

**DECISION SUPPORT SYSTEMS FOR SOLVING DISCRETE
MULTICRITERIA DECISION MAKING PROBLEMS**

BY

THERON VAN ZYL VAN DYK

THESIS

Submitted in fulfillment of the
requirements for the degree of

**MASTER OF SCIENCE IN THE DEPARTMENT
OF STATISTICAL SCIENCES OF THE
UNIVERSITY OF CAPE TOWN**

SUPERVISOR: Prof. T.J. Stewart

SEPTEMBER, 1992

The copyright of this thesis vests in the author. No quotation from it or information derived from it is to be published without full acknowledgement of the source. The thesis is to be used for private study or non-commercial research purposes only.

Published by the University of Cape Town (UCT) in terms of the non-exclusive license granted to UCT by the author.

CONTENTS

	Page
1. Introduction	
1.1 Types of decision problems and environments	1
1.2 Definitions of MCDM concepts	3
1.2.1 Decision maker	3
1.2.2 Alternatives	4
1.2.3 Criteria	4
1.2.4 Attributes	4
1.2.5 Objectives	5
1.2.6 Goals	6
2. Interactive Multicriteria Decision Making	
2.1 Historical review on MCDM	8
2.2 Why interactive decision making?	11
2.2.1 Requirements of interactive Multicriteria Decision Making	15
3. Design Philosophies of Decision Support Systems	
3.1 Historical review on DSS	19
3.1.1 Definitions	21
3.2 A framework for the design of a DSS	25
3.2.1 Representations	26
3.2.2 Memory aids	27
3.2.3 Control aids	28
3.2.4 Variety of styles	29
3.2.5 User–friendliness	31

3.3	Development approaches for Decision Support Systems	32
3.4	The components of a DSS with specific reference to the MCDM problem	35
3.4.1	Data Component	35
3.4.2	Model Component	38
3.4.3	Dialog Component	39
4. The Multicriteria Decision Support System		
4.1	The multicriteria decision making process via the use of a DSS	43
4.2	A discussion of the possibility of integrating DSS and MCDM and possible disparities between them	47
5. A Discussion of Different Methodologies for solving the Discrete MCDM Problem		
5.1	Introduction	53
5.2	Sequential elimination methods	56
5.2.1	The STEM method	56
5.2.2	The PRIAM method	60
5.2.3	The method of STEUER and CHOO	62
5.3	Spatial proximity methods	65
5.3.1	The TOPSIS method	65
5.3.2	The REFERENCE POINT method of Wierzbicki	68
5.3.3	Interactive multiple goal programming (IMGP)	70
5.3.4	A combined approach	73

5.4	Aggregation procedures	76
5.4.1	An adaptation of the ZIONTS–WALLENIOUS method for the discrete case	76
5.4.2	Multiattribute Value Functions (The Simple Multiattribute Rating Technique – SMART)	79
5.4.3	Visual Interactive Sensitivity Analysis (VISA)	83
5.5	Outranking methods	85
5.5.1	The ELECTRE method	85
5.5.2	The PROMETHEE method	89
5.6	Conclusion	94

6. Implementation

6.1	Introduction	96
6.2	Program start–up	97
6.3	Conventions used in the system	97
6.4	Options on the main menu	98
6.4.1	Set–up	98
6.4.2	Alternatives	99
	(i) Create file	99
	(ii) Modify file	99
	(iii) View file	100
	(iv) Delete file	100
6.4.3	Criteria	101
6.4.4	Combined file	101
	(i) Create file	101
	(ii) Modify file	102

(iii)	View file	102
(iv)	Delete file	103
6.4.5	Solve	103
(i)	New user	103
(ii)	Experienced user	104
(iii)	Input needed	104
6.4.6	Output	107
(i)	View file	108
(ii)	Print file	108
(iii)	Delete file	108
6.4.7	Exit/Save	108
6.5	Hardware requirements	108

7. Testing and Evaluating the Decision Support System

7.1	Introduction	110
7.2	Evaluation criteria	110
7.2.1	Ease of use	110
7.2.2	The type of input information required	111
7.2.3	The output provided by the system	111
7.2.4	Graphical representations	111
7.2.5	The methods implemented	111
7.2.6	The elapsed time to reach a final satisfactory solution	112
7.2.7	The confidence in the solution reached	112
7.3	Methodology	112
7.4	Analysis of the evaluation process	113
7.5	Conclusion	118

References

Figures

Appendix I – AN EXAMPLE

Appendix II – THE HELP SYSTEM

Appendix III – PROBLEMS USED FOR EVALUATING THE SYSTEM

Appendix IV – EVALUATION QUESTIONNAIRE

Appendix V – STATISTICAL RESULTS

ACKNOWLEDGEMENTS

This is not an attempt in thanking everybody who supported me during the writing of this thesis, but rather an attempt in mentioning a few names, without who the completion of this thesis would have been impossible, namely:

- (i) my supervisor, Professor Theo Stewart whose constructive comments and constant assistance guided me through the years I have been busy with this study;
- (ii) the CSIR, and specifically Mr Hans Ittmann for providing me with the facilities and opportunity to work on this thesis while I was employed in the Division for Information Technology;
- (iii) Prof Paul Fatti and the students of the University of the Witwatersrand for testing and evaluating the decision support system;
- (iv) my parents and friends for providing their support and encouragement at all times.

PREFACE

The aim of this study was the design and implementation of an interactive decision support system, assisting a single decision maker in reaching a satisfactory decision when faced by a multicriteria decision making problem.

There are clearly two components involved in designing such a system, namely the concept of decision support systems (DSS) and the area of multicriteria decision making (MCDM). The multicriteria decision making environment as well as the definitions of the multicriteria decision making concepts used, are discussed in chapter 1. Chapter 2 gives a brief historical review on MCDM, highlighting the origins of some of the more well-known methods for solving MCDM problems. A detailed discussion of interactive decision making is also given.

Chapter 3 is concerned with the DSS concept, including a historical review thereof, a framework for the design of a DSS, various development approaches as well as the components constituting a decision support system. In chapter 4, the possibility of integrating the two concepts, MCDM and DSS, are discussed. A detailed discussion of various methodologies for solving MCDM problems is given in chapter 5. Specific attention is given to identifying the methodologies to be implemented in the DSS.

Chapter 6 can be seen as a theoretical description of the system developed, while Chapter 7 is concerned with the evaluation procedures used for testing the system.

A final summary and concluding remarks are given in Chapter 8.

The system is discussed by means of an example in Appendix I, while the help facility implemented is briefly discussed in Appendix II. The problems used in evaluating the system and the questionnaire completed by the respondents are given in Appendix III and IV respectively, while some statistical results are given in Appendix V.

Finally a list of references are given.

CHAPTER 1 – Introduction

Most people face the dilemma of making decisions, varying in complexity and importance, in the course of their daily activities. The environment in which many of these decisions have to be made, is often unstructured and the consideration of conflicting multiple criteria is the rule rather than the exception. As human beings we do not always have the ability to take cognisance of all of these multiple influences when attempting to make a rational and meaningful decision.

Over the past twenty years, many methodologies have been developed and exist in the literature regarding the solution of these complex decision problems. This thesis forms part of an in–depth study towards the development of an interactive decision support system with the aim of assisting decision makers in reaching satisfying decisions. It comprises mainly of a literature survey of the various components involved in such a system, as well as a discussion of the methodologies that will be implemented for assisting decision makers.

The field of multiple criteria decision making has expanded rapidly over the last decade and continues to do so. New approaches and developments of existing approaches appear regularly in the literature. This study will, however, only focus on that part of this area defined in the next section.

1.1 Types of decision problems and environments

Multiple criteria decision making problems and the methodologies for solving them, can broadly be categorised as being either continuous or discrete. In the first case the solution space is defined implicitly by sets of constraints, thus leading to an infinite

number of feasible solutions as in the case of linear programming. In the case of the discrete problem we are faced with a choice among a number of discrete alternatives. Each of the above cases can further be categorised as either stochastic or deterministic depending on whether or not provision is made for risky and/or uncertain outcomes.

In this thesis we will deal only with those problems that can be classified as being of a deterministic and discrete character, and that have to be solved by a single decision maker. According to Korhonen *et al* [1981], the aim of these decision methodologies is '(the) finding of a complete or partial rank ordering of the alternatives (or finding) the best alternative or the best subset of alternatives'. In order to structure the decision problem in question, we can represent it mathematically as follows:

$$\begin{aligned} & \text{Maximize } F(d) \\ & \text{subject to } d \in D \end{aligned}$$

where d is a decision (alternative)
 D is the discrete set of possible decisions (alternatives)
 $F(.)$ constitutes some measure of satisfaction to the decision maker with the decision reached.

In order to simplify this problem, each alternative is further characterised by a set of attributes. These attributes are measured on a specific scale (cardinal, ordinal or nominal) and the resulting values are used as performance measures. Once measures of performance have been attached to the different attributes, they are defined to be criteria. Hence, each alternative d can entirely be represented in terms of its associated vector, (c_1, c_2, \dots, c_k) , of performance measures.

Many approaches have been developed for solving this class of decision making problem.

Examples include multiattribute value theory, outranking methods, reference point approaches and weighting methods. Some of these methodologies are reviewed in Chapter 5.

The definitions and meanings of the concepts and terminology used in the context of the discrete alternative environment, are not unanimously agreed upon among the different researchers in this field. A discussion of our conceptualisation thereof is thus essential.

1.2 Definitions of MCDM concepts

Certain concepts and terminologies are used throughout the area of multiple criteria decision making. Different meanings can, however, be attached to these concepts depending on the specific area of interest. Although some of them may have identical dictionary definitions, it is useful and necessary, for a clear understanding thereof, to distinguish among them in a specific decision making context.

1.2.1 Decision maker

At the core of any decision problem, is the decision maker or group of decision makers. For the present study, we will only be concerned with a single decision maker or alternatively a group of decision makers that can act as a single entity in providing input during the decision process.

The final outcome of any decision problem considered will depend on the judgement of this single entity which has to bear the responsibility of the decisions made. It is important to realise that the system (or computer) acts only as a decision making tool, assisting the decision maker. The outcome of the decision process is therefore only as good as the input provided by the human decision maker operating the 'tool' provided.

1.2.2 Alternatives

The different options available to the decision maker and from which a decision has to be made, are termed alternatives. For the decision environment we are concerned with, the set of alternatives is discrete. This set can be seen as fixed for a particular analysis to be performed. The decision maker may, however, decide to add or delete specific alternatives during the decision process in order to determine the effect on the outcome. Examples of such a set of alternatives include the different makes of cars in the problem of buying a car or the different candidates applying for a job from which one has to be chosen.

Although criteria, attributes and objectives are very closely linked to each other a technical difference can be made among their definitions:

1.2.3 Criteria

A criterion can be viewed as a basis of evaluation or comparison of alternatives, according to one point of view. A criterion does not need to be quantitatively measurable and can be an abstract concept or express a specific point of view, for example the reliability of a car or the social welfare of a nation.

1.2.4 Attributes

Attributes can be seen as the physical and/or psychological characteristics of the alternatives defined above. These characteristics are measured for the purpose of evaluating an alternative according to one or other criterion. An attempt is made to describe the alternatives of choice in terms of these attributes, which can thus be viewed as information sources available to the decision maker for attaching more practical and workable definitions to the alternatives. As with the set of alternatives, the choice of

attributes depends entirely on the decision maker and needs to reflect the issues of importance to him. Attributes have meaning only in so far as they serve to represent a particular criterion.

According to Keeney and Raiffa [1976] the set of attributes should possess the following properties in order to describe the alternatives in the most comprehensive way:

- **complete**: all aspects of the alternatives needed to make a sound decision, should be represented by the attributes;
- **operational**: attributes must be easy to work with and self-explanatory to the decision maker;
- **decomposable**: a simplification of the evaluation process may be possible by decomposing the attributes into parts;
- **nonredundant**: no aspect of the alternatives should be accounted for more than once by different attributes;
- **minimal**: no other complete set of attributes, describing the same set of alternatives, with a smaller number of elements, should exist.

Attributes can be measured on different types of scales. A cardinal scale is one involving direct numerical values, for example the price of a car. On an ordinal scale, attributes are simply ranked relative to each other, either individually or in equivalence classes. A nominal scale of measurement is used when the attribute represents a purely qualitative property, for example the colour of a car. All three these types of measurement scales can be part of the same decision problem and are handled differently during the solution process. This will be discussed in more detail at a later stage.

1.2.5 Objectives

An objective represents an expression of a criterion in terms of striving towards an extreme value of some attribute and can thus be seen as a criterion in which one strives

to an absolute minimum or maximum. Because they can be considered as directions of improvement or preference within the set of attributes, it is clear that objectives are closely linked to the needs and desires of the decision maker and according to Zionts [1987] 'indicates the direction desired.'

Objectives can often exhibit a hierarchical structure which helps in defining a direct relationship between objectives and attributes. One or more attribute(s) can form an objective and two or more objectives can form a higher-level objective. The highest level of objectives are usually vaguely stated and therefore difficult to measure and to use. The lower level objectives can actually be seen as 'attributes' representing the higher-level ones. For example the objective 'to improve the well-being of the employees of a company', can be viewed as being composed of lower-level objectives , for example 'improve the attitude of the employees' and 'provide a pleasant working-environment.' The first one of these can again be broken down to 'salaries paid' and 'leave granted' which can be seen as the actual operational attributes.

1.2.6 Goals

Goals can be seen as the concepts of decision making that tie up most closely with the needs and desires of the decision maker. In contrast to objectives, which state the direction of increasing preference, goals state a desired result of a decision process expressed in terms of attributes, which is or is not achieved in any particular case. A goal is therefore an expression of a criterion in terms of achieving a desirable value on some attribute. The values of these goals are functions of either attributes or objectives.

A distinction is made between minimum acceptable goals and those goals that the decision maker aims to achieve. The former is called restriction levels, while the latter is known as aspiration levels. An example of a restriction level in the problem of buying a car, can be an engine capacity of at least 1200cc, while an aspiration level for the same

attribute (engine capacity) can be 1800cc. These goals are only attainable within the constraints defined implicitly by the set of alternatives. Although they are normally *a priori* determined at the start of the decision process, they usually are revisable as the process continues.

The concepts discussed above can be illustrated by means of an example:

Consider the problem of buying a new car:

'economy' is a criterion, possibly subdivided into initial costs and long run costs; relevant attributes for these two criteria are purchase price and fuel consumption; an objective is to minimize fuel consumption and a goal is to get a car for less than R60 000.

In our discussion of a specific decision problem and possible solution methods ample use will be made of the abovementioned terms and it is therefore important to have a clear understanding of their respective meanings.

CHAPTER 2 – Interactive Multicriteria Decision Making

2.1 Historical review on MCDM

Decision making as a human activity has been with us since the beginning of mankind. Upto about five decades ago little formal attempt was made to study the process of decision making. It was only during World War II with the emerging of the new discipline of Operations Research/Management Science (OR/MS), that man became aware of the importance of quantifying his decision process. Since our concept of the world tends to be very simple in situations of war and emergencies, the aim of the decision maker in such circumstances is to maximize or minimize one objective function only (to win and/or survive). The 'cost' involved or the influence that the decision may have on other variables are not considered. This may lead to short term satisfaction for the decision maker, with possible extreme repercussions for future planning processes.

Morgenstern and von Neumann [1947] were two of the first researchers to realise the 'dilemma of conflicting objectives'. It is, however, only during the last two decades that there has been an increased awareness of the need to identify and consider simultaneously several objectives in the analysis and solution procedure of decision problems. Multicriteria decision making (MCDM), as it is known today, has evolved in response to these more practical needs.

The origins of some of the more well-known methods for solving MCDM problems can shortly be summarised as follows:

The concept of vector-dominated solutions used today in some multicriteria decision making methodologies was for example first used by Koopmans [1951]. In the same year Kuhn and Tucker [1951] derived optimality conditions for the existence of efficient

solutions in a vector maximization problem. The most tangible outcome of the fifties was, however, the goal programming approach of Charnes and Cooper [1961]. The foundations of what is known today as Multi-attribute Utility theory (MAUT) were being laid down in works of Adams, May and Fagot [1959]. No significant follow-up work was done. At this time in history, the OR/MS world was mainly occupied with refining computer-based mathematical algorithms and too overwhelmed by the mathematical success and elegance of single-criterion problems to pay further attention to the developments in the field of MCDM. The foundations for problems with multiple conflicting objectives were first laid by Johnson [1968].

These methodologies are best suited for solving the continuous class of decision making problems. Very little or no interaction is required from the decision maker. As stated by Hwang and Yoon [1981] 'Their orientation and motivation is mostly to explain, rationalize, understand, or predict decision behavior – not to guide decision making.'

Most of these methodologies may not directly be used in the discrete alternative case, but they have been the basis in the development of solution methods for the discrete class of problems. Although Churchman, Ackoff and Arnoff [1957] first treated a discrete multiple criteria decision problem in a formal way, many potentially useful concepts and methods had been laid aside until 1968 when McCrimmon reviewed the existing methods and applications of this area. It is rather surprising that little effort has been given to any further reviews since then.

Due to the vast number of approaches used in the field of discrete multiple criteria problems, an attempt will not be made to give a full account thereof. The origin of some of the main thrusts that will be focused on can be summarised as follows while a more detailed discussion of the most applicable methods to be implemented into a Decision Support System (DSS) follows in Chapter 5:

Benayoun, Roy and Sussman [1966] were the first to suggest a method making use of an outranking type of relationship between the alternatives. This method was improved by Roy [1971] and is since known as the ELECTRE method. Some other methodologies based on the same idea have since be developed and include the PROMETHEE class of methods developed by Brans *et al* [1984]. Methods that make use of weights assessed either internally or that are provided by the decision maker, include the AHP developed by Saaty [1980], the SMART method developed by Edwards [1984] and the TOPSIS method developed by Hwang and Yoon [1980]. Some other developments that are of importance for this study, include methods that make use of the reference point ideas (goals) such as PRIAM developed by Levine and Pomerol [1986].

These methodologies form part of developments in recent years where the emphasis has been on methods that require frequent interaction with the decision maker. This led to solution procedures that are easier to understand and that can be used with greater confidence as the decision maker has become an important part of the actual decision process. The decision maker is not granted entire freedom to make a decision according to his own discretion nor is any decision forced upon him. These types of solution procedures have only really gained support from developers of solution methods and from decision makers themselves since the vast advances in the computer industry. The development of the MCDM field can thus not be seen in isolation from simultaneous developments in computer technology. As the applicable technological tools improve, greater emphasis can be placed on the decision technique and scientific decision analysis has therefore become an important area of reasearch.

According to Zionts [1987] 'the challenges in the field (of MCDM) today are far greater than they have ever been. And the problems that we study become even more and more difficult.' Fortunately, the tools for assisting decision makers have likewise become more sophisticated and one can therefore state with confidence that MCDM has become

established as one of the most dynamic and widely applied fields of Operations Research and Management Science. Lively interest in MCDM problems and their solutions is likely to continue well into the future.

2.2 Why Interactive Decision Making?

There are mainly two approaches for modelling human decision making. The main reason for distinguishing between them, is the availability of relevant information needed from the decision maker. In the case where all information needed for reaching a solution can be elicited *a priori* to the start of the decision process, no interaction is needed with the decision maker. The solution process then becomes a mere mechanical procedure, presenting the final solution to the human decision maker without any further involvement by him. The classical way of reaching the outcome, is by optimising a single objective function related to the known utility function of the decision maker.

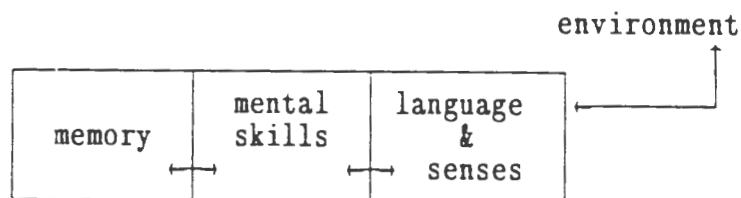
When considering unstructured and more complex problems, where multiple considerations have to be taken into account, the decision maker is often unable or unwilling to supply sufficient preference information *a priori* to enable the use of the above solution procedure. The solution of such problems by a human being on his/her own can thus become very tiresome or even impossible and new interactive solution techniques had to be developed.

Research on the appropriateness of interactive approaches for solving multicriteria decision making problems appeared in the early 1970's for the first time. The term 'interactive approach' was initiated by Geoffrion *et al* [1972]. He states 'There are, of course, other approaches that might be termed 'interactive mathematical programming' ... – but in these other approaches the interaction is usually superimposed in an *ad hoc*

manner rather than being dictated by the mathematical programming algorithm itself.'

Since these early stages and with the simultaneous developments in the computer industry, the interactive approach for solving decision problems (and specifically MCDM problems), has gain great support. This trend is also indicative of a more realistic approach towards decision making. Human choice behavior alone does not always appear to be rational which as March [1987] states 'is not necessarily a fault in human choice to be corrected, but often a form of intelligence to be refined by the technology of choice, rather than ignored by it.' The idea is not to replace the human decision maker, but rather try to improve the decision maker's ability to make decisions via the use of computers. According to Jacob *et al* [1989] humans and computers do have different information processing capabilities which need to be understood and exploited to their full potential.

The human information processor can be viewed as consisting of the following components:



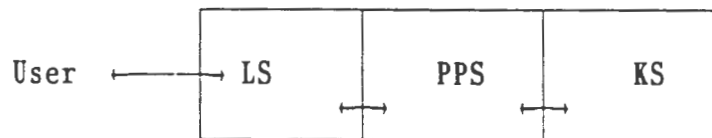
The limitations of the human being as an information processor and decision maker on his own arise because of the limited capabilities of these three components.

- (i) Humans are not capable of processing large volumes of information. This leads to selective selection of environmental influences on which to base

decisions. By doing this, important information may be ignored and decisions based on less important data.

- (ii) Human memory also has a limited capacity. Due to the fact that recently 'stored' information is the easiest to retrieve, decisions may be based on this data alone.
- (iii) The computational skills of humans are also limited. Simplifying assumptions may be made or heuristics employed when making decisions. This may lead to solving the problem faster, but the solution is to a great extent based on judgmental biases.
- (iv) Time, cost and accuracy are three other factors that suffer when humans alone are employed for making complex decisions.

The information processing capabilities of the computer as a decision support system, on the other hand, can be represented as follows:



where LS: language system (link between the human and the computer)

KS: knowledge system (information base)

PPS: problem processing system (computing skills)

Harmon and King [1985] attribute the increasing use of computers in an interactive way to four factors:

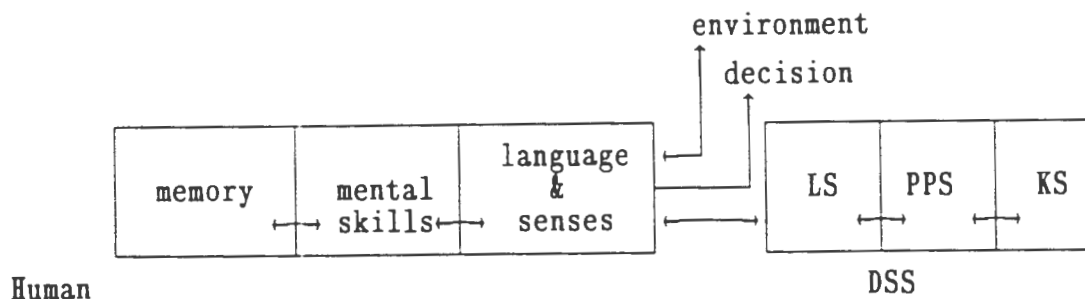
- (i) a detailed and comprehensive knowledge base;
- (ii) computers consider every known possibility;
- (iii) they never jump to conclusions and

- (iv) the possibility of constant updating of information

The memory capability and speed of processing data result in increasingly complex problems to be solved faster by a computer. The knowledge base is able to deal with facts and data that a human cannot remember. Biases are also to a great extent eliminated, especially when decisions have to be made under time pressure. The computer as a decision tool can therefore process data, translate them into relevant information and process this information more effectively than a human decision maker.

One main disadvantage of only using computers without the involvement of the human decision maker, is the lack of creativity of computers. For example, computers do not have the ability to consider novel situations, while humans can relate this to possible past experiences. The computer as part of the interactive procedure should, however, not suppress the creative thinking of the human decision maker, but rather enhance it.

The interactive procedure can thus be viewed as being of the following form:



From this schematic display one can see that an interactive procedure for solving a decision problem consists of an alternation of stages of calculation and discussion. The calculation stage allows the computer to select an action to put to the decision maker during the discussion stage. The specific action will depend on the methodology employed to solve the specific decision problem. The discussion stage, on the other hand, allows the human decision maker to consider the proposition put to him, and to provide

supplementary information regarding his preferences and desires. This additional information is then introduced into the model in the next calculation stage. A decision rejected at the start of the decision process, may even in light of information acquired by the decision maker during the decision process, be chosen as the final decision. The outcome of the decision process, however, still depends on the judgement of the decision maker.

With the development and increasing support for interactive decision making '...we can no longer see the final solution as a process of *discovering* but as one of *constructing* a process which continues throughout the whole interaction.' as stated by Roy [1987]. The outcome of the decision process is thus in a way constructed by the decision maker via the inputs provided by him.

2.2.1 Requirements of Interactive Multicriteria Decision Making

When a discrete alternative MCDM problem is solved interactively via the use of a Decision Support System (DSS), the system should ideally adhere to certain properties in order to maximize its effectiveness. These properties will be discussed in some detail in what follows.

(i) A variety of procedures exist for solving the discrete alternative MCDM problem, for example the Analytic Hierarchy Process (AHP) (Saaty [1980]), the Simple Multiple Attribute Rating Technique (SMART) (Edwards *et al* [1986]), the PROMETHEE method (Brans *et al* [1986]), a discrete version of the Zionts–Wallenius method (Korhonen, Wallenius and Zionts [1984]) and the ELECTRE method (Roy *et al* [1971]). These approaches differ mainly in the input needed from and the output desired by the decision maker during the decision process. The preference for a specific solution method is, among other aspects, a direct result of the cognitive biases existing among different

decision makers. It is, however also true that the structure (size, type of criteria) of the problem can limit the choice of the solution procedure to a great extent. The AHP can for example only handle upto seven different alternatives and/or attributes, while some methodologies do not make provision for attributes that can be evaluated on a qualitative as well as a quantitative scale.

When considering the development of a general-purpose multicriteria decision support system, one has to provide for as wide a range of decision makers and class of problems as possible. In doing this, the limitations of one approach have to be overcome by (possibly) combining it with another more powerful (in this respect) approach.

(ii) With our understanding of interactiveness, there should be no need for the decision maker to have an explicitly known value function before solving the problem at hand. This function is computed internally with the intelligent use of the input information provided by the decision maker. The inputs needed from the decision maker should be reduced to the minimum and must be requested in a simple and easy-to-understand manner. Provision has to be made for different types of input information demanded, for example direct comparisons of alternatives, providing trade-off information or giving reference points for the criteria. By requesting input information that can be understood and fairly easily provided by the decision maker, he will have a sense of control over the solution procedure. His confidence in the outcome of the decision process will thus be increased and the final solution presented to him will be easier accepted as the 'best' one under the current circumstances.

(iii) As the decision maker plays a major role in the outcome of the decision, and is not always consistent in his responses, this should be made known to him. Provision for 'backtracking' needs thus to be made to allow the decision maker to change previous inconsistent responses. The search for the preferred solution is thus constrained as little

as possible by the system itself. If the decision maker decides to terminate the procedure before a final solution has been reached, a satisficing and feasible intermediate solution should still be presented to him.

(iv) User–friendliness is an important aspect of any computerised system and a decision support system is no exception. User–friendly support should therefore be inherent to the system from the moment it is started until the output is displayed. One way of making provision for this is to have sufficient on–line help available at any stage of the decision process. That part of the system seen by the decision maker (front–end) has to be designed in a self–explanatory way and pleasant to work with. The use of menus is one way of achieving this.

A different way of viewing this aspect is what Elam and Mead [1987] call a 'fun computing environment'. By 'fun' is meant enjoyment as well as becoming deeply engrossed in an activity. The structure of the system should thus be to allow decision makers with limited knowledge of computers and even of the solution procedure, to use it with confidence and understanding.

(v) The choice situations need to be structured in a similar manner at each stage of the solution process. This will allow the decision maker to continue using a familiar procedure.

(vi) The decision maker has to be allowed to reach a preliminary satisficing solution fairly quickly. He may lose interest if the process takes too long and may thus not investigate all possibilities in order to find the 'best' solution.

A vast amount of research and development towards achieving the above requirements have taken place since Power and Aldag [1986] have stated that '... the only benefits

clearly evident at present are that the programs offer students experience with a new decision technology and provide a structured experience in decision making.' The possible limitations, as mentioned by Korhonen and Laakso [1986], have also been overcome to a great extent and the decision maker is provided with a set of simple but effective tools that enables him, by trial-and-error and if-then type of statements, to obtain the 'best' (most satisfactory) solution. (Malakooti [1988]).

It is, however, important to note that even if a specific interactive decision support system for assisting decision makers adheres to all of the above-mentioned requirements, it can be stated with great confidence that it will not be accepted by all parties involved as different decision makers still have different expectations and perceptions of the system employed. The interactive system can thus not be viewed as the panacea for all decision problems for all decision makers, but rather as an assisting tool for the majority thereof.

CHAPTER 3 – Design philosophies of Decision Support Systems

The ultimate purpose of this study is the development of a Decision Support System (DSS), incorporating multicriteria methodologies, for assisting in the solution of multi-criteria decision making problems. The DSS concept needs to be investigated in detail which will allow a better understanding of how to integrate these two thrusts in an usable way.

3.1 Historical review on DSS

In the quest to simplify the life of modern man the concept of a DSS has been developing over many years as the needs, and specifically the decision making needs, of generations changed. The foundations for the development of decision support tools were laid by the concept of Electronic Data Processing (EDP), with Management Information Systems (MIS) and developments in computer technology being important building blocks. The main purpose of EDP was to handle and manipulate vast amounts of data electronically in a relatively easy way, aiding managers in interpreting these data. Unnecessary duplication of data and the human element are thus reduced to the minimum. As the needs of managers become more intense and diverse, these systems lost their value and a new concept, MIS, arose. Initially these were just viewed as supplying decision makers with the necessary information to make decisions. MIS however continued to evolve with more emphasis being placed on the processing and presentation of the data to be used for decision making. Both EDP and MIS are basically two simple, devoted computer aids for assisting managers in their day-to-day business problems with no or very little user interactivity. A major shortcoming of both these systems are their limited use for real managerial decision making besides providing the decision maker with the data needed. In the current, highly competitive

world, this is quite a serious shortcoming.

The concept of a Decision Support System emerged in the early 70's as a direct result of this shortcoming. It was introduced by Gary and Scott–Morton in 1971 in their classic article 'A Framework for Information System Design' as a new class of information systems. A new and practical approach was initiated for applying computers and information science to the decision making problems faced by management. The developers of these systems finally realised that there were, in fact, better ways of building systems for solving management problems than the traditional and cumbersome approach of EDP and Management Information Systems. As Ackoff [1979] said: ' there is a greater need for decision making systems that can learn and adapt quickly and effectively in rapidly changing situations than there is for systems that produce optimal solutions that deteriorate with change.'

As decisions became more complicated, the need for solving them with the assistance of computers became an important issue. Perhaps by coincidence or as a result of these needs, tremendous developments in the computer industry also took place in the late 1970's and early 1980's. These include the following:

- (i) Drastic improvements in hardware and software, involving mainly ease-of-use and understanding of the computer, as well as storage capability and response time;
- (ii) an increasing capability and decreasing cost of computers in general;
- (iii) increasing availability of public databases and external data sources and
- (iv) a growth in Artificial Intelligence (AI) techniques such as Expert Systems and natural language processing.

Due to the nature and expectations of DSS as a computer–based decision tool, these developments had a major impact on their development. The synergy between increased

understanding of the concept on the part of the users and improved tools and techniques for the builders of such systems initiated a definite move towards increased DSS usage. As the technological tools available for developing efficient DSS become more sophisticated, the drawbacks in the development process were the skill and abilities of the designers and users of the system. However, with time the experience of these developers and users increased, leading to more powerful DSS being developed.

The discussed relationship between Electronic Data Processing (EDP), Management Information System (MIS) and DSS, show that DSS is only one of several important (not mutually exclusive) technological aids for improving organizational performance. The emphasis is however on increased individual and organizational effectiveness rather than on increased efficiency in processing of data. The need for such systems is growing daily and as Sprague [1986] states '...with a broad view of DSS's role in the overall mission of information systems in organizations, the future is exceedingly bright.'

3.1.1 Definitions

Although Decision Support Systems have been with us for quite some time and have been used quite extensively by managers, no established definition of this concept exists. This may be due to the broad spectrum of its abilities as well as the lack of consensus as to what characteristics such a system should have to qualify as a DSS. Many authors have, however, attempted to define a DSS. A few of these ideas are:

- (i) Bui [1984]: 'DSS are computer-based problem-solving methods that attempt to support the decision maker who deal with unstructured, ill-structured or underspecified problems.'
- (ii) Keen and Wagner [1979]: 'A DSS is a computer-based system that is used personally on an ongoing basis by managers and their immediate staff in direct support of managerial activities, which include decision making. Another term that can be used to describe these systems is 'executive

mind—support systems’.

- (iii) Wynne [1984] attempts to define a DSS according to its impact. ‘The impact of a DSS is on decisions where sufficient structure exists for analytic aids to be of value but where managerial judgement is essential.’
- (iv) The original definition by Gerrity [1971]: DSS are basically ‘an effective blend of human intelligence, information technology and software which interact closely to solve complex problems.’

Without trying to give an encompassing definition of a DSS some of the comprehensive characteristics that are apparent from the definitions will be discussed. In doing this the need for a DSS and its abilities in solving MCDM problems should be clear.

(i) Support

Almost every attempt at defining a DSS, emphasises the supporting ability thereof. A distinction can be made between two different levels of support, namely passive and normative support. In the case of passive support no attempt is made to change the existing modes of decision making. A decision support tool of this type can be seen as a computer—based system automating the existing decision process. The way decisions are presently made by decision makers is kept intact. Very little room for actually influencing the outcome exists. With normative support, consideration is, however, also given to how decisions should be made besides considering how they are actually made.

The DSS ideally needs to support the decision maker in reaching a decision throughout the entire decision process. In accordance with Simon [1960] the decision process can be viewed as consisting of three interrelated phases, namely intelligence, design and choice. As for the first phase, support is needed in the structuring of the problem as well as providing the initial data needed in the most effective way. The main supporting

function is however needed in the design phase of the decision process, where the problem is analysed and the decision maker assisted in reaching the 'best' solution under the current circumstances. The decision maker needs to be led systematically towards a decision satisfying his needs without ever losing track of the cognitive biases inherent to different decision makers.

With this active type of support, a DSS attempts to apply the very best of analytical reasoning to solving complex decision problems. The benefits to decision makers using DSS actively in reaching decisions, are far more than the passive role played by other information systems and this will thus be the type of support provided within the Multi-Criteria Decision Support System (MCDSS) to be developed.

(ii) Type of problem

Most of the existing definitions of DSS focus on the support provided for solving problems of an unstructured or at least an ill-structured nature. One way of distinguishing between structured and unstructured decision problems, is to determine whether or not the decision process can be described in detail prior to making the decision. Donovan [1976] described the characteristics of problems to be solved through the use of a DSS as follows:

- (i) the problem is continuously changing;
- (ii) the answers are needed quickly;
- (iii) data are continuously changing and come from a variety of sources;
- (iv) data must be processed into different kinds of data representations; and
- (v) when computer support is required, one is more concerned with rapid implementation than with long-term efficiency.

A problem may be classified as unstructured due to novelty, time constraints, lack of knowledge, need for non-quantifiable data, complexity and so on. The MCDM problem

with its many conflicting objectives and no pre-set rule as to how a decision should be reached, can thus be seen as definitely forming part of the class of unstructured problems.

The determination of the type of problems to be solved is closely related to the level of support needed. The need for active as opposed to passive support in solving these classes of problems is apparent when considering unstructured or ill-structured problems.

(iii) Effectiveness versus efficiency

One of the key aspects of a DSS is the improvement of a decision maker's effectiveness. Keen and Scott-Morton [1978] have defined effectiveness as the identification of what should be done followed by ensuring the relevance of the chosen criteria. This is distinguished from efficiency which relates to performing a given task as best as possible with respect to certain criteria. Another way of making this distinction is by defining effectiveness to be the 'maximization' of a specified goal, bearing in mind the limited resources available. Efficiency is then on the other hand, defined as the 'minimization' of the resources in order to achieve a desired level of satisfaction.

Although efficiency can usually be measured in terms of cost and time, effectiveness is difficult to quantify. A detailed understanding of all variables affecting the performance is needed. If the confidence of the decision maker has improved via the use of a DSS, one can generally state that the effectiveness of the decision process has been improved.

The absence of an established definition has had some damaging consequences on the DSS field and its development due mainly to a loss of credibility among developers and users. Definitions in general are, however, created and are neither true nor false and thus not of utmost importance. What is important, is that potential users of a DSS must

realise the benefits as well as the limitations thereof. It is only then that decision support systems will gain increasing support and will be applied with confidence.

3.2 A framework for the design of a DSS

When designing a Decision Support System (DSS) for solving Multiple Criteria Decision Making problems (MCDM), in fact for any DSS, a clear understanding of what we aim to achieve in this development, is needed. This can act as a framework and guideline when the actual development process starts.

According to Sprague [1986] there are two main technological levels, namely a specific DSS or a DSS generator. A specific DSS is 'the hardware/software that allows a specific decision maker or group of decision makers to deal with a specific set of related problems', while a DSS generator can be referred to as '... a platform or staging area from which specific DSS can be constantly developed and modified with the cooperation of the user, and without heavy consumption of time and effort.' As we are aiming at the development of a DSS covering a wide range of solution procedures within the class of discrete, deterministic MCDM problems, each with its own characteristics, it is clear that the development of a DSS generator needs to be our aim.

In the development of a DSS framework, there are two main influencing aspects, namely the actual decision process and the decision maker making the decision. The expectations of the decision maker need to be fulfilled and the DSS must not hinder the flow of the decision process in any way.

As mentioned before, Simon [1960] defines the process of decision making as consisting

of three inter-related phases, namely Intelligence, Design and Choice. Decision support is needed during all three these stages of decision making. In order to be able to build a supporting function into the framework of the DSS, a short list of actions taking place during each of these phases is needed.

(i) **Intelligence**

During this phase, data is gathered, objectives identified and the problem is diagnosed and structured.

(ii) **Design**

During the design phase, the data is manipulated and the most suitable solution procedure is employed to fit the structure of the problem. A preliminary solution is presented to the decision maker.

(iii) **Choice**

During this phase, various solution procedures can be employed in order to help the decision maker reaching the final, most satisfactory solution.

From the above discussion, we can therefore conclude that a DSS should support a variety of decision processes to cater for a variety of decision problems. It also needs to be very flexible as far as the data processing abilities are concerned.

When supporting the decision maker (as the second main concern) during the abovementioned three phases of decision making, the DSS should ideally adhere to certain requirements to simplify the decision process from the decision maker's point of view. These requirements need to form part of the framework of the DSS.

3.2.1 Representations

Decision makers often have trouble in describing a decision making process without relying on some conceptualization of the decision. According to Fischer [1985], 'A picture of a problem, even if merely a rough diagram 'to yourself' to help in understanding the problem, has always been a helpful tool in operations

research/management science, as well as other disciplines of the mathematical sciences.' If the decision maker is able to see graphically what happens if certain goals (for example) are relaxed or strengthened, he often understands the problem better and can thus make a better decision. It is also true that the decision maker is often forced to deal with concepts and representations unfamiliar to him – this should be avoided as far as possible or at least not without a proper explanation. Possibilities of conceptualisation aids include histograms, a statistical representation of the number of times the values of attributes have been changed in the search for a satisfactory solution, special data entry forms, etc. An example of a method that has been implemented using some of these ideas, is the visual interactive method for solving multicriteria mathematical programming problems developed by Korhonen and Laakso [1986]. The Visual Interactive Sensitivity Analysis (VISA) developed by Belton and Vickers [1990] also make use of graphical representations in order to enhance the understanding of the decision process and displaying the result of different approaches taken by the decision maker. Some of these ideas will be implemented in the DSS (see Chapter 5).

3.2.2 Memory aids

One of the main benefits of computer–assisted decision making, is the memory capability of the computer. This needs to be exploited to its fullest potential. More emphasis should therefore be given to physical memory aids as opposed to mental ones. The physical aids that should be provided by a MCDSS to be compatible with the needs of decision makers can include the following:

(i) Data bases, containing all data relevant to the specific decision problem should be provided. Data can be extracted from sources internal and/or external to the actual decision environment. In this study, we will mainly be concerned with internally located data bases, of a relational type, that have been prepared by the decision maker before actually starting to solve the problem. These data bases should be easily accessible to the user if changes need to be made and the decision maker should be able to view them

whenever he desires.

(ii) The use of workspaces for saving intermediate results is another important aid for the decision maker. This is especially of great help in solving large problems where a lot of time is spent on the actual solution process. The decision maker can interrupt the process, save the intermediate results and can start the program from this point at a later stage. These aids provide a means for accumulating results of the operations on the representations.

(iii) Another aid to the decision maker is the use of triggers. A definition of triggers is given by Jelassi *et al* [1987]: 'Triggers are certain prescribed conditions which, when true, invoke the use of rule sets.' Triggers can also be seen as reminders to the decision maker of the order of actions to take during the solution process of a MCDM problem. One example is the presentation of a menu of actions to the decision maker and by flashing the next one in line.

The decision maker cannot always remember what actions have been taken previously while the computer can store this information for a practically indefinite period of time. A 'history' of what has been done before as well as the next best action to take, can be presented to the decision maker at any stage of the decision process via the use of these memory aids.

3.2.3 Control aids

In order to improve the credibility of a solution provided by the MCDSS, the user has to have personal control over the decision process. The DSS should never be perceived as being the actual decision maker, but only as an assisting tool where the decision maker is able to change the course of the decision process whenever he desires. As was stated by Carlson [1982] 'The DSS control aids are intended to help decision makers use representations, operations and memories to synthesize a decision making process based on their individual styles, skills and knowledge.' Examples of such control aids include

menus for operation selection, on-line help facilities and edit facilities for changing a previous input made by the decision maker.

3.2.4 Variety of styles

As different decision makers have different styles, preferences, skills and knowledge abilities, any DSS should ideally be designed to support these differences. The existence of cognitive constraints is one of the main hindrances in decision making. Thorough attention should therefore be given to the relations between cognition and the DSS generator and to possible remedies for overcoming this problem. Although many authors have over the years published articles regarding this phenomena, most DSS still do not make provision for these biases inherent to all decision makers. One way of attempting to get a handle on this issue, is what is called 'attribution theory'. Hughes and Gibson [1987] have defined this to be '...a cognitive process involving perception and inference to deduce causation about observed behavior or events in the environment.'

As Maier [1973] suggests, the manager (decision maker) considers two factors before deciding whether to implement a decision or backtrack through the decision process again. These factors are (1) the anticipated quality of the decision and (2) the anticipated acceptance of the decision. If the probability of an effective decision (1) is below his level of satisfaction, the decision maker will backtrack. Attribution theory helps in evaluating the human factor (2). In short, the probability of acceptance is a function of distinctiveness, consensus, consistency over time and consistency over modality. It is clear that depending *inter alia* on the decision maker's specific preferences, the decision will be accepted or not.

Cognitive biases can occur during all the stages of decision making. These stages can in contrast to Simon [1960] be termed as information acquisition, information processing

and output and feedback:

(i) Information Acquisition

Information relevant to the decision at hand, is gathered during this stage of the decision process. This data is however often acquired in a selective way consistent with the decision maker's preconceived notions. For example, a manager may pay attention only to positive performance measures regarding a candidate for promotion about which a preconceived favorable impression exists. The availability of information also influences the decision maker towards the acquisition of data. The use of the system should not be influenced by the ease with which certain information can be recalled.

(ii) Information Processing

Due to the difficulty of processing large amounts of information decision makers may resort to the use of simplifying heuristics. The DSS therefore needs to constantly remind the decision maker of alternative ways of solving the decision problem at hand. Human beings in general choose the path of least resistance, which may not necessarily lead to the most satisfactory solution.

(iii) Output and Feedback

When a decision is made, a need exists for feedback on the decision made. Decision makers learn about their mistakes and successes through this feedback. They can however not learn from their mistakes when they automatically attribute 'good' decisions to their own abilities and 'poor' ones to uncontrollable factors. The need for recording the process by which a decision has been reached, becomes apparent in order to give the reasons for either success or failure.

Fischhoff [1982] has categorised debiasing methods into three broad classes:

- (i) correcting flaws in the experimental design surrounding the decision

- (ii) demonstrating to decision makers the existence of biases and teaching them ways to avoid them
- (iii) correcting for possible mismatches between the decision maker and the task.

More specifically, decision makers must be forced to search for information discrepancies, offered alternative formulations and solution procedures to the problem. The DSS we are about to develop, will not take formal cognisance of these possible biasing factors. We will rather try to take individual differences into consideration via a properly structured interface between the decision maker and the computer. By this we mean that different decision makers with different decision making styles and cognitive preferences will feel equally at ease in using the system. For example, people having different cognitive styles prefer different amounts of data, have preference for qualitative versus quantitative information, for graphical versus tabular representations etc. By using question-and-response dialogues extensively the user will be asked both substantive and procedural questions. The former entail questions relevant to the specific decision situation, while procedural questions ask for program control guidance, for example 'Do you want any help?'

As was stated by Kydd [1989] '...merely telling individuals about pitfalls of cognitive biases is not a strong enough treatment to help them overcome the effects of bias.' A more plausible remedy to this problem, is to build something into the system itself. This something clearly forms part of the framework of the DSS.

3.2.5 User-friendliness

One important aspect of any DSS is its user-friendliness. This means that the system need to be prompt, meaningful, pleasant to work with and error-tolerant. No preliminary training should be necessary before the decision maker can use the system — the system should rather explain itself as the decision maker proceeds. The

user–friendliness is to a great extent met by the user–system dialog component of a DSS which will be discussed in some detail at a later stage.

In developing MCDSS, cognisance will be taken of all of the abovementioned characteristics, thus providing the decision maker with a DSS that will be easy to use and will comply with most of the decision maker's needs in the solution of the specific class of MCDM problems.

3.3 Development Approaches for Decision Support Systems

Although Decision Support Systems are in certain aspects closely related to information systems such as Management Information Systems (MIS) and Data Processing Systems (DP), the differences that exist are substantial enough to require a different design approach. In the traditional life–cycle approach, the assumption is made that the requirements and structure of the system can be determined prior to the start of the design and development process. Although we have attempted to describe a framework for the development of a DSS, the exact objectives of the system and how it will be achieved are often not clear.

A DSS should therefore be developed in a very flexible manner. The user of the system should ideally be actively and continuously involved during the development process. This is to ensure that the developer stays on the right track and to permit changes to be made quickly and easily. Keen [1980] gives four reasons for this:

- (i) the user may not be able or is unwilling to provide functional specifications right from the start of the development process;
- (ii) users may not know precisely what they want and designers may not understand what the users need;
- (iii) the users' concepts of the task or the decision situation are mainly shaped by

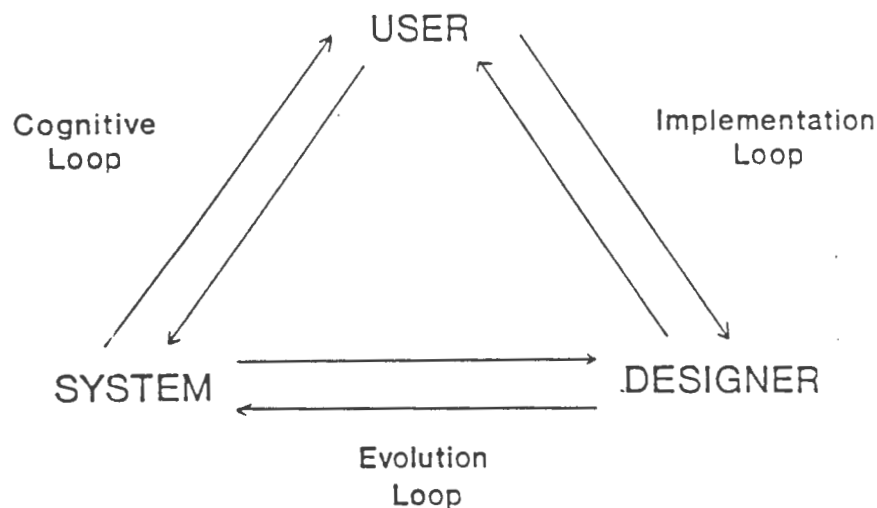
- the system; and
- (iv) users require sufficient autonomy to handle the task in a variety of ways and thus prevent the system from being standardized.

This difficulty of initially specifying the information requirements needed, and the possible changes in the decision making environment and the decision maker's task, requires an adaptive design approach. In this type of approach, the four traditional system development activities, namely requirement analysis, design, development and implementation, are combined into a single phase and are repeated iteratively. An important aspect of this type of design is the continuous involvement of the user. The decision maker actually becomes the guiding force within the development process and as Vazsonyi [1978] stated 'DSS rely on the decision maker's insights and judgements at all stages of problem solving – from problem formulation, to choosing the relevant data to work with, to picking the approach to be used in generating solutions, and on to evaluating the solutions presented.'

This form of development is called the middle-out design by Keen [1980] as opposed to the usual bottom-up or top-down approaches. An initial system is designed and developed to support decision making aspects that are of immediate relevance. The user is presented with something workable as soon as possible and as he becomes more familiar with the system and his requirements become more clearly defined, the system is expanded. This process of evaluation, modification and expansion of the system is repeated iteratively until a relatively stable and satisfactory (for the user) system is obtained. In this approach, the development process begins with a much less global view of the decision environment and the focus is shifted from delaying implementation until a final system has been developed, to starting the development as quickly as possible with the information available. Continuous feedback on the structure of the system and on the solution techniques of the relevant problem area, is therefore needed. Thus, there

is no clear break between the classic design and implementation phases. There may not even be a precise end to implementation: the evolution of new ideas and the user's ongoing learning process, lead to continuous adjustments and developments.

According to Sprague *et al* [1986], this evolutionary, interactive design approach can be visualized as follows, where each arrow indicates an adaptive influence:



The importance of the continuous interaction between user, designer and system is clearly emphasised, as well as the 'never-ending' implementation loop of the approach.

This adaptive development approach can be summarised in the words of Courbon *et al* [1979] '...methodology based on the progressive design of a DSS, going through multiple as-short-as-possible cycles in which successive versions of the system under construction are utilized by the end-user.' According to Sol [1987] there are, however, certain drawbacks with this type of development approach, namely: the prototyping idea may overemphasize the activity of solution finding, while the activity of understanding the problem situation may get too little attention; not every organization can bear the 'throw away' aspects of prototyping; a prototype is easily taken away as a pilot system or as a final product and developing prototypes may lead to isolated thinking or a 'tunnel vision', neglecting the overview of the total system.

In the case of the Muticriteria Decision Support System, no specific user–market has been identified prior to the start of the development process. The structure of the system and the choice of solution techniques to be implemented are entirely done by the developer thereof which exclude the possibility of implementing the true nature of the evolutionary approach. The system will, however, allow the **formulation** of problems using a 'middle–out' approach. The user will be able to change the definition of the specific problem to be solved, as specified by the alternatives and criteria, interactively and as often as is desired.

3.4 The Components of a DSS with specific reference to the MCDM problem

Irrespective of what the framework of the DSS looks like or what development approach is employed, all Decision Support Systems do have the same three basic components. These are normally referred to as a Dialog –, Data – and Model component respectively. The dialog component comprises a software interface through which the decision maker directs actions, to be performed by the system, and receives responses from the DSS regarding these actions. The database component has functions such as providing information in response to queries from the user, supplying data for building, updating and running of models, and storing intermediate and final results from the analyses. The model component consists of the actual solution models that can be used for solving the specific MCDM problems.

These three components can be thought of as the architecture of a DSS.

3.4.1 Data Component

For any decision problem data is needed. For some a more formally designed and

structured data base is required because of the nature and quantity of the data while others only need a loosely-structured file containing data. Data analysis needs are more important in Multicriteria Decision Support Systems than in most other computer-based information systems. A vast amount of data is used when solving a MCDM problem via a DSS. In representing this data in the most convenient way, depending on the type of data to be used, the actual use thereof can be made fairly simple.

When designing the data component of a DSS, we need to remember that it consists of four basic sub-systems:

(i) Query-Language-Facility

This can be seen as the link between the model and the data component as the data needed for the specific solution procedure will be extracted from the data component via this facility. It is also via this facility that the data component will be managed and maintained by the user of the system.

(ii) Generalized View Processor

This forms the link between the query-language-facility and the data base management system and processes the query of the user into machine-understandable language.

(iii) Data Dictionary

All the definitions of the data concepts used, are stored in this sub-system, for example the identification names of criteria and alternatives.

(iv) Data Base Management System (DBMS)

This can be seen as the primary sub-system and will be discussed in more detail. The following general objectives can be identified:

- (i) minimize redundancy in data stored;
- (ii) supply consistent information for the decision making process;
- (iii) application programmes can be developed, enhanced and maintained faster and more economically if the 'correct' DBMS is used;

- (iv) simplify the physical reorganization of data stored when this is needed;
and
- (v) centralizing the control of the data base.

The DBMS can thus be seen as the central unit for controlling and organizing data in such a way as to allow the decision maker to easily use and/or modify the stored data.

A variety of data entities exist in the solution of a MCDM problem. These include alternatives, criteria, attributes, their identification names and intermediate results. The specific data needs for these type of decision problems can be summarised as follows:

- (i) mechanisms for extracting data from a data base via some language facility;
- (ii) a command language for convenient and direct access to the data base;
- (iii) the possibility to modify the ways in which solution techniques may be used
— it must for example be possible to intercept the execution of a modelling procedure manually and provide intermediate data;
- (iv) the analysis of several criteria and alternatives at once;
- (v) provision for qualitative as well as quantitative evaluations of criteria and
- (vi) a variety of techniques are involved for solving these type of decision problems, each having specific data requirements.

Three basic possible structures of data models exist in the literature, namely a relational, hierarchical and network model. The differences among them are based on the representation of data, what operations are possible within each structure and the constraints on these representations. Since both MCDM methods and a relational data structure typically present data to the decision makers in the form of a two-dimensional table, with rows corresponding to entities and columns to properties, the relational model seems to be the most appealing possibility. The tables in a relational data structure have the following properties:

- (i) column-homogeneity: in any one column the items are of the same kind;

- (ii) each item is a simple number or a string;
- (iii) rows are distinct and can be uniquely identified;
- (iv) the ordering of rows are immaterial; and
- (v) the ordering of columns are immaterial.

For each possible usable relation between data within the context of the specific decision problem, such a relational table is acquired. An example is the relation between alternatives (models of cars) and criteria (speed, comfort and price):

Car Model	Speed	Comfort	Price
Car1	120	Good	20000
Car2	140	Poor	40000
Car3	130	Poor	35000

3.4.2 Model Component

The model component of a DSS comprises mainly of the different solution techniques that can possibly be employed in order to reach a satisfactory decision. 'A systematic environment must be provided in order for managers to solve semi-structured problems using model units.' as stated by Suh *et al* [1989] and this is achieved by the two most important units of the model component, namely the Model Base Management System (MBMS) and the Model Execution System. The MBMS must ensure that multiple executions of solution techniques are performed in a logically consistent manner and that they are properly matched with the most current data. It must also facilitate sensitivity analysis in providing the user with the necessary detail of the consequences of each possible step that can be taken. This will ensure that the optimal solution to a MCDM problem will be reached in the fewest possible steps. The MBMS, furthermore provides a logical independence and a form of interface between the MCDM technique and the data required as well as between the MCDM problem and the user.

The Model Execution sub-system, on the other hand, ensures that the necessary computations are done logically and that the results that are obtained are within reasonable bounds of credibility.

The key capabilities that a DSS needs to have as far as the model component is concerned, include the following:

- (i) the ability to create new models quickly and easily – in a MCDM sense this may entail the use of different MCDM methods for solving the same problem;
- (ii) the ability to catalog and maintain a wide range of models; the outcome of the different solution techniques have to be stored for future reference;
- (iii) the ability to combine the different models, using appropriate linkages, via the use of the data base. As the data used, is the only common factor in all of these different methods, the interrelation between the different solution techniques has to be done through the data base.

3.4.3 Dialog Component

The dialog component 'manages the two-way flow of information between the decision makers and the Model Manager and Data Manager components of the system.' as stated by Jelassi *et al* [1984]. The success or failure of any DSS depends to a great extent on this component of the DSS and it is mainly in the structure of the dialog component that the various requirements discussed in a previous chapter will have to crystalize. The way in which data is requested from, and feedback provided to, the decision maker, is of utmost importance and has to be done in a meaningful way. It is through this user-interface system that the user communicates with the system. To the user this component is the system and much of the power, flexibility and usability of a DSS are derived from capabilities inherent to the user-system-interface.

Three different modes of interaction with the decision maker can be distinguished, namely:

(i) The Action Language

This is the facilities available to the user to communicate with the system and includes the availability of a regular keyboard, the use of the function keys to perform certain actions and many more;

(ii) The Presentation Language

This consists of ways of presenting the outcome of the decision process to the user when running the system. Possibilities include screen output, printer output and graphical or tabular displays and

(iii) The Knowledge Base

Everything the user needs to know in order to run the system as well as all essentials that have to be brought to the decision making session in order to effectively use the system, form part of the knowledge base.

After identifying the different levels of communication between the system and the user, a specific dialogue style has to be decided upon. A number of possibilities exist:

(i) Question—Answer

This type of dialog style is most successfully used by inexperienced and infrequent users who are not familiar with the problem to be solved. The questions should be asked in a self-explanatory manner or sufficient on-line help has to be provided. In order to make provision for more experienced and frequent users, default answers to the questions can be given.

(ii) Command Language

In this case, the interaction with the user is far more complicated. Only certain specified commands can be used by the user to communicate with the system. A simplified possibility is to provide the user with a list of possible commands and their respective meanings. For simple applications this language is easily learned, while the more

complicated the problems to be solved, the closer this resembles a programming language and more skill is needed.

(iii) Menu

All possible actions that can be taken by the user at any stage of the decision making process, is presented to him in the form of a menu. The specific action to be taken is then selected from this menu of options. This dialog style is most effective in the case of inexperienced or infrequent users who are, however, familiar with the problem.

(iv) Input Form/Output Form

This option provides for input forms in which the user has to enter commands and data and output forms on which the responses of the DSS are recorded. After viewing an output form, an edited input form can be completed and the process can be repeated.

(v) Combinations

Various combinations of these possible dialog styles can be used, for example a question–answer approach can be used as part of an on–line feature in conjunction with a command or menu–driven interface.

When designing a DSS for the solving of different MCDM problems, we will mainly make use of the question–answer approach combined with a simplified menu–driven system. The main options available to the user for running the system will be presented in a menu format, while the actual decision making process will be completed via the use of a question–answer session. This form of dialog–style is the easiest to use and caters for all possible levels of users that may want to make use of the system. It also assists the decision maker in the understanding of the actual process as well as supporting the evaluation and interpretation of the outcome thereof. By requesting and describing certain input information, the decision maker can feel in control of the decision process and has confidence in the output produced.

In the above section a theoretical discussion has been given of the three basic

components of a DSS with specific reference to the class of discrete, deterministic MCDM problems. The practical implications of this will be discussed during the implementation phase of the Muticriteria Decision Support System. We can conclude with Jelassi *et al* [1984] 'The three basic components interact with each other...they provide the basis for building an integrated MCDSS.'

CHAPTER 4 – The Multicriteria Decision Support System

4.1 The multicriteria decision making process via the use of a DSS

The aim of developing a Multicriteria Decision Support System (MCDSS) is to support and facilitate the process of constructing a good decision, when such a decision involves multiple and conflicting goals. This requires, *inter alia*, some modelling of human decision making in this context, for which purpose one can distinguish between basically two modelling approaches, namely a process-oriented and an outcome-oriented approach. The process-oriented approach is based on the view that if one understands the decision process, one can correctly predict the outcome of that process. The outcome-oriented approach, on the other hand, reasons that if one can correctly predict the outcome of the decision process, then one has a clear understanding of the decision process itself. According to Zeleny [1982] 'The emphasis must be on the process, not on the act or the outcome of making a decision...' By doing this the emphasis will be on how decisions should be made, rather than on how they are actually made.

The use of multicriteria decision making methodologies may at times be perceived by the decision maker as a black box, into which certain inputs go, and out of which come a final recommended decision. By embedding a variety of MCDM methods in one DSS as part of its model base, the decision maker is encouraged to experiment with options, leading to a far greater degree of transparency.

The input information required by such a DSS consists mainly of the alternatives, the attributes characterizing them, and the preference information from the decision maker required by the relevant MCDM methods. The latter are provided in whatever format is most acceptable to the decision maker's style and skill. The output comprises of the

final decision reached, presented in a format preferred by the decision maker, that is, a single alternative, a partial ranking of alternatives or a total ranking of alternatives. What remains to be discussed, is what is done with the input provided in order to reach the output produced. All relevant aspects that have been discussed in previous chapters, need to be incorporated in order to form a well-structured decision process. A more in-depth discussion of the 'Tools Component', as Bui [1984] refers to it, is therefore needed in order to determine how a 'DSS functions to bring data, models, software interfaces, and the user together into an effective decision-making system.' (Floyd *et al* [1989])

As soon as the decision maker perceives the need to decide upon a specific action to take, in order to improve his current state of affairs, an initial conflict situation arises. The decision maker soon realises that it is usually not a matter of simply deciding and implementing the decision. Due to the conflict between what is desirable and what is achievable with the resources and alternatives available to him, the decision maker is 'forced' to make use of a decision making tool. The session with the DSS is started.

As was mentioned in Chapter 2, the interactive decision process consists mainly of an alternation of stages of discussion and calculation. All initial information (data) available to the decision maker and applicable to the decision problem, will first have to be entered into the system. This data is entered in a tabular format (in correspondence to the relational database discussed in Chapter 3, section 3.4.1) and consists of the alternatives, the attributes with their identification labels, the criteria values, as well as whether the different attributes need to be maximised or minimised. The interaction with the decision maker at this pre-decision stage is mainly done via the Query-Language-Facility as part of the Dialog component of the DSS.

The next step is to decide upon the specific methodology to be employed for solving the

specific decision problem. This is done by employing an question–answer type of approach, thus allowing the system to use the method most suitable to the decision maker’s skill and decision making style. As soon as an applicable method has been decided upon, a further interactive session with the decision maker is started. During this session, specific preferences are required from the decision maker and intermediate results are presented to him. A process of careful re–interpretation and re–assessment of criteria values ensues in order to seek greater divergence in the attractiveness of the various attribute scores among the different alternatives. Interactively this is achieved by forcing the decision maker to re–evaluate previous scores and/or preferences. The type of input that has to be re–evaluated will depend on the specific method that has been decided upon. Via this greater divergence, the alternatives can be distinguished more clearly from each other and a sounder basis for a final decision exists. The effort towards conflict resolution is therefore replaced by an attempt at conflict reduction.

The Query–Language–Facility is again employed during this stage as it ‘receives queries about the data needed as inputs for the model, and requests the insertion of intermediate results in the database’, Jelassi *et al* [1984]. It passes back to the Model Manager the answers to its queries. The Model Execution component executes the specific solution methodology, using whatever input has been provided, to reach a satisfactory decision. This Model–Dialog link supports the interaction between the decision maker and the MCDM Model Manager. The user, however, determines the sequence of the exchanged data and messages. This sequence of actions is not a fixed procedure for each use of the model.

It is also during this stage of the decision process that the need for some form of physical conceptualisation of the actions taken, becomes apparent. The user has to be assisted in deciding what action next to take within the sequence of actions. One possible way of achieving this, is by presenting him with a form of sensitivity analysis, showing the

effect of various changes in attribute values, on the decision reached. Throughout this interaction with the decision maker, sufficient on-line help is needed and the questions presented to the decision maker have to be stated in an understandable and self-explanatory manner. The user must also be able to return to any previous stage of the decision process (backtracking) and to exit the system whenever he desires.

As soon as a satisfactory solution to the decision problem has been reached, all that remains is to present the outcome to the decision maker in a format acceptable to him. The implementation of this final decision can then be seen as the final step in the decision process and according to Festinger [1984] 'Avoiding post-decision dissonance can also be accomplished to some extent by psychologically revoking the decision as soon as it is made.'

In this chapter we have attempted, in the words of Minch *et al* [1986], 'to describe a system whose physical states are manipulated in well-structured ways by the computer, but whose knowledge states are controlled jointly by the computer and human decision makers.' This decision process can alternatively be graphically represented as in Figure 1.

A more detailed discussion, applicable to the specific solution methodologies chosen, will be given in Chapter 6. A discussion of these various methodologies that will form part of the Model component, follows in the next chapter.

4.2 A Discussion of the Possibility of Integrating DSS and MCDM and Possible Disparities between them

In order to reach a conclusion as to whether it is at all possible to integrate Decision Support Systems and Multicriteria Decision Making, we need to reinforce previous ideas as to what is meant by these two broad concepts. An attempt is not made to give fully encompassing definitions of DSS and MCDM, but rather to emphasize their respective use(s) for decision makers.

In its most basic and simple form, a DSS can be described as any interactive computer-based tool designed for assisting decision makers in the process of decision making. It is, however, true that fairly limited (if any) formal assistance is needed when decision makers are faced with a well-structured problem. In this case few uncertainties exist regarding the way in which such a problem has to be solved. A pre-formulated, mathematically consistent and proven method of reaching a decision exists and all that remains to be done by the decision maker is to formulate the problem in a manner consistent with the 'needs' of the specific methodology. The only possible assistance that may be needed by the decision maker is in the formulation of the problem or in other words during the design phase of the decision process. (Simon [1960])

A true DSS can rather be seen as aiming at problems of a semi- or unstructured nature. As soon as decision makers are faced with problems of this nature a more complex and uncertain way of dealing with them becomes the rule. More extensive assistance is needed in the formulation of the problem as well as additional assistance in the choice of the methodology to be used and the way these methodologies operate. This corresponds to the 'Tools phase' of the decision process as discussed by Bui [1984]. The decision process is thus extended beyond the mere implementation of a fixed methodology.

The activities that form part of the decision process now include the improvement of the model of the problem (as opposed to only defining it), the acquisition of useful and significant data as well as the choice of the appropriate method. The style of the decision maker as well as his decision making skills also become important playing partners in reaching a satisfactory solution.

The basic structure of a DSS, consisting of three components, a Data- , Model- and Dialog component, was thus 'invented' to incorporate the complex nature of these decision problems. A more interactive way of decision making is proposed in order to adhere to the ever-changing perception of the problem by the decision maker. It is mainly via the Dialog component that communication with the decision maker takes place. By interacting with the DSS the decision maker is actually forced to structure his decision process and thus to a certain extent simplify it. In the last decade more extensive use has been made of Artificial Intelligence (AI) techniques in order to accomplish this and the 'extended DSS' concept as perceived by Keen [1986] has taken off. As it is impossible to identify all pre-defined kinds of behavior as mental models to be represented in a formalized way, DSS is distinctly different from Expert Systems. 'The knowledge base is not classic factual knowledge as in most Expert System applications.' according to Balestra and Tsoukias [1990].

In summary the following can be seen as characteristics (or activities) that need to be accomplished by a DSS:

- (i) deals mainly with semi- and unstructured decisions;
- (ii) support multiple decision processes and decision making styles;
- (iii) provide assistance during all phases of the decision process;
- (iv) easy and convenient to use;
- (v) decision maker must be able to change the system as he gains more experience which enhances the flexibility; and

- (vi) capture and reflect the way decision makers think.

As was discussed in Chapter 1, the general discrete MCDM problem can be seen as consisting of a discrete number of alternatives, each being evaluated by a number of criteria. The aim is then to find a solution (an existing alternative) that maximizes the satisfaction of the decision maker. By determining how these decision problems respond to the characteristics of DSS as mentioned above, the disparity and/or similarity between these two concepts should be apparent.

(i) Although MCDM in general hardly ever deals with totally unstructured decision situations, the discrete class of problems is definitely more ill-structured than the continuous multi-objective class of problems. The decision maker has difficulty in determining *a priori* how such a decision problem should be approached. The structure of the problem is therefore not well-defined and the decision maker must be able to add or drop alternatives and to evaluate and re-evaluate his judgements as often as he desires. An *ad hoc* data analysis capability is needed to perform these activities.

(ii) Various methodologies have been developed during the past number of years to assist decision makers having different decision making styles and skills in reaching a satisfactory decision. These methodologies do by their very nature support the decision maker in decision making as they are concerned with decision problems that are faced by most people on a day-to-day basis. Whether the quality of the decision is also improved, is difficult to assess. The decision maker is at least forced to structure his problem properly and is supported by logically consistent procedures in reaching a decision. His confidence in the outcome should thus be improved.

(iii) The majority of MCDM methodologies focus on the design and choice phase of the

decision process with very little attention given to the intelligence phase. By this is meant that the problem is set-up in a certain format and executed to reach a decision without much assistance given during this execution phase. Via the process-oriented approach of decision making, the design phase has gained more importance. The ability of a specific methodology to produce the outcome desired by the decision maker, is hardly ever considered and the emphasis is more on the affordability of the methodology (in terms of time and money).

(iv) The design of many technically sophisticated MCDM methodologies has not paid enough attention to the issue of ease-of-use. This may undermine the confidence of the user as he does not always fully understand the process by which a decision is reached. On the other hand, many methodologies of lesser technical and mathematical sophistication are relatively easy to use.

(v) Most MCDM algorithms use local information on the decision maker's preferences to construct a stepwise procedure leading to the final most satisfactory solution. They do, however, differ in the way the consecutive steps are taken as well as in the information needed for each step. The class of MCDM methodologies that can be termed interactive MCDM do allow, to some extent, the user to change his input and/or perception of the problem as more experience is gained during the course of the procedure. It is, however, true that once the decision maker has decided to employ a specific method, he cannot really change his approach arbitrarily without employing a new methodology from start.

(vi) The descriptive class of MCDM methodologies does to a large extent capture and reflect the way decision makers think. Some methodologies, however, make use of a normative approach, employing the way decision makers should make the decisions. This might cause some aversion from decision makers as they do not feel comfortable with the way the methodology is structured. Most decision makers should, however, be

able to find a method compatible with their specific level of experience and decision making style. No formal structure exists assisting them in employing the method best suited for their needs.

From the above it can be concluded that, depending on the specific methodology employed, MCDM do respond, although rather weakly in some instances, to the reference criteria of DSS.

One way of overcoming the shortcomings is to view MCDM not as a stand-alone DSS, but to integrate it into the model component of a DSS. By using a database structure best suited for the application field we aim at, in our case this will be a relational structure, an easy-to-understand and easy-to-use interface can be built between the Data and Model components of a Muticriteria Decision Support System. By making full use of a properly structured man-machine interface (Dialog component), the MCDM methodologies can become much more flexible and in line with the decision maker's style and skills. The decision maker's choice of decision possibilities can be expanded and his priority of established goals can easily be altered. This will have a major contribution in the design and choice phases of the decision process. A properly structured Dialog component, backed by a relational Data component, will also assist the decision maker in the gathering and validation of data as well as in the identification of the desired objectives. The intelligence phase is thus also supported.

By using a DSS for solving MCDM problems the two types of knowledge present, namely methodological and self-referential knowledge, can be exploited to the fullest (Balestra and Tsoukias [1990]). The decision maker interacts with the system and via a learning process the knowledge inherent to the method and the user respectively becomes more usable. The physical states of the process are controlled by the computer while the knowledge state is controlled by the computer and the human decision maker.

We can thus conclude that the stages of MCDM as mentioned by Jelassi *et al* [1984], namely method selection, designation of criteria, restriction of the set of alternatives and the execution of the model, can all best be supported by a DSS. If one of these stages can be seen as the most important and difficult to be perform, it will be the method selection process and a DSS is ideally structured for finding the best match between the decision maker, the decision problem and the technique to be used (Gershon and Duckstein [1982]).

CHAPTER 5 – A Discussion of Different Methodologies for Solving the Discrete MCDM Problem

5.1 Introduction

A large number of multicriteria decision making techniques for solving discrete MCDM problems have evolved since decision makers and analysts first realized the importance and relevance of this class of decision making problems. The following features may be used to distinguish among the different methodologies:

- (i) the kind of information and the stage of the decision process where this information is requested from the decision maker, that is *a priori*, during the process or *a posteriori*;
- (ii) the degree of interactivity between the formal process and the decision maker;
- (iii) the measurement scales of the data and of the results (that is cardinal, ordinal or nominal) and
- (iv) the way in which the information provided by the decision maker is transformed to be used as input to a specific methodology.

As was discussed in Chapter 2, section 2.2, we will mainly be concerned with methodologies requiring a fair amount of interaction between decision maker and decision process. It was also seen that such an interactive approach consists of alternating stages of discussion and calculation. It is during the calculation phase that the information provided by the decision maker is transformed into a form needed for a specific methodology.

The type of information requested from the decision maker and the way in which it is transformed, allow us to classify the methodologies into the following broad classes (it

should, however, be mentioned that these classifications are not necessarily disjoint and some methods may equally belong to more than one class):

(i) Sequential elimination methods

This class of methodologies are all characterised by a process for sequentially considering alternatives on the basis of criteria evaluations. This causes alternatives to be either retained or eliminated from further consideration.

Methods that can be classified as belonging to this class include the STEM (Benayoun *et al* [1971]) , PRIAM (Levine and Pomerol [1986]) and Steuer and Choo (Steuer and Choo [1983]) procedures.

(ii) Spatial proximity methods

This class of methodologies relate different alternatives through the use of spatial representations and distance measures. The alternatives are either related to other feasible alternatives or to imaginary (infeasible) points.

Methods that can be classified as belonging to this class include the TOPSIS procedure (Hwang and Yoon [1980]) , the Reference point approach (Wierzbicky [1980]) and the Interactive Multiple Goal Programming approach (Spronk [1981]).

(iii) Aggregation of the criteria

In these approaches the criteria are aggregated (using some form of aggregation function) to form a unique meta-criterion, which subsequently leads to the formation of a unique unattainable alternative. The other feasible alternatives are then compared to this one, using different comparison procedures.

Methods that can be classified as belonging to this class include the Korhonen-Zionts-Wallenius approach (Korhonen, Wallenius and Zionts [1984]), the SMART technique developed by Edwards *et al* [1986] and VISA (an extension of SMART) developed by Belton and Vickers [1990].

(iv) Outranking approach

The methodologies based on outranking relations make use of the binary relations of preference, under the form of pairwise comparisons of the alternatives. If we compare two alternatives a and b , ' a outranks b ' if and only if the decision maker takes the risk of asserting that, given his present state of knowledge, action a is *at least as good* as action b .

Methods belonging to this class of approaches include the ELECTRE method (Roy *et al* [1971]) and PROMETHEE (Brans *et al* [1986]; Diakoulaki and Koumoutsos [1991]).

A more detailed discussion of the various methodologies is given below, highlighting the possibility of implementing them in a Decision Support System.

The following terminology will be used throughout the discussions:

the user is confronted by a set, S , of p alternatives $S = \{z^1, z^2, \dots, z^p\}$, when each z^i is evaluated according to k different criteria, formally represented by the relation $z^i = (z_1^i, z_2^i, \dots, z_k^i)$. It is also assumed, without loss of generality, that all criteria need to be maximised.

Ideal: the infeasible alternative obtained by individually maximising each criterion;

Nadir: the infeasible alternative obtained by individually minimising each criterion.

In many of the methodologies to be discussed the additive difference independence assumption is made. This requires that the relative strength of preference between two alternatives that have identical fixed levels in some criteria does not change when these criteria are fixed at some other level. The Zionts–Wallenius method, the combined approach and the Simple Multiple Attribute Technique can be mentioned as applicable methods.

5.2 Sequential elimination methods

5.2.1 The STEM method

The method proceeds as follows:

1. By individually finding the maximum value for each criterion, the ideal (unattainable) criterion vector (alternative), $z^* \in R^k$ is obtained. A $(k \times k)$ payoff table is then constructed with the j th row being the criteria values of the alternative resulting from individually maximizing each of the criteria.

	z^1	z^2	z^j	z^k
z^1	z^*_1	z^1_2	z^1_j	z^1_k
z^2	z^2_1	z^*_2	z^2_j	z^2_k
\vdots						
z^j	z^j_1	z^j_2	z^*_j	z^j_k
\vdots						
z^k	z^k_1	z^k_2	z^k_j	z^*_k

2. Let iteration counter $h = 0$. Let m_i be the minimum value in the i th column of the payoff table. Calculate π_i values as follows:

$$\pi_i = \frac{z^*_i - m_i}{z^*_i} \left[\frac{1}{\sqrt{\sum (z^i_j)^2}} \right] \quad \text{if } z^*_i > 0$$

$$\pi_i = \frac{m_i - z^*_i}{m_i} \left[\frac{1}{\sqrt{\sum (z^i_j)^2}} \right] \quad \text{if } z^*_i \leq 0$$

The greatest weight is thus placed on the criterion with the greatest relative ranges. The

second term normalizes the criteria values according to the L_2 -norm, which allows for comparisons over criteria with different measuring scales.

3. Let $S^{(1)} = S$ and index set $J^* = \phi$

The entire feasible region, S (that is, including all alternatives), is thus used during the first iteration of the procedure. The set J^* designates the criterion values to be relaxed during the next iteration in order to allow greater achievement of the others.

4. Let $h = h + 1$ and calculate $\lambda_i^{(h)}$ minimax (Tchebycheff) weights as follows:

$$\lambda_i^{(h)} = 0 \quad i \in J^*$$

$$\lambda_i^{(h)} = \frac{\pi_i}{\sum \pi_j} \quad i \notin J^*$$

On the first iteration the $\lambda_i^{(h)}$ sum to one and on all subsequent iterations to less than one because J^* is not empty for all iterations greater than 1.

5. Calculate a compromise solution z^h by solving the weighted minimax program:

$$\min \{ \alpha \}$$

$$\text{s.t. } \alpha \geq \lambda_i^{(h)} (z_i^* - z_i) \quad 1 \leq i \leq k.$$

An alternative formulation to the above, which is more suited to the discrete case is the following:

$$\min_{z \in S} \max_{j=1 \dots k} \{ |\lambda_j^{(h)} (z_j^* - z_j)| \}$$

In this step we solve for the alternative in the reduced feasible region $S^{(h)}$ whose

criterion vector is closest to z^* according to the weighted Tchebycheff metric defined by $\lambda^{(h)} \in \mathbb{R}^k$.

6. Let $z^{(h)}$ be the alternative found in step 5 and compare $z^{(h)}$ with z^* .

z^* can usually be considered a good reference point for assessing the quality of a candidate criterion vector (alternative) assuming that it does not lie too far outside the feasible region.

7. If all the criteria values of $z^{(h)}$ are satisfactory, stop with $z^{(h)}$ as the final choice. Else continue with step 8.

8. Specify the index set J^* of criterion values to be relaxed and specify the amounts $(\Delta_j, j \in J^*)$ by which they are to be relaxed.

This step is executed only if components of $z^{(h)}$ exist for which the decision maker is willing to sacrifice achievement in order to improve other components.

9. Form the reduced feasible region:

$$S^{(h+1)} = \{z \in S \mid \begin{array}{ll} z_j \geq z_j^{(h)} - \Delta_j & j \in J^* \\ z_j \geq z_j^{(h)} & j \notin J^* \end{array} \}$$

Then go to step 4.

By adding additional constraining factors as above, we iterate through progressively smaller subsets of S .

(i) Comments:

1. The STEM method can be seen as a combination of the constraint method and a modified reference point method. It compares the ideal vector and a calculated feasible solution at each stage. The user is led to a solution by a sequential revision of feasible

aspiration levels. The method was originally posed in a linear programming context. By using it in a discrete sense, the decision maker may find it difficult to revise his aspiration levels at each iteration. A form of sensitivity analysis, assisting the decision maker in determining which criteria values to relax and the amount of relaxation is, however, possible. This can be done in the following way:

Given two alternatives z^j and z^k , the upper limit of an increase in z_1^k when z_1^j decreases by one, can be determined for each pair of criteria and presented to the decision maker. This will allow him to make a more effective choice. This may, however, not solve the problem of Δ 's set at levels insufficient to cause a move from the current position. A possible solution to this problem is suggested below (under conclusions).

2. The main drawback of this method is its 'irrevocability': when a concession has been made on a criterion, it is definitely registered in the model. If the decision maker wishes to change his mind, he is obliged to start the procedure again. The evolving target procedure developed by Roy [1976] provides a possible remedy to this problem.
3. The method might be quite cumbersome as the number of criteria increases, although the calculation steps are simple to perform.
4. Due to the irrevocability and the fact that at least one λ_i is set to zero at each iteration, the procedure stops after at most k iterations.
5. According to Brockhoff [1985], decision makers have higher difficulty in handling, a better understanding, lower instrumentality and more satisfaction with the STEM method than the method of Steuer and Choo.

(ii) Conclusion

The STEM method is easily understandable to any decision maker and almost no mathematical or computer sophistication is involved in using the method. The method uses a cognitive appealing principle in determining the best solution and lends itself to assisting the decision maker in determining this solution via the use of a DSS. STEM is certainly one of the best known interactive procedures. It has also been the first such

method proposed in the literature and has opened a fruitful field of research. More recent procedures are probably better adapted to the needs of decision makers. Most of the indicated drawbacks could be overcome by relaxing the irrevocability assumption. The possibility of incorporating the ideas of STEM into some other procedure will be considered.

One possible way of assisting the decision maker in determining sufficient revised aspiration levels at each iteration, is by displaying a table of adjacent alternatives for each criterion together with possible revised evaluations. The decision maker has the opportunity of indicating which changes (Δ_i 's) he is prepared to consider. This will decrease the strain put on the decision maker to specify actual changed evaluations for certain criteria and should increase the attractiveness of STEM as a decision making methodology.

5.2.2 The PRIAM method

This method is based upon the pairwise comparison principle for determining the most satisfactory decision and proceeds as follows:

1. A starting alternative is set by the decision maker or alternatively if he does not have the ability to do this, the decision maker is presented with the negative ideal (nadir) alternative as the starting point. This starting solution becomes the previous best as well as the best alternative considered so far.
2. The decision maker is requested to specify certain modifications to the levels of criteria constituting the starting solution. Four scenarios are possible as a result of these modifications:
 - (i) the resulting modified alternative has already been chosen – try again and repeat step 2.
 - (ii) the resulting modified alternative is infeasible (not part of S) – try

again and repeat step 2.

- (iii) the decision maker does not want to continue with the search process – go to step 6.
- (iv) in all other cases go to step 3

3. The decision maker has to indicate whether the modified alternative is tentative or imperative in nature, meaning that the decision maker can either alter these modifications at a later stage (if so desired) or he is not prepared to change them at any stage.

4. If the resulting alternative is preferred to the previous best one, it becomes the new previous best alternative, else it is added to the list of already met alternatives.

5. If the previous best alternative is not in the already met list, it becomes the new best alternative, the modified alternative is added to the already met list and is set to the new starting point. If the previous best alternative has already been met, it is pointed out to the decision maker.

6. At this step, the decision maker does not want to, or cannot continue along the current path because he has arrived to an alternative. Then the algorithm goes back to the first tentative alternative which precedes the starting point. If such an alternative exists, it becomes the new starting point and the procedure returns to step 2. If it does not exist, stop with the current alternative as the best choice.

(i) Comments

1. The decision maker only has to make a small number of pairwise comparisons in order to reach a satisfactory final solution.
2. The method does not assume that the decision maker will always progress in the right

direction and allows for 'backtracking'.

3. PRIAM relies on artificial intelligence concepts, thus enhancing the cognitive ease of the procedure.
4. The method follows a search tree and the exploration only theoretically stops when, along the last path, the decision maker is, at each step, satisfied with his choice.
5. Only the path which is followed, is memorized, so that it is quite possible to meet twice, or more often, the same alternative (actual or hypothetical).
6. The tendency of the decision maker to try various strategies by emphasizing one criterion rather than another, is one of the most important acquisitions of the algorithm.
7. The method can be adapted to accept 'I don't know' responses.

(ii) Conclusion

The method is mathematically simple to understand and operate and should be appealing to decision makers because of the backtracking ability. In order to assist the decision maker in enlarging his search tree sufficiently, a statistical routine can be used to memorize how many times the decision maker has tried to change each criterion. Those that have been considered less than the average are pointed out to the decision maker. This is very appealing from a DSS point of view. PRIAM gives in short time a small set of Pareto optimal alternatives.

PRIAM is definitely a possible choice for implementing into a DSS.

5.2.3 The method of STEUER and CHOO

Although this method is originally based on the use of linear programming, it can also be employed in the discrete case.

This procedure presents samples of progressively smaller subsets of nondominated alternatives to the decision maker, from which he has to select one as his most preferred.

The method proceeds as follows:

1. Calculate the ideal point, z^* as in the case of the STEM procedure and let $z^{**} = z^* + \epsilon$ where ϵ is a sufficiently small positive number. Let $\Lambda^1 = \{\lambda \in \mathbb{R}^k : \lambda_j \in [0,1], \sum \lambda_j = 1\}$ be the initial set of weighting vectors. Let $h = 1$.

2. Randomly generate a large number ($\approx 50 \times p$) of weighting vectors from Λ^h . This can be achieved by employing the LAMBDA procedure discussed in Steuer [1986], which provides a set discretization capability for gathering representatives from set Λ^h .

Filter this set to obtain a fixed number ($2 \times p$) of representative weighting vectors. By filtering an attempt is made in obtaining the ($2 \times p$) vectors the most different from one another.

3. For each representative weighting vector λ , solve the associated augmented Tchebycheff program:

$$\begin{aligned} & \min (\mu - \sum \rho_j z_j) \\ \text{s.t. } & \mu \geq \lambda_j (z_j^{**} - z_j) \quad (j = 1, 2, \dots, k) \\ & z \in S \end{aligned}$$

where ρ_j are sufficiently small positive values.

Filter the $2 \times p$ resulting nondominated points to obtain P solutions.

4. Present the P compromise solutions and ask the decision maker to select his most preferred one. Let z^h be this point.

5. (a) If $h = t$ then STOP with z^h as the most preferred solution (where t is a prespecified number of iterations), else do (b).

(b) Let λ^h be a weighting vector which generates z^h . Its components are given by:

$$\lambda_j^h = \frac{1}{z_j^{**} - z_j^h} \left(\sum \frac{1}{z_j^{**} - z_j^h} \right)^{-1} \quad (j = 1, 2, \dots, k)$$

Determine the reduced set of weighting vectors:

$$\Lambda^{h+1} = \{\lambda \in \mathbb{R}^k : \lambda_j \in [l_j, u_j], \sum \lambda_j = 1\}$$

where

$$[l_j, u_j] = \begin{cases} [0, r^h] & \text{if } \lambda_j^h \leq r^h/2 \\ [1-r^h, 1] & \text{if } \lambda_j^h \geq 1 - r^h/2 \\ [\lambda_j^h - r^h/2, \lambda_j^h + r^h/2] & \text{otherwise} \end{cases}$$

in which r is a prespecified 'convergence factor' ($0 < r < 1$).

Let $h = h + 1$ and go to step 2.

(i) Comments

1. This method is similar to the weighted sums/filtering approach of Steuer (Steuer [1986]).
2. It possesses a degree of definitiveness as the number of solutions and iterations are pre-determined.
3. The procedure requires a fair amount of Computer Processing Unit (CPU) time as a large number of weighted Tchebycheff sums need to be calculated at each iteration.
4. The decision maker may change his mind (as an error-correcting ability), but only to a certain extent because of the monotonic reduction of the set of weighting vectors performed in step 5(b).
5. Many technical parameters (P , t , r), without any preferential meaning have to be prespecified.
6. The stopping rule in step 5(a) is somewhat artificial and a better way is to let the decision maker stop the procedure when he wishes.

(ii) Conclusion

The algorithm requires more than a fair amount of mathematical sophistication in order to implement, especially in determining the random weighting vectors in step 1 as well as filtering them. Although computer-based procedures have been developed by Steuer

(Steuer [1986]) for doing this (LAMBDA and FILTER), they are not always readily available and will require some sophisticated computer knowledge to develop. The method is quite appealing from a cognitive point of view. By allowing the decision maker to alter his aspirations during the course of the solution process, it becomes even more attractive. The determination of P , t and r may, however, be difficult for the user to do. Another disadvantage of the method is the fact that the decision maker is forced to choose one alternative from P different ones in each iteration which may not be easy (depending on the size of P and the degree of difference among the alternatives). Due to the above problems, this method may not be a good candidate for implementing into a DSS.

5.3 Spatial proximity methods

5.3.1 The TOPSIS (Technique for order preference by similarity to the ideal solution) method

This method proceeds as follows:

1. The decision matrix, consisting of the alternatives as rows and the criteria vectors (evaluations of alternatives) as columns, is normalized in order to transform the criteria dimensions into nondimensional vectors allowing comparison across the different criteria. A possible way of calculating an element r_{ij} of the normalized matrix is as follows:

$$r_{ij} = z_j^i / \sqrt{\sum (z_j^i)^2} \quad i = 1 \dots p, j = 1 \dots k$$

2. The decision maker is requested to supply a set of weights for the criteria which is then accommodated into the decision matrix by multiplying each column with its associated weight w_j . These weights could also be determined from rank orders assigned

to the criteria depicting their order of preference (see the section on weight assessment discussed for the SMART procedure). The weighted normalized decision matrix V is then calculated with each element having the form:

$$v_{ij} = w_j r_{ij} \quad i = 1 \dots p, j = 1 \dots k$$

3. The ideal, v^* and negative-ideal (nadir), v^- alternatives are then obtained as consisting of the best and worst values for the different criteria respectively. They thus depict the most preferable and the least preferable (usually unobtainable) alternatives respectively.

4. The distances between each alternative and the ideal and nadir solutions are then calculated as follows:

ideal solution:
$$S_{i^*} = \sqrt{\sum_j (v_{ij} - v_j^*)^2} \quad i = 1, 2, \dots, p$$

nadir solution:
$$S_{i^-} = \sqrt{\sum_j (v_{ij} - v_j^-)^2} \quad i = 1, 2, \dots, p$$

This distance is thus calculated using the Euclidian (L_2 -norm) distance measure. A more robust distance measure that can alternatively be used, is the L_∞ -norm (defined as $\max(|v_{ij} - v_j^*|)$ in the case of the ideal solution).

5. The relative closeness of alternative z^i to the ideal solution is calculated as follows:

$$C_{i^*} = S_{i^-} / (S_{i^*} + S_{i^-}), \quad 0 < C_{i^*} < 1 \quad i = 1, 2, \dots, p$$

Thus $C_{i^*} = 1$ if $z^i = z^*$ and $C_{i^*} = 0$ if $z^i = z^-$. An alternative is closer to z^* as C_{i^*} approaches 1.

6. Rank the alternatives according to the descending order of C_{i^*} .

(i) Comments

1. It can be proven that the TOPSIS method is a special case of the Simple Additive Weighting (SAW) method, using the city block distance as distance measure instead of the Euclidean distance measure as described above (Hwang and Yoon [1981]). The SAW method can be considered as one of the best known and easiest to understand methods, which increases the popularity of the principles used in TOPSIS.
2. The weights that are needed from the decision maker can either be directly provided by him or can alternatively be deduced from rank numbers given to the criteria. Most decision makers are able to at least rank the criteria in order of preference if not attaching specific weights to them.
3. The indifference curves formed by the utility function of the TOPSIS method indicate a variation of hyperbola where the weighted distance from two fixed points (ideal and nadir) is zero. For $C_{i*} \geq 0.5$ the indifference curves are convex to the origin, which indicates the property of diminishing marginal rates of substitution observed in most indifference curves. For $C_{i*} < 0.5$, the curves are concave to the origin, which may be interpreted as a risk-prone attitude resulting from a pessimistic situation. Hence, this approach can be viewed as an amalgamation of optimistic and pessimistic decision methods.
4. The weights are the factors in this procedure that will ultimately determine the final outcome. One way of assisting the decision maker in assessing these weights, is by some form of graphical representation depicting the outcome (order of alternatives) using different weight assignments for the different criteria. This can be done using the VISA ideas of Belton and Vickers [1990].

(ii) Conclusion

Due to the fact that decision makers generally perceive their 'best' alternative to be the one closest to the ideal alternative and furthest from the nadir point, this method should

be cognitively appealing to them. The input information needed (weights or rank numbers) can be supplied relatively easy and with the assistance of a graphical sensitivity routine, their effect on the outcome can be determined. As the need for mathematical and/or computer sophistication is a minimum, there is basically no restriction to the number of alternatives and/or criteria that TOPSIS can handle. The weights used are directly related to the outcome produced, thus increasing the understandability of the method and the belief in the outcome given. What is achieved in this method can be directly related to what is given. The method also allows for any evaluation scale for the criteria (nominal, ordinal or cardinal).

One can thus conclude that TOPSIS is definitely a viable method for implementation into a DSS. It may, however, be necessary to combine the ideas with some other methodology. This will be discussed at a later stage.

5.3.2 The REFERENCE POINT method of Wierzbicki

The method proceeds as follows:

1. Compute the ideal (usually infeasible) alternative $z^* = (z_1^*, z_2^*, \dots, z_k^*)$ as consisting of the maximum attainable values for the k different criteria.
2. The decision maker has to express his aspirations in the form of a $y^{(1)} \in R^k$ reference alternative where $y^{(1)} < z^*$. The elements of the $y^{(1)}$ vector represents the levels of criteria values the decision maker would like to attain.
3. The λ -vector (weight) corresponding to $y^{(1)}$ is calculated as follows:

$$\lambda_i^{(1)} = \frac{1}{|z_i^* - y_i^{(1)}|} \left[\sum_j^k \frac{1}{|z_j^* - y_j^{(1)}|} \right]^{-1} \quad i = 1 \dots k$$

It will be noted that the λ 's sum to one.

4. The following procedure is then solved to obtain $z^{(1)}$, which is a projection of $y^{(1)}$ onto the nondominated set:

$$\begin{aligned} & \min \{ \alpha + \rho \sum_i \lambda_i^{(1)} (y_i^{(1)} - z_i) \} \\ \text{s.t. } & \alpha \geq \lambda_i^{(1)} (y_i^{(1)} - z_i) \quad 1 \leq i \leq k \end{aligned}$$

α is unrestricted, thus it is not essential for the $y^{(1)}$ to be a feasible alternative.

This $z^{(1)}$ can be seen as a projection of $y^{(1)}$ onto the nondominated set of alternatives.

5. Assuming that $z^{(1)}$ is sufficiently stimulating to cause the decision maker to update his aspirations, a second iteration is started by specifying a new reference alternative $y^{(2)} < z^*$ and repeat the procedure from step 3. This new $y^{(2)}$ can be seen as reflecting the decision maker's updated aspiration levels.

6. If the decision maker is satisfied with the solution reached, stop.

(i) Comments

1. The reference method is a very simple way of leading the decision maker towards a final solution. No heavy calculations are involved, allowing the CPU time needed to be minimal.

2. Due to its simplicity and cognitive appealing way of reaching a final outcome, decision makers should have no resistance towards using the method and will have confidence in the solution reached.

3. An appealing feature of the method is that the decision maker is allowed to be uninhibited when specifying reference alternatives – these reference points do not need to be feasible.

4. The reference point method makes a single probe of the nondominated set per iteration.

(ii) Conclusion

The method is closely related to the goal programming approach in that it tries to find a solution as close as possible to a specified aspiration level, using the augmented weighted Tchebycheff metric. The method requires substantial active involvement from the decision maker and he must seriously contemplate his tradeoffs at each iteration, because the method terminates unless the decision maker keeps specifying new reference alternatives. The method also facilitates experimentation as the decision maker can, via new reference points experiment with new ideas at any time and thus abruptly redirects the search process. Although aspiration levels can be rather easily specified by the decision maker at the beginning of the procedure and at specific iterations when he wishes to reorient his explorations, it may be difficult for him to provide new levels at each iteration. The decision maker will therefore have to be assisted in doing this (see TOPSIS for a possible way). The method can also be enhanced by placing lower bounds on the criteria (or alternatively using the nadir alternative as the lower bound).

5.3.3 Interactive Multiple Goal Programming (IMGP)

Although Interactive Multiple Goal Programming was originally developed for assisting decision makers in solving continuous multiple objective decision problems, it can also, with some minor modifications, be applied to discrete decision problems with explicitly given alternatives. The algorithm to be discussed is based on reducing the set of feasible alternatives at each iteration until the decision maker is able and willing to choose the most satisfactory alternative among those remaining.

1. A potency matrix is constructed as consisting of the ideal, z^* and nadir, z^- (usually infeasible) alternatives.

$$P = \begin{bmatrix} * & * & \dots & * \\ z_1^- & z_2^- & \dots & z_k^- \end{bmatrix}$$

where z^* is a vector consisting of the maximum values for each criterion considered separately and z^- is a vector consisting of the minimum values for each criterion considered separately.

2. Given the pessimistic solution, that is the lower row of P , the decision maker has to indicate which goal value(s) (criterion values) should be improved in value. Let this be column j .

3. If the decision maker is unable or unwilling to supply the exact value of improvement, a new goal level, g_j equal to $(z_j^* + z_j^-)/2$ is assumed (all other pessimistic goal levels remaining equal). If an exact desired goal level can be supplied, g_j is set equal to this value.

4. The set of feasible alternatives is reduced by deleting all alternatives for which $z_j \geq g_j$ is not true assuming z_j is to be maximised.

5. The decision maker is presented with the new ensuing potency matrix calculated from the reduced set of alternatives. If he is prepared to accept this restriction and its consequences on the other criteria values, this potency matrix becomes the new matrix to be considered with the second row thereof the new pessimistic starting solution. If the decision maker is unhappy with what is presented to him, a new goal level, \hat{g}_j is calculated as being halfway between the previously accepted level and g_j . This process is repeated until the decision maker is satisfied with the potency matrix and the ensuing solution presented to him.

6. The decision maker is then asked again to indicate which goal level should be improved and the procedure is thus repeated from step 2. If the list of feasible alternatives has been reduced sufficiently to enable the decision maker to choose his most satisfactory solution from those remaining, the procedure terminates.

(i) Comments

1. The class of problems for which IMG P can be used is quite large. Because no strong assumptions about the decision maker's preference function are made, IMG P can be used to generate a unique final solution, a series of efficient solutions or a set of satisficing solutions.
2. IMG P is a relatively simple method which is easily understood by the decision maker. The cognitive appeal of the method is thus enhanced.
3. The simplicity of the method implies that it can easily be computerized. It thus becomes feasible in terms of time and cost to carry out many iterations within a short period. The decision maker can thus benefit extensively from the learning effects of using interactive procedures.
4. The types of input demanded from the decision maker appear to be rather simple: (i) is the given solution acceptable or not?; (ii) which goal value needs to be improved?; (iii) how much at the least should this goal value be improved (optional)?; (iv) do you accept the consequences of the proposed improvement of the value of the indicated goal variable? The user thus has the opportunity to revise any previously stated improvements.
5. A possible disadvantage of the method is that there is no formal guarantee that the decision maker will ever stop changing his mind, thus leading to the possibility of repeating the procedure indefinitely.
6. The method is not restricted in any way by the way in which alternatives are evaluated. Normative, ordinal as well as cardinal evaluations can be handled with the same ease.

7. The method can also be adapted to make provision for specifying more than one goal value to be improved in step 2.

(ii) Conclusion

We can conclude that the interactive multiple goal approach to solving multiple criteria decision making problems is definitely an option to consider when developing a decision support system for handling these kind of problems. The method is closely related to the STEM as well as to the PRIAM procedures and a possible combination of the ideas used in these approaches may be more appealing to the decision maker than considering them separately. The use of graphical representations will also enhance the cognitive appeal of the combined approach.

5.3.4 A combined approach

Due to the similarities between STEM, the reference point approach, IMGP and PRIAM, a possible way of combining them into a single encompassing methodology needs to be considered. This method can be constructed to proceed as follows:

1. Let $Z_A^1 = S$, the entire set of alternatives and set the iteration counter $h = 1$.
2. Determine the ideal and negative ideal (nadir) alternatives, z^{h*} and z^{h-} , both elements of R^k by finding the maximum and minimum values for each individual criterion respectively considering all alternatives belonging to the set Z_A^h . Present these alternatives to the decision maker.
3. Assuming that he is at the nadir point, z^{h-} , the decision maker is asked to specify which criteria values he would like to improve, as well as the desired aspiration levels for each criterion of those specified. If the decision maker is unable or unwilling to specify these levels, the method can alternatively make use of fixed directions as in the case of

STEM (see step 2) or IMGp (see step 3).

4. Let J^* be the set of those criteria the decision maker would like to improve and Δ_j ($1 \leq j \leq k$) the amount of these changes. These new aspiration levels, specified or calculated for each attribute, form a new goal alternative, $z^{h+1} \in R^k$ which can be defined as

$$\begin{aligned} z_j^{h+1} &= z_j^{h-} + \Delta_j && j \in J^* \\ &= z_j^{h-} && \text{otherwise.} \end{aligned}$$

The new set of potential outcomes becomes:

$$Z_A^{h+1} = \{z \in Z_A^h : z_j \geq z_j^{h+1}\}$$

Set $h = h + 1$.

If this set is empty, the decision maker is reverted back to step 3. Otherwise, the new ideal and nadir alternatives, considering all alternatives belonging to the new set Z_A^{h+1} , are calculated.

5. Calculate a compromise solution, z^c from the reduced set of alternatives, closest to z^h in an additive difference sense by using the following Wierzbicki scalarizing function:

$$\begin{aligned} \min_i \{ \alpha + \rho \sum_j \lambda_j (z_i^h - z_j) \} \quad & i = 1 \dots p \\ \text{s.t. } \alpha \geq \lambda_j (z_i^h - z_j) \quad & 1 \leq j \leq k \\ z \in Z_A^h \end{aligned}$$

where $\lambda_j = \frac{1}{|z_i^* - z_i^h|} \left(\sum \frac{1}{|z_i^* - z_i^h|} \right)^{-1}$

6. z^c , as well as the new ensuing ideal and nadir alternatives of the reduced set are presented to the decision maker.

(a) if he is satisfied and prepared to accept z^c as the final solution, STOP

(b) otherwise, go to step 3

(i) Comments

1. Within the framework of this method, no consistency is required from the decision maker. He is free to change his mind. The main purpose of such a procedure is to support learning of preferences in a trial and error fashion.
2. The preference information required in step 3 is similar to STEM, although no irrevocability is involved here, which make the information easier to supply.
3. The preference information required in step 3 is only used to guide the search within Z_A^h . We believe that such information, which increases the cognitive strain imposed on the decision maker, is unnecessary in a learning-oriented approach. Indeed, it would be possible to use a fixed direction of preferences (as in STEM or IMG P) and possibly to allow the decision maker to specify this information when he really wishes to. The number of alternatives are reduced during each iteration of the method by considering only those which fulfill the decision maker's specified goal levels.

(ii) Conclusion

This method is simple to understand and no mathematical sophistication is required in order to solve a decision problem using this combined approach. The method is aimed at implementing the 'best' concepts of the reference point, the STEM, IMG P and PRIAM procedures discussed previously and will be implemented in addition to the STEM and PRIAM procedures.

5.4 Aggregation procedures**5.4.1 An adaptation of the ZIONTS–WALLENIOUS method for the discrete case**

This procedure can be classified as belonging to the class of methods making use of some form of aggregation function for the different criteria in order to evaluate them. The

method proceeds as follows:

1. Let $\Lambda^{(0)} = \Lambda = \{\lambda \in \mathbb{R}^k \mid \lambda_j > 0, \sum_j \lambda_j = 1 \text{ (optional)}\}$ be the weighting vector space.

Using an arbitrary $\lambda \in \Lambda^{(0)}$, find the alternative with the highest weighted score as follows:

$$\sum_j \lambda_j (z_j^i) \quad \text{over all alternatives } i = 1 \dots p.$$

Let $z^{(0)}$ denote this alternative and iteration counter $h = 0$.

2. Generate all efficient alternatives, z^1 , adjacent to $z^{(h)}$. In order to clarify what is meant by 'adjacent efficient alternatives' we need to define the concepts of 'dominance' and 'convex dominance'.

Dominance: Alternative z^m is said to be dominated by alternative z^n if $z_i^n \geq z_i^m$ for all i with a strict inequality for at least one i .

Convex dominance: Alternative z^m is said to be convex dominated by the alternatives in set Z if z^m is not dominated and there exist multipliers μ_n such that

$$\sum_{\substack{n=1 \\ n \neq m}}^p \mu_n z_i^n \geq z_i^m \quad \text{for } i = 1 \dots k$$

$$\sum \mu_n = 1, \quad \mu_n \geq 0, \quad n \neq m, \quad n = 1 \dots p$$

Intuitively, a blend of alternatives dominates alternative z^m .

Adjacent efficient alternatives: Alternative z^q is an adjacent efficient neighbor of alternative z^r if no convex combination of z^q and z^r is convex dominated by any convex combinations of alternatives in either of the sets $S_1 = \{z^t, t = 1 \dots p, t \neq q\}$ or $S_2 = \{z^t, t = 1 \dots p, t \neq r\}$. Thus, if z^r is not convex dominated, then z^q is an adjacent efficient neighbor of z^r if the following linear programming problem has a feasible solution with a positive objective function value:

$$\begin{aligned} & \max \sum \lambda_i (z_i^q - z_i^r) \\ \text{s.t. } & \sum \lambda_i (z_i^t - z_i^r) \leq 0 \quad \text{for all } t = 1 \dots p, t \neq q, t \neq r \\ & \lambda_i \geq 0, \quad i = 1 \dots k \end{aligned}$$

We will use the terms 'adjacent alternatives' and 'adjacent efficient alternatives' interchangeably.

If there are no adjacent solutions, go to step 6.

3. Ask the decision maker to compare each of these adjacent alternatives to $z^{(h)}$.

(a) If the adjacent solution, z^n is preferred, generate a constraint

$$\lambda_j (z_j^n - z_j^{(h)}) \geq \epsilon \quad (\epsilon > 0 \text{ and small}) \text{ and go to step 4.}$$

(b) If $z^{(h)}$ is preferred, generate a constraint $\lambda_j (z_j^{(h)} - z_j^n) \geq \epsilon$ ($\epsilon > 0$ and small) and go to step 4.

(c) If the decision maker is unsure about his preference, do not generate any constraints.

If all adjacent efficient alternatives have been compared to $z^{(h)}$, go to step 4. Else repeat step 3. If no z^n is preferred to $z^{(h)}$, go to step 6.

4. Use the constraints generated in step 3 (from this and possibly previous iterations) to form $\Lambda^{(h+1)}$ by finding a feasible solution to the linear program described above, adding the following set of constraints:

$$\begin{aligned} \lambda_j (z_j^n - z_j^{(h)}) &\geq \epsilon \quad (\text{for } z^n \text{ preferred to } z^{(h)}) \\ \lambda_j (z_j^{(h)} - z_j^n) &\geq \epsilon \quad (\text{for } z^{(h)} \text{ preferred to } z^n) \\ \sum \lambda_j &= 1, \quad \lambda_j \geq 0 \end{aligned}$$

If the above set of constraints has no feasible solution, delete the oldest set of constraints on the weights and repeat step 4. (By mistake or from changes in the

decision maker's aspirations, there may be inconsistencies in the λ -constraints to the extent that $\Lambda^{(h+1)} := \phi$.)

5. Employing the λ -constraints to reduce $\Lambda^{(h)}$, we form $\Lambda^{(h+1)}$ and return to step 1.
6. Delete the oldest responses generated in step 3 and go to step 2. If all responses have been dropped, rank order the alternatives and stop.

(i) Comments

1. So long as the decision maker is able to choose between pairs of alternatives the set of alternatives becomes successively smaller and the procedure converges.
2. This method can be very effective in rapidly reducing the weighting vector space in the early iterations.
3. This procedure has an error-correcting capability in that it deletes its oldest λ -constraints when inconsistencies occur.
4. Although the method relies on fairly mathematically sophisticated procedures, the only information needed from the decision maker is pairwise comparisons.

(ii) Conclusion

This method requires only from the user to compare two alternatives at one time, thus reducing the cognitive strain. It has also been proved that for any number of criteria, the number of pairwise comparisons required increases with the number of alternatives at a decreasing rate (Korhonen *et al* [1984]). The method converges to the efficient extreme point of greatest utility when the decision maker's implicit utility function is pseudoconcave, thus allowing a wider applicability of the method. The ability of the method to implicitly take cognisance of inconsistencies, is also a great advantage. Although the method has been adapted for the discrete case, it still needs some linear programming (LP) software for determining the adjacent efficient alternatives. The

method relies on repeated executions of the LP with an increasing set of constraints during subsequent executions. This causes an increase in the computational overhead, which reduces the response time to a large extent. Another major disadvantage of the method, as indicated by Korhonen *et al* [1984], is the relative large number of judgments required from the decision maker for problems having five or more criteria. The decision maker may thus find the method too cumbersome. These factors can cause us to argue that the method does not entirely fit into the framework of the DSS and will thus not be implemented.

5.4.2 Multiattribute Value Functions (The Simple Multiattribute Rating Technique – SMART)

The SMART technique (Edwards *et al* [1986]) belongs to the class of Multiattribute Utility Techniques (MAUT), relying on an additive value function in order to obtain a single importance value for each alternative. The AHP method (Saaty [1972]) also produces such a value function, but in view of the greater simplicity of SMART and the controversy surrounding the AHP (Dyer [1990]), SMART falls more in our terms of reference.

The simplest form of the SMART technique will be considered and proceeds as follows:

1. Weight assessment:

In order to assign weights to the different criteria in the decision problem, the following possibilities exist:

- (i) The decision maker is asked to rank the criteria in order of importance, assigning a value of 10 to the least important criterion and increasing this assigned value according to the rate by which the importance increases according to the decision maker. These values then constitute importance weights for the criteria.
- (ii) The decision maker can alternatively rank the criteria in order of importance, assigning rank numbers, R_i , ranging from 1 (most important) to k (least important). By

using the rank sum weighting procedure, weights are estimated from:

$$w_i = (k+1-R_i)/\Sigma R_i$$

(iii) The concept of swing weights can also be used. The decision maker is presented with the ideal and nadir points and the nadir point is assumed to be the starting solution. The user is allowed to change only one criterion from its worst to its best level. He has to indicate which one should be changed. After this change has been performed, the next one to be changed must be specified. This process is repeated until the last criterion to be changed has been indicated. It is implicitly assumed that the criterion that seems to make the most difference in value will be changed first, the one with the second greatest impact next and so on. Ratio-scale weights are then obtained from these rank orders by arbitrarily assigning a value difference of 100 points to that criterion that was selected first for improvement. Equally arbitrarily assign a value difference of 0 to the criterion for which it would make no difference if one moved it from worst to best. All other value differences are then expressed as percentages of 100.

2. The assigned weights of step 1 are then normalized as follows:

$$w_i = \frac{w_i}{\Sigma w_i}$$

where w_i is the unnormalized ratio weight assigned in step 1.

3. The different alternatives are then evaluated on a scale of 0 to 100 on each criterion, with the best one receiving a value of 100 and the worst alternative a value of 0. The actual assessment of each alternative on the different criteria are thus converted to value numbers,

$$v^i = (v_1^i, v_2^i, \dots, v_k^i) \quad \begin{array}{l} k = \text{number of criteria} \\ i = 1 \dots p \end{array}$$

This can either be provided directly by the decision maker or alternatively can be determined indirectly by interpolating between 0 and 100 according to the actual

evaluation of the alternative in terms of the specific criterion.

4. The simple additive model is then used to aggregate the weights of step 2 and the value numbers of step 3 to obtain a single importance value for each alternative as follows:

$$x^i = \sum_j w_j v_j^i \quad i = 1, \dots, p$$

5. The alternatives are then ranked according to the x_i values. If the decision maker is not satisfied with the outcome, any of the values assigned in the previous steps can be changed, leading to a different outcome. The visual interactive sensitivity aid (VISA) developed by Belton and Vickers [1990] can be used to assist the decision maker in changing previously assigned evaluations (see discussion in next section).

(i) Comments

1. Initially SMART was justified by its simplicity, later by studies that demonstrated the robustness of additive multiattribute models, and still later by difference measurement, which gave it theoretical support.
2. SMART actually comprises a number of techniques rather than a single procedure with the main communality being their reliance on direct numerical estimation methods.
3. Due to the fact that each alternative is in essence re-evaluated on a scale of 0 to 100, the actual scale of measurement (nominal, cardinal or ordinal) has no restrictive effect on the range of applications of the method.
4. The method in effect makes use of trade-off information provided by the decision maker in the form of the ratings given in step 3 as the decision maker should keep in mind that the rating provided should reflect how far in value the level under consideration is from the extremes. This way of assigning scores to alternatives has the characteristic that if the decision maker wishes to extend the analysis to include an additional alternative, rescaling of all criteria values may be needed to maintain a 0 to

100 scale and a consequent reevaluation of criteria weight. This is a recognition of how real decision making works – if new alternatives are introduced which shift the ranges of values available, then values and preferences can and frequently do change. An alternative approach is to define a *global scale* on which the 0 and 100 points are defined by some globally worst and best possibility for each criterion. Decision makers are, however, hardly ever concerned with the range of possible values, and are only concerned with the actual range of values defined by the specific decision problem.

5. The method also can be used for ranking alternatives for which the criteria form a hierarchical structure – the lower levels of criteria are evaluated according to their contribution to the higher level ones and the final weights of the lowest level of criteria are obtained by multiplication of the lower level weights by the upper level ones – as with the AHP method.

6. The (v^i) 's determined in step 3 express both the order of preference and the relative strength of preference among the alternatives.

7. The method, using rating and ratio estimation procedures, is based on direct numerical judgments and can ideally be applied to any set of alternatives, but is especially useful if the set is relatively small. If a natural scale, however, exists the decision maker may tend simply to match one set of numbers with another one, which can be seen as a disadvantage of the method. This can, however, be overcome to a large extent by using graphical displays as in VISA.

(ii) Conclusion

Due to everything that has been said and discussed above, we can conclude that the SMART methodology does not rely on a high level of mathematical or computer sophistication and is easily understood by any type of decision maker. Its application range is not restricted in any way. The CPU time needed for reaching a solution is also minimal. Various ways for eliciting the evaluations of the alternatives exist (direct rating and ranking; swing weighting) and by determining the decision maker's needs and

abilities via a DSS, the best-suited way can be implemented. This flexibility is quite a strong advantage of the method, especially for implementation into a DSS.

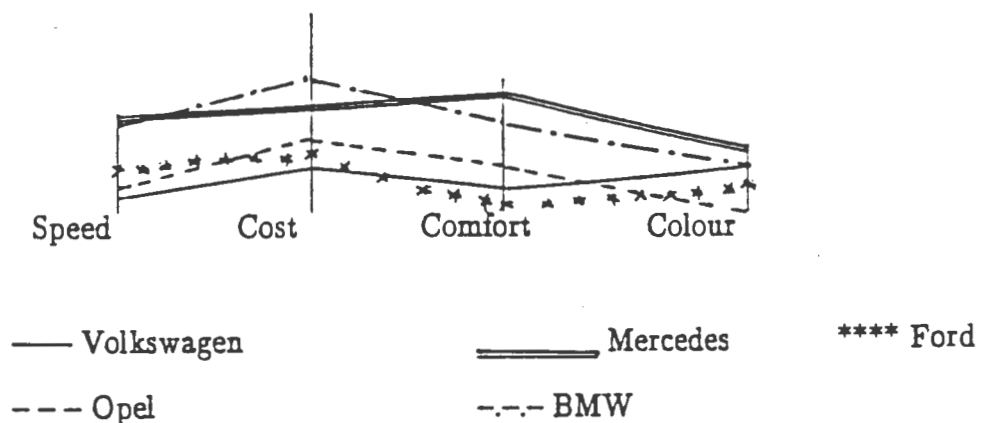
The method is definitely suited for implementation into a DSS.

5.4.3 Visual Interactive Sensitivity Analysis (VISA)

VISA is an implementation of the SMART methodology, incorporating and emphasizing sensitivity analysis using graphical representations. These representations can be used for assisting decision makers in determining what action to take in order to have the desired effect. The analysis is based on a simple weighted multi-attribute value function.

The use of graphical representations enhances the conceptualization of the problem by the decision maker (see Chapter 3, section 3.3). One way of achieving this right from the start of the decision process is by displaying the evaluations of the alternatives by the different criteria in the following way:

Each criterion is represented by a vertical line, the height of which depicts the importance of that specific criterion, and the performance of each alternative is illustrated by the point at which the line depicting its performance crosses the criterion line. A graphical representation of the following form is thus displayed:

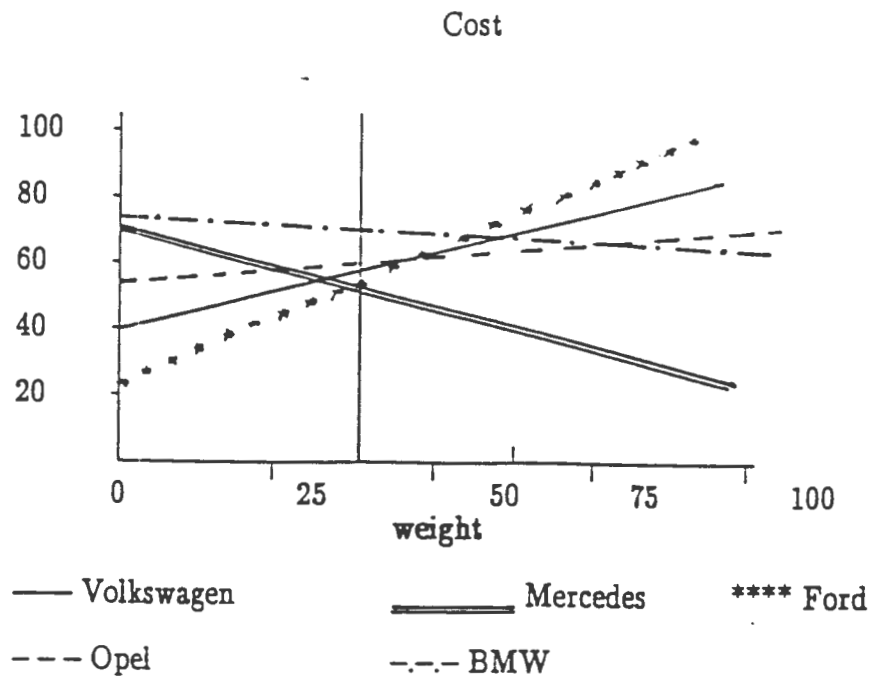


This allows the decision maker to identify the alternative with the highest score on the most significant criteria, although it may have significantly lower scores on all of the

other criteria. The decision maker can thus see at a glance which alternative has the most desired value for the different criteria.

It is very useful to do a thorough sensitivity analysis on the criteria weights assigned. One way of doing this is as follows:

A graph is displayed for each criterion with the horizontal axis representing the weight assigned to the selected criterion and the vertical axis the overall score (according to some aggregation function).



By moving the vertical line, depicting the weight, the effect of the weight assigned to that specific criterion, on the overall ranking of the alternatives can be observed. For example, in the above representation, the BMW is ranked first for a weight of 25 attached to cost. By moving the vertical line to the right, thus increasing the weight to 75 (say), the Opel will be ranked first.

By allowing the decision maker to change the weights interactively as well as graphically he is assisted and invited not to accept the outcome of the analysis unquestioningly.

According to Belton and Vickers [1990], the effectiveness of the decision making process is enhanced and 'the increased likelihood of implementation through the use of the visual interactive interface is not at the expense of a reduction in the quality of decisions.'

The sensitivity idea of VISA is applicable to and of vital importance in many other methodologies, such as TOPSIS and ELECTRE and will thus be implemented as a separate module which can be called from various stages in the DSS.

5.5 Outranking methods

5.5.1 The ELECTRE method

The ELECTRE class of methodologies can be considered as the classical outranking type of method. The ELECTRE I method proceeds as follows:

1. The decision matrix is again normalized as in the case of the TOPSIS method discussed previously to form the normalized decision matrix, R with elements:

$$r_j^i = z_j^i / \sqrt{\sum (z_j^i)^2} \quad i = 1, 2, \dots, p; j = 1, 2, \dots, k$$

2. For validating a comprehensive outranking relation S, we need to characterize what is usually referred to as the "greater of lesser importance" given to each criterion (see Roy [1989]). An importance coefficient therefore needs to be assigned to each criterion. These coefficients are intrinsic to the criteria and do not depend on the nature of the scale chosen for evaluating performances of the criteria. They are not used in a compensating manner. These coefficients, w_j are provided by the decision maker (either directly or by giving rank numbers for the criteria – see the section on weight assessment discussed for the SMART procedure).

3. For each pair of alternatives z^t and z^s with corresponding scores r_t and r_s , the set of criteria is divided into two distinct subsets. The concordance set C_{ts} of z_t and z_s is composed of all criteria for which z^t is preferred to z^s . Thus:

$$C_{ts} = \{j | r_j^t \geq r_j^s\}$$

The complimentary subset is called the discordance set, which is defined as:

$$D_{ts} = \{j | r_j^t < r_j^s\}$$

4. The relative value of the concordance set is measured by the concordance index which is equal to the sum of the weights associated with those criteria contained in the concordance set. Therefore the concordance index c_{ts} between z^t and z^s is defined as:

$$c_s^t = \frac{1}{w} \sum_{j \in C_{ts}} w_j \text{ with } w = \sum_j w_j$$

This index reflects the relative importance of z^t with respect to z^s . A higher value of c_s^t indicates that z^t is preferred to z^s as far as the concordance criteria are concerned. The concordance matrix C is formed from these values with element (t,s) of this matrix depicting the c_s^t value described above.

5. So far no attention has been paid to the degree to which the evaluations of a certain z^t are worse than the evaluations of competing z^s . Therefore a second index, the discordance index, has to be defined:

$$d_s^t = \frac{\max_{j \in D_{ts}} |r_j^t - r_j^s|}{\max_{j \in J} |r_j^t - r_j^s|}$$

A higher value of d_s^t implies that, for the discordance criteria, z^t is less favorable than z^s . The discordance indices form the discordance matrix D with element (t,s) of this matrix being the d_s^t value defined above.

It should be noted that information contained in the concordance matrix differs significantly from that in the discordance matrix, making the information content of C and D complementary; differences among weights are represented by means of the

concordance matrix, whereas differences among criteria values are represented by means of the discordance matrix.

6. The concordance dominance matrix is calculated with the aid of a threshold value for the concordance index. z^t will only have a chance of dominating z^s if its corresponding concordance index c_s^t exceeds at least a certain threshold value \bar{c} , that is $c_s^t \geq \bar{c}$.

This threshold value can initially be determined as the average concordance index (for example):

$$\bar{c} = \frac{\sum_t \sum_s c_s^t}{k(k-1)} \text{ where } k = \text{number of criteria}$$

A boolean matrix F is then constructed, with elements defined as:

$$\begin{aligned} f_s^t &= 1 \text{ if } c_s^t \geq \bar{c} \\ f_s^t &= 0 \text{ if } c_s^t < \bar{c} \end{aligned}$$

7. The discordance dominance matrix, G is constructed in a way analogous to the F matrix on the basis of a threshold value \bar{d} to the discordance indices. The elements of this matrix are calculated as:

$$\begin{aligned} g_s^t &= 1 \text{ if } d_s^t \leq \bar{d} \\ g_s^t &= 0 \text{ if } d_s^t > \bar{d} \end{aligned}$$

where

$$\bar{d} = \frac{\sum_t \sum_s d_s^t}{k(k-1)}$$

The decision maker will have to have the ability to interactively change the threshold values \bar{c} and \bar{d} in order to determine their effect on the final outcome of the procedure. The decision maker can be assisted in doing this by using the ideas of Vetschera [1986].

8. The intersection of F and G is calculated next and the elements of the resulting matrix, E is defined as

$$e_s^t = f_s^t \cdot g_s^t$$

9. The aggregate dominance matrix E gives the partial–preference ordering of the alternatives. If $e_s^t = 1$, then z^t is preferred to z^s for both the concordance and discordance criteria, but z^t still has the chance of being dominated by other alternatives.

Hence, the condition that z^t is not dominated by ELECTRE procedure is,

$$\begin{aligned} e_s^t &= 1 \text{ for at least one } s, s = 1, 2, \dots, k; t \neq s \\ e_t^i &= 0 \text{ for all } i = 1, 2, \dots, p; i \neq t, i \neq s \end{aligned}$$

Many other partial rank orders are also possible.

(i) Comments

1. The input needed from the decision maker can easily be provided as either weights for the criteria or by ranking the criteria in order of importance.
2. The method examines both the degree to which the preference weights are in agreement with pairwise dominance relationships and the degree to which weighted evaluations differ from each other.
3. A possible weak point of the method is the use of concordance and discordance threshold values which may be difficult for the decision maker to provide, although their impact on the final solution may be significant. A sensitivity analysis can be performed on these values, explaining to the decision maker the effect of changing them (Vetschera [1986]).
4. The ELECTRE I method structures a partial ordering of alternatives which is stronger than the incomplete ordering implied by nondominance, but still allows some incomparability to remain.
5. The method does not require the preference relationships to be transitive – it recognizes that the reasons which allow one to decide z^1 is preferred to z^2 and those which allow z^2 to be preferred to z^3 may be too distinct to allow z^1 to be preferred to z^3 .

(ii) Conclusion

The ELECTRE type of methodology can be considered to be one of the best multicriteria decision methods. It is based on simple logic and can be used irrespective of the type of criteria used for evaluating the alternatives (ordinal, nominal or cardinal), thus increasing the area of applicability. A fair amount of assistance can be given to the decision maker during an interactive session with the method. Although only a partial ordering of alternatives is constructed using the ELECTRE I methodology, the size of the feasible region is reduced, allowing the decision maker to more easily reach a final decision – either directly or by applying a different method to the reduced set of alternatives.

The ELECTRE I method is thus ideally suited for implementation into a DSS.

5.5.2 The PROMETHEE method

The class of methodologies that can be referred to as PROMETHEE can also be seen as belonging to the class of outranking methodologies and proceeds as follows:

1. The decision maker has to associate a generalized criterion to each criterion describing the alternatives of the decision problem. These extensions of the notion of criterion is based on the introduction of a preference function giving the preference of the decision maker for an alternative z^1 with regard to alternative z^2 . Such a function is defined separately for each criterion with a value between 0 and 1 – the smaller the value, the greater the indifference, and the closer to 1 the greater the preference of the decision maker for one alternative above the other. This function can be defined as follows:

let $f(\cdot)$ be a particular criterion, criterion h , say and z^1 and z^2 two particular alternatives. The associated preference function $P_h(z^1, z^2)$ of z^1 with regard to z^2 will be defined as:

$$P_h(z^1, z^2) = \begin{cases} 0 & \text{if } f(z^1) \leq f(z^2) \\ p[f(z^1), f(z^2)] & \text{if } f(z^1) > f(z^2) \end{cases}$$

It seems reasonable to choose for $p(\cdot)$ functions of the type:

$$p[f(z^1), f(z^2)] = p[f(z^1) - f(z^2)].$$

The following six types of functions cover most of the assumed forms that are supposed to represent different types of strength of preferences (we write $x = f(z^1) - f(z^2)$):

(i) Usual criterion:

$$p(x) = \begin{cases} 0 & \forall x \leq 0 \\ 1 & \forall x > 0 \end{cases}$$

As soon as $f(z^1)$ and $f(z^2)$ are different, the decision maker has strict preference for the alternative with the greater value.

(ii) Quasi criterion:

$$p(x) = \begin{cases} 0 & -r \leq x \leq r \\ 1 & x < -r \text{ or } x > r \end{cases}$$

z^1 and z^2 are indifferent as long as the difference between $f(z^1)$ and $f(z^2)$ does not exceed r ; else the preference becomes strict. The value r can be seen as the greatest value of the difference between two evaluations, below which the decision maker considers the corresponding alternatives to be indifferent.

(iii) Criterion with linear preference:

$$p(x) = \begin{cases} x/m & -m \leq x \leq m \\ 1 & x < -m \text{ or } x > m \end{cases}$$

This allows the decision maker to prefer progressively z^1 to z^2 for progressively larger deviations between $f(z^1)$ and $f(z^2)$. The intensity of preference increases linearly until this deviation equals m ; after this value the preference is strict. The preference threshold, m , can thus be seen as the lowest value above which the decision maker

considers that there is strict preference of one of the alternatives.

(iv) Level criterion:

$$p(x) = \begin{cases} 0 & |x| \leq r \\ 1/2 & r < |x| \leq m \\ 1 & m < |x| \end{cases}$$

If the difference between the evaluations lies between r and m , there is a weak preference situation.

(v) Criterion with linear preference and indifference area:

$$p(x) = \begin{cases} 0 & |x| \leq r \\ (|x| - r)/(m-r) & r < |x| \leq m \\ 1 & m < |x| \end{cases}$$

In this case the decision maker considers that his preference increases linearly from indifference to strict preference in the area between the two thresholds r and m .

(vi) Gaussian criterion:

$$p(x) = 1 - \exp(-x^2/2\delta^2)$$

The preference of the decision maker still grows with the deviation x . The value of δ , the standard deviation, is not always easily determined, although experience obtained with the Normal distribution in Statistics could assist in attaching a value to it. It can be defined as the distance between the origin and the point of inflexion of the curve of differences.

3. For each pair of alternatives z^1 and z^2 a preference index for z^1 with regard to z^2 over all criteria, is defined:

$$\pi(z^1, z^2) = 1/k \sum_h P_h(z^1, z^2) \quad \text{where } k = \text{number of criteria.}$$

This index gives a measure of the preference of z^1 over z^2 for all the criteria; the closer to 1, the greater the preference.

4. Define for each alternative the outgoing flow as follows:

$$\phi^+(z^1) = \sum_{x \in S} \pi(z^1, x)$$

and the incoming flow as:

$$\phi^-(z^1) = \sum_{x \in S} \pi(x, z^1)$$

where S is the set of alternatives.

The larger $\phi^+(z^1)$, the more z^1 dominates the other alternatives. The smaller $\phi^-(z^1)$, the less z^1 is dominated.

5. By defining a total preorder as some relation imposed on the set of alternatives the decision maker is allowed to do a distinct comparison between each pair of alternatives. Using this preorder, all alternatives are comparable, justifying the expression "**total preorder**". Define the two total preorders (P^+, I^+) and (P^-, I^-) such that:

$$\begin{aligned} z^1 P^+ z^1 & \text{ iff } \phi^+(z^1) > \phi^+(z^1); \\ z^1 P^- z^2 & \text{ iff } \phi^-(z^1) < \phi^-(z^2); \\ z^1 I^+ z^2 & \text{ iff } \phi^+(z^1) = \phi^+(z^2); \\ z^1 I^- z^2 & \text{ iff } \phi^-(z^1) = \phi^-(z^2). \end{aligned}$$

A partial preorder is defined similarly to a total preorder, although provision is made for some pairs of alternatives to remain incomparable. The following partial preorder $(P^{(1)}, I^{(1)}, R)$ is then obtained by considering their intersection:

$$z^1 \text{ outranks } z^2 \text{ (} z^1 P^{(1)} z^2 \text{):} \quad \text{if } \begin{array}{l} z^1 P^+ z^2 \text{ and } z^1 P^- z^2, \\ z^2 P^+ z^2 \text{ and } z^1 I^- z^2, \\ z^1 I^+ z^2 \text{ and } z^1 P^- z^2, \end{array}$$

$$z^1 \text{ is indifferent to } z^2 \text{ (} z^1 I^{(1)} z^2 \text{):} \quad \text{if } z^1 I^+ z^2 \text{ and } z^1 I^- z^2,$$

z^1 and z^2 are incomparable ($z^1 R z^2$): otherwise

This is the PROMETHEE I partial relation.

If a total preorder has been requested by the decision maker, the net-flow for each alternative can be considered as follows:

$$\phi(z^1) = \phi^+(z^1) - \phi^-(z^1)$$

which can easily be used for ranking the actions:

$$\begin{array}{ll} z^1 \text{ outranks } z^2 & \text{iff } \phi(z^1) > \phi(z^2) \\ z^1 \text{ is indifferent to } z^2 & \text{iff } \phi(z^1) = \phi(z^2) \end{array}$$

This is the PROMETHEE II complete relation. All the alternatives of S are now completely ranked.

(i) Comments

1. The parameters to be fixed by the decision maker can be seen as a major disadvantage of the PROMETHEE class of methods. They may however have an economic significance to the decision maker which makes the determination of their values easier. By carefully constructed questions posed to the decision maker, it is possible to determine their respective values.
2. The mathematical sophistication of the method is more complex than some of the other methods discussed, but still not a major drawback.
3. A form of sensitivity analysis can be performed on the threshold values, discussing the influence of changing them, on the final ranking of the alternatives.
4. It was determined by Brans *et al* [1984] that the size of the problem does not influence the sensitivity of the rankings. The difficulty (the disagreement between the criteria) and the proximity (the average distance between a difference of two evaluations and the thresholds) of the problem are two determinant factors of instability.

5. Weights, or alternatively rank numbers to be converted into weights, for the different criteria can be used in the calculation of the preference index (step 3) which may prove the final ranking to be more realistic.
6. The interaction with the decision maker is minimum and involves only the determination of the threshold values. These can be seen as some form of trade-off information provided.
7. The method is equally applicable to criteria measured on a cardinal, ordinal or nominal scale.

(ii) Conclusion

Due to the possible difficulty of fixing the threshold values needed in the method and the possible aversion of the decision maker in doing this, the PROMETHEE methods are not ideally suited for implementation into a DSS. However, by providing the decision maker detailed assistance in determining their values (by explaining their meaning in detail and eliciting their values by means of simple questions), this difficulty will hopefully be overcome. Extensive use of graphical representations depicting the change in the outcome of the problem as these values are altered (sensitivity analysis) will also enhance the usefulness and willingness to solve the decision problem using PROMETHEE.

5.6 Conclusion

The above discussion represents a selective number of methodologies existing in the literature for solving discrete multicriteria decision making problems. It is by no means a comprehensive discussion, but an attempt was rather made to discuss the most applicable and useful approaches for implementation into a decision support system.

The aim of the DSS to be developed, as was mentioned before, is to support and facilitate the process of constructing a good decision. In order to achieve this, provision need to be made for as wide a variety of decision making styles as possible. A representative set of methodologies allowing for a fully compensatory approach, a goal-based approach as well as an outranking approach is therefore needed. By selecting such a set, different types of input information can be supplied by the decision maker, including goal levels, pairwise comparisons, trade-off information and strength of preference information. Provision should also be made to allow the decision maker to either supply this input information *a priori* to the actual decision process or alternatively *a posteriori* to the process itself. One further important aspect to consider in selecting which methodologies to implement, is the form of the output desired by different decision makers. This includes a full ranking of the entire set of alternatives, a single alternative as 'best' choice and a partial ranking of the set of alternatives. In summary, the set of methodologies chosen should restrict the decision maker as little as possible in reaching the desired outcome.

Considering all of the above criteria, the following methodologies are implemented: STEM, PRIAM, TOPSIS, the combined approach, SMART, ELECTRE and PROMETHEE.

Where possible, the visual interactive sensitivity ideas will be used to enhance the decision maker's perception and understanding of the problem as well as the output.

The challenge of implementing them into a DSS now remains and the degree of success thereof will only become apparent after proper testing procedures have been performed.

CHAPTER 6 – Implementation

6.1 Introduction

In implementing the methodologies discussed in the previous chapter, the decision process is perceived to follow the steps mentioned below:

- (i) define the list of alternatives;
- (ii) define the list of criteria;
- (iii) attach specific values, using the list of criteria, to each alternative;
- (iv) define the format of the output desired;
- (v) define the preference information that can be supplied;
- (vi) execute the proposed methodology; and
- (vii) evaluate the output produced and start process again if desired.

The interactive system is menu-driven and provides for the above steps in the form of menus presented to the decision maker. Provision is made for seven options in the main menu:

- (i) Set-up
- (ii) Alternatives
- (iii) Criteria
- (iv) Combined file
- (v) Solve
- (vi) Output
- (vii) Exit/Save

The alternatives and criteria are stored in two separate files, termed the alternative and the criteria file, respectively. These files are then combined into one file, termed the combined file, which can be seen as the file describing the decision problem entirely.

This file is used as input to the methodology that will be executed. Various actions can be performed on these three files and are discussed in separate sections, according to the specific option on the menus presented to the decision maker.

This chapter is divided into seven sections, each describing one of the above-mentioned options. Ample use is made of illustrations to enable the reader to follow the text descriptions more easily.

The use of the system is explained, by applying a simple example, in Appendix I.

6.2 Program start-up

The system can only be run from a hard disk. To do this the user needs to create a directory called MCDM (say) on the hard disk and copy all the files, on the floppy disks supplied, into this directory. The program is then executed by typing MCDM.

Once this has been done, an introductory screen is displayed (see Figure 6.1). By pressing any key a menu is displayed across the screen showing the seven options available to the user (see Figure 6.2). A specific option is selected by moving the bar or cursor over the desired choice, using the `Left_Arrow` or the `Right_Arrow` key, and then pressing the `Return` key. The action linked to the chosen option will then be executed. The seven options will be discussed separately in the remainder of this user guide.

6.3 Conventions used in the system

Certain keys do have specific functions throughout the use of the system:

- (i) The **Left_Arrow**, **Right_Arrow**, **Down_Arrow** and **Up_Arrow** keys can all be used to move the cursor in the desired direction.
- (ii) The **Return key** is used to select a specific action from a menu. It also has to be pressed in order to save new or modified data.
- (iii) The **Esc key** will always return the user to the previous screen displayed and is also used in some instances to cancel a chosen operation. Messages to this effect will appear on the screen where applicable.
- (iv) The **F1 key** is used to access the help screens that have been implemented.

6.4 Options on the main menu

6.4.1 Set-up

Whenever the system is used, it is important to specify the disk drive and the directory from which files containing the needed information will be read, and on which new files will be saved. Therefore, each time the system is started afresh, this option has to be executed immediately. The user can then either choose the current drive and directory or alternatively choose to use a different set-up by selecting the second option on the window displayed (see Figure 6.3). The user must then enter a new drive and directory in the following format: 'C:\NEW'. C: depicts the drive specification and NEW the directory, on the C: drive, to be used (see Figure 6.4).

When any of the other options on the main menu is chosen without execution of the set-up option first, a message will be displayed, informing the user to run the set-up option (see Figure 6.5).

6.4.2 Alternatives

By choosing this option from the main menu, the user has the opportunity to perform various actions on the file containing the alternatives. This file contains a list of the possible outcomes to the decision problem.

The sub-menu shown in Figure 6.6 is displayed when this option is chosen. Each one of the sub-options of this menu will now be discussed in detail:

(i) Create file

If the user is faced with a new decision problem, the list of alternatives from which a decision will have to be made, needs to be stored in a file. This file is created by executing the first option. The user is prompted to enter the name of the file to be created, using a maximum of eight characters (see Figure 6.7). If a file with this name already exists in the directory specified in the set-up option, the user is informed thereof and can either cancel the creation operation or overwrite the existing file (see Figure 6.8).

As soon as a valid filename has been specified, the screen shown in Figure 6.9 is consecutively displayed, allowing the user to enter as many different alternatives as desired. By entering a 'blank alternative' description, the system assumes that all needed alternatives have been supplied, and the user is returned to Figure 6.6. There is no limit to the number of alternatives that can be entered.

(ii) Modify file

The user can modify any existing file of alternatives, by choosing the second option on the menu shown in Figure 6.6. A list of all the existing alternative files is displayed (see Figure 6.10). The user has to choose the desired one by placing the cursor, using the Up_Arrow and/or Down_Arrow keys, over the desired filename. As soon as this has

been done, two additional options become available, as is shown in Figure 6.11.

Add a new alternative

By choosing this option, the user can add as many additional alternatives, to those already existing, as desired. A similar screen to the one shown in Figure 6.12 is displayed and by entering a blank description the system again assumes that the operation has been completed. If a description is entered that already exists in the filename specified, the system will not add the duplicate alternative to the file.

Delete an existing alternative

This option is chosen whenever an alternative needs to be deleted from an already existing alternative file. The user has to enter the description of the alternative to be deleted on a screen similar to the one depicted in Figure 6.13. If a description for an alternative, that does not exist in the file chosen, is entered, the user is informed thereof (see Figure 6.14) and can either cancel the deletion operation or enter a different name. If a valid description has been supplied, the specific alternative will be permanently removed from the file.

(iii) View file

Anyone of the existing alternative files can be viewed by choosing this option from Figure 6.6. A list of all existing filenames is displayed (see Figure 6.10) from which one has to be chosen. A screen similar to the one shown in Figure 6.15 is then displayed and the user can scroll up or down by using the `Up_Arrow` and `Down_Arrow` keys respectively.

(iv) Delete file

By choosing this option, the same list of filenames is displayed (see Figure 6.10) and the user has to choose the desired one. On pressing the Return key, the specific file will be

permanently removed from the hard disk. Alternatively, the user is returned to Figure 6.6 by pressing the Esc key.

6.4.3 Criteria

The alternatives, from which a decision has to be made, are evaluated by a number of criteria. This option on the main menu allows the same actions as discussed for the alternatives (see Section 5), to be performed on the file(s) containing the criteria. They include: 'create a new file', 'modify an existing file', 'view an existing file' and 'delete an existing file'.

The steps that can be performed by the user are identical to those already discussed. (Wherever the word 'alternatives' is used it is to be substituted by 'criteria'.) The only difference is in the way the criteria are described – besides a descriptive name, the user also needs to state whether the specific criterion is to be minimised or maximised. This has to be done whenever a new criterion name is specified (see Figure 6.6).

6.4.4 Combined file

The criterion and alternative files are created independently from each other and has to be combined to form the file describing the specific decision problem completely. This is done by choosing the fourth option from the main menu (see Figure 6.2).

A similar menu to the one shown in Figure 6.6 is displayed allowing similar actions to be performed on the combined file.

(i) Create file

A new file is created by combining any existing alternative file with any existing criterion file. When this option is therefore chosen, the user has to supply a filename on a screen similar to the one shown in Figure 6.17. The user is informed if a file by the

selected name already exists and can either cancel the operation or enter a different filename (see Figure 6.8). If a valid name has been supplied, the screen shown in Figure 6.18 is displayed. Both options on this menu have to be executed in order to create the file specified. The order in which these options are executed is irrelevant.

For both options, a list of all existing alternative and criterion files, respectively, is displayed from which one of each has to be chosen (see Figure 6.10). As soon as both options have been performed, a screen similar to Figure 6.19 is displayed for each alternative in the chosen alternative file. The evaluations for that alternative, by the different criteria forming part of the chosen criterion file, must then be supplied by the user. By pressing the Escape key, the evaluations for the next criterion has to be supplied. This is repeated until all alternatives have been evaluated on all criteria.

As soon as all evaluations have been supplied, the user is returned to Figure 6.6.

(ii) Modify file

By choosing this option, a list of all existing combined files is displayed (see Figure 6.10) from which a specific one has to be selected using the Up_Arrow, Down_Arrow and Return key. As soon as this has been done, a screen similar to the one shown in Figure 6.20 is displayed. The user is then allowed to change any information displayed by moving the cursor over the field and modifying the data. The existing information are then overwritten by the new data.

(iii) View file

A list of all existing combined files is displayed and on choosing a specific one a screen similar to the one shown in Figure 6.20 is displayed. The user is then allowed to scroll up or down using the Up_Arrow and Down_Arrow keys. No changes to the displayed information are, however, allowed.

(iv) Delete file

The same list of all existing combined files is displayed (see Figure 6.10) and a specific one has to be chosen. As soon as the Return key has been pressed, the chosen file will be permanently removed from the hard disk.

6.4.5 Solve

On selecting this option, a choice has to be made between new and experienced users of the system (see Figure 6.21).

(i) New user

When the user chooses this option, a list of all decision problems that have been created (combined files) is displayed (see Figure 6.10) from which the specific one has to be selected. The screen depicted in Figure 6.22 is then displayed, allowing the user to specify the type of output desired. The output desired can range from a single 'best' alternative to ranking the entire list of alternatives in order of importance. As was discussed in Chapter 3, the output desired is one of two crucial factors needed for deciding which methodology to use for solving the decision problem.

A series of questions is then posed to the user in order to determine the type of input information he is able to supply. This information forms the other crucial factor in determining the specific methodology to be used for reaching a decision. The various questions are displayed in Figure 6.23 to Figure 6.28. A schematic display of the questions asked is given in Figure . As soon as the system has enough information 'to decide' what methodology to use, the name of the recommended solution procedure is displayed (see Figure 6.29) and by pressing the Return key, that methodology will be executed. Alternatively, the Esc key can be pressed if the method is not to be executed and the user wants to return to the previous screen.

(ii) Experienced user

When this option is chosen, it is assumed that the user is familiar with the procedures to be followed that will allow the system to recommend a specific methodology. This becomes most apparent in the choice of input information to be provided – the user is directly prompted by a menu of options as opposed to prompting a new user with questions leading him towards supplying the relevant information.

A list of all existing combined files is displayed (see Figure 6.10) allowing the user to choose the specific problem to be solved. The user then gets the opportunity to specify the type of input information he is willing and able to supply by choosing the specific option from the menu displayed in Figure 6.30. The user is then prompted to choose the type of output he desires from a menu as displayed in Figure 6.22. Depending on what has been chosen, the system informs the user of the specific methodology that will be used for solving the decision problem (see Figure 6.29) and by pressing the Return key that methodology will be employed.

A discussion of the screens applicable to the various methodologies follows. The discussion is done according to the type of input provided and the methodology employed for solving the decision problem.

(iii) Input needed**(a) Ranking or weighting criteria**

If the user is able to supply some form of ranking information on the criteria, the TOPSIS (Hwang and Yoon [1981]) or ELECTRE methodology (Goicoechea *et al* [1982]) is employed. The user has a choice of only ordering the criteria in terms of their importance or supplying numeric values (weights) depicting their relative importance

(see Figure 6.31). In the case where the criteria can only be ordered, this information is supplied on a screen similar to Figure 6.32 and when weights can be supplied, the user has to complete a screen similar to Figure 6.33.

The information supplied can be altered by the user if he is not satisfied with the solution reached using the supplied information. This is achieved by entering an 'N' (for No) to the prompted question (see Figure 6.35).

(b) Trade-off information

If a single choice or an identification of a subset of alternatives is desired as output (see Figure 6.22), the STEM methodology is proposed as the best methodology. On selecting to rank all alternatives, the PROMETHEE II methodology has been implemented. For any other type of output, the PROMETHEE I methodology (Brans and Vincke [1985]) is proposed.

Procedure followed for the STEM methodology

The ideal and nadir alternatives are displayed to the user on the screen depicted in Figure 6.40. On pressing the Escape key, the current best alternative is calculated and displayed on the screen depicted in Figure 6.41. Any current criteria values can then be changed by the user in an attempt to better unsatisfactory values. This is done on a screen similar to Figure 6.42. If changes have been made such that no alternatives are feasible, the user is informed thereof (see Figure 6.43). However, if some remaining alternatives are still feasible, they are displayed on a screen similar to Figure 6.44. This process is repeated until the user is satisfied that no more changes would improve his current outcome.

Procedure followed for the PROMETHEE class of methodologies

The user is prompted with a series of questions, for each criterion, in order to determine

the specific classification type of that criterion. These questions are closely related to trade-off information and are displayed in Figures 6.36 to 6.39. Depending on the answers to these questions, the most satisfactory solution is determined.

(c) Specify goal levels

The combined approach discussed in chapter 5 is the only methodology implemented when minimum acceptable and/or maximum desired levels for the criteria can be supplied.

A screen is displayed informing the user that the problem is being set-up (see Figure 6.45). As soon as this has been done, a screen similar to that shown in Figure 6.46 is displayed, allowing the user to change the current level of evaluations of specific criteria. Once this has been done, the user is informed of the number of remaining feasible alternatives on the screen depicted in Figure 6.47. If the user is not prepared to accept the alternative displayed as the final solution, he is returned to Figure 6.46 and is allowed to specify different goal levels. This process is repeated until the user is satisfied.

(d) Paired comparison information on the alternatives

If the user can compare any two alternatives and is able to tell which one is the more preferred, only the PRIAM methodology (Levine and Pomerol [1986]) has been implemented.

The user is again confronted by Figure 6.45, while the problem is being set-up. As soon as this has been done, starting levels of achievement have to be supplied for all criteria on a screen similar to Figure 6.48. The screen depicted in Figure 6.34 is then displayed while the first iteration of the methodology is performed. The current best feasible alternative is then displayed on a screen similar to Figure 6.49. The user now has the option of either continuing with determining new and (hopefully) better feasible

alternatives or to return to a previously determined alternative. By choosing to continue, the user gets the opportunity to change any of the desired levels specified previously for the criteria. The user has to indicate whether the changes are imperative or tentative. This is accomplished on a screen depicted in Figure 6.50. As soon as this has been done, the remaining feasible alternatives are displayed (see Figure 6.51) and the most preferred alternative has to be entered by the user. The previous 'best' alternative is then compared with this new one on a screen similar to Figure 6.52. The user is then returned to Figure 6.49.

By choosing to return to previous changes made, the user is referred back to the previous tentative step in the solution process. If the user has gone back to the starting point, Figure 6.55 is displayed and the user gets the opportunity to display the results (see Figure 6.54). By indicating that he is satisfied, the system returns to the main menu, else the solution process is started again.

(e) Strength of preference information

The method that is implemented to consider strength of preference information on the alternatives is called SMART (Edwards and von Winterfeldt [1986]). The user has to supply the strength of preference rate between the alternatives on a screen depicted in Figure 6.55. This rate can range from 0 for the least important to 100 for the most important. The ratio of importance for the different criteria also has to be supplied and can range from 10, for the lowest ranked criterion, to any value for the highest ranked one (see Figure 6.56).

6.4.6 Output

By choosing this option the user can perform certain operations on the output files created when a decision has been reached. (The extensions of the output filenames are directly related to the methodology that has been used for solving the problem, that is

TOP = TOPSIS; PR1 = PROMETHEE1; PR2 = PROMETHEE2; PRI = PRIAM;
STM = STEM; ELE = ELECTRE; SMT = SMART; GL = COMBINED
APPROACH). The menu depicted in Figure 6.57 is displayed. This allows the following operations to be performed:

(i) View file

A list of all output files created, is displayed and a specific one has to be chosen (see Figure 6.58). As soon as a file has been chosen, a screen similar to Figure 6.59 is displayed, allowing the user to view the output created. No modifications are allowed.

(ii) Print file

Anyone of the output files created, can be printed by choosing this option. A list of all existing output files is displayed (see Figure 6.58) and the specific one to be printed has to be chosen. The user then gets the opportunity to either print this file or cancel the operation (see Figure 6.60).

(iii) Delete file

By choosing this option, the same list of output files is displayed (see Figure 6.58) and the specific one that is chosen will be permanently removed from the hard disk.

6.4.7 Exit/Save

The user is returned to the DOS prompt by choosing this option from the main menu (see Figure 6.2). All files that have been created and/or modified during that session of the system are automatically saved to the directory specified in the set-up option.

6.5 Hardware requirements for running the system

The multicriteria decision support system can run on any IBM personal computer (PC) or on a computer that is compatible with the IBM PC. The computer should have at least one floppy disk drive and a hard disk, and should also have at least 640Kb (kilobytes) of memory. Both monochrome and colour screen configurations are allowed although different versions of the system are needed for the different configurations.

If printing is required, any printer, including a laser printer, can be used and has to be linked to the PC via the parallel port.

CHAPTER 7 – Testing and evaluating the Decision Support System

7.1 Introduction

The development of any DSS is ultimately aimed at assisting the decision maker in making a satisfactory decision. Although the perceived needs and desires of decision makers have been taken into consideration in designing and developing the DSS, the system can only really be evaluated when used by real decision makers faced by real world decision problems. A properly structured evaluation strategy therefore needs to be designed.

The number of articles published, addressing this issue is quite limited. Most of them are also mainly concerned with comparing different methodologies using a specified set of criteria. The approach we will follow will be along the same lines, with the exception of using criteria more specifically aimed at evaluating the system as such instead of only the methodologies implemented.

7.2 Evaluation criteria

The first step in designing an evaluation procedure is to determine the criteria that will be used in the evaluation process. For our purposes we propose the following list:

7.2.1 Ease of use

This criterion can be seen as consisting of a number of sub-criteria:

- (i) the general appeal of the interface structure;
- (ii) the logical flow (or lack thereof) of the steps to follow in order to be able to execute a specific methodology;
- (iii) the help facility implemented;

- (iv) the error—tolerance of the system; and
- (v) the level of computer literacy needed to be able to use the system.

7.2.2 The type of input information to be provided

Via this criterion, the aim is to determine whether the user can identify and feels comfortable with providing the information needed for at least one category of the possibilities implemented.

7.2.3 The output provided by the system

This criterion is concerned with all aspects of the system associated with the output. We need to determine whether the possibilities implemented, allowing various classes of output information to be generated, are sufficient for the needs of the decision maker. Furthermore, the way in which this output is presented, needs to be evaluated by the decision maker in order to determine the understandability thereof to the user.

7.2.4 Graphical representations

By using this criterion, the aim is to determine whether the level of graphical representations implemented is sufficient and relevant to the decision maker. The main reason for using graphical displays is to enhance the decision maker's understanding of the different procedures where it is used. It should be established if this aim has been achieved.

7.2.5 The methods implemented

Via this criterion, we are attempting to determine the compatibility of the available procedures to the decision maker's specific decision making style and preference. Another aspect is to determine whether the user has confidence in trying different approaches for solving the same decision problem. The flexibility of the system in that it allows the decision maker to change his mind anytime during the solution process also

needs to be determined.

7.2.6 The elapsed time to reach a final satisfactory solution

A possible way of establishing if the time involved in reaching a satisfactory solution is a drawback of the system, is to determine whether the decision maker loses interest during the course of the solution process. Alternatively, it can be established whether the time taken to reach a solution is justifiable by the quality of that solution.

7.2.7 The confidence in the solution reached

We somehow need to determine how the solution provided compares to the decision maker's intuitive ('gut') feel for the most satisfactory outcome. If the user is prepared to use the system for other discrete decision problems, this may be an indication of his confidence in the solution reached. The degree of confidence is also to some extent related to the degree of control the decision maker is perceived to have over the outcome reached.

7.3 Methodology

In order to evaluate the system on the abovementioned criteria, a group of ten students, acting as decision makers were given two hypothetical decision problems to solve. The problems comprise of a fair number of alternatives and criteria in order to allow the outcome to be non-trivial. The first problem consisted of 31 alternative boarding locations, each one evaluated on 4 different criteria. The second one consisted of 42 alternative cars evaluated on seven different criteria (see Appendix III for a description of the decision problems). The students were allowed some time to ask questions related to the problem and it was assumed that each one had a clear understanding of what is expected from him before using the system and subsequently evaluating it.

A questionnaire was designed (see Appendix IV) for evaluation purposes taking the criteria discussed previously into consideration. The answers to the questions posed were assessed on a point scale of 1 to 10. One further question that was posed to the decision makers was to rank the above criteria in order of importance.

7.4 Analysis of the evaluation process

Ideally, one would have liked to compare the DSS developed with existing systems which allow the user flexibility and guidance in the choice of the MCDM method, from within a variety of such methods. This would have generated defensible hypothesis tests, but unfortunately no such systems are readily available.

Emphasis will therefore rather be on identifying the strong and weak points of the system, particularly in relation to the importance placed on these points, taken across all respondents. This will highlight where improvements in the system are necessary.

As is apparent from the questionnaire (Appendix IV), the system was evaluated on seven different categories, namely:

ease of use, input information needed, output provided, graphical representations, the methodologies implemented, the time taken and the confidence in the solution produced. Different questions were constructed for each of these categories and a single score per respondent for each specific category will be calculated as the average score of these different questions. It was furthermore required from the respondents to indicate the importance of each of these categories in judging the performance of a decision support system such as the one considered. The level of computer literacy of the respondents ranges from three to nine, which simulates a range of decision makers that may want to use the system.

The following matrices of scores can be constructed from the completed questionnaires:

(i) The matrix of average category rating:

The categories are numbered from 1 to 7 and the respondents from 1 to 10:

	1	2	3	4	5	6	7
1	5.25	5	7	5	4.67	6	5
2	5.25	7	5	5	4.67	6	6
3	6	6	4	5	5.5	5	5
4	6.5	5	7.5	4	4	6.5	6
5	6	7	7.5	1	4.75	3.5	5
6	5.25	8	8.5	4	5.7	7	6
7	5.5	7	8.5	4	4.67	5.5	6
8	5.2	6	7.5	8	5.2	6.5	5
9	6.5	6	6.5	6	5.5	5	6
10	4.6	5	5.4	6	5.3	4	5

(ii) The matrix of importance ratings for the seven criteria

	1	2	3	4	5	6	7
1	6	5	7	6	4	4	10
2	5	6	7	4	7	3	10
3	6	7	7	5	8	6	8
4	7	7	7	6	6	4	8
5	8	7	7	6	6	8	7
6	7	8	8	6	7	4	9
7	5	6	8	6	6	4	9
8	8	8	8	7	8	7	10
9	9	8	8	7	9	8	10
10	6	6	9	5	7	5	9

In order to evaluate these results, one can as a first step perform a two-way analysis of variance (ANOVA) on each of the matrices. The results of these analyses are included in Appendix V. In interpreting these results, it can be seen that the first matrix of performance values reveals highly significant differences between categories (significant level = $0.0039 < 0.05$), but not between respondents (significant level = $0.2662 > 0.05$). There are thus real differences in performance across categories. The results, however, have an absolute interpretation which is not significantly user-dependent. By employing the same procedure on the matrix of importance ratings, highly significant differences between categories (significant level = $0.0 < 0.05$) and between respondents (significant level = $0.0002 < 0.05$) are apparent. The absolute values of the importance ratings are thus of lesser interest (as they are clearly respondent-dependent), although the ranking of the criteria by importance is meaningful. These results are re-inforced by the degree of overlapping among the confidence intervals depicted in the tables of means. By averaging these importance ratings across all respondents, the following ranking is determined (with the average value in brackets):

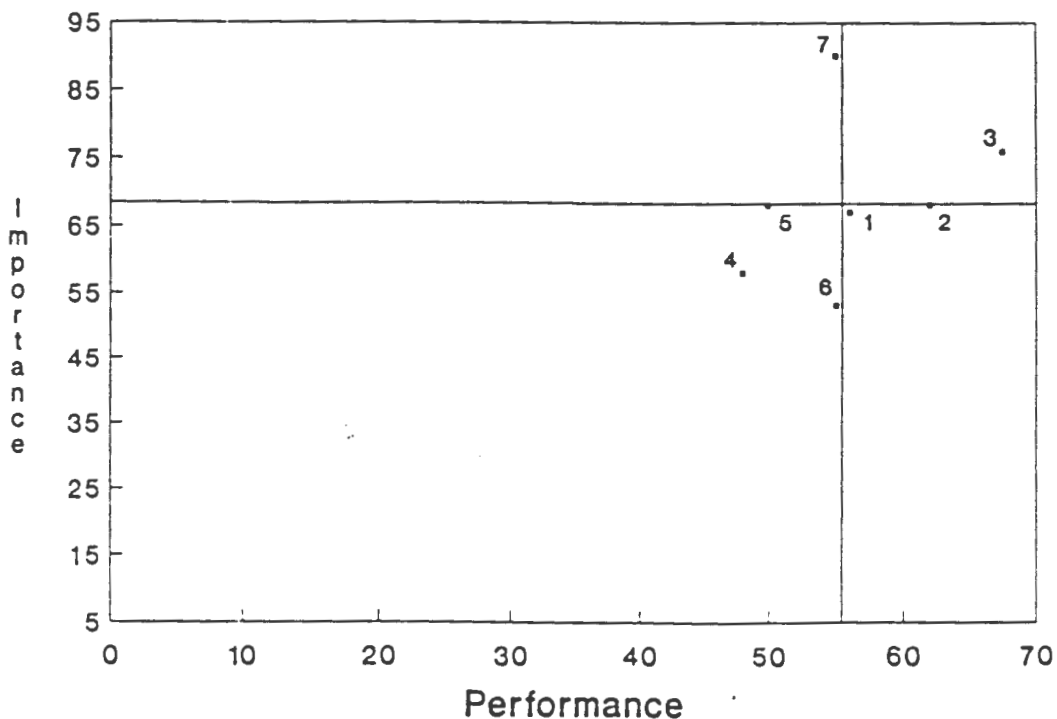
1. the confidence in the solution produced (9)
2. the type of output provided by the system (7.6)
3. the methodologies implemented (6.8) as well as the type of input information to be provided (6.8)
4. the time taken to reach a satisfactory solution (6.7)
5. the graphical representations implemented (5.8) and
6. the ease-of-use of the system (5.3).

Not many similar studies are recorded in the literature which makes the comparison of the above results with other studies very difficult. The results from studies done by Wallenius [1975] and Buchanan and Daellenbach [1987] can, however, be summarised as follows:

The importance of the information provided by the methodology (output) as well as the

confidence in the solution were emphasized in the study by Wallenius [1975]. Also, ease-of-use and the understanding of the methodology were given priority over the time taken to reach a satisfactory solution. The same criteria were also used by Buchanan and Daellenbach [1987], although the relative preference for using each methodology was accorded greater importance than the other criteria, since 'it reflects the decision maker's actual willingness to use the methods and should therefore implicitly include the other criteria.' A study done by Reeves and Gonzales [1989] used the following criteria: quality of solutions generated, simplicity/user friendliness, type of solutions generated, number of iterations needed to identify a satisfactory solution and flexibility. Rothermel and Schilling [1984] measured the performance of three methodologies on the following criteria: ease-of-use, level of understanding and the confidence in each method and in the solutions. In both these latter two studies, it was not expected from the respondents to rank the criteria in order of importance.

In order to identify possible weak areas within the system, the average performance measures can be plotted against the average importance ratings, where the averages are calculated across all respondents. The following XY-scatterplot can be constructed, with the X-axis representing the average performance ratings and the Y-axis the average importance ratings.



The graph can be sub-divided into four quadrants by drawing a line parallel to the X-axis through the median of the performance ratings (5.5) and also drawing a line parallel to the Y-axis through the median of the importance ratings (6.8). The possible weak areas of the system can thus be identified as those categories (indicated by the corresponding letter) for which a score in the top left-hand quartile is depicted. This means that such a category has an average importance measure of at least 6.8 and an average performance rating of not more than 5.5.

Although no category falls strictly into the top left quadrant, three are on or close to its boundary and can thus be identified as possible weak areas of the decision support system, namely the ease-of-use of the system (1), the methods implemented (5) and the confidence in the solution produced (7). These areas need to be investigated in more detail in order to determine future research needs.

As far as the ease-of-use of the system is concerned, possible ways of overcoming the problem, are to have a more applicable help facility implemented at various levels of the system as well as to make more use of graphical representations and/or triggers assisting the decision maker in deciding the order in which actions should be performed. The use of a question-answer approach at all levels should also enhance the understanding and use of the system (especially for novel users thereof). As stated by Klein *et al* [1986], 'There is a need to associate decision styles to questioning methodology', thus catering for a wider range of decision makers.

The confidence in the solution reached was ranked as the most important criterion, although only an average performance rating of 5.5 was determined. The level of confidence is closely linked to the other weak area identified, namely the methodologies implemented. Many respondents commented on the lack of attention given in explaining the methodologies implemented, thus enhancing the decision maker's perception of the

system as being a 'black box'. A more thorough explanation of the specific methodology recommended by the system, as well as how the solution is derived, should bring about a better understanding and a higher level of confidence in the solution produced.

Another drawback of the system and a need for future research, is the inability of backtracking and changing previous inconsistent responses. By increasing this ability, it should enhance the level of control the decision maker has on the solution produced, which should increase the confidence in the solution produced.

However, as stated by Buchanan and Daellenbach [1987], 'the responsibility for achieving a successful man-machine interface does not lie only with the machine.' A certain amount of responsibility also falls with the decision maker to learn his/her preferences as outcomes are progressively revealed and to shed prior expectations, if need be. Any MCDM problem, when first approached by a decision maker, is a potentially ambiguous situation. Gimpl [1985] has argued that intolerance of ambiguity is likely to result in the specification of inappropriate goals, which are to a great extent based on biased information. This causes the decision maker to have less confidence in the solution produced.

7.5 Conclusion

Although the respondents used in evaluating the decision support system, cannot be considered as real-world decision makers, it can be assumed that the weak areas identified by them will also be identified as weak areas by any other group of respondents. It can finally be concluded that although some areas in need of improvement were identified, these areas are not rated sufficiently important to have an overall negative effect on the use of the system. This is further emphasised by the fact that 50% of the respondents indicated that they will use the system to solve similar

decision problems, while the other 50% was unsure. Despite important future research needs identified, the use of the current system for similar discrete decision making problems should not be drastically influenced.

The identified problem areas will be attended to in future versions of the multicriteria decision support system.

CHAPTER 8 – Concluding Remarks

In order to meet the ultimate aim of this thesis, namely the development of a decision support system (DSS) for solving discrete multicriteria decision making (MCDM) problems, the two concepts of DSS and MCDM, were studied in parallel, with specific reference to the discrete decision making problem. It was consequently concluded that it may be possible to combine these concepts by developing a multicriteria decision support system (MCDSS).

The system developed was subsequently tested and evaluated by a group of students, simulating real-world decision makers. Although specific areas in need of improvement were identified as a result of the evaluation process, these shortcomings should not detract from use of the system, even in its current form.

The development of any computerised system is, however, an evolutionary process and as different needs may arise, updated versions of the system will have to be developed. It should, however, be stressed that due to the diverse nature of different decision makers, no system will ever be able to meet all needs and fulfill all desires of the users thereof. The system described in this thesis should, however, meet the most important of these needs and can be regarded as a first step towards filling a gap in the day-to-day decision making problems faced by decision makers.

References

1. Ackoff R L, **The Future of Operational Research is Past**, Journal of the Operational Research Society, Vol 30, No 3, 1979
2. Adams, May and Fagot, **A Model of Riskless Choice**, Behavioral Science, Vol 4, 1959
3. Badran Y, **Preference Ranking of Discrete Multiattribute Instances**, European Journal of Operational Research 34, 1988
4. Balestra G, Tsoukias A, **Multicriteria Analysis Represented by Artificial Intelligence Techniques**, Journal of Operational Society, Vol 41, No 5, 1990
5. Belton V, Vickers S, **Use of a Simple Multi-Attribute Value Function incorporating Visual Interactive Sensitivity Analysis for Multiple Criteria Decision Making**, in Readings in Multiple Criteria Decision Aid, Carlos A Bana e Costa (ed.), 1990
6. Belton V, Gear T, **On a Shortcoming of Saaty's Method of Analytic Hierarchies**, Omega, Vol 11, No 3, 1982
7. Benayoun R, De Montgolfier J, Tergny J, Laritchev O, **Linear Programming and Multiple Objective Functions: STEP Method (STEM)**, Mathematical Programming, 1 (3), 1971
8. Benayoun R, Roy B, Sussman B, **ELECTRE: Une Methode Pour Guider le Choix en Presence de Points de Vue Multiple**, Sema, Direction Scientifique, Note de Travail No 49, Paris, June 1966
9. Brans J P, Mareschal B, Vincke Ph, **PROMETHEE: A new family of outranking methods in multicriteria analysis**, Operational Research '84, J P Brans (editor), Elsevier Science Publishers B.V. (North-Holland) IFORS 1984
10. Brans J P, Vincke Ph, Mareschal B, **How to Select and How to Rank Projects: The PROMETHEE Method**, European Journal of Operational Research 24, 1986
11. Brockhoff K, **Experimental Test of MCDM Algorithms in a Modular Approach**, European Journal of Operational Research 22, 1985
12. Buchanan P, **Solution Methods for Multiple Objective Decision Models**, European

Journal of Operational Research 29, 1987

13. Buchanan J T, Daellenbach H G, **A Comparative Evaluation of Interactive Solution Methods for Multiple Objective Decision Models**, European Journal of Operational Research 29, 1987
14. Bui T, **Building Effective Multiple Criteria Decision Models: A Decision Support System Approach**; Systems, Objectives, Solutions 4, 1984
15. Carlson E D, **An Approach for Designing Decision Support Systems**, in Building Decision Support Systems, Bennett J L (ed), Addison–Wesley Publishing Company, 1983
16. Chankong V, Haimes Y Y, **Multiobjective Decision Making: Theory and Methodology**, North–Holland, 1983
17. Charnes A, Cooper W, **Management Models and Industrial Applications of Linear Programming**, Vol 1, Wiley, New York, 1961
18. Churchman C W, Ackoff R L, Arnoff E L, **Introduction to Operations Research**, Wiley, New York, 1957
19. Courbon J C, Grajew J and Tolovi J, **Design and Implementation of Decision Support Systems by an Evolutive Approach**, Unpublished working paper, January 1979
20. Diakoulaki D, Koumoutsos N, **Cardinal Ranking of Alternative Actions: Extension of the PROMETHEE Method**, European Journal of Operational Research 53, 1991
21. Donovan J, **Database System Approach to Management Decision Support**, ACM Transactions on Database Systems, Vol 1, No 4, 1976
22. Dyer J S, **A Time–sharing Computer Program for the Solution of the Multiple Criteria Problem**, Management Science, Vol 19, No 12, August 1973
23. Dyer J S, **Remarks on the Analytic Hierarchy Process** (with responses by T L Saaty, P T Harker and L J Vargas), Management Science, Vol 36, March 1990
24. Edwards W, von Winterfeldt D, **Decision Analysis and Behavioral Research**, Cambridge University Press, 1986
25. Elam J J, Mead M, **Designing for Creativity: Considerations for DSS Development**,

Information & Management 13, 1987

26. Evren R, **Interactive Compromise Programming**, Journal of the Operations Research Society, Vol 38, No 2, 1987
27. Festinger L (ed), **Conflict Decision and Dissonance**, Stanford, California, 1984
28. Fischer M L, **Interactive Optimization**, Annals of Operations Research, Vol 5, 1985/6
29. Fischhoff B, **Debiasing**, In **Judgement Under Uncertainty: Heuristics and Biases**, Kahneman D, Slovic P and Tversky A (eds), Cambridge Univ Press, 1982
30. Floyd S A, Turner III C F, Davis K R, **Model-Based Decision Support Systems: An Effective Implementation Framework**, Computers and Operations Research, Vol 16, No 5, 1989
31. Geoffrion A M, Dyer J S, Feinberg A, **An Interactive Approach for Multi-Criterion Optimization, with an Application to the Operation of an Academic Department**, Management Science, Vol 19, December 1972
32. Gerrity T P, **The Design of Man-Machine Decision Systems: An Application to Portfolio Management**, Sloan Management Review, Vol 12, No 2, 1971
33. Gershon M, Duckstein L, **An Algorithm for Choosing a Multiobjective Technique**, Multiple Criteria Decision making (5th International Conference, Mons, Belgium, 1982), Springer-Verlag series: Lecture notes in Economics and Mathematical Systems
34. Gibson M, Bernnardo J J, Chung C and Badinelli R, **A comparison of interactive multiple objective decision making procedures**, Computers and Operations Research, Vol 14, No 2, 1987
35. Goicoechea A, Hansen D R, Duckstein L, **Multiobjective Decision Analysis with Engineering and Business Applications**, John Wiley and Sons, 1982
36. Gorry G A, Scott-Morton M S, **A Framework for Management Information Systems**, Sloan Management Review, 12, No 1, Fall 1971
37. Harmon P, King D, **Expert Systems: Artificial Intelligence in Business**, John Wiley and Sons, New York, 1985

38. Hughes C T, Gibson M L, **An Attribution Model of Decision Support Systems (DSS) Usage**, *Information and Management* 13, 1987
39. Hwang C-L, Yoon K, **TOPSIS (Technique for Order Preference by Similarity to Ideal Solution – A Multiple Attribute Decision Making)**, 1980
40. Hwang C-L, Yoon K, **Multiple Attribute decision Making: Methods and Applications**, Springer-Verlag series: Lecture Notes in Economics and Mathematical Systems, 1981
41. Jacob V S, Moore J C, Whinston A B, **An Analysis of Human and Computer Decision-Making Capabilities**, *Information and Management* 16, 1989
42. Jelassi M T, Jarke M, Stohr E A, **Designing a Generalized Multiple Criteria Decision Support System**, from *Decision Making with Multiple Objectives* (Proceedings, Cleveland, Ohio 1984), Haimes Y Y and Chankong V (eds), Springer Lecture Notes in Economics and Mathematical Systems, No 242
43. Jelassi M T, Williams K, Fidler C S, **The Emerging Role of DSS: From Passive to Active**, *Decision Support Systems*, Vol 3, 1987
44. Johnson E, **Studies in Multiobjective Decision Models**, Lund, Sweden: Studentlitteratur, 1968
45. Khairullah Z Y, Zionts S, **An Approach for Preference Ranking of Alternatives**, *European Journal of Operational Research* 28, 1987
46. Keen P G W, **Adaptive Design for DSS**, *Data Base*, Vol 12, No 1 & 2, Fall 1980
47. Keen P G W, **Decision Support Systems: the next decade**, In *DSS: a decade in perspective*, McLean E & Sol H G (eds), North-Holland, Amsterdam, 1986
48. Keen P G W, Scott-Morton M S, **Decision Support Systems: An Organizational Perspective**, Reading, Massachusetts, Addison-Wesley, 1978
49. Keen P G W, Wagner G R, **DSS: An Executive Mind-Support System**, *Datamation*, 1979
50. Keeney R L, Raiffa H, **Decisions with Multiple Objectives: Preferences and Value Tradeoffs**, John Wiley and Sons, New York, 1976

51. Klein G, Moskowitz H, Ravindran A, **Comparative evaluation of prior versus progressive articulation of preference in bicriterion optimization**, Naval Research Logistics Quarterly, Vol 33, 1986
52. Koopmans T C, **Analysis of Production as an Efficient Combination of Activities**, Activity Analysis of Production and Allocation, Cowles Commission Monograph 13, Wiley, New York, 1951,
53. Korhonen P J, Laakso J, **A Visual Interactive Method for Solving the Multiple Criteria Problem**, European Journal of Operational Research, Vol 24, 1986
54. Korhonen P, Soismaa M, **An Interactive Multiple Criteria Approach to Ranking Alternatives**, Journal of the Operational Research Society, Vol 32, 1981
55. Korhonen P, Wallenius J, Zionts S, **Solving the Discrete Multiple Criteria Problem using Convex Cones**, Management Science, Vol 30, No 11, 1984
56. Kuhn, Tucker, **Nonlinear Programming**, in J Neyman (ed), Proceedings of the Second Berkeley Symposium on Mathematical Statistics and Probability, University of California Press, Berkeley, 1951
57. Kydd C T, **Cognitive Biases in the Use of Computer-Based Decision Support Systems**, OMEGA International Journal of Management Science, Vol 17, No 4, 1989
58. Levine P, Pomerol J-Ch, **PRIAM, an interactive program for choosing among multiple attribute alternative**, European Journal of Operational Research 25, 1986
59. Lofti V, Stewart T, Zionts S, **An aspiration-level interactive model for multiple criteria decision making**, Working Paper No 701, August 1988
60. MacCrimmon K R, **Decision Making among Multiple-Attribute Alternatives: A Survey and Consolidated Approach**, Santa Monica, California, RAND Memorandum, RM4823-ARPA, 1968
61. Maier N F F, **Psychology in Industrial Organizations**, Houghton-Mifflin, Boston, 1973
62. Malakooti B, **A Decision Support System and a Heuristic Interactive Approach for Solving Discrete Multiple Criteria Problems**, IEEE Transactions on Systems, Man and

Cybernetics, Vol 18, No 2, March/April 1988

63. March J G, **Bounded Rationality, Ambiguity and the Engineering of Choice**, Bell Journal of Economics 9, 1987
64. Minch R P, Sanders G L, **Computerized Information Systems Supporting Multicriteria Decision Making**, Decision Sciences, Vol 17, 1986
65. Morgenstern O, von Neumann J, **Theory of Games and Economic Behavior**, Second Edition, Princeton University Press, Princeton, 1947
66. Power D J, Aldag R J, **An Empirical Assessment of Computer-Assisted Decision Analysis**, Decision Sciences Vol 17, 1986
67. Reeves G R, Franz L S, **A simplified interactive multiple objective linear programming procedure**, Computers and Operations Research, Vol12, No 6, 1985
68. Reeves G R, Gonzalez J J, **A Comparison of two Interactive MCDM Procedures**, European Journal of Operational Research 41, 1989
69. Rothermel M A, Schilling D A, **A comparative study of three methods of eliciting preference information**, OMEGA International Journal of Management Science, Vol 12, No 4, 1984
70. Roy B, **Problems and Methods with Multiple Objective Functions**, Mathematical Programming, Vol 1, No 2, 1971
71. Roy B, **From Optimization to Multicriteria Decision Aid: Three Main Operational Attitudes**, in Multiple Criteria decision Making: Jouy-en-Josas, France, Thiriez H and Zionts S (eds.), Springer-Verlag: Berlin 1976
72. Roy B, **Meaning and Validity of Interactive Procedures as Tools for Decision Making**, European Journal of Operational Research 31, 1987
73. Roy B, **The Outranking Approach and The Foundations of ELECTRE Methods**, Document du Lamsade, Laboratoire d'Analyse et modelisation de Systemes pour l'Aide a la Decision, Universite de Paris-Dauphine, Document No 53, July 1989
74. Saaty T, **The Analytic Hierarchy Process**, New York, McGraw-Hill, 1980

75. Simon H A, **The New Science of Management Decisions**, New York: Harper and Row, 1960
76. Sol H G, **Conflicting Experiences with DSS**, Decision Support Systems 3, 1987
77. Sprague R H, **A Framework for the Development of Decision Support Systems**, in Decision Support Systems: Putting Theory into Practice, Sprague R.H. and Watson H.J. (eds.), Prentice–Hall, Englewood Cliffs, New Jersey, 1986
78. Sprague R H, Watson H J (ed), **Decision Support Systems: Putting theory into practice**, 1986
79. Spronk J, **Interactive Multiple Goal Programming: Applications in Financial Planning**, International Series in Management Science/Operations Research, Ignizio J P (ed.), 1981
80. Steuer R E, Choo E–U, **An Interactive Weighted Tchebycheff Procedure for Multiple Objective Programming**, Mathematical Programming, Vol 26, No 1, 1983
81. Steuer R E, **Multiple Criteria Optimization: Theory, Computation, and Application**, John Wiley & Sons, 1986
82. Stewart T J, **Inferring Preferences in Multiple Criteria Decision Analysis using a Logistic Regression Model**, Management Science, Vol 30, No 9, September 1984
83. Suh E–H, Hinomoto H, **Use of a Dialogbase for Integrated 'Relational' Decision Support Systems**, Decision Support Systems, Vol 5, 1989
84. Vazsonyi A, **Decision Support Systems: The new Technology of decision making?**, Interfaces, Vol9, No 1, 1978
85. Wallenius J, **Comparative Evaluation of some Interactive Approaches to Multicriterion Optimization**, Management Science, Vol 21, No 12, August 1975
86. Weistroffer H R, **Multiple Criteria Decision Making with Interactive over–achievement Programming**, Operations Research Letters, Vol 1, No 6, December 1982
87. Wierzbicki A P, **The use of Reference Objectives in Multiobjective Optimization**, Working Paper 78–66, International Institute for Applied Systems Analysis, Laxenburg,

Austria, 1979

88. Wynne B E, **A Domination Sequence – MS/OR: DSS and the Fifth Generation**, Interfaces, Vol 14, No 3, 1984
89. Zeleny M, **Multiple Criteria Decision Making**, McGraw–Hill Book Company, 1982
90. Zeleny M, **A Concept of Compromise Solutions and the Method of the Displaced Ideal**, Computers and Operations Research, Vol 1, 1974
91. Zionts S, **A Multiple Criteria Method for choosing among Discrete Alternatives**, European Journal of Operational Research 7, 1981
92. Zionts S, **Multiple Criteria Mathematical Programming: an Updated Overview and Several Approaches**, State University of New York, June 1987
93. Zionts S, Wallenius J, **An Interactive Multiple Objective Linear Programming Method for a Class of Underlying Nonlinear Utility Functions**, Management Science, Vol 29, No 5, 1983

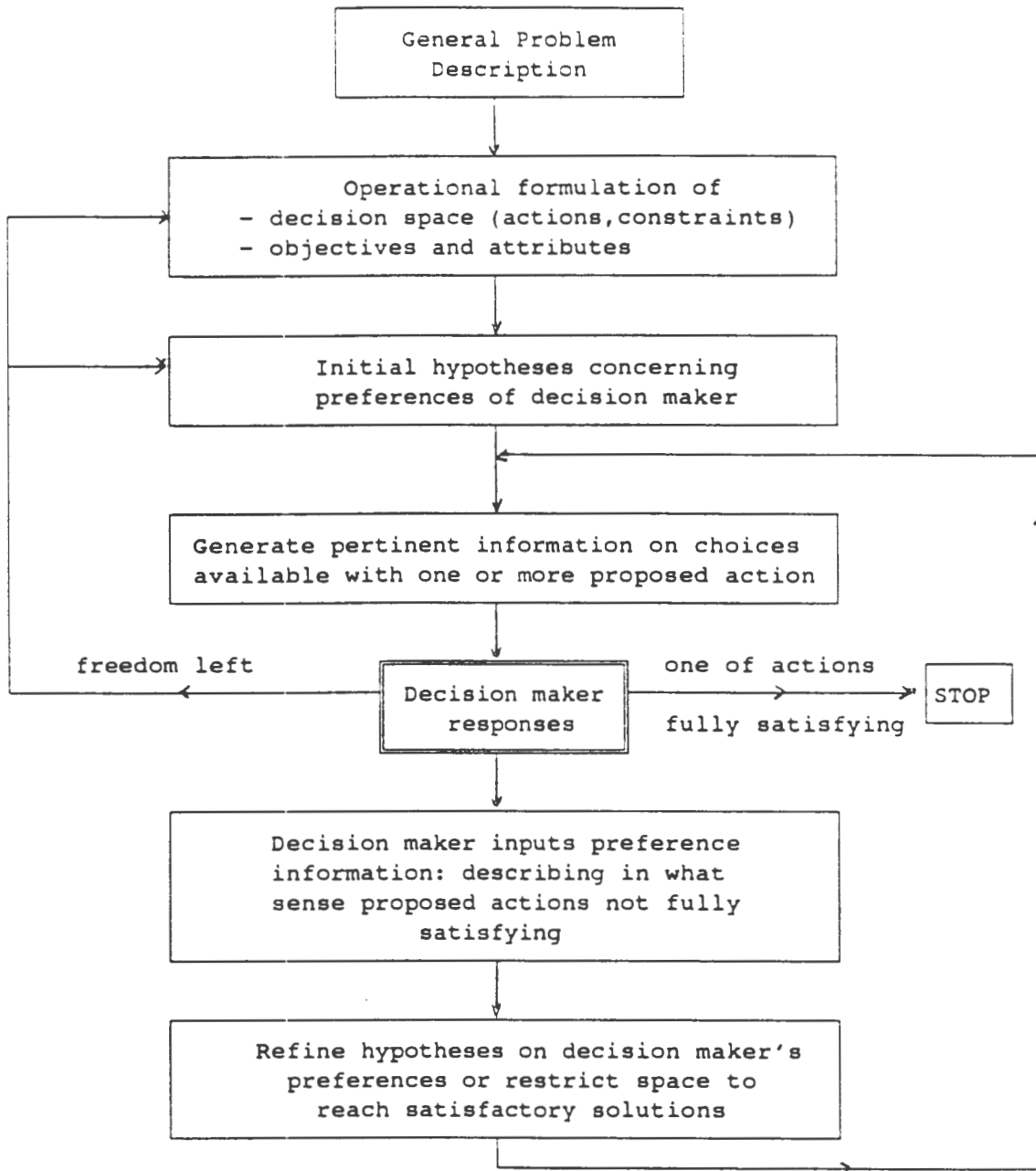


Figure 1: A Graphical Display of the Decision Process

Delete all dominated alternatives

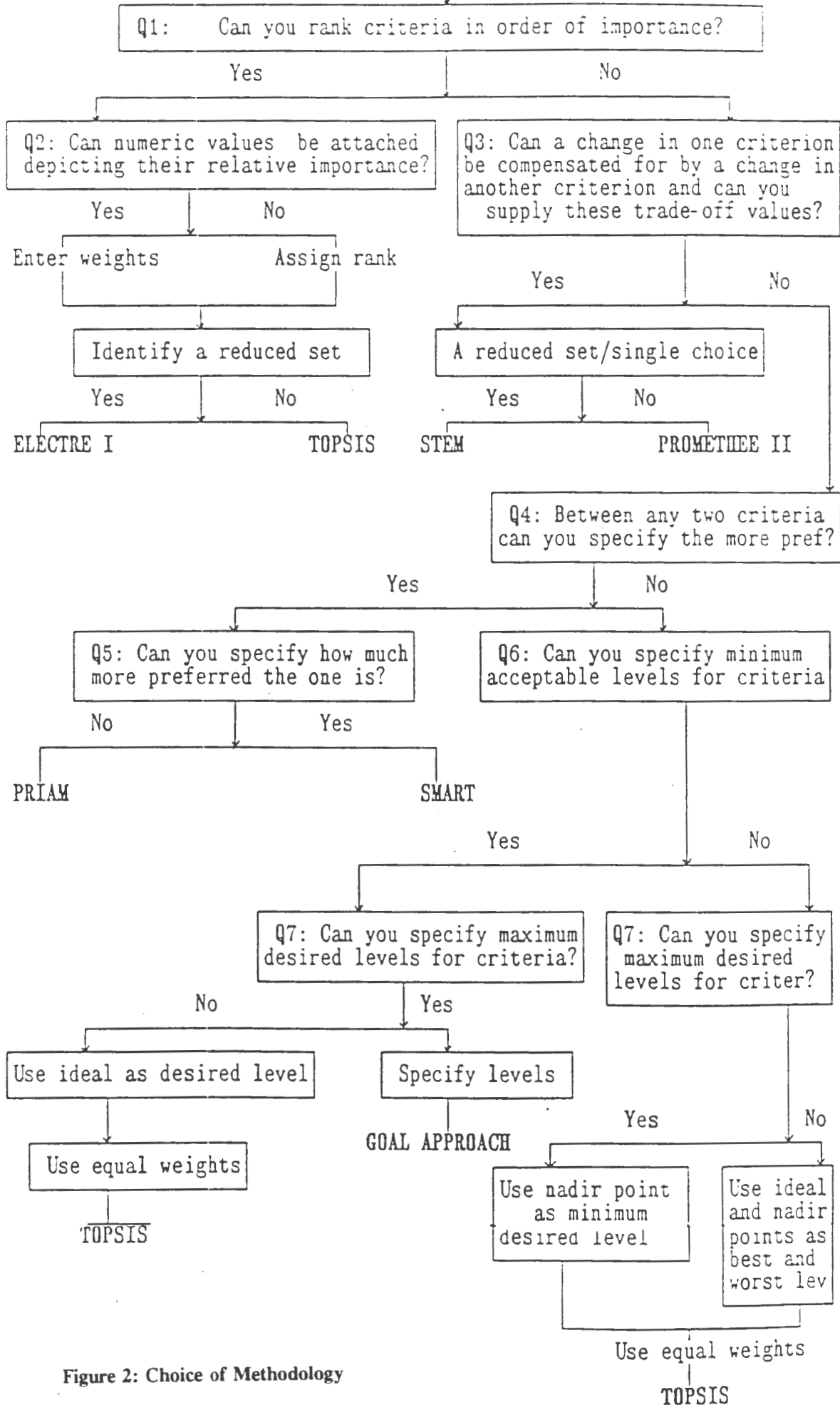


Figure 2: Choice of Methodology

A MULTICRITERIA DECISION SUPPORT SYSTEM
for solving discrete multicriteria
decision making problems

Developed by T van Dyk

Copyright (1992) Version 1

Press any key to continue

Figure 6.1

Set-Up	Alternatives	Criteria	Combined File	Solve	Output	Exit
--------	--------------	----------	---------------	-------	--------	------

F1 - Help Esc - Cancel

Figure 6.2

Current Directory
New Directory

Figure 6.3

Enter directory (e.g. c:\MCDM):

Figure 6.4

No directory specified. Do Set-Up

Figure 6.5

Create File
Modify File
View File
Delete File

Figure 6.6

Name of alternative file:

Figure 6.7

File already exists
Overwrite Cancel

Figure 6.8

Description of alternative:

Figure 6.9

Name
CAR1
CAR2
DATA1
FILE1
FILE2
.....
.....
.....

Figure 6.10

Add a new alternative
Delete an existing alternative

Figure 6.11

Alternative to be added:

Figure 6.12

Alternative to be deleted:

Figure 6.13

Alternative by that name does not exist.

Cancel

Try again

Figure 6.14

Alternative
MERCEDES
BMW
FORD
VW
.....

Figure 6.15

Description of criterion:
Minimize or Maximize? (MIN/MAX):

Figure 6.16

Name of file:

Figure 6.17

Alternative File Criteria File

Figure 6.18

Alternative: A1

Criteria	Value
C1 (MIN)
C2 (MAX)
C3 (MAX)
C4 (MIN)
.....

Figure 6.19

CAR1

Alternativ	Criteria	Min/Max	Value
MERCEDES	SPEED	MAX	200
MERCEDES	COMFORT	MAX	6
MERCEDES	FUEL CONSU	MIN	18
MERCEDES	PRICE	MIN	80000
BMW	SPEED	MAX	220
BMW	COMFORT	MAX	5
BMW	FUEL CONSU	MIN	18
BMW	PRICE	MIN	90000
FORD	SPEED	MAX	180
.....
.....

Press Esc when done viewing

Figure 6.20

New user
Experienced user

Figure 6.21

Single choice
Rank all alternatives
Reduced set containing best
Rank a subset of alternatives

Figure 6.22

Can you rank the criteria in order of importance? (Y/N):

Figure 6.23

Can numeric values be attached depicting their relative importance? (Y/N):

Figure 6.24

Can the change in one criterion be compensated for by a change in another criterion and can you supply these trade-offs? (Y/N):

Figure 6.25

Between any two criteria and/or alternatives can you specify the more preferred? (Y/N):

Figure 6.26

Can you specify minimum acceptable levels for all criteria? (Y/N):

Figure 6.27

Can you specify maximum desired levels for all criteria? (Y/N):

Figure 6.28

Recommended methodology

TOPSIS

Enter to execute or Esc to cancel

Figure 6.29

Rank/Weight criteria
Do trade-offs
Specify goal levels
Do paired comparisons
Strength of preference

Figure 6.30

Weights
Rank criteria

Figure 6.31

Enter rank numbers for criteria

Criteria	Rank
SPEED	...
COMFORT	...
FUEL CONSUMP	...
PRICE	...

Press Esc when done editing

Figure 6.32

Supply weights

Criteria	Weights
SPEED	...
COMFORT	...
FUEL CONSUMP	...
PRICE	...

Press Esc when done editing

Figure 6.33

Please be patient - problem is being solved

Figure 6.34

CAR1.TOP

Alternative	Score
FORD	0.876
OPEL	0.785
VW	0.774

Are you satisfied? (Y/N):

Figure 6.35

Criterion: SPEED

Considering evaluations of alternatives on this criterion, can you specify a maximum difference between evaluations below which any two alternatives can be considered indifferent? (Y/N):

Specify maximum difference value:

Figure 6.36

Criterion: COMFORT

Considering evaluations of alternatives on this criterion, can you specify a minimum difference between evaluations above which any one alternative can be considered to be preferred to any other alternative? (Y/N):

Specify minimum difference value:

Figure 6.37

Criterion: PRICE
Does your preference for one alternative above another increase linearly with the difference value between ... and ... ? (Y/N):

Figure 6.38

Criterion: FUEL CONSUMPTION
Does your preference for one alternative above another increase non-linearly with the difference value between ... and ...? (Y/N):
Specify the difference value where your preference starts flattening:

Figure 6.39

Your Ideal and Nadir alternatives are:

Criteria	Ideal	Nadir
SPEED	220	120
COMFORT	6	2
FUEL CONSUM	12	20
PRICE	40000	90000

Press Esc to return

Figure 6.40

You are at alternative: A

Criteria	Value
SPEED	140
COMFORT	4
FUEL CONSUM	12
PRICE	25000

Press Esc to return

Figure 6.41

Enter relaxations next to the relevant criteria
To decrease a value precede it with a negative sign

Criteria	Value	Change
SPEED	140	...
COMFORT	3	...
FUEL CONSUM	15	...
PRICE	40000	...

Press Esc when done editing

Figure 6.42

<p>No alternatives are feasible</p> <p>Return to menus (M) or start again (S)?:</p>

Figure 6.43

The following alternatives are feasible:

Alternativ	Criteria	Value
A	SPEED	110
A	PRICE	18000
A	COMFORT	3
A	FUEL	12
B	SPEED	112
B	PRICE	18500
B	COMFORT	3.2
B	FUEL	14

Press Esc to return

Figure 6.44

Please be patient - problem is being set up

Figure 6.45

Assuming you are at alternative with values equal to the Nadir enter an Y next to those criteria values you would like to change and the desired values if possible

Criteria	Ideal	Nadir	Change_Y_N	Desire_Val
SPEED	220	120
COMFORT	6	2
FUEL CONSUM	12	19
PRICE	40000	90000

Press Esc when done editing

Figure 6.46

Feas_Max indicates your ideal (infeasible) alternative;
 Feas_Min indicates your worst (infeasible) alternative;
 Desire_Val indicates your current goal alternative and
 Value indicates your current best alternative

Criteria	Feas_Max	Feas_Min	Desire_Val	Value
SPEED	220	120	180	150
COMFORT	6	2	5	4
FUEL CONSUM	12	19	14	13
PRICE	40000	90000	38000	43000

Number of remaining feasible alternatives: ...
 Accept feas_min as the new starting solution (Y/N)?: ...

Figure 6.47

Supply levels to start from or 0 if unsure

Criteria	Currnt_Min	Currnt_Max	Desire_Min
SPEED	2	5	...
COMFORT	1	6	...
FUEL CONSU	18	12	...
PRICE	90000	40000	...

Press Esc when done editing

Figure 6.48

You are at level:

Criteria	Desire_Min
Price	80000
Speed	1
Fuel consum	18
Comfort	2

Go (A)head or (B)ackwards? :

Figure 6.49

Enter your modification for each criterion.
 For decreasing a value enter a - before the value.

Criteria	Desire_Min	Change
Price	80000	...
Speed	120	...
Comfort	1	...
Fuel Consum	18	...

Is this change (I)mperative or (T)entative?:

Figure 6.50

Remaining feasible alternatives

Alternative	Criteria	Value
FORD	SPEED	2
FORD	COMFORT	2
FORD	FUEL CONSUM	12
FORD	PRICE	45000
BMW	SPEED	5
BMW	COMFORT	6
BMW	FUEL CONSUM	18
BMW	PRICE	90000

Enter the preferred one:

Figure 6.51

Compare the following two alternatives

Alternative	Alternative
FORD	OPEL
FORD	OPEL
FORD	OPEL
FORD	OPEL

Enter the preferred one:

Figure 6.52

You have gone back to your starting point
Display results? (Y/N):

Figure 6.53

Results

Your best alternative is:
You have encountered the following Pareto optimal alternatives:
FORD
OPEL

Are you satisfied? (Y/N):

Figure 6.54

Criterion: Price

Min

Alternative	Value	Rate
BMW	150000	0
MERCEDES	120000	...
FORD	45000	...
VW	43000	...
OPEL	32000	100

Enter the strength of preference rate between alternatives
100 = best and 0 = worst
Press Esc when done

Figure 6.55

Enter the ratio of importance between criteria
10 = minimum (lowest rank)

Criteria	Rate
PRICE	10
FUEL	...
COMFORT	...
SPEED	0

Press Esc when done editing

Figure 6.56

View file
Print file
Delete file

Figure 6.57

Name
CAR.TOP
CAR.PR1
CAR.SMT

Figure 6.58

Alternative	Score
FORD	0.867
OPEL	0.785
VW	0.774
MERCEDES	0.423
BMW	0.358

Figure 6.59

Printer ready?	
Print	Cancel

Figure 6.60

APPENDIX I – AN EXAMPLE

In this Appendix an example problem is solved using this multiple criteria decision support system. The problem is defined and the user is then taken through the decision process step by step using this system.

A decision maker has to buy a new car. The following is a list of possible cars from which 'best' one has to be chosen: Volkswagen, Mercedes, BMW, Ford, Opel. The criteria that will influence the choice of what car to buy, can be listed as follows:

price, fuel consumption, speed, comfort. The first two of these criteria have to be minimised, while the other two have to be maximised. The criteria are conflicting in nature, as a decrease in price will usually mean a decrease in comfort (say) – one criterion (price) is thus improved, while another one (comfort) is decreased. This problem is, therefore, clearly a discrete multicriteria decision making problem which can be solved with the aid of this system.

The decision process will be discussed in logical steps:

1. The decision maker will first have to decide where the files to be created should be stored. This is done by the 'Set-up' option on the main menu (Figure 6.2).
2. The list of possible choices of cars will now have to be entered into a file. This is achieved by the 'Alternative' option on the main menu (Figure 6.2). Assuming that this problem has not been addressed before, the user has to create the file and therefore needs to enter a filename under which the information is to be stored. This is done by option 1 on Figure 6.6. The filename, ALTCAR (say), is entered and a screen similar to Figure 6.9 is displayed. The five different makes of cars are then entered consecutively. After this has been done, the user presses the Return key (accepting a blank description) and is returned to Figure 6.6. The alternative file, ALTCAR, has been created.

3. The criteria that will be used for choosing the 'best' car, also have to be stored in a separate file. This is done by choosing the third option ('criteria') on the main menu (see Figure 6.2). Assuming that the problem has not been addressed before, the criteria file will have to be created by choosing the first option on Figure 6.6. The name of the file is again entered, CRITCAR (say) and will be saved in the specified directory. As soon as this has been done, the descriptive names of the criteria as well as whether they should be minimised or maximised are entered on the screen displayed in Figure 6.16. By pressing the Return key without entering a description, the user is returned to Figure 6.6.

4. The two separate files, ALTCAR and CRITCAR, have to be combined into a single file in which values will be attached to the alternatives for each of the different criteria. This is achieved by the fourth option ('combined file') on the main menu (see Figure 6.2). The user has to specify a name for the file to be created ('CARS' say) as in Figure 6.17. A screen similar to Figure 6.18 is then displayed and both options have to be executed. In both instances, a list of existing files is displayed and the ALTCAR file has to be chosen from the list of alternative files, while CRITCAR is chosen from the list of criteria files. A screen similar to Figure 6.19 is then displayed for each make of car. The values of each specific car displayed, for the different criteria, have to be entered (for example: for Volkswagen – price = 40000; fuel consumption = 12 l/100km; speed = 2 (on a scale of 1 to 5) and comfort = 2.5 (on a scale of 1 to 5)) Once this has been done for all the alternatives, the problem can be solved.

5. The user chooses option 5 from the main menu and has to specify whether he is a new or experienced user by choosing the relevant option from the menu displayed in Figure 6.21. A list of all the decision problems that have been defined, is displayed and the 'CARS' file has to be chosen. The type of output desired by the user has then to be chosen from a menu as displayed in Figure 6.22. Assuming that the user has never used

the system before ('New user'), the user is prompted by a series of questions in order to determine the type of preference information the user is able to provide. This series of questions is listed from Figure 6.23 up to Figure 6.28. These questions are closely related to the menu that is presented to an experienced user of the system as displayed in Figure 6.30. The specific information that can be supplied by the decision maker will have to be entered on a screen specifically designed for that. Assuming the user is able to rank the different criteria of the CARS decision problem, the different rank numbers will have to be supplied on a screen similar to Figure 6.32. The most important criterion has a rank number of 1, the next most important a rank number of 2 and so on until a rank number has been attached to all the criteria. The TOPSIS methodology will be suggested by the system as the most suitable methodology to be used (see Figure 6.29). The output of the methodology is displayed on a screen similar to Figure 6.35 and the user gets the opportunity to enter different rank numbers than before by entering an 'N' to the prompted question. Once he is satisfied, he is returned to the previous menu by pressing the Esc key.

6. The CARS decision problem can be solved using anyone of the methodologies implemented as long as the user is able to supply the relevant preference information needed for the specific methodology. An output file is created each time the problem is solved, where the extension of the file depicts the specific methodology that has been used for producing that specific output file. In our case this file will be called 'CARS.TOP' and the file can be viewed, printed or deleted by choosing the sixth option on the main menu (see Figure 6.2). An example of such an output file is displayed in Figure 6.59 where the score displayed depicts the order of importance of the alternatives.

7. The user is returned to the disk operating system (DOS) by choosing the seventh option from the main menu (see Figure 6.2).

APPENDIX II – HELP

The help system that has been implemented and that is accessible by the user on pressing the F1 key is briefly discussed in this appendix. Use is made of illustrations allowing the reader to follow the text descriptions more easily.

Help is available at almost all the stages of the decision process. An example of a help screen, is given below:

Help Screen 2
The user can specify where the files to be constructed should be saved or, alternatively, from which directory to retrieve already existing files. The format in which to enter a new directory is C:\NEW, meaning the files will be resident in the directory 'NEW' on the C drive.

Press any key to return to previous screen

All the other help screens have a similar format to the one displayed above. If the F1 key is pressed at any stage of the decision process for which help has not been implemented, the command will be ignored and will have no effect on the system.

APPENDIX III – PROBLEMS USED FOR EVALUATING THE SYSTEM

(i) Problem 1 - Choosing a boarding location

The problem consists of 31 alternative locations, each one evaluated on 4 different criteria. The alternatives have been numbered from 1 to 31. The following criteria are used:

- | | |
|---|-----------------|
| (i) number of bathrooms | to be maximised |
| (ii) age of the house (yrs) | to be minimised |
| (iii) rent to be paid (p/m) | to be minimised |
| (iv) distance to university (scale from 0 to 7) | to be maximised |

The evaluations for the different alternatives are as follows:

Alternative	#Bathrooms	Age	Rent	Distance
A	4	48	900	1.5
B	2	22	380	2.8
C	2	25	382	3.5
D	1	45	290	0.2
E	1	16	300	1.4
F	1	34	377	2.1
G	2	12	382	0.3
H	5	40	650	3.2
I	2	45	340	6.3
J	1	30	370	1.7
K	2	20	590	1.3
L	2	22	455	3.5
M	2	14	650	3.2
N	2	17	500	1.4
O	3	9	530	0.4
P	2	20	455	3.4
Q	5	60	800	4.6
R	4	7	745	2.7
S	3	11	645	6.3
T	2	15	500	1.3
U	3	3	860	3.9
V	3	18	650	0.3
W	1	16	580	4.1
X	2	4	655	0.7
Y	2	28	295	2.1
Z	2	27	450	1.2
AA	3	15	660	5.3
BB	2	14	350	0.3
CC	2	32	520	1.4
DD	1	35	375	0.5
EE	2	7	510	4.9

(ii) Problem 2 - choosing a car to buy

The problem consists of 42 alternative cars that can be bought, each one evaluated on 7 criteria. The cars are numbered from 1 to 42. The following criteria are considered:

- | | |
|--|-----------------|
| (i) Price (R) | to be minimised |
| (ii) Max speed (km/h) | to be maximised |
| (iii) Fuel consumption (km/l) | to be maximised |
| (iv) Appearance (scale of 1 to 10) | to be maximised |
| (v) Mileage (km) | to be minimised |
| (vi) Maintenance needed (scale of 1 to 10) | to be minimised |
| (vii) Re-sale value (scale of 1 to 10) | to be maximised |

The evaluations for the different alternatives are as follows:

Altern	Price	Speed	Fuel	Appear	Mileage	Maint
A	25000	130	19	5	85	2
B	25000	92	15	2	100	6
C	22000	92	15	4	85	5
D	22000	105	11	3	90	4
E	23000	92	15.5	5	52	4
F	25000	105	15	6	60	3
G	23000	117	15	3	117.5	7
H	23000	117	16	3	90	5
I	24000	117	15	3	90	6
J	17000	117	15	3	90	6
K	19000	130	11	4	92	7
L	18000	117	15.5	3	117.5	5
M	20000	117	15.5	3	90	5
N	22000	105	15.5	4	117.5	4
O	22000	92	18	3	90	4
P	21000	92	18	3	90	6
Q	17000	105	11	3	117.5	4
R	24000	105	15.5	3	90	4
S	25000	92	15.5	3	90	4
T	17000	117	15.5	3	90	4
U	21000	117	13	4	85	3
V	25000	117.5	15.5	4	94	3
W	20000	105	15.5	3	94	2
X	19000	130	13	5	94	4
Y	21000	105	11	2	117.5	5
Z	20000	105	15.5	2	85	4
AA	13000	160	14	5	90	6
BB	12000	117	15.5	3	90	5
CC	25000	117	16	4	90	4
DD	20000	105	15.5	3	90	3
EE	10000	92	8	5	60	6
FF	9000	160	7	4	145	5
GG	9000	160	15.5	6	117.5	7

Altern	Price	Speed	Fuel	Appear	Mileage	Maint
HH	8500	130	12	3	85	5
II	7500	130	15	4	85	6
JJ	5000	142	15	5	90	7
KK	6500	155	7.5	6	85	7
LL	6000	142	5	4	117.5	6
MM	7000	130	5	4	110	5
NN	7500	117	8	6	85	6
OO	9500	105	11	3	60	4
PP	9500	105	11.5	4	140	4

Alternative	Re-Sale value
-------------	---------------

A	10
B	7
C	7
D	8
E	7
F	7
G	8
H	7
I	7
J	7
K	9
L	8
M	8
N	8
O	8
P	9
Q	7
R	8
S	8
T	8
U	9
V	8
W	9
X	8
Y	9
Z	8
AA	9
BB	7
CC	8
DD	7
EE	8
FF	8
GG	7
HH	7
II	8
JJ	8
KK	8
LL	9
MM	7
NN	7
OO	8
PP	7

APPENDIX IV – EVALUATION QUESTIONNAIRE

1. Ease of use

(i) How user–friendly did you find the interface structure ?

1 2 3 4 5 6 7 8 9 10

(1 = totally unfriendly, 10 = very user–friendly)

(ii) How helpful did you find the help facility implemented ?

1 2 3 4 5 6 7 8 9 10

(1 = totally useless, 10 = very helpful)

(iii) How easy did you find it to get to a stage where a methodology could be executed ?

1 2 3 4 5 6 7 8 9 10

(1 = very difficult, 10 = very easy)

(iv) How would you rank your level of computer literacy ?

1 2 3 4 5 6 7 8 9 10

(1 = no computer literacy, 10 = extensive literacy)

(v) How error–tolerant was the system ?

1 2 3 4 5 6 7 8 9 10

(1 = no tolerance, 10 = very tolerant)

2. Input information needed

(i) How easily could you relate to at least one of the possible categories for input information needed ?

1 2 3 4 5 6 7 8 9 10

(1 = impossible, 10 = very easy)

3. Output provided

(i) How close did the categories of output provided relate to what was needed ?

1 2 3 4 5 6 7 8 9 10

(1 = no relation, 10 = an exact relation)

(ii) How easy was it to interpret the output provided ?

1 2 3 4 5 6 7 8 9 10

(1 = very difficult, 10 = very easy)

4. Graphical representations

(i) How helpful were the graphical representations used in enhancing your understanding of the methodologies implemented ?

1 2 3 4 5 6 7 8 9 10

(1 = no help, 10 = very useful)

5. The methodologies implemented

(i) How compatible were the methodologies implemented to your specific needs and abilities ?

1 2 3 4 5 6 7 8 9 10

(1 = not compatible at all, 10 = very compatible)

(ii) How would you rate your confidence in trying different approaches for solving the same decision problem ?

1 2 3 4 5 6 7 8 9 10

(1 = no confidence, 10 = very confident)

(iii) How flexible was the system in allowing you to change your mind during the

solution process ?

1 2 3 4 5 6 7 8 9 10

(1 = no flexibility, 10 = very flexible)

(iv) How easy did you find it to understand the method ?

1 2 3 4 5 6 7 8 9 10

(1 = very difficult, 10 = very easy)

6. The time taken

(i) How quickly did you reach a satisfactory solution ?

1 2 3 4 5 6 7 8 9 10

(1 = very slowly, 10 = very quickly)

(ii) How would you rate the level of interest you had during the solution process ?

1 2 3 4 5 6 7 8 9 10

(1 = lost interest entirely, 10 = very high level of interest)

7. The confidence in the solution

(i) How satisfied were you with the solution presented to you ?

1 2 3 4 5 6 7 8 9 10

(1 = not satisfied, 10 = completely satisfied)

(ii) How would you rate the level of control you had in determining the final solution ?

1 2 3 4 5 6 7 8 9 10

(1 = no control, 10 = entire control)

(iii) Would you use the system to solve other discrete decision making problems ?

1 2 3

(1 = no, 2 = unsure, 3 = yes)

8. General

How would you rate the importance of the following criteria in judging the performance of a decision support system such as the one used ? (1 = unimportant, 10 = extremely important)

(i) Speed

1 2 3 4 5 6 7 8 9 10

-(ii) Quality of solution

1 2 3 4 5 6 7 8 9 10

-(iii) Easy of use

1 2 3 4 5 6 7 8 9 10

-(iv) Use of graphical representations

1 2 3 4 5 6 7 8 9 10

(v) The type of input demanded

1 2 3 4 5 6 7 8 9 10

(vi) The format of the output provided

1 2 3 4 5 6 7 8 9 10

(vii) The confidence in the solution reached

1 2 3 4 5 6 7 8 9 10

(viii) The methods implemented

1 2 3 4 5 6 7 8 9 10

APPENDIX V - STATISTICAL RESULTS

Analysis of Variance for matrix of performances

Source of var	Sum of Squares	d.f.	Mean Square	F-Ratio	Sig. level
MAIN EFFECTS	40.856564	15	2.7237710	2.245	0.0155
Respondents	14.040556	9	1.5600617	1.286	0.2662
Categories	26.816009	6	4.4693348	3.683	0.0039
RESIDUAL	65.529734	54	1.2135136		
TOTAL (CORR.)	106.38630	69			

0 missing values have been excluded.

Table of means for matrix of performances

Level	Count	Average	Std. Error (internal)	Std. Error (pooled s)	95% Conf for mean
Respondents					
1	7	5.4171429	0.3068016	0.4163641	4.5821928 6.252092
2	7	5.5600000	0.3076872	0.4163641	4.7250500 6.394950
3	7	5.2142857	0.2640604	0.4163641	4.3793357 6.049236
4	7	5.6428571	0.5084323	0.4163641	4.8079071 6.477807
5	7	4.9642857	0.8388423	0.4163641	4.1293357 5.799236
6	7	6.3500000	0.5985101	0.4163641	5.5150500 7.184950
7	7	5.8814286	0.5652746	0.4163641	5.0464786 6.716379
8	7	6.2000000	0.4498677	0.4163641	5.3650500 7.034950
9	7	5.9285714	0.2389034	0.4163641	5.0936214 6.763521
10	7	5.0428571	0.2389034	0.4163641	4.2079071 5.877807
Categories					
1	10	5.5000000	0.3574602	0.3483552	4.8014307 6.198569
2	10	5.6050000	0.1965607	0.3483552	4.9064307 6.303569
3	10	4.8000000	0.5734884	0.3483552	4.1014307 5.498569
4	10	6.2000000	0.3265986	0.3483552	5.5014307 6.898569
5	10	6.7400000	0.4759085	0.3483552	6.0414307 7.438569
6	10	5.5000000	0.1666667	0.3483552	4.8014307 6.198569
7	10	4.9960000	0.1669677	0.3483552	4.2974307 5.694569
Total	70	5.6201429	0.1316659	0.1316659	5.3561085 5.884177

Analysis of Variance for matrix of importances

Source of var	Sum of Squares	d.f.	Mean Square	F-Ratio	Sig. level
MAIN EFFECTS	128.02857	15	8.535238	8.450	0.0000
Respondents	40.85714	9	4.539683	4.494	0.0002
Categories	87.17143	6	14.528571	14.384	0.0000
RESIDUAL	54.542857	54	1.0100529		
TOTAL (CORR.)	182.57143	69			

0 missing values have been excluded.

Table of means for matrix of importances

Level	Count	Average	Std. Error (internal)	Std. Error (pooled s)	95% Conf for mean	
Respondents						
1	7	6.0000000	0.7867958	0.3798595	5.2382539	6.761746
2	7	6.0000000	0.8728716	0.3798595	5.2382539	6.761746
3	7	6.7142857	0.4205600	0.3798595	5.9525397	7.476032
4	7	6.4285714	0.4809288	0.3798595	5.6668254	7.190318
5	7	7.0000000	0.3086067	0.3798595	6.2382539	7.761746
6	7	7.0000000	0.6172134	0.3798595	6.2382539	7.761746
7	7	6.2857143	0.6441785	0.3798595	5.5239682	7.047460
8	7	8.0000000	0.3779645	0.3798595	7.2382539	8.761746
9	7	8.4285714	0.3688556	0.3798595	7.6668254	9.190318
10	7	6.7142857	0.6441785	0.3798595	5.9525397	7.476032
Categories						
1	10	5.3000000	0.5783117	0.3178133	4.6626775	5.937323
2	10	6.7000000	0.4229526	0.3178133	6.0626775	7.337323
3	10	5.8000000	0.2905933	0.3178133	5.1626775	6.437323
4	10	6.8000000	0.3265986	0.3178133	6.1626775	7.437323
5	10	7.6000000	0.2211083	0.3178133	6.9626775	8.237323
6	10	9.0000000	0.3333333	0.3178133	8.3626775	9.637323
7	10	6.8000000	0.4422166	0.3178133	6.1626775	7.437323
Total	70	6.8571429	0.1201221	0.1201221	6.6162576	7.098028