } \Theta \in (0,1).$$

Therefore, there must exist some subsequence $\{x^{k_j}\}$ of $\{x^k\}$ which has a limit point $\tilde{x}$, say, such that

$$(f(v) - f(x^{k_j})).u^{k_j} \to 0 \quad \text{as} \quad j \to \infty; \tag{7.31}$$

otherwise we have that there exists $v \in X$ with

$$\max\ y = \min_k (f(v) - f(x^k)).u^k \geq \varepsilon > 0$$

which contradicts $y^k \to 0$ as $k \to \infty$. Assume that $\tilde{x}$ is not optimal. Then

$$(f(\bar{x}) - f(\tilde{x})).\nabla U(f(\tilde{x})) > 0 \qquad \text{since } \tilde{x} \notin \Omega$$

$$\text{i.e.} \quad (f(\bar{x}) - f(\tilde{x})).u^*(\tilde{x}) > 0. \tag{7.32}$$

In addition, letting $j \to \infty$ in (7.31), gives

$$(f(v) - f(\tilde{x})).\tilde{u} = 0 \tag{7.33}$$

where

$$\tilde{u} = \lim_{j \to \infty} u^{k_j} = u^*(\tilde{x})$$

by Assumption 7.10 and the fact that $y^{k_j} \to 0$ as $j \to \infty$. Now

$$(f(v) - f(x^{k_j})).u^{k_j}$$

$$= \theta(f(\bar{x}) - f(x^{k_j})).u^{k_j} + (1-\theta)(f(z) - f(x^{k_j})).u^{k_j}.$$

Letting $j \to \infty$, the left hand side tends to zero (from (7.33)), while the right hand side has a limit which is positive (from (7.29), (7.32) and $\theta \in (0,1)$). This contradiction means that $\tilde{x} \in \Omega$ and the theorem is proved. □

<u>REMARK</u> 7.16 We did not make use of (7.21) in the proof of the above theorem. It is sufficient to have that $\phi^{k+1} \to 0$ as $k \to \infty$ in (7.20). However, use was made of (7.21) in Theorem 7.12. From Theorems 7.12 and 7.15, we have that *if X is a bounded set and if the DM satisfies* Assumptions 7.10 and 7.14, then Algorithm 7.3 will either terminate after a finite number of iterations with the DM's most preferred solution, or there is a subsequence of $\{x^k\}$ which converges to the DM's most preferred solution. □

7.4 THE PRACTICAL IMPLEMENTATION OF THE CUTTING PLANE ALGORITHM

In a practical implementation of Algorithm 7.3 on a real problem, the procedure will be stopped after a finite number of iterations. This could be done at the end of iteration k when the DM is presented with a solution x^{k+1} which he considers to be satisfactory - i.e. we have found a satisfactory rather than the most preferred solution. An upper bound on $\|x^{k+1} - \bar{x}\|$, where $\bar{x}$ is the most preferred solution, can be found by solving the problem

$$\begin{aligned} &\underset{x}{\text{maximize}} && \| x - x^{k+1} \| \\ &\text{subject to} && (f(x) - f(x^j)).u^j \geq 0; \qquad j = 1,\dots,k; \\ &&& x \in X; \end{aligned}$$

where we are assuming that $\bar{x}$ satisfies the additional constraints.

It is possible that at some iteration the optimal solution is 'cut out' of the constraint set by inaccurate information regarding the weights. If we assume that the DM's estimates of his marginal rates of substitution (given in the matrix D) are becoming more accurate as the algorithm progresses, then it is more likely that this will happen at earlier, rather than later, iterations. As a result, some constraints need to be dropped from the constraint set (possibly). This could be done at the end of iteration k by looking for the earliest constraints in (7.6) which are binding once problem (7.5) - (7.7) has been solved for (x^{k+1},y^{k+1}). The binding constraints satisfy

$$(f(x^{k+1}) - f(x^j)).u^j = y^{k+1}; \qquad 1 \leq j \leq k.$$

Then drop all the binding constraints j, where $j \ll k$, from (7.6).

Bounds on errors on the responses which are made by the DM in the normalised matrix D can be computed using the ad-hoc procedure which was described in Chapter 6. These will give the limits on the elements of $D^*(x^k)$ which ensure that Assumption 7.10 is satisfied. If the DM has given a consistent matrix D, then there is probably no point (from a practical point of view) in computing the bounds on the elements of D unless the utility function is assumed to be linear (or almost linear?) - see Note 6.21. In general, when U is not linear, $u^*(x^k)$ can be quite different to $u^*(x^{k+1})$. As a result, $D^*(x^k)$ can be quite different to $D^*(x^{k+1})$, even if the inquiry patterns are the same. Therefore, computing the ranges in which the elements of $D^*(x^k)$ are assumed to lie may be of little assistance to the DM when he estimates $D^*(x^{k+1})$. The ranges may be useful, however, when the DM has given an inconsistent matrix D in an attempt to 'iron out' the inconsistencies before the algorithm continues - see Note 6.20.

7.5 CONCLUSION

In this chapter we have proposed a cutting plane method for solving problem (7.1) (7.2) when the (unknown) utility function is assumed to be *concave and twice continuously differentiable.* After an interaction with the DM, a constraint is defined which reduces the set of points of interest. If we assume that the DM's most preferred solution is always inside the reduced constraint set, and that the DM's estimates become increasingly more accurate as the algorithm progresses, then the algorithm will find the DM's most preferred solution as the limit of some subsequence.

An assumption of the form given in Assumption 7.14 needs to be made (or possibly Assumption 7.10 can be tightened). The question arises as to whether a 'better' assumption can be made. It appears to us that 'better' in this case means finding an assumption which does not involve the unknown optimal solution $\bar{x}$.

Musselman and Talavage (1980) also recognise the possibility of $\bar{x}$ being made infeasible in their proposed cutting plane method. They suggest, however, that once a desirable solution has been found, limits for each trade-off ratio at every assessment point of the algorithm should be obtained. "If a DM establishes a range within which a particular trade-off is known to exist, this range can be judged against these limits to indicate to what extent his trade-off uncertainty undermines the final result". (Musselman and Talavage (1980, page 1431)). Zionts and Wallenius (1983) recognise the same problem from a different perspective: viz. the true weights $u^*(\bar{x})$, at the optimal solution $\bar{x}$, may not satisfy the constraints (2.6) (2.7). They suggest that certain (earlier?) constraints on the weights need to be dropped. This is similar to the approach which we suggested in Section 7.4 (although in a different context). The algorithm still needs to be tested on several real problems in order to determine how serious this can be and, if constraints do need to be dropped, which ones they should be. (i.e. how early is earlier?)

Problem (7.1) (7.2) is a multiple objective linear programming problem. Consider the problem

$$\text{maximize} \quad U(f_1(x),\ldots,f_r(x))$$

$$\text{subject to} \quad g(x) \leq 0,$$

where $f_i : R^n \to R^1$ is concave for $i = 1,\ldots,r$, and $g : R^n \to R^m$ is concave.

The proofs of convergence of Algorithm 7.3 did not explicitly use the structure of the constraint set X. All that was required was that X be convex and bounded. Algorithm 7.3 will converge (as the limit of a subsequence) to the DM's most preferred solution (subject to Assumptions 7.10 and 7.14) over any bounded, convex constraint set. In the proofs of convergence, however, the linearity of f was exploited. Using the property

$$f(\theta x + (1-\theta)y) \geq \theta f(x) + (1-\theta)f(y) \qquad \theta \in [0,1]$$

for concave functions, appears to lead to the same conclusions regarding convergence as before.

Finally, further investigations need to be carried out (on real problems) to discover what effects the introduction of a steplength procedure into Algorithm 7.3 (see Remark 7.4 (ii) (iii)) will have on the rate of convergence.

CHAPTER 8 AN APPLICATION TO A PROBLEM IN WILDLIFE MANAGEMENT

Peddie (1984) is concerned with a problem in wildlife management. The aim of his research is to determine the stocking density and mix of animal species which will optimize certain objectives in the Pilanesburg Game Reserve, Bophuthatswana, Southern Africa. Peddie's brief from his supervisors was to collect the relevent data by means of field observations and then to formulate a mathematical model which described the interaction of the animal species with the available resources in the park. The resulting model turned out to be a linear programming problem with multiple objectives. It was decided that Algorithm 7.3 would be used in an attempt to find a compromise solution (or a satisfactory solution), where Peddie would act as the DM since he had a good working knowledge of the problem and knew what kind of solution the management of the park was interested in. In addition to this, he had been mandated by the management to recommend a solution to them for their consideration.

It is not the function of this chapter to comment on the validity of the biological data or the structure of the final multiple objective linear programming problem. We first present the mathematical model of the problem. The relevent data is included in the Appendix for the sake of completeness. Finally, we report on the results of the implementation of Algorithm 7.3 to the problem in the search for a satisfactory solution.

8.1 DEFINITION OF THE PROBLEM

The management of the Pilanesberg Game Reserve are considering stocking the park with 21 species of large herbivores. These are:

1. Elephant	2. White Rhino	3. Black Rhino
4. Giraffe	5. Buffalo	6. Eland
7. Zebra	8. Roan	9. Sable
10. Waterbuck	11. Wildebeest	12. Hartebeest
13. Gemsbuck	14. Kudu	15. Tsessebe
16. Reedbuck	17. Impala	18. Warthog
19. Bushbuck	20. Mountain Reedbuck	21. Springbuck.

Peddie found that the park could be broken down into 10 distinctly different habitats. These are:

1. Grassland
2. Secondary grassland
3. Encroached secondary grassland
4. *Summit savanna*
5. Valley savanna
6. Mesocline savanna (moist hillside)
7. Xerocline savanna (dry hillside)
8. Valley thicket
9. Talus/ kloof thicket
10. Riverlines.

As a result, 210 variables were defined, where

> x_{ij} is the number of animals of species i found in habitat j; $i = 1,\ldots,21$; $j = 1,\ldots,10$.

Each species has a diet consisting of various proportions of 7 food types. These are:

f_1. Browse below 1,5m

f_2. Browse below 2,5m (including f_1)

f_3. Browse below 5,0m (including f_1 and f_2)

f_4. Forbs

f_5. Leaf litter

f_6. Fine (leaf and sheath) grass

f_7. Coarse (stem and inflorescence) grass.

These food types occur in different quantities in each of the habitats.

Objectives

Peddie stated the following 7 objectives:

Maximize

1. The amount of available food consumed
2. The biomass of the animals in the park
3. The number of animals in the park
4. The value of the animals sold for conservation
5. The value of the animals shot for trophies
6. The value of the animals captured and sold live
7 The value of the animals shot and sold as carcasses and skins.

The last 4 objectives are to be realised from the sustained yield (i.e. the rate of increase) of each of the species. The values for each of the objectives over each of the species is given in Table 1 of the Appendix, where, for example, a white rhino (second row) eats

20,6 kgs. of food per day; has a biomass of 1500 kgs.; has a conservation value of R2000, a trophy value of R8500, a live sale value of R2000 and no carcass value.
This necessitates the introduction of 43 additional variables (corresponding to the 43 nonzero entries in columns 3 - 7 of Table 1 in the Appendix):

Y_{im} *denoting the number of animals of species* i which contribute towards objective m; $i = 1,\ldots,21$; $4 \leq m \leq 7$.

Constraints

1. There are constraints on the amount of food available.

i.e. $\sum_{i=1}^{21} f_{ji} x_{ik} \leq F_{jk}$; $\quad j = 1,\ldots,7$; $k = 1,\ldots,10$;

where: f_{ji} is the amount of food of type j eaten by species i per day (given as element i,j of Table 2 in Appendix),

F_{jk} is the total amount of food of type j which is available in habitat k (given as element j,k of Table 3 in the Appendix - divided by 100).

From Table 3 in the Appendix we see that there is no leaf litter (f_5) in habitats 1 and 2. These 2 constraints are excluded and the animals diet in habitats 1 and 2 have their leaf litter component added to the course grass component of their diets in these habitats. (Given in Table 2 of the Appendix.)

2. A minimum number of each species must occur in the solution to ensure a diversity of species (for various reasons).

i.e. $\sum_{k=1}^{10} x_{ik} \geq X_i$; $\quad i = 1,\ldots,21$;

where X_i is given in Table 4 of the Appendix for each species.

3. Each species has a given rate of sustained yield (harvesting) which is given by its annual rate of increase. The numbers allocated to objectives 4 - 7 are constrained by this rate and, in the case of trophies (objective 5), by the rate of trophy production (Table 4 of the Appendix).

$$\text{i.e.} \quad \alpha_i \sum_{k=1}^{10} x_{ik} = \sum_{m=4}^{7} \beta_{im} Y_{im} ; \qquad i = 1,\ldots,21;$$

$$Y_{i5} \leq T_i \sum_{k=1}^{10} x_{ik} ; \qquad i = 1,\ldots,21;$$

where: α_i is the sustained yield of species i (Table 4 in the Appendix),

T_i is the trophy yield of species i (Table 4 in the Appendix - where $T_i = 0$ if species i is not used for trophies), and

$\beta_{im} = 1$ if species i is used for objective m

$= 0$ if it is not (can be seen from Table 1 of Appendix).

4. There may be an upper bound on the number of species i in the park and/or a bound on species i in habitat k.

$$\text{i.e.} \quad \sum_{k=1}^{10} x_{ik} \leq U_i ; \qquad 1 \leq i \leq 21;$$

$$x_{ik} \leq u_{ik} ; \qquad 1 \leq i \leq 21; \quad 1 \leq k \leq 10.$$

Peddie found difficulty in obtaining most of these values (U_i and u_{ik}) beforehand in an information vacuum. As a result, it was decided to include these *constraints interactively* during the decision making process as they were needed.i.e. the current set of weights are used to find a solution (including all the previously generated cutting planes). If this solution had too many animals of a particular species in a habitat, or in the park, then Peddie made estimates on the upper bounds ('density' constraints) and these were included. The problem was resolved (sometimes further new bounds needed to be included) until a 'feasible' solution was achieved. At this stage the trade-off information was obtained and the algorithm was continued with the new cutting plane included in the constraint set.

The upper bound right hand sides (viz. U_i and u_{ik}) can be found in Table 5 of the Appendix. The second number in each block indicates the stage at which the bounds were included in the constraint set.

5. At a certain stage of the algorithm, Peddie included bounds on the number of animals which could be used for trophy purposes.

i.e. $Y_{i5} \leq 30$ for $i = 7,11,12,13,14,17,18,19,20$.

No solution ever arose which made it necessary to impose a bound on Y_{25}. The stage at which these additional 9 constraints were included is indicated in the next section.

8.2 RESULTS OF THE COMPUTER RUNS AND DECISION MAKER INTERACTIONS

ITERATION 0

The aim at the first stage is to find a nondominated, feasible solution. The weights of the objective functions were all set to 1 and the objective functions simply added. The problem

$$\text{maximize} \quad y$$

$$\text{subject to} \quad \sum_{m=1}^{3} f_m(x) + \sum_{m=4}^{7} f_m(Y) \geq y$$

was solved. Initially there were 124 constraints. (i.e. 68 food, 21 *minimum number*, 21 *sustained yield* + 10 *trophy*, 3 upper bounds – on species 1,2 and 3 in Table 5 of Appendix – and 1 sum of objectives.)

A solution was computed and this was found to have too many of certain species either in the park or in a particular habitat. Upper bounds were then included and the problem solved again. This procedure was continued until a 'feasible' solution was found. The upper bounds which were included in this process at this stage can be found in Table 5 of the Appendix marked with a [1]. The solution is given as Solution 1 in Table 6 of the Appendix.

The values of the 7 objective functions at this solution are:

Food	96 464 (kgs)
Biomass	4 607 970 (kgs)
Numbers	21 729
Conservation	1 596 500 (R)
Trophy	527 800 (R)
Live Sale	414 110 (R)
Carcass	21 600 (R).

ITERATION 1

Peddie gave the following indifference trade-offs at this solution (rounded to 4 decimal places). The matrix gives the normalised trade-offs. The actual trade-offs which were given for each pairwise comparison can be obtained by multiplying each row by the number on the same row outside the matrix (subject to the rounding error from normalisation).

$$\begin{array}{ccccccccc} 1 & 2 & 3 & 4 & 5 & 6 & 7 & & \\ \end{array}$$

$$\left(\begin{array}{ccccccc} 1 & 2 & 3 & 4 & 5 & 6 & 7 \\ -0{,}008 & 1 & & & & & \\ -0{,}1333 & & & & & 1 & \\ & 1 & -0{,}0067 & & & & \\ & & & & & 1 & -0{,}04 \\ & & & 1 & -0{,}2 & & \\ & & -0{,}0462 & & 1 & & \\ & & & 1 & & & -0{,}008 \end{array}\right) \begin{array}{c} \\ 500\ 000 \\ 75\ 000 \\ 600\ 000 \\ 50\ 000 \\ 500\ 000 \\ 130\ 000 \\ 500\ 000 \end{array}$$

Objectives (1,2), (1,6),(2,3),(6,7),(4,5),(3,5) and (4,7) were chosen for the pairwise comparisons.

The weights u^1 were computed to be (to 4 decimal places)

(0,2378; 0,0028; 0,5643; 0,0058; 0,0262; 0,0316; 0,7895).

The objective constraint used at Iteration 0 was dropped from the constraint set (since the weightings were arbitrary) and the following problem was then solved:

$$\text{maximize} \quad y$$

$$\text{subject to} \quad \sum_{m=1}^{3} u_m^1 f_m(x) + \sum_{m=4}^{7} u_m^1 f_m(Y) - y \geqslant \sum_{m=1}^{7} u_m^1 f_m(x^1, Y^1)$$

(including all the previous constraints).

The solution which was found, after additional upper bounds had been added (indicated by a ^{2} in Table 5 of the Appendix), is given as Solution 2 in Table 6 of the Appendix. The values of the objective functions are given as $f(x^2)$ in Table 8 of the Appendix.

ITERATION 2

At $f(x^2)$, Peddie gave the indifference trade-off matrix found in Table 7 of the Appendix. The weights were computed to be

$$u^2 = (0,2310;\ 0,0035;\ 0,2477;\ 0,0693;\ 0,3588;\ 0,8669;\ 0,0106).$$

These weights were then used to construct another cutting plane which was included in the constraint set. The solution which was computed (including additional bounds) is given as Solution 3 in Table 6 of the Appendix. The objective functions at this solution are given as $f(x^3)$ in Table 8 of the Appendix.

ITERATION 3

At $f(x^3)$, Peddie gave the indifference trade-off matrix found in Table 7 of the Appendix. The weights were computed to be

$$u^3 = (0,7172;\ 0,0193;\ 0,6555;\ 0,1086;\ 0,1093;\ 0,1668;\ 0,0639).$$

Another cutting plane was then incorporated into the constraint set. The solution which was computed (including additional bounds) is given as Solution 4 in Table 6 of the Appendix and the values of the objective functions at this solution are given as $f(x^4)$ in Table 8 of the Appendix. It was at this stage (i.e. while Solution 4 was being determined) that Peddie decided that he needed to include the additional 9 trophy constraints mentioned in Section 8.1.

ITERATION 4

At $f(x^4)$, Peddie gave the indifference trade-off matrix found in Table 7 of the Appendix. The weights were computed to be

$$u^4 = (0,8775;\ 0,0103;\ 0,3059;\ 0,0227;\ 0,0234;\ 0,1177;\ 0,3534).$$

A cutting plane was included in the constraint set and the solution which was computed (including additional bounds) is given as Solution 5 in Table 6 of the Appendix. The values of the objective functions at this solution are given as $f(x^5)$ in Table 8 of the Appendix.

ITERATION 5

At this stage, Peddie decided to drop objectives 1 and 2 from the problem (i.e. set their weights to zero) because he felt that the values that had been generated so far for these two objectives were more or less satisfactory. In addition, it appeared that the remaining objectives would ensure that the first two would retain reasonable levels. As a result, he estimated his indifference trade-offs at $f(x^5)$ using pairwise comparisons between objectives 3 – 7 only.(See Table 7 of the Appendix.) The weights were computed to be

$$u^5 = (0;\ 0;\ 0,9361;\ 0,006;\ 0,3287;\ 0,1192;\ 0,0379).$$

A further cutting plane was then included in the constraint set and the solution which was computed is given as solution 6 in Table 6 of the Appendix. This solution includes a further reduction on the upper bound previously imposed on species 18. (See Table 5 of the Appendix.) The values of the objective functions at this solution are given as $f(x^6)$ in Table 8 of the Appendix. It can be seen from the table that objectives 1 and 2 did retain reasonable levels (in comparison to the previous results).

ITERATION 6

At $f(x^6)$ Peddie indicated that he felt that all the objectives had, for the first time, all achieved desirable levels and that this solution was the most satisfactory to date. It was decided, however, to continue the algorithm in a search for a 'better' solution. The indifference trade-off matrix which was given (again between objectives 3 - 7) led to the weights

$$u^6 = (0;\ 0;\ 0{,}9554;\ 0{,}0711;\ 0{,}2223;\ 0{,}1462;\ 0{,}1070).$$

The cutting plane was included in the constraint set and the solution which was computed is given as Solution 7 in Table 6 of the Appendix. The values of the objectives are given as $f(x^7)$ in Table 8 of the Appendix.

ITERATION 7

Peddie indicated that he preferred $f(x^6)$ to $f(x^7)$. Indifference trade-offs which were taken at $f(x^7)$ led to the weights

$$u^7 = (0;\ 0;\ 0{,}901;\ 0{,}0238;\ 0{,}15;\ 0{,}2254;\ 0{,}3382).$$

The solution which was computed (including the new cutting plane) is given as Solution 8 in Table 6 and the values of the objective functions is given as $f(x^8)$ in Table 8 of the Appendix. Peddie still preferred $f(x^6)$ at this point.

ITERATION 8

At this stage, it was decided to return to $f(x^6)$ and to look at a convex combination of the objective functions between $f(x^6)$ and $f(x^7)$ and between $f(x^6)$ and $f(x^8)$. A step length of 0,1 was chosen and two tables, each with 11 columns (2 end points + 9 intermediate points), were constructed. Peddie chose the third column of the second table (i.e. beween $f(x^6)$ and $f(x^8)$) as the most preferred column. This column occurred where the step length was 0,2. As a result of this, levels for objectives 2 - 7 were set at

Biomass	4 285 000 (kgs)
Numbers	27 000
Conservation	997 000 (R)
Trophy	150 000 (R)
Live Sale	289 800 (R)
Carcass	399 500 (R)

where Food Consumption was to be maximized. The lower levels on Numbers and Trophy were set by Peddie as he was prepared to sacrifice these two objectives in order to increase some (or all?) of the remaining objectives. The cutting planes which had been generated so far were dropped from the constraint set and the levels for the 6 objective functions above were included as constraints. The solution which was obtained is given in Table 9 of the Appendix and the objective functions were computed to be

Food	97 336
Biomass	4 420 995
Numbers	30 565
Conservation	1 000 000

Trophy	150 000
Live Sale	289 720
Carcass	399 430.

The last two values are due to the solution being rounded to the nearest integers. Peddie felt that this was the most satisfactory solution to date in terms of the objectives. He was, however, dissatisfied with the mix of animals (i.e. the decision variables x) *which made up this* solution. Various attempts were made to find a more satisfactory mix of animals by settings bounds on 6 of the objectives and maximizing the *remaining objective*. This approach failed to find a suitable solution (in terms of the decision variables).

It was then decided to consider a convex combination of the decision variables which constituted Solution 6 of Table 6 and Table 9 of the Appendix (since Peddie preferred the animal mix which was found in Solution 6). A step length of 0,2 was chosen and a *table with* 6 columns was constructed. Peddie chose the 4^{th} column as the most desirable. This coincided with a step length of 0,6. As a result, the levels *of* the objective *functions were* adjusted to:

Food	96 500	(97350 - 95275)*0,6 + 95275 = 96520
Biomass	4 300 000	dropped from 4 389 041
Numbers	27 000	still kept low
Conservation	980 000	dropped from 985 360
Trophy	150 000	still kept low
Live Sale	270 000	dropped from 280 952
Carcass	400 000	dropped from 408 136.

The lower bounds for the species were then raised in accordance with

Peddie's chosen mix of the decision variables. Certain bounds were not raised to their full extent, however, in order for some competing species (with respect to food) to improve their values. The final bounds can be found in Table 10 of the Appendix.

It was decided to maximize Conservation and to set the remaining objectives at the above levels. The solution which was obtained is given in Table 10 of the Appendix. The objective function values at this solution are:

Food	95 993 (kgs)	due to rounding error
Biomass	4 335 780 (kgs)	
Numbers	32 029	
Conservation	1 085 750 (R)	
Trophy	150 000 (R)	
Live Sale	269 800 (R)	due to rounding error
Carcass	399 875 (R)	due to rounding error.

Peddie felt that he had a desirable solution, both in terms of the objective functions and the decision variables. As a result, the procedure was terminated.

8.3 GENERAL COMMENTS

1. *Indifference trade-offs* between objectives i and j were obtained from Peddie by presenting him with 3 vectors at the solution x^k: viz.

$$\begin{array}{lll} f_1(x^k) & f_1(x^k) & f_1(x^k) \\ \cdot & \cdot & \cdot \\ f_i(x^k) & f_i(x^k) + A & f_i(x^k) - A \\ \cdot & \cdot & \cdot \\ f_j(x^k) & f_j(x^k) - B & f_j(x^k) + B \\ \cdot & \cdot & \cdot \\ f_7(x^k) & f_7(x^k) & f_7(x^k), \end{array}$$

where $A, B > 0$. A and B were initially chosen to be somewhere between 10 and 20% of $f_i(x^k)$ and $f_j(x^k)$ respectively. If Peddie indicated that he preferred one of the points, or there was a point that he disliked, then A and/or B was adjusted until he *was indifferent* between the 3 vectors.

2. As can be seen from Table 7 in the Appendix, the indifference trade-offs differ markedly at different solutions. As a result, we assumed that the implicit utility function was concave and chose to implement Algorithm 7.3 rather than Algorithm 4.1.

3. When the interactive procedure began, Peddie took approximately 45 minutes to estimate the first set of indifference trade-offs. It was found that *the time for this interaction* decreased markedly (down to about 15 minutes) as the number of interactions increased. It should be reported that several false starts were made to solve this problem. After a few interactions had been performed, it was

discovered that errors had been made in the formulation of the problem. After these had been corrected, the algorithm was restarted from the beginning. At the final stage, because of the previous interactions, Peddie took less than 10 minutes to estimate the indifference trade-offs at each interaction.

4. The interactions were performed between Peddie and a human analyst (viz. ourselves). It is quite probable that this influenced the time that was taken by Peddie to give the indifference trade-offs as we found that we could often predict what his reaction would be to certain trade-offs (as a result of learning from his reactions to previous trade-offs).

5. 3 types of upper bounds were included during the implementation of the algorithm.

   a) Bounds were included when it was found that too many of a species occurred in a habitat and/or the park. These were included before the indifference trade-offs were made.

   b) The upper bound on species 18 was reduced from 8000 to 5000.

   c) Additional bounds were incorporated for animals which are used for trophy purposes.

   (b) and (c) imply that indifference trade-offs were made at 'infeasible' solutions. It was felt, however, that this did not invalidate the estimates which had been made previously (similar to establishing the full utility function - Section 2.3.2). The worry was that the previously generated cutting planes might have led to there being no feasible solutions when the new bounds were included, or that the most preferred 'feasible' solution might have been cut out of the constraint set. It turned out (for this problem) that neither of these two possibilities did arise. Firstly, because a feasible solution (with the cutting plane constraints included) could be found after the new bounds were added, and secondly, because the cutting plane constraints were omitted from the final stages of the solution procedure. However, the effect of taking trade-offs at essentially nonfeasible solutions may well have had an effect on the speed of convergence of the Algorithm to a satisfactory solution (i.e. we suspect that it may have speeded it up).

6. Once all the necessary constraints had been incorporated, it would have been quite simple to restart the algorithm from the beginning and to report the results for a well defined problem. We have, however, opted to follow the 'ill defined' route as we believe that in problems of this type, difficulties of the kind which we have mentioned do tend to arise. This necessitates a pragmatic approach to the solution procedure and, in our case, this made things easier for our DM who was also resposible for the mathematical formulation of the problem. If Algorithm 7.3 is used as the basis of the solution procedure, and the constraints are being included interactively, then care needs to be exercised in case the problems mentioned in the last point do arise (which can happen quite easily).

7. Given the size of the problem which was solved, the interactive algorithm found a satisfactory solution (in terms of the objective function values) fairly quickly. Although a few more interactions were performed (which led to solutions which were not as 'good' as $f(x^6)$) using Algorithm 7.3, this allowed time and provided a framework for further fine-tuning of the solution on a more ad hoc basis.

8.4 CONCLUSIONS

The problem which we considered in this chapter has two interesting features. These are:

a) the problem was not 'well defined' at the start in the sense that all the constraints are known beforehand. This led to an 'interactive' formulation of the problem where certain constraints were included as it was found necessary to do so.

b) at the last stage, attention was focussed on finding suitable values for the decision variables. This implies that there was some implicit objective of 'achieving a suitable mix of animal species'. It is, however, difficult to see how this could have been included in the formulation of the problem at the start.

The solution procedure appears to have performed fairly efficiently on the problem. Algorithm 7.3 found a satisfactory solution after 5 iterations. Since the theoretical implementation of Algorithm 7.3 requires increasing accuracy from the DM at later iterations, and there is always the chance that the most preferred solution has been 'cut' out of the constraint set by inaccurate information, we adopted an approach in the final stages which is similar to interactive goal programming.

From the experience which was gained while solving the problem, it is likely that the switching over from a utility based algorithm to a GP approach would still have been desirable to our DM if some other utility based agorithm had been used instead of Algorithm 7.3. It appears that for (some) MCDM problems, an argument can be made in favour of using a utility based interactive algorithm in the initial stages of the solution procedure to find satisfactory values for the objectives, and then switching over to an interactive GP approach for further 'fine tuning' of the solution.

CONCLUSION

In this thesis we have paid particular attention to linear programming problems with more than one objective function. Certain basic results from utility theory and decision analysis were presented and used in the work which was developed later on. A survey of the most important methods which have been cited in the literature was given and some of these methods were discussed in a reasonable amount of detail.

A new method of estimating the direction of the marginal utility vector was proposed. This is based, to a large extent, on the approach which was suggested by Rosinger (1981). In this method, the DM is asked to chose pairs of objective functions that he feels he is able to estimate indifference trade-offs for. These indifference trade-offs are then included in an *indifference trade-off* matrix and this is used to compute a vector which gives the estimated direction of the marginal utility vector. The approach also enables the DM to give conflicting information (to some extent) when he makes these estimates. It is hoped that this approach will help to reduce the difficulties that DMs have been found to experience when using interactive methods which require estimates of the marginal rates of substitution between objectives to be made (eg. Wallenius (1975)). We found that when a DM did apply this approach (reported in Chapter 8) on a real problem which required 7 objective functions to be compared somehow, he experienced some difficulty initially (and took some time over) giving the indifference trade-offs. As the number of interactions increased, however, he found the procedure increasingly easy to apply and his response time decreased markedly as a result.

Two interactive algorithms for solving multiple objective linear programming problems were then proposed. The first of these algorithms is simply a restatement of the simplex algorithm, where the estimated weights (marginal utility vector) are used to reduce the multiple objective functions to a single one. This is suggested for solving problems where it is believed that the DM's preference structure can be reasonably represented as a linear sum of the objective functions. The second of the algorithms, when the DM's preference structure is thought to be more complex (assumed concave), uses the estimated weights to construct cutting planes which are then included in the constraint set. Provided that the estimates of the weights satisfy certain assumptions, both of the algorithms were shown to converge to the DM's most preferred solution.

Intervals on the estimated weights can be computed as a consequence of the assumptions which were imposed in order to guarantee the convergence of the algorithms. It was shown, by means of an example, how these intervals can be calculated in a practical manner. Since the estimated wieghts are directly responsible for the performance of the algorithms (i.e whether they converge and, if so, the rate of convergence) these intervals can be presented to the DM as an aid in the decision making process.

When the DM is using the approach which we have suggested, where indifference trade-offs are given (and the weights are then computed from these), intervals need to be computed on the original trade-offs made by the DM (rather than the weights). It was shown how this could be achieved. In order to do so, however, additional assumptions needed to be imposed on the estimated trade-off ratios and it was shown that certain problems can arise as a result of the assumptions which were made and the line of reasoning which was followed. The approach which was followed, however, did give reasonable intervals on the trade-off ratios and it is suggested that these might be of assistance to the DM during the decision making process, especially when he has given conflicting information in the indifference trade-off matrix.

If different assumptions are imposed so as to ensure that the algorithms converge, then these will lead to different bounds being calculated. Further research needs to be conducted to establish whether there are 'better' assumptions which can be made, and further investigations should be carried out in order to determine whether the bounds (on the weights and/or indifference trade-offs) are of any practical use and do, in fact, assist (hinder? confuse?) the decision making process.

In the final chapter of the thesis, the proposed cutting plane algorithm is implemented as the basis of a solution procedure to solve a linear programming problem with multiple objectives which arose in a wildlife management context. It is reported that the method found a satisfactory solution in only 5 iterations and that a pragmatic utilization of the algorithm led the DM to the final solution after a few more interactions. The algorithm still needs to be tested on some more real problems before any generals claims about its efficiency or robustness can be made, and any conclusions can be drawn about the relative success (or lack of success) when compared to the other interactive approaches which were surveyed in Chapter 2.

It is generally accepted that the successful implementation of an algorithm on a programming problem with multiple objectives is problem and/or DM dependent. Gershon and Duckstein (1983) treat the choice of a MCDM method itself as a multiple criteria decision making problem and develop an algorithm for making this choice.

From our own experiences with a real practical problem (Chapter 8), a pragmatic and sometimes ad hoc solution procedure often needs to be adopted. This could entail the switching from one algorithm which is suitable in the early stages of the solution procedure to another one (or more?) which appears to be suitable at a later stage. This may be especially true when the problem is being solved over several days (rather than hours). It appears, therefore, that for certain multiple objective programming problems, not only is the choice of algorithm a difficulty, but that this could be further complicated by switching from one algorithm to another one during the decision making process. We feel that even if the decision analyst has decided on a certain method, he should approach the overall solution procedure with an open mind and be prepared to adapt or change the method he uses if it becomes apparent that this could facilitate the entire procedure. PRAGMATISM IS THE KEY.

APPENDIX

TABLE 1

Species	Food (kg)	Biomass (kg)	Conservation (R)	Trophy (R)	Live Sale (R)	Carcass (R)
Elephant	24,2	1800				2200
White Rhino	20,6	1500	2000	8500	2000	
Black Rhino	14,9	860	15000			
Giraffe	14,0	750			2000	920
Buffalo	8,9	450	4000			
Eland	7,3	360			500	440
Zebra	5,7	210		450	220	250
Roan	4,5	200	6000			
Sable	4,2	185	2500			
Waterbuck	3,8	165	750			
Wildebeest	3,7	160		340	125	195
Hartebeest	3,7	155		500	220	190
Gemsbuck	3,6	150		500	200	180
Kudu	3,3	135		500	220	160
Tsessebe	2,9	120	1500			
Reedbuck	1,7	50			100	60
Impala	1,2	40		100	50	50
Warthog	1,2	40		100	25	50
Bushbuck	1,1	30		200		35
Mountain Rdbck	1,0	28		200	60	35
Springbuck	1,0	28	300			

TABLE 2

Herbivore diet constituents: daily consumption (kgs. dry mass): dry season.

Species	Food Type						
	Browse ≤ 1,5m	Browse ≤ 2,5m	Browse ≤ 5,0m	Forbs	Leaf Litter	Grass, fine	Grass, coarse
Elephant			20,57	1,21	0,48	1,21	0,73
White Rhino				1,03		11,33	8,24
Black Rhino		12,66	12,66	1,49	0,45		0,30
Giraffe			13,72	0,14	0,14		
Buffalo				0,09	0,09	5,16	3,56
Eland		6,20	6,20	0,73		0,22	0,15
Zebra						3,13	2,57
Roan		0,37	0,37			2,56	1,57
Sable		0,21	0,21			2,73	1,26
Waterbuck		0,08	0,08			2,28	1,44
Wildebeest				0,04		2,96	0,70
Hartebeest				0,08		2,66	0,96
Gemsbuck		0,11	0,11	0,07		1,98	1,44
Kudu		2,31	2,31	0,83	0,16		
Tsessebe						2,32	0,58
Reedbuck						1,53	0,17
Impala	0,50	0,50	0,50	0,08	0,20	0,25	0,17
Warthog				0,06		0,84	0,30
Bushbuck	0,71	0,71	0,71	0,22	0,06	0,07	0,05
Mountain Rdbck				0,01		0,89	0,10
Springbuck	0,50	0,50	0,50	0,10	0,05	0,25	0,10

TABLE 3

Biomass of available food over a 100 day period (1982).

Food Type	HABITAT										
	1	2	3	4	5	6	7	8	9	10	Total
f_1	1157	765	6999	3307	44000	66944	345754	139167	709518	55456	1373067
f_2	1575	1119	9338	5094	81940	152180	751759	287949	1166308	142901	2600163
f_3	2441	1782	13224	9301	163331	366436	1642627	621442	1981353	344103	5146040
f_4	52979	35968	28107	224372	171990	140394	251154	91022	59406	43321	1098713
f_5	0	0	18125	116325	626592	320736	1853478	546394	1099048	311815	4892513
f_6	105183	138426	106488	431690	468406	439470	1461834	171475	76457	140980	3540409
f_7	128509	169238	130239	527472	506617	536616	1786989	209398	93240	172310	4260628

* These figures must be divided by 100 to get the maximum daily food consumption

TABLE 4

Species	Minimum Nos.	Sustained Yield (%)	Trophy Yield (%)
Elephant	40	4	
White Rhino	120	8	4
Black Rhino	30	8	
Giraffe	100	10	
Buffalo	200	10	
Eland	200	15	
Zebra	200	10	10
Roan	200	10	
Sable	200	10	
Waterbuck	200	15	
Wildebeest	200	20	5
Hartebeest	200	20	5
Gemsbuck	200	15	5
Kudu	250	15	5
Tsessebe	150	10	
Reedbuck	100	15	
Impala	500	20	5
Warthog	150	25	5
Bushbuck	150	25	5
Mountain Rdbck	300	25	5
Springbuck	150	20	

TABLE 5

ANIMAL DENSITIES AND UPPER BOUNDS.

	HABITATS				
SPECIES	4	6	7	9	BOUNDS
1					150[0]
2					300[0]
3			50[1]		150[0]
4		75[1]	150[1]	50[1]	
5					1500[1]
6				250[2]	
7					
8					500[1]
9					500[1]
10	25[1]	50[1]	50[1]		1000[1]
11					
12					4000[2]
13			2500[4]		3500[5]
14					
15					500[1]
16	25[1]				
17		600[2]	1500[2]		10000[2]
18	500[4]	750[5]	1000[2]		8000[2] 5000[6]
19		300[2]	400[2]		3000[2]
20			2500[3]		6000[3]
21	50[1]	100[1]	150[1]		3000[1]

TABLE 6

Results of the different Computer Runs

SPECIES	Solution 1 x_i	Y_{i4}	Y_{i5}	Y_{i6}	Y_{i7}	Solution 2 x_i	Y_{i4}	Y_{i5}	Y_{i6}	Y_{i7}
1	150				6	150				6
2	300	12	12	0		120	0	5	5	
3	150	12				30	2			
4	730			73	0	730			0	73
5	1500	150				200	20			
6	932			140	0	454			0	68
7	200		20	0	0	200		0	0	20
8	500	50				200	20			
9	500	50				200	20			
10	1000	150				200	30			
11	200		10	0	30	200		0	0	40
12	1784		357	0	0	4000		0	0	800
13	5255		263	525	0	5000		0	0	705
14	3828		191	383	0	3591		0	0	539
15	500	50				150	15			
16	100			15	0	100			0	15
17	500		25	75	0	9741		0	0	1948
18	150		8	0	30	8000		0	0	2000
19	150		8		30	1354		0		338
20	300		15	60	0	300		0	0	75
21	3000	600				150	30			

	Solution 3					Solution 4				
1	150				6	150				6
2	300	0	12	12		300	0	12	12	
3	100	8				150	12			
4	730			73	0	730			73	0
5	200	20				1500	150			
6	492			0	74	287			43	0
7	200		0	0	20	200		20	0	0
8	200	20				500	50			
9	200	20				500	50			
10	200	30				200	30			
11	200		0	0	40	200		10	0	30
12	4000		0	0	800	200		30	10	0
13	1102		0	0	165	4350		30	623	0
14	3659		0	435	114	3716		30	527	0
15	150	15				500	50			
16	100			15	0	100			15	0
17	9916		0	0	1983	7187		30	0	1407
18	8000		0	0	2000	7283		30	0	1791
19	727		36		145	600		30		120
20	6000		300	1200	0	600		30	120	0
21	150	30				3000	600			

TABLE 6 (Continued)

	Solution 5					Solution 6				
SPECIES	x_i	Y_{i4}	Y_{i5}	Y_{i6}	Y_{i7}	x_i	Y_{i4}	Y_{i5}	Y_{i6}	Y_{i7}
1	150				6	150				6
2	120	0	5	5		300	0	12	12	
3	150	12				150	12			
4	730			73	0	730			73	0
5	1500	150				745	74			
6	297			0	45	289			0	43
7	200		0	0	20	300		30	0	0
8	500	50				500	50			
9	500	50				500	50			
10	200	30				200	30			
11	200		0	0	40	600		30	0	90
12	1774		0	0	355	2760		30	0	522
13	3500		0	0	525	3500		30	0	495
14	3694		0	0	554	3714		30	405	122
15	150	15				150	15			
16	100			0	15	100			15	0
17	8008		0	0	1602	9916		30	0	1953
18	8000		0	0	2000	5000		30	0	1220
19	700		30		145	700		30		145
20	300		15	0	60	600		30	120	0
21	2196	439				288	58			

	Solution 7					Solution 8				
1	150				6	150				6
2	300	0	12	12		300	0	12	12	
3	150	12				150	12			
4	730			73	0	730			73	0
5	1500	150				950	95			
6	285			0	43	321			0	48
7	200		20	0	0	200		20	0	0
8	500	50				500	50			
9	500	50				200	20			
10	1000	150				200	30			
11	200		10	0	30	600		30	0	90
12	1483		30	0	267	2801		30	0	597
13	600		30	0	60	600		30	0	60
14	3837		30	0	545	3793		30	539	0
15	500	50				150	15			
16	100			15	0	100			15	0
17	7205		30	0	1411	7207		30	0	1411
18	5000		30	0	1220	5000		30	0	1220
19	700		30		145	700		30		145
20	5203		30	1271	0	6000		30	1470	0
21	3000	600				3000	600			

TABLE 7

INDIFFERENCE TRADE-OFFS GIVEN BY THE D.M.

OBJECTIVES 1	2	3	4	5	6	7	SCALING FACTORS
			Trade -offs at $f(x^2)$				
-0,0171	1						700 000
	1	-0,0143					700 000
		1			-0,2857		7 000
			-0,15			1	200 000
				-0,03		1	200 000
	1			-0,008			500 000
-0,3			1				50 000
			Trade-offs at $f(x^3)$				
-0,2333					1		60 000
	1	-0,0233					300 000
		-0,1667	1				30 000
			-0,6			1	50 000
		-0,1667		1			30 000
	1				-0,14		500 000
-0,0875						1	80 000
			Trade-offs at $f(x^4)$				
-0,025			1				400 000
	1			-0,05			400 000
			1			-0,0667	300 000
					1	-0,3333	60 000
		-0,075		1			40 000
-0,02	1						500 000
	1	-0,04					100 000
			Trade-offs at $f(x^5)$				
		-0,35		1			20 000
			1		-0,05		400 000
					-0,4286	1	70 000
				-0,075		1	80 000
		-0,1333			1		30 000

TABLE 8

VALUES OF THE OBJECTIVE FUNCTIONS.

	SOLUTIONS						
OBJECTIVES	$f(x^2)$	$f(x^3)$	$f(x^4)$	$f(x^5)$	$f(x^6)$	$f(x^7)$	$f(x^8)$
FOOD	94 727	91 167	96 606	95 499	95 275	93 273	90 337
BIOMASS	4 101 385	4 017 535	4 422 080	4 260 788	4 339 984	4 247 344	4 068 050
NUMBERS	35 070	36 776	32 253	32 969	31 192	33 143	33 652
CONSERVATION	334 000	424 000	1 482 500	1 381 700	963 400	1 572 500	1 135 000
TROPHY	42 500	169 200	177 400	51 500	188 700	177 400	184 200
LIVE SALE	10 000	339 200	442 940	156 000	267 800	247 760	378 280
CARCASS	700 695	462 725	183 150	484 565	421 195	323 325	312 725

TABLE 9

SOLUTION DURING ITERATION 8

SPECIES	x_i	Y_{i4}	Y_{i5}	Y_{i6}	Y_{i7}
1	150				6
2	300	0	12	12	
3	150	12			
4	730			73	0
5	1 036	104			
6	283			0	42
7	1 278		0	0	128
8	500	50			
9	200	20			
10	200	30			
11	200		0	0	40
12	1 908		18	0	364
13	3 500		30	0	495
14	3 787		30	521	17
15	150	15			
16	100			15	0
17	9 916		0	0	1983
18	5 000		0	0	1250
19	727		30		152
20	300		15	60	0
21	150	30			

TABLE 10 THE FINAL SOLUTION

SPECIES	BOUNDS	x_i	Y_{i4}	Y_{i5}	Y_{i6}	Y_{i7}
1	140	150				6
2	290	300	0	12	12	
3	140	145	12			
4	700	730			73	0
5	900	946	95			
6	280	280			0	42
7	800	932		0	0	93
8	400	500	50			
9	300	300	30			
10	300	300	45			
11	350	350		0	0	70
12	1 000	1 309		12	0	250
13	1 000	3 500		30	0	495
14	3 500	3 783		30	305	232
15	300	300	30			
16	300	300			45	0
17	7 500	9 004		0	0	1800
18	150	5 000		0	0	1250
19	700	700		30		145
20	2 000	2 000		30	470	0
21	1 200	1 200	240			

REFERENCES

Avriel M (1976). *Nonlinear Programming: Analysis and Methods.* Prentice Hall, Englewood Cliffs, New Jersey.

Belensen S M and Kapur K C (1973). An Algorithm for Solving Multicriteria Linear Programming Problems with Examples. *Operational Research Quarterly,* Vol. 24, No. 1, pp 65 - 77.

Bell D E, Keeney R L and Raiffa H (eds.) (1977). *Conflicting Objectives in Decisions.* International Series in Applied Systems Analysis, Vol. 1, John Wiley and Sons.

Benayoun R, de Montgolfier J, Tergny J and Laritchev O (1971). Linear Programming with Multiple Objective Functions: Step Method (STEM). *Mathematical Programming,* Vol. 1, No. 3, pp 366 - 375.

Chankong V and Haimes Y Y (1977). The Interactive Surrogate Worth Trade-Off (ISWT) Method for Multiobjective Decision Making. *Multiple Criteria Problem Solving,* (S Zionts ed.), Springer, Berlin.

Cochrane J L and Zeleny M (eds.) (1973). *Multiple Criteria Decision Making.* University of South Carolina Press, Columbia S.C.

Cohon J L (1978). *Multiobjective Programming and Planning.* Academic Press, New York.

Dauer J P and Kreuger R J (1977). An Iterative Approach to Goal Programming. *Operational Research Quarterly,* Vol. 28, No. 3, pp 671 - 681.

de Samblannckx S, Depraetere P and Muller H (1982). Critical Consideration Concerning the Multicriteria Analysis by the Method of Zionts and Wallenius. *European J. of Operational Research,* 10, pp 70 - 76.

Dyer J (1972). Interactive Goal Programming. *Management Science,* Vol. 19, No. 1, pp 62 - 70.

Dyer J (1973a). An Impirical Investigation of a Man-Machine Interactive Approach to the Solution of the Multiple Criteria Problem. *Multiple Criteria Decision Making,* Cochrane and Zeleny (eds.), pp 202 - 216.

Dyer J (1973b). A Time Sharing Computer Program for the Solution of the Multiple Criteria Problem. *Management Science*, Vol. 19, No. 12, pp 1379 - 1383.

Dyer J (1974). The Effect of Errors in the Estimation of the Gradient in the Frank Wolfe Algorithm, with Implications for Interactive Programming. *Operations Research*, Vol. 22, No. 1, pp 675 - 689.

Evans J P and Steuer R E (1973). Generating Efficient Extreme Points in Linear Multiple Objective Programming. *Multiple Criteria Decision Making*, Cochrane and Zeleny (eds.), pp 349 - 365.

Fandel G and Gal T (eds.) (1979). *Multiple Criteria Decision Making Theory and Applications*. Springer Verlag, Lecture Notes in Economics and Mathematical Systems, Vol. 177.

Ferguson C E (1969). *Microeconomic Theory*. Richard D Irwin Inc., Irwin Series on Economics.

Fichefet J (1976). GPSTEM: An Interactive Multi-Objective Optimization Method. *Progress in Operations Research*, A Prékopa (ed.), Vol. 1, North Holland, pp 317 - 332.

Frank M and Wolfe P (1956). An Algorithm for Quadratic Programming. *Naval Research Logistics Quarterly*, 3, 95 - 110.

Geoffrion A, Dyer J and Feinberg A (1972). An Interactive Approach for Multicriteria Optimization with an Application to the Operation of an Academic Department. *Management Science*, Vol. 19, No. 4, pp 357 - 368.

Gershon M and Duckstein L (1983). An Algorithm for Choosing of a Multiobjective Technique. *Essays and Surveys on Multiple Criteria Decision Making*, Hansen R (ed). Springer Verlag, Lecture Notes in Economics and Mathematical Systems, Vol. 209.

Goicoechea A, Hansen D R and Duckstein L (1982). *Multiobjective Decision Analysis with Engineering and Business Applications*. John Wiley.

Hadley G (1974). *Linear Programming*. Addison-Wesley, Eighth Printing.

Haimes Y Y (1979). The Surrogate Worth Trade-off Method and its Extensions. *Multiple Criteria Decision Making Theory and Applications*, Fandel and Gal (eds.), pp 85 - 108.

Haimes Y Y and Hall W A (1974). Multiobjectives in Water Resources *Systems* Analysis: The Surrogate Worth Trade-off Method. *Water Resources Research*, Vol. 10, No. 4, pp 615 - 624.

Haimes Y Y, Hall W A and Friedman H T (1975). *Multiobjective Optimization in Water Recources Systems: The Surrogate Worth Trade-off Method.* Elsevier, The Netherlands.

Hall W A and Haimes Y Y (1975). The Surrogate Worth Trade-off Method with Multiple Decision Makers. *Multiple Criteria Decision Making: Kyoto 1975*, Zeleny (ed.), pp 207 - 233.

Harrald J, Leotta J, Wallace W A and Wendell R E (1979). A Note on the Limitations of Goal Programming as Observed in Resource Allocations for Marine Environmental Protection. *Naval Research Logistics Quarterly*, Vol. 25, pp 733 - 739.

Hwang C L and Masud A S M (1979). *Multiple Objective Decision Making: Methods and Applications.* Springer Verlag, Lecture Notes in Economics and Mathematical Systems, Vol. 164.

Hwang C L and Masud A S M (1981). Interactive Sequential Goal Programming. *J. Operational Research Society*, Vol. 32, pp 391 - 400.

Hwang C L, Paidy S R, Yoon K and Masud A S M (1980). Mathematical Programming with Multiple Objectives: A Tutorial. *Computers and Operations Research*, Vol. 7, pp 5 - 31.

Hwang C L and Yoon K (1981). *Multiple Attribute Decision Making: Methods and Applications.* Springer Verlag, Lecture Notes in Economics and Mathematical Systems, Vol. 186.

Ignitzio J P (1976). *Goal Programming and Extensions.* Heath, Boston, Mass.

Isermann H (1976). Existence and Duality in Multiple Objective Linear Programming. *Multiple Criteria Decision Making,* Thiriez and Zionts (eds.), pp 64 - 75.

Isermann H (1977). The Relevence of Duality in Multiple Objective Linear Programming. *Multiple Criteria Decision Making,* Starr and Zeleny (eds.), pp 241 - 262.

Johnsen E (1968). *Studies in Multiobjective Decision Models.* Studentlitteratur, Lund.

Keen P G W (1977). The Evolving Concept of Optimality. *Multiple Criteria Decision Making,* Starr and Zeleny (eds.), pp 31 - 57.

Keeny R L and Raiffa H (1976). *Decisions with Multiple Objectives: Preferences and Value Trade-offs.* John Wiley and Sons, New York.

Kelley J E (1960). The Cutting Plane Method for Solving Convex Programs. *J. SIAM,* Vol. 8, pp 703 - 712.

Lee S M (1972). *Goal Programming for Decision Analysis.* Auerbach, Phil.

Mac Crimmon K R (1973). An Overview of Multiple Objective Decision Making. *Multiple Criteria Decision Making,* Cochrane and Zeleny (eds.), pp 18 - 44.

Monarchi D E, Kisiel C C and Duckstein L (1973). Interactive Multiobjective Programming in Water Resources: A Case Study. *Water Resources Research,* Vol. 9, No. 4, pp 837 - 850.

Musselman K and Talavage J (1980). A Tradeoff Cut Approach to Multiple Objective Optimization. *J. Operations Research,* Vol. 28, No. 6, pp 1424 - 1435.

Oppenheimer K R (1978). A Proxy Approach to Multiattribute Decision Making. *Management Science,* Vol. 24, No. 6, pp 675 - 689.

Pearman A (1977). A Weighted Maximin and Maximax Approach to Multiple Criteria Decision Making. *Operational Research Quarterly,* Vol. 28, No. 3, pp 584 - 587.

Peddie D (1984). *Optimum Stocking Densities of Large Ungulates to Achieve Multiple Goals in the Pilansberg Game Reserve.* Ph. D. Thesis, Department of Botany, University of the Witwatersrand, Johannesburg. To be submitted.

Philip J (1972). Algorithms for the Vector Maximization Problem. *Mathematical Programming,* Vol. 2, pp 207 - 229.

Roy B (1971). Problems and Methods with Multiple Objective Functions. *Mathematical Programming,* Vol. 1, pp 239 - 266.

Rosinger E E (1981). Interactive Algorithm for Multiobjective Optimization. *J. of Optimization Theory and Applications,* Vol. 35, pp 339 - 365.

Saaty T L (1980). *The Analytic Hierarchy Process.* MacGraw-Hill, New York.

Sakawa M (1982). Interactive Multiobjective Decision Making by the Sequential Proxy Optimization Technique: SPOT. *European J. of Operational Research,* Vol. 9, pp 386 - 396.

Simon H A (1957). A Behavioral Model of Rational Choice. *Models of Man,* John Wiley, New York.

Spronk J (1981). *Interactive Multiple Goal Programming.* Martinus Nijhoff.

Stadler W (1979). A Survey of Multicriteria Optimization or the Vector Maximization Problem: Part I 1776 - 1960. *J. of Optimisation Theory and Applications,* Vol. 20, No. 1, pp 1 - 52.

Starr M K and Zeleny M (eds.) (1977). *Multiple Criteria Decision Making.* North- Holland, Studies in the Management Sciences, Vol. 6.

Steuer R E (1976). A Five Phase Procedure for Implementing a Vector-Maximum Algorithm for Multiple Objective Linear Programming Problems. *Multiple Criteria Decision Making,* Thiriez and Zionts (eds.), pp 159 - 169.

Steuer R E (1977). An Interactive Multiple Objective Linear Programming Procedure. *Multiple Criteria Decision Making,* Starr and Zeleny (eds.), pp 225 - 239.

Steuer R E and Schuler A T (1978). An Interactive Multiple Objective Linear Programming Approach to a Problem in Forest Management. *Operations Research,* Vol. 26, No. 2, pp 254 - 269.

Stewart T J (1981). An Interactive Approach to Multiple Criteria Decision Making Based on Statistical Inference. *IEEE Trans. on Systems, Man and Cybernetics,* SMC-11, pp 733 - 740.

Stewart T J (1983). Interaction Between Analyst and Decision-Maker in Support of Multiple Criteria Decision-Making Problems: An Overview of Methodology. *Interactive Decision Analysis and Modelling,* Stewart (ed.), TWISK 294, NRIMS, C.S.I.R., Pretoria.

Stewart T J (1984). Inferring Preferences in Multiple Criteria Decision Analysis Using a Logistic Regression Model. *Management Science,* Vol. 30, No. 9.

Thiriez H and Zionts S (eds.) (1976). *Multiple Criteria Decision Making: Jouy-en-Josas, France 1975.* Springer Verlag, Lecture Notes in Economics and Mathematical Systems, Vol. 130.

van Delft A and Nijkamp P (1977). *Multi-Criteria Analysis and Regional Decision Making.* Martinus Nijhoff, Leiden/ The Hague.

Wallenius J (1975). Comparative Evaluation of Some Interactive Approaches to Multicriteria Optimization. *Management Science,* Vol. 21, No. 12, pp 1387 - 1396.

Wallenius J and Zionts S (1977). A Research Project on Multicriteria Decision Making. *Conflicting Objectives in Decisions,* Bell, Keeney and Raiffa (eds.), pp 76 - 96.

White D J (1980). Multiobjective Interactive Programming. *J. Operational Research Society,* Vol. 31, pp 517 - 523.

Wilkinson J H (1965). *The Algebraic Eigenvalue Problem.* Oxford University Press, London.

Wolfe P (1970). Convergence Theory in Nonlinear Programming. *Integer and Nonlinear Programming*, J Abadie (ed.), American Elsevier, New York.

Yu P L (1973). A Class of Solutions for Group Decision Problems. *Management Science*, Vol. 19, No. 8, pp 936 - 946.

Yu P L and Zeleny M (1975). The Set of All Nondominated Solutions in Linear Cases and a Multi-Criteria Simplex Method. *J. of Mathematical Analysis and Applications*, Vol. 49, pp 430 - 468.

Zeleny M (1973). Compromise Programming. *Multiple Criteria Decision Making*, Cochrane and Zeleny (eds.), pp 262 - 301.

Zeleny M (1974). *Linear Multiobjective Programming*. Springer Verlag, New York.

Zeleny M (1976). The Theory of the Displaced Ideal. *Multiple Criteria Decision Making*, Zeleny (ed.), pp 153 - 206.

Zeleny M (ed.) (1976). *Multiple Criteria Decision Making: Kyoto 1975*. Springer Verlag, Lecture Notes in Economics and Mathematical Systems, Vol. 123.

Zeleny M (1982). *Multiple Criteria Decision Making*. MacGraw-Hill.

Zionts S (1979). Methods for Solving Management Problems Involving Multiple Objectives. *Multiple Criteria Decision Making Theory and Applications*, Fandel and Gal (eds.), pp 540 - 558.

Zionts S and Wallenius J (1976). An Interactive Programming Method for Solving the Multiple Criteria Problem. *Management Science*, Vol. 22, No. 6, pp 652 - 663.

Zionts S and Wallenius J (1983). An Interactive Multiple Objective Linear Programming Method for a Class of Underlying Nonlinear Utility Functions. *Management Science*, Vol. 29, No. 5, pp 519 - 529.