Preferences in Evolutionary Multiple Criteria Decision Making Optimisation

By

Alejandra Duenas

A thesis submitted for the candidature of the degree of Doctor of Philosophy

Department of Automatic Control and Systems Engineering

The University of Sheffield

Mappin Street

Sheffield, S1 3 JD

United Kingdom

January 2003

ABSTRACT

Despite the number of approaches established for Multiple Criteria Optimisation Problems, few of them have been developed for the decision making process. This research work proposes a new methodology for the solution of optimisation problems that involve multiple criteria emphasising the Decision-Maker's (DM's) preferences model and the use of evolutionary computation techniques and fuzzy logic. The use of genetic algorithms (GAs) is of vital importance to the development of this research. The use of operations research (OR) techniques and decision analysis is also considered vital. The aim of this project is to provide a definition of hybrid approaches that combine the strengths of GA and decision analysis. For this reason four hybrid models are proposed: 1. The GA-SEMOPS. 2. The fuzzy multiobjective genetic optimiser. 3. The GA-PROTRADE. 4. The interactive procedure for multiple objective optimisation problems. The main characteristics of these approaches are that they handle the DM's preferences in an interactive way and their objective functions are formulated using goal levels and surrogate functions.

In order to demonstrate that these models can be used in different optimisation problems they have been applied to different case studies covering examples from environmental systems to land and human resource allocation. Each model was studied in depth, comparing the results found with those available in literature. In the majority of the cases, it was found that they performed better than existing methods.

The investigations carried out showed that the proposed hybrid models can be considered as a very powerful tool for the solution of a wide variety of optimisation problems in situations from business to science and engineering.

ACKNOWLEDGEMENTS

First, I would like to thank my supervisor, Dr. Neil Mort, who has carefully read this manuscript and provided me with valuable comments and discussions regarding this research work. I am grateful for his supervision throughout this research programme.

I acknowledge support from Consejo Nacional de Ciencia y Tecnología (México) and Instituto Tecnológico y de Estudios Superiores de Monterrey- Campus Estado de México. I am also very grateful to Emilio Alvarado and Juan López, who gave me all their help at the beginning of this degree.

I would also like to thank my parents for their love and support. Special thanks go to my sisters Andrea and Anabel Dueñas, my brother in law Mario López and Natividad Ortíz for believing in me. I thank Daniela María López Dueñas for being my source of inspiration.

I would like to take this opportunity to extend my deepest gratitude to Joaquín Aguado for the hours we spent discussing different problems, his academic advice, his invaluable friendship and because we started this journey together. I also thank all my friends for their encouragement and comprehension especially Karen Koenig, Mariela Jiménez, Mónica Castro, Marina Monter, Laura Hidalgo, Daniela Coimbra, Rui Pedro Pereira and Filipa Lã.

Special thanks are due to Stephen McCabe for his support, his decisive help, for tolerating my ups and downs and for making my life easier and happier.

Finally, I thank God for being the main inspiration in my life and for giving me the courage and strength to finish this research programme. I dedicate this thesis to Him and to my parents Daniel and Tony Dueñas.

CONTENTS

CHAPTER 1

Introduction

1.1 Operational Research

The first scientific area where the study of the *design* and *decision processes* were formally defined is *operational research* (OR), as it is called in Britain, or o*perations research* or *management science* (OR/MS), as it is known in the USA. As the main objective of this thesis is the study of decision-making, the different definitions or views of different authors of OR will be discussed in this chapter.

Generally, it is considered that OR originated in the World War II period. Cook and Shutler (1991) presented a creditable review of the history of OR. They stated that although the idea of applying science to the solution of decision problems can be traced from the Greeks, it was not until the mid-1930s that a country's national objectives and science objectives became one. At that time, a group of scientists led by Robert Watson-Watt was working on the development of radar to defend the United Kingdom from German air attack. This group was interdisciplinary with scientists from different areas such as mathematics, physics, statistics, electronics and psychology and developed a new activity or a new way of visualising and attacking a problem. This team had an office in Operational Headquarters, according to Cook and Shutler (1991) the door where this group of scientist was working was labelled "operational research" that means "research into operations" or perhaps " research within operational headquarters". In 1940 Watson-Watt and his team decided to name the new activity "operational research" in order to describe the general activity and make it known.

Professor Patrick Blackett (physicist) who was an active member of the OR group contacted his fellow professor at the Massachusetts Institute of Technology in Boston, Philip Morse, to share with him how operational research was used in different areas of the war. Blackett and other colleagues presented operational research in a special conference, to scientists and defence chiefs of the United States. They adopted the idea but changed the name to "operations research", because of their use of "adjectival nouns"(Cook and Shutler, 1991), setting up OR groups in the different areas of the USA's armed forces (land, sea and air).

After the war some scientists were convinced that OR could be as useful in peace as it had been in war, for this reason, in this period of time the main focus of OR was on quantitative approaches to support decision-making. The simplex method proposed by George Dantzig in 1947, led to the application of OR to non-military problems. This combined with the development and growth of digital computers resulted in an extensive range of methods and applications. In 1957, Churchman, Ackoff and Arnoff published the first OR textbook, the main objective of their book was to visualise OR as a *process* rather than just a simple set of techniques to solve problems. They stated as mentioned above that the main objective of OR was to improve operations and for this reason defined a procedure for conducting OR shown in Figure 1.1, understanding that a project can be also called *research*. The phases of this procedure are explained in more detail below.

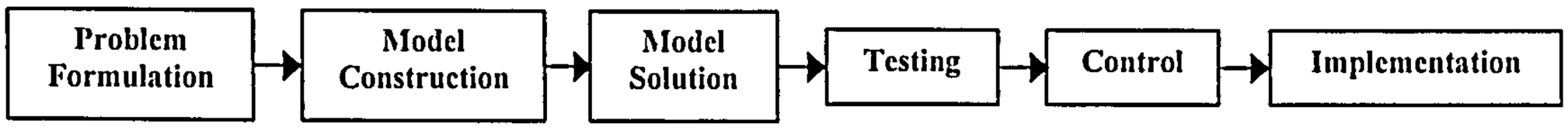


Figure 1.1 Phases of the Research

Problem formulation

Churchman et al. (1957) defined the formulation of a problem (first phase of the research), also called the orientation period, as a sequential process that starts with an initial formulation and is continuously reformulated until a solution is found. Generally speaking, a problem arises when an individual or a group within the organisation wants to improve some aspects of current practices and for this reason a decision has to be made. Churchman et al. (1957) presented an idealised problem-formulation procedure where the first step was to define the components of the problem.

The components of a problem are:

1. *Decision-maker* (DM). It is important to identify the person or group of people who is going to make the decision once the problem is solved.

2. *Objectives.* These outline the desires of the DM as well as the possible outcomes of the problem.

3. *System* or *environment*. The principal components of the system were listed by Churchman et al. (1957) as "management, men, machines, materials, consumers, competitors, and government and the public".

4. *Alternative courses of action*. The researchers have to list all the possible alternative courses of action.

Once the components of the problem are defined the next step is to transform the problem into a research problem. To perform the transformation the steps listed below are followed:

1. Editing the list of objectives.

2. Editing the list of alternative courses of action.

3. Defining the measure of effectiveness.

Construction of the model

A scientific model is understood as "a representation of the system under study"; it is also understood as an instrument that helps in the evaluation of the possible alternatives of action.

Churchman et al. (1957) classified models into three categories: iconic, analogue and symbolic.

- Iconic models. These models' main characteristics are that they "look like what they represent", their properties are the same as those of the original and usually are scaled up or down making the decision or design process easier to use and more economic.

- Analogues. It is not always possible to represent all the properties of the original system, when this is the case it is necessary to make a "substitution of one property for another according to some transformation rules" (Churchman et al., 1957). A good example of these kinds of models is graphs because they use distance to represent different properties such as time, weight, money, etc. It is easier to modify an analogue than an iconic model and also it is considered more general.

- Symbolic models. In these models, "the components and their interrelations" (Churchman et al., 1957) are represented by mathematical or logical symbols. The models to be studied in this chapter will be mainly symbolic.

Once the models are described, it is necessary to list the steps in model construction:

1. Make a list of the components of the system.

2. Pertinence of the components. To decide whether each of the components should be considered in the model.

3. Combining and dividing the components. Sometimes, it is better to group some components.

4. Substituting symbols. Once the final list of components is completed, it is necessary to determine whether each component has a variable or fixed value.

Solution of the model

Churchman et al. (1957) stated that to derive a solution it is necessary to follow procedures that can be classified into two types: analytic and numerical. They defined the analytic procedure as deductive and the numerical procedure (trial and error) as inductive. There are certain occasions where none of the procedures mentioned above can be applied. These cases are normally those where a term in the equation has to be previously evaluated. Therefore to evaluate the term it is necessary to apply the Monte Carlo technique. It is important to bear in mind that when the term solution appears it refers to the solution of the model and not to the solution of the real system. Analytic and numerical solutions as well as the Monte Carlo technique are briefly described below:

- Analytic solutions. When a solution is given in the form of an equation or a set of equations it is considered analytic. It is important to bear in mind that some systems are somehow restricted. These restrictions are called constraints and are normally represented by inequalities. In a more general way, a solution will be considered analytic when it is possible to solve the problem using mathematical deduction or when different types of mathematical analysis are needed to derive a solution.

- Numerical solutions. This kind of solution is given by the substitution of the symbols by numbers and finding which set of numbers gives the "maximum effectiveness" by trying every possible combination of numbers. These kinds of solutions are founded on a trial an error basis and perform several iterations.

- Monte Carlo Technique. When probability concepts are involved it is not possible to apply either analytic or numerical techniques; therefore, a technique called Monte Carlo is necessary. This technique is a procedure that helps when mathematical expressions are constructed of probability distribution functions and yields approximate evaluations of these expressions.

Churchman et al. (1957) reviewed the different types of problems that frequently appear in industry and government such as inventory problems, allocation problems, waiting-line problems, combine inventory-allocation-queuing problems, replacement and maintenance problems, and competitive problems.

Model and solution testing

The testing will be divided into two categories: model and solution. The kind of data used for testing depends on the kind of test to be performed. In the case of model testing, it measures the adequacy of the model. In other words, it has to be assessed if the *pertinent variables* have been included or not. These variables are those that have a significant effect on the effectiveness of the system. It is also important to be sure that the model accurately represents the relationships between the effectiveness measure and the independent variables. Finally, the parameters contained in the model should be evaluated properly in order to yield good results.

Testing the solution. It is supposed that a solution given by OR must lead to an improvement over the current system, so to know if this improvement has occurred it is necessary to compare it to past tests. Therefore the solution testing would determine whether the DM adopts that solution or not. For this purpose it is necessary to obtain historical data. Sometimes there are no records of this data and then the effectiveness of the new solution must be compared with that of the current system.

Control of the solution

According to Churchman et al. (1957) a "*decision rule* is a solution that can be applied repetitively". The system's parameters are the variables which define it. It is necessary to bear in mind the systems in reality are unstable and for this reason are very likely to change. In other words, the relationships between the parameters and the parameters themselves have to be adjusted and re-evaluated every time the system changes. Summarising the solution has to be controlled due to changes in the system.

The steps followed for the design of a control system are listed below:

1. Make a list of the variables, parameters, and relationships that are included or should be included in the solution.

2. Develop a procedure to detect changes in the parameters and relationships.

3. Definition of the adjustments to be made in the solution when a significant change occurs.

Implementation

This stage occurs once the solution has been found and tested. An implementation plan is needed to assure the operations' improvement. To develop the plan, Churchman et al. (1957) proposed answering the following questions, bearing in mind that "the implementation of a solution involves people taking action":

1. Who should do what?

2. When?

3. What information and facilities are required to do it?

Finally, the most important point in the implementation procedure is the continuous co-operation and communication between managers, operators, and researchers.

After discussing the Churchman et al. (1957) method for OR, it is necessary to discuss some other authors' points of view.

Pidd (1991) named the OR methodology as a "process model". From Figure 1.2 it is possible to see that this process model is considered a continuous cycle; in other words, the process can be repeated as many times as necessary during the development of the OR project.

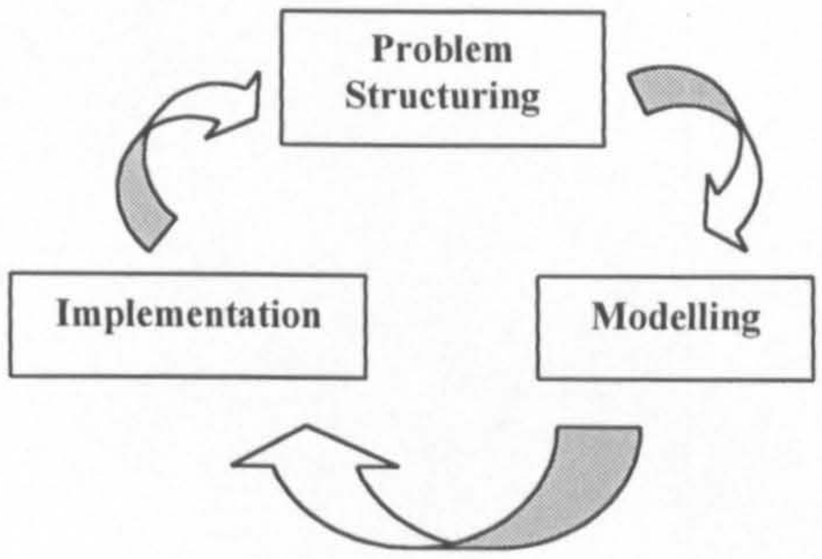


Figure 1.2 Process model
Adapted from Pidd (1991)

The process model is defined by three stages:

1. Problem structuring. This is defined as the process of extracting some agreement from a mess. Where mess occurs when there is no agreement or "common view about what constitutes an acceptable solution" (Pidd, 1991).

2. Modelling. According to Pidd (1991) this is considered as the heart of OR and involves the use of analysis methods such as mathematics, statistics and computer science. The risk of modelling can be the oversimplification of the system which is being studied.

3. Implementation. After the modelling phase is performed, some changes will be recommended. It is possible to say that an OR approach is implemented when these suggestions are taken into practice.

One of the main aspects of OR is the use of quantitative analysis. According to Anderson et al. (1996), quantitative analysis is divided into four steps: model definition, data preparation, model solution and report generation.

Model definition

Anderson et al. (1996) define a model as "the representations of real objects or situations". The kind of model used to represent a problem through mathematical relationships and symbols is called a mathematical model; according to Churchman et al. (1957) this kind of model is known as symbolic. A mathematical model is needed when the time and cost of experimentation have to be reduced. Another important fact is that the risk of experimenting with a real situation is considerably greater than that associated with experimenting with a mathematical model. Therefore the better the model represents the real situations the better the decisions and conclusions will be. In order to have a good model it is necessary to define the objectives and constraints associated to the problem the most accurately possible. The

inputs of a model can be classified as either controllable or uncontrollable. The controllable inputs are those determined by the DM whilst the uncontrollable are those that cannot be influenced.

Data preparation

The data that is going to be used by the model has to be prepared. On the one hand, Anderson et al. (1996) understand data as "the values of the uncontrollable inputs". Sometimes a separate data preparation, after the model has been defined, is needed. On the other hand, Curwin and Slater (1991) define a process called data collection where two aspects are considered: the need for data and its quality. They focus on numerical data although they bear in mind that to solve most problems; it is necessary to take into account the people, the enterprise, the culture and the environment. A very important element is the "completeness" of data, this is the DM's responsibility because he or she has to determine whether the data collected is sufficient or not. A full understanding of the purpose of the data is vital for the problem's solution. Some elements attained to data are grouped as follows (Curwin and Slater 1991):

1. Population. In order to collect data, it is necessary most of the time to do a survey that can be costly, for this reason it is essential to identify the relevant population.

2. Sources of data. Once the relevant population has been defined, the next step is to obtain the data. This is classified as primary and secondary. Primary data has to do with the "collection of new data" through observation, questionnaires and group discussions, whilst secondary data has to do with "existing data". This means that is data collected previously for other purposes.

3. Numbers and selection. The first type is census. Once the relevant population is identified, a census will be applied to the complete and enumerated population, without applying any selection procedure. If the relevant population is too big it is recommended to select a sample of it.

   The second type is random and non-random selection. To select a random sample each element of the population has the same probability of being selected. Usually, a computer generates a series of random numbers. When some judgement is required to make the selection the sample is called non-random. The most used non-random sampling is the selection of a "quota sample". This consists of identifying the characteristics sought by the survey, then the proportions of people for each characteristic will be determined and a quota will be assigned to the interviewer. Another important aspect to consider is the sample's size; to calculate a good sample size a procedure that uses an error level is required. The error level is determined and the sample's size is the one that attains that level.

4. Asking questions. Having identified the relevant population and the sample, it is necessary to determine the kinds of questions to be asked. In most cases, a questionnaire is designed. The main characteristics of the questionnaire are twofold: "logical structure" and "thought-out questions" (Curwin and Slater 1991). There are two kinds of questions: open and pre-coded. Open questions are those that allow the respondents to express their ideas and thoughts. Pre-coded questions are those that offer the respondents possible answers or alternatives to be chosen. Once the relevant population has been selected and the questionnaire has been designed, an interview has to take place. The main element of an interview is the interviewer. There are two aspects to consider regarding the interviewer's profile, namely, that this person must have had appropriate training and must have the appropriate attitude. Nowadays some interviews are made by telephone or using postal questionnaires.

5. Non-response. This case basically appears when the survey is applied on a human population and it depends on the kind of survey that is being developed. Normally, the researcher would like to reduce the non-response to a minimum.

6. Types of data. There are different types of data such as discrete, continuous, categorical or nominal, ordinal and cardinal. Nevertheless, sometimes data is classified in quantitative and qualitative.

Model solution

According to Anderson et al. (1996), in this phase the analyst or researcher (Churchman et al., 1957) tries to identify the values of the *optimal solution* for the model. For this optimal solution, identification of any one of the three solutions (analytic, numerical, Monte Carlo technique) defined by Churchman et al. (1957) will be used. After the solution is found, the DM and the analyst would like to consider its feasibility, for this reason Anderson et al. (1996) suggest a model testing and validation phase (as conducted by Churchman, Ackoff, and Arnoff). If for any reason the model shows inaccuracies a corrective action has to be applied until the testing and validation phase is satisfied. It is possible to note that this process is very similar to that proposed by Churchman et al. (1957).

Report generation

The reports are prepared using the solution of the model. As shown in Figure 1.3 to make a decision the DM needs both qualitative and quantitative information. In this context, the main emphasis is on the quantitative approach.

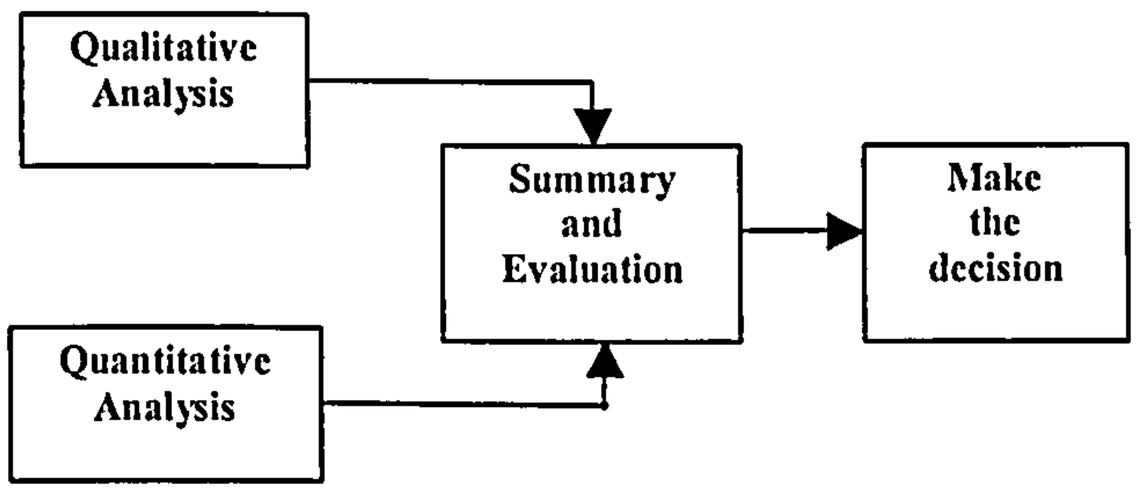


Figure 1.3 Problem analysis
Adapted from Anderson et al. (1996)

Normally, it is expected that the report gives the DM information about the results and recommends him or her a possible decision to make.

When the amount of data is very large, it is necessary to group it together in order to make the numbers more comprehensible. Curwin and Slater (1991) stated that there are three ways to present the data: tabulation of data, visual presentation and graphical presentation.

- Tabulation of data. One way is to present the data in numerical order, from the lowest to the highest values or vice versa. Examples of this kind of presentation are "frequency distribution and cross-tabulation".

- Visual presentation. This way of representing the data is through a chart or a diagram. The data will be divided into two types: discrete and continuous. The presentation of discrete data can be made through "Pie charts, Bar charts and pictograms". The presentation of continuous data can be made through "histograms".

- Graphical presentation. This is normally used when the relationship between two variables has to be shown. Curwin and Slater (1991) consider four types: graph plotting, plotting a time series, logarithmic graphs and the Lorenz curve.

One of the most popular textbooks in the study of OR is the one written by Taha (1997). According to him, a typical OR mathematical model is one that relates the variables, constraints and objective functions. Figure 1.4 shows the structure of a mathematical model, where the objective is to optimise (maximise or minimise) the objective function satisfying the constraints. The solution is known as *optimum*.

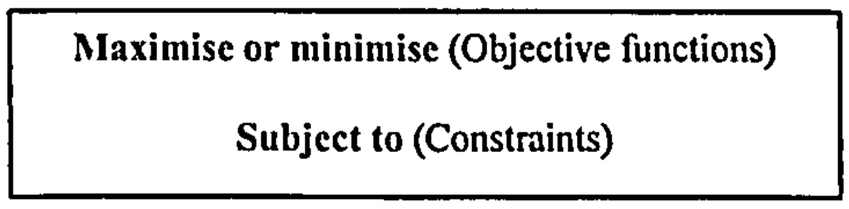


Figure 1.4 Typical OR mathematical model
Adapted from Taha (1997)

According to Taha (1997), OR should be viewed in the context of the decision-making process. The most used technique to solve optimisation problems is *linear programming*, where all the elements (objectives and constraints) of the model are linear and the variables are continuous, although there are other techniques such as integer programming, non-linear programming, goal programming, dynamic programming and network programming. Practically, the whole of these techniques are computational algorithms.

Taha (1997) states that an OR team is defined by two essential components: the OR analyst and the client. He also states that there are five principal phases for implementing OR:

1. Definition of the problem. The analyst and the client carry out this process, involving three issues: description of the decision alternatives, determination of the objective and specification of the limitations.

2. Construction of the model. Once the problem has been defined, it is necessary to transform it into mathematics (*mathematical relationships*).

3. Solution of the model. Taha (1997) considers this phase as one of the simplest because it consists of the use of standard mathematical algorithms mentioned above (e.g. linear programming). It can also involve the use of a heuristic approach or the use of simulation, and sensitivity analysis.

4. Validation of the model. In this phase, it is necessary to check if the model is giving a reasonable output and it is behaving as was expected. A way to validate if the output is suitable or not is comparing it with historical results (outputs) using the same or similar input conditions. In the case that there is no historical data of the problem, for comparison purposes, it is necessary to use simulation for the verification of the output.

5. Implementation of the solution. If the model is considered valid, the next step to follow is the *translation* of the results into *operating instructions*. These instructions have to be understandable to the person or group of people who will administer the system.

Taha (1997) considers the following areas as those of relevant interest within the OR study: linear programming as mathematical technique, data envelopment analysis, inventory control, project management, queuing theory, computer simulation methods and decision analysis.

1.2 Handling Preferences

Preferences are essential in the modelling of decision-making. According to Vincke (1992) in the preference structure, it is assumed that a decision-maker has to compare two actions *a* and *b* where:

*a*P*b* means that action *a* is preferred to action *b*
*a*I*b* means that action *a* is indifferent to action *b*

Some authors use indistinctively the terms criteria or preference to denote criteria, for effects of this thesis it is important to understand the difference between these two terms.

The classification of Hwang and Masud (1979) described different stages where the information from the DM could be needed. These stages are:

1. No Articulation of Preference Information.

2. A Priori Articulation of Preference Information.

3. Progressive Articulation of Preference Information (Interactive Methods).

4. A Posteriori Articulation of Preference Information (Nondominated Solutions Generation Methods).

1.3 Overview

The main objective of this research is to find the solution of optimisation problems (multiple criteria decision-making) using OR techniques that can handle the DM's preferences in an interactive way. The originality of this research is the way in which it modernises two interactive methods proposed in the 1970's and then refines them through contemporary areas of research such as genetic algorithms and fuzzy logic. These methods were selected because not only can they handle the DM's preferences in an interactive way but they also offer a direct way for the development of a hybrid approach using OR and evolutionary techniques.

Based on this objective the structure of this thesis is described below:

Chapter One introduces the basic concepts of OR, visualising it more as a methodology than as a set of techniques.

Chapter Two describes the main characteristics of the evolutionary algorithm, giving the general procedure and its formal definition. Furthermore, the definitions of genetic algorithms, along with their genetic operators are presented. This chapter also describes the basic techniques of evolutionary computation to solve a well-known problem called the *traveling salesman problem* (TSP). This problem was chosen because it is not only an NP-complete problem but also an optimisation problem and for the purpose of this research gives the general idea on how to attack this kind of problem using evolutionary algorithms. This chapter presents the most commonly used genetic operators in the solution of the TSP.

The area of decision theory, which is defined as making a decision by mathematical means, is described in Chapter Three. In this chapter the effect of the incorporation of risk and

uncertainty in a problem is also discussed, defining some basic characteristics of risk analysis. Moreover, an area that involves OR and decision theory called multiple criteria decision-making (MCDM) is defined in detail along with the most popular methods for the solution of MCDM problems. Finally, the most important operators of multiple objective evolutionary algorithms are discussed.

Chapter Four describes a GA-OR approach where the GA uses an interactive method called *Sequential Multiobjective Problem Solving* (SEMOPS). This new approach is named GA-SEMOPS and can be understood as the support system used to help the DM to solve the decision problem. In this chapter two case studies are solved: the Bow river case study and the nurse-scheduling problem.

- The Bow River Valley is a hypothetical example of an artificial river basin and its pollution problems. In this problem there are three pollution sources, the Pierce-Hall Cannery, and the towns of Bowville and Plympton towns and the main objective is to improve the water quality considering the DM's preferences. The problem is solved using two different techniques a real-valued genetic algorithm and a *mutliobjective genetic optimiser* using a fuzzy rule-based system. This case study is the one presented by the authors of the SEMOPS method and it was selected for comparison purposes.

- The nurse-scheduling problem consists of generating a schedule of working days and days off for each nurse in a hospital subject to hard and soft constraints. The approach presented in this thesis considers a multiple objective nurse scheduling problem involving the decision making process attached to it. In order to manage this process it has been decided to use the SEMOPS method in conjunction with a genetic algorithm (GA), developing a hybrid approach that uses the strengths of both operations research and evolutionary computation techniques.

In Chapter Five a GA-OR approach is described where the GA uses the *Probabilistic Trade-off Development* (PROTRADE) method. The main characteristics of PROTRADE are that it incorporates uncertainty and risk analysis in the solution of a problem, and it also allows the DM to introduce his or her preferences. The new model is called GA-PROTRADE and is used to solve a land allocation problem. This case study is a multiple use approach to land reclamation and it is solved by means of a real-valued genetic algorithm, considering risk and uncertainty. This problem is the one presented by the authors of the PROTRADE method and it was selected for comparison purposes.

Chapter Six introduces a new proposal for the solution of multiple objective optimisation problems and describes its main characteristics. This model is called the *interactive procedure for multiple objective optimisation problems* (IPMOOP) and is used for the solution of a resource allocation problem in the Automatic Control and Systems Engineering at the University of Sheffield. This problem is selected in order to implement a real life application of the IPMOOP and to validate it.

Finally, Chapter Seven outlines the conclusions obtained from the results of this research and the future applications for these methods.

Figure 1.5 shows the relationships between the different chapters and the types of codification used in each case study. It is possible to observe that Chapters Two, Four, Five and Six use the concepts outlined in Chapter Two, whilst Chapters Four, Five and Six also use the concepts presented in Chapter Three. Furthermore, it is also evident that the genetic algorithm's codification used in Chapter Two is integer, in Chapter Four for the first case study it is real-valued whilst in the second one it is binary string, in Chapter Five it is real-valued and in Chapter Six it is binary string.

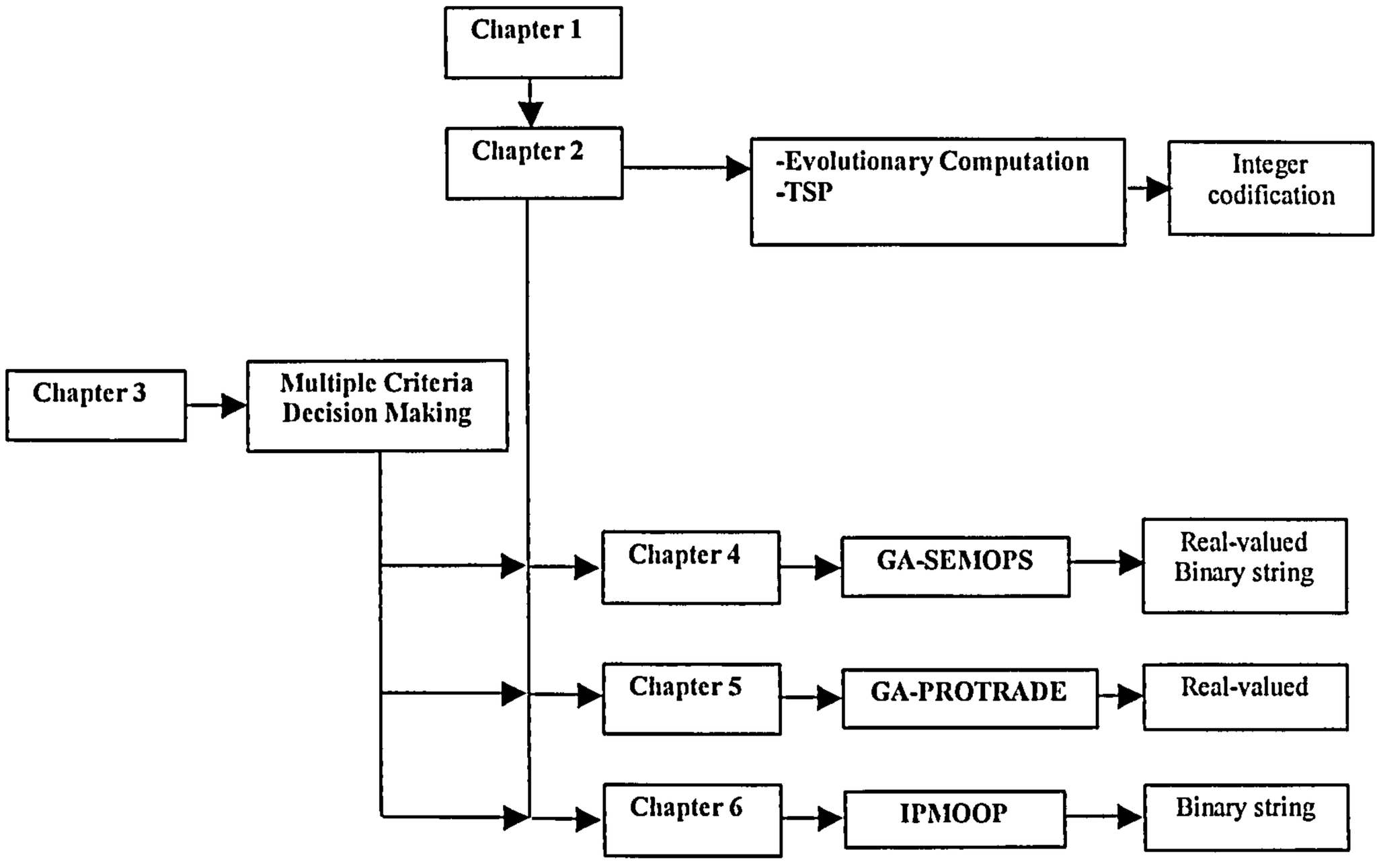


Figure 1.5 Thesis Structure

1.4 Contributions

The main contributions of this research work are:

- The development of a hybrid model called GA-SEMOPS (Chapter Four). The main characteristics of this approach are: the use of a surrogate function that can be directly understood as the objective function and mapped as the GA's fitness function and the use of a set of auxiliary problems reducing the solution space. Additionally, another important feature is the use of goal and aspiration levels and finally, the sequential decision-making process with the direct intervention of the DM. Another contribution is the demonstration of the use of this model to solve different multiple objective optimisation problems and different kinds of GAs.

- The development of a fuzzy multiple objective genetic optimiser (Chapter Four). This model is based on a fuzzy rule-based system (FRBS) in order to achieve an automated process emulating the DM. Considering that the problem has several objectives, a number of fuzzy sets will be assigned to each objective. These sets are defined in terms of standard deviations and aspiration levels. Once the fuzzy sets are defined, it is important to determine the shapes of the membership functions. Finally the set of rules is created, using both the fuzzy set and the membership functions. This set of rules is programmed into the automated algorithm.

- The development of a hybrid approach called GA-PROTRADE (Chapter Five). This model allows the introduction of risk and uncertainty in the solution of a problem. One of its most important features is that it handles the DM's preferences in two different levels.

The first level is based on pairwise comparisons made by the DM to determine the importance of each goal compared to the others. This process yields a classification of goals from the most preferred to the least important. The second level for handling preferences is based on the assignment of a probability of achievement to each goal. The DM is asked to define the probability of achievement of each goal creating a new solution space and a new surrogate objective function. Therefore, the GA has to be applied again finding new solutions. Finally, the DM and the analytic programmer have to decide whether or not the solutions found are acceptable and select which one is more suitable.

- The development of the interactive procedure for multiple objective optimisation problems (IPMOOP), in Chapter Six. This procedure is focussed on the decision-making process followed when a real life problem is solved. The main objective of this model is to visualise the decision-making process as a continuous interaction between the DM and the analytic programmer. For this reason a unit called the *decision-making process group* (DMPG) is defined, showing this interaction. This model is based on the formulation and solution of an *initial problem*. In this model an initial problem is proposed because it is expected that in the first attempt at a solution of a problem not all the aspects attached to it can be visualised. Therefore, after the solution of the initial problem a *final problem* has to be formulated and solved. Once the final problem is solved the DMPG unit decides whether an acceptable solution has been found or not. This procedure can be applied as many times as necessary until a solution found satisfies the DM.

CHAPTER 2

Evolutionary Computation

2.1 Introduction

Since the main objective in this research, as outlined in Chapter One, is the solution of optimisation problems using evolutionary algorithms, it is important here to describe evolutionary computation. Moreover, most optimisation methods perform a search to find a single-optimal solution whilst evolutionary computation performs a search to find a population of solutions. Therefore the outcome in evolutionary computation methods is also a population of solutions.

The origins of evolutionary computation date from the late 1950's when some computer scientists started to use the Darwinian evolution theory ("Struggle for Existence") to solve optimisation problems (Bäck et al., 2000). In general, evolutionary computation systems consisted of using operators that were defined as processes in natural selection (reproduction, crossover, and mutation). Despite the flexibility and adaptability of evolutionary computation in the searching task it was not until the 1980's when this set of methodologies became well known in the scientific community.

The main purpose of evolutionary computation is to model the evolution process and simulate it using a computer. The concept of evolution may be used as one search method to find the best solution out of a great number of possible solutions (Mitchell, 1996). The term evolutionary computation was defined in 1991 as including three avenues: genetic algorithms, evolutionary programming and evolution strategies (Bäck et al., 1997). If these avenues are considered as algorithms then the term "Evolutionary Algorithms" emerges.

Firstly, this chapter describes the main characteristics of the evolutionary algorithm, giving the general procedure and its formal definition. Additionally, the definitions of genetic algorithms along with their genetic operators are presented. In subsection 2.2.2, a description of evolutionary programming and its operators is given. In subsection 2.2.3, the evolution strategies are shown as well as the definition of their recombination, mutation and selection operators. A comparison between genetic algorithms, evolutionary programming and evolution strategies is presented in section 2.3.

Secondly, this chapter describes a well-known problem called the Traveling Salesman Problem (TSP). This problem concerns a salesman who has to visit n cities with the restriction that he must visit each city once. The solution expected is the tour of all the cities which implies the minimal cost (Hamiltonian cycles).

In sub-section 2.4.1 some traditional search methods for the TSP are explained. To solve this problem using genetic algorithms a considerable number of crossovers have been developed sub-section 2.4.2 contains a summary of these operators.

Additionally, in sub-section 2.4.3 other methodologies to solve this problem as genetic local search, evolution strategies, and evolutionary programming are analysed. Genetic local search is a hybrid that consists of the traditional methodologies used in a genetic algorithm. Evolution strategies are based on mutation and recombination operators. Genetic algorithms sometimes use these mutation operators.

Finally, in section 2.5 some applications of TSP for real-world problems are discussed, especially for scheduling manufacturing problems.

2.2 Evolutionary Algorithms

Evolutionary algorithms (EAs) are instances of algorithms that work with evolutionary principles. Genetic algorithms, evolutionary programming and evolution strategies are subsets of evolutionary algorithms (Bäck, 2000).
Michalewicz (1994) describes the following structure of an evolutionary algorithm (he called it *evolution program*):

Procedure ***evolution program***

```
begin
        t ← 0
        initialise P(t)
        evaluate P(t)
        while (not termination-condition) do
        begin
                t ← t + 1
                select P(t) from P(t - 1)
                alter P(t)
                evaluate P(t)
        end
end
```

In the above, P(t) is a population of individuals, which is initialised at $t = 0$, and this is considered as the starting point in the program. An evaluation function is applied and returns the fitness, comparing among the solutions. A termination-condition is stated, determining the number of iterations the program is going to run. The counter t is incremented by one; an individual is selected from the previous population of individuals and is altered by a "genetic" operator. It is necessary to evaluate the new P(t); the program will run until the termination-condition is reached and it will return the solution of the problem.

Basically, the genetic operators will be classified in three asexual, sexual and panmictic as shown in the following definition (Bäck, 1996):

Definition 2.1
A genetic operator $\upsilon_\theta : I^P \to I^q$ *is called*

$$
\begin{aligned}
\text{sexual} \quad &: \leftrightarrow \quad \exists \upsilon'_\theta : I^2 \to I \ : \\
&\upsilon_\theta(\vec{a}_1, \cdots, \vec{a}_p) = (\upsilon'_\theta(\vec{a}_{i1}, \vec{a}_{j1}), \ldots, \upsilon'_\theta(\vec{a}_{iq}, \vec{a}_{jq})) \\
&\textit{where } \forall k \in \{1, \cdots, q\} \ \ i_k, j_k \in \{1, \cdots, p\} \\
&\textit{are chosen at random,}
\end{aligned}
\tag{2.1}
$$

$$
\text{asexual} \quad : \leftrightarrow \quad \exists \upsilon'_\theta : I \to I \ : \\
\upsilon_\theta(\vec{a}_1, \cdots, \vec{a}_p) = (\upsilon'_\theta(\vec{a}_1), \cdots, \upsilon'_\theta(\vec{a}_p)) \wedge p = q,
$$

$$
\text{panmictic} \quad : \leftrightarrow \quad \exists \upsilon'_\theta : I^p \to I \ : \\
\upsilon_\theta(\vec{a}_1, \cdots, \vec{a}_p) = \underbrace{(\upsilon'_\theta(\vec{a}_1, \cdots, \vec{a}_p), \cdots, \upsilon'_\theta(\vec{a}_1, \cdots, \vec{a}_p))}_{q}
$$

where $I = A_x \times A_s$ is the space of individuals, A_x, A_s denote arbitrary sets, υ_θ is the genetic operator, p is the initial population size and q is the population size after the genetic operator is applied (q and p can be equal), and $\vec{a}$ is an individual of the population selected for the genetic operation.

Recombination is an example of a sexual or panmictic operator whilst mutation is an asexual operator; recombination and mutation operators will be explained in more detail in sub-sections 2.2.1.2, 2.2.1.3, 2.2.3.1, and 2.2.3.3.

2.2.1 Genetic Algorithms

John Holland first proposed genetic algorithms (GAs), and Goldberg (1989) followed up his research; these algorithms basically simulate genetics (Fogel, 1994).
As stated in (Goldberg, 1989):

> "Genetic algorithms are search algorithms based on the mechanics of natural selection and natural genetics. They combine survival of the fittest among string structures with a structured yet randomised information exchange to form a search algorithm with some of the innovative flair of human search. In every generation, a new set of artificial creatures (strings) is created using bits and pieces of the fittest of the old, an occasional new part is tried for good measure. While randomised, genetic algorithms are no simple random walk. They efficiently exploit historical information to speculate on new search points with expected improved performance."

GAs are stochastic algorithms that use Darwinian struggle for survival based on natural evolution. The main purpose of the original GA was to solve optimisation problems using binary strings (0's and 1's). These strings (chromosomes) are taken from a certain domain space. GAs start with a population of strings instead of using a single point, generating successive populations of strings (offspring). The population of strings is considered as a set of potential solutions; these solutions are evaluated by an objective function that plays the role of an environment (Bäck, 2000).

Normally a GA has three operators: 1. Reproduction, 2. Crossover, and 3. Mutation. The definitions of these three operators are presented to give a general idea of the GA's operators; later on they will be presented more in detail.

1. Reproduction is the process in which strings with the maximum objective function values have more probability of being copied (living), while the strings with the minimum objective values will not be used (die) in the next generation.

2. Crossover is the process in which the reproduced strings are mated randomly and each pair of strings swaps their characters starting from a position of the strings that is randomly selected.

3. Mutation is the process that alters one or more genes (chromosome positions), with a determined probability (mutation rate). This process is considered as a secondary mechanism (Goldberg, 1989). Michalewicz considers mutation to be an important genetic operator; sometimes it is utilised prior to crossover.

Eshelman et al. (1989) stated that the effectiveness of GAs is based on the exploitation of the result found (past) joined to the exploration of new areas in the search space. Then, the selection operator is a mean for exploitation and the mutation and crossover operators are means for exploration.

The terminology used in GA has its roots not only in the field of genetics but also in computer science. Table 2.1 presents the correspondence between "natural" and "artificial" terminology.

Natural	**Genetic Algorithm**
Chromosome	String
Gene	Feature, character, or detector
Allele	Feature value
Locus	String position
Genotype	Structure
Phenotype	Parameter, set, alternative solution, a decoded structure
Epistasis	Non-linearity

Table 2.1 Comparison of Natural and GA terminology
Adapted from Goldberg (1989)

In GA, the chromosomes are often binary strings, and it will be necessary to modify or convert the original problem into a convenient form, taking potential solutions and converting these into binary representation. In this case the problem will be modified to support the binary representation as it is shown in Figure 2.1.

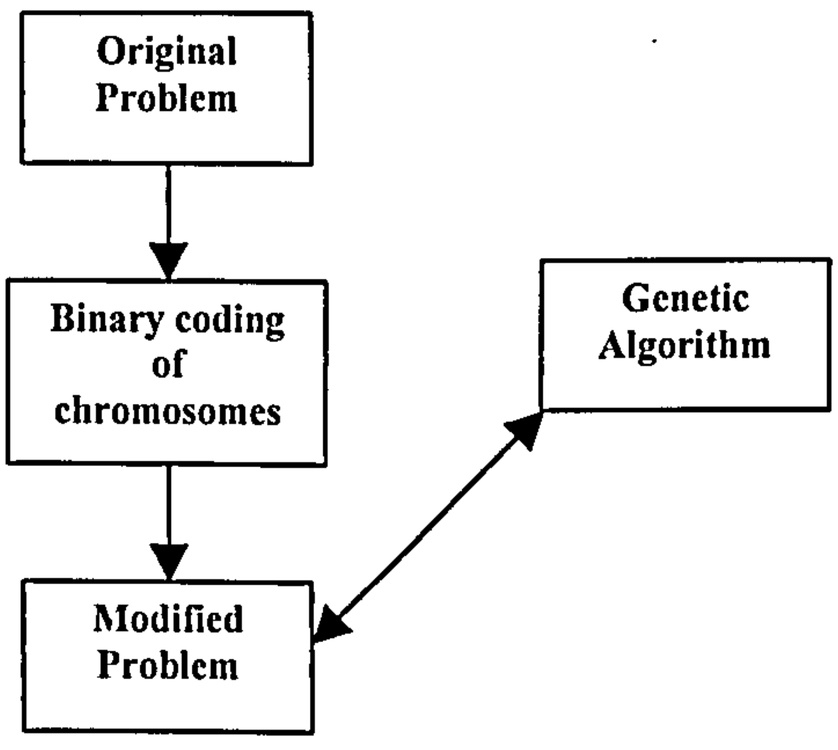


Figure. 2.1 Codification process for GAs.
Adapted from Michalewicz (1994).

Some applications of Genetic Algorithms are optimisation, automatic programming, machine learning, economics, immune systems, ecology, population genetics, evolution and learning, and social systems (Mitchell, 1996).

There are other codification structures for GAs such as real-valued vector or floating-point representation; these are discussed in greater depth in Chapter Four.

2.2.1.1 Selection in GA

Selection can be considered as an operator in GA (Deb, 2000). The main objective of this operator is to select the best solutions from the population and delete the poor solutions. In some cases the selection operator is the combination of reproduction and selection. To define a selection procedure it is necessary to divide it into 3 principal steps (Grefenstette, 2000):

1. The objective function must be mapped to fitness function.

2. The probability distribution is created using the fitness as a basis.

3. The samples are drawn from the probability distribution.

The objective function is mapped to fitness function. Defining the objective function as:

$$f : K_x \rightarrow R \tag{2.2}$$

where R is the set of real numbers and K_x is the object variable space, the objective function measures are cost and reward. Reward is measured in a maximising problem and cost in a minimising problem. The fitness function is defined as follows:

$$\phi : K_x \rightarrow R^+ \tag{2.3}$$

where R^+ is the set of positive real numbers, then the fitness function maps the scores of the objective function to a non-negative set. The fitness function can be defined in terms of the objective function:

$$\phi(k_i(t)) = g(f(k_i(t))) \tag{2.4}$$

where $k_i \in K_x$. The next issue to consider is if an individual has a very high performance, the fitness function will tend to assign similar fitness values to the members of the population, leading to premature convergence. To solve this problem fitness scaling is necessary; this scaling is defined as a time-varying linear transformation of the objective value (Grefenstette, 2000):

$$\phi(k_i(t)) = \alpha[f(k_i(t)) - \beta(t)] \tag{2.5}$$

where α is –1 for minimisation problems and +1 for maximisation problems and $\beta(t)$ is the worst value seen in the last few generations.

Different methods of selection exist: proportional selection, tournament selection, rank-based selection, boltzman selection and other selection methods (Blickle, 2000), (Grefenstette, 2000b).

The proportional selection method consists of creating a number of offspring in proportion to an individual's fitness. This method works generating a probability distribution:

$$P_{prop}(i) = \frac{\phi(i)}{\sum_{i=1}^{\mu} \phi(i)} \tag{2.6}$$

where μ is the population size, then this procedure of calculating the proportional probability will be applied to each individual of the population. With this probability distribution it is possible to select one parent to use for reproduction purposes. One of the best known methods of proportional selections is the roulette wheel. In the roulette wheel selection each individual in the population will have a slot of the wheel, the size of the slot depending on the fitness of the individual. Every time it is necessary to select an individual the roulette wheel will be spun. Then the individuals with better fitness will be more likely to be reproduced as part of the offspring.

In tournament selection a group of individuals is chosen randomly (Blickle, 2000). These individuals participate in a tournament where the one with the best fitness value wins. The winner is inserted into the next population. In order to obtain the new population this process

is repeated x times. This selection procedure could be implemented in a polynomial time complexity $O(x)$, and scaling and translation do not affect it.

$$
\begin{aligned}
&r \in \{1, 2, \dots, x\} \\
&P(t) = \{p_1, p_2, p_3, \dots, p_n\} \\
&P(t) \in I^x \\
&P'(t) = \{p'_1, p'_2, p'_3, \dots, p'_x\}
\end{aligned}
\tag{2.7}
$$

where $P(t)$ is the initial population, n is the number of individuals and r is the tournament size (arbitrary). To denote the population after selection $P'(t)$ is used.

The algorithm that describes the tournament selection procedure is presented below, where *bestind* is defined as the function that selects the individual with the best fitness from the subset taken from $P(t)$ of size r.

Tournament selection algorithm

```
Algorithm ( int i, )
i ←1;
r ← random( );
while (i ≤ r)
{
      p'i ← bestind(P(t), r)
      i ← i + 1;
}
```

In rank based selection (Grefenstette, 2000b) the probability of selection is determined by the rank of the individual's fitness. Following the steps of the general selection process: 1. The objective function must be mapped to fitness function. 2. The probability distribution is created using the fitness as a basis. 3. The samples are drawn from the probability distribution.

1. Mapping the objective function to fitness is represented by:

$$\phi(a_i) = \delta f(a_i) \tag{2.8}$$

where ϕ is the fitness function, f is the objective function and δ is -1 when it is desired to minimise and $+1$ when it is desired to maximise.

2. For ranking there are different methods such as linear ranking, non-linear ranking, (μ, λ), $(\mu+\lambda)$, and threshold selection (the last three methods are used in evolution strategies). As an example of these ranking methods linear ranking is explained below.

If the size of the population is μ, fitness is ranked from zero for the worst fitness individual and μ-1 for the best. Once, the individuals are ranked a selection probability is assigned. This selection probability is proportional to the individual's rank. The selection probability for each individual is defined as follows:

$$P_{rank}(i) = \frac{\alpha_{rank} + \left[rank(i) / (\mu - 1) \right] (\beta_{rank} - \alpha_{rank})}{\mu} \tag{2.9}$$

where α_{rank} is the number of offspring assigned to the worst individual, and β_{rank} is the expected number of offspring to be assigned to best individual during each generation (Grefenstette, 2000b). Given $\alpha_{rank} = 2 - \beta_{rank}$, and $1 \leq \beta_{rank} \leq 2$, then the expected number of offspring of the best individual is no more than twice the number of the population average. This prevents the algorithm from having a premature convergence when a very good individual (super individual) appears.

2.2.1.2 Crossover in GA

Crossover is a recombination operator. In GAs there are several kinds of crossover such as one point, two point, multiple-point, uniform, segmented and shuffle.

As Goldberg (1989) defined the one-point crossover consists of selecting randomly two individuals from the parent population, selecting a random position in the chromosomes (to choose the position a uniform distribution is used) and swapping the segments to the right hand side of the position selected. This will generate two new individuals (offspring).

In Eshelman et al. (1989) the two-point crossover consists of a chromosome considered as a ring and two numbers randomly selected that will segment the chromosome and will produce two offspring.

De Jong (1975) proposed a generalised crossover model where a new variable was considered. This variable is the number of crossover points and is called CP, to have the one-point crossover case CP is equal to 1, for the two-point crossover CP is equal to 2. The chromosome is still considered as a ring for an even CP and for an odd CP one of the crossing points is considered fixed in position 0. Both of these cases produce two new children. If CP is greater than 2 the crossover is known as multiple-point (Figure 2.2).

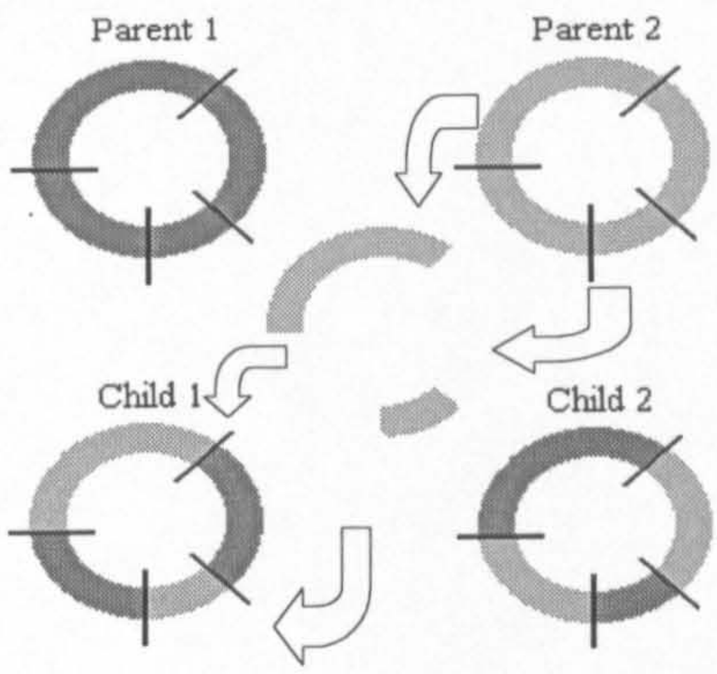


Figure 2.2 Multiple-point crossover with CP = 4

According to Syswerda (1991) the uniform crossover exchanges bits with fixed probability; this operator uses a crossover mask. This mask consists of a string the same size than the chromosomes where for each bit the mask parity determines which of the two parents will give its bit to the child. This crossover is called uniform due to the fact that the distribution of 1-bits is uniform with a bit probability of occurrence of 0.5. Hence the mask is randomly generated and for each mask there is a corresponding inverse mask.

Segmented crossover is considered a variant of multiple-point crossover that allows the variation of the crossover positions (points).

Eshelman et al. (1989) introduced the shuffle crossover. This technique is very similar to one-point crossover. It consists of a random shuffle of the bit positions in both parents, having

swapped the segments to the right of the crossover position it is necessary to unshuffle the offspring to recover the original ordering positions.

2.2.1.3 Mutation in GA

Basically, mutation is understood as the process of generating a new child from one and only one parent, $m : I^p \rightarrow I^p$. To apply the mutation operator to a binary string (chromosome) $\vec{a} = (a_1, \cdots, a_p) \in I = \{0,1\}^p$ it is necessary to fix a probability of mutation p_m (this probability is small) and to determine positions i_j randomly, where $i_j \in \{1, \ldots, p\}$ (such positions have the same probability p_m of being chosen). In other words, p_m is the probability of independently inverting each a_i thus after the mutation operator is applied the new individual is $\vec{a}' = m(\vec{a})$ where $m : \{0,1\}^p \rightarrow \{0,1\}^p$ and

$$a'_i = \begin{cases} a_i & u > p_m \\ 1 - a_i & u \leq p_m \end{cases} \tag{2.10}$$

where u denotes a uniform random variable U({0,1}) (Bäck et al., 2000).

2.2.2 Evolutionary Programming

Fogel et al. (1966) developed evolutionary programming (EP). Initially their objective was to develop artificial intelligence through the simulation of natural evolution. They showed that to have intelligent behaviour it was necessary to predict environment (the organism is only as intelligent as the quality of predictions are). Environment was posed as a sequential source of symbols. An organism was needed for transforming a sequence of input symbols into a sequence of output symbols. This organism was represented mathematically and was considered as a "parent". The organism was a finite-state machine with an alphabet (finite) of input symbols, an alphabet (finite) of output symbols and a finite number of internal states.

Evolutionary programming techniques consisted of two operators: selection and mutation. Mutation was considered to be the production of an offspring through a single modification of the parent; an average cost of the errors was calculated for both parent and offspring. Comparing the parent and offspring cost of the errors, if the offspring was superior (in terms of lower cost) the parent was discarded and the offspring served as a parent to produce a new offspring. If the parent was superior the offspring was discarded and the parent was used again to produce a new offspring. This was considered the selection operator. Evolutionary programming is still considered an area of active research (Mitchell, 1996).

In Figure 2.3 a finite state machine is shown, considering the left of the slash as the input symbol (1,0) and the right of the slash as the output symbol (α, β, γ). The machine starts in state A (De Jong et al., 2000) (Fogel et al., 1966).

The population is a set of finite-state machines so the parents and the offspring will be machines as well. Moreover, there are five possible modes of mutation: change an output symbol, change a state transition, add a state, delete a state or change the initial state (Porto, 2000) (Fogel et al., 1966). The mutation operator is applied randomly, based on a probability distribution typically uniform. After mutation the machines that have the best ability to predict each next symbol become parents of the next generation of offspring.

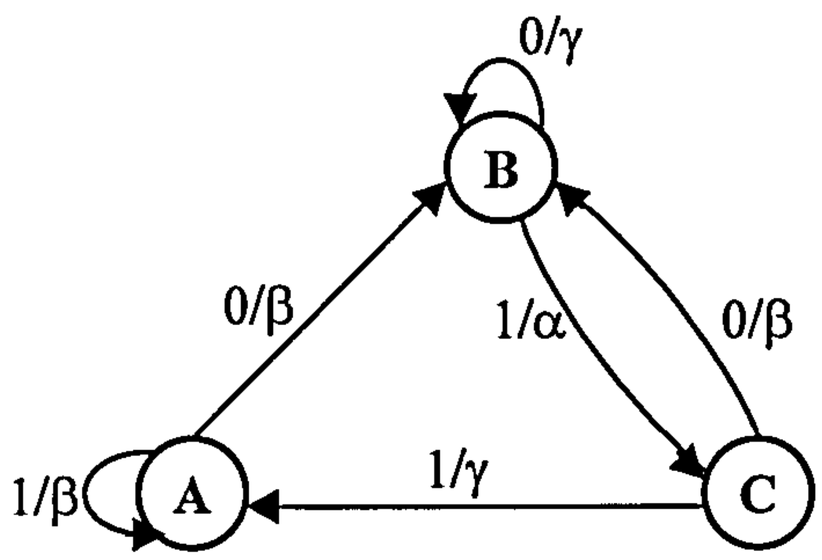


Figure 2.3 Finite-state machine.
Adapted from Fogel et al. (1966)

Fogel (1991) retook evolutionary programming where the following sequence of operations was proposed:

1. Generating the initial population P_i setting a random vector V_i, a uniform distribution in n dimensions ranged from a to b inclusive $U(a,b)^n$, then:

$$P_i = V_i \sim U(a,b)^n, \forall\ i = 1, \ldots, k \tag{2.11}$$

   where k is the number of parents.

2. Assigning a fitness score $\phi_{pi} = G(F(p_i),v_i)$, where v_i represents random alteration in p_i or random variation on the $F(p_i)$ evaluation. Fitness function F maps p_i to the real number set $p_i \rightarrow \mathrm{R}$ and G determines the fitness score to be assigned.

3. Altering p_i and generating p_{i+k}:

$$p_{i+k,j} = p_{i,j} + N(0, \beta_j \phi_{pi} + z_j), \forall j = 1, \cdots, n \tag{2.12}$$

   where $N(\mu,\sigma^2)$ is a Gaussian random variable with variance σ^2 and mean μ:

$$\mu = 0 \text{ and } \sigma^2 = \beta_j \phi_{pi} + z_j \tag{2.13}$$

   where $z_j > 0$ represents a positive offset and $\beta_j > 0$ represents a positive constant of proportionality to ϕ_{pi}. This is considered the mutation operator.

4. Assigning a fitness score to each p_{i+k}:

$$\phi_{p_{i+k}} = G(F(p_{i+k}), v_{i+k}) \tag{2.14}$$

5. Assigning a weight w_i to each p_i.

6. Ranking p_i in descending order according to their w_i values.

EP has been applied successfully to problems in prediction, identification and automatic control. Other areas where EP has had success are optimisation of neural networks, optimal routing, drug design, game theory, control of heating, ventilation and air conditioning (Syswerda, 1991). Because EP can be used to solve NP-complete problems especially in commercial and military domains, it is expected that EP will be better developed in future research.

2.2.3 Evolution Strategies

Evolution strategies (ES) were developed in Germany in 1964 (Rechenberg, 1965) and, in the same way as genetic algorithms and evolutionary programming, they follow the principles of natural evolution. The aim of these strategies was to solve optimisation problems. Normally these strategies are algorithms completely based on recombination, selection and mutation operators even though the first evolution strategy ((1+1)-strategy) worked just with selection and mutation operators. This strategy is called "two membered evolution strategy" (Michalewicz, 1994) and the selection mechanism is represented by the term (1+1)-selection. The population is defined with one individual, which is considered the parent. The mutation operator is applied to the parent resulting in an offspring. If the offspring's fitness is better than the parent's, the offspring replaces its parent, otherwise the parent remains unchanged. The representation of the parent is made using a pair of float-valued (real-valued) vectors:

$$v = (x, \lambda)$$

$$x \in \mathrm{R}^n$$
$$\lambda \in \mathrm{R}$$
$$v : \mathrm{R}^n \times \mathrm{R} \rightarrow \mathrm{R}^{n+1} \qquad (2.15)$$

where vector x represents a point in the search space and λ represents a vector of standard deviations. Mutations are applied by adding a normally distributed random vector $Z \sim N(0,\lambda)$ that is multiplied by a scalar $\sigma > 0$:

$$x_{t+1} = x_t + \sigma_t Z_t \qquad (2.16)$$

where N(0, λ) is a vector with a mean of zero and standard deviations λ. Defining f as the objective function to be maximised, the offspring replaces the parent only if $f(x_{t+1}) > f(x_t)$.

It is obvious that in the (1+1)-ES the concept of population is not used. For this reason a multi-membered evolution strategy (μ+1)-ES was created, where $\mu > 1$. In this ES μ parents are recombined to obtain the offspring to which the mutation operator is applied, the recombination operator is explained in detail in section 2.2.3.1. The resulting offspring of this procedure will substitute the worst parent only if it is better. The (μ+1)-ES was the foundation to the (μ+λ)-ES and (μ,λ)-ES (Bäck, 1996). In particular, the (μ+λ)-ES or (μ,λ)-ES implies that μ parents will produce λ offspring.

2.2.3.1 The Recombination Mechanisms

The recombination operator will be called r, and basically is the selection of two individuals from the population μ. The main difference between a sexual and a panmictic recombination operator according to Bäck (1996) lies in the way the parents are chosen. In other words, in the sexual recombination operator the two individuals (parents) are randomly chosen from the initial population; this includes the possibility of choosing the same parent twice in order to generate the new individual (offspring). In the panmictic recombination operator one parent is

chosen randomly and it maintains fixed then it is necessary to select randomly a second parent for each vector of the first.
As regards the (1+1)-ES there is no recombination operator conversely (μ+1)-ES, (μ+λ)-ES, and (μ,λ)-ES will have the following operator (Bäck et al., 1991):

$$r \quad : \quad I^{\mu} \rightarrow I \qquad \text{recombination operator} \tag{2.17}$$

Moreover there is another classification for evolution strategies (ES) recombination operators: discrete and intermediate. Both classifications can be mixed; for example it is possible to have a panmictic intermediate recombination or a sexual discrete recombination.

On the one hand, to obtain the components of the offspring in the intermediate case it is necessary to calculate the arithmetical mean of the corresponding components of the two parents selected for recombination. Schwefel presented a generalised intermediate recombination operator that consists of setting arbitrary weights within an interval [0,1] instead of having the unique value of 0.5 as in the arithmetical mean (Bäck, 1996). The sexual intermediate recombination operator *r* is presented in the following definition,

Definition 2.2

The recombination operator r is:

$$r(P') = a' = (x', \sigma') \in I$$

$$x'_i = \frac{1}{2}(x_{a,i} + x_{b,i}) \qquad i = 1, \cdots, n$$
$$\sigma'_i = \frac{1}{2}(\sigma_{a,i} + \sigma_{b,i}) \tag{2.18}$$

where the parents $a = (x_a, \sigma_a)$ and $b = (x_b, \sigma_b) \in I$, P' is the population, and x'_i and σ'_i represent the offspring.

On the other hand, Bäck et al. (1991) proposed that in order to obtain the components of the offspring in the discrete recombination operator these are copied from one of the parents randomly selected. The sexual discrete recombination operator *r* is presented in the following definition:

Definition 2.3

The recombination operator r is (Definition 2.2)

$r(P') = a' = (x', \sigma') \in I$ *where* $x' \in \mathrm{R}^n, \sigma' \in \mathrm{R}^n$

$$x'_i = \begin{cases} x_{a,i}, & X \leq 1/2 \\ x_{b,i}, & X > 1/2 \end{cases} \qquad \forall i \in \{1, \cdots, n\}$$
$$\sigma'_i = \begin{cases} \sigma_{a,i}, & X \leq 1/2 \\ \sigma_{b,i}, & X > 1/2 \end{cases} \qquad \forall i \in \{1, \cdots, n\} \tag{2.19}$$

where the parents $a = (x_a, \sigma_a)$ and $b = (x_b, \sigma_b) \in I$, X is a uniform random variable within the interval [0,1]. The probability of mating for each element of the population is the same.

2.2.3.2 Selection Operators

According to Bäck (1996) the selection mechanisms in ES are mainly deterministic. Considering (μ+1)-ES as the first multi-membered evolution strategy, the selection operator *s* is defined as the one that removes the individuals with the worst fitness value of the population. This operator is applied before the next offspring generation is produced (Bäck et al., 1991).

$$s \quad : \quad I^{\mu+1} \rightarrow I^{\mu} \tag{2.20}$$

In Bäck (1996) the (μ+λ)-selection operator selects μ individuals (the best) from the set that is constituted by the union between the parent set and the offspring set. The set resultant from the application of the selection operator is used as the new parent set for the next generation. The (μ+λ)-selection operator is represented as follows:

$$s_{(\mu+\lambda)} \quad : \quad I^{\mu+\lambda} \rightarrow I^{\mu} \tag{2.21}$$

The (μ,λ)-selection operator selects the μ best individuals from the offspring set to generate the new parent set for the next generation. The (μ,λ)-selection operator is represented as follows:

$$s_{(\mu,\lambda)} \quad : \quad I^{\lambda} \rightarrow I^{\mu} \tag{2.22}$$

2.2.3.3 Mutation Operators

As mentioned above mutation *m* is an asexual operator. This operator is presented in the following definition (Bäck, 1996):

Definition 2.4

The mutation operator m is:

$$m(\vec{x},\vec{\sigma},\vec{\alpha}) = (\vec{x}',\vec{\sigma}',\vec{\alpha}') \qquad \forall i \in \{1,\cdots,n\}, \forall j \in \{1,\cdots,n\cdot(n-1)/2\}:$$

$$\begin{aligned}
\sigma'_i &= \sigma_i \cdot \exp(\tau' \cdot N(0,1) + \tau \cdot N_i(0,1)) \\
\alpha'_j &= \alpha_j + \beta \cdot N_j(0,1) \\
\vec{x}' &= \vec{x} + \vec{N}(\vec{0}, C(\vec{\sigma}',\vec{\alpha}'))
\end{aligned}$$

where $N(0,1)$ is a vector with a mean of zero and standard deviation 1, σ and α are the standard deviation and rotation angles respectively, and $\vec{x}$ is the object variable vector.

Evolutionary strategies have several applications one of which is developed in (Lohman, 1991). This problem used a self-optimisation method to the structure of a local filter in Visual Systems.

2.3 Comparison among Algorithms

Hoffmeister and Bäck (1990) showed the similarities and differences between Genetic Algorithms (GAs) and Evolution Strategies (ESs). Table 2.2 shows a table adapted from (Hoffmeister and Bäck, 1990). Nevertheless, Bäck et al. (1991) concluded that the most significant difference between these two algorithms is the *two-level learning* in ESs.

ES	GA
Real valued representation of individuals	Binary coding for the representation of individuals
Knowledge of the number of variables in the objective function	Knowledge of objective function's properties
(μ, λ), $(\mu+\lambda)$ selection	Proportional, ranking, tournament selection
Mutation is the main operator	Recombination (crossover) is the main operator
Recombination schemes as secondary role	Mutation as secondary role

Table 2.2 GA and ES differences.

Fogel (1991) after running and comparing a GA to an EP concluded that in some cases the GA was unable to find solutions close to the optimal while the EP was more accurate. EP is robust to multiple optima and random noise because of the Gaussian relation between the parents and the children (offspring). This relation guarantees that every point in the space could be reached, allowing the algorithm to throw appropriate solutions. Additionally, EP has a very efficient parallel search mechanism. It is important to notice that even though EP is a very versatile tool computationally speaking in a serial computer the cost increases compared to that in a parallel-processing machine; understanding computational cost as the complexity of the overall system.

EP can be used as a procedure to search in general non-linear functions. As in the case of ES and GA comparison, in EP the representation of the decision variables is a real-valued vector while in GA is a binary string or Gray-coded string. It is important not to forget that in the first approaches of EP the representation was through finite-state machines.

In the ES and EP case one characteristic in common is that the evolutionary operators (mutation and crossover) are used variable by variable. Besides both methods use strategy parameters in order to provide more freedom to the search algorithm (variance and covariance along with other decision variables) and a Gaussian random variable is added to each parent involved in the genetic operator's procedures.

Finally, it is important to mention that in GA it is possible to use direct real-valued vectors, and are called real-valued GA. For this kind of GA it is necessary to modify the genetic operators as is discussed in Chapter Three.

2.4 Traveling Salesman Problem (TSP)

The origins of TSP date from the 1920's when some mathematicians and economists (Applegate *et al.*) proposed it. TSP became more known due to being a prototype of a hard problem in combinatorial optimisation. Because of this it has been studied in research operations, linear programming, and genetic algorithms as well.

To specify the Traveling Salesman Problem the following graph theoretical definitions are necessary (Reinelt, 1994). An undirected graph (or graph) $G = (V,E)$ consists of a finite set of vertexes[1] V and a finite set of edges E. Each edge $e = \{a,b\}$ or $e = ab$ has two endvertexes a,b. This graph is called undirected because there is no difference between the edges $e_1 = \{a,b\}$ and $e_2 = \{b,a\}$. It is said that edge e is incident to a and to b if $e = \{a,b\}$. The number of edges incidents to a vertex a is defined as $\delta(a)$.

[1] **Nodes or points**

A graph G' = (V',E') is a subgraph of G = (V,E) if V'⊆V and E'⊆E. A graph G(V,E) is complete if for all $a,b \in V \mid a \neq b \; \exists e = (a,b) \in E$. The complete graph of n vertexes is denoted by $Y_n=(V_n,E_n)$ and $V_n=\{1,2,\ldots n\}$. If a set of edges $P = \{ e_1(a_1a_2), e_2(a_2a_3), \ldots , e_{k-1}(a_{k-1}a_k)\}$ is given this is called a walk or $[a_1,a_k]$-walk. If $a_i \neq a_j$ for all $i \neq j$ then the set P is called path or $[a_1,a_k]$-path. A cycle is a set of edges $C= e_1(a_1a_2), e_2(a_2a_3), \ldots , e_{k-1}(a_{k-1}a_k), e_k(a_ka_1)$ where $a_i \neq a_j$ for $i \neq j$. $|C|$ denotes the length of a cycle.

The graph G = (V,E) is connected if it contains for every pair of vertexes (a,b) a path that connect them. A Hamiltonian cycle (Hamiltonian tour) is a simple cycle of length $|C| = n$ in a graph of n vertexes. For every edge $e = \{a,b\} \in E$ a weight function is defined as follows $C : E \rightarrow Q$ (Q is the set of rational numbers) for the edge $e = \{a,b\}$ its weight will be denoted by $C(e)$ or C_{ab} . A weight of a set of edges Z where $Z \subseteq E$ is defined as follows:

$$C(Z) := \sum_{e \in Z} C(e) \quad . \tag{2.23}$$

If the weight of a tour is considered as the length of the tour, the shortest tour will be the one with the shortest length. The TSP can be defined in a formal way:

Defining H as the set of all the Hamiltonian cycles of a graph:

π : TSP
Parameters: $G =(V,E)$ with $W : E \rightarrow R^+$

Answer:
Precondition: G is complete and finite
$TSP_A(G,i) \rightarrow t$
Postcondition:
$t \in H \mid \forall r \in H, C(t) \leq C(r)$
(2.24)

The aim of the Symmetric Traveling Salesman Problem (STSP) is to find the shortest Hamiltonian cycle of a complete graph Y_n with a weight $C(e)$ in every edge $e = \{a,b\} \in Y_n$. The Euclidean TSP is a set of points in the plane where the graph contains a vertex for every point and the weights are defined as the Euclidean distance between every pair of vertexes.

Describing the TSP in other words a salesman has to visit different cities in a certain tour; the constrains of this problem are:

1. There is a starting point
2. Each city is visited once.
3. The cost of travel between the cities is given.
4. The distance between city A and city B is the same distance between city B and city A. Considering the distance directly proportional to cost, travelling from A to B costs as much as travelling from B to A (symmetric).

The problem is to find the tour with the minimum total cost. To solve the problem the integer vector representation is chosen instead of using the binary string representation; it will help with the application of crossover and mutation operators. A vector $\mathbf{v} = \langle i_1 i_2 \ldots i_n \rangle$ represents a tour: from i_1 to i_2, etc. The tour is a Hamiltonian cycle that is, a cycle passing through all the vertices of the minimum total weight.

In the TSP the search space is composed of permutations of the different cities to visit. This condition increases the complexity of the problem. The TSP is an NP-Hard problem, which

means that the algorithms to compute this problem will require a computational time exponential in n, where n is the number of cities. To solve the TSP different optimisation and search methods have been used in addition to GA, evolution strategies and other methodologies. The TSP problem solved using GA needs to develop different crossover and mutation operators.

2.4.1 Traditional Optimisation and Search Methods for the TSP

To solve the Traveling Salesman Problem different heuristic methods have been developed. Some of these methods are called construction procedures (Reinelt, 1994). It is necessary to consider the following: the complete undirected graph will be called K_n, the edge weight or length are C_{ab} for every pair of vertexes a and b. The main objective is to find good Hamiltonian tours.

The first procedure to study is nearest neighbour heuristics, in this is necessary to set an initial city. This city is the starting point of the tour, and the next city is the nearest city to it. Then the salesman visits the nearest city not visited yet, until he has visited all the cities and returns to the initial city. This procedure run in a time complexity $O(n^2)$.

The second procedure is insertion heuristics, in this a subtour is chosen to start. This subtour has a size l where $l \geq 1$, then it is necessary to insert those cities that are not part of the subtour. This insertion of cities is based on different criteria, such as nearest insertion, farthest insertion, cheapest insertion, random insertion, largest sum insertion and smallest sum insertion. Comparing these procedures the better results were farthest insertion and random insertion (Reinelt, 1994).

The third procedure is one based on a greedy algorithm given the set of edges E $=\{e_1, e_2, \ldots, e_m\}$ where $m = n(n-1)/2$ and n is the number of cities. The paths are considered to have a length of zero and then the procedure checks if the shortest edge that has not been considered can join two paths. In other words, the procedure executes a sort of E from the shortest to the largest. Then it checks m times if adding the edge e_i still has a Hamiltonian tour where $i \leq m$ and it varies from 1 to m. If this is possible the edge e_i will be added to the tour.

These construction procedures are considered as algorithms with a moderate performance. This is the reason to improve some of them by means of doing alterations to the tours. It is necessary to consider a current tour to do the movements (alterations) on it.

Turning to the node insertion procedure it is necessary to start from the basis that a node (vertex) from the current tour will be moved and will be reinserted in a new position that should be the best. This means that this insertion will result in a decrease in the tour total distance (tour length). This procedure has a time complexity of $O(n^2)$ because it is necessary to check every possible insertion in every node. A failure will appear when any improvement can be found.

The edge insertion procedure is similar to the node insertion but instead of inserting a node an edge is inserted. This means an edge is removed from the current tour and it is reinserted in the best position to decrease the most the tour length. It is important to note that the edge can be inserted in two possible ways; one is inserting it conserving the vertex order and the other is switching the position of the vertexes in the edge. This procedure has a time complexity of $O(n^2)$. These procedures were applied after the construction methods described above (Reinelt, 1994). However, the results were very poor. In some cases the tour found was twice as long as the shortest tour found in the construction procedures. For this reason, other methods were developed; these methods are: 2-Opt Exchange; 3-Opt Heuristic; Lin-Kernighan; and are described below:

2-Opt Exchange

This procedure is the elimination of two edges in a tour and the connection of the two resulting paths in a different way to form a new tour. A 2-Opt move involves the edges between the vertex that is analysed and the vertex that follows it in the current tour. Then it checks if it is possible to reduce the tour. This procedure is run for all the vertexes in the initial tour, choosing the best of all the new tours generated. Reinelt (1994) ran this procedure using the construction techniques to generate the initial tour and then compared the result obtained with the results of the insertion heuristics. The results of the 2-Opt were much better than the results with the other methodologies. The time complexity for this procedure is $O(n^2)$.

3-Opt Heuristic

This procedure is an improvement of the 2-Opt heuristic, the main functions is to separate the tour in three parts and combine the paths in the best way possible. Considering different types of 3-Opt moves, this implies a time complexity of $O(n^3)$, which makes this procedure very time consuming. The next procedure is applied to all the vertexes in the initial tour. This means to all the cities in the problem (n cities) but a new constraint must be defined. With a set of vertexes $N(x)$ with a constant cardinality and independent of the number of cities n, the constraint is that once the three edges are eliminated they have at least one endvertex in $N(x)$. After this the best tour of all the tours generated will be chosen.

Lin-Kernighan Heuristics

Lin and Kernighan (1973) developed this heuristic method. They considered that if a great modification was applied to the tour but it was composed by simple moves, these were not always decreasing the tour length but they could increase the possibilities of improvement the results. Having an initial tour and running the procedure for every vertex in the tour; the main objective is to find a move that improves the tour. This move consists of 2-Opt and node insertion moves (called sub-moves). Comparing this method with 2-Opt, 3-Opt node insertion and edge insertion, the Lin-Kernighan methods had better results.

2.4.2 Crossovers for the TSP

Goldberg (1989) proposed a simple crossover that consisted in generating an integer number randomly between 1 and the string length less one. Considering two strings P_1 and P_2 with 7 bits and the random selected integer $a = 3$ the crossover yields:

$$P_1 = 1\ 0\ 1 \mid 0\ 1\ 0\ 1$$
$$P_2 = 1\ 1\ 1 \mid 1\ 0\ 1\ 0$$

The new strings:

$$P'_1 = 1\ 0\ 1\ 1\ 0\ 1\ 0$$
$$P'_2 = 1\ 1\ 1\ 0\ 1\ 0\ 1$$

Using this crossover in the TSP, two parents P_1 and P_2 are assumed with 5 elements (cities) and an integer $a = 3$ the crossover yields:

$$P_1 = (3\ 1\ 2|\ 5\ 4)$$
$$P_2 = (2\ 5\ 4|\ 3\ 1)$$

The strings are illegal tours:

$$P'_1 = (3\ 1\ 2\ 3\ 1)$$
$$P'_2 = (2\ 5\ 4\ 5\ 4)$$

In conclusion, the simple crossover is not appropriate in the TSP solution. First of all it is necessary to determine the representation used in this problem as well as the genetic operators.

In genetic algorithms as was mentioned before the typical way to represent chromosomes is using binary vectors. In some cases this representation is not enough and other representation methods have been used considering the syntax of the chromosomes as an important component of the genetic algorithm (Goldberg, 1989).

2.4.2.1 Adjacency, Ordinal, Path Representation and their Crossovers

In the GA community there is an agreement that binary representation is not valid in the TSP (Michalewicz, 1994). To solve this problem many vector representations have been developed; the three most known are adjacency, ordinal, and path representations (Michalewicz, 1994). Grefenstette *et al.* (1985) studied the adjacency and the ordinal representations.

In adjacency representation it is necessary to use a repair algorithm, because it does not support the normal crossover. The tour is a list of cities and each tour has only one adjacency representation. Three crossover operators were defined (Michalewicz, 1994):

1. Alternating edges. This crossover generates an offspring choosing an edge from the first parent then selects another edge from the second parent and alternating parents, taking care not to select an edge that introduces a premature cycle.

2. Subtour-chunks. This crossover generates an offspring choosing a subtour with random length from one parent and choosing another subtour from another parent with random length as well.

3. Heuristic crossover. This crossover generates an offspring choosing randomly a city and setting this city as the starting point. Then the two edges of this city are compared and the shorter is selected as the better.

In ordinal representation the classical crossover works. In this representation there is a list A of n cities that serves as a reference point and a list R of references, for example:

$$A = (2\ 1\ 3\ 4)$$

The list of references:

$$R = (1\ 3\ 2\ 1)$$

- The first number on the list R is 1, it takes the first city on the list A that is **2**, this city is removed from list A. The list A now is $A = (1\ 3\ 4)$
- The next number on the list R is 3, it takes the third element on the list A that is **4**, this city is removed from list A. The list A now is $A = (1\ 3)$

- The next number on the list R is 2, it takes the second element on the list A that is **3**, this city is removed from list A. The list A now is $A = (1)$
- The next number on the list R is 1, it takes the first element on the list A that is **1**,

The tour is:

$$T = (2 \quad 4 \quad 3 \quad 1)$$

Some experimental results show that this representation is not appropriate for the TSP (Michalewicz, 1994).

In path representation three crossovers have been defined: PMX, OX, CX.

PMX (partially-mapped). Goldberg and Lingle (1985) pay special attention to the crossover operation in evolutionary search. They defined this crossover with two crossing sites (randomly selected), as is shown in the next example (Goldberg, 1989):

$$A = 8\ 9\ |5\ 3\ 4|\ 6\ 1\ 7\ 2$$
$$B = 9\ 4\ |6\ 7\ 1|\ 2\ 5\ 3\ 8$$

PMX maps from A to string B and from string B to string A. The 5 and the 6, the 3 and the 7, and the 4 and the 1, exchange places. The 6 an the 5, the 7 and the 3, the 1 and the 4, exchange places, yielding the following offspring:

$$A' = 8\ 9\ 6\ 7\ 1\ 5\ 4\ 3\ 2$$
$$B' = 9\ 1\ 5\ 3\ 4\ 2\ 6\ 7\ 8$$

OX (order). The order crossover works in a similar way than to PMX (Goldberg, 1989). Two crossing points were defined as is shown in the next example:

$$A = 8\ 9\ |5\ 3\ 4|\ 6\ 1\ 7\ 2$$
$$B = 9\ 4\ |6\ 7\ 1|\ 2\ 5\ 3\ 8$$

OX maps from A to B leaving holes in cities 6, 7, and 1 (represented by H). Mapping from B to A will leave holes in cities 5, 3, and 4 holes (represented by H):

$$A = 8\ 9\ 5\ 3\ 4\ H\ H\ H\ 2$$
$$B = 9\ H\ 6\ 7\ 1\ 2\ H\ H\ 8$$

The holes are filled with a sliding motion that starts from the second crossing site:

$$A = 5\ 3\ 4\ H\ H\ H\ 2\ 8\ 9$$
$$B = 6\ 7\ 1\ H\ H\ H\ 2\ 8\ 9$$

The Hs are filled in the case of A with cities from B and in the case of B with cities from A, these cities are between the two crossing sites. For A the cities are 6, 7, and 1 for B cities are 5, 3, and 4:

$$A' = 5\ 3\ 4\ 6\ 7\ 1\ 2\ 8\ 9$$
$$B' = 6\ 7\ 1\ 5\ 3\ 4\ 2\ 8\ 9$$

CX (cycle). Oliver *et al.* (1987) proposed this crossover. The main constraint of this crossover is that each city in the offspring comes from one of the parents. Assume two parents A and B (this example was taken from (Michalewicz, 1994):

$A = 1\ 2\ 3\ 4\ 5\ 6\ 7\ 8\ 9$

$B = 4\ 1\ 2\ 8\ 7\ 6\ 9\ 3\ 5$

The first city from parent A is taken to begin the first offspring:

$A' = 1\ \_\ \_\ \_\ \_\ \_\ \_\ \_\ \_$

The next city is city number 4, because it is the first city in parent B, city 4 appears in the fourth place in parent A:

$A' = 1\ \_\ \_\ 4\ \_\ \_\ \_\ \_\ \_$

City 8 is the city that appears in fourth place in parent B and is in eighth place in parent A:

$A' = 1\ \_\ \_\ 4\ \_\ \_\ \_\ 8\ \_$

Continuing this process, the next cities to place are 3 and 2. City number 2 requires the selection of city 1, this city is already placed in the offspring and the cycle is completed:

$A' = 1\ 2\ 3\ 4\ \_\ \_\ \_\ 8\ \_$

The next city to consider is the first city that appears in parent B and does not appear in the offspring (A'). This city is 7 and is placed in the fifth position of the offspring:

$A' = 1\ 2\ 3\ 4\ 7\ \_\ \_\ 8\ \_$

This city 7 appears in the fifth place of parent B and the city that appears in this place in parent A is 5. The city 5 is in ninth place of parent B and in this place in parent A appears city 9. This city is in seventh place in parent B as it is placed in the offspring as follows:

$A' = 1\ 2\ 3\ 4\ 7\ \_\ 9\ 8\ 5$

City 9 is related to city 7 and this city is already part of the offspring. Then a second cycle is completed. The only city that has not been placed is city 6. The offspring is as follows:

$A' = 1\ 2\ 3\ 4\ 7\ 6\ 9\ 8\ 5$

The second offspring is obtained following the same procedure:

$B' = 4\ 1\ 2\ 8\ 5\ 6\ 7\ 3\ 9$

Oliver *et al.* (1987) analysed and compared the PMX, OX and CX crossovers solving the TSP with 30 cities. It is shown that the crossover with the best performance is the OX.

All the representations mentioned above are appropriate to GAs. In most of the cases it is necessary to use a repair algorithm to allow the proper function of crossover.

2.4.2.2 The Edge Recombination Crossover (ER)

Whitley *et al.* (1991) developed a crossover called the edge recombination crossover (ER). This crossover transfers more than 95% of the edges from the parents to a single offspring while the adjacency representation transfers just 30 % of the edges from the parents (Grefenstette *et al.*, 1985).
The process is to create a list of edges; this list has the edges of each city in both parents. This means that each city can have at least two cities and at most four cities. For example, for the two parents:

$$P_1 = (2 \quad 3 \quad 4 \quad 1 \quad 5)$$
$$P_2 = (1 \quad 4 \quad 3 \quad 5 \quad 2)$$

The edge list is:

city 1: edges to other cities: 2,4,5
city 2: edges to other cities: 1,3,5
city 3: edges to other cities: 2,4,5
city 4: edges to other cities: 1,3
city 5: edges to other cities: 1,2,3

To construct the offspring the number of edges of each city is analysed. The first step is to select the city with the least edges. In the example above this is city 4; this city has cities 1 and 3 connected to it. Cities 1 and 3 have three edges, which means a random selection is needed between these two cities. Suppose that city 3 is chosen, then the offspring is as follows:

$$O = (4 \quad 3 \quad \ldots \quad \ldots \quad \ldots)$$

Analysing city 3, it is connected to cities 2, 4, and 5. City 4 having been selected then cities 2 and 5 can be examined. Each city has three edges, then a random selection is made. Assuming city 2 has been selected it has edges 1, 3, and 5. Since city 3 is already in the offspring, cities 1 and 5 will be analysed. Both cities have 3 edges. A random selection is applied again, and city 5 is selected. City 5 has edges 1, 2, and 3. Cities 2 and 3 are part of the offspring, then city 1 is selected.

The offspring is:

$$O = (4 \quad 3 \quad 2 \quad 5 \quad 1)$$

Additionally the concept of failure has to be introduced. A failure occurs when a city has been chosen and it does not have any city remaining in its edge list. In this case the operator chooses another city and this is introduced in the offspring continuing with the procedure mentioned above.

Whitley *et al.* (1991) presented the results for the 30 city problem comparing them with the results found by Oliver *et al.* (1987) obtaining a better performance for the edge recombination crossover. Additionally, they used this crossover to solve sequencing problems due to the good solutions found for the TSP. The results of this experiment are shown in subsection 2.5.1.

Starkweather *et al.* (1991) developed an improved version of the edge recombination crossover. In this version, an adjacency table is constructed in the same form as in the edge recombination crossover but with the difference that if a city is adjacent to another in both parents it will be represented in the adjacency table with a negative sign. For example, for the two parents used previously:

$$P_1 = (2 \quad 3 \quad 4 \quad 1 \quad 5)$$
$$P_2 = (1 \quad 4 \quad 3 \quad 5 \quad 2)$$

The edge list is:

city 1: edges to other cities: 2,-4,5
city 2: edges to other cities: 1,3,-5
city 3: edges to other cities: 2,-4,5
city 4: edges to other cities: -1,-3
city 5: edges to other cities: 1,-2,3

Selecting the first city of the offspring randomly and analysing the edge list of this city the next city of the offspring will be determined. When following a scheme of priorities, the first priority is to place the flagged cities in the offspring and the cities with a lower number of edges in their lists will have second priority. Once the cities are placed in the offspring they have to be removed from the edge lists. When none of the priorities mentioned above could be followed to determine the next city to place in the offspring it will be determined randomly. This edge recombination operator version was used in small TSPs. Mathias and Whitley (1992) called this improved version as Edge-2 recombination operator and they introduced a new version called Edge-3 recombination operator that was designed with a guarding failure mechanism. This version is based on the construction of partial tour or subtours. The city that is at the beginning or at the end of a partial tour is called terminal. A terminal can have two states, to live or die; a live terminal exists when its edge list still has edges; otherwise it is a dead terminal. The algorithm of the Edge-3 recombination operator, considering that the edge table has being done previously, is shown below:

Edge-3 Algorithm

Algorithm (int *i*, *j*, *live*)

```
c1 ← randomcity (parentA);      /* randomcity selects randomly a city from parent A */
live ← 0;
i = 2;
while (live = 0 && i < n+1)             /*  n is the number of cities of the tour */
{
      ci ← buildtour (edgelist, ci-1);      /* buildtour selects the edge from the edgelist of ci-1
                                               following the priority conditions */
      if (edges(edgelist, ci) = 0) then     /* edges counts the number of city that are in the
                                               edgelist */
            live ← 1;
      i ← i + 1;
}
if (i < n) then
{
      live ← 0;
      j ← i;
      while (live = 0 && j < n + 1)
      {
```

```
            c' ← inverse (c_l, c_i);          /* inverts the partial tour from c_i to c_l */
            c_j' ← buildtour (edgelist, c_j-1');
            if (edges(edgelist, c_j') = 0) then
            live ← 1;
            j ← j + 1;
        }
}
```

When two failures occur, the Edge-3 algorithm continues adding cities to the tour in a random form. The Edge-4 recombination operator was proposed by Dzubera and Whitley (1994), and follows the same idea of Edge-3 recombination operator. The main difference is that once the first and the final cities have suffered a failure, instead of reversing all the tour in the offspring, it will reverse a part of it. Considering the following example:

5 8 10 6 1 4 3 2

suppose that city 2 and city 5 are dead and the city 2 edge list is 4, 6 and 8. Then Edge-4 recombination operator takes the adjacent cities to city 2 edge list:

5 8 10 6 1 4 3 2

These cities are city 10, city 1 and city 3, then the one with the least number of cities in its edge list will be chosen. Lets suppose this is city 6 then the subtour from city 6 to city 2 will be reversed:

5 8 10 2 3 4 1 6

allowing the procedure to continue with city 6.

Comparing the results using Edge-3 and Edge-4 recombination crossovers, it is found that Edge-4 consumes greater resources than Edge-3. However, Edge-4 has a better performance than Edge-3.

Tang and Leng (1994) proposed a new variant of Edge-2 called edge recombination nearest-neighbour (EdgeNN). It is necessary to generate a matrix that contains the distances among the cities. Figure 2.4 shows the distance matrix:

	1	**2**	**3**	**4**	**...**	***n***
1	—	C_{12}	C_{13}	C_{14}	...	C_{1n}
2	C_{21}	—	C_{23}	C_{24}	...	C_{2n}
3	C_{31}	C_{32}	—	C_{34}	...	C_{3n}
4	C_{41}	C_{42}	C_{43}	—	...	C_{4n}
⋮	⋮	⋮	⋮	⋮	—	⋮
n	C_{n1}	C_{n2}	C_{n3}	C_{n4}	...	—

Figure 2.4 Distance matrix

where C_{ij} is the distance between city i and city j. If the problem is symmetric $C_{ij} = C_{ji}$.

The algorithm of the EdgeNN recombination operator is shown below (the initialisation of the population P is generated randomly and $S_i, S_j \in P$, $i \neq j$):

EdgeNN recombination crossover algorithm

```
Algorithm (int k, int t)
t ← 0;
parent1 ← S_i;
parent2 ← S_j;
length ← n/4;
k ← n - length;
point ← randompoint(parent1, n, h);
offspring ← segment (parent1, point, length);
newparent ← parent1- segment;
map ← edgemap(parent2, newparent);
c ← lastcity(offspring);
while (t < k)
{
    map ← remove(c, map);
    elements ← edgelist(map,c);
    if (elements ≠ 0) then
    {
        if (negative ≠ 0) then
        {
            c ← selectcity(c, map);
            offspring ← offspring add c;
        }
        else
            c ← nearest(matrix, c);
            offspring ← offspring add c;
    }
    else
    {
        if(t < k) then
        c ← nearest(matrix, c);
    }
    t ← t + 1;
}
```

where:

randompoint	:	*tour* → *city*	randomly select a city
segment	:	*tour* → *segment*	copy a tour segment from parent 1 beginning from the city selected
edgemap	:	*tour, segment* → *edge list*	build the list of edges for each city
remove	:	*edge list* → *edge list*	remove all occurrences
Edgelist	:	*edge list* → *int*	calculate the number of cities that *c* (current city) has in its edge list
selectcity	:	*edge list* → *city*	select a city randomly
nearest	:	*matrix* → *city*	determine the nearest city to *c* in its edge list

Tang and Leng (1994) report very good results for the EdgeNN compared to Edge-2. However, the Edge-2 is a blind recombination operator, so it was decided to use Edge-3 with a 2-Repair procedure (variation of the 2-Opt) to have a more balanced comparison. The results of this comparison were better for the 2-Repair-Edge-3 hybrid operator. The authors of this method concluded that this is useful to generate the initial population using local hill-climbing procedures. All the methods explained above are known as the Edge recombination family.

2.4.2.3 The Maximal Preservative Crossover

Mühlenbein (1991) developed the maximal preservative crossover (MPX). The aim of this crossover is to preserve subtours contained in the parents. MPX uses two parents, one of them is called the donor and the other is called the receiver. Two points of crossover are defined randomly and they are applied to the donor parent. The string of edges between these two points is extracted; this string is called the crossover string. The crossover string is copied to the offspring and the next cities are added to the offspring in a consecutive form that follows certain rules. The procedure to follow is described as follows:

1. The last city in the offspring is taken and this city is called X.
2. It is verified if in the receiver there is another city that follows city X and it has not been placed in the offspring yet. If it exists it is placed in the offspring and goes back to step 1.
3. If there is not another city in the receiver that follows city X, it is verified in the donor. If in the donor there is a city that follows city X it is placed in the offspring and goes back to step 1.
4. If there is not another city in the donor that follows city X, the city that comes consecutively in the receiver parent is placed in the offspring and goes back to step 1.

This procedure is followed until all the cities from the parents are placed in the offspring. The following example shows how this procedure works:

$A = 1\ 4\ 3\ 5\ 6\ 7\ 8\ 9\ 2$ *donor*
$B = 8\ 2\ 7\ 6\ 9\ 4\ 1\ 3\ 5$ *receiver*

A and B are the parents, defining A as the donor parent and B as the receiver parent and choosing the crossover point in 2 and 6:

$A = 1\ 4\ |3\ 5\ 6\ 7|\ 8\ 9\ 2$

Copying the crossover string to the offspring:

$O = 3\ 5\ 6\ 7\ \_\ \_\ \_\ \_\ \_$

Following the procedure, the last city in the offspring is city 7. City 7 in the receiver is followed by city 6 and if this city is already part of the offspring then it is necessary to check in the donor parent. In the donor the city that follows city 7 is city 8. City 8 is placed in the offspring and it goes back to step 1:

$O = 3\ 5\ 6\ 7\ 8\ \_\ \_\ \_\ \_$

Now the last city in the offspring is city 8. City 8 in the receiver is followed by city 2. If city 2 is not in the offspring then it is placed in it and it goes back to step 1:

$O = 3\ 5\ 6\ 7\ 8\ 2\ \_\ \_\ \_$

The last city in the offspring now is city 2. City 2 in the receiver is followed by city 7. If city 7 is already placed in the offspring then it proceeds to verify in the donor. If in the donor parent city 2 does not have any city that follows it then it is necessary to apply step 4. The next city in the receiver is city 9 and this is placed in the offspring and goes back to step 1:

$O = 3\ 5\ 6\ 7\ 8\ 2\ 9\ \_\ \_$

The last city in the offspring is city 9. City 9 in the receiver is followed by city 4. City 4 is placed in the offspring and it goes back to step1:

$O = 3\ 5\ 6\ 7\ 8\ 2\ 9\ 4\ \_$

The last city in the offspring now is city 4. City 4 in the receiver is followed by city 1. City 1 is placed in the offspring and the procedure is finished having completed the offspring:

$O = 3\ 5\ 6\ 7\ 8\ 2\ 9\ 4\ 1$

Mühlenbein defined that the length of the string crossover should be determined randomly while Ulder *et al.* (1990) proposed that this length should not be longer than 1/3 of the number of cities of the problem.

2.4.2.4 The Complete Subtour Exchange Crossover

Katayama *et al.* (1998) proposed the *complete subtour exchange crossover* (CSEX). This crossover is based on the enumeration of all common subtours. To enumerate the subtours they developed an algorithm which consumes O(n) time where n is the number of cities in the problem. The process consists in finding the common subtours from two parents considering that they must have the same direction or the opposite direction. For example, for the two parents:

$P_A = (1\ \ 4\ \ \overrightarrow{2\ \ 3}\ \ 5\ \ 6\ \ \overrightarrow{9\ \ 8\ \ 7})$
$P_B = (\overleftarrow{3\ \ 2}\ \ 1\ \ 5\ \ 4\ \ 6\ \ \overleftarrow{7\ \ 8\ \ 9})$

The common subtours are **2 3** and **9 8 7** in parent A, and **3 2** and **7 8 9** in parent B because they have the opposite direction. Having located the common subtours the next step is to generate the offspring:

$P_A = (1\ \ 4\ \ [2\ \ 3]\ \ 5\ \ 6\ \ [9\ \ 8\ \ 7])$
$P_B = ([3\ \ 2]\ \ 1\ \ 5\ \ 4\ \ 6\ \ [7\ \ 8\ \ 9])$

$O_1 = (1\ \ 4\ \ [3\ \ 2]\ \ 5\ \ 6\ \ [9\ \ 8\ \ 7])$
$O_2 = (1\ \ 4\ \ [2\ \ 3]\ \ 5\ \ 6\ \ [7\ \ 8\ \ 9])$
$O_3 = (1\ \ 4\ \ [3\ \ 2]\ \ 5\ \ 6\ \ [7\ \ 8\ \ 9])$
$O_4 = ([3\ \ 2]\ \ 1\ \ 5\ \ 4\ \ 6\ \ [9\ \ 8\ \ 7])$
$O_5 = ([2\ \ 3]\ \ 1\ \ 5\ \ 4\ \ 6\ \ [7\ \ 8\ \ 9])$
$O_6 = ([2\ \ 3]\ \ 1\ \ 5\ \ 4\ \ 6\ \ [9\ \ 8\ \ 7])$

Katayama *et al.* (1998) defined the number of common subtours as k and the maximum number of offspring generated as $2 \times 2^k - 2$. The algorithm to determine the number of common subtours satisfies the following statement:

$$\{ \pi_A(i) \mid i \in [l_A, r_A] \} = \{ \pi_B(i) \mid i \in [l_B, r_B] \} \tag{2.25}$$

Where π_A and π_B are the permutations and $[l_A, r_A]$ and $[l_B, r_B]$ are the subtour sections.
The procedure of interchanging the common subtours in the parents to generate the offspring is called "2-swap". The CSEX selects the best offspring among all those generated by the 2-swap procedure. To know the performance of CSEX Katayama *et al.* (1998) compare it with edge recombination crossover (IERX) and typical heuristics like nearest neighbor (NN), 2-opt (LS) and 2-Opt with nearest neighbor (NN +LS). It is necessary to define the concept of quality, which is the percentage excess from the optimal value:

$$quality = \frac{Fitness - Optimal}{Optimal} \times 100(\%) \tag{2.26}$$

In Table 2.3 the results of the comparison are shown; this table was adapted from Katayama *et al.* (1998).

NN	**quality (%) min.**	**quality (%) max.**	**Quality (%) Avg.**	**Cpu Time (sec) avg.**
ei151	13.8	20.9	18.0	0.00
kroA100	20.5	26.7	25.0	0.01
LS	**quality (%) min.**	**quality (%) max.**	**Quality (%) Avg.**	**Cpu Time (sec) avg.**
ei151	1.2	13.3	6.1	0.11
kroA100	5.8	14.1	10.3	0.61
NN + LS	**quality (%) min.**	**quality (%) max.**	**Quality (%) Avg.**	**Cpu Time (sec) avg.**
ei151	2.1	8.0	5.1	0.10
kroA100	4.0	9.2	7.9	0.40
GA (CSEX)	**quality (%) min.**	**quality (%) max.**	**Quality (%) Avg.**	**Cpu Time (sec) avg.**
ei151	2.1	5.4	3.8	5356
kroA100	3.2	7.0	5.5	11838
GA (IERX)	**quality (%) min.**	**quality (%) max.**	**Quality (%) Avg.**	**Cpu Time (sec) avg.**
ei151	4.5	7.5	6.2	5026
kroA100	8.5	44.7	21.9	16049

Table 2.3 Comparison of crossover operator (GA) with heuristics.

It is obvious that the CSEX had a better performance than the other operators did even though it consumes more computation time than all the others except for the IERX. It was considered as well to make a hybrid (HGA) using genetic algorithms and a local search method as 2-Opt. Table 2.4 shows the results of these HGA using MPX, IERX and CSEX. This table was adapted from Katayama *et al.* (1998); the TSP instance used is kroA100.

Algorithms	quality (%) min.	quality (%) max.	Quality (%) Avg.	Cpu Time (sec) avg.
MPX	0.0	0.287	0.066	2015
IERX	0.0	0.301	0.033	1727
CSEX	0.0	0.301	0.085	1091

Table 2.4 Comparison of crossover operator in a HGA.

2.4.2.5 The Inver-over Operator

Tao and Michalewicz (1998) developed an evolutionary algorithm using the inver-over operator. This algorithm has two main features: it possesses a strong pressure in selection and uses an adaptive operator. The algorithm of the inver-over operator is shown below (the initialisation of the population P is generated randomly):

Inver-over operator Algorithm

Algorithm (int k, float p)

```
k ← 0;
while (unchanged < k)
{
    ∀ Si ∈P do
        offspring ← Si ;
        c ← randomcity (offspring);
        end ← 0;
        repeat
        {
            if (random( ) ≤ p) then
        c' ← randomcity (offspring – {c});
        else
        {
            Sj ← randompop (P – { Si});
            c' ← nextcity (Sj, c);
        }
        if (c' = nextcity (offspring, c) or c' = prevcity (offspring, c)
            end ←0;
        else
        {
            offspring ← inverse (nextcity (offspring, c), c', offspring);
            c ← c';
        }
} until (end = 1)
if (eval (offspring) ≤ eval (Si))
{
            Si ← offspring;
            k ← 0;
        }
        else
            k ← k+1;
}
```

where:

randomcity	:	*tour* → *city*	randomly select a city

nextcity	:	*tour* × *city* → *city*	select the next city
prevcity	:	*tour* × *city* → *city*	select the previous city
inverse	:	*city, city* → *subtour*	place the elements of the subtour in reverse order
eval	:	*tour* → *fitness*	calculate the total distance of the tour

To test the algorithm the following values were used: population size r =100, probability p = 0.02 and the termination condition *unchanged* = 10 (the best solution did not change the last 10 iterations).

The optimal solutions of the test cases were known (TSPLIB, 1995); the test cases were EIL30, EIL51, EIL76, EIL101, ST70, KROA100, KROC100, KROD100, LIN105, CHN144, PCB442, PR2392 and RAN10000. Tao and Michalewicz (1998) presented the results after ten runs and gave an average value of the results on them comparing this value with the optimal solution, an average computational time in seconds, the total number of inversion and the total number of iterations. In the cases of EIL30, EIL51, EIL76, ST70, KROA100, KROC100, KROD100 and LIN105 the optimal was found in all the runs of the algorithm. In EIL101 the algorithm failed once in ten runs, the optimum value was 629 and the average value was 629.2. In the case of 144 cities (CHN144) the average value was above the optimum 0.04%, with 442 cities (PCB442) the average was above the optimum 0.63%, and for the 2392 cities case the average was above the optimum 2.66% and the average computational time was 5366.23 seconds (1.49 hrs). From the random case with 10000 cities the average was above the optimum 3.56% and the average computational time was 167501 seconds (46.52 hrs.).

It can be concluded that a good feature of this algorithm is that it combines mutation (inversion) and crossover operators in one and it has three parameters: population size, probability of random inversion and the number of iterations (the number of times the while loop is processed).

2.4.3 Other Methodologies to Solve the Traveling Salesman Problem

2.4.3.1 Genetic Local Search for the Traveling Salesman Problem

Ulder *et al.* (1990) discussed how the speed of classical Local Search algorithms could be increased using genetic algorithms in their trials to solve the TSP. A Local Search technique consists of a certain kind of iterative heuristics. The 2-Opt, the Lin-Kernighan (1973) and the k-Opt algorithms are well known Local Search algorithms to solve the TSP. Having analysed all the solutions proposed by different authors developed to solve the problem, they summarised the procedure as follows: 1. Initialise. This is the construction of the initial population. 2. Improve. Replace each solution from the initial population with a better solution using a Local Search algorithm. The best method of Local Search is the one proposed by Lin and Kernighan (1973). 3. Recombine. This main objective is to recombine solutions to add new solutions extending the current population. 4. Improvement. The use of a Local Search technique to improve again the solutions. 5. Selection. Following rules previously described select the best solutions. 6. Evolution. This consists of the repetition of steps from 3 to 5. The repetition procedure will be practised until a termination condition is reached. The multi-start Local Search algorithm consists of a Local Search algorithm that is repeated several times and it retains the best local solution found.

The performance of multi-start Local Search (Mult2-Opt and MultLK) is compared to the performance of the Genetic Local Search algorithms (Gen2-Opt and GenLK). In the improvement stage Gen2-Opt used the Local Search 2-Opt and in GenLK the Local Search Lin-Kernighan is used. For the recombination stage two tours are selected randomly and a

subtour of one of them is chosen; this subtour contains at most one third of all the cities in the complete tour.

The selection stage consists of collecting the best tours once the recombination stage is performed. The termination condition is achieved when all the tours in the population have the same fitness or when the fitness of the best tour stops having an improvement after five successive generations. For the experiments different instances of the TSP already known (GRO48, TOM57, EUR100, GRO120, LIN318, GRO442, GRO532, GRO666) were used. Then the average of five times running each algorithm is compared to the optimum solution. In all the cases the GenLK algorithm had the best solutions having an average deviation from the optimum of 0.11%, Gen2-Opt had an average deviation of 1.84%, MultLK had an average deviation of 0.28% and Mult2-Opt had an average deviation of 5.39%. The algorithms that used the Lin-Kernighan method are more effective than the others are, but it is important to notice that even though they are good the best one is the Genetic Local Search algorithm (GenLK).

2.4.3.2 Evolution Strategies for the Traveling Salesman Problem

Herdy (1991) presented a method to solve the TSP using evolution strategies. As it has been mentioned above these strategies use two operators mutation and recombination. The mutation operators used are four: 1. Reciprocal exchange, 2. Inversion of a segment of the tour, 3. Insertion of a city at another point of the tour, 4. Displacement of a tour segment.

1. Reciprocal exchange. This operator selects two cities randomly and they interchange their places with each other. Having the following parent *A* suppose that the cities 4 and 10 were selected randomly:

   $A = 2\ (4)\ 6\ 7\ 1\ 9\ 3\ 8\ (10)\ 5$

   the final offspring *O* after apply the operator is:

   $O = 2\ (10)\ 6\ 7\ 1\ 9\ 3\ 8\ (4)\ 5$

2. Inversion of a segment of a tour. This operator reverses the order of the cities in a subtour. Having parent *A* suppose that the subtour selected was from city 7 to city 3:

   $A = 2\ 4\ 6\ [7\ 1\ 9\ 3]\ 8\ 10\ 5$

   the offspring *O* is:

   $O = 2\ 4\ 6\ [3\ 9\ 1\ 7]\ 8\ 10\ 5$

3. Insertion of a city at another point of the tour. This operator selects randomly a city to be inserted between two other cities that are placed consecutively. Having parent *A* suppose that the city selected is city 1 and it should be inserted between city 3 and 8:

   $A = 2\ 4\ 6\ 7\ \underline{1}\ 9\ 3\ 8\ 10\ 5$

   the final offspring is:

   $O = 2\ 4\ 6\ 7\ 9\ 3\ \underline{1}\ 8\ 10\ 5$

4. Displacement of a tour segment. In this operator a subtour is selected and this is displaced by a certain number of cities. Having parent A suppose that the subtour selected is from city 3 to city 10 and is displaced four places to the left:

$$A = 2\ 4\ 6\ 7\ 1\ 9\ \boxed{3\ 8\ 10}\ 5$$

the offspring is:

$$O = 2\ 4\ \boxed{3\ 8\ 10}\ 6\ 7\ 1\ 9\ 5$$

However, in this evolution strategy to solve the TSP not only the mutation operation is applied but also a recombination operator. This operator only produces valid tours and to achieve this condition heuristic and deterministic components are used.

2.4.3.3 Evolutionary Programming for the Traveling Salesman Problem

In order to solve the TSP using evolutionary programming (EP) instead of evolving finite state machines it is necessary to evolve a population of tours (Fogel, 1991) and a technique called probabilistic survival is implemented. This technique consists of placing each tour of the population in a competition against the 10% of the remaining tours. Each individual of the population (tour) has a probability of winning defined as follows; having two competing parents A and B, where each parent has a tour length C_A and C_B respectively, the "win" probability of A P_{Awin} is:

$$P_{Awin} = \frac{C_B}{C_A + C_B} \tag{2.27}$$

and the "win" probability of B P_{Bwin} is:

$$P_{Bwin} = \frac{C_A}{C_A + C_B} \tag{2.28}$$

For each encounter it is necessary to calculate the probability of winning of each involved tour. Each tour has a counter associated; this counter is the number of winnings of that tour. The tours with the biggest number of winnings are part of the offspring and will be used as parents in the next generation. It is important to note that in EP the genetic operator used is only mutation. Fogel (1991) implemented this procedure in TSP of 25, 50, 75, 100, 125 and 150 cities obtaining efficient and effective results.

2.5 Applications of the TSP

Reinelt (1994) explains some of the possible applications of the Traveling Salesman Problem. Some of these applications will be described below.

The drilling problem for printed circuit boards (PCBs): This problem consists of drilling the holes that are needed to connect the pins of integrated circuits or conductors in a circuit board; these holes can have different diameters. If two consecutive holes have different diameters the head of the machine has to be moved, this process is pretty time consuming. Creating a set of all the holes with the same diameter and assuming that each hole can be a city this can be seeing as a symmetric TSP. Where the parameter to optimise is the time that it takes to move the head from one place or position to the others. This procedure can be applied to each different diameter creating a sequence of TSP, one for each set of holes with the same diameter.

The order-picking problem warehouse: In a warehouse a vehicle is used to collect materials. If a customer makes an order, that is a subset of all the stored materials. The objective is to optimise the tour this means to minimise the total distance of the tour or route that should be followed by the vehicle. Considering the location of each item as a vertex, this problem can be solved as a TSP.

X-ray crystallography: To obtain information about the structure of a crystalline material an x-ray diffractometer is used. In order to measure the intensity of the reflections of the crystal the detector has to be placed in different positions. The number of different positions can be quite big (30,000 positions). In this problem the main objective is to optimise the time of positioning the detector and to achieve this it is necessary to determine the best sequence for the measurements. The problem is posed as a TSP.

Finally the application of the TSP that will be studied the most is scheduling problems in sub-section 2.5.1 these applications are studied in more detail.

2.5.1 Scheduling

Cleveland and Smith (1989) used a GA to solve a scheduling problem. Job-sequencing problem can be studied as timing and a sequencing problem. If the problem is analysed as a sequencing problem it has common characteristics with the symmetric TSP. The job-sequencing problem consists of finding an order of jobs to release such that the processing cost of the jobs will be minimal. This problem has $n!$ possible solutions and so is a NP-complete problem. The problem has an overall facility composed by sectors, once a job enters a sector it has to be moved from station to station automatically, each station processes the job in a first-come, first served manner. To formulate the sequence of the problem the PMX, subtour-swap operator and the subtour-chunk operator were applied (Cleveland *et al.*, 1989). The features of the problem were a sector had five workstations and each workstation had three identical machines. The dispatch rules were:

1. Earliest due date first. This means that the most pressing contract has to be finished before it is possible to move to the next contract.

2. Shortest processing time first. This means that the selection of the jobs will be to process first the type that requires the least processing time.

3. Least slack time first. This means the difference between the due date and the total processing time of a job. The jobs will be processed in order of their urgency.

In the tests that were run, in the first case the subtour chunk operator had a slightly better result, so it was not possible to say that one operator was the best. In the second test the operator with the best result was the subtour chunk. It is possible to conclude that especially the flow shop problem can be treated as a TSP and some times it is possible to use the same crossover operator for both problems.

Whitley *et al.* (1991) developed the edge recombination operator (as it was mentioned above) and used it in sequencing problems. This was a production line scheduling problem; above all it is necessary to describe the features and constraints of the problem.

Firstly, it has six workcells in sequence and each one has two machines (identical) which operate in parallel and in an independent form. The workcells have an input that is used to receive work and an output to deliver work once it is completed. The machines have a cost associated to the kind of work they do. Lastly, there are twenty different kinds of products to be produced in a single schedule with certain set-up and processing fixed times.

Figure 2.5 shows the production line where L represents the input/output of the workcells and M represents the machines, as it is defined the workcells have two machines and A and B represent these machines respectively. L_1 represents the first job sequence.

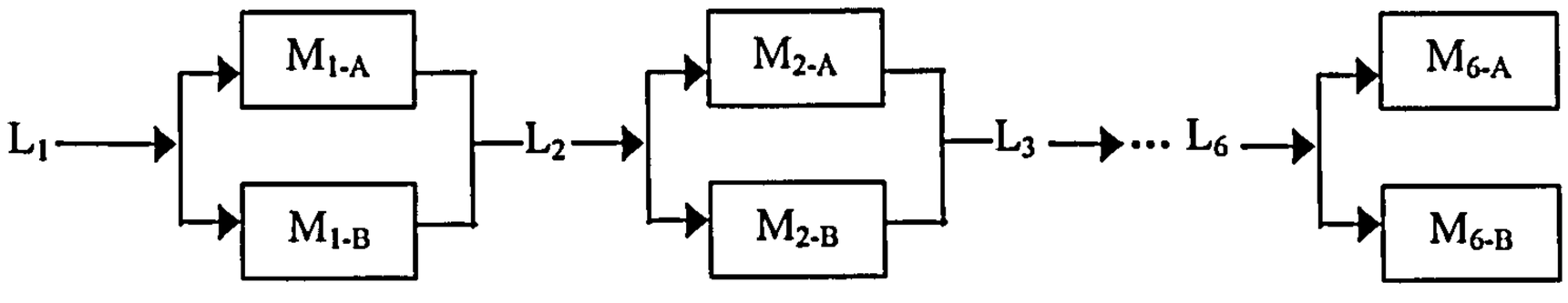


Figure 2.5 Production line layout

Another important part is the encoding of the problem. The key is to visualise the scheduling problem as sequencing one to allow the treatment of it as a TSP using the edge recombination operator. Good results were obtained not only in this problem but also in a real world scheduling application.

2.5.1.1 Solution of a Scheduling Problem using GA

Syswerda (1991) solved a scheduling problem optimiser based on GA. The main objective was to optimise a schedule. The problem was the System Integration Test Station (SITS) of the U.S Navy, where the resources to be scheduled were F-14 airframes, the flight and radar simulation environment generators, and numerous pieces of support equipment such as computers, radios and recorders. A scheduling problem appears because the laboratory can be used simultaneously by different users. Four components of the GA were proposed: chromosome syntax, chromosome interpretation, evaluation, and the genetic operators.

In the chromosome syntax a list of tasks is used, this list could be represented as the list of cities in the TSP. In order to interpret the chromosome a schedule builder is used, and as its name indicates it is used to produce legal schedules. The chromosome evaluation is made considering task priority and time preferences. In the design of a genetic algorithm the use of operators is one of the most important steps. Syswerda (1991) compares some of them in the solution of the scheduling problem. The first operator to analyse is the Random search, from the basis that the schedule builder generates good lists of tasks.

The second operator is mutation considering that this operator has two functions. It can be used in the search and it also provides diversity to the population. Three different mutation operators were studied. Position-based mutation is where two tasks are selected randomly and the second task is placed before the first one. Order-based mutation is where in the same way two tasks are selected randomly and they interchange their positions in the list of tasks. Scramble mutation considers the adjacency of tasks; in this operator a sublist of tasks is selected and the order of tasks in the sublist is scrambled. Figure 2.6 shows the performance of these operators in 3000 evaluations. It is obvious that the order-based mutation operator was superior compared to the others (this figure was adapted from Syswerda (1991)).

The third operator is crossover, and three crossover operators were considered: order-based crossover, position-based crossover and edge recombination crossover. These were explained in subsections 2.4.2.1 and 2.4.2.2. Figure 2.7 shows the performance of the crossover operators in 3000 evaluations (this figure was adapted from (Syswerda, 1991). It is obvious that the edge recombination operator was inferior compared to the others even though good results were reported by Whitley *et al.* (1991). Syswerda concluded that adjacency is not so important to the SITS problem due to the characteristics of the schedule builder.

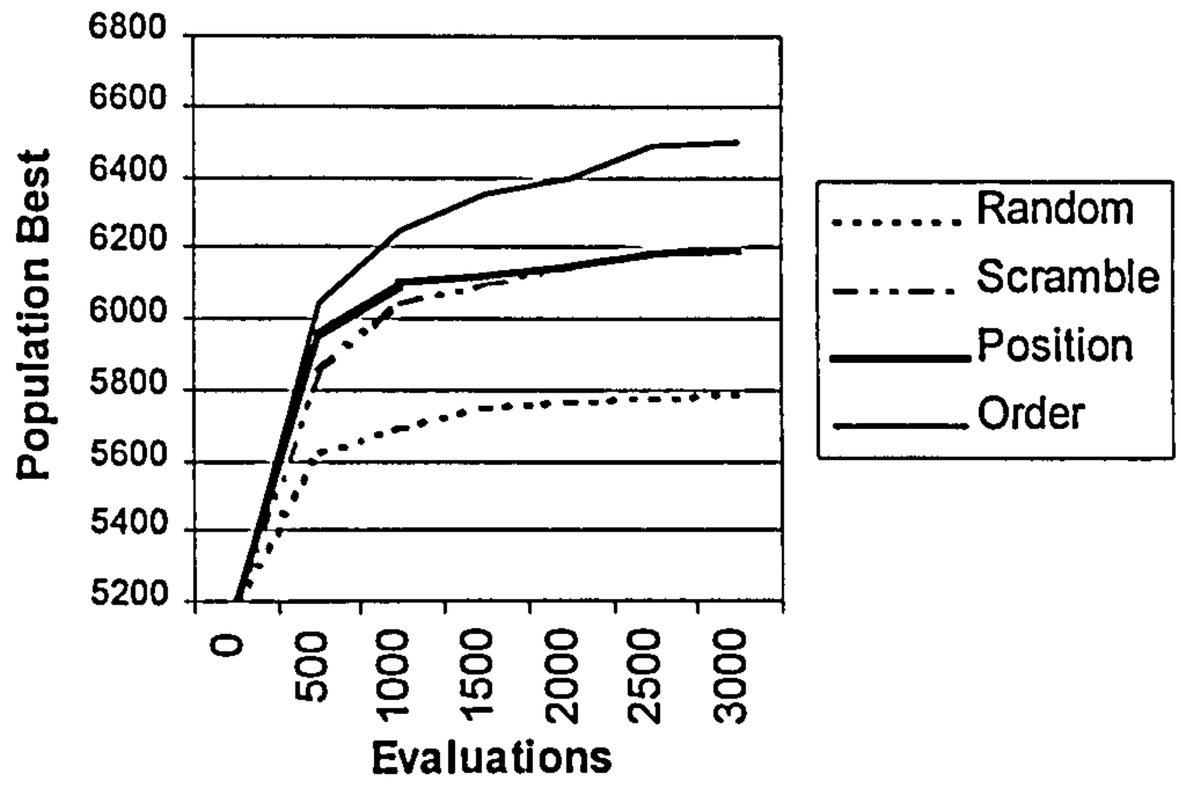


Figure 2.6 Comparison among the mutation operators and the random search

Finally, the operators are combined. In this case the order-based mutation that was the most effective is used with the three different crossovers with the same probability to being applied (this was called fixed). If Figure 2.6 and Figure 2.7 are compared it is noticed that for the first 1000 evaluations the crossover operators give better values than the mutation operators, but after this point the crossover tends to increase very slowly.
At this moment the crossover could not do anything else. As mentioned above the mutation operator could give diversity to the population. Then order-based mutation is applied after crossover operation in every run (this was called variable). Figure 2.8 shows the results of these combinations (this figure was adapted from Syswerda, 1991).

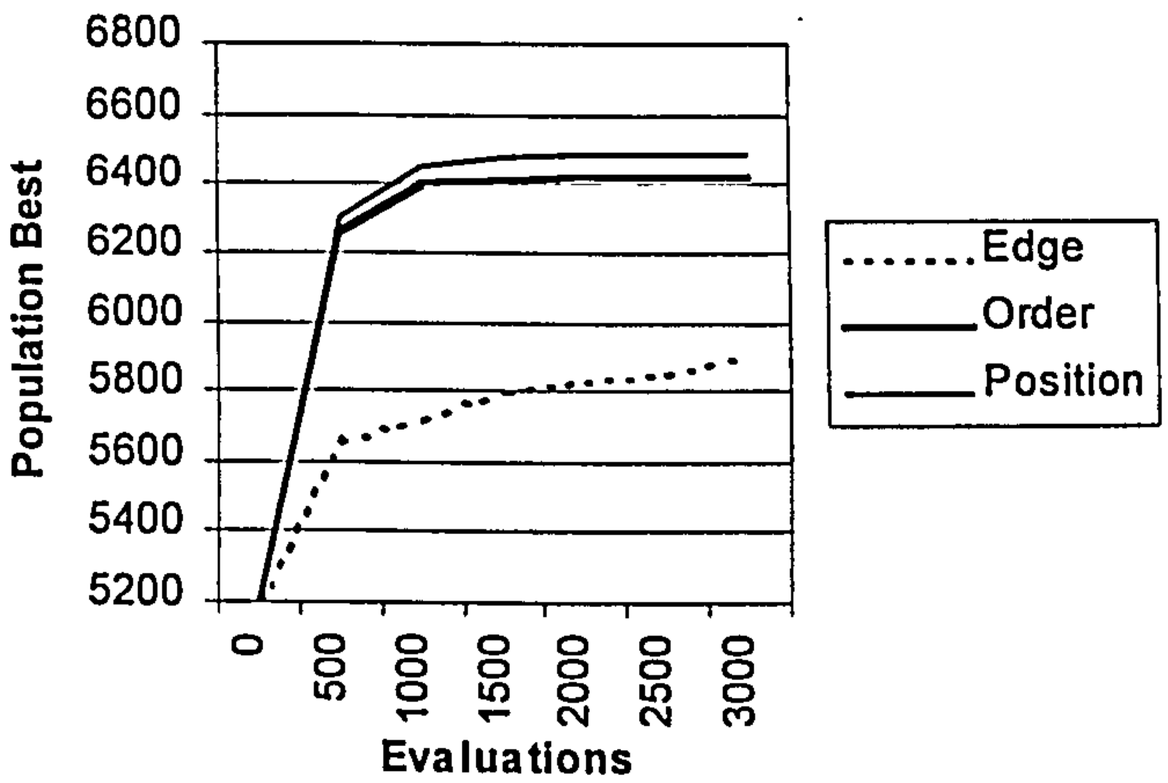


Figure 2.7 Comparison among the crossover operators

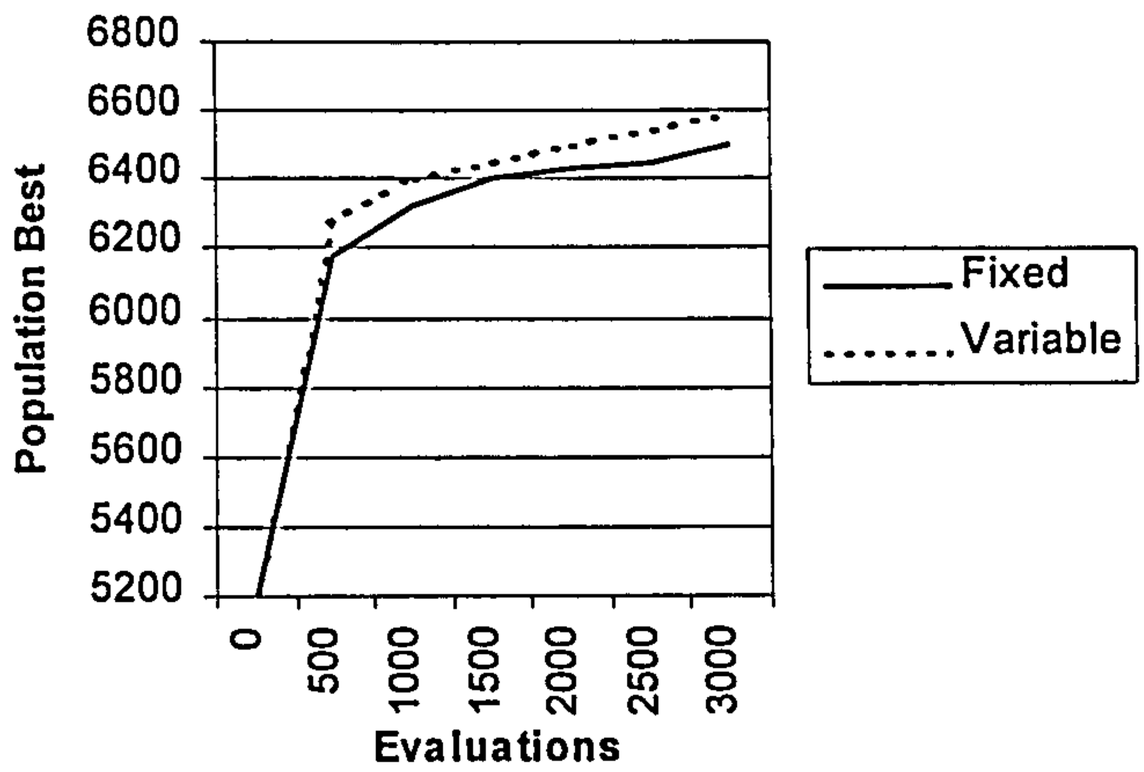


Figure 2.8 Performance of the GA using variable and fixed combinations

It can be concluded that using a GA to solve a scheduling problem will follow the steps presented by Syswerda that consist of:

1. Chromosome Syntax. This will depend on the nature of each problem and is related to chromosome representation.
2. Chromosome Interpretation. This is related to the conversion of the list of task to a schedule.
3. Chromosome Evaluation. This is related to the objective function.
4. Operators. This is one of the most important steps in the design of GA, the development of mutation and crossover operators to be used in the algorithm.

The most important point to notice is that the edge recombination crossover had very poor results in the solution of the SITS problem while in other scheduling problems it was really efficient. It can be concluded that the use of operators and their performance depend directly on the kind of problem to be solved. This point makes GAs very flexible and powerful.

2.6 Summary

This chapter presents the basic concepts of evolutionary computation, where evolutionary algorithms are defined as instances of algorithms that work with evolutionary principles. Genetic algorithms (GA), evolutionary programming (EP) and evolution strategies (ES) are subsets of evolutionary algorithms.

Generally, GAs are stochastic algorithms based on natural evolution principles, that perform a search starting from an initial population to which certain genetic operators are applied to find an optimal solution (Michalewicz, 1994). Some of the most popular selection operators, crossover operators and mutation operators of GAs have already been described.

EP is based on a finite-state machine (organism) considered as a parent. Two operators described previously are applied to this organism: selection and mutation. It has been concluded that EP has been applied successfully to problems in prediction, identification and automatic control especially in the solution of NP-complete problems.

ES are algorithms completely based on recombination, selection and mutation operators even though the first evolution strategy ((1+1)-strategy) worked just with selection and mutation operators. The representation of the parent is made using a pair of float-valued (real-valued)

vectors. Other strategies were also discussed such as (μ+λ)-ES or (μ,λ)-ES. Furthermore, the differences between the three different evolutionary algorithms were outlined. There were also outlined the bases for the understanding of GAs, which form the main search mechanism used throughout the development of this thesis.

This chapter has investigated the application of evolutionary computation in real world optimisation problems, but not all these problems can be solved by the use of such methods. In general, many approaches have been developed to find the most suitable solution for the problems.

The Traveling Salesman Problem is a combinatorial optimisation problem, which has been treated using genetic algorithms with several different crossover operators. This report has presented some of them but in recent years other crossovers have been developed (Michalewicz, 1994). It is important to notice that this problem is NP-complete and that almost all the scheduling problems are NP-complete as well. This means that the type of problem under consideration is the same.

In this chapter some applications of scheduling problems have been explained, but it is important to bear in mind that the number of such problems could be unlimited. It is also important to see that in this kind of problem, the codification of the GA's chromosomes is integer. Therefore it is shown how a GA can be represented using a different codification than binary string. Also, it is a good example of optimisation, giving a better view of the optimisation of NP-complete problems. Moreover, this chapter outlines the bases for the understanding of evolutionary computation applied to optimisation that will be used in the treatment of the case studies and problems presented in Chapters Four, Five, and Six.

In the next chapter the concepts of decision analysis, risk analysis, and multiple criteria decision-making methods are discussed.

CHAPTER 3

Multiple Criteria Decision Making

3.1 Introduction

Every day people try to solve personal, employment or social problems by making decisions; some decisions are simpler than others. The most common method of making decision is intuitive, particularly with simple decisions. Nowadays the complexity of decisions to be made especially in organisations has increased; for this reason in most cases it is impossible to make them intuitively. To solve this problem a new process emerges called *analytic decision-making*.

Analytic decision-making is part of a wider area called *decision theory* or *decision analysis*. This area involves all the concepts related to making a decision by mathematical means. The aim of decision theory is related to the choices made when solving a problem and in a certain way the selection of a course of action to obtain a solution. Harrison (1975) considers three elements in the decision analysis: the decision-making process, the decision-maker, and the decision itself. These concepts are described in more detail in section 3.2 of this chapter.

In real-world problems it is necessary to identify, quantify, evaluate and incorporate risks and uncertainties into the decision making process. These considerations will improve the process by itself but at the same time they will make it more and more complicated. To have the necessary tools to understand the incorporation of risk and uncertainty in a problem, section 3.3 has been dedicated to *risk analysis*, emphasising the discussion of risk related to decisions.

Moreover, one important objective of the decision-making problems is the search for an optimal solution (Pareto optimal solution). The area that involves this search is called multiple criteria decision-making (MCDM). The definitive boom in this area of research was in the late 1970's when several conferences and publications appeared (Zeleny, 1982). MCDM can be understood as the support system used to help the decision-maker to solve a decision problem and is explained in section 3.4. A decision problem normally includes attributes, objectives, goals and criteria (Hwang and Masud, 1979). These concepts are defined in subsection 3.2.2.

This research is focused basically on the solution of MCDM problems. For this reason in section 3.4 the most popular methods for the solution of this kind of problems are discussed. Finally, it is important to bear in mind that one of the main objectives of this research is the implementation of the decision-making process using evolutionary computation and fuzzy logic and for this reason the most important operators of multiple objective evolutionary algorithms are discussed in section 3.5.

It is very significant to remember that all the concepts and definitions presented throughout this chapter will be used in the development of this thesis.

3.2 Decision Theory

The beginning of the *decision theory* or *decision analysis* dates from the Second World War essentially with the spread of operations research theory. After that, in order to formalise the theory some basic concepts were defined. These definitions will be presented in this section to have a better understanding of the concept of decision theory or decision analysis.

Firstly, Tannenbaum (1950) defined the action to decide as "the coming to a conclusion". He also states that decision-making "involves a conscious choice or selection of one behaviour alternative from among a group of two or more behaviour alternatives". Saaty (1994) states that there are two kinds of decision: "analytical and intuitive"; if the decision to be made is very complex then an analytical solution method will be needed.

Goodwin and Wright (1998), understanding the concept of analysis as the process of dividing something into its most elemental particles, define decision analysis as a formal mechanism that helps to integrate the results of each problem after dividing the decision problem into a set of smaller problems.

Decision analysis can be very useful when a decision has to be made by a group of decision-makers (DMs) because it can give a better picture of each individual's position when facing the problem (Goodwin and Wright, 1998). However, it is not realistic to expect decision analysis to solve the problem or to give an optimal solution; it is better to consider that it will give the DM an idea of what should be done. Nevertheless, this idea could be the opposite of what the DM's intuitive feelings say.

Clemen (1991) states that the decision analysis helps the DM to have a better understanding of the problem, giving the tools required for solving it analytically. According to Moore and Thomas (1988) the decision analysis concept could be considered not only as a set of techniques but also as an approach to the decision-making process under risk and uncertainty. To perform risk analysis and represent uncertainty, mathematical tools are required; these are discussed in section 3.3.

Forgionne (1986) defines an area called "quantitative decision making", where he states that decision analysis is one part of it. He also presents some areas where quantitative decision-making has been applied, dividing them into "private institutions" and "public institutions". Particularly, the applications found in private institutions are in the following areas: finance (e.g. budgets, financial planning, management), marketing (e.g. advertising media selection, assessment of competitive marketing strategies, location of distribution facilities), and production (e.g. allocation of production resources, inventory control, production planning and scheduling). The applications in public institutions are in the following areas: health (e.g. evaluation of health care delivery systems, hospital staffing), military (e.g. missile allocation for national defence, war game simulation, weapon system analysis), social and environmental (e.g. courtroom scheduling, educational planning and scheduling, air traffic control, urban planning), and economical (e.g. forecasting economic conditions, economic development). In conclusion, it is possible to define general activities, such as allocation, planning, analysis, scheduling, control and forecast, where quantitative decision-making is applied.

Summarising, Harrison (1975) states that decision theory has to be focused on three elements: the decision-making process, the DM, and the decision itself.

1. The decision-making process is discussed in the following subsection (3.2.1).
2. The DM is the individual that recognises the problem, sets the objectives and selects the decision to be made. The DM can be one individual or a group of individuals.
3. The decision will be the process of selecting a course of action to follow.

3.2.1 Decision-Making Process

It is said that the decision-making process involves an individual and a social phenomenon. This process includes the choice of one action to follow among different alternatives (Harrison, 1975). The choice is made by means of an evaluation process that considers the DM's expectations. In other words the objective is to find the alternative that better attains the DM's objectives.

Gregory (1988) states that the decision-making process has two main aspects: objective and subjective. The objective aspect can be covered with decision-making quantitative methods. The subjective aspect can be classified as psychological. He recommends that the aspects should not be separated at the moment of making the decision although most of the methods presented in literature are basically mathematical. In this research the methods used are analytical and the main focus is on how the DM's preferences can be converted and represented in a mathematical way. Chapters Four and Five show how two different methods can be applied to solve MCDM, taking into account the DM. In Chapter Six the focus has to be put on the psychological way of understanding and representing the DM's desires and preferences, solving a real problem involving the optimisation of the allocation of lecturers' activities.

Forgionne (1986) defines the decision-making process in a very accurate way because he considers both types of information to be qualitative and quantitative. The DM observes and recognises a problem to solve, identifies alternative courses of action and gathers the information (qualitative and quantitative) to evaluate the alternatives according to the defined criteria (Figure 3.1). Once the courses of action are evaluated the DM selects the alternative that he or she considers the best or most preferable and implements it.

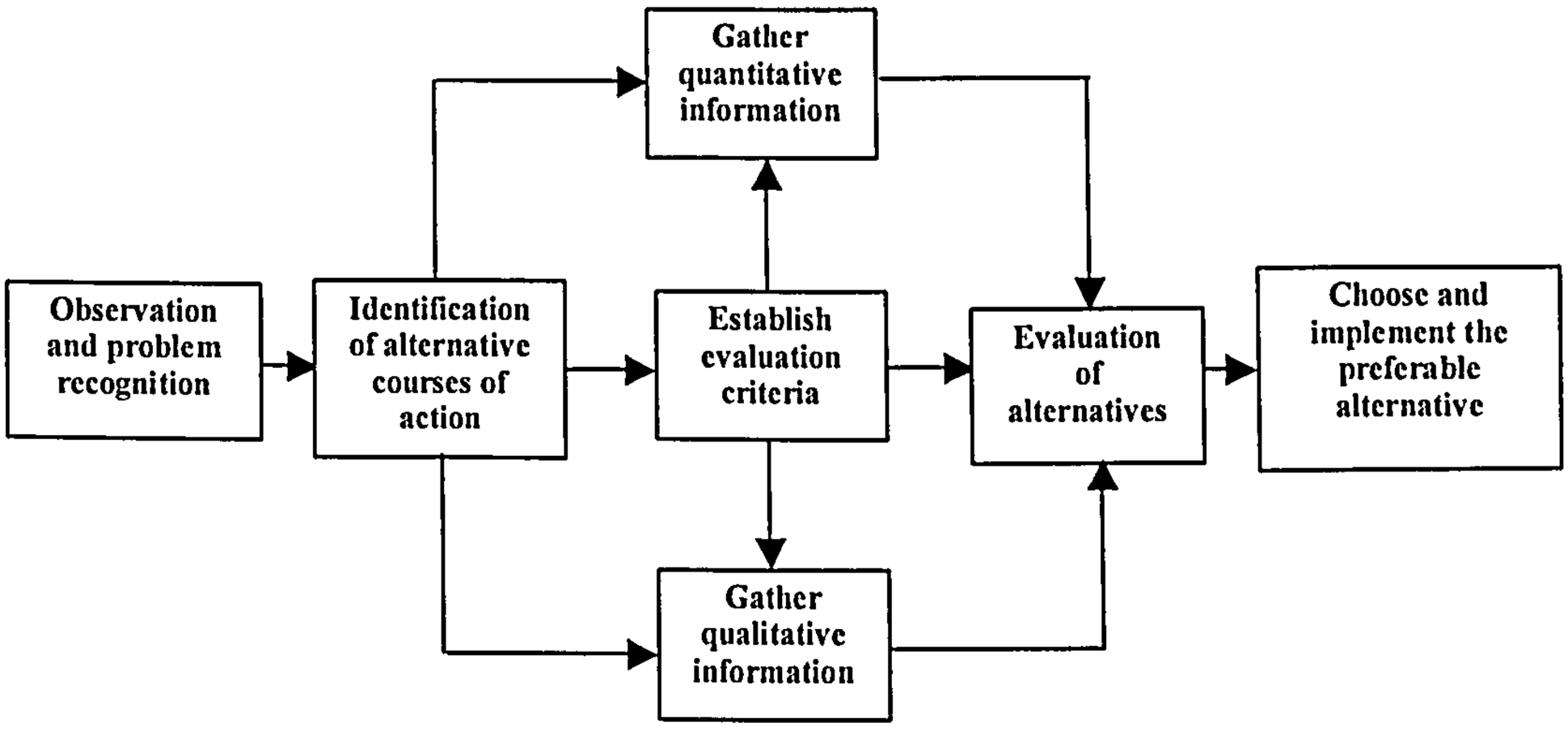


Figure 3.1 Decision-making process
Adapted from Forgionne (1986)

Gore et al. (1992) define the decision process as all the activities carried out to make a decision, and not only the moment of decision. They proposed the decision process shown in Figure 3.2.

In an ideal situation it is expected that the DM can choose freely from a set of alternatives but in reality this is not possible because of some factors that limit or restrict the selection. These factors are called constraints and can be classified. Tannenbaum (1950) presented five different types of constraints. It is important to bear in mind that his main objective was to describe the managerial decision-making process and in a certain way his classification is oriented in this sense. This constraint classification is presented as follows:

1. *Authoritative* constraints. This kind of constraint is of human type because it is defined by an individual and is imposed on another individual, normally following a hierarchy defined in an enterprise (manager- subordinate relation).
2. *Biological* constraints. These constraints can be either permanent or temporary (changeable). They are directly related to the individual's behaviour. It is said that they are changeable when the individual does not have the knowledge of an activity but can learn it in a defined period of time. It is said that a biological constraint is permanent when it is not possible to change its nature ("human beings cannot fly") (Tannenbaum, 1950).
3. *Physical* constraints. These constraints have to do with the physical environment (physical laws, climate, geography, etc) and they are present at every moment.
4. *Technological* constraints. These constraints depend on the technological advances and resources available at the moment the decision is made.
5. *Economic* constraints. These constraints are related to optimisation (maximisation) and depend on economic forces as well as economic resources.

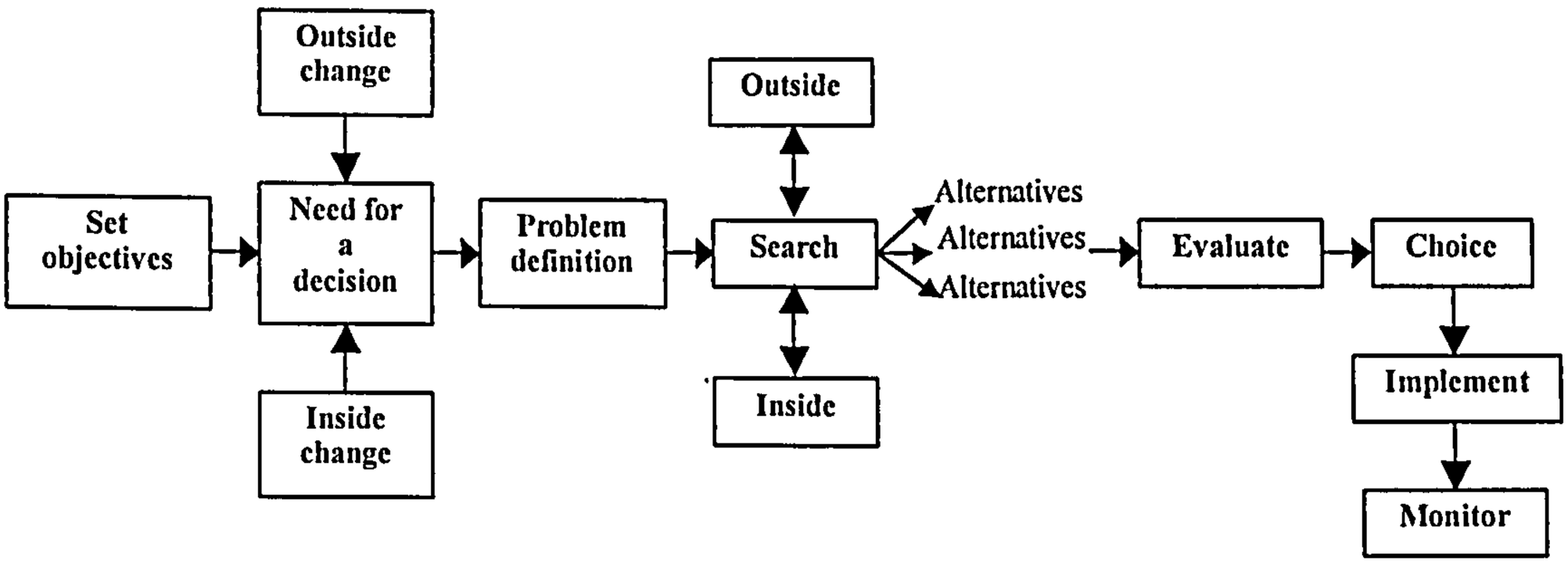


Figure 3.2 Decision process
Adapted from Gore (1992)

After the concept of constraint is determined it is possible to understand that the alternative set will be narrowed due to the fact that the alternatives that do not meet the constraints will be discarded. Finally, Tannenbaum (1950) defined decision-making as "the judgement exercised within constraints".

In the next subsection the terms: objectives, attributes, goals and criteria will be discussed, due to the fact that they are an essential part in the decision-making process.

3.2.2 Objectives, Attributes, Goals and Criteria

It is important to consider that a universal definition of objectives, attributes, goals and criteria terms does not exist, and therefore most of the authors define these in an informal manner. In this chapter, more suitable definitions of these terms, with respect to this research, are considered.

Objectives. Keeney and Raiffa (1976) define an objective as the element that gives the direction to follow in order to achieve a better outcome. Ackoff (1978) presents a more simple definition where he states that an objective is a 'desired outcome'. Hwang and Masud (1979) as Keeney and Raiffa, define an objective as the direction in which it is expected "to do better" but they also include the DM's perception. In other words, the definition of objective considered for purposes of this research will be the direction to follow to find a better outcome as perceived by the DM.

Attributes. Hwang and Masud (1979) define attributes as the "characteristics, qualities or performance parameters" used to describe alternatives. The alternatives normally represent a set of available actions to be followed or chosen by the DM. According to Keeney and Raiffa (1976) the attributes are those used to measure the objectives as well as the scalar quantities that are used to measure the attributes. French (1986) defines the attributes as the dimensions used to represent the alternatives and states that the way the DM perceives the problem is key for their representation. Triantaphyllou (2000) considered the set of alternatives to be finite.

French (1986) states that an objective has basically two components, a dimension and an indication of the "good and bad". With this definition, and considering that an attribute is a dimension used to represent alternatives, it is possible to conclude that an objective and an attribute always go hand-in-hand. Moreover, in real world problems, it is necessary to deal with several objectives or attributes that are usually in conflict. These kinds of problem are called *multiple objective decision-making* (MODM) problems and *multiple attribute decision-making* (MADM) problems and they are studied in the next section.

Goals. A goal is something desired by the DM that helps to clearly identify a level of achievement or a target. To determine this definition both Keeney and Raiffa (1976) and Hwang and Masud (1979) were considered.

Criteria. Criteria indicate attributes and/or objectives (Hwang and Masud, 1979). Consequently, the term multiple criteria decision-making (MCDM) involves either multiple objectives or multiple attributes.

For example, in the case of a plant manager an objective can be "to minimise cost of production", then the attribute will be determined in sterling pounds and can be "thousands of sterling pounds spend in a month". A goal for this case can be "to spend a maximum of 20,000 sterling pounds in a month".

As the main objective of decision theory is the solution of problems, it is essential to bear in mind that one of the most important aspects of problem solving is the complete understanding and definition of the objectives. Keeney and Raiffa (1976) suggested that in the solution of a problem it is necessary to specify the objectives and attributes following a generation and selection process respectively.

In the objective generation process two different processes can be considered. The first one consists of specifying an overall objective that gives the DM a global panorama of the problem; this objective can be considered as a starting point to the further definition of more specific objectives (*low-level* objectives). The whole process consists mainly of three steps:

1. Looking for similar problems in literature; it is possible that some of the objectives in problems already solved are close to those of the problem to solve.
2. Building an analytical model of the problem based on the identification of input and output variables.
3. Observing people to analyse the way they are making decisions related to the problem.

The second objective generation process consists of generating low-level objectives and then using them to generate an overall objective or more general objectives. A technique suggested is the use of surveys asking people which objectives they consider should be included in the problem's model. For this process it is recommended to have a group of experts for the identification of objectives.

Basically, in the attribute selection process the attributes are related to the information given to the DM. This information has to be useful to the decision-making process; for this reason an attribute has two main characteristics, which are *comprehensiveness* and *measurability*

(Keeney and Raiffa, 1976). An attribute is comprehensive if it gives the DM the information needed and it is measurable if it allows assessment of the objective. Consequently, if an attribute is going to be selected by the DM it must have both characteristics. In some cases the objectives cannot use a scalar quantity to be measured, therefore a subjective index has to be defined.

It has been said that the aim of solving a problem is to choose one alternative from a set of alternatives. Now it is necessary to establish that in most of the problems the DM wants to optimise one or several objectives. To have a better understanding of the optimisation concept it is necessary to define an *optimum*. Optimising is the process of identifying the *best* alternative. The best alternative will then be called the *optimum* or *optimal choice* (French, 1986).

Finally, in the real world most decisions are based to some extent on uncertain forecasts and the appearance of risk. In the next section the area of risk management related to decision-making is presented.

3.3 Risk Management

Baker et al. (1999) define *risk management* as "the procedure that consists of risk analysis, evaluation and control" although these steps are further divided into five: risk identification, analysis, evaluation, response and monitoring. Using this risk management classification, they propose a risk management life cycle, shown in Figure 3.3.

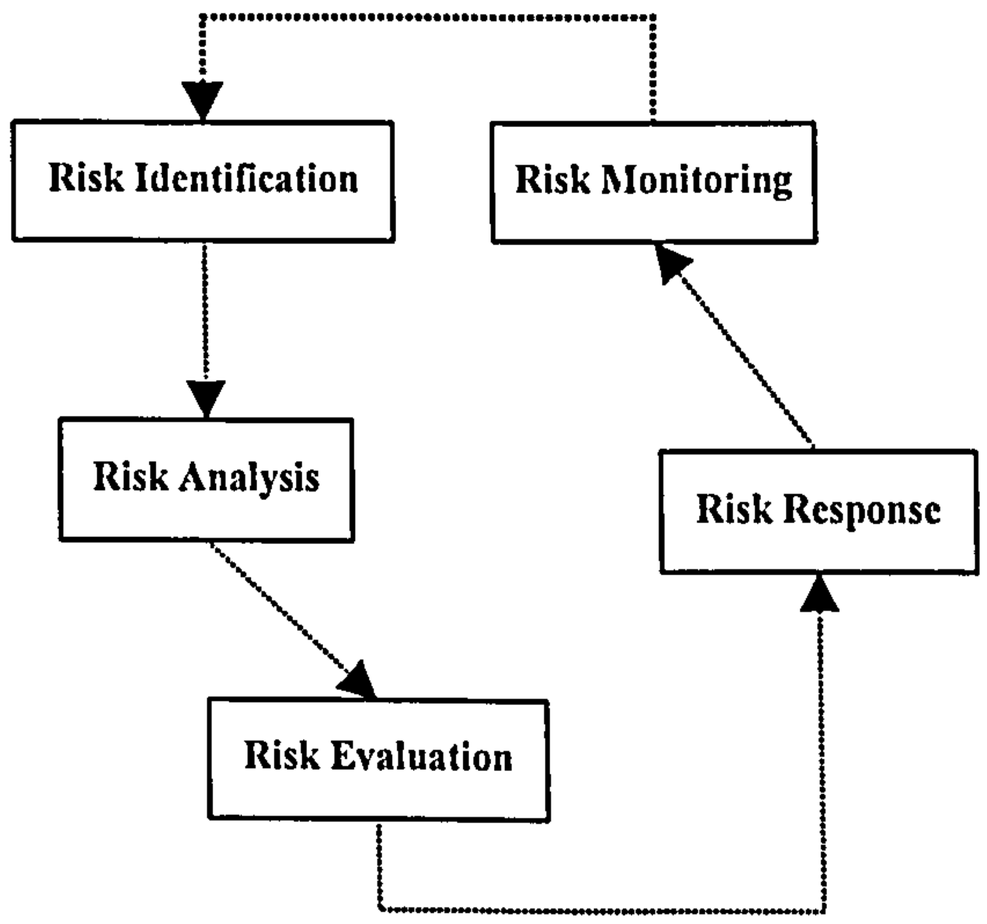


Figure 3.3 Risk Management Life Cycle
Adapted from Baker et al. (1999)

According to Haimes (1998), risk management is distinguished from *risk assessment* although some authors address the whole process of assessment and management as risk management. Considering both risk assessment and risk management, as independent processes different questions have to be asked. In the risk assessment process the following questions have to be answered:
"What can go wrong? What is the likelihood that it would go wrong? What are the consequences?" (Haimes, 1998). Having posed these questions, the risk analyst can identify, measure, quantify and evaluate the risk. Risk management will be constructed from the risk assessment and some other kinds of questions will be asked:
"What can be done? What options are available, and what are their associated trade-offs in terms of all costs, benefits, and risks? What are the impacts of current management decisions on future options?" (Haimes, 1998).

Finally, there are four sources of failures that have to be considered in the process: hardware, software, organisational, and human failures.

First of all to discuss *risk analysis* it is necessary to understand the difference between the concepts of risk and uncertainty. The main distinction between risk and uncertainty consists in the nature of the problem outcomes. If the outcomes can be described using probability distributions then it is said to be a situation of risk, otherwise it is a situation of uncertainty.

The main focus of this chapter is to describe all the tools that the DM can use to obtain the big picture of a problem including assessing uncertainty and the possible outcomes.

Hertz and Thomas (1983) define a logical sequence of steps for handling risk:

1. Risk identification. It is necessary to make a diagnosis of the problem and its structure. In this step the manager has to identify possible scenarios like the best, the most likely and the worst scenario. This will help to understand the impact of risk in future events.
2. Risk measurement. This step has to do with the assessment of situations that involve risk. These risk situations have to be classified, including a judgement about how high the situation's risk is.
3. Risk evaluation and re-evaluation. In this part of the process the manager can perform the *sensitivity analysis* and can estimate the correctness of the risk classification. The manager is expected to make a judgement about the decision selected, for example if a project will be adopted or not.

Haimes (1998) defines risk assessment and risk management processes as a whole process with the following five steps:

1. Risk identification. This step, considered the first and most important step in the risk assessment process, is understood as the identification of the nature of risk and the sources of failure. These sources could be related to elements such as environmental, technological, political, and economic aspects.
2. Risk quantification and measurement. This step consists of assessing the failure factors using probabilities and determining the relationship between the sources of risk and their impacts. In this step the input-output relationships with the system variables and the objective functions are also quantified.
3. Risk evaluation. In this step the link between the assessment and the management process is made by means of generating and measuring the trade-offs in terms of cost, benefits and risks.
4. Risk acceptance and avoidance. In this step the DM plays a very important role due to the fact that he or she evaluates the impact of current decisions on future options.
5. Risk management. This is the step where the decisions are executed or implemented.

Haimes (1998) suggests that these five steps will form a loop or a cycle where they (steps) can be performed several times.

When risk management is discussed it is necessary to speak about models. A mathematical model will be understood as a set of equations that explains and represents the real problem or system. Moreover, to generate mathematical models different types of variables will be used. These variables can be classified as follows:

1. Decision variables. These are controlled by the DM and normally are represented by a vector of decision variables $\mathbf{x} = (x_1, x_2, \ldots, x_n)$.
2. Input variables. These are not necessarily controlled by the DM but by the individual parties involved in the system.

3. Exogenous variables. These are related to external factors that can affect the system and they are exclusive of decision and inputs variables.
4. Random variables. Random events or processes that can be represented by probability distributions (risk) or are completely uncertain introduce these variables.
5. State variables. These represent the quality and quantity levels of the system.
6. Output variables. These are normally represented in terms of the decision variables, state variables, and sometimes the random variables.

3.3.1 Sensitivity Analysis

In this subsection the sensitivity analysis will be discussed. As mentioned in section 3.3 this analysis belongs to the risk evaluation stage (Hertz and Thomas, 1983). Frey and Patil (2002) state that the sensitivity analysis can be useful to verify and to validate a model, to identify critical control points, and to prioritise data.

Frey and Patil (2002) classify the sensitivity analysis methods in three categories:

1. Mathematical. These methods assess how sensitive the output of a model is to variations in the input values.
2. Statistical. These methods involve running simulations where the inputs are described by probability distributions and the main objective is to assess the effect of the inputs (variance) on the output distribution.
3. Graphical. These methods are characterised by the representation of sensitivity using graphs, charts, or surfaces. They can be used as a complement to statistical and mathematical methods.

Frey and Patil (2002) compare 10 different sensitivity analysis methods, analysing their advantages and disadvantages:

1. Nominal range sensitivity, also known as local sensitivity analysis or threshold analysis, is used to solve deterministic models. This method is most validly used with linear models. The sensitivity is represented by percentages that measure the positive or negative change to the nominal solutions.

2. Difference in log-odds ratio (ΔLOR) is used in deterministic models where the output is a probability. Odds are the ratio of the probability that an event occurs P to the probability that the event does not occur (1 - P), odd ratio = $P/(1-P)$. After the ratios are calculated a logarithm is applied to them as shown in equation 3.1:

$$\Delta LOR = \log\left[\frac{\Pr(event|\text{ with changes in input})}{\Pr(No\ event|\text{ with changes in input})}\right] - \log\left[\frac{\Pr(event\ |\text{ without changes})}{\Pr(No\ event|\text{ without changes})}\right] \quad (3.1)$$

   The value of ΔLOR can be positive or negative. The greater the value of $|\Delta LOR|$ the greater the influence of the input over the output.

3. Break-even analysis is considered a concept, whose main objective is "to evaluate the robustness of a decision to changes in inputs".

4. The automatic differentiation technique (AD) is used for calculating local sensitivities when the models are very big, and, as its name indicates, is automated. The local sensitivity (one input, some inputs) analysis method is based on the evaluation of the partial derivatives of the output with respect to the inputs selected.

5. Regression analysis can be used as a probabilistic analysis method (Frey and Patil, 2002). To use this method it is necessary to previously identify the relation between the inputs and the output. This technique is based on fitting the relationship between the inputs and the output using equations such as the one shown below:

$$Y_i = \beta_0 + \beta_1 X_{1,i} + \beta_2 X_{2,i} + \ldots + \beta_m X_{m,i} + \varepsilon_i \quad (3.2)$$

where, β_j are the regression coefficients for the jth input, $X_{j,i}$ is the ith input data point for the jth input, Y_i is the ith output data point for the ith input data point, and ε_i is the error for the ith data point.

6. Analysis of variance (ANOVA). This method determines if there is a "statistical association between an output and one or more inputs" (Frey and Patil, 2002). In this method the inputs are called *factors* and the values of these inputs are called *factor levels*. The output is called *response variable*. There are two types of ANOVA: *single-factor* that studies the effect of one factor on the response variable, and *multifactor* that studies the effect of interactions between factors.

7. Response surface method (RSM). This method is used to represent the relation between one or more inputs and the output (response variable); it can be applied to any deterministic model but also can be used in probabilistic analysis. It is better to identify the most important inputs using a method like normal range sensitivity analysis and having them identify curvatures in the response surface to find higher-order effects.

8. Fourier amplitude sensitivity test (FAST). This method can be also used for uncertainty analysis. FAST is used to calculate the expected values and variance of the output and how the inputs affect this variance. The output variance is evaluated by using Fourier coefficients, then the contributions of the inputs to the total variance is calculated using these coefficients, fundamental frequency and higher harmonics of the frequency. This method is better used in probabilistic models.

9. Mutual information index (MII). Considering that each input can give information about the output, the aim of this method is to provide a measure of this information. After the measures are provided it is necessary to compare them to determine which inputs are giving useful information about the output. This method can be used in systems with continuous outputs and it is very complex to programme.

10. Scatter plots. This is a graphical sensitivity analysis method and for this reason is used for visual assessment of the effects of each input on the output. This method can be also used as a guide to select the appropriate sensitivity analysis method to use in a certain model. It is mainly applied to probabilistic models.

3.3.2 Risk Management Applications

Baker et al. (1999) performed a comparative investigation to obtain the risk management practices conducted by the oil and gas industry and the construction industry in the United Kingdom. In the first stage of this research, 100 construction companies and 27 oil companies (Baker et al., 1999) were selected. These companies were asked to answer a questionnaire; only 52 replied. Once the questionnaire was applied, it was possible to analyse different aspects of the overall risk management practices. One of the most important aspects was what risk analysis techniques were used in both the oil and the construction industry. The risk analysis techniques were divided in two categories: qualitative and quantitative. 80 % of the companies use a combination of qualitative and quantitative methods while the remaining

20% concentrated only on qualitative methods. The qualitative techniques used were based on personal and corporate experience, and engineering judgement. The quantitative methods used in the construction industry were: expected monetary value (EMV), break-even analysis, and scenario analysis. These techniques belong to sensitivity analysis and are related to the evaluation of financial risk. On the other hand, the techniques used in the oil industry analysed financial and safety aspects and were: EMV, expected net present value, algorithms, decision matrix, decision tree, break-even analysis, scenario analysis, and simulation. Finally, Baker et al. (1999) presented an order of importance of six types of risk; these are presented in Table 3.1.

	Construction industry	Oil industry
Types of risk	**Position**	**Position**
Financial	1	1
Technical	2	2
Operational	4	3
Time	3	5
Environmental	5	4
Political	6	6

Table 3.1 Order of importance of six types of risk
Adapted from Baker et al. (1999)

Evans and Olson (2002) outline as applications of risk analysis the following areas: operations management, finances, marketing, and engineering. They use simulation to perform risk analysis in specific problems such as emergency rooms, local area networks, job shops, and supply chain systems.

Langford (2002) designed a survey to understand the way people perceived risk. It was stated that for discussing risk perception and communication in contemporary society it was necessary to consider the concept of trust. For example, if trust is lost it can modify people's worldview. In other words, it is important to consider trust because the act of trusting is a risky activity. On the one hand, making a study of human thinking, feeling and being can deliver some answers to the question of why people perceive environmental and health risk as they do. On the other hand, it can help the DMs to understand why they are looking for a particular solution that might be influenced by personal, social, cultural and political interests. This paper offers a perspective of a risk-oriented society and how people's perception of risk can affect the decision-making process.

Oka et al. (2001) present the application of risk assessment on a case of land-use conversion in Japan. The land in question was used as rice-growing fields for hundreds of years; this area supported many species of aquatic plants, many species of fish, and insects. A gas company planned the construction of a liquefied natural gas plant and it was necessary to assess the environmental impact. The method used to measure if the probability of extinction increased was simulation using a time horizon of 1, 10 and 100 years. The diversity of species was also measured using pairwise measurement and priorities setting. Finally, they performed a risk-benefit analysis, concluding that the method proposed allowed them to obtain an indicator of the diversity of the environment as a whole.

Romerio (2002) presents a comparison between three different paradigms to manage the risk of ionising radiation. In this paper risk is understood as "the probability of developing a solid cancer or leukaemia" and a paradigm is defined as the concept that represents "the set of problems, hypotheses, laws, and methods of analysis accepted and used by members of a scientific community". The paradigms that are compared are that of the National Commission of Radiological Protection, one that affirms that the effect of low doses is relatively weak, and

one that affirms that the effect of low doses is relatively serious. Risk estimation was performed for each paradigm using a dose-effect model. The conclusion presented is that there are no reasons to prefer one model to another. In the risk management stage dose limit to the public is analysed; this differs in each model. Thus with the information presented in this paper, a DM would be capable of selecting a paradigm to use in a problem related to the use of radiation. This is a good example of how risk analysis can contribute to making decisions related to energy, environmental and health policies.

Moreover, Deisler (2002) presents a survey where he analyses whether the application of risk analysis (risk assessment, risk management and risk communication) can help to lessen the impact of terrorist attacks. He emphasises three areas to reduce risk: 1) risk communication, this consists of improving the communication techniques of governmental leaders and spokespersons about risk, 2) chemical and biological risk analysis, this consists of dealing with the consequences of an attack, training and educating, doing research, detecting diseases, and having better emergency response capabilities, 3) technological risk analysis, this consists of detecting places or facilities that are vulnerable to attack and installing warning systems with the objective of reducing risk.

In this subsection some applications of risk analysis, risk sensitivity, risk assessment and risk management have been presented, however, it is important to bear in mind that these are just a small portion of the whole range of possibilities for these areas of study.

Having defined objectives, attributes, goals, criteria, constraints, variables, decision theory, and risk management now the analysis of the relations between these terms and the decision-making process is needed. In order to perform this analysis it is necessary to define an area of study called multiple criteria decision-making, this area is discussed in the next section.

3.4 Multiple Criteria Decision-Making

On the one hand, according to Zimmermann (1996) MCDM is divided into two classes of models: multiple objective decision-making (MODM) and multiple attribute decision-making (MADM). This division is based on the type of decision space they have. MODM primarily focuses on continuous decision spaces whilst MADM focuses on discrete decision spaces. Despite this classification some authors use the terms objectives or attributes indistinctly to refer to MCDM. MODM problems are defined as the maximisation of a vector, $\mathbf{Z}(x) = (\mathbf{z}_1(x),\ldots, \mathbf{z}_k(x))$, where $\mathbf{x} \in X$ (solution space); $\mathbf{x} \in \Re^k$. It is evident that its decision space is continuous, whilst in MADM the decision space is defined as a finite set of alternatives $X = \{x_i \mid i = 1,\ldots, n\}$ and in order to find the optimal alternative a finite set of goals, $G = \{g_j \mid j = 1,\ldots, m\}$ is used. Methods to solve both of these models are discussed below in this section. Yoon and Hwang (1995) presented the same classification of MCDM problems.

On the other hand, according to Triantaphyllou (2000), there are three classes of methods to solve MCDM problems: deterministic, stochastic and fuzzy. These methods will be discussed in subsections 3.4.1, 3.4.2 and 3.4.3.

Another way of classifying MCDM methods is by the type of information provided by the decision-maker (DM). It is important to have in mind that this classification is mainly used for deterministic methods and was proposed by Chen and Hwang (1991) and then adapted by Yoon and Hwang (1995). It is important to notice that this classification or taxonomy not only focuses on the type of information managed but also on the fact that these methods are only used to solve MADM problems. This taxonomy is shown in Figure 3.4.

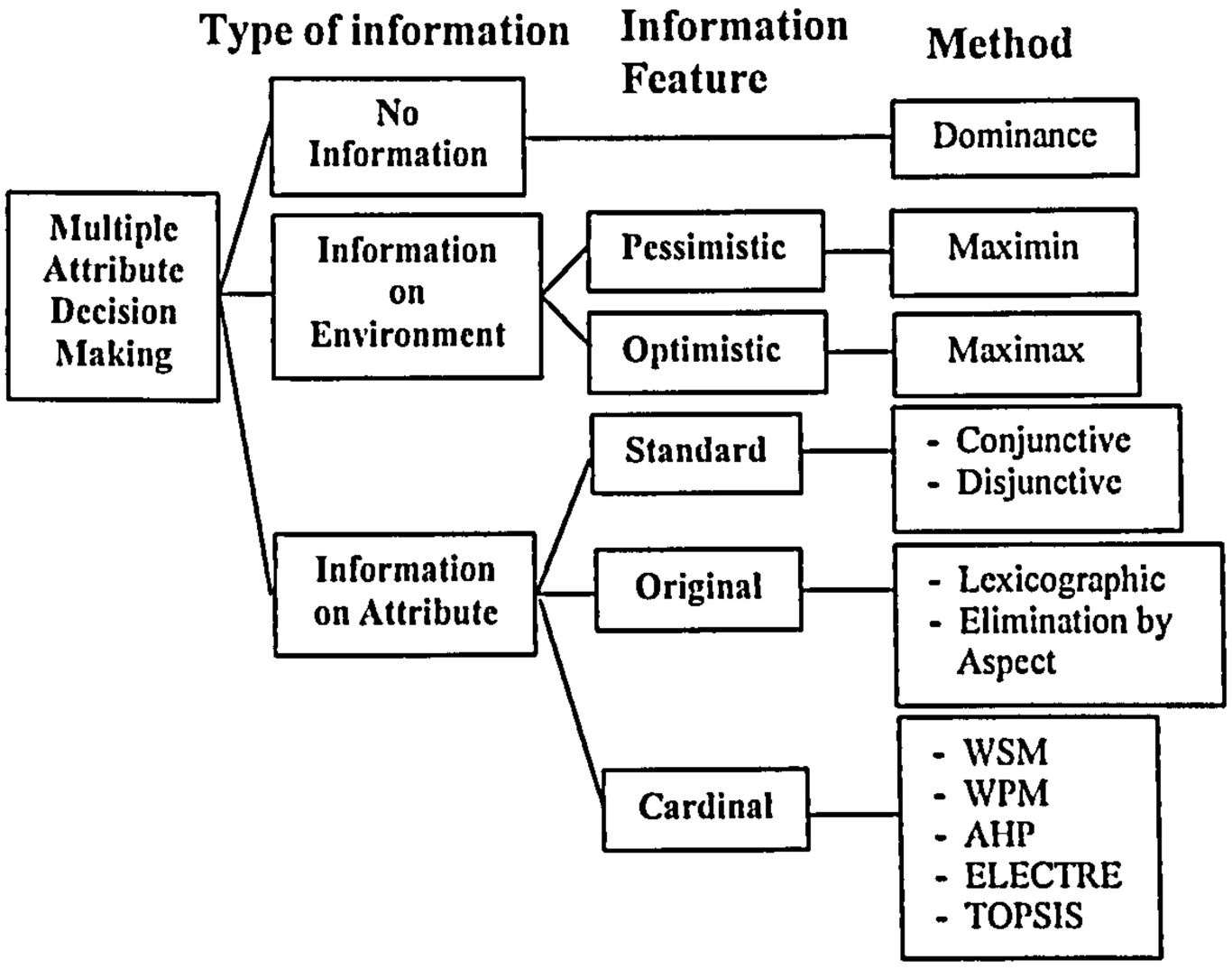


Figure 3.4 Taxonomy of MADM
Adapted from Yoon and Hwang (1995)

MCDM is also known as *multiple criteria decision-aid* (MCDA) or 'analyse multicritère' in French. Vincke in his book "Multi-criteria decision-aid" first published in French in 1989 ("L'Aide Multicritère À La Décision") and then translated to English in 1992, states the main objective of MCDA is to provide the DM with the necessary tools to solve a multiple objective or attribute problem. He defines a set of actions, where these could be solutions or decisions. The set of actions denoted by A, according to Vincke (1992) can be *stable* (defined a priori and not changeable) or *evolutive* (modified during the solution process). The definition of A depends not only on the nature of the problem and the decision procedure but also on the criteria definition, the preferences model, and the decision-aid method to be applied.

Vincke (1992) defined a criterion as a function g on A, where g takes certain values and represents the decision-maker's preferences. It is necessary to define a family $F=\{g_1, g_2, \dots, g_j, \dots, g_n\}$; this includes several criteria where each g represents a single criterion. It is expected that the family of criteria F represents all the aspects of the problem to be solved.

The need to define a structure for the preferences appears when the DM has to compare two actions. The possible reactions of the DM are as follows (Vincke, 1992):

a) Preference for one of the actions. Having actions x and y, to say that action x is preferred to action y, it is written that xPy.
b) Indifference between the actions. Having actions x and y, to represent indifference, it is written that xIy.
c) Inability to compare the actions. Having actions x and y, to represent incomparability, it is written that xJy.

After the preferences are defined the next question can be posed: how can these preferences be represented in an analytical way? To answer this, a new area study has to be developed, which is called multiple attribute utility theory.

The basis of this theory is that the DM commonly tries to optimise (maximise) a certain unknown function. This function U (Vincke, 1992) should include all the different criteria and aspects of the problem and is called "utility function". The decision-maker's preferences are

represented by an analytical function as additive, multiplicative or mixed. And as mentioned above, the basis to define a function *U*:

$$U(a) = \sum_{i=1}^{n} U_i(f_i(a)) \tag{3.3}$$

This function is the one used in the additive model, where $f_i(a)$ is the *i*th criterion and a is an action that belongs to the space of possible action to be followed. The additive model is one of the most used because of its simple form.

The multiplicative model is defined as follows:

$$U(a) = \frac{\prod_{i=1}^{n}[1 + k\,k_i U_i(a_i)] - 1}{k} \tag{3.4}$$

where k and k_i are scaling constants.

Having discussed the basic concepts and definitions of MCDM the next step to follow is the descriptions of the methods for solving this kind of problem. For the purposes of this research the classification used is that proposed by Triantaphyllou (2000) where three classes of method are considered: deterministic, stochastic and fuzzy. This classification will be used throughout this chapter.

3.4.1 Deterministic MCDM Methods

In this section some of the most used deterministic methods for solving MCDM problems are discussed. These methods are: weighted sum model, weighted product model, analytic hierarchy process, elimination and choice translating reality method and the method for the technique for order preference by similarity to ideal solution.

In most of these methods the decision problem is defined using a decision matrix **A** that consists of a set of alternatives, a set of criteria, and performance values for each alternative. As mentioned in subsection 3.2.2, for this kind of method the set of alternatives is considered to be finite. Consequently these methods are for solving MADM problems. Yoon and Hwang (1995) define the basic steps to follow in the solution of a MADM, and as mentioned below these are followed in most of the methods. The steps will complete a procedure followed by the DM, defined as follows:

1. Attribute Generation,
2. Attribute Weighting,
3. Quantification of Qualitative Ratings,
4. Normalisation of Attribute Ratings.

Each step will be defined in this section according to the method to be explained.

Generally speaking the main part of a problem solution is to pose the problem. In the MADM case the problem will be posed using a *decision matrix*.

Having a set of m alternatives $A = \{A_1, A_2, A_3, \ldots, A_m\}$ and a set of n decision criteria, $C = \{C_1, C_2, C_3, \ldots, C_n\}$ the decision matrix **A** is represented as follows (Triantaphyllou 2000, Hwang and Yoon 1995),

Alternatives	Criteria				
	C_1	C_2	C_3	...	C_n
A_1	a_{11}	a_{12}	a_{13}	...	a_{1n}
A_2	a_{21}	a_{22}	a_{23}	...	a_{2n}
⋮	⋮	⋮	⋮	⋮	⋮
A_m	a_{m1}	a_{m2}	a_{m3}	...	a_{mn}

where the a_{ij} values (for $i = 1, 2, 3, \ldots, m$ and $j = 1, 2, 3, \ldots, n$) are the performance values of each alternative in terms of each criterion. For instance, a_{23} is the performance value of alternative A_2 in terms of criterion C_3, or a_{m2} is the performance value of alternative A_m in terms of criterion C_2.

Weighted sum model (WSM)

The decision problem solved using the WSM method is defined as follows (Triantaphyllou, 2000):
Having a decision matrix **A**, where all the criteria have the same units, it is necessary to calculate the A_i values (equation 3.5) to find the alternative that satisfies equation 3.6:

$$A_i = \sum_{j=1}^{n} a_{ij} w_j \quad \text{for} \quad i = 1, 2, 3, \ldots, m \tag{3.5}$$

$$\text{score} = \max A_i \tag{3.6}$$

To show this method the following example is presented. Having the decision matrix **A** with 4 alternatives and 5 criteria:

Alternatives	Criteria				
	C_1 (w_1	C_2 w_2	C_3 w_3	C_4 w_4	C_5 w_5)
A_1	a_{11}	a_{12}	a_{13}	a_{14}	a_{15}
A_2	a_{21}	a_{22}	a_{23}	a_{24}	a_{25}
A_3	a_{31}	a_{32}	a_{33}	a_{34}	a_{35}
A_4	a_{41}	a_{42}	a_{43}	a_{44}	a_{45}

Using equation 3.5 to calculate the value of each alternative gives the following results:

$A_1 = a_{11}w_1 + a_{12}w_2 + a_{13}w_3 + a_{14}w_4 + a_{15}w_5$
$A_2 = a_{21}w_1 + a_{22}w_2 + a_{23}w_3 + a_{24}w_4 + a_{25}w_5$
$A_3 = a_{31}w_1 + a_{32}w_2 + a_{33}w_3 + a_{34}w_4 + a_{35}w_5$
$A_4 = a_{41}w_1 + a_{42}w_2 + a_{43}w_3 + a_{44}w_4 + a_{45}w_5$

The following step is to identify which of these alternatives has the maximum value. This alternative will be considered to be the best.

Weighted product model (WPM)

Basically, this model uses multiplication instead of addition and is based on the comparison of each alternative with the others. In order to compare two alternatives it is necessary to calculate a ratio and raise it to the power of the weight related to that criterion (Yoon Hwang, 1995), as shown in equation 3.7:

$$R(A_x/A_y)=\prod_{j=1}^{n}(a_{xj}/a_{yj})^{w_j} \tag{3.7}$$

Then it is possible to conclude that one of the main characteristics of this method is that it is dimensionless. Therefore it allows the comparison between criteria with different units.

For instance, having the following decision matrix:

	Criteria			
Alternatives	C_1 (w_1	C_2 w_2	C_3 w_3	C_4 w_4)
A_1	a_{11}	a_{12}	a_{13}	a_{14}
A_2	a_{21}	a_{22}	a_{23}	a_{24}
A_3	a_{31}	a_{32}	a_{33}	a_{34}

The calculated ratios are as follows:

$$R(A_1/A_2)=(a_{11}/a_{21})^{w_1}\times(a_{12}/a_{22})^{w_2}\times(a_{13}/a_{23})^{w_3}\times(a_{14}/a_{24})^{w_4}$$
$$R(A_1/A_3)=(a_{11}/a_{31})^{w_1}\times(a_{12}/a_{32})^{w_2}\times(a_{13}/a_{33})^{w_3}\times(a_{14}/a_{34})^{w_4}$$
$$R(A_2/A_3)=(a_{21}/a_{31})^{w_1}\times(a_{22}/a_{32})^{w_2}\times(a_{23}/a_{33})^{w_3}\times(a_{24}/a_{34})^{w_4}$$

Finally, the ratio values are compared to rank the alternatives. For example, if the ratio values are the following:

$R(A_1/A_2)=0.85$, because $0.85<1$ then A_2 is greater than A_1
$R(A_1/A_3)=1.35$, because $1.35>1$ then A_1 is greater than A_3
$R(A_2/A_3)=1.63$, because $1.63>1$ then A_2 is greater than A_3

hence $A_2>A_1>A_3$ is the ranking found.

Analytic Hierarchy Process (AHP)

Saaty (1994) describes this method as a process to identify which objective weighs more than others. The main characteristic of the AHP method is the creation of a hierarchy through paired comparisons made by humans based on their ability to relate the data available and their experience. This method also assesses the benefits, the cost and the risk of the feasible solutions. AHP allows the DM to be a central element in the control of the outcome and can be also used to quantify some DM emotional factors.

According to Goodwin and Wright (1998), this method is divided into 5 stages:

Stage 1: Set up the hierarchy of the decision. This process starts stating the general objective of the decision then the general attributes. After the general attributes are determined, these are divided into more detailed attributes to conform to the next level of the hierarchy. Once all the attributes (criteria) are specified the possible alternatives (courses of action) are placed. Figure 3.5 shows the hierarchy resultant from this process.

Stage 2: Make pairwise comparisons. In this process each attribute is compared in terms of its importance; the comparison is made in pairs.
Since the pairwise comparison method is one of the most commonly used to solve MCDM problems, it is important to discuss some of its main characteristics and properties. Mainly,

the comparisons are used to determine the relative importance of each alternative in the problem, and to generate ratio scales (e.g. the Kelvin scale used to measure temperature). This method also helps to assign a weight to each DM's preferences and to translate them to a utility function. When pairwise comparisons are made, it is necessary to consider how the DM expresses his or her opinions (choices) linguistically and how to quantify them.

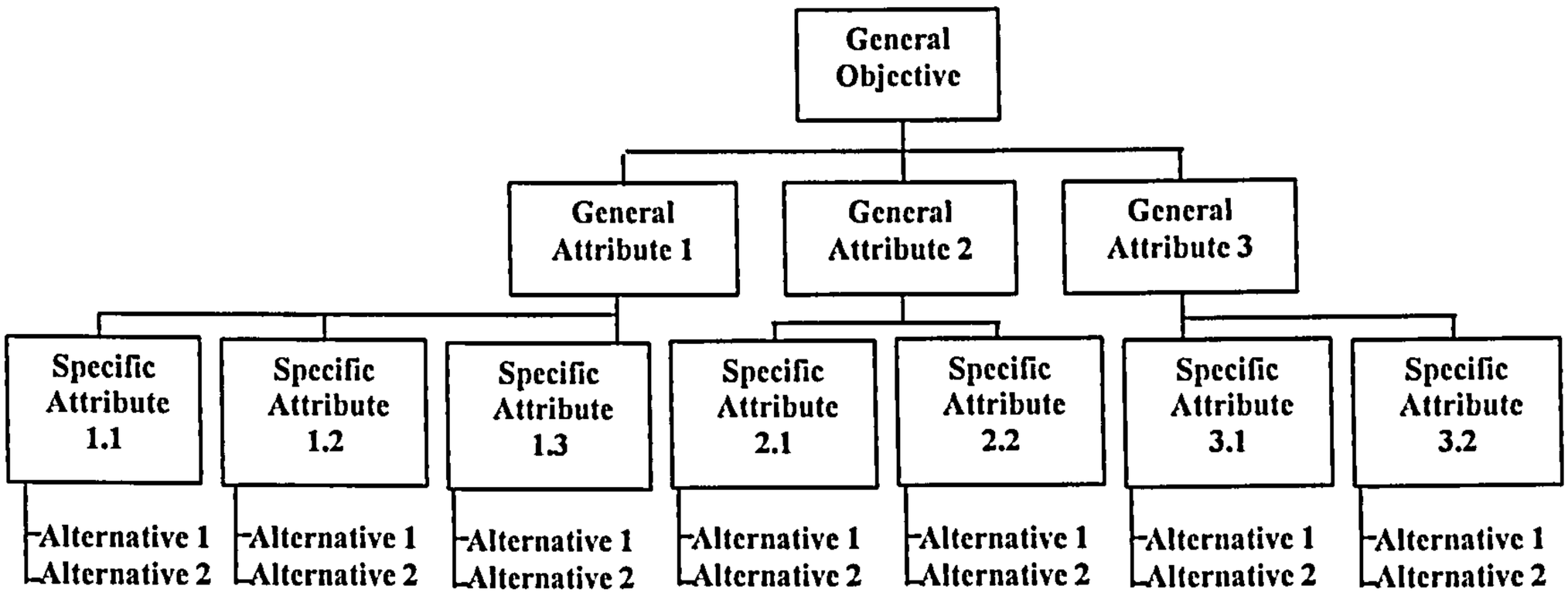


Figure 3.5 General hierarchy for a decision problem

Having defined the hierarchy (stage 1, Figure 3.5), the first step is to make paired comparisons between the elements of each level. Saaty (1994) proposes to ask two questions when making the comparisons. The first question is related to *dominance*, in other words, which of the two elements better attains the criterion. The second question is related to *intensity*, in other words, by how much is it better?

The method of pairwise comparisons is used in Chapter Five of this thesis to identify and represent the DM's preferences.

Stage 3: Transform the comparisons into weights. After the pairwise comparisons are made these have to be transformed into weights. The weights have to be normalised, this means, that the sum of weights is equal to one. In other words, the weights for the objective and attributes of the hierarchy in Figure 3.6 are represented as follows:

If w_o is the weight assigned to the general objective then $w_o = 1.0$.
If w_1 is the weight assigned to the general attribute 1, w_2 is the weight assigned to the general attribute 2, and w_3 is the weight assigned to the general attribute 3 then $w_1 + w_2 + w_3 = 1.0$.
If $w_{1.1}$ is the weight assigned to the specific attribute 1.1, $w_{1.2}$ is the weight assigned to the specific attribute 1.2, and $w_{1.3}$ is the weight assigned to the specific attribute 1.3 then $w_{1.1} + w_{1.2} + w_{1.3} = 1.0$.
If $w_{2.1}$ is the weight assigned to the specific attribute 2.1, and $w_{2.2}$ is the weight assigned to the specific attribute 2.2 then $w_{2.1} + w_{2.2} = 1.0$.
If $w_{3.1}$ is the weight assigned to the specific attribute 3.1, and $w_{3.2}$ is the weight assigned to the specific attribute 3.2 then $w_{3.1} + w_{3.2} = 1.0$.
If $w_{1.1a1}$ is the weight assigned to the alternative 1 that follows the path of the specific attribute 1.1, and $w_{1.1a2}$ is the weight assigned to the alternative 2 that follows the path of the specific attribute 1.1 then $w_{1.1a1} + w_{1.1a2} = 1.0$.
If $w_{1.2a1}$ is the weight assigned to the alternative 1 that follows the path of the specific attribute 1.2, and $w_{1.2a2}$ is the weight assigned to the alternative 2 that follows the path of the specific attribute 1.2 then $w_{1.2a1} + w_{1.2a2} = 1.0$.

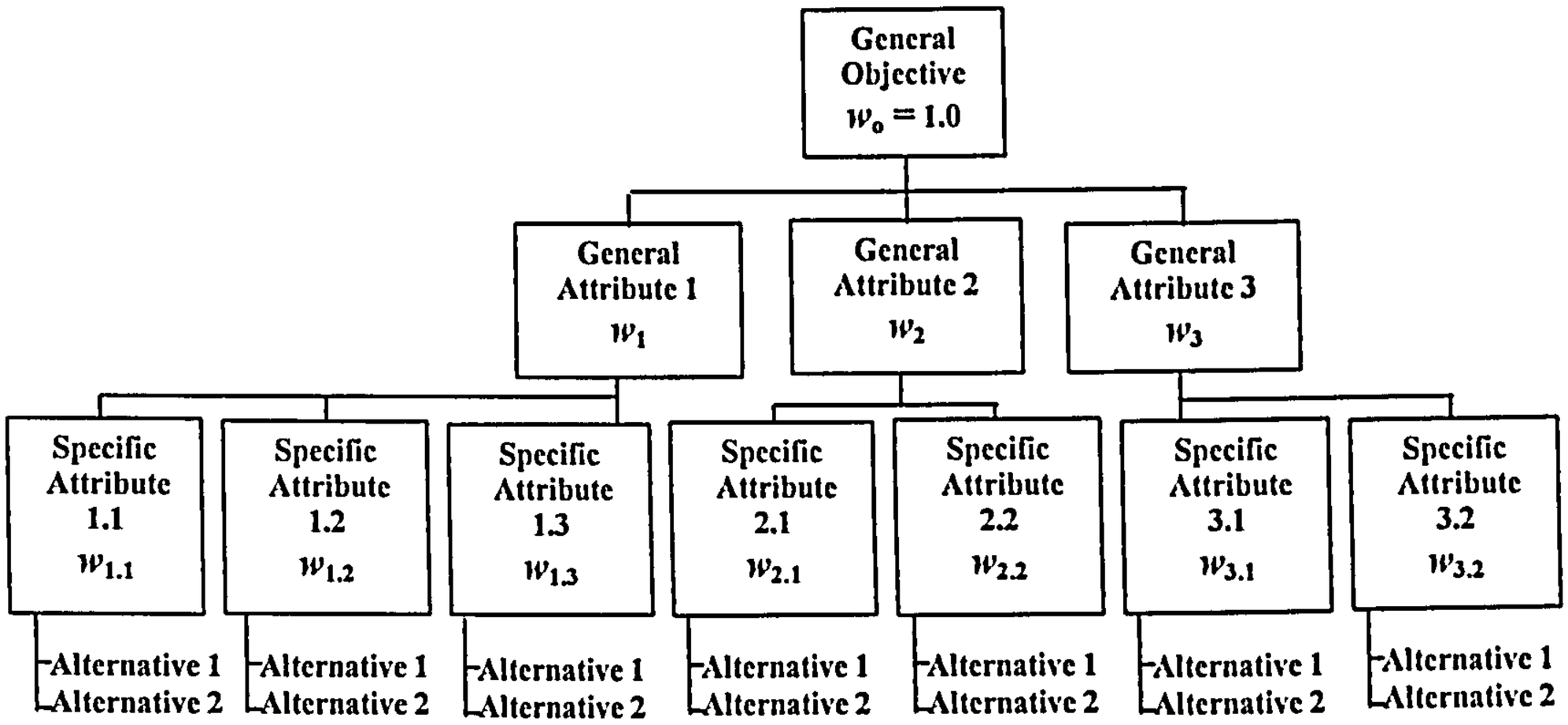


Figure 3.6 Weights of the hierarchy for a decision problem

Then for each alternative in each path for each specific attribute the weights are described as follows:

$w_{1.3a1} + w_{1.3a2} = 1.0,$
$w_{2.1a1} + w_{2.1a2} = 1.0,$
$w_{2.2a1} + w_{2.2a2} = 1.0,$
$w_{3.1a1} + w_{3.1a2} = 1.0,$
$w_{3.2a1} + w_{3.2a2} = 1.0.$

The procedure to transform comparisons into weights was developed first by Saaty (1977), where he proposed to build a square matrix of pairwise comparisons to express the intensity of dominance. The matrix's size is $n \times n$, where n is the number of criteria in the second level of the hierarchy (General Attributes, Figure 3.5). This matrix is shown in Figure 3.7 and it is reciprocal, this means that the element $a_{ji} = 1/a_{ij}$, where $i = 1, 2,\ldots, n$ and $j = 1, 2,\ldots, n$.

	Criteria				
Criteria	C_1	C_2	C_3	...	C_n
C_1	1	$1/a_{21}$	$1/a_{31}$	...	$1/a_{n1}$
C_2	a_{21}	1	$1/a_{32}$	...	$1/a_{n2}$
C_3	a_{31}	a_{32}	1	...	$1/a_{n3}$
⋮	⋮	⋮	⋮	⋮	⋮
C_n	a_{n1}	a_{n2}	a_{n3}	...	1

Figure 3.7 Matrix of pairwise comparisons

After the comparisons matrix is built, a priorities column is added. These priorities are the components of the eigenvector of the matrix. To calculate the eigenvector the procedure shown below is followed:

Having the matrix of pairwise comparisons **A** and the condition $\mathbf{A}w = \lambda_{max}w$ that can be also expressed as $(\mathbf{A}-\lambda_{max}\mathbf{I})w = 0$, it is said that the determinant $|\ \mathbf{A}-\lambda_{max}\mathbf{I}| = 0$, can be expressed as the equation (*characteristic equation*) of a nth degree polynomial in λ_{max} (n is the order of matrix **A**) equal to zero. Where λ_{max} is a root of the characteristic equation. This root is known as the eigenvalue of the matrix **A**.

Saaty gives a more detailed explanation of this method in his book 'Fundamentals of Decision Making and Priority Theory' (Saaty, 1994).

Stage 4: Make a provisional decision. Once the weights are determined it is necessary to calculate scores for each of the specific alternatives. For example, in order to calculate the score of alternative 1, it is necessary to follow all the paths that lead to it (Figure 3.6), multiply the weights of each path and add these results. In particular, one path can be: general attribute 1 → specific attribute 1.1 → alternative 1. This procedure is shown below:

Score for alternative 1

$$\begin{aligned} &= w_1 \times w_{1.1} \times w_{1.1a1} \\ &+ w_1 \times w_{1.2} \times w_{1.2a1} \\ &+ w_1 \times w_{1.3} \times w_{1.3a1} \\ &+ w_2 \times w_{2.1} \times w_{2.1a1} \\ &+ w_2 \times w_{2.2} \times w_{2.2a1} \\ &+ w_3 \times w_{3.1} \times w_{3.1a1} \\ &+ w_3 \times w_{3.2} \times w_{3.2a1} \end{aligned}$$

Score for alternative 2

$$\begin{aligned} &= w_1 \times w_{1.1} \times w_{1.1a2} \\ &+ w_1 \times w_{1.2} \times w_{1.2a2} \\ &+ w_1 \times w_{1.3} \times w_{1.3a2} \\ &+ w_2 \times w_{2.1} \times w_{2.1a2} \\ &+ w_2 \times w_{2.2} \times w_{2.2a2} \\ &+ w_3 \times w_{3.1} \times w_{3.1a2} \\ &+ w_3 \times w_{3.2} \times w_{3.2a2} \end{aligned}$$

After the scores are calculated, they are compared and the alternative that has the highest value is considered the best.

Stage 5: Perform sensitivity analysis. This procedure measures how sensitive to changes the alternative selected is. According to Haimes (1998), sensitivity is related to "changes in the system's output to possible variations in the decision variables (inputs)". The sensitivity analysis techniques are discussed in subsection 3.3.1.

According to Saaty (1994), some of the areas of application of AHP are "planning, generating a set of alternatives, setting priorities, choosing a best policy after finding a set of alternatives, allocating resources, determining requirements, predicting outcomes, designing systems, measuring performance, insuring the stability of a system, optimising, and resolving conflict". It is also said that AHP is a very powerful method for decision-making because it gives a structure to the decision.

Elimination and Choice Translating Reality method (ELECTRE)

The ELECTRE ("Elimination et choix traduisant la réalité" in French) method developed by Roy (1977) is based on *outranking relations*. These relations are obtained using pairwise comparisons of the alternatives. Therefore an outranking relation will be defined as:
Having two alternatives A_p and A_q, then A_pRA_q read as A_p outranks (dominates) A_q and means that A_p is preferred to A_q.

In the first place it is considered that the decision matrix is not dimensionless so it is necessary to normalise each of the alternatives by using equation 3.8:

$$b_{ij} = \frac{a_{ij}}{\sqrt{\sum_{k=1}^{m} a_{kj}^2}} \tag{3.8}$$

where m is the number of alternatives, and a_{ij} is the performance value of ith alternative in terms of the jth criterion.

After normalising, a new *normalised decision matrix* **B** is built:

$$\mathbf{B} = \begin{bmatrix} b_{11} & b_{12} & b_{13} & \cdots & b_{1n} \\ b_{21} & b_{22} & b_{23} & \cdots & b_{2n} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ b_{m1} & b_{m2} & b_{m3} & \cdots & b_{mn} \end{bmatrix} \tag{3.9}$$

where m is the number of alternatives, n is the number of criteria and b_{ij} is the normalised a_{ij} value.

In addition to the normalisation process it is necessary to weight matrix **B**. Weighting a matrix is the process of multiplying the normalised decision matrix **B** by the matrix of weights **W**. The matrix of weights is defined as follows:

$$\mathbf{W} = \begin{bmatrix} w_1 & 0 & 0 & \cdots & 0 \\ 0 & w_2 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & w_n \end{bmatrix} \tag{3.10}$$

where n is the number of criteria and $\sum_{i=1}^{n} w_i = 1$. Having defined **W** it is possible to define a *weighted matrix* **E** as follows:

$$\mathbf{E} = \mathbf{B}\,\mathbf{W} \tag{3.11}$$

$$\mathbf{E} = \begin{bmatrix} e_{11} & e_{12} & e_{13} & \cdots & e_{1n} \\ e_{21} & e_{22} & e_{23} & \cdots & e_{2n} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ e_{m1} & e_{m2} & e_{m3} & e_{m4} & e_{mn} \end{bmatrix} = \begin{bmatrix} w_1 b_{11} & w_2 b_{12} & w_3 b_{13} & \cdots & w_n b_{1n} \\ w_1 b_{21} & w_2 b_{22} & w_3 b_{23} & \cdots & w_n b_{2n} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ w_1 b_{m1} & w_2 b_{m2} & w_3 b_{m3} & \cdots & w_n b_{mn} \end{bmatrix} \tag{3.12}$$

As mentioned above, the ELECTRE method is based on the generation of outranking relationships. To formulate these relationships it is necessary to obtain a *concordance set* and a *discordance set* (Yoon and Hwang, 1995). Having two alternatives A_p and A_q, the set of criteria that meet that $A_p \mathrm{P} A_q$ (A_p is preferred to A_q) is called concordance set and is defined as:

$$C_{pq} = \{j, e_{pj} \geq e_{qj}\}, \qquad \text{for } j = 1, 2, 3, \ldots, n \tag{3.13}$$

where e_{pj} is the element of the weighted matrix **E** that corresponds to the alternative A_p and the jth criterion and e_{qj} is the element that corresponds to the alternative A_q and the jth criterion.

Therefore the discordance set is the complement of the concordance set and is defined as:

$$D_{pq} = \{j, e_{pj} < e_{qj}\}, \qquad \text{for } j = 1, 2, 3, ..., n \tag{3.14}$$

In other words, the discordance set contains the criteria that do not meet the preference $A_p \mathrm{P} A_q$.

Once the concordance set is defined, it is possible to obtain the *concordance index* c_{pq} using equation 3.15:

$$c_{pq} = \sum_{j \in C_{pq}}^{n} w_j, \quad 0 \le c_{pq} \le 1 \tag{3.15}$$

where w_j is the weight associated to the jth criterion contained in the concordance set.

Having found the concordance indices, the *concordance matrix* **C** is:

$$\mathbf{C} = \begin{bmatrix} - & c_{12} & c_{13} & \cdots & c_{1m} \\ c_{21} & - & c_{23} & \cdots & c_{2m} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ c_{m1} & c_{m2} & c_{m3} & \cdots & - \end{bmatrix} \tag{3.16}$$

where the value c_{pq} is not defined when $p = q$.

In a similar way as the concordance index was defined, the *discordance index* d_{pq} is calculated using equation 3.17 and the discordance set:

$$d_{pq} = \frac{\max\limits_{j \in D_{pq}} |e_{pj} - e_{qj}|}{\max\limits_{j} |e_{pj} - e_{qj}|} \tag{3.17}$$

Having found the discordance indices, the *discordance matrix* **D** is:

$$\mathbf{D} = \begin{bmatrix} - & d_{12} & d_{13} & \cdots & d_{1m} \\ d_{21} & - & d_{23} & \cdots & d_{2m} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ d_{m1} & d_{m2} & d_{m3} & \cdots & - \end{bmatrix} \tag{3.18}$$

where the value d_{pq} is not defined when $p = q$.

The next step is to obtain the *concordance* and *discordance dominance matrices*. Triantaphyllou (2000) states that the concordance dominance matrix is constructed using a threshold value related to the concordance index. Therefore having a threshold value $\underline{c}$ it is said that the alternative A_p dominates the alternative A_q if the concordance index $c_{pq} \ge \underline{c}$. The value for $\underline{c}$ is calculated using equation 3.19:

$$\underline{c} = \frac{1}{m(m-1)} \sum_{p=1}^{m} \sum_{q=1}^{m} c_{pq}, \quad \text{where } p \ne q \tag{3.19}$$

Essentially, the threshold value is considered as the average of the concordance indices. The elements of the concordance matrix **F** are determined using the following function:

$$f_{pq} = \begin{cases} 1, & \text{if } c_{pq} \geq \underline{c} \\ 0, & \text{if } c_{pq} < \underline{c} \end{cases} \tag{3.20}$$

In the same way the discordance matrix **G** is constructed by using a threshold value $\underline{d}$ defined as follows (average of the discordance indices):

$$\underline{d} = \frac{1}{m(m-1)} \sum_{p=1}^{m} \sum_{q=1}^{m} d_{pq} \quad \text{where } p \neq q \tag{3.21}$$

The discordance matrix **G** is determined using the following function:

$$g_{pq} = \begin{cases} 1, & \text{if } d_{pq} \geq \underline{d} \\ 0, & \text{if } d_{pq} < \underline{d} \end{cases} \tag{3.22}$$

Once the concordance and discordance dominance matrices have been calculated, *the aggregate dominance matrix* **H** can be constructed using the following equation:

$$h_{pq} = f_{pq} \times g_{pq} \tag{3.23}$$

Finally, the less favourable alternatives from the aggregate dominance matrix have to be eliminated. In other words, if $h_{pq} = 1$, this means that $f_{pq} = 1$ ($c_{pq} \geq \underline{c}$) and $g_{pq} = 1$ ($d_{pq} \geq \underline{d}$) meaning that alternative A_p is preferred to A_q. Therefore if any column in matrix **H** (aggregate dominance) has an element equal to 1, then it is said that the column is dominated by the row. Having stated this, it is possible to eliminate all the columns that have an element equal to 1.

According to Yoon and Hwang (1995), one disadvantage of the ELECTRE method is that it relies on threshold values and these are, in general, arbitrary.

Technique for Order Preference by Similarity to Ideal Solution (TOPSIS)

Yoon and Hwang developed this method in 1981 (Yoon and Hwang, 1995). They established that if a MADM problem is defined by m alternatives and n attributes (criteria) it might be analysed as a geometric system. This system has m points in the n-dimensional space, and for that reason it is possible to speak about Euclidean distances. Furthermore, the TOPSIS method is based on the assumption that the alternative selected should have the longest distance from the *negative-ideal solution* and the shortest distance form the *positive-ideal solution*.

What is an *ideal solution*? An ideal solution is "a collection of ideal ratings in all attributes considered" (Yoon and Hwang, 1995). Nevertheless, in most of the cases the ideal solution is not feasible. Thus the positive-ideal solution is defined as the set of the best values for the attributes among all the alternatives available. Additionally, the negative-ideal solution is defined as the set of the worst values for the attributes among all available alternatives. Mainly, finding an alternative that is the farthest to the negative-ideal solution and the closest to the positive-ideal solution is not an easy task. That is why an index called *similarity* is defined.

Generally speaking, this method follows 6 steps presented below (Yoon and Hwang, 1995):

Step 1: Calculate normalised ratings. (Using the same notation used in the ELECTRE method).

$$b_{ij} = \frac{a_{ij}}{\sqrt{\sum_{k=1}^{m} a_{kj}^2}} \tag{3.24}$$

where $i = 1, 2, \ldots, m$ and $j = 1, 2, \ldots, n$.

Step 2: Calculate weighted normalised ratings.

$$e_{ij} = w_j b_{ij} \tag{3.25}$$

where $i = 1, 2, \ldots, m, j = 1, 2, \ldots, n$ and w_j is the weight of the jth attribute.

Step 3: Identify the positive-ideal solution $\mathbf{A}^+$ and the negative-ideal solution $\mathbf{A}^-$. These solutions are defined in terms of the weighted normalised values e_{ij}:

$$\mathbf{A}^+ = \{e_1^+, e_2^+, \ldots, e_j^+, \ldots e_n^+\} = \left\{\left(\max_i e_{ij} \middle| j \in J_1\right), \left(\min_i e_{ij} \middle| j \in J_2\right) \middle| \; i = 1, \ldots, m\right\} \tag{3.26}$$

$$\mathbf{A}^- = \{e_1^-, e_2^-, \ldots, e_j^-, \ldots e_n^-\} = \left\{\left(\min_i e_{ij} \middle| j \in J_1\right), \left(\max_i e_{ij} \middle| j \in J_2\right) \middle| \; i = 1, \ldots, m\right\} \tag{3.27}$$

where J_1 is a set of benefit attributes and J_2 is a set of cost attributes.

Step 4: Calculate separation measures. The separation between alternatives will be measured by means of the n-dimensional Euclidean distance. Hence the distance between the positive-ideal solution and each alternative is given by the following equation:

$$S_i^+ = \sqrt{\sum_{j=1}^{n} \left(e_{ij} - e_j^+\right)^2} \tag{3.28}$$

where S_i^+ is the separation (distance) between the positive-ideal solution e^+ and the ith alternative, $i = 1, 2, \ldots, m$, and e_{ij} are the weighted normalised values .

The distance (separation) between the negative-ideal solution and each alternative is given by the following equation:

$$S_i^- = \sqrt{\sum_{j=1}^{n} \left(e_{ij} - e_j^-\right)^2} \tag{3.29}$$

Step 5: Calculate similarities to positive-ideal solutions.

$$L_i^+ = \frac{S_i^-}{\left(S_i^+ + S_i^-\right)} \tag{3.30}$$

where $i = 1, 2, \ldots, m$ and $0 \leq L_i^+ \leq 1$.

Step 6: Rank preference order. This procedure consists of ranking alternatives in descending order with respect to L_i^+ .

3.4.2 Stochastic MCDM Methods

Probabilistic Trade-off Development method (PROTRADE)

PROTRADE is a multiple objective stochastic method and was developed by Goicoechea et al. (1979) with the purpose of solving non-linear problems considering the DM's preferences. This method not only allows the DM's interaction but is also capable of handling risk. Basically, it consists of the formulation of a surrogate function that is modified by the DM every time the algorithm is performed. The surrogate function is defined using the following procedure:

1. A vector of objective functions is defined using the coefficients' expected values:

$$\begin{aligned} &\mathbf{z}(\mathbf{x}) = [z_1(\mathbf{x}), z_2(\mathbf{x}), \ldots, z_p(\mathbf{x})], \\ &g_q(\mathbf{x}) \le 0 \quad \textit{where} \quad q \in I[1, Q] \\ &\mathbf{x} > 0, \\ &z_i(\mathbf{x}) = \sum_{j=1}^{n} c_{ij}\, x_j, z_i(\mathbf{x}) = E[\mathbf{x}] \end{aligned} \tag{3.31}$$

where p is the number of objectives to optimise, q is the number of constraints, c_{ij} are the coefficients' expected values and n is the number of decision variables.

2. Vectors $\mathbf{U}_1$ and $\mathbf{M}$ are defined as having the maximum and minimum values of the objective functions respectively. This means that vector $\mathbf{U}_1$ has the values of the maximisation of each objective separately (e.g. max $z_1(\mathbf{x})$, max $z_2(\mathbf{x})$, max $z_3(\mathbf{x})$, max $z_4(\mathbf{x})$, max $z_5(\mathbf{x})$), subject to constraints $g_q(\mathbf{x}) \le 0$. In addition vector $\mathbf{M}$ has the minimum values found following the same procedure, in other words, minimising each objective separately subject to the constraints.

$$\begin{aligned} &z_i(\mathbf{x}_i^*) = \max \quad z_i(\mathbf{x}), \qquad i \in I[1, p] \\ &\mathbf{U}_1 = \begin{bmatrix} z_1(\mathbf{x}_1^*) \\ z_2(\mathbf{x}_2^*) \\ \vdots \\ z_p(\mathbf{x}_q^*) \end{bmatrix} \\ &\mathbf{M} = \begin{bmatrix} z_{1\min} \\ z_{2\min} \\ \vdots \\ z_{3\min} \end{bmatrix} \end{aligned} \tag{3.32}$$

It is important to bear in mind that $\mathbf{M}$ may not exist in practice. Therefore, in such cases some other techniques have to be applied.

3. An initial surrogate function is formulated:

$$F(\mathbf{x}) = \sum_{i=1}^{p} G_i(\mathbf{x}) \tag{3.33}$$

where

$$G_i(\mathbf{x}) = \frac{z_i(\mathbf{x}) - z_{i\min}}{z_i(\mathbf{x}_i^{\bullet}) - z_{i\min}} \tag{3.34}$$

This surrogate function will be modified every time the algorithm is run.

The method also includes the use of multiple attribute utility functions used to introduce DM's preferences, and in the same way as some of the methods discussed in subsection 3.4.1, it uses pairwise comparisons between the alternatives. It is important to mention that PROTRADE is not used to mathematically represent the function to optimise, but the DM's preferences. This method will be fully discussed in Chapter Five together with a case study of multiple use land allocation.

Surrogate worth trade-off method (SWT)

Haimes (1998) defines the multi-objective function as follows:

$$\min_{\mathbf{x}\in X} \{f_1(\mathbf{x}), f_2(\mathbf{x}), \ldots, f_n(\mathbf{x})\} \text{subject to } \mathbf{x} \in X \tag{3.35}$$

where $\mathbf{x}$ is an n-dimensional vector of decision variables, and X is the set of feasible solutions. In other words, X consists of the vectors of decision variables $\mathbf{x}$ that meet the constraints defined as follows:

$$X = \{\mathbf{x} | g_i(\mathbf{x}) \le 0, i = 1,2,\ldots,m\} \tag{3.36}$$

Therefore the trade-off function is defined:

$$f_j(\mathbf{x}) \le \varepsilon_j, \; j = 2,3,\ldots,n \tag{3.37}$$

where ε_j are maximum tolerable levels. Using the generalised Lagrangian, L is found:

$$L = f_1(\mathbf{x}) + \sum_{j=2}^{n} \lambda_{1j}[f_j(\mathbf{x}) - \varepsilon_j] \tag{3.38}$$

where λ_{1j} are generalised Lagrange multipliers associated with the jth constraint, where the objective function is $f_1(\mathbf{x})$. Therefore λ_{ij} represents the generalised Lagrange multiplier associated with the ith objective function and the jth constraint. Defining Ω as the set of all λ_{ij}, where $j = 2, 3,\ldots, n$ that satisfies the Kuhn-Tucker conditions:

$$\lambda_{1j}[f_j(\mathbf{x}) - \varepsilon_j] = 0, \qquad \lambda_{1j} \ge 0; \; j = 2,3,\ldots,n \tag{3.39}$$

$$\lambda_{1j}(\varepsilon_j) = -\frac{\partial L}{\partial \varepsilon_j}, \; j = 2,3,\ldots,n \tag{3.40}$$

The value of λ_{1j} indicates the marginal benefit of the objective function $f_1(\mathbf{x})$.

The surrogate worth function W_{ij} acts as the interface between the DM and the mathematical model and it is defined below (Haimes, 1998):

$$W_{ij} = \begin{cases} > 0 & \text{when } \lambda_{ij} \text{ marginal units of } f_i(\mathbf{x}) \text{ are preferred over one marginal unit of } f_j(\mathbf{x}), \\ & \text{given the satisfaction of all objectives at level } \varepsilon_k, k = 1,2,\ldots,n \\ = 0 & \text{when } \lambda_{ij} \text{ marginal units of } f_i(\mathbf{x}) \text{ are equivalent to one marginal unit of } f_j(\mathbf{x}), \\ & \text{given the satisfaction of all objectives at level } \varepsilon_k, k = 1,2,\ldots,n \\ < 0 & \text{when } \lambda_{ij} \text{ marginal units of } \mathrm{f_i}(\mathbf{x}) \text{ are not preferred over one marginal unit of } \mathrm{f_j}(\mathbf{x}), \\ & \text{given the satisfaction of all objectives at level } \varepsilon_k, k = 1,2,\ldots,n \end{cases} \tag{3.41}$$

The W_{ij} value is a DM assessment using an ordinal scale, where $W_{ij} = 0$ implies indifference. When a degree of difference but not the specific amount of difference measures something, it is said that an ordinal scale is used. In particular, an ordinal scale could be to ask the DM to classify solutions as excellent, very good, good, fair, or poor.

The DM obtains the trade-off values through the trade-off function and these are associated with the Pareto optimal solutions also known as non-inferior solutions (Haimes, 1998). It is necessary to define the concept of the indifference band as a subset of the non-inferior set where to improve one objective function it is necessary to degrade another. Hence an optimum solution is understood as that non-inferior feasible solution that is an element of the indifference band.

The band of indifference is determined as follows:

Having two different values of λ_{ij}, the DM is asked to assess them using an ordinal scale and the $W_{ij}(\lambda_{ij})$ values are obtained. Then a linear interpolation of the two W_{ij} is made and λ_{ij}^* is the value over the line that gives a $W_{ij} = 0$. The indifference band exists in the neighbourhood of λ_{ij}^*, as shown in Figure 3.8.

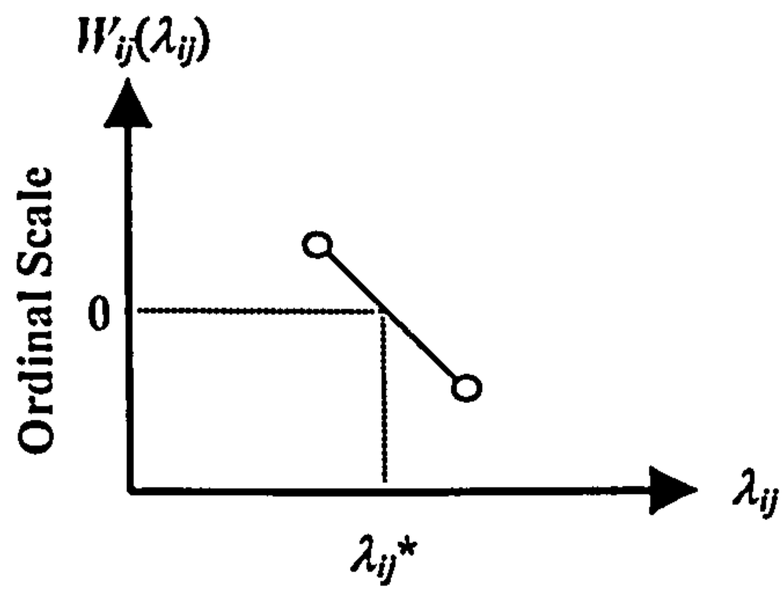


Figure 3.8 Indifference band at λ_{ij}^*
Adapted from Haimes (1998)

Thus, the surrogate function is used to assign a scalar value to the non-inferior solutions.

3.4.3 Fuzzy Sets and Decision Making

Carlsson (1984) clarifies the term "fuzziness" as meaning "vagueness" rather than "uncertainty". Kickert (1978) studied fuzzy set theory applied to the decision-making field.

Fuzzy theory, broadly speaking, is a mathematical theory and the term fuzziness is related in a certain way to uncertainty. In the past, probability theory was the only area of human knowledge that was involved with uncertainty; this is based on predictions of events, where to validate them the passage of time or testing is necessary (Terano et al. (1992)). For fuzzy theory the uncertainty must be understood as the ambiguity in the meaning of words, for

instance, "tall person" or "young person". This give everyone an involvement with uncertainty and consequently with fuzziness.

3.4.3.1 Basic Definitions of Fuzzy Sets

In the traditional set theory given a set A, a characteristic function f_A and a universe of discourse $X = \{x\}$, the function f_A will be described as follows (Kickert , 1978):

$f_A(x) = 0$ if and only if $x \notin A$
$f_A(x) = 1$ if and only if $x \in A$

These sets are called *ordinary sets* or *sets* (Zadeh, 1965).

In the theory of fuzzy sets, the characteristic function is generalised, allowing it to take values within the interval [0,1]. This means that the number of values that f_A can assume is infinite. Then, to define a fuzzy set, it is necessary to consider a membership function $\mu_A(x)$, which associates to each $x \in X$ a real number in the interval [0,1]. The value of $f_A(x)$ represents the grade of membership. This means that the closer the value of $f_A(x)$ is to one, the higher the grade of membership (Zadeh, 1965). In other words, a fuzzy set is a mapping from the universe of discourse into a closed interval ([0,1]).

Zadeh (1965) established that the nature of a fuzzy set is non-statistical, although the fuzzy set membership function could be similar to a probability function. The differences can be seen in the basic properties and combination rules of membership functions.
The following definitions were taken from the paper presented by Zadeh (1965):

Definition 3.1

A fuzzy set is empty if and only if its membership function is identically zero on X.

Definition 3.2

Two fuzzy sets A and B are equal:

$$A = B \text{ if and only if } f_A(x) = f_B(x), \forall x \in X \tag{3.42}$$

Definition 3.3

Complement. The complement of a fuzzy set A, A', is defined by:

$$f_{A'}(x) = 1 - f_A(x) \tag{3.43}$$

Definition 3.4

Containment. "A is a subset of B" is defined by:

$$A \subset B \Leftrightarrow f_A(x) \leq f_B(x) \tag{3.44}$$

Definition 3.5

Union. The union of two fuzzy sets A and B is a fuzzy set C, written as $C = A \cup B$, and is defined by:

$$f_C(x) = \text{Max}\,[f_A(x), f_B(x)], \qquad x \in X \tag{3.45}$$

The associative property for the union is:

$$A \cup (B \cup C) = (A \cup B) \cup C \tag{3.46}$$

Definition 3.6

Intersection. The intersection of two fuzzy sets A and B is a fuzzy set C, written as $C = A \cap B$ and is defined by:

$$f_C(x) = \text{Min}\,[f_A(x), f_B(x)], \qquad x \in X \tag{3.47}$$

The intersection also has the associative property.

Having defined the operations of complementation, union and intersection, it is possible to use some of the basic identities of ordinary sets in fuzzy sets such as De Morgan's and distributive laws (Zadeh, 1965).
These laws in terms of membership function will be represented as follows (the identities were verified by Zadeh, 1965):

De Morgan's laws:

$$(A \cup B)' = A' \cap B' \tag{3.48}$$

$$1 - \text{Max}\,[f_A(x), f_B(x)] = \text{Min}\,[1 - f_A(x),\ 1 - f_B(x)] \tag{3.49}$$

$$(A \cap B)' = A' \cup B' \tag{3.50}$$

$$1 - \text{Min}\,[f_A(x), f_B(x)] = \text{Max}\,[1 - f_A(x),\ 1 - f_B(x)] \tag{3.51}$$

Distributive laws:

$$C \cap (A \cup B) = (C \cap A) \cup (C \cap B) \tag{3.52}$$

$$\text{Min}\,[f_C(x), \text{Max}\,[f_A(x), f_B(x)]] = \text{Max}\,[\text{Min}\,[f_C(x), f_A(x)], \text{Min}\,[f_C(x), f_B(x)]] \tag{3.53}$$

$$C \cup (A \cap B) = (C \cup A) \cap (C \cup B) \tag{3.54}$$

$$\text{Max}\,[f_C(x), \text{Min}\,[f_A(x), f_B(x)]] = \text{Min}\,[\text{Max}\,[f_C(x), f_A(x)], \text{Max}\,[f_C(x), f_B(x)]] \tag{3.55}$$

3.4.3.2 Fuzzy Rule-Based Systems

For the purposes of this research it is necessary to define what a fuzzy rule-based system (FRBS) is and how it works. Mainly, the FRBS is applied to a fuzzy controller where desired states of the process are defined. The controller objective is to reach these desired states by adjusting the input values using a closed loop that goes from the system outputs to the system inputs. Generally speaking, fuzzy controllers are systems based on written rules (i.e. fuzzy rules) that relate the input variables with the control variables. To make this relation possible it is necessary to use linguistic variables.

Zadeh (1973) defines a linguistic variable "as that whose values are sentences in a natural or artificial language". These sentences are formed by labels (small, tall, young, old, large),

connectives (and, but, or), negation (not), and hedges (very, more or less, quite). Therefore a linguistic variable defines a physical variable and a fuzzy rule is constructed following a condition-action structure. The type of fuzzy rule used in this work is "*if* <condition> *then* <action>".

For example, if the variable to control is temperature, linguistically it can be expressed as "the temperature is too high" (condition) and the action to follow could be "decrease it by a bit". Thus, the rule is "*if* the temperature is too high *then* decrease it by a bit".

The process of defining all the rules and the fuzzy sets will be called fuzzy classification. For the fuzzy classification a "how to measure acceptability" process is defined. The measure process usually uses a deviation or an error to specify the fuzzy sets. The fuzzy sets are also defined linguistically using predicates. For example, they can be written as "the deviation is tolerable" and the "deviation is too high". Once the fuzzy sets are established, the membership functions of each of them are determined. The fuzzy classification includes not only the fuzzification process but also the rule-based system construction. To perform the control actions a defuzzification process is needed. This process is basically the transformation from linguistic variables to crisp variables.

To summarise, the fuzzy control has the following modules (Zimmermann, 1996):

1. Fuzzification
2. Rule base
3. Defuzzification

Figure 3.9 shows a fuzzy controller where the three modules mentioned above are represented. This fuzzy controller is known as "Mamdani" and it is explained in great depth in subsection 3.4.3.3.

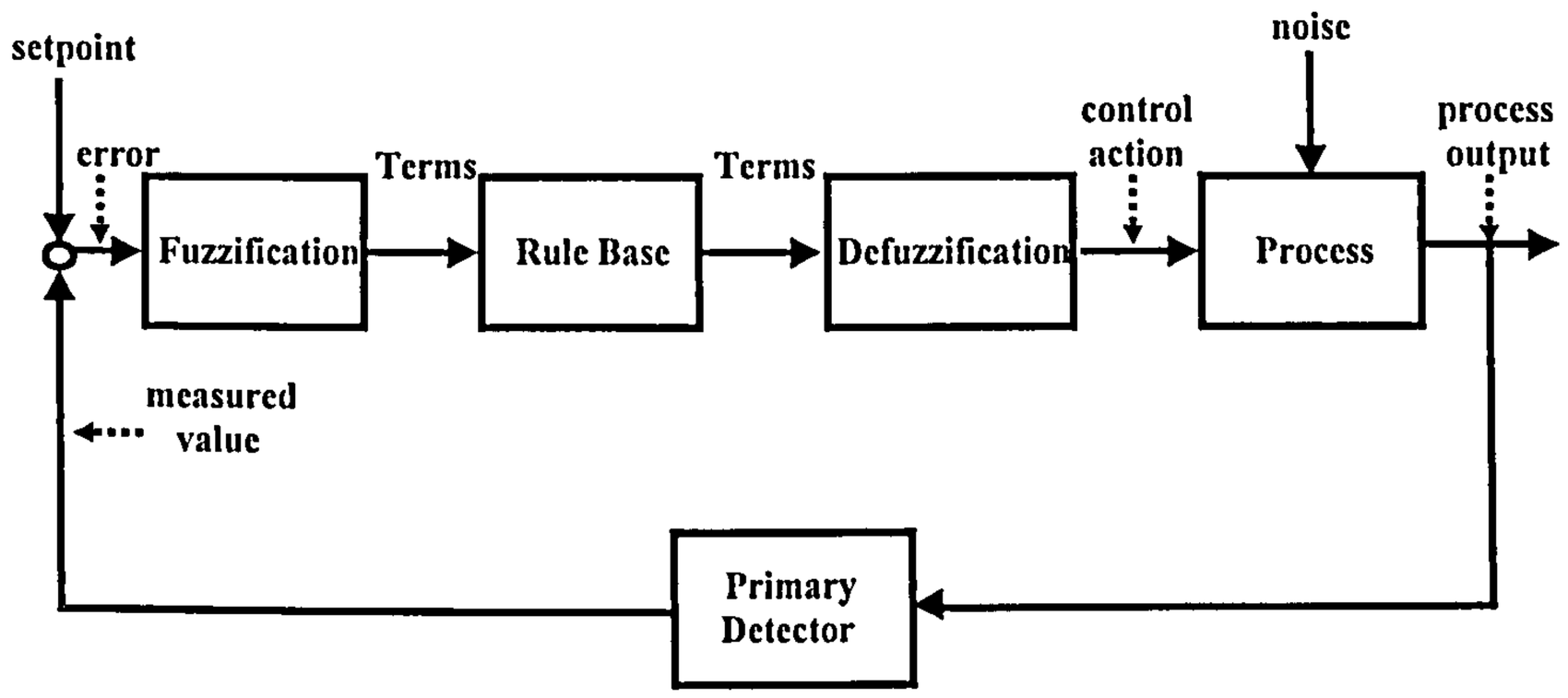


Figure 3.9 Mamdani Fuzzy Controller
Adapted from Zimmermann (1996)

3.4.3.3 Fuzzy Controllers

Mamdani controller

The Mamdani controller is based on the description of process states using linguistic variables as inputs to control rules. It is understood that a linguistic variable is a fuzzy set with a certain shape. This shape can be linear, triangular, trapezoidal, etc. Each linguistic variable consists of different terms and each term will have a membership function associated with it (Zimmermann, 1996). As mentioned above, when the linguistic variable "temperature" is used different terms can describe it: "low" (l), "comfortable" (c) and "high" (h). Figure 3.10 shows

how these terms are represented using a triangular shape. The terms of each linguistic variable are described by membership functions.

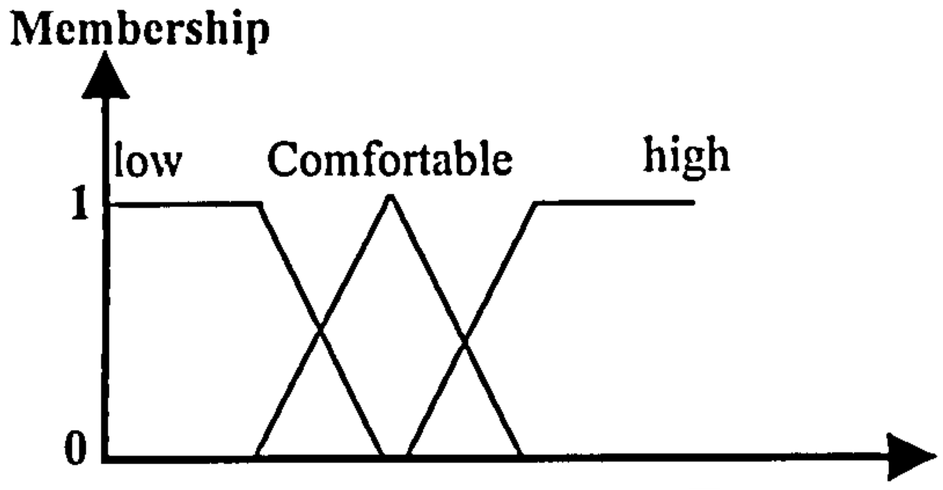


Figure 3.10 Graphical representation of a linguistic variable

Having said this, considering n as the number of linguistic variables and m as the number of terms for each variable, the membership function will be:

$$\mu_i^j(x) \tag{3.56}$$

where $i = 1, 2,\ldots, n$ indicates which linguistic variable is analysed and $j = 1, 2,\ldots, m$ which term of the linguistic variable i is used.

The number of rules is in function of the linguistic variables and the number of terms each of these has. Mamdani and Assilian (1975) implemented a fuzzy controller in a plant; this process involved a steam engine and a boiler combination. This controller considered four input variables and two output variables. Each of the input variables (pressure, speed, change in pressure, and change in speed) was defined as an error that measures the difference between the present variable value and the set point or the difference between the present value and the last. The outputs were defined as actions to follow (heat change and throttle change) and seven fuzzy subsets were mapped to points in a universe of discourse (positive big, positive medium, positive small, nil, negative small, negative medium and negative big). The rules used in this problem utilised fuzzy conditional statements like "*If* the pressure error is negative small *then* the heat change is positive small" or "*If* the speed error is negative big *then* the throttle change is positive big". With this example it is possible to visualise that the rules connect the input variables with the output variables. In general the rule has the following form:

$$\text{if } x_1 \text{ is } A_1^{j_1} \text{ and} \ldots \text{and } x_n \text{ is } A_n^{j_n} \text{ then } u \text{ is } A^j \tag{3.57}$$

where the antecedents are fuzzy and the consequent is fuzzy.

Sugeno controller

The idea of this controller is based on that of Mamdani's even though some modifications are made. Sugeno defines the fuzzy control rules as *antecedent-consequent* where the consequent is a crisp function and the antecedent is fuzzy. For this reason with this kind of controller the procedure of defuzzification is avoided (Zimmermann, 1996). In order to determine the membership of each input the rule is defined as follows:

$$\text{if } x_1 \text{ is } A_1^{j_1} \text{ and} \ldots \text{and } x_n \text{ is } A_n^{j_n} \text{ then } u \text{ is } f_r(x_1, \ldots x_n) \tag{3.58}$$

Using the temperature example, having two fuzzy antecedents like temperature and change of temperature and a consequent that is a linear function that will control the power in a heating system, then the rule can be built as follows:
if the temperature is high and the change of temperature positive small then power = temp-25.

3.4.3.4 Design of Fuzzy Controllers

Zimmermann (1996) states that, in order to design a fuzzy controller it is necessary to consider three main aspects:

1. Scaling factors. A scaling factor is defined for each rule and basically scales the base variables of the linguistic variables. Therefore the variable used in the rule is defined as $x' = s \cdot x$, where s is the scaling factor.

2. Fuzzy sets. First of all the shape of the fuzzy sets has to be determined, for example, this can have a trapezoidal shape. The peak value or modal value is where the membership function is equal to one. The cross point is the value of the variable where two membership functions have the same membership value (must be different to zero). The width is the value of the variable where the membership function of study has a value of zero on the left or right side of the modal value. Then a membership function is symmetric if the right and the left width are equal. The cross point level is the value of the membership function at the cross point. Therefore a cross point ratio can be defined as the number of cross points two membership functions have. Figure 3.11 illustrates all the values described above using three fuzzy sets with trapezoidal shapes.

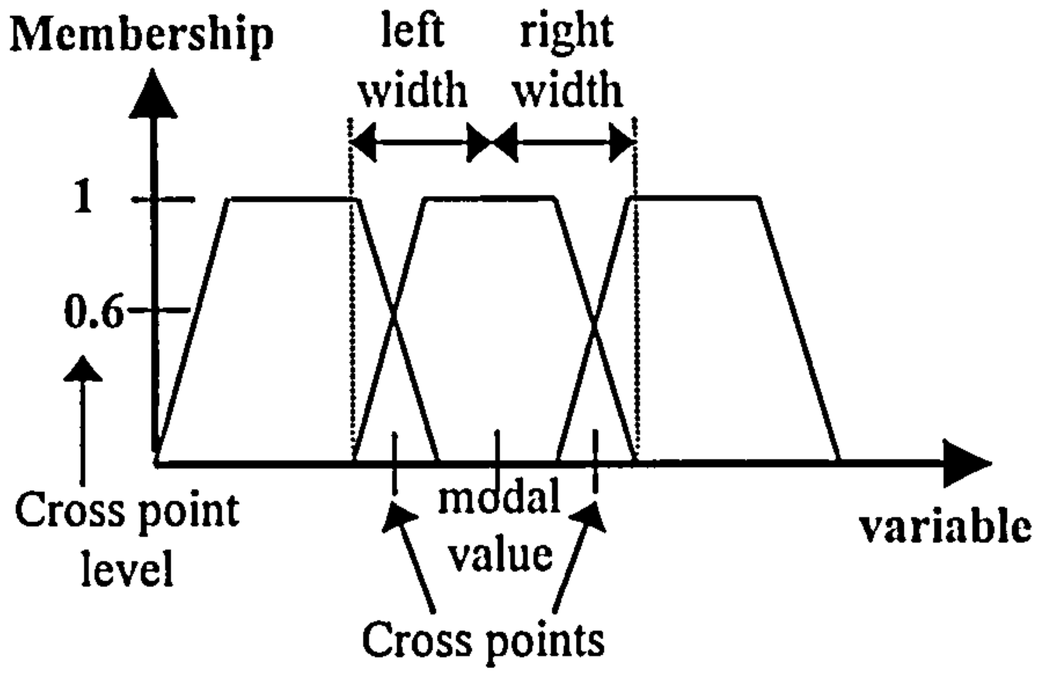


Figure 3.11 Fuzzy set parameters
Adapted from Zimmermann (1996)

3. Rules. Terano et al. (1992) state that the process of determining the control rules should be divided into two parts: determination of the antecedent and determination of the consequent.

In Chapter Four of this thesis the FRBS will be used to represent the preferences of the DM to have an automated decision-making process in the solution of a multiple objective optimisation problem.

3.4.3.5 Defuzzification Methods

The main objective of the defuzzification process is to transform a fuzzy set into a crisp number. In fuzzy control this process is understood as the mapping from a domain set of

fuzzy control actions into a set of crisp control actions (Lee, 1990). Nevertheless there is not an established procedure to select a defuzzification method. For that reason the most widespread methods are presented below:

Max-Membership method (MAX)

The crisp number resultant w_c is the one that corresponds to the maximum membership function value as is shown in Figure 3.12.

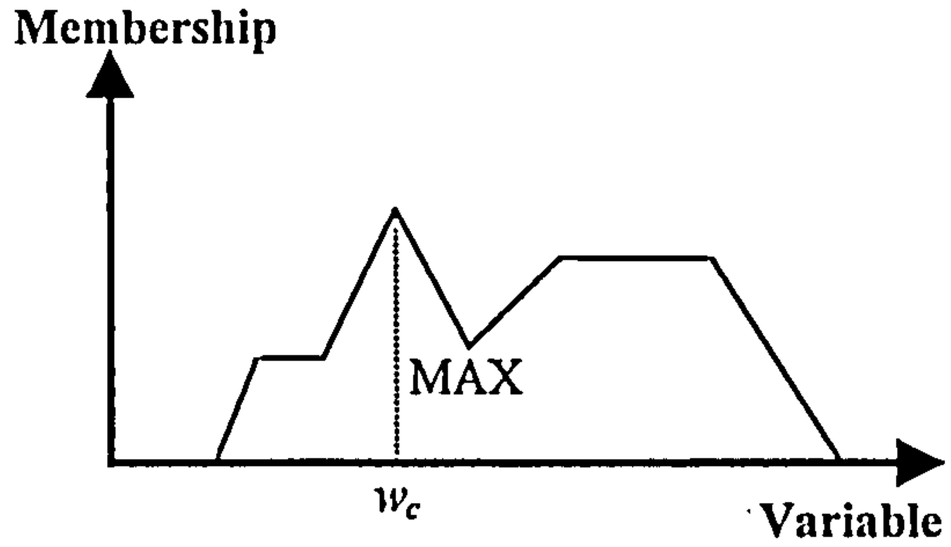


Figure 3.12 Max-membership defuzzification strategy

w_c is selected such that $\forall\ w \in U$, $\mu_A(w_c) \geq \mu_A(w)$, where U is the universe of discourse, and $\mu_A(w_c)$ and $\mu_A(w)$ are the values of the membership function evaluated in w_c and w respectively.

Centre of Area method (COA)

This method is also known as "Centre of Gravity" and its objective is to select the control action w_c that represents the centre of the shaded area shown in Figure 3.13.

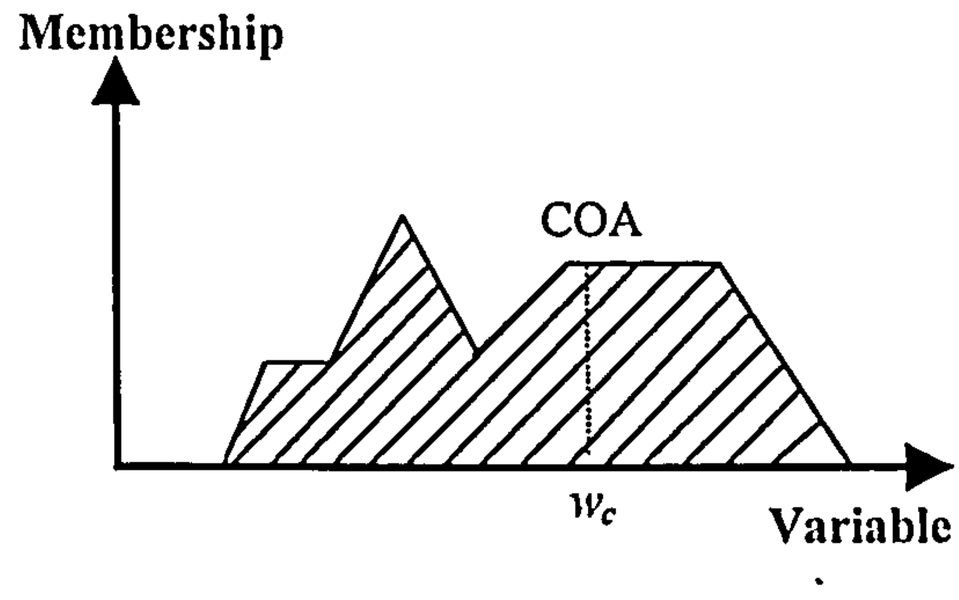


Figure 3.13 Centre of Gravity defuzzification strategy

w_c can be calculated in two ways (continuous case and discrete case):

$$w_c = \begin{cases} \dfrac{\int \mu_A(w) \cdot w \cdot dw}{\int \mu_A(w) \cdot dw} & \textit{continuous} \\ \dfrac{\sum_{i=1}^{n} \mu_A(w_i) \cdot w_i}{\sum_{i=1}^{n} \mu_A(w_i)} & \textit{discrete} \end{cases} \qquad (3.59)$$

$n =$ number of quantization levels

Mean of Maximum method (MOM)

This method generates the control action w_c that represents the mean value of all variable values that reach the maximum of the membership function (Figure 3.14).

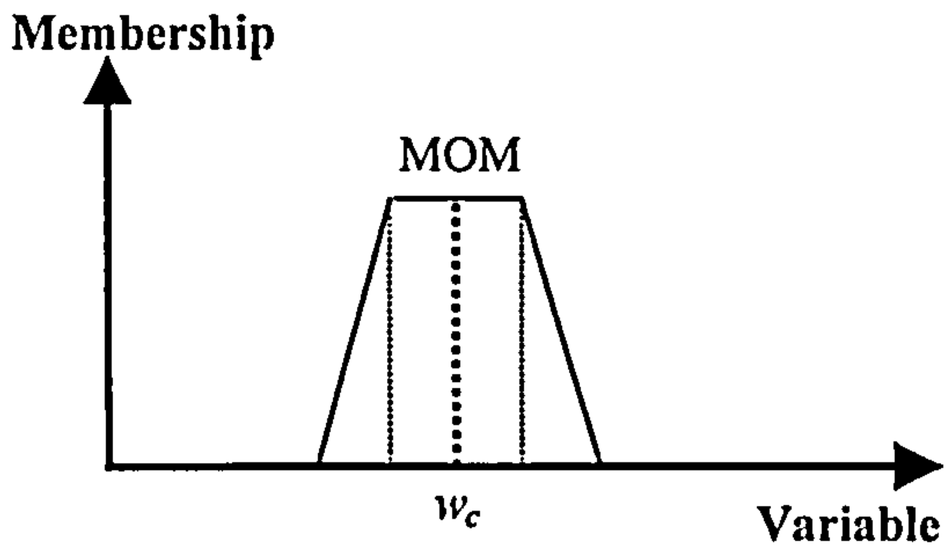


Figure 3.14 Mean of Maximum defuzzification strategy

3.5 Multiple Objective Evolutionary Optimisation (MOEO)

The following definitions are used to formally describe multiple objective optimisation problems (MOP):

Definition 3.7

Objectives are represented by mathematical functions and they could be to minimise cost or maximise profit. These functions could be linear or non-linear. Normally they are represented as follows:

$$\textit{maximise } f(x) \qquad (3.60)$$

$$\textit{minimise } f(x) \qquad (3.61)$$

Definition 3.8

A general multi-objective MOP is defined starting from the basis that the main purpose is to find the maximum/minimum solution for a certain problem, having a decision variable vector x of dimension n and k objectives the problem is represented as:

$$\textit{max/min } [f_1(x), f_2(x), \ldots, f_k(x),] \qquad (3.62)$$

subject to:

$$g_i(x) \leq 0, \ i = 1, 2, \ldots, m \tag{3.63}$$

where m is the number of constraints (Hwang and Masud, 1979).

Definition 3.9

A non-dominated solution exists where it is not possible to improve one objective without decreasing the other objectives. In a formal way x^* is a non-dominated solution if and only if there is not any $x \in X$ (where X is the feasible set of variables that satisfies the constraints) such that $f_i(x^*) \leq f_j(x)$ for all i, and $f_j(x^*) < f_j(x)$ for at least one j (Hwang and Masud, 1979).

Definition 3.10

The members of the population that are non-dominated constitute the Pareto optimal set. The applicability of these concepts in practice depends on how much information is needed from the DM during the computation of Pareto optimal solutions.

As defined in section 3.4, one of the main characteristics of multiple objective optimisation problems is that the search space is continuous. Therefore binary-coded GAs (discussed in Chapter Two) not always are good for solving MOPs because the string length has to be defined in advance for precision purposes. For this reason there exists a number of real-parameter GAs (Deb, 2001), where there is no need of a string coding. This kind of GA is better known as real-valued GA, and there exist a number of crossovers and mutation operators specially defined for them.

3.5.1 Real-Valued Genetic Algorithms

In this section the *Real-Valued Genetic Algorithms*, or *Real-Coded Genetic Algorithms* (RCGA) as Herrera et al. (1998) called them, will be defined. The first approach to real-valued representation was developed for ES, as was discussed in Chapter Two of this thesis.

Herrera et al. (1998) define the RCGA's chromosome as a vector of floating point numbers with the same size as the solution's vector. Each gene of the chromosome represents a variable of the problem and its value remains within the interval previously defined for each variable.

One of the most important characteristics of RCGA is that the domain can be bigger and can have greater precision than that represented with binary-coding. Understanding the *genotype* as the coding and the *phenotype* as the search space, it is clear that in RCGA there is no difference between them. Another important characteristic of RCGA is that not having a coding-decoding process increases the speed of the GA.

Broadly speaking, RCGA cannot use the same operators as binary-coded GAs. Therefore for a better understanding of RCGA it is important to define their correspondent genetic operators (crossover and mutation).

3.5.1.1 Crossover Operators

Linear crossover

This crossover generates three solutions from two parents. Having parents P_1 and P_2, the solutions are as follows:

$$\begin{aligned} &0.5(P_1 + P_2) \\ &(1.5\,P_1 - 0.5\,P_2) \\ &(-0.5\,P_1 + 1.5\,P_2) \end{aligned} \tag{3.64}$$

Naïve crossover

This crossover is very similar to the one used in binary-coded GA because a cross point is selected (single-point, two-point, or n-point) to produce the offspring. For instance, having two parents, n decision variables x_{ij} (where $i = 1, 2,\ldots, n$ and $j = 1, 2$), and a cross site at the fourth position produces the following offspring:

$$\begin{aligned} P_1&: \; x_{11} \;\; x_{21} \;\; x_{31} \;\; x_{41} \;\; x_{51} \;\; \cdots \;\; x_{n1} \\ P_2&: \; x_{12} \;\; x_{22} \;\; x_{32} \;\; x_{42} \;\; x_{52} \;\; \cdots \;\; x_{n2} \\ O_1&: \; x_{11} \;\; x_{21} \;\; x_{31} \;\; x_{41} \;\; x_{52} \;\; \cdots \;\; x_{n2} \\ O_2&: \; x_{12} \;\; x_{22} \;\; x_{32} \;\; x_{42} \;\; x_{51} \;\; \cdots \;\; x_{n1} \end{aligned} \tag{3.65}$$

The search power of this crossover is not so good.

Blend crossover

Having two parent solutions $x_i^{(1,t)}$ and $x_i^{(2,t)}$, where t represents the generation number, and having a range $[x_i^{(1,t)} - \alpha(x_i^{(2,t)} - x_i^{(1,t)}),\ x_i^{(2,t)} + \alpha(x_i^{(2,t)} - x_i^{(1,t)})]$, then a solution is randomly selected within this range. In other words, having a randomly selected number $u_i \in [0,1]$, the offspring is defined:

$$x_i^{(1,t+1)} = (1 - \gamma_i)\, x_i^{(1,t)} + \gamma_i\, x_i^{(2,t)} \tag{3.66}$$

where $\gamma_i = (1 + 2\alpha)u_i - \alpha$. Deb (2001) suggests a value for $\alpha = 0.5$.

Arithmetical crossover

Michalewicz (1994) defines this operator as "a linear combination of two vectors", having two parents x_v^t and x_w^t to be crossed, where t is the generation number. The resultant offspring is defined by:

$$\begin{aligned} x_v^{t+1} &= a \cdot x_w^t + (1 - a) \cdot x_v^t \\ x_w^{t+1} &= a \cdot x_v^t + (1 - a) \cdot x_w^t \end{aligned} \tag{3.67}$$

where the parameter a can be either constant or variable. If a is constant the operator is called *uniform arithmetical crossover*, if not the operator is called *non-uniform arithmetical crossover*.

Simulated binary crossover (SBX)

This crossover was developed by Deb and some of his students in 1995 (Deb, 2001), and is defined as follows:

Having two parent solutions $x_i^{(1,t)}$ and $x_i^{(2,t)}$, where t represents the generation number, to compute the offspring $x_i^{(1,t+1)}$ and $x_i^{(2,t+1)}$ the following procedure is performed:

Step 1: Select a random number $u_i \in [0,1)$

Step 2: Calculate β_{qi} using the following equation:

$$\beta_{qi} = \begin{cases} (2u_i)^{\frac{1}{\eta_c+1}}, & \text{if } u_i \le 0.5; \\ \left(\frac{1}{2(1-u_i)}\right)^{\frac{1}{\eta_c+1}}, & \text{otherwise} \end{cases} \tag{3.68}$$

Step 3: Calculate the offspring using the following equations:

$$x_i^{(1,t+1)} = 0.5[(1+\beta_{qi})x_i^{(1,t)} + (1-\beta_{qi})x_i^{(2,t)}] \tag{3.69}$$

$$x_i^{(2,t+1)} = 0.5[(1-\beta_{qi})x_i^{(1,t)} + (1+\beta_{qi})x_i^{(2,t)}] \tag{3.70}$$

Unfair Average Crossover

This operator was developed by Nomura and Miyoshi (1995) and is defined as follows:

Having two parent solutions $x_i^{(1,t)}$ and $x_i^{(2,t)}$, where t represents the generation number, to compute the offspring $x_i^{(1,t+1)}$ and $x_i^{(2,t+1)}$ the following equations are applied:

$$x_i^{(1,t+1)} = \begin{cases} (1+\alpha)x_i^{(1,t)} - \alpha x_i^{(2,t)}, & \text{for } i = 1,\ldots,j, \\ -\alpha x_i^{(1,t)} + (1+\alpha)x_i^{(2,t)}, & \text{for } i = j+1,\ldots,n \end{cases} \tag{3.71}$$

$$x_i^{(2,t+1)} = \begin{cases} (1-\alpha)x_i^{(1,t)} + \alpha x_i^{(2,t)}, & \text{for } i = 1,\ldots,j, \\ \alpha x_i^{(1,t)} + (1-\alpha)x_i^{(2,t)}, & \text{for } i = j+1,\ldots,n \end{cases} \tag{3.72}$$

where j is a integer parameter randomly selected between 1 and n, that indicates the cross site.

Fuzzy Connectives Based (FCB)

This crossover was developed in 1994 by Herrera et al. (1998) and consists of considering each bounded variable x_i (gene) domain $[x_i^L, x_i^U]$, dividing it into four regions: $[x_i^L, x_i^{1,t}]$, $[x_i^{1,t}, x_i^{2,t}]$, $[x_i^{2,t}, x_i^U]$, where the fourth region is an overlapping region (y_i^1, y_i^2), where $y_i^1 \le x_i^{1,t}$ and $y_i^2 \ge x_i^{2,t}$. After the regions have been identified, one solution has to be selected from each of them; user-defined fuzzy connective functions make the selection. Two normalised parents are defined:

$$s = \frac{x_i^{1,t} - x_i^L}{x_i^{2,t} - x_i^{1,t}} \tag{3.73}$$

$$s' = \frac{x_i^{2,t} - x_i^L}{x_i^{2,t} - x_i^{1,t}} \tag{3.74}$$

Four fuzzy connectives (T, G, P, and $\hat{C}$) are defined:

$$x_i^{1,t+1} = x_i^L + (x_i^U - x_i^L)\mathrm{T}(s,s') \tag{3.75}$$
$$x_i^{2,t+1} = x_i^L + (x_i^U - x_i^L)\mathrm{G}(s,s') \tag{3.76}$$
$$x_i^{3,t+1} = x_i^L + (x_i^U - x_i^L)\mathrm{P}(s,s') \tag{3.77}$$
$$x_i^{4,t+1} = x_i^L + (x_i^U - x_i^L)\hat{\mathrm{C}}(s,s') \tag{3.78}$$

where $\mathrm{T}(s, s') = \min(s, s')$, $\mathrm{G}(s, s') = \max(s, s')$, $\mathrm{P}(s, s') = (1-\lambda)s+\lambda s'$, and $\hat{\mathrm{C}}(s, s') = \mathrm{T}^{1-\lambda}\mathrm{G}^{\lambda}$. After this procedure is performed, four solutions are generated, two of them directly replace the parents, and the other two replace two solutions randomly selected from the population waiting to be crossed.

3.5.1.2 Mutation Operators

Random mutation

Michalewicz (1994) defines this operator as that applied to a floating point number and the result is a random value from the domain <LB,UB>, where LB, UB are the lower and upper bounds respectively. This operator can be considered as analogous to the one used in binary-coded GAs.

Uniform mutation

This operator is defined as follows (Michalewicz, 1994):

Having a chromosome $s_v^t = \langle v_1, \ldots, v_n \rangle$ where n is the number of variables, t is the generation number, and having selected the element v_k for mutation (where each element of the chromosome has the same probability of being selected), then the new offspring is defined as:

$$s_v^{t+1} = \langle v_1, \ldots, v'_k, \ldots, v_n \rangle \quad \text{where} \quad 1 \le k \le n \tag{3.79}$$

where v'_k is a randomly generated value within the variable's domain.

Non-Uniform mutation

This operator was developed by Michalewicz (1994), and is defined as follows:

Having the following chromosome $s_v^t = \langle v_1, \ldots, v_n \rangle$ where n is the number of variables, t is the generation number, and having selected the element v_k for mutation then the new offspring is defined as follows:

$$s_v^{t+1} = \langle v_1, \ldots, v'_k, \ldots v_n \rangle \tag{3.80}$$

where v'_k is defined as follows:

$$v'_k = \begin{cases} v_k + \Delta(t, UB - v_k) & \text{if a random digit is 0,} \\ v_k - \Delta(t, v_k - LB) & \text{if a random digit is 1} \end{cases} \tag{3.81}$$

where LB and UB are lower and upper domain bounds of the variable v_k. Michalewicz (1992) defines the function $\Delta(t, y)$ as:

$$\Delta(t, y) = y \cdot \left(1 - r^{\left(1-\frac{t}{T}\right)^b}\right) \tag{3.82}$$

$\Delta(t, y)$ returns a value in the range $[0, y]$ and r is a random number in [0,1], b is the degree of dependency on iteration number ($b = 5$, normally), and T is the maximal generation number.

This operator is called non-uniform because it searches the space uniformly when t is small, and locally when t is increased.

Normally distributed mutation

This operator is based on the use of a zero-mean Gaussian probability distribution (Deb, 2001), represented below:

$$y_i^{(1,t+1)} = x_i^{(1,t+1)} + N(0, \sigma_i) \tag{3.83}$$

where σ_i is a fixed parameter defined by the user (DM). This operator is very similar to that discussed in Chapter Two for ES.

Polynomial mutation

In this operator the probability distribution used is polynomial (Deb, 2001):

$$y_i^{(1,t+1)} = x_i^{(1,t+1)} + (x_i^{(u)} - x_i^{(L)})\,\bar{\delta}_i \tag{3.84}$$

where the parameter $\bar{\delta}_i$ is calculated from the polynomial probability distribution represented by equation 3.81 and 3.82:

$$P(\delta) = 0.5(\eta_m + 1)(1 - |\delta|)^{\eta_m} \tag{3.85}$$

$$\bar{\delta}_i = \begin{cases} (2r_i)^{1/(\eta_m+1)} - 1, & \text{if } r_i < 0.5, \\ 1 - [2(1-r_i)]^{1/(\eta_m+1)}, & \text{if } r_i \geq 0.5 \end{cases} \tag{3.86}$$

η_m should have a fixed value. The parameter $\bar{\delta}_i$ distribution is very similar to that of the non-uniform mutation operator. Moreover, another similarity between the polynomial mutation operator and the non-uniform mutation operator is that both are used for handling bounded decision variables.

3.5.2 Developments of MOEO

Schaffer (1985) proposed a vector evaluated genetic algorithm (VEGA) for the solution of multi-objective optimisation problems. This algorithm was based on the generation of a number of sub-populations at each generation. In order to generate the sub-populations the proportional selection operator was performed for each objective. Therefore, the number of sub-populations is equal to the number of objectives and the sub-populations size is the population size divided by the number of objectives. Additionally, all the elements of the sub-populations are set together and shuffled, generating a new population. This new population is used for the application of the crossover and mutation operators.

In 1987 one of the first attempts to solve multi-modal function optimisation problems using GAs was made by Goldberg and Richardson (1987). They developed the method of sharing functions. This method permits the formation of stable subpopulations of different strings. Through the use of sharing functions, niche-like and species-like (Goldberg, 1989) subdivisions of the environment and population are formed. Sharing allows the maintenance of a more diverse population and avoids a premature convergence. Goldberg and Richardson (1987) pointed out that the premature convergence of the simple GAs was a normal characteristic of these algorithms and usually drives to points that are not close enough to the optimal points. The results obtained with the GA using the sharing function were satisfactory.

Fonseca and Fleming (1993) proposed a rank-based fitness assignment method for multiobjective optimisation GA (MOGAs) implementing the idea proposed by Goldberg (1989). Treating the problem as a MOP, the solution expected is a family of points (Pareto-optimal solution). The Pareto optimal was necessary to assign rank-based fitness in terms of non-dominated individuals giving a rank of 1 to the best and a rank n to the worst. Additionally, the average fitness of individuals with the same rank was calculated. Fonseca and Fleming (1993) introduced a new parameter in the GA called the "niche size" implementing fitness sharing in the objective value domain. This method was implemented for a Pegasus gas turbine engine application.

Horn and Nafpliotis (1993) proposed an algorithm to find the Pareto optimal set called Niched Pareto GA. They worked at the same time as Fonseca and Fleming but independently. Horn and Nafpliotis (1993) proposed a Pareto domination tournament to give more domination pressure. A comparison set is selected randomly from the initial population, then two individuals are selected at random and each of these individuals is compared to every individual in the comparison set. This comparison leads the process to the following cases: 1. One of the two individuals is nominated by the comparison set, the other is non-dominated against the comparison set, and the non-dominated is selected; and 2. Neither of the two individuals is dominated by the comparison set, the use of sharing is needed. The sharing process defined is the degradation of an individual fitness having a niche count and a fitness function. Golberg (1992) proposed to sample the population to estimate the niche count - this was called niche count sampling. Both techniques were used in their Niched Pareto GA, showing the algorithm appropriate for finding the Pareto set and maintaining a large number of subpopulations.

Fonseca and Fleming presented an overview of EA in MO in 1995. In this review the multiobjective approaches using EAs developed at that time were discussed. The concept of Pareto optimal is the set of objective vectors that is non-dominated; this is considered the base of most of the approaches to solve multiobjective optimisation problems. Basically three techniques were used: 1. Pareto-based approaches, 2. Niche induction techniques, and 3. Pareto-based in conjunction with Niche induction techniques. On the other hand, Fonseca and Fleming (1995) concluded that one of the most important characteristics of the multiobjective evolutionary approach is the DM's intervention in the search and preferences setting.

One of the areas that has been developed in GA is the optimisation of design problems. Gero and Louis (1995) used the Pareto optimality to solve different design problems with shape grammars. Shape grammars is a formal method of shape generation used in architecture and is usually a set of grammatical rules used for mapping one shape into a different shape. A GA was developed using crossover to generate new grammars and then recombination to allow the restructuring of the problem formulation. In this way, it was possible to have an evolving state space, which could drive the problem formulation to an automatic reformulation of the optimisation problem.

Coello and Christiansen (1995) developed a technique that combines the global criterion method and GA. This technique places all the objectives at the same level of importance and

it is possible to solve either minimisation or maximisation problems. The approach used was the global criterion equation to generate the set of non-dominated solutions giving a certain weight to each objective. The GA used tournament selection and two-point crossover. Moreover, the algorithm was applied in the solution of the design of an I-beam and the design of a machine tool spindle obtaining good results in both cases.

Shaw and Fleming (1996) solved a real-life scheduling problem for a manufacturer of chilled ready meals for sale in supermarkets. The problem identified was the allocation of the orders to the production lines. The manufacturer needed a flexible system, which allowed rescheduling and change of priorities during manufacturing. They used the MOGA proposed by Fonseca and Fleming. The number of objectives to minimise was three. The edge recombination operator and splice mutation operator were used. The weighted sum GA (standard single-objective) was programmed in order to compare with MOGA. The results found by MOGA were better in terms of effectiveness.

Additionally, Camponogara and Talukdar (1997) developed a GA for constrained and multiobjective optimisation. A Pareto scheme was proposed. This scheme consisted of two roles: evolution selection and combination operator. The evolution selection consisted of keeping only non-dominated solutions from one generation to another. Whilst the combination operator consisted of creating offspring from the solutions in Pareto sets used to calculate improvement by searching the line defined by them. The main characteristic of the GA was to convert the constrained optimisation problem into an unconstrained multiobjective problem. Managing the constraints as objectives to minimise. In this way the problem was treated as a multiobjective problem minimising all the objectives.

Lis and Eiben (1997) proposed a Multisexual Genetic Algorithm (MSGA) based on giving sex or gender to individuals. In the recombination process one individual of each sex is used. It is necessary to map from each optimisation criteria to one sex in order to assign the sexes. In such a way that it is expected to have as many sexes as criteria. One important characteristic of this algorithm is the use of multi-parent crossover that performs recombination with one parent of each sex. Each sex will be ranked separately. The results showed that MSGA has a high capacity for finding Pareto optimal solutions.

Coello (1998) introduced a new multiobjective optimisation technique using a GA based on the min-max optimum concept. The main idea of this technique is to ensure all the individuals created are feasible solutions avoiding the violation of any constraint. A weight vector is provided in order that there are as many processes as weight combinations. Then the min-max approach is used with a weight combination for generating a single solution to formulate the Pareto set. This technique transforms the multiobjective problem into single objective problems. The problem solved with this multiobjective optimisation technique was the design of a Machine Tool Spindle. The technique showed better overall results and proved to be very robust.

In 1999, Hiroyasu et al. (1999) presented a new model of MOGA using parallel processing. The model is called Divided Range GA (DRGA) where the individuals are separated into sub-populations based on their objective function values. This model is a distributed GA using the migration operation. This operation consists of the movement of individuals randomly selected into other islands. DRGA have three abilities: local search, global search, and efficient search. In the overall process, a crossover called gravity crossover is used with no mutation operator. A different terminal condition than the number of generations is needed. The result showed that DRGA is an effective model in finding the Pareto solutions.

Zitzler and Thiele (1999) introduced a new evolutionary approach to multiple criteria optimisation called strength Pareto evolutionary algorithm (SPEA). This algorithm consists of storing the non-dominated solutions found externally, assigning scalar fitness values to

individuals, and performing clustering to reduce the number of stored non-dominated solutions. First, the non-dominated solutions in the external set are ranked, and then the fitness assignment procedure is applied. This procedure consists of two steps: each solution is assigned a real value, called strength, this value is proportional to the number of members plus one; the fitness of the individual is calculated by adding the strength values of the external non-dominated solutions. This algorithm is compared to other EAs achieving the best assessment.

Srinivas and Deb (1994) proposed the non-dominated sorting GA (NSGA). They showed that the non-dominated sorting has a high computational complexity. Deb et al. (2000) suggested a non-dominated sorting based multiobjective GA (NSGA-II) to avoid this complexity. They proposed a fast non-dominated sorting approach that returns a list of the non-dominated fronts. The algorithm proposed reduces the complexity from $O(mN^3)$ to $O(mN^2)$. After the application of this sorting it is necessary to estimate the density of solutions surrounding a point with the *crowding distance*. Additionally, a crowded comparison operator is used. This operator guides the selection process. The results showed a very good performance for the algorithm.

In 2000, Laumanns et al. (2000) presented a unified model of multiobjective evolutionary algorithms with elitism. The elitism term was defined as follows: individuals from the gene pool of population could not be discarded to allow worse individuals to participate. A unified model for multiobjective evolutionary algorithms was defined. This model consists of using probabilistic operators though in some cases the operators will be completely deterministic. Initially, an archive of elite individuals and a normal offspring population will be used to generate the current offspring population. The selection of the elite individuals is based on a probabilistic operator that determines whether an individual is chosen to function as a parent or not. In addition, the elitism strategy determines which individuals must be stored in order to update the archive of elite individuals. Recombination and mutation are applied in the generation of the new offspring population. The elitism technique was included in three MOEAs showing a regular behaviour. The authors concluded that this model is beneficial to speed up convergence to the Pareto set.

Knowles and Corne (2000) introduced a Pareto archived evolution strategy ((1+1)-PAES). This algorithm is divided into three parts: 1. A "candidate solution generator", that acts as simple random mutation hillclimbing; it maintains a "single current solution" and produces a new candidate using random mutation. 2. A "candidate solution acceptance", that consists of using a comparison set to select between the mutated and the current solution. 3. A non-dominated solution archive, which records all the non-dominated solutions previously found. They also proposed an "adaptive grid algorithm" to maintain diversity.

Sarker et al. (2002) presented a new multi-objective evolutionary algorithm (MEA). This algorithm is called $(\mu + \lambda)$ MEA and consists of discarding the dominated individuals in each generation. In other words, in order to calculate the number of offspring in each generation, the number of non-dominated solutions is multiplied by a fixed ratio. This means that the population size in each generation varies. The algorithm uses a real-valued codification, discrete recombination and Gaussian mutation. The performance of the MEA is compared with that of strength Pareto evolutionary algorithm (SPEA) in five benchmark problems, where MEA outperforms SPEA on all five occasions.

It is important to mention that this survey is an attempt to show the recent developments in this area but it does not cover all the work that has been done until now.

3.6 Summary

This chapter's main focus is on decisions, their nature and the way they are made. The decision analysis concept has been defined as a formal mechanism that helps to integrate the results of each problem after dividing it into a set of smaller problems. It helps the DM to have a better understanding of the problem and gives the tools for solving it analytically. It is also an approach to handling the decision-making process under risk and uncertainty.

Decision analysis is focused on the decision-making process, the DM and the decision. The decision-making process can be summarised as the observation and recognition of a problem, identification of alternatives, establishment of evaluation criteria, alternatives evaluation, selection of an alternative and implementation.

The risk analysis procedure is defined as risk identification, risk quantification and measurement, risk evaluation, risk acceptance or avoidance, and risk management. Some applications of risk management have been outlined, such as decision-making on energy, environmental and health policies, operations management, finances, marketing, engineering, and even risk analysis for terrorist attacks.

MCDM is a branch of decision-making and it is divided into two classes of models: multiple objective decision-making and multiple attribute decision-making. This division is based on the type of decision spaces; these are continuous or discrete. The methods to solve MCDM problems can be classified in deterministic, stochastic and fuzzy. These methods have been described in subsections 3.4.1, 3.4.2 and 3.4.3.

Finally, the core of this research (multiple objective evolutionary optimisation) has been discussed, presenting different developments on evolutionary algorithms with different genetic operators, different representation (binary, real) and different applications.

Once the basic concepts have been defined and studied, the next step is the solution of MODM and MADM problems. In Chapters Four, Five and Six three case studies and a practical problem are solved using multiple objective/attribute evolutionary optimisation.

CHAPTER 4

Sequential Multi-objective Problem Solving (SEMOPS)

4.1 Introduction

As discussed in the previous chapter, several methodologies for the optimisation of single-objective problems exist, although in practice, problems are often multi-objective. This implies that multiple objective problems (MOPs) have to be reformulated into single-objective problems or in the best case a method has to be used that allows the problems to be treated as multiple objective, without modifying them. One important characteristic of multiple objective optimisation problems is that a very large set of solutions is acceptable. These solutions are considered equivalent. Coello (2000) states that to solve MOPs, three stages are needed: measurement, search, and decision-making. Hwang and Masud (1979) defined the concept of decision-making as the selection process of an alternative to follow from all the alternatives available.

This chapter focuses on two new approaches. The first approach principally consists of adding an evolutionary algorithm to a *sequential multi-objective problem solving* (SEMOPS) method developed by Monarchi et al. (1973). The main objective of this hybrid approach is to handle the DM's preferences in an interactive (progressive) way, using the strengths of both OR and evolutionary computation techniques. The second approach is an innovative model developed with the aim of automating the DM using a fuzzy rule-based system. This new model is called *fuzzy multiobjective genetic optimiser* and is defined in section 4.5. In order to demonstrate the power of these two new approaches two decision-making optimisation problems both with a high level of difficulty have been solved.

The first problem to be solved is the Bow River Valley case study; this is a multiple objective problem that consists of minimising the pollution in an artificial river basin, and is described in section 4.4. The main purpose of this experiment is to demonstrate that it is possible to combine a method proposed in OR with a GA, visualising the GA as a search algorithm. The GA uses the interactive SEMOPS method to describe the fitness function based on goal levels and aspiration levels. The aspiration levels are directly related to the decision-maker's preferences.

An interesting question to posit is "what happens if the system has to be automated?" One answer to this question is to implement a fuzzy rule-based system. The fuzzy rule-based system is programmed with the classification given by the decision-maker. The optimiser is applied to the Bow River Valley case study. The results of this fuzzy genetic algorithm are compared with those found using a human decision-maker instead of the fuzzy rule-based system (section 4.4).

The second problem to be solved is a very common problem in hospitals: the nurse-scheduling problem (NSP) also known as the nurse rostering system, as described in section 4.6. There are many organisations that divide the work of their staff into different shifts such as hotels, hospitals and service centres. Normally, the staff members work around rosters. Cheng et al. (1996) define roster as the process of determining the staff shifts over a period of

time. Valouxis and Housos (2000) define a roster as a set of work and rest days assigned to a staff member. The NSP is defined as the generation of a schedule of working days and days off for each nurse in a hospital. In order to manage this process, it has been decided to use the SEMOPS-GA approach.

Finally, the conclusions are presented in section 4.7.

4.2 Handling Preferences

Following the classification of information for preferences presented in Chapter One, the stage used in this chapter is an interactive method which follows a two-step algorithm. The first step is to identify a non-dominated solution. In the second step the DM will provide his or her preferences and trade-off information. This information is used to modify the problem. For this reason these methods are known as methods of *Progressive Articulation of Preferences* (Goicoechea et al., 1982).

Fonseca and Fleming (1993) made a first attempt to deal with preferences of the DM in a GA for multiple objective optimisation. This multiobjective genetic optimiser consists of presenting to the DM a set of points to evaluate in each generation of a genetic algorithm. After the assessment is made, the DM communicates his/her preferences to the GA. The GA proceeds with the next generation (Figure 4.1).

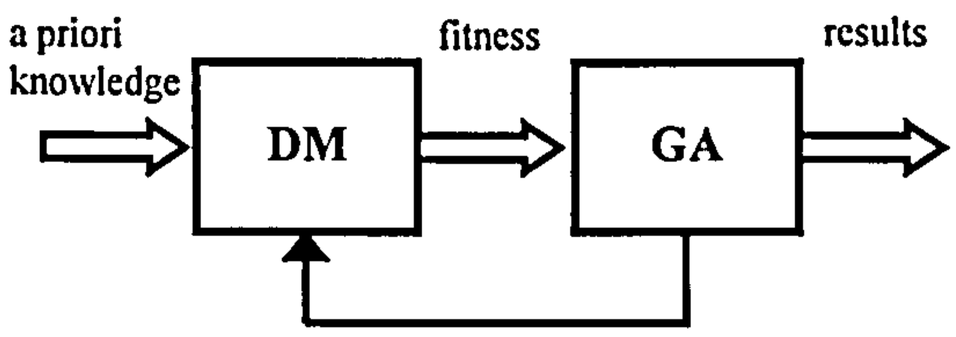


Figure 4.1 General Multiobjective Genetic Optimiser
Adapted from Fonseca and Fleming (1993)

Once the MOP and the genetic optimiser have been defined, it is necessary to present the method (SEMOPS) that will be used throughout this chapter.

4.3 SEMOPS: Method

Sequential Multiobjective Problem Solving (SEMOPS) is an interactive method developed by Monarchi et al. (1973), which involves the DM in the search for a *satisfactum* (satisfactory alternative of action). This algorithm has two different levels: goal and aspiration. The goal levels are defined as "conditions imposed on the DM by external forces" and the aspiration levels are "attainment levels of the objectives which the DM personally desires to achieve" (Goicoechea et al., 1982).

Simon (1959) concluded that aspiration levels specify the conditions for satisfaction. He considered satisfying models have a better performance than optimising models. The satisfying model offers the DM the possibility of searching for new alternatives of action. Finally, Simon (1959) stated that most entrepreneurs want to achieve a satisfactory alternative of action rather than the optimal.

The decision problem has p goals, n decision variables, and a constraint set X. Each goal is connected to an objective function, so for p goals there will be p objective functions. These functions are represented in a set $\mathbf{z} = (z_1, z_2, ..., z_p)$ and will be used to evaluate how well the

goals have been accomplished. It is necessary that the set of constraints X be continuous, and the constraint and objective functions have to be at least first order differentiable (Monarchi et al., 1973).

The goals are transformed into a function $d(\mathbf{x})$ in the real positive set. If $d < 1$ the goal is satisfied. Each objective function will be compared to its correspondent aspiration level AL and will be transformed into a d function as follows:

At most:

$$z_i(\mathbf{x}) \leq AL_i; \quad d_i = \frac{z_i(\mathbf{x})}{AL_i} \tag{4.1}$$

At least:

$$z_i(\mathbf{x}) \geq AL_i; \quad d_i = \frac{AL_i}{z_i(\mathbf{x})} \tag{4.2}$$

Equal:

$$z_i(\mathbf{x}) = AL_i; \quad d_i = \frac{1}{2}\left[\frac{AL_i}{z_i(\mathbf{x})} + \frac{z_i(\mathbf{x})}{AL_i}\right] \tag{4.3}$$

Within an interval:

$$AL_{iL} \leq z_i(\mathbf{x}) \leq AL_{iU}$$
$$d_i = \left[\frac{AL_{iU}}{AL_{iL} + AL_{iU}}\right]\left[\frac{AL_{iL}}{z_i(\mathbf{x})} + \frac{z_i(\mathbf{x})}{AL_{iU}}\right] \tag{4.4}$$

The main purpose of SEMOPS is to generate information in order to help the DM to make a decision. It is important to notice that the algorithm by itself does not solve the problem.

Another important characteristic of this method is the use of a surrogate objective function. This surrogate objective function s is based on the goal and aspiration levels. Thus:

$$s = \sum_{i=1}^{P} d_i \tag{4.5}$$

SEMOPS is defined as a three-step algorithm, (1) *set-up* transforms the original problem into the surrogate objective function, (2) *iteration* is cycling until a satisfactum is obtained, and (3) *termination* is when a satisfactum is found or when the DM decides that there is no satisfactum.
Then, for the first iteration the problem will be described as follows:

Principal problem

$$\min \quad s_1 = \sum_{i=1}^{P} d_i \tag{4.6}$$

subject to

$$\mathbf{x} \in \mathbf{X} \tag{4.7}$$

auxiliary problems, $l = 1,2,\ldots, P$

$$\min \quad s_{1l} = \sum_{\substack{i=1 \\ i \neq l}}^{P} d_i \tag{4.8}$$

where P is the number of auxiliary problems
subject to

$$\begin{aligned} &\mathbf{x} \in \mathbf{X} \\ &z_l(\mathbf{x}) \geq AL_l \end{aligned} \tag{4.9}$$

The solution of this iteration will be used in the assessment process for the DM to decide the changes for the next iteration. In this way the process is interactive to take into account the DM's preferences. Figure 4.2 shows an adaptation of the genetic optimiser proposed by Fonseca and Fleming (1993), considering the DM preferences and the SEMOPS method.

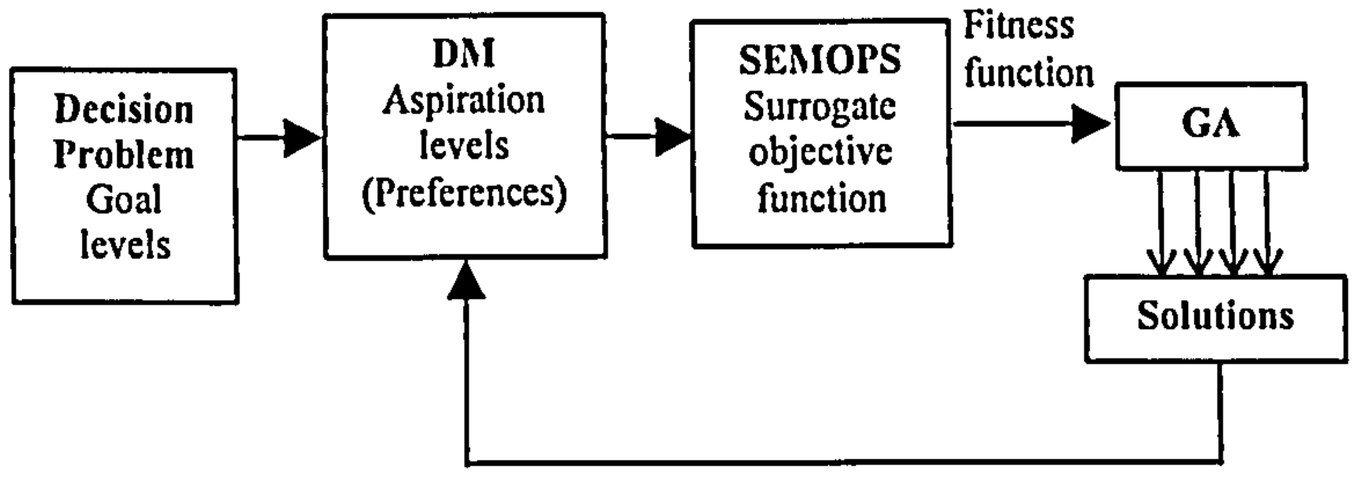


Figure 4.2 Genetic optimiser using the SEMOPS method

4.4 Bow River Valley Case Study

The Bow River Valley is a hypothetical example of an artificial river basin and its pollution problems. As Figure 4.3 shows, there are three pollution sources, the Pierce-Hall Cannery, and the towns of Bowville and Plympton. The water quality will be measured by the concentration of dissolved oxygen (DO). The number of pounds of biochemical oxygen demanding material (BOD) will measure the waste. The BOD is divided into nitrogenous (BOD_n) and carbonaceous (BOD_c) material.

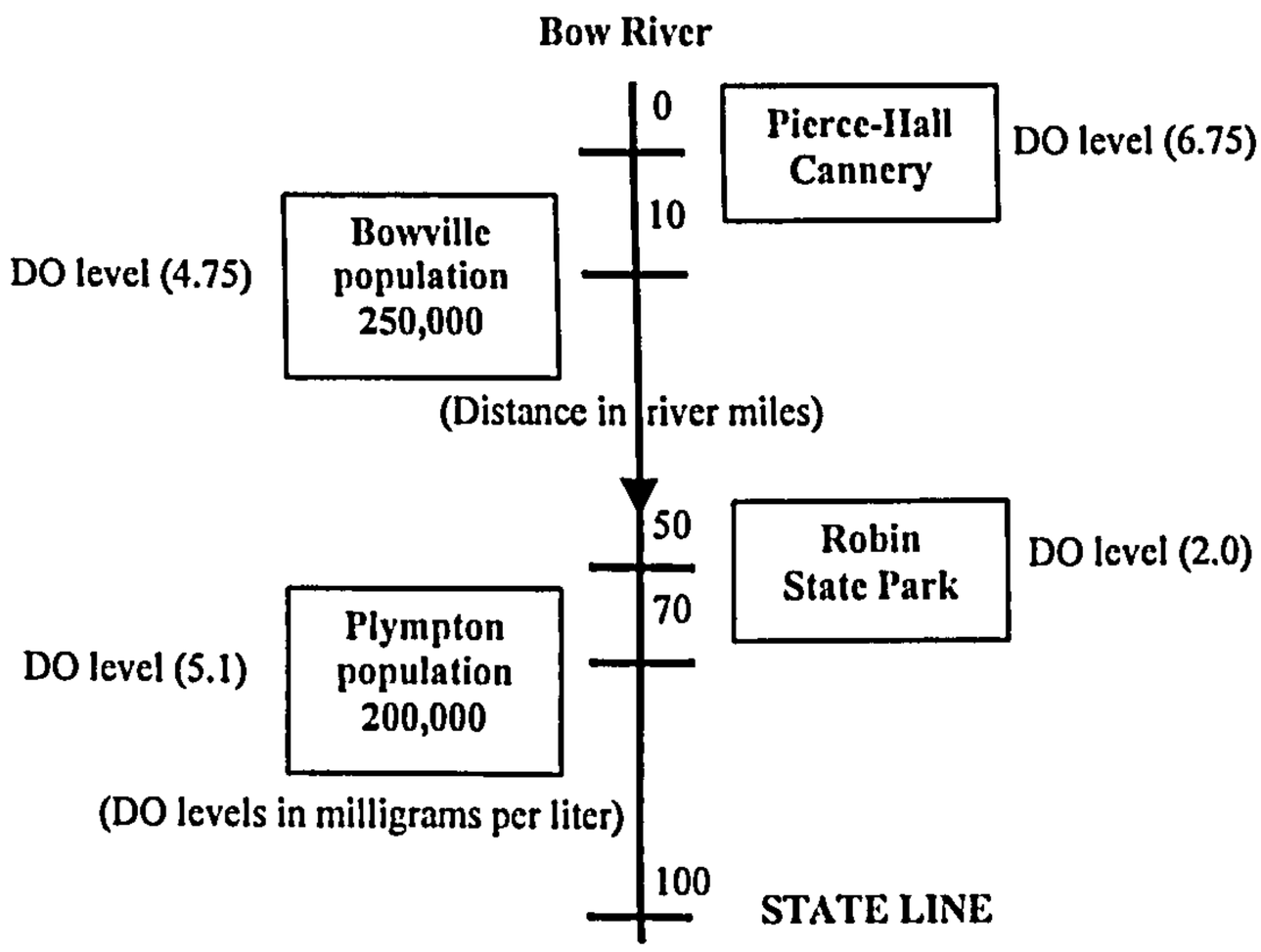


Figure 4.3 Bow River Valley Basin
Adapted from Monarchi et al. (1973)

The BOD is reduced by 30% of the total with primary treatment facilities installed in the Pierce-Hall cannery, and in the towns of Bowville and Plympton.

The goals for this case study are:

1. To raise the dissolved oxygen concentration (DO) to at least 6 mg/l at Bowville.

2. To raise the dissolved oxygen concentration (DO) to at least 6 mg/l at Plympton.
3. To raise the dissolved oxygen concentration (DO) to at least 6 mg/l at Robin State Park.
4. To maintain a percentage of return on investment above 6.5% at Pierce-Hall.
5. To hold the tax rate below $1.5 per $1000 of assessed value at Bowville.
6. To hold the tax rate below $1.5 per $1000 of assessed value at Plympton.

The proportionate reduction in BOD_n (w_i) is in terms of the BOD_c (x_i), as follows:

$$w_i = \frac{0.39}{(1.39 - x_i^2)} \tag{4.10}$$

The DO level q_j is specified by:

$$q_j = -\sum \left[d_{ij}^c L_i^c (x_i - 0.3) + d_{ij}^n L_i^n (w_i - 0.3) \right] + \bar{q}_j \tag{4.11}$$

where

$*d_{ij}^c$ = carbonaceous transfer coefficient between points i and j;

$*d_{ij}^n$ = nitrogenous transfer coefficient between points i and j;

$*L_i^c$ = carbonaceous *BOD* load for source i;

$*L_i^n$ = nitrogenous *BOD* load for source i;

x_i = proportionate reduction in L_i^c;

w_i = proportionate reduction in L_i^n;

$\bar{q}_j$ = current *DO* level at point j;

i,j = points

* The values of the transfer coefficients and the BOD waste levels are in Appendix A

Furthermore, it is possible to define the decision variables for this problem:

x_1 = proportionate reduction in carbonaceous *BOD* load at the Pierce-Hall Cannery.
x_2 = proportionate reduction in carbonaceous *BOD* load at Bowville.
x_3 = proportionate reduction in carbonaceous *BOD* load at Plympton.

The only constraint on this problem is that the DO level at the state line has to be at least 3.5 mg/l and is calculated using equation 4.11. The constraint is called q_4 and is expressed as follows:

$$q_4 \geq 3.5 \text{ mg/l} \tag{4.12}$$

The next set of equations describes the incremental cost for reduction above 30% in BOD_c (additional waste treatment) where the units of C_i are in thousand dollars/year:

Pierce-Hall Cannery

$$C_1 = \left[59 / (1.09 - x_1^2) \right] - 59 \tag{4.13}$$

Bowville

$$C_2 = \left[532 / (1.09 - x_2^2) \right] - 532 \tag{4.14}$$

Plympton

$$C_3 = \left[450 / (1.09 - x_3^2) \right] - 450 \tag{4.15}$$

The percentage return on investment r will be determined in terms of the additional annual cost in the Pierce-Hall Cannery:

$$r = \frac{100}{5{,}000{,}000}(375{,}000 - 0.6\,C_1) \tag{4.16}$$

The town's tax rate will be affected by the additional waste treatment:

Bowville

$$\Delta t_2 = (2.4 \times 10^{-3})(0.75\,C_2) \tag{4.17}$$

Plympton

$$\Delta t_3 = (3.3 \times 10^{-3})(0.75\,C_3) \tag{4.18}$$

For the first iteration the goal levels and the aspiration levels are considered to be the same.

Thus, the goals are calculated as follows:

Goal 1 DO level at Bowville ($z_1 \geq AL_1$)

$$z_1 = q_1 \tag{4.19}$$

Goal 2 DO level at Robin State Park ($z_2 \geq AL_2$)

$$z_2 = q_2 \tag{4.20}$$

Goal 3 DO level at Plympton ($z_3 \geq AL_3$)

$$z_3 = q_3 \tag{4.21}$$

where q_1, q_2 and q_3 are calculated using equation 4.11.

Goal 4 Percentage of return on investment at Pierce-Hall ($z_4 \geq AL_4$)

$$z_4 = r \tag{4.22}$$

where r is calculated using equation 4.16.

Goal 5 Tax rate at Bowville ($z_5 \leq AL_5$)

$$z_5 = \Delta t_2 \tag{4.23}$$

where Δt_2 is calculated using equation 4.17.

Goal 6 Tax rate at Plympton ($z_6 \leq AL_6$)

$$z_6 = \Delta t_3 \tag{4.24}$$

where Δt_3 is calculated using equation 4.18.

4.4.1 The Genetic Algorithm

For the Bow River Valley problem it is necessary to use real-valued coding and decoding for the GA. This real-valued representation will make the coding of genotypes easier. However, in this type of GA it is necessary to define different selection, crossover and mutation operators.

Selection operator

Tournament selection is used specially for maximisation and minimisation problems. It allows the algorithm to select the individual with the lowest fitness value within a group of y individuals randomly selected. The tournament size y used is 3 and represents the number of individuals chosen from the initial population to participate in the tournament (Chapter Two).

The selection procedure is a very important part of the GA, not only because it allows the selection of the best solution after each iteration, but also because it reduces the size of the solution set. If the size of the solution set is reduced, the DM will have a better view of the possible solutions to make a decision.

Crossover operators

The recombination operators selected are (as defined in Chapter Three):

Arithmetic crossover (Michalewicz, 1994). Having two parents p_1 and p_2, and a number α within [0,1], the offspring ch is:

$$ch_i = \alpha\, p_{1i} + (1-\alpha)\, p_{2i} \tag{4.25}$$

where $i = 1,\ldots, n$ and n is the number of decision variables.

Unfair average crossover (Nomura and Miyoshi, 1995). Having two parents p_1 and p_2, and a number α within [0,0.5], the offspring ch_1 and ch_2 are:

$$ch_{1i} = \begin{cases} (1+\alpha)\, p_{1i} - \alpha\, p_{2i}, & \text{for } i = 1,\ldots, j, \\ -\alpha\, p_{1i} + (1+\alpha)\, p_{2i}, & \text{for } i = j+1,\ldots, n \end{cases} \tag{4.26}$$

$$ch_{2i} = \begin{cases} (1-\alpha)\, p_{1i} + \alpha\, p_{2i}, & \text{for } i = 1,\ldots, j, \\ \alpha\, p_{1i} + (1-\alpha)\, p_{2i}, & \text{for } i = j+1,\ldots, n \end{cases} \tag{4.27}$$

where j is a number randomly generated between 1 and n (number of decision variables).

Simulated binary crossover (Deb, 2001). Having two parents p_1 and p_2, the offspring ch_1 and ch_2 are:

$$ch_{1i} = 0.5[(1+\beta_{qi})\,p_{1i} + (1-\beta_{qi})\,p_{2i}] \tag{4.28}$$

$$ch_{2i} = 0.5[(1-\beta_{qi})\,p_{1i} + (1+\beta_{qi})\,p_{2i}] \tag{4.29}$$

where β_{qi} is calculated as shown in Chapter Three and n is the number of decision variables.

Mutation operator

The mutation operator selected is the one proposed by Michalewicz (1994) that is applied to a floating point number and the result is a random value from the domain <LB,UB>, where LB, UB are the lower and upper bound respectively. The probability of mutation is equal to 15%. This operator was defined in Chapter Three.

Goal and aspiration levels

The goals are:

$$\mathbf{z} = (z_1, z_2, z_3, z_4, z_5, z_6)$$

$$\begin{aligned} z_1 &\geq 6.0, \quad z_2 \geq 6.0, \\ z_3 &\geq 6.0, \quad z_4 \geq 6.5, \\ z_5 &\leq 1.5, \quad z_6 \leq 1.5, \end{aligned}$$

For the first iteration the DM's aspiration levels will have the goal's values:

$$\mathbf{AL} = (AL_1, AL_2, AL_3, AL_4, AL_5, AL_6)$$

$$\begin{aligned} AL_1 &= 6.0, \quad AL_2 = 6.0, \\ AL_3 &= 6.0, \quad AL_4 = 6.5, \\ AL_5 &= 1.5, \quad AL_6 = 1.5, \end{aligned}$$

For the following generations, the goals remain the same and the DM changes the aspiration levels depending on the solutions obtained in the previous iteration.

4.4.2 Experimental Results

The GA is implemented in C, the initial population is expressed in floating point vectors generating randomly x_i vectors used to generate w_i vectors, d_i vectors and the s solution.

In this work, the algorithm was run for 50 cycles for each principal and auxiliary problem, and $\alpha = 0.5$ (crossover operator) finding the solutions showed in Table 4.1. These conditions for comparison purposes will be considered as the first iteration in the traditional method (Monarchi et al., 1973).

The first row in Table 4.1 (grey) represents the solution of equation 4.6 (surrogate function) and the following rows represent the solution for the auxiliary problems. The aspiration levels are:

A= (6.0, 6.0, 6.0, 6.5, 1.5, 1.5)

	Decision variables			Goals						Fitness
	x_1	x_2	x_3	z_1	z_2	z_3	z_4	z_5	z_6	s
Aspiration Levels				**≥6.0**	**≥6.0**	**≥6.0**	**≥6.5**	**≤1.5**	**≤1.5**	
Surrogate	0.806	0.653	0.849	5.899	3.729	5.664	6.602	0.486	1.914	6.269
Auxiliary 1	0.786	0.668	0.843	5.854	3.764	5.677	6.709	0.530	1.844	5.203
Auxiliary 2	0.638	0.389	0.847	5.518	2.552	5.279	7.170	0.063	1.892	4.433
Auxiliary 3	0.819	0.644	0.849	5.929	3.711	5.657	6.519	0.461	1.934	5.222
Auxiliary 4	0.983	0.601	0.857	6.300	3.955	5.709	2.530	0.358	2.035	5.115
Auxiliary 5	0.743	0.942	0.853	5.756	5.431	6.207	6.891	3.764	1.971	5.371
Auxiliary 6	0.816	0.669	0.917	5.922	3.813	5.691	6.538	0.533	3.408	4.991

Table 4.1 Results using the goals values as aspiration levels

The solution of the surrogate function yields to the following decision variable values:

$x_1 = 0.806$, 80.6 % Pierce-Hall Cannery
$x_2 = 0.653$, 65.3 % Bowville
$x_3 = 0.849$, 84.9 % Plympton
where only goals 4 and 5 are met.

It was decided to follow the same procedure as Monarchi et al. (1973) to make their results comparable with those obtained with the GA. For this reason goal 6 is managed as a constraint with an aspiration level $z_6 \leq 1.8$ (tax rate at Plympton), obtaining the results presented in Table 4.2. The aspiration levels are:

A= (6.0, 6.0, 6.0, 6.5, 1.5, 1.8)

	Decision variables			Goals					*Const	Fitness
	x_1	x_2	x_3	z_1	z_2	z_3	z_4	z_5	z_6	s
Aspiration Levels				**≥6.0**	**≥6.0**	**≥6.0**	**≥6.5**	**≤1.5**	**≤1.8**	
Surrogate	0.832	0.662	0.836	5.959	3.810	5.689	6.427	0.513	1.754	4.989
Auxiliary 1	0.779	0.676	0.838	5.839	3.788	5.685	6.740	0.554	1.776	3.973
Auxiliary 2	0.799	0.321	0.837	5.884	2.521	5.258	6.639	0.012	1.768	3.148
Auxiliary 3	0.819	0.647	0.836	5.931	3.725	5.662	6.514	0.469	1.751	3.932
Auxiliary 4	0.943	0.655	0.834	6.216	4.031	5.746	4.671	0.491	1.725	3.826
Auxiliary 5	0.762	0.989	0.832	5.800	6.038	6.390	6.817	7.560	1.707	3.920

Table 4.2 Results considering goal 6 as a constraint

* goal 6 acts as a constraint

It is possible to see that once goal 6 is considered as a constraint the number of auxiliary problems is reduced by one. In other words, the initial number of auxiliary problems is six and in the second generation five. For each generation, the number of auxiliary problems will be reduced by one.

The solution of the surrogate function yields to the following decision variables values:

$x_1 = 0.832$, 83.2 % Pierce-Hall Cannery
$x_2 = 0.662$, 66.2 % Bowville
$x_3 = 0.836$, 83.6 % Plympton

where aspiration levels 5 and 6 are met.

Analysing the results, it is obvious that to attain the aspiration level of goal 2 ($z_2 \geq 6.0$) the attainment level of goal 5 (tax rate at Bowville) is modified tremendously, for the last

auxiliary problem (more than 500%). Monarchi et al. (1973) suggested modifying the aspiration level of goal 2 to $z_2 \geq 5.0$ mg/l. For flexibility reasons the GA was defined with a goal 2 aspiration level of $z_2 \geq 4.9$ mg/l; the results found are shown in Table 4.3. The aspiration levels are:

A= (6.0, 4.9, 6.0, 6.5, 1.5, 1.8)

	Decision variables			Goal	*Const	Goals			*Const	Fitness
	x_1	x_2	x_3	z_1	z_2	z_3	z_4	z_5	z_6	s
Aspiration Levels				**≥6.0**	**≥4.9**	**≥6.0**	**≥6.5**	**≤1.5**	**≤1.8**	
Surrogate	0.890	0.849	0.832	6.090	4.902	6.034	5.829	1.635	1.703	4.185
Auxiliary 1	0.880	0.852	0.833	6.068	4.901	6.035	5.961	1.673	1.710	3.199
Auxiliary 2	0.901	0.845	0.835	6.115	4.902	6.033	5.665	1.592	1.741	3.189
Auxiliary 3	0.994	0.799	0.838	6.328	4.927	6.023	1.213	1.159	1.768	2.717
Auxiliary 4	0.709	0.898	0.833	5.679	4.987	6.071	7.002	2.428	1.708	2.973

Table 4.3 Results considering goal 2 as a constraint

* goal 2 and 6 act as constraints

The solution of the surrogate function yields to the following decision variables values:

$x_1 = 0.890$, 89 % Pierce-Hall Cannery
$x_2 = 0.849$, 84.9 % Bowville
$x_3 = 0.832$, 83.2 % Plympton

where aspiration levels 1, 2, 3 and 6 are met.

Modifying the return on investment in the Pierce-Hall Cannery from 6.5 % to 6.0 % ($z_4 \geq 6.0$) will safeguard the future of the Cannery. In the solutions found by Monarchi et al. (1973) (who used a cutting-plane programming technique) there appeared an apparent inconsistent constraint set while using the GA the results are shown in Table 4.4. The aspiration levels are:

A= (6.0, 4.9, 6.0, 6.0, 1.5, 1.8)

	Decision variables			Goal	*Const	Goal	*Const	Goal	*Const	fitness
	x_1	x_2	x_3	z_1	z_2	z_3	z_4	z_5	z_6	s
Aspiration Levels				**≥6.0**	**≥4.9**	**≥6.0**	**≥6.0**	**≤1.5**	**≤1.8**	
Surrogate	0.873	0.861	0.839	6.052	4.952	6.051	6.045	1.794	1.796	3.179
Auxiliary 1	0.850	0.861	0.837	6.000	4.907	6.039	6.279	1.794	1.761	2.189
Auxiliary 2	0.869	0.858	0.835	6.045	4.924	6.043	6.083	1.754	1.742	2.162
Auxiliary 3	0.857	0.869	0.834	6.017	4.978	6.061	6.210	1.901	1.726	1.987

Table 4.4 Results considering goal 4 as a constraint

* goal 2, 4 and 6 act as constraints

The solution of the surrogate function yields to the following decision variables values:

$x_1 = 0.873$, 87.3 % Pierce-Hall Cannery
$x_2 = 0.861$, 86.1 % Bowville
$x_3 = 0.839$, 83.9 % Plympton

where aspiration levels 1, 2, 3, 4 and 6 are met.

Finally, the aspiration level of goal 5 is modified; in the GA the aspiration level will be $z_5 \leq 1.8$ while Monarchi et al. (1973) considered $z_5 \leq 1.9$. After the fifth iteration the results for the cutting-plane programming technique are:

$x_1 = 0.877$, 88 % Pierce-Hall Cannery
$x_2 = 0.869$, 87 % Bowville
$x_3 = 0.818$, 82 % Plympton

Considering $\mathbf{z} = (z_1, z_2, z_3, z_4, z_5, z_6)$ then the results are:

$z = (6.07, 5.03, 6.07, 5.96, 1.92, 1.57)$

The results obtained by the GA are:

$x_1 = 0.873$, 87 % Pierce-Hall Cannery
$x_2 = 0.861$, 86 % Bowville
$x_3 = 0.839$, 84 % Plympton

$\mathbf{z} = (6.05, 4.95, 6.05, 6.05, 1.79, 1.79)$

It is important to mention that in all the cases the constraint q_4 is met.

If the results of Monarchi et al. (1973) are compared to the results of the GA, it is possible to conclude that even though the results are not the same they are very close. It is important to note that to be able to compare the results the criteria followed by the DM in each method were the same. Table 4.5 shows the comparison between goal values, the cutting-plane programming technique and the GA final results.

Goal	Initial Aspiration level (IAL) (goal's value)	Cutting-plane Technique (CPT)	Absolute difference |IAL-CPT| (%)	Genetic Algorithm (GA)	Absolute difference |IAL-GA| (%)	|CPT-GA|
z_1	≥ 6.0	6.07	0.07 (+1.17%)	6.05	0.05 (+0.83%)	0.02
z_2	≥ 6.0	5.03	0.97 (-16.17%)	4.95	1.05 (-17.5%)	0.08
z_3	≥ 6.0	6.07	0.07 (+1.17%)	6.05	0.05 (+0.83%)	0.02
z_4	≥ 6.5	5.96	0.54 (-8.3%)	6.05	0.45 (-6.9%)	0.09
z_5	≤ 1.5	1.92	0.42 (-28%)	1.79	0.29 (-19.33%)	0.13
z_6	≤ 1.5	1.57	0.07 (-4.6%)	1.79	0.29 (-19.33%)	0.22

Table 4.5 Comparison of the cutting-plane programming technique and GA final results.

Considering the results from Table 4.5, goals achievements for both techniques are analysed as follows:

Difference between the initial aspiration levels (goal values) and final results.

Cutting-plane programming technique	*Genetic Algorithm*
- $z_1 \geq 6.0$	
+1.17 %	+0.83 %

where the "+" means that the final result is within the goal range.

- $z_2 \geq 6.0$	
-16.17 %	-17.5 %

where the "-" means that the final result is outside the goal range, for instance a value equal or greater than 6 was expected and the values found are 5.03 and 4.95.

- $z_3 \geq 6.0$	
+1.17 %	+0.83 %

where the "+" means that the final result is within the goal range.

- $z_4 \geq 6.5$	
-8.3 %	-6.9 %

where the "-" means that the final result is outside the goal range, for instance a value equal or greater than 6.5 was expected and the values found are 5.96 and 6.05.

- $z_5 \leq 1.5$	
-28 %	-19.33%

where the "-" means that the final result is outside the goal range, for instance a value equal or less than 1.5 was expected and the values found are 1.92 and 1.79.

- $z_6 \leq 1.5$	
-4.6 %	-19.33%

where the "-" means that the final result is outside the goal's range, for instance a value equal or less than 1.5 was expected and the values found are 1.57 and 1.79.

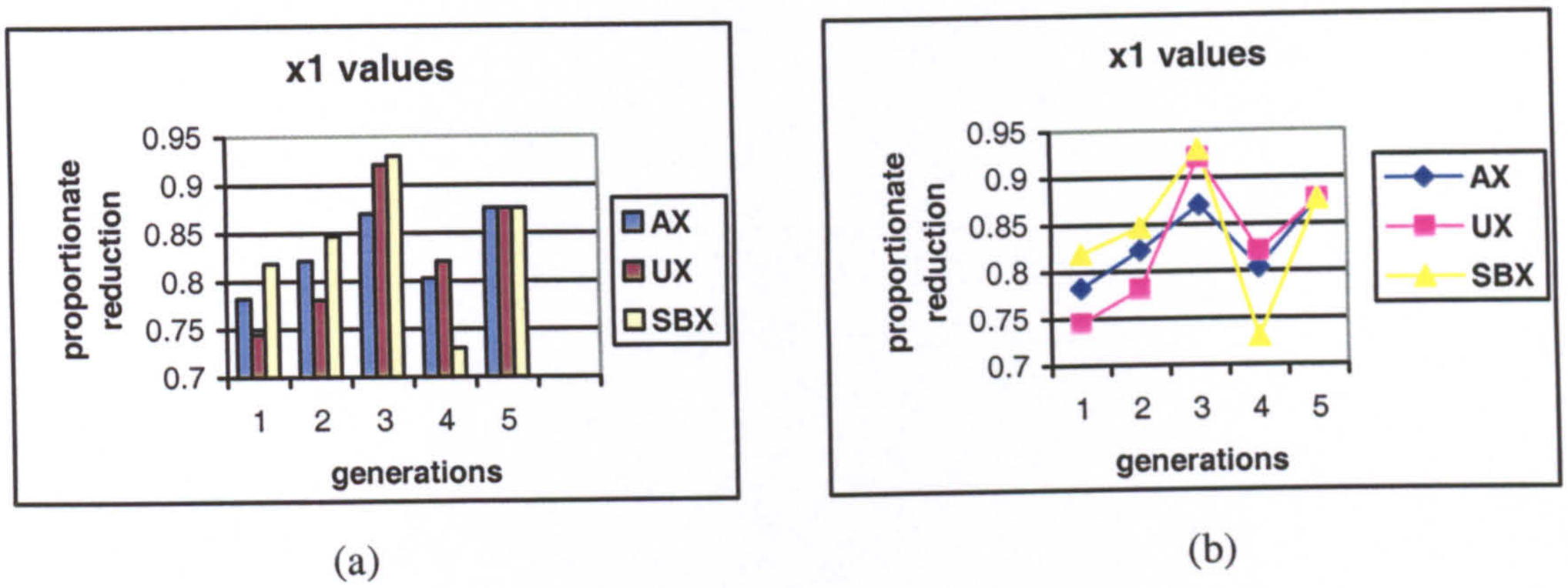
Once it has been shown that the results found by the GA are very similar to those found by Monarchi et al. (1973), it is possible to conclude that the GA is working adequately and, therefore, it is an interesting exercise to run it with different crossover operators.

In this research the programme was run with three different crossovers: arithmetic (AX), unfair (UX) and simulated binary (SBX). For reasons of space, it is not feasible to include all the results found, therefore, they are presented in Appendix A. In this chapter, only the final results are included along with charts for comparison purposes.

Table 4.6 presents the final values of the decision variable x_1 (proportionate reduction in carbonaceous BOD load at the Pierce Hall Cannery) for the three different crossovers (AX, UX and SBX) and the slope change in each generation. An increment on the decision variable is represented by a positive slope whilst a decrement is represented by a negative one. Additionally, Figure 4.4 is constructed using these values.

Generation	AX (x_1)	Slope Value	Slope Sign	UX (x_1)	Slope value	Slope sign	SBX (x_1)	Slope value	Slope sign
1	0.782	------	------	0.745	------	------	0.819	------	------
2	0.822	0.04	+	0.781	0.036	+	0.847	0.028	+
3	0.87	0.048	+	0.921	0.14	+	0.93	0.083	+
4	0.803	-0.067	-	0.821	-0.1	-	0.73	-0.2	-
5	0.876	0.073	+	0.876	0.055	+	0.876	0.146	+

Table 4.6 Final values of x_1 and slope values for the AX, UX and SBX



(a) (b)

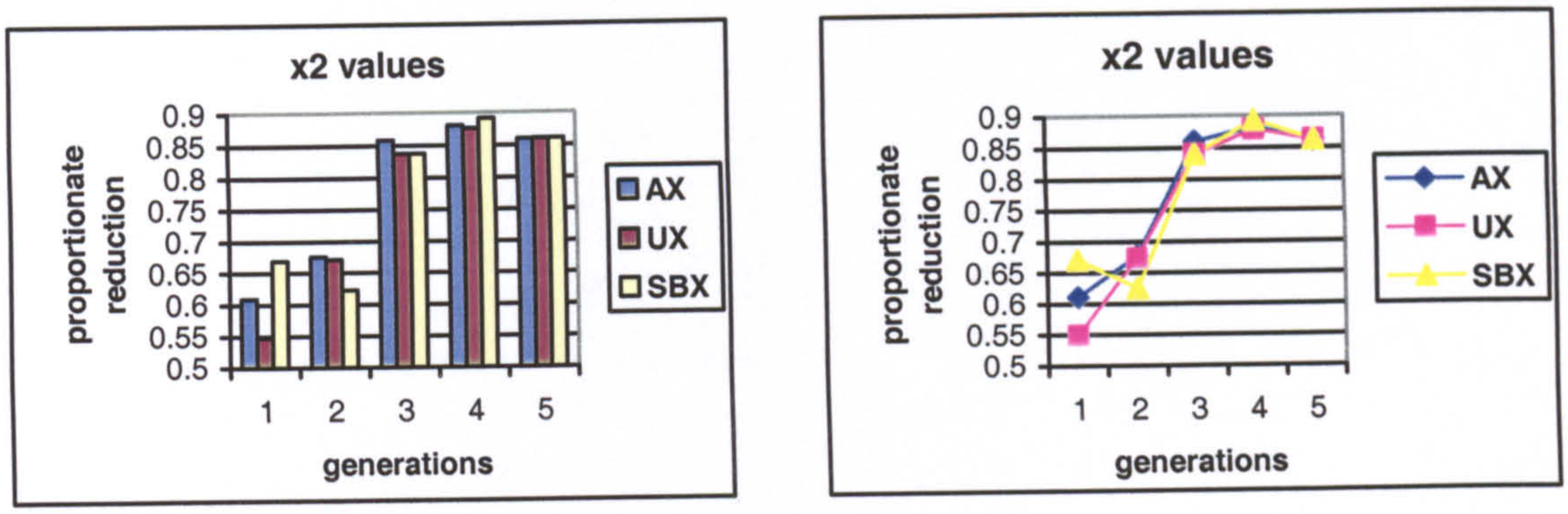
Figure 4.4 Proportionate reduction in carbonaceous BOD at the Pierce-Hall Cannery

From Figure 4.4(a), it can be seen that the proportionate reduction in carbonaceous BOD load at the Pierce-Hall Cannery after the fifth generation is 87.6% for the three crossovers (Table 4.6). It is also possible to see that the behaviour of the contour line on each crossover is the same (Figure 4.4(b)), where in the third generation x_1's value increases considerably with respect to the second (5.84% for AX, 17.93% for UX and 9.8% for SBX) and in the fourth generation the value decreases considerably with respect to the third (7.7% for AX, 10.86% for UX, 21.51% for SBX). Finally, the contour line stabilises in the fifth generation finding an x_1 value between the third and fourth generations' values.

Table 4.7 shows the final values of the decision variable x_2 (proportionate reduction in carbonaceous BOD load in the town of Bowville) for the three different crossovers (AX, UX and SBX) and the slope change in each generation. Additionally, Figure 4.5 is constructed using these values.

Generation	AX (x_2)	Slope Value	Slope sign	UX (x_2)	Slope value	Slope sign	SBX (x_2)	Slope value	Slope sign
1	0.609	------	------	0.548	------	------	0.67	------	------
2	0.676	0.067	+	0.672	0.124	+	0.623	-0.047	-
3	0.859	0.183	+	0.838	0.166	+	0.838	0.215	+
4	0.881	0.022	+	0.877	0.039	+	0.892	0.054	+
5	0.86	-0.021	-	0.861	-0.016	-	0.861	-0.031	-

Table 4.7 Final values of x_2 and slope values for the AX, UX and SBX



(a) (b)

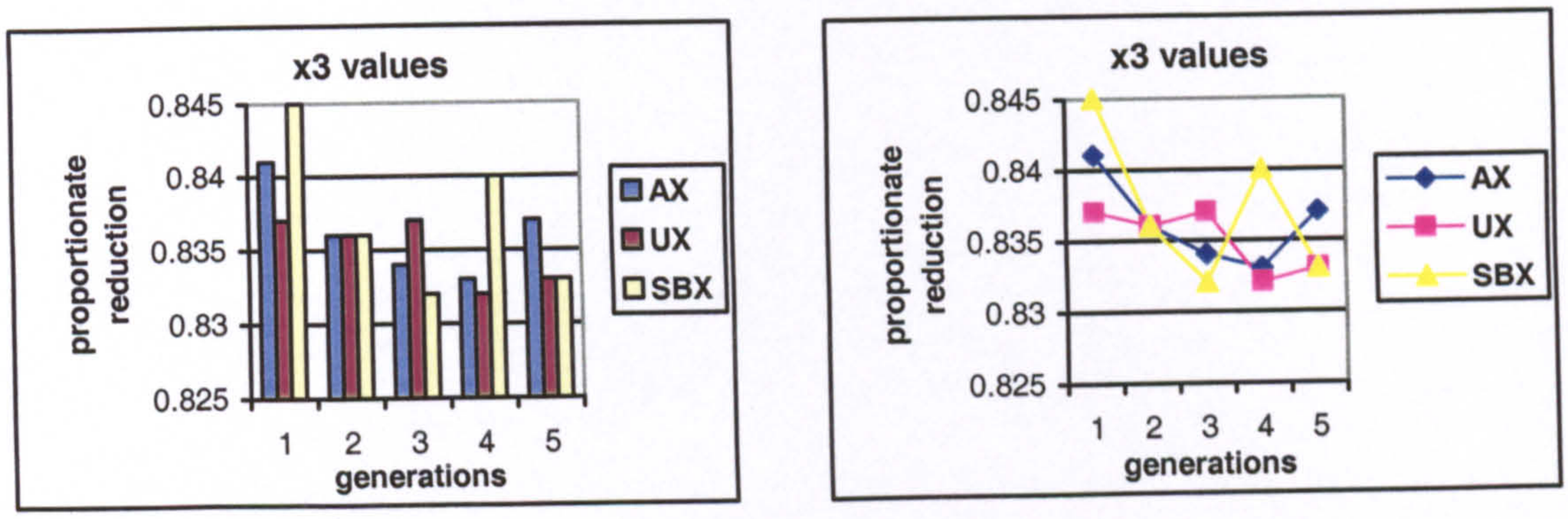
Figure 4.5 Proportionate reduction in carbonaceous BOD in Bowville

From Figure 4.5(a), it is possible to see that the proportionate reduction in carbonaceous BOD load in Bowville after the fifth generation using the UX and SBX is the same (86.1%). The results with the AX differ from those found with the UX and SBX (Table 4.7). AX produced 86% BOD reduction in Bowville. It is obvious that the behaviour of the contour line on each crossover is almost the same (Figure 4.5(b)) with the exception of the second generation of SBX where the x_2 value decreases instead of increasing as in the two other crossovers.

Table 4.8 shows the final values of the decision variable x_3 (proportionate reduction in carbonaceous BOD load at the town of Plympton) for the three different crossovers (AX, UX and SBX) and the slope change in each generation. Additionally, Figure 4.6 is constructed using these values.

Generation	AX (x_3)	Slope Value	Slope sign	UX (x_3)	Slope value	Slope sign	SBX (x_3)	Slope value	Slope sign
1	0.841	------	------	0.837	------	------	0.845	------	------
2	0.836	-0.005	-	0.836	-0.001	-	0.836	-0.009	-
3	0.834	-0.002	-	0.837	0.001	+	0.832	-0.004	-
4	0.833	-0.001	-	0.832	-0.005	-	0.84	0.008	+
5	0.837	0.004	+	0.833	0.001	+	0.833	-0.007	-

Table 4.8 Final values of x_3 and slope values for the AX, UX and SBX



(a) (b)

Figure 4.6 Proportionate reduction in carbonaceous BOD at Plympton

From Figure 4.6(a), it is possible to see that the proportionate reduction in carbonaceous BOD load at Plympton after the fifth generation using the UX and SBX is the same (83.3%). The

results with the AX differ from those found with the UX and SBX (Table 4.8), where the value of 83.3% is met in the fourth generation. AX produced 83.7% BOD reduction at Plympton in the fifth generation. It is obvious that the behaviour of the contour lines in Figure 4.6(b) differ from those of Figure 4.4 (b) and 4.5(b), where the initial values of x_3 for the three crossovers (AX, UX, SBX) are high and the slopes for the next generation are negative. In the SBX case the x_3 values vary in a more abrupt and profound way. These variations can be interpreted as the way the search is responding in order to obtain the optimal or satisfactory x_3 value.

Once each of the three crossover operators has been analysed and compared, the next step to follow is the analysis of the three decision variables values for each crossover (AX, UX and SBX).

Table 4.9 presents the comparison between the x_1, x_2 and x_3 values for the AX. It also shows the slope values for each generation. Figure 4.7 is built using these values.

Generation	AX (x_1)	Slope Value	Slope Sign	AX (x_2)	Slope Value	Slope sign	AX (x_3)	Slope Value	Slope sign
1	0.782	------	------	0.609	------	------	0.841	------	------
2	0.822	0.04	+	0.676	0.067	+	0.836	-0.005	-
3	0.87	0.048	+	0.859	0.183	+	0.834	-0.002	-
4	0.803	-0.067	-	0.881	0.022	+	0.833	-0.001	-
5	0.876	0.073	+	0.86	-0.021	-	0.837	0.004	+

Table 4.9 Final values of x_1, x_2 and x_3 for the AX

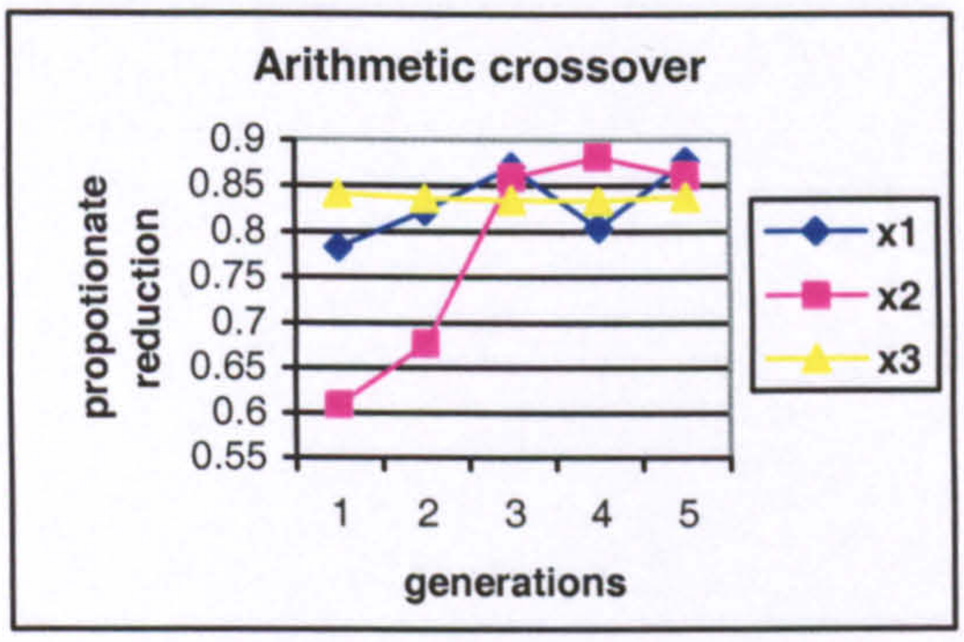


Figure 4.7 Proportionate reduction in carbonaceous BOD in the Pierce-Hall Cannery, Bowville and Plympton

Table 4.9 shows that the decision variable that starts with the lowest value is x_2, for this reason the slopes in each generation are positive until the last generation where it stabilises with a negative one. In the opposite way, the decision variable x_3 starts with a high value, therefore, in the following generations the slope is negative until the last one where it stabilises with a positive slope. However, the percentile difference in each decision variable differs considerably. For instance, the x_1 value increases 12.02% from the first to the last generation, the x_2 value increases 41.22% from the first to the last generation, whilst the x_3 value decreases 0.476% from the first to the last generation. Therefore the conclusion is that the changes in the x_3 values are very smooth whilst in x_2 values are bigger due to their dependence on goal 2 (Figure 4.7). The analysis of the goal values will be shown in Figures 4.10, 4.11 and 4.12.

Table 4.10 presents the comparison between the x_1, x_2 and x_3 values for the UX. It also shows the slope values for each generation. Figure 4.8 is built using these values.

Generation	UX (x_1)	Slope value	Slope Sign	UX (x_2)	Slope Value	Slope sign	UX (x_3)	Slope value	Slope sign
1	0.745	------	------	0.548	------	------	0.837	------	------
2	0.781	0.036	+	0.672	0.124	+	0.836	-0.001	-
3	0.921	0.14	+	0.838	0.166	+	0.837	0.001	+
4	0.821	-0.1	-	0.877	0.039	+	0.832	-0.005	-
5	0.876	0.055	+	0.861	-0.016	-	0.833	0.001	+

Table 4.10 Final values of x_1, x_2 and x_3 for the UX

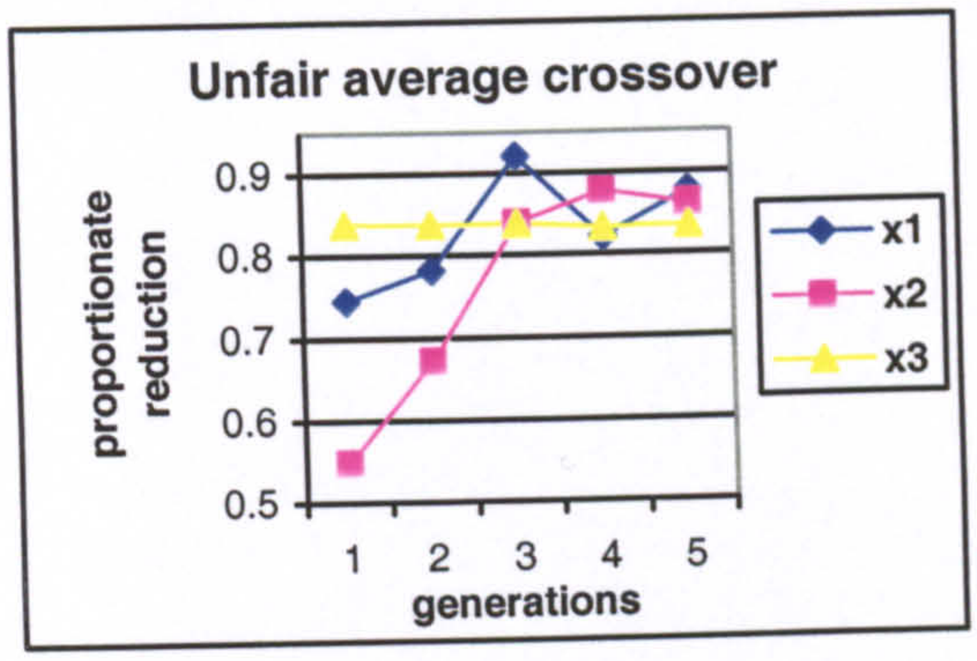


Figure 4.8 Proportionate reduction in carbonaceous BOD in the Pierce-Hall Cannery, Bowville and Plympton

From Table 4.10, it is possible to see that the decision variable that has the lowest value is x_2 and as in the AX case, the slopes in each generation are positive until the last generation where it is negative. Moreover, if the percentile difference in each decision variable (considering the first and the last values) is analysed, the following results are found: x_1 increases 17.58%, x_2 increases 57.12% and x_3 decreases 0.478%. By comparison, the curves for x_1, x_2 and x_3 depicted in Figure 4.7 and Figure 4.8 have the same shape.

Table 4.11 presents the comparison between the x_1, x_2 and x_3 values for the SBX. It also shows the slope values for each generation. Figure 4.9 is built using these values.

Generation	SBX (x_1)	Slope value	Slope sign	SBX (x_2)	Slope Value	Slope sign	SBX (x_3)	Slope value	Slope sign
1	0.819	------	------	0.67	------	------	0.845	------	------
2	0.847	0.028	+	0.623	-0.047	-	0.836	-0.009	-
3	0.93	0.083	+	0.838	0.215	+	0.832	-0.004	-
4	0.73	-0.2	-	0.892	0.054	+	0.84	0.008	+
5	0.876	0.146	+	0.861	-0.031	-	0.833	-0.007	-

Table 4.11 Final values of x_1, x_2 and x_3 for the SBX

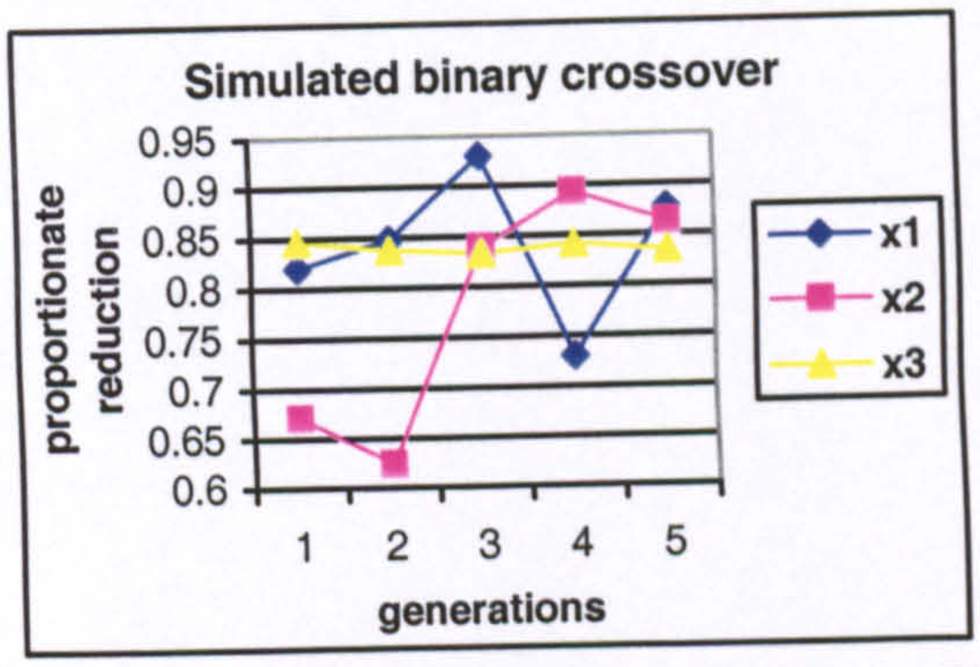


Figure 4.9 Proportionate reduction in carbonaceous BOD in the Pierce-Hall Cannery, Bowville and Plympton

From Table 4.11, it can be seen that the decision variable that starts with the lowest value is x_2. Nevertheless, in this case (SBX) the second generation has a negative slope making the second x_2 value even smaller. In the two next generations the slopes are positive and finally in the last generation the slope is negative again, stabilising the solution. Furthermore, the percentile difference in each decision variable considering the first and the last values yields the following results: x_1 increases 6.96%, x_2 increases 28.51% and x_3 decreases 1.42%. After comparing Figure 4.7, Figure 4.8 and Figure 4.9, the conclusion can be that x_1 and x_3 have very similar shapes and that x_2 varies only in the SBX case.

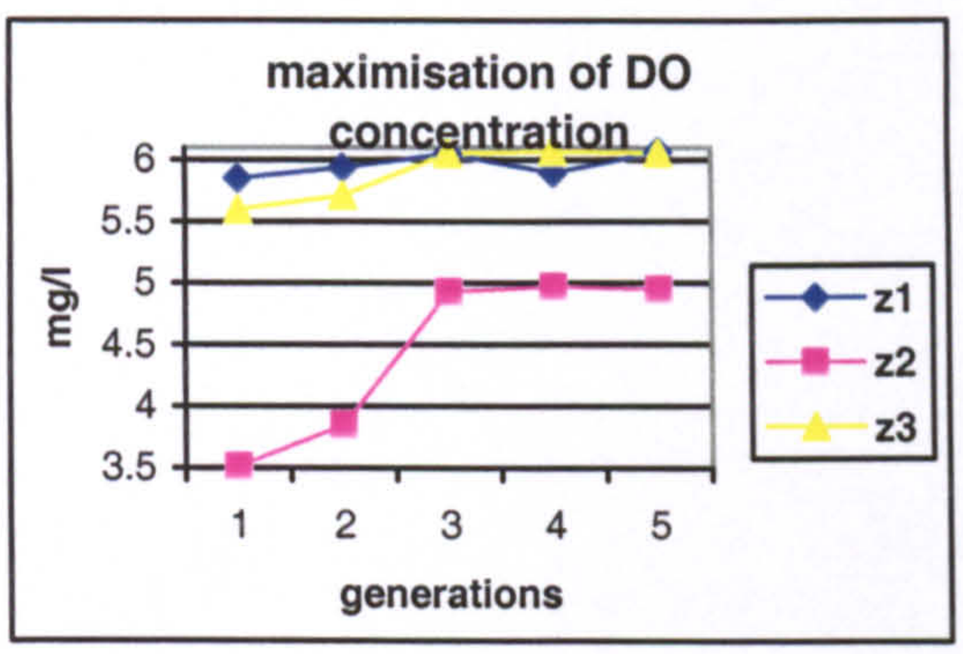
It could be also interesting to analyse the final results of each goal (z_1, z_2, z_3, z_4, z_5 and z_6) for each of three crossovers.

Table 4.12 shows the goal values using the AX and Figure 4.10 is built using those values.

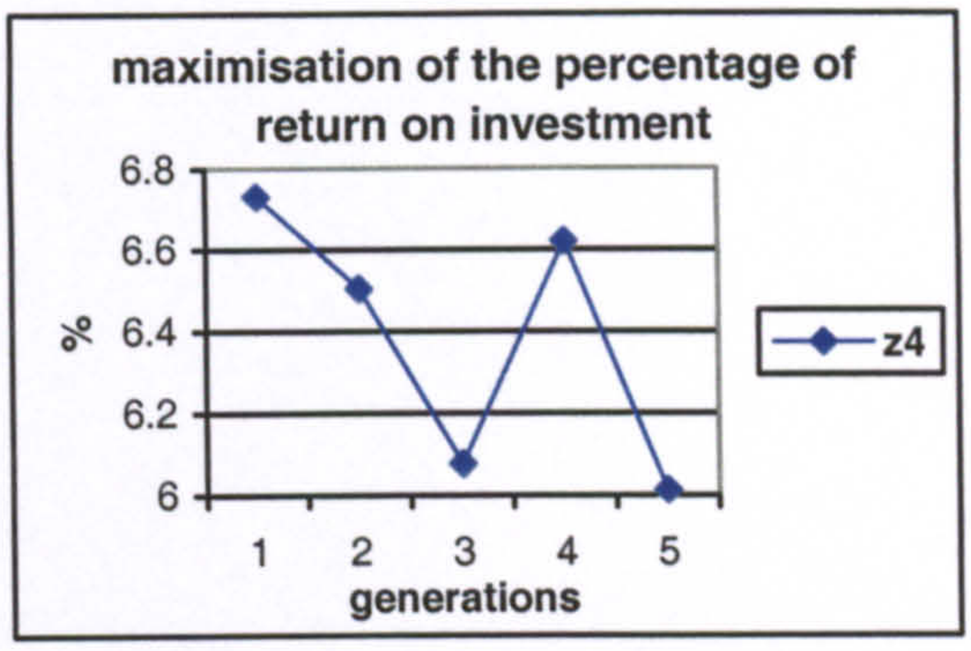
AX

Generation	z_1	z_2	z_3	z_4	z_5	z_6
1	5.845	3.513	5.595	6.729	0.375	1.809
2	5.935	3.852	5.703	6.503	0.555	1.748
3	6.046	4.927	6.044	6.077	1.757	1.727
4	5.892	4.974	6.064	6.620	2.100	1.718
5	6.059	4.948	6.050	6.012	1.775	1.761

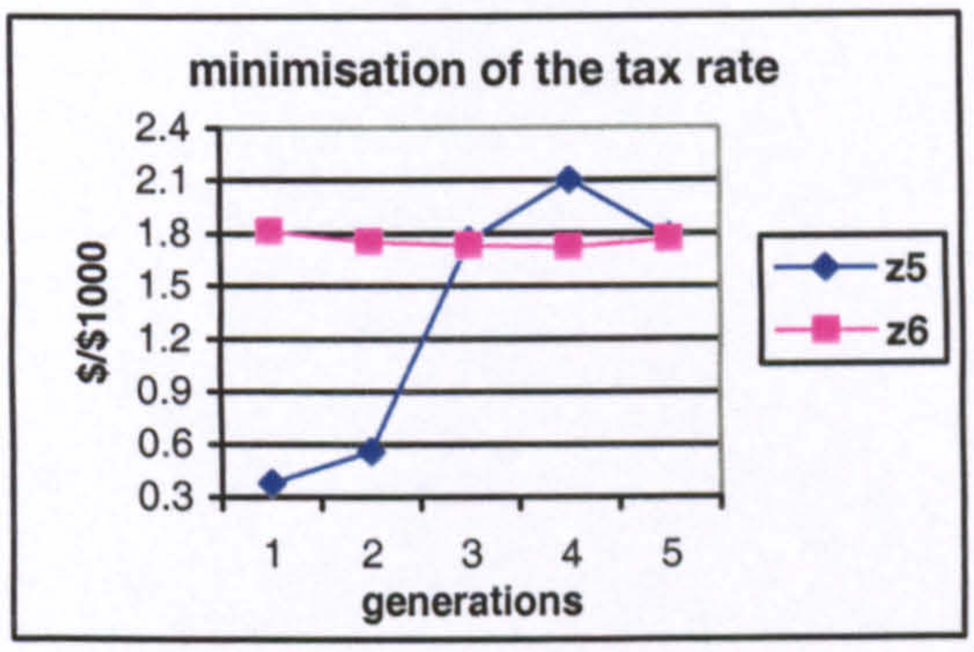
Table 4.12 Final values of z_1, z_2, z_3, z_4, z_5 and z_6 for the AX



(a)



(b)



(c)

Figure 4.10 (a) Maximisation of DO concentration for the AX (b) Maximisation of the percentage of return on investment for the AX and (c) Minimisation of the tax rate for the AX

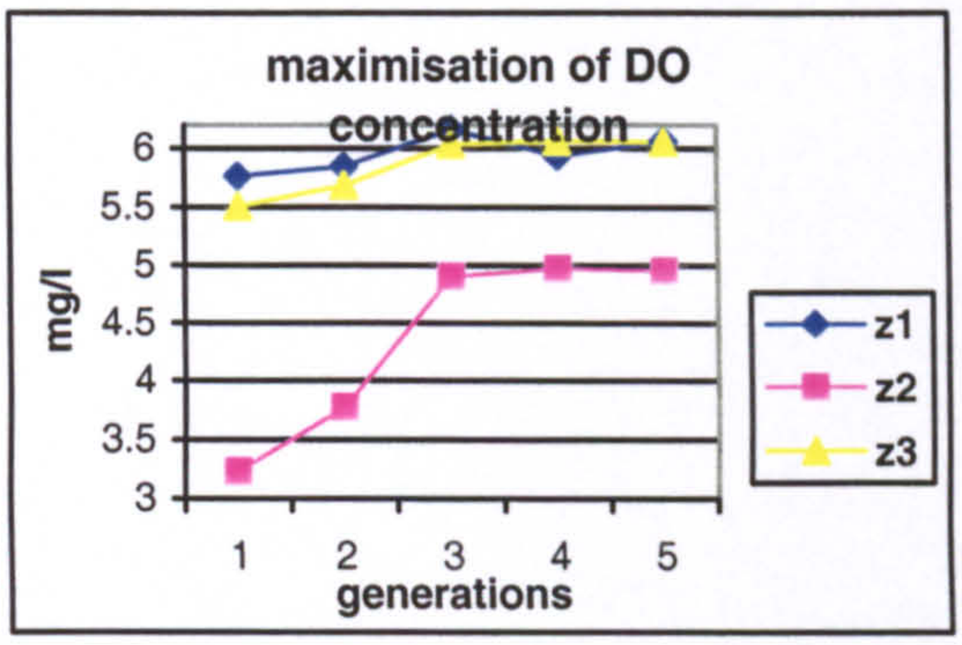
From Figure 4.10(c), it is evident that z_6 maintains almost a constant value throughout the five generations, because it becomes a constraint in the second generation. In comparison with z_2 that becomes a constraint in the third generation (Figure 4.10(a)), it is more evident the change of values and how it stabilises. For z_4 (Figure 4.10(b)) it would seem that the values are not stable at all, but the abrupt change of value from generation 4 to 5 is due to the change of value in the aspiration level from 6.5 to 6. The z_1 values remain almost constant throughout all generations. After this analysis has been done, it could be concluded that the charts reflect the changes in the aspiration levels.

Table 4.13 shows the goal values using the UX and Figure 4.11 is built using those values.

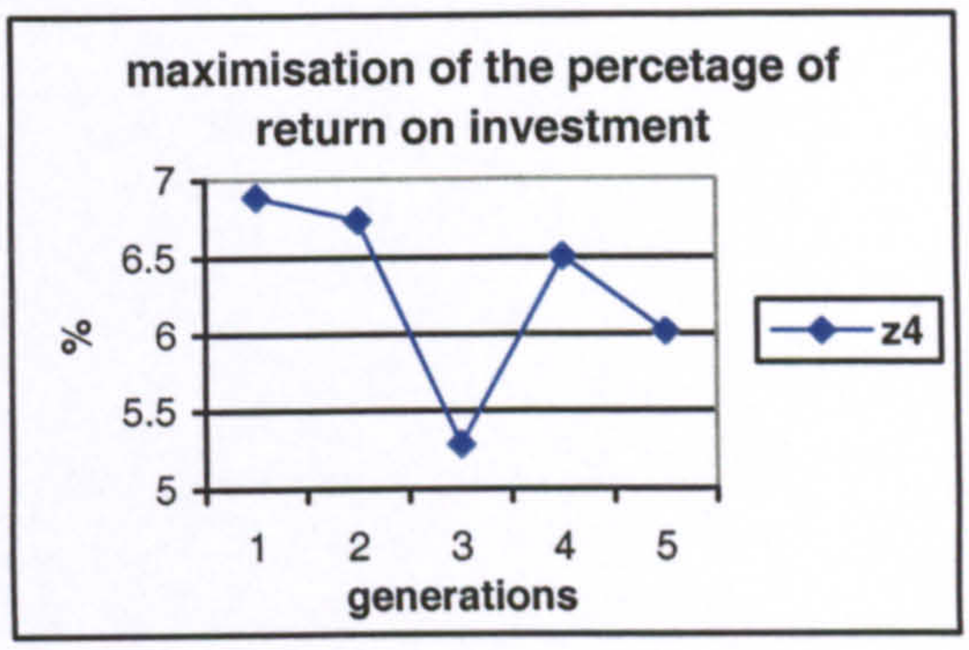
UX

Generation	z_1	z_2	z_3	z_4	z_5	z_6
1	5.761	3.228	5.502	6.884	0.256	1.767
2	5.842	3.774	5.681	6.735	0.544	1.748
3	6.161	4.903	6.031	5.282	1.508	1.761
4	5.935	4.975	6.063	6.503	2.035	1.702
5	6.059	4.953	6.052	6.012	1.785	1.718

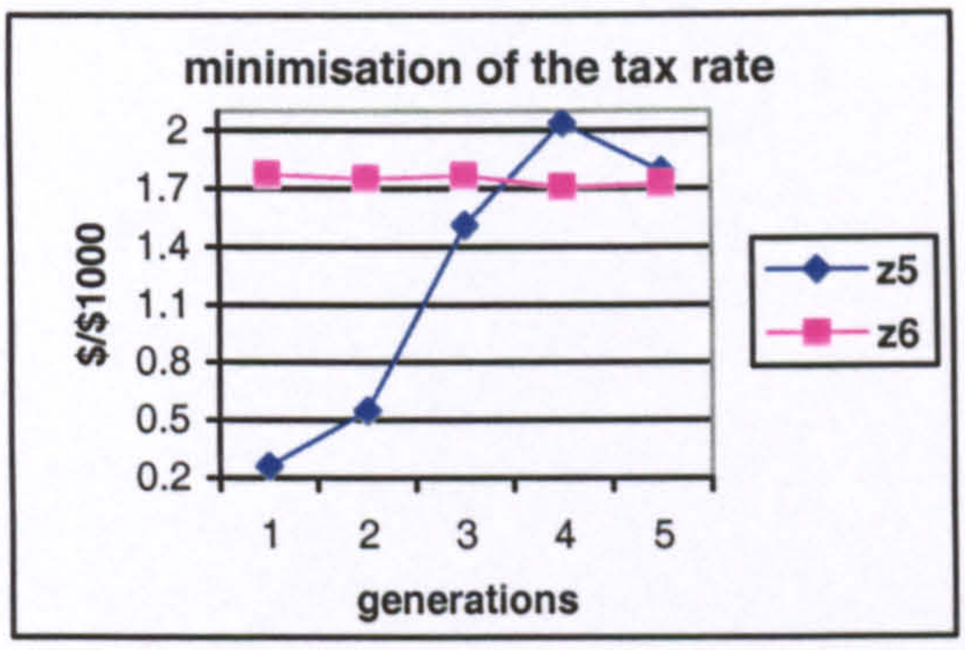
Table 4.13 Final values of z_1, z_2, z_3, z_4, z_5 and z_6 for the UX



(a)



(b)



(c)

Figure 4.11 (a) Maximisation of DO concentration for the UX (b) Maximisation of the percentage of return on investment for the UX and (c) Minimisation of the tax rate for the UX

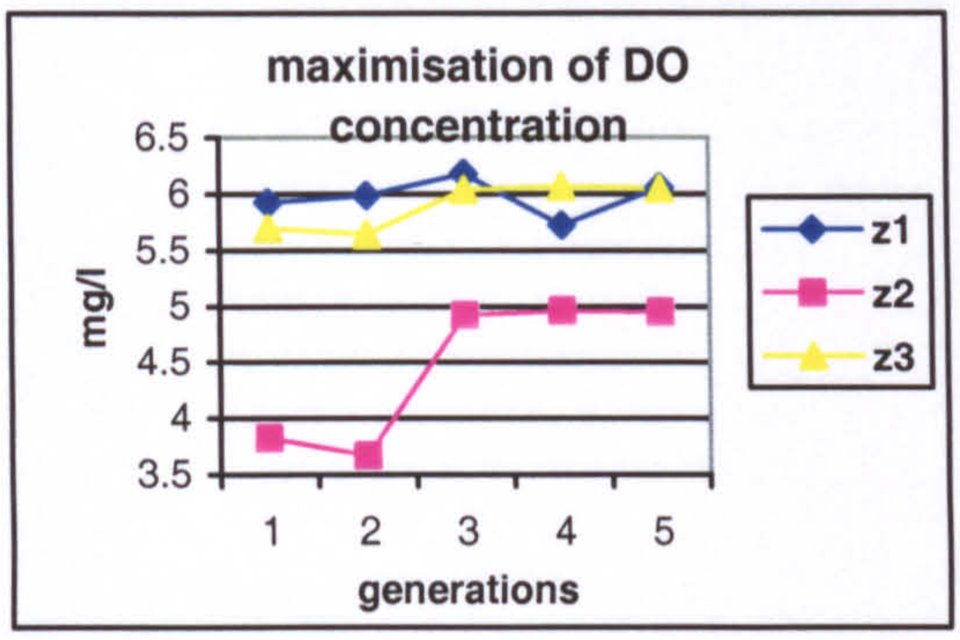
If the contour lines in Figure 4.10 are compared from those in Figure 4.11, it is evident that they have almost the same shapes.

Table 4.14 shows the goal values using the SBX and Figure 4.12 is built using those values.

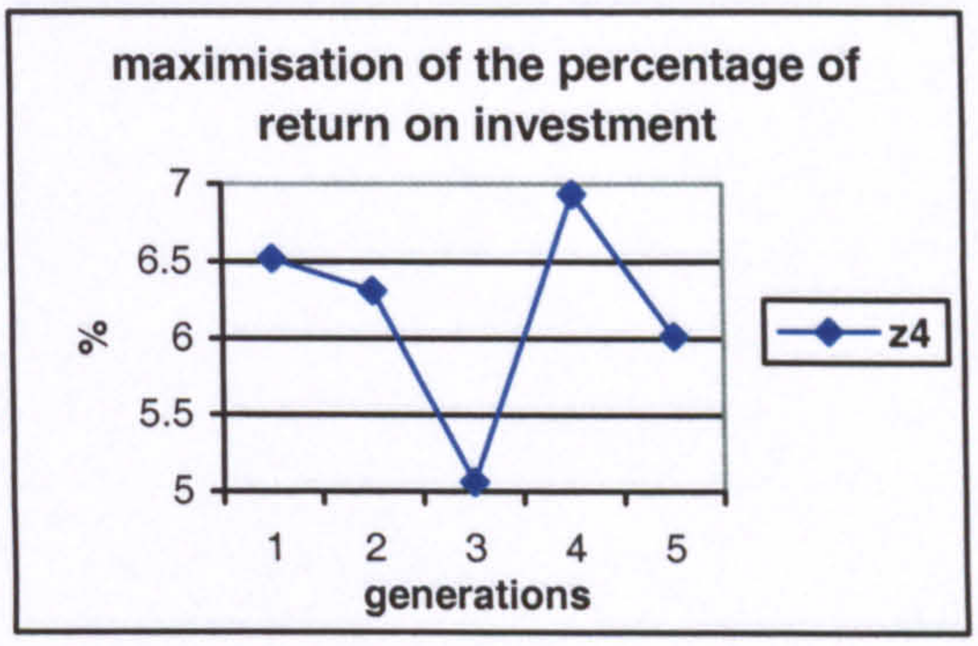
SBX

Generation	z_1	z_2	z_3	z_4	z_5	z_6
1	5.928	3.823	5.694	6.521	0.537	1.860
2	5.992	3.670	5.641	6.310	0.406	1.752
3	6.181	4.928	6.037	5.063	1.508	1.702
4	5.728	4.962	6.063	6.935	2.300	1.796
5	6.059	4.953	6.052	6.012	1.785	1.718

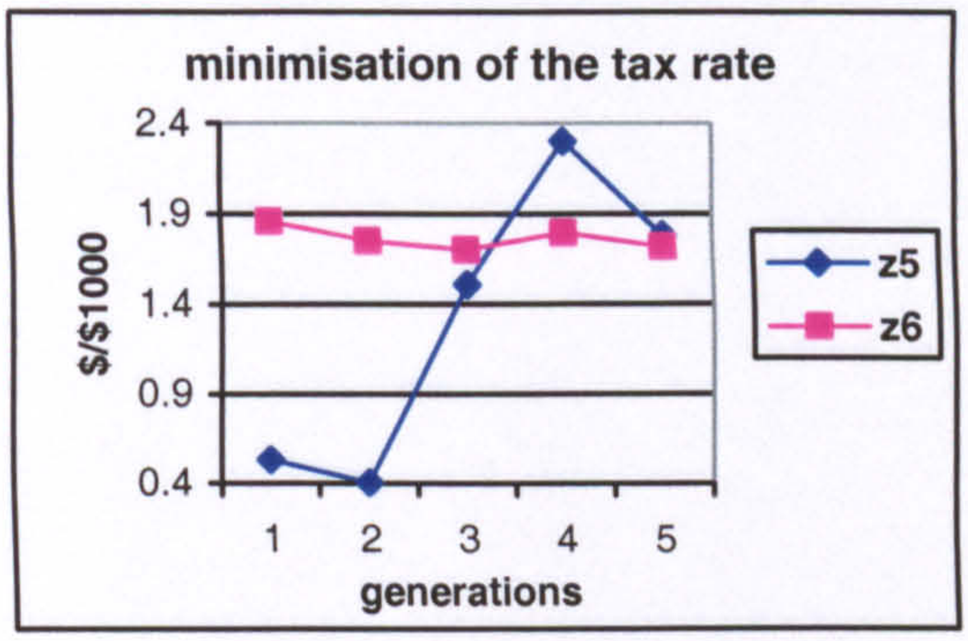
Table 4.14 Final values of z_1, z_2, z_3, z_4, z_5 and z_6 for the SBX



(a)



(b)



(c)

Figure 4.12 (a) Maximisation of DO concentration for the SBX
(b) Maximisation of the percentage of return on investment for the SBX and
(c) Minimisation of the tax rate for the SBX

If the contour lines in Figure 4.12 are compared from those in Figure 4.10 and 4.11, it is evident that they have almost the same shapes. Consequently, it is clear that the results for the three crossovers are very similar. Thus, the main conclusion is that the method is robust enough to generate feasible and valid solutions without mattering the kind of genetic operator used and moreover, SEMOPS can be used with different search methods yielding accurate results.

The next issue to consider is what happens if the aspiration levels are modified. After analysing the result with the following final aspiration levels:

$$AL_{F1} = 6.0,\ AL_{F2} = 4.9,$$
$$AL_{F3} = 6.0,\ AL_{F4} = 6.0,$$

$$AL_{F5} = 1.8,\ AL_{F6} = 1.8,$$

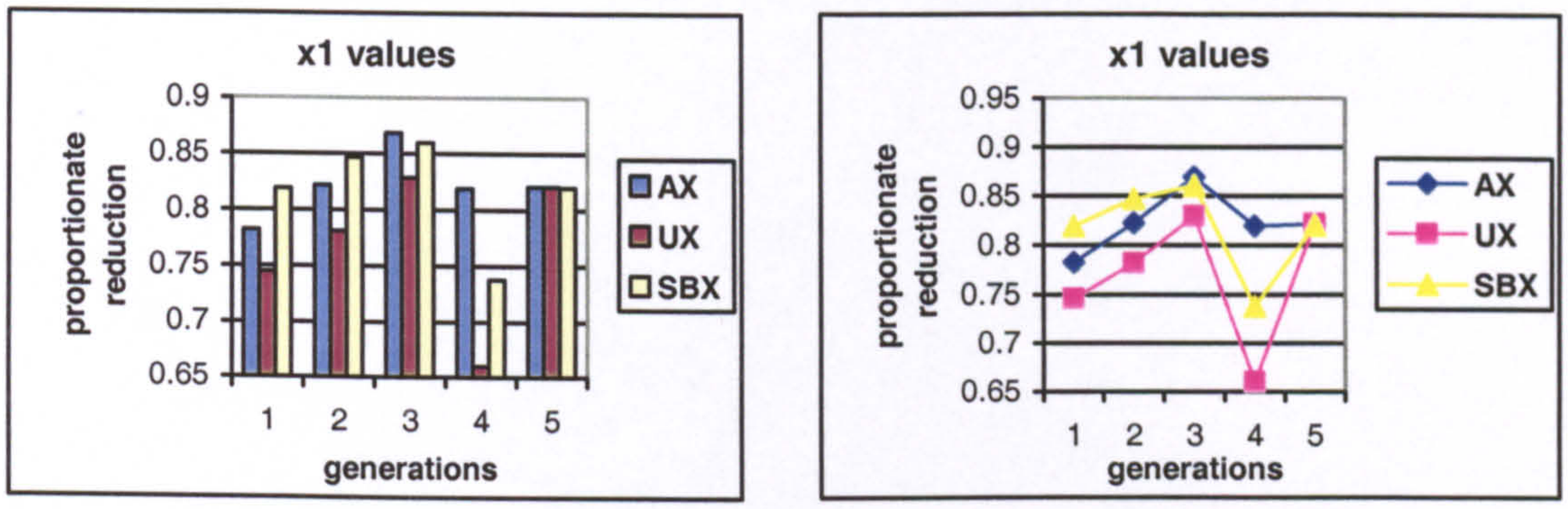
The problem was solved using different aspiration levels. Once again the problem was solved using three different crossovers (AX, UX and SBX). The results are presented in Appendix A and in the following Tables and Figures just the final results are analysed. In this occasion the final aspiration levels are:

$$AL_{F1} = 5.9,\ AL_{F2} = 5.0,$$
$$AL_{F3} = 6.0,\ AL_{F4} = 6.5,$$
$$AL_{F5} = 2.3,\ AL_{F6} = 1.8,$$

Table 4.15 presents the final values of the decision variable x_1 (proportionate reduction in carbonaceous BOD load at the Pierce Hall Cannery) for the three different crossovers (AX, UX and SBX) and the slope change in each generation, using the final aspiration levels defined above. Additionally, Figure 4.13 is constructed using these values.

Generation	AX (x_1)	Slope Value	Slope Sign	UX (x_1)	Slope value	Slope sign	SBX (x_1)	Slope value	Slope sign
1	0.782	------	------	0.745	------	------	0.819	------	------
2	0.822	0.04	+	0.781	0.036	+	0.847	0.028	+
3	0.869	0.047	+	0.829	0.048	+	0.860	0.013	+
4	0.819	-0.05	-	0.660	-0.169	-	0.738	-0.122	-
5	0.821	0.002	+	0.821	0.161	+	0.820	0.082	+

Table 4.15 Final values of x_1 and slope values for the AX, UX and SBX



(a) (b)

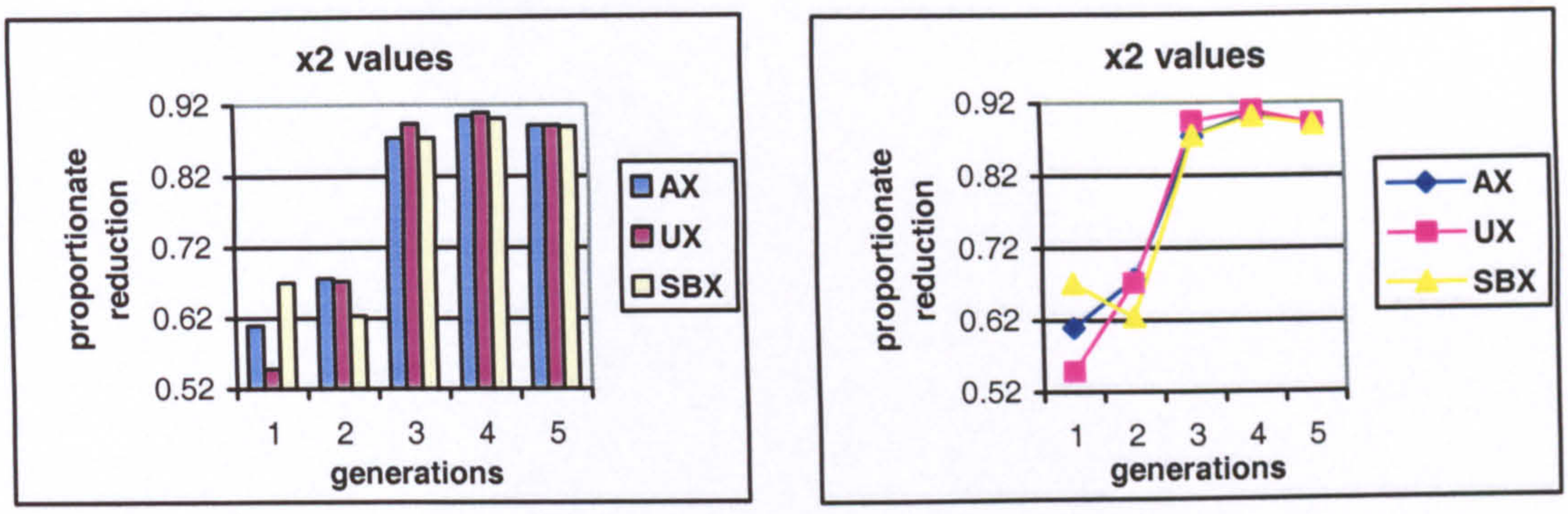
Figure 4.13 Proportionate reduction in carbonaceous BOD at the Pierce-Hall Cannery

From Figure 4.13(a), the results indicate that the proportionate reduction in carbonaceous BOD load in the Pierce-Hall Cannery after the fifth generation using the AX and UX is the same (82.1%), whilst the SBX is 82% (Table 4.15). The difference between these values is 0.12%. If Figure 4.13 is compared with Figure 4.4, it can be seen that the shape of the contour lines for both figures are very similar, although the final x_1 value in Figure 4.4 differs approximately 6% from the value in Figure 4.13.

Table 4.16 shows the final values of the decision variable x_2 (proportionate reduction in carbonaceous BOD load in the town of Bowville) for the three different crossovers (AX, UX and SBX) and the slope change in each generation. Additionally, Figure 4.14 is built using these values.

Generation	AX (x_2)	Slope Value	Slope sign	UX (x_2)	Slope value	Slope sign	SBX (x_2)	Slope value	Slope sign
1	0.609	------	------	0.548	------	------	0.670	------	------
2	0.676	0.067	+	0.672	0.124	+	0.623	-0.047	-
3	0.874	0.198	+	0.894	0.222	+	0.874	0.251	+
4	0.905	0.031	+	0.909	0.015	+	0.901	0.027	+
5	0.891	-0.014	-	0.891	-0.018	-	0.889	-0.012	-

Table 4.16 Final values of x_2 and slope values for the AX, UX and SBX



(a) (b)

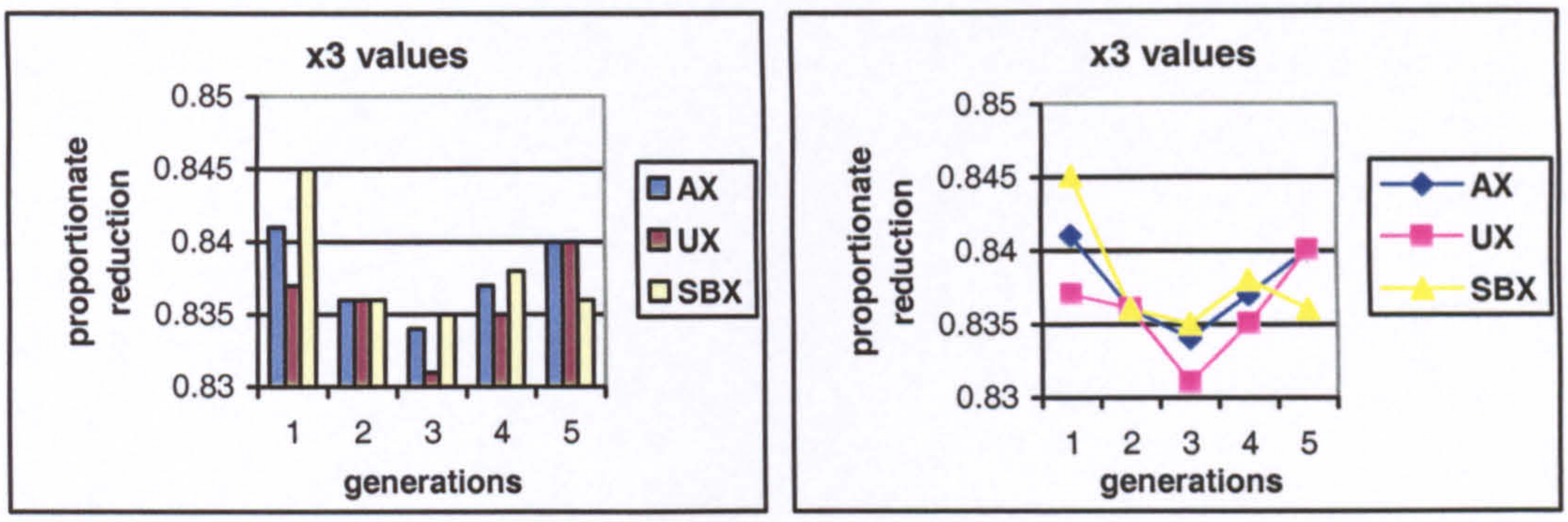
Figure 4.14 Proportionate reduction in carbonaceous BOD in Bowville

From Figure 4.14(a), it can be seen that the proportionate reduction in carbonaceous BOD load in Bowville after the fifth generation using the AX and UX is the same (89.1%). The results with the SBX (88.9%) differ from those found with the AX and UBX (Table 4.16). It is obvious that the behaviour of the contour line of each crossover is almost the same (Figure 4.14(b)) with the exception of the second generation of SBX where the x_2 value decreases instead of increasing as in the two other crossovers. If Figure 4.14 is compared with Figure 4.5, it can be seen that the contour lines for both figures are very similar, although the final x_2 value in Figure 4.5 differs approximately 3% from the value in Figure 4.14.

Table 4.17 shows the final values of the decision variable x_3 (proportionate reduction in carbonaceous BOD load at the town of Plympton) for the three different crossovers (AX, UX and SBX) and the slope change in each generation. Additionally, Figure 4.15 is constructed using these values.

Generation	AX (x_3)	Slope Value	Slope sign	UX (x_3)	Slope value	Slope sign	SBX (x_3)	Slope value	Slope sign
1	0.841	------	------	0.837	------	------	0.845	------	------
2	0.836	-0.005	-	0.836	-0.001	-	0.836	-0.009	-
3	0.834	-0.002	-	0.831	-0.005	-	0.835	-0.001	-
4	0.837	0.003	+	0.835	0.004	+	0.838	0.003	+
5	0.840	0.003	+	0.840	0.005	+	0.836	-0.002	-

Table 4.17 Final values of x_3 and slope values for the AX, UX and SBX



(a) (b)

Figure 4.15 Proportionate reduction in carbonaceous BOD in Plympton

From Figure 4.15(a), it is possible to see that the proportionate reduction in carbonaceous BOD load at Plympton after the fifth generation using the AX and UX is the same (84%). The results with the SBX differ from those found with the UX and SBX (Table 4.17), where the value after the fifth generation is 83.6%. It is obvious, that the results represented in Figure 4.15(b) differ from those in Figures 4.13(b) and 4.14(b), where the initial values of x_3 for the three crossovers (AX, UX, SBX) are low and the slopes for the next generation are positive. Despite the results in Figure 4.15 appearing to be distinct from each other, the x_3 values are very close in each generation, this is shown in Table 4.17 where the slopes' values are very small. These values indicate that the changes from one generation to the other are very smooth. If Figure 4.15 is compared with Figure 4.6, it can be seen that the shape of each contour line for both figures is not similar at all. Therefore it is possible to conclude that the decision variable where the changes on the aspiration levels are reflected, is x_3.

4.5 Fuzzy Rule-Based System

In this section the multiobjective genetic optimiser proposed by Fonseca and Fleming has been modified with a fuzzy rule-based system (FRBS) taking the place of the DM in order to achieve an automated process to solve the Bow River case study presented in section 4.4. Duenas and Mort (2001) proposed this modified model, where the optimiser is divided into the GA and the FRBS as shown in Figure 4.16.

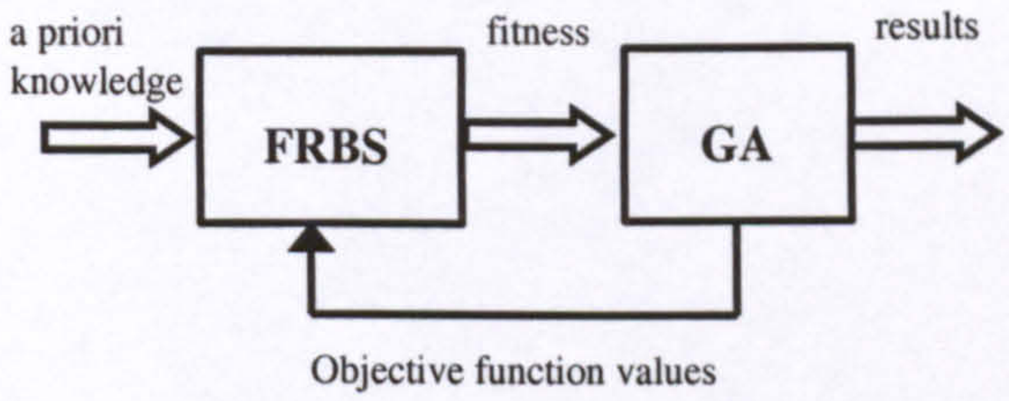


Figure 4.16 Fuzzy Multiobjective Genetic Optimiser

Voget and Kolonko (1998) developed a fuzzy genetic algorithm using fuzzy rule-based control of the selection procedure and the fitness function as an adaptive selection mechanism. In this section the same procedure is followed, but using the FRBS to determine which goal will act as a constraint in the next generation. The classification system for the fuzzy rule-based system was defined following the DM's decision process.

For the multiobjective genetic optimiser presented in Figure 4.16, the fuzzy rule-based system is responsible for simulating the decision making process assuming the DM's role. As argued

by Voget and Kolonko (1998) to create a fuzzy classification it is necessary to define how to measure the deviation for the goals to be able to specify the fuzzy sets.

Consider that, for each generation, the GA will give a solution for each goal that consists of a principle problem and P auxiliary problems (see equations 4.6 to 4.9). It is possible to calculate the mean $\overline{\mu_i}$ and the standard deviation Δ_i for each objective. The problem with this solution is that it is impossible to determine which goal is closest to its aspiration level because the measure in the standard deviation is calculated with respect to the mean. To define a deviation Δ_i that considers the aspiration level AL for each goal z the following equation is used:

$$\Delta_i = \sqrt{\frac{\sum_{i=0}^{P}(z_i - AL)^2}{P+1}} \tag{4.30}$$

Having the deviation for each goal means it is now possible to define the fuzzy sets (Voget and Kolonko, 1998). There will be two fuzzy sets T_i and R_i for each goal where:

T_i represents the predicate "deviation Δ_i is acceptable",
R_i represents the predicate "deviation Δ_i is too high", and $i = 1, 2, \ldots, n$ (n is the number of goals in each generation).

It is necessary to determine the membership functions m for the fuzzy sets. If it is considered that the fuzzy sets T_i and R_i are complementary and that $m_{T_i}(t)$ represents the degree of acceptance of a deviation t and can take a value within the interval [0,1]. This deviation must be normalised in order to have values between 0 and 1. In order to normalise the highest value of all the deviations will be considered.

Having the deviation as the measure of acceptability it is possible to define a unique membership function for the six goals:

$$m_{T_i}(t) = t \qquad \text{and} \qquad m_{R_i}(t) = 1 - t \tag{4.31}$$

where $0 \leq t \leq 1$.

The complete fuzzy classification with respect to the twelve predicates will be the sequences of their membership values. Γ_n represents the result for the first generation

$$\begin{aligned} M(\Gamma_n) &= (m_{T_1}, m_{T_2}, m_{T_3}, m_{T_4}, m_{T_5}, m_{T_6}, m_{R_1}, \\ &\quad m_{R_2}, m_{R_3}, m_{R_4}, m_{R_5}, m_{R_6})(\Gamma_n) \\ &= (\Delta_1, \Delta_2, \Delta_3, \Delta_4, \Delta_5, \Delta_6, 1-\Delta_1, 1-\Delta_2, 1-\Delta_3, \\ &\quad 1-\Delta_4, 1-\Delta_5, 1-\Delta_6) \end{aligned} \tag{4.32}$$

It is necessary to consider that this fuzzy classification will be modified for each generation because in each generation one goal will become a constraint.

The next step is to create the set of rules. The kind of rules used are "IF <condition> THEN <action>". Voget and Kolonko (1998) defined the conditions as AND-combinations of the predicates T_i and R_i. The actions to reduce deviation in this problem will be defined as the selection of a goal to act as a constraint in the next generation of the GA. Table 4.18 describes the rules used in this application.

IF	R1	R2	R3	R4	R5	R6
	*	*	*	*	*	T_1
AND	T_2	R_2	R_2	T_2	T_2	T_2
AND	*	*	T_3	T_3	T_3	T_3
AND	*	T_4	R_4	R_4	R_4	T_4
AND	R_5	T_5	R_5	T_5	R_5	R_5
AND	R_6	T_6	T_6	T_6	R_6	T_6
THEN	G_6	G_2	G_2	G_4	G_4	G_5

Table 4.18 Set of Fuzzy Rules

* means it could be either acceptable or too high.

This means that in rule 1 (R1), if the deviation is acceptable in ("T_2"), and if it is too high in ("R_5"), and if it is too high in ("R_6"), the goal which will act as a constraint in the next generation is goal 6 "G_6".

4.5.1 Experimental Results

The GA is implemented in C; the initial population is expressed in floating point vectors generating randomly x_i vectors used to generate w_i vectors, d_i vectors and the s solution.
In this work, the algorithm was run for 50 cycles for each principal and auxiliary problem, finding the solutions shown in Table 4.19. The first row represents the solution of equation number 4.6 and the following rows represent the solution for the auxiliary problems. The aspiration levels are:

A= (6.0, 6.0, 6.0, 6.5, 1.5, 1.5)

Generation 1

	Decision variables			Goals						Fitness
	x_1	x_2	x_3	z_1	z_2	z_3	z_4	z_5	z_6	s
Surrogate	0.806	0.653	0.849	5.899	3.729	5.664	6.602	0.486	1.914	6.269
Auxiliary 1	0.786	0.668	0.843	5.854	3.764	5.677	6.709	0.530	1.844	5.203
Auxiliary 2	0.638	0.389	0.847	5.518	2.552	5.279	7.170	0.063	1.892	4.433
Auxiliary 3	0.819	0.644	0.849	5.929	3.711	5.657	6.519	0.461	1.934	5.222
Auxiliary 4	0.983	0.601	0.857	6.300	3.955	5.709	2.530	0.358	2.035	5.115
Auxiliary 5	0.743	0.942	0.853	5.756	5.431	6.207	6.891	3.764	1.971	5.371
Auxiliary 6	0.816	0.669	0.917	5.922	3.813	5.691	6.538	0.533	3.408	4.991
	Deviation			0.246	2.286	0.392	1.515	1.336	0.831	
	Normalised Deviation			0.107	1.00	0.17	0.659	0.581	0.361	
	Fuzzy set			T_1	R_2	T_3	R_4	R_5	T_6	

Table 4.19 Results after first generation

Comparing Table 4.18 to Table 4.19, it is possible to conclude that the FRBS determines that goal 2 has to be managed as a constraint with an aspiration level $z_2 \geq 5$ mg/l (DO level at Robin State Park), obtaining the results presented in Table 4.20 at the next generation. The aspiration levels are:

A= (6.0, 5.0, 6.0, 6.5, 1.5, 1.5)

Generation 2

	Decision variables			Goals					*Const	Fitness
	x_1	x_2	x_3	z_1	z_2	z_3	z_4	z_5	z_6	s
Surrogate	0.894	0.867	0.839	6.099	5.036	6.076	5.779	1.873	1.787	5.536
Auxiliary 1	0.812	0.891	0.856	5.914	5.062	6.090	6.563	2.270	2.029	4.842
Auxiliary 2	0.966	0.847	0.840	6.262	5.108	6.088	3.711	1.617	1.796	4.985
Auxiliary 3	0.939	0.847	0.833	6.202	5.014	6.063	4.806	1.609	1.710	4.170
Auxiliary 4	0.789	0.936	0.836	5.860	5.428	6.204	6.696	3.504	1.748	4.127
Auxiliary 5	0.894	0.866	0.921	6.099	5.026	6.073	5.776	1.852	3.542	4.332
Deviation				0.161	0.183	0.010	1.399	0.903	0.887	
Normalised Deviation				0.115	0.130	0.078	1.00	0.645	0.633	
Fuzzy set				T_1	T_2	T_3	R_4	R_5	R_6	

Table 4.20 Results after the second generation

Now comparing Table 4.20 to Table 4.18 the FRBS determines that modifying the return on investment in the Pierce-Hall Cannery from 6.5 % to 6.0 % ($z_4 \geq 6.0$) will safeguard the future of the Cannery. The results of this are shown in Table 4.21 (next generation). The aspiration levels are:

A= (6.0, 5.0, 6.0, 6.0, 1.5, 1.5)

Generation 3

	Decision variables			Goal	*Const	Goals			*Const	Fitness
	x_1	x_2	x_3	z_1	z_2	z_3	z_4	z_5	z_6	s
Surrogate	0.824	0.886	0.856	5.940	5.042	6.084	6.488	2.180	2.018	4.795
Auxiliary 1	0.607	0.922	0.851	5.447	5.086	6.103	7.227	3.021	1.955	4.301
Auxiliary 2	0.664	0.907	0.861	5.577	5.010	6.079	7.117	2.623	2.104	4.228
Auxiliary 3	0.782	0.951	0.834	5.844	5.581	6.251	6.730	4.202	1.727	3.137
Auxiliary 4	0.845	0.894	0.926	5.987	5.139	6.112	6.328	2.328	3.729	3.536
Deviation				0.320	0.270	0.141	0.852	1.550	1.082	
Normalised Deviation				0.207	0.175	0.091	0.550	1.00	0.698	
Fuzzy set				T_1	T_2	T_3	R_4	R_5	R_6	

Table 4.21 Results after the third generation

Once again comparing Table 4.21 to Table 4.18 the FRBS identifies that goal 6 has to be managed as a constraint with an aspiration level $z_6 \leq 1.8$ (tax rate at Plympton), obtaining the results presented in Table 4.22. The aspiration levels are:

A= (6.0, 5.0, 6.0, 6.0, 1.5, 1.8)

Generation 4

	Decision variables			Goal	*Const	Goal	*Const	Goal	*Const	Fitness
	x_1	x_2	x_3	z_1	z_2	z_3	z_4	z_5	z_6	s
Surrogate	0.846	0.885	0.834	5.991	5.075	6.092	6.316	2.168	1.727	3.431
Auxiliary 1	0.773	0.902	0.837	5.825	5.098	6.103	6.769	2.510	1.762	2.656
Auxiliary 2	0.778	0.899	0.837	5.836	5.075	6.096	6.747	2.430	1.758	2.648
Auxiliary 3	0.824	0.982	0.830	5.941	6.040	6.389	6.483	6.722	1.677	1.949
Deviation				0.124	0.525	0.212	0.609	2.720	0.233	
Normalised Deviation				0.045	0.193	0.078	0.224	1.00	0.086	
Fuzzy set				T_1	T_2	T_3	T_4	R_5	T_6	

Table 4.22 Results after the fourth generation

Finally, the aspiration level of goal 5 is modified to $z_5 \leq 1.8$. The results presented by the GA using the DM opinion after the fifth iteration are $x_1 = 0.873$, 87 % Pierce-Hall Cannery, $x_2 = 0.861$, 86 % Bowville, and $x_3 = 0.839$ 84 % Plympton.

Considering $z = (z_1, z_2, z_3, z_4, z_5, z_6)$ then the results for the GA are $z = (6.05, 4.95, 6.05, 6.05, 1.79, 1.79)$. These results were obtained following the DM's goal selection procedure: goal 6 (z_6) followed by goal 2 (z_2) followed by goal 4 (z_4), followed by goal 5 (z_5).

$z_6 \rightarrow z_2 \rightarrow z_4 \rightarrow z_5$

The results of the multiobjective optimiser are $x_1 = 0.875$ (88 % Pierce-Hall Cannery), $x_2 = 0.882$ (88 % Bowville), and $x_3 = 0.847$ (85 % Plympton) and the results of the GA are $z = (6.06, 5.11, 6.1, 6.02, 2.11, 1.81)$. These results were obtained following the FRBS's goal selection procedure: goal 2 (z_2) followed by goal 4 (z_4) followed by goal 6 (z_6), followed by goal 5(z_5).

$z_2 \rightarrow z_4 \rightarrow z_6 \rightarrow z_5$

If the results of the GA are compared to the results of the multiobjective optimiser, it is possible to conclude that even though the results are not the same they are very close. It is important to notice that to be able to compare the results the criteria followed by the DM in each method were the same.

Having solved the Bow River Valley case study (using GA and fuzzy logic) and having compared the results with those found by the Authors of the method (Monarchi et al., 1973), it was decided to solve another problem using the same method (SEMOPS) and GAs. This decision was made in order to prove the consistency of the method as well as using another kind of GA. The problem selected is the nurse-scheduling problem using binary coding and multiple-point crossovers.

4.6 The Nurse-Scheduling Problem (NSP)

The NSP is defined as the generation of a schedule of working days and days off for each nurse in a hospital, where a working day will be divided into three different shifts: day shift (8:00-16:00), evening shift (16:00-24:00), and night shift (00:00-8:00). Normally, this problem is solved by hand by head-nurses and it takes a long time and a great deal of effort.

The NSP involves the optimisation of several objectives such as maximisation of nurse satisfaction, maximisation of fairness, and minimisation of hospital costs. In this section the NSP is solved using a hybrid approach of SEMOPS in conjunction with a GA.

There are different approaches in the literature to solving the NSP problem. Berrada et al. (1996) solved the NSP with a multi-objective approach using tabu-search and objectives' prioritisation. Cheng et al. (1996) developed a redundant modelling approach that consists of generating a slot model, a shift model, and an aggregate model (the first two models are connected) generating constraints for each of the models. Dowsland (1998) used tabu-search and strategic oscillation focused on the nurses' working practices rather than the minimisation of costs. Jaumard et al. (1998) divided the NSP into two: a master and an auxiliary problem. The master problem minimises the salary cost and maximises the employee preferences, while the auxiliary problem considers as a feasible solution an acceptable schedule for a given nurse. Jan et al. (2000) used a co-operative genetic algorithm optimising the individual fitness of each nurse and also the fitness of the entire schedule. Finally, Valouxis and Housos (2000) developed hybrid optimisation techniques utilising the strengths of operations research and

artificial intelligence. Basically, their solution merges integer linear programming and local search techniques.

It is possible to say that the approach presented in this report could be considered as a hybrid approach that uses the strengths of both operations research and evolutionary computation techniques.

The NSP is a good example of considering the preferences of the nursing staff as much as those of the hospital. To measure the performance of the GA this problem was solved using a two-point crossover and a four-point crossover in such a way as to compare the results obtained.

4.6.1 Problem Definition

The NSP is a multiple objective optimisation problem (MOP). The main objective of the NSP is to find a schedule that meets all the constraints.
Table 4.23 contains the symbols used to denote the different shifts throughout the solution of this problem.

Shift	Symbol
day-shift	*d*
evening-shift	*e*
night-shift	*n*
day off	*o*

Table 4.23 The symbols used to represent each shift.

The decision variables are defined as follows (Jan et al., 2000):

$$x_{ijw} = \begin{cases} 1 & \text{if nurse } i \text{ works } w \text{ shift on day } j \\ 0 & \text{otherwise} \end{cases} \tag{4.33}$$

where x_{ijw} represents the decision variable that expresses nurse i working a wth shift on a day j.

Jan et al. (2000) proposed representing the NSP using a matrix $M \times N$, having N number of nurse to be scheduled and M number of days to be scheduled. Berrada et al. (1996) stated that to make the problem easier to solve, it is better to have a number of short-term horizons rather than a single long time horizon. They suggested a term horizon ranging from 2 to 4 weeks. However, in this approach, the number of nurses to be scheduled is 8 and the short-term horizon is 1 week (7 days). It is supposed there are two different categories of nurses according to their experience: junior and senior.

4.6.1.1 Constraints Definition

Generally, constraints are divided into two categories: hard and soft constraints Deb (2001). On the one hand, hard constraints are those that have to be satisfied in order to obtain a feasible solution. Soft constraints, on the other hand, are those that it is desirable to satisfy. In other words, if a soft constraint is not satisfied the solution is still feasible.

Hard constraints

1. Each nurse can work only one shift a day

$$\sum_{w=1}^{3} x_{ijw} = 1 \quad i = 1,\ldots, \mathrm{N}; \qquad j = 1,\ldots, \mathrm{M} \tag{4.34}$$

2. For each day, there must be at least 1 nurse working the day shift, 1 nurse working the evening shift, and 1 nurse working the night shift.
3. Each nurse must have one day off per week.

$$\sum_{w=1}^{3} \sum_{j=1}^{M} x_{ijw} = 6 \tag{4.35}$$

Soft constraints

Soft constraints are defined as preference rules (Cheng et al., 1996). In other words, these are the nurses' preferences and they will be satisfied as far as possible, although their violation does not affect the schedule's feasibility.

1. After a night shift a nurse prefers not to have a day shift.
2. At least one senior nurse is present on each shift.

4.6.1.2 Objective Functions Definition

The first objective to consider is the minimisation of hospital costs subject to the hard constraints:

$$\min \sum_{j=1}^{M} \sum_{i=1}^{N} c_i \sum_{w=1}^{3} x_{ijw} \tag{4.36}$$

where c_i is a weight assigned to the ith nurse, where values are 2 for a senior nurse and 1 for a junior nurse. This weight represents the relationship between the different levels of nurses and their salaries.
The second objective to consider is the minimisation of the entire schedule fitness T subject to the constraints. This objective is defined as follows:

$$\min T \tag{4.37}$$

where T is the addition of the individual schedule fitness I_i for each nurse:

$$T = \sum_{i}^{N} I_i \tag{4.38}$$

I_i is the fitness of the ith nurse individual schedule, calculated using the values presented in Table 4.24

Days pattern	Assigned value
n d	60
n e	10
e d	20

Table 4.24 Assigned fitness values

The other possible combinations have an assigned value of 0. For instance, the assigned value to "o n" or "o d" is 0.

The aspiration level of the entire schedule fitness *T*, subject to the constraints, is 180. In the first iteration the aspiration levels are equal to the goal levels.

4.6.2 Genetic Algorithm Definition

To solve the problem a GA is implemented; the initial population of 960 is expressed in binary code and is generated randomly. For this problem the algorithm is run for 100, 200 and 500 cycles. The selection method used is tournament selection (Bäck et al., 2000) where the tournament size is two (binary tournament).

The initial crossover used is a two-point crossover, as defined by Eshelman et al. (1989). The two-point crossover consists of a chromosome considered as a ring and two numbers randomly selected which will segment the chromosome and will produce two offspring. The mutation operator's probability is 0.15.

In an attempt to optimise the GA it has been decided to use a four-point crossover operator. De Jong (1975) proposed a generalised crossover model where a new variable was considered. This variable is the number of crossover points and is called CP. In the one–point crossover case CP is equal to 1; for the two-point crossover CP is equal to 2. The chromosome is still considered as a ring for an even CP; for an odd CP one of the crossing points is considered fixed in position 0. Both of these cases produce two new children. If CP is greater than 2 the crossover is known as multiple-point.

4.6.3 Experimental Results

Tables 4.25, 4.26 and 4.27 show the best-obtained schedule using the two-point crossover with 100, 200 and 500 cycles. It is also shown how all the hard constraints are met. This makes these preliminary schedules feasible solutions.

100 cycles

	Day 1	**Day 2**	**Day 3**	**Day 4**	**Day 5**	**Day 6**	**Day 7**	**Days worked**	**Fitness**
nurse 1	n	n	e	o	d	n	n	6	10
nurse 2	e	e	d	n	e	e	o	6	30
nurse 3	n	n	o	n	d	d	n	6	60
nurse 4	o	d	d	e	e	d	n	6	20
nurse 5	o	n	e	e	n	d	e	6	70
nurse 6	d	e	e	e	e	o	d	6	0
nurse 7	e	o	d	d	d	d	e	6	0
nurse 8	o	n	n	e	n	n	n	6	10
# nurses day-shift	1	1	3	1	3	4	1	**Total fitness**	***T*=200**
# nurses evening-shift	2	2	3	4	3	1	2		
# nurses night-shift	2	4	1	2	2	2	4		
# nurses day off	3	1	1	1	0	1	1		
Total of nurses	8	8	8	8	8	8	8		

▢ Represents a night shift followed by a day shift

Table 4.25 Best solution for 100 cycles using two-point crossover

200 cycles

	Day 1	Day 2	Day 3	Day 4	Day 5	Day 6	Day 7	Days worked	Fitness
nurse 1	d	e	e	e	n	o	d	6	0
nurse 2	o	e	d	e	d	d	n	6	40
nurse 3	d	n	e	n	n	o	n	6	10
nurse 4	d	e	o	d	n	n	e	6	10
nurse 5	d	e	n	o	d	n	n	6	0
nurse 6	e	n	n	o	e	d	d	6	20
nurse 7	o	d	d	n	d	e	d	6	80
nurse 8	n	e	e	n	e	o	d	6	20
# nurses day-shift	4	1	2	1	3	2	4	**Total fitness**	***T*=180**
# nurses evening-shift	1	5	3	2	2	1	1		
# nurses night-shift	1	2	2	3	3	2	3		
# nurses day off	2	0	1	2	0	3	0		
Total of nurses	8	8	8	8	8	8	8		

▢ Represents a night shift followed by a day shift

Table 4.26 Best solution for 200 cycles using two-point crossover

500 cycles

	Day 1	Day 2	Day 3	Day 4	Day 5	Day 6	Day 7	Days worked	Fitness
nurse 1	n	o	e	n	e	d	d	6	30
nurse 2	n	n	e	o	d	n	n	6	10
nurse 3	o	d	d	d	n	e	e	6	10
nurse 4	e	e	d	e	e	e	o	6	20
nurse 5	d	d	o	d	e	n	n	6	0
nurse 6	d	n	o	n	e	e	d	6	30
nurse 7	d	e	e	d	d	d	o	6	20
nurse 8	o	n	n	e	n	e	n	6	20
# nurses day-shift	3	2	2	3	2	2	2	**Total fitness**	***T*=140**
# nurses evening-shift	1	2	3	2	4	4	1		
# nurses night-shift	2	3	1	2	2	2	3		
# nurses day off	2	1	2	1	0	0	2		
Total of nurses	8	8	8	8	8	8	8		

Table 4.27 Best solution for 500 cycles using two-point crossover

100 cycles

	Day 1	Day 2	Day 3	Day 4	Day 5	Day 6	Day 7	Days worked	Fitness
nurse 1	n	n	o	d	d	d	d	6	0
nurse 2	d	d	d	o	e	e	e	6	0
nurse 3	e	e	o	e	d	d	d	6	20
nurse 4	n	n	e	o	n	n	e	6	20
nurse 5	e	o	n	e	e	e	e	6	10
nurse 6	n	e	o	n	n	n	e	6	20
nurse 7	n	o	e	d	e	n	d	6	80
nurse 8	d	d	e	n	o	n	n	6	0
# nurses day-shift	2	2	1	2	2	2	3	**Total fitness**	***T*=150**
# nurses evening-shift	2	2	3	2	3	2	4		
# nurses night-shift	4	2	1	2	2	4	1		
# nurses day off	0	2	3	2	1	0	0		
Total of nurses	8	8	8	8	8	8	8		

▢ Represents a night shift followed by a day shift

Table 4.28 Best solution for 100 cycles using four-point crossover

200 cycles

	Day 1	Day 2	Day 3	Day 4	Day 5	Day 6	Day 7	Days worked	Fitness
nurse 1	e	o	d	d	e	d	e	6	20
nurse 2	e	n	e	o	e	d	d	6	30
nurse 3	n	o	n	n	n	n	n	6	0
nurse 4	d	e	n	n	n	o	d	6	0
nurse 5	d	e	d	d	n	o	e	6	20
nurse 6	e	d	o	e	n	e	d	6	50
nurse 7	d	d	e	d	o	d	n	6	20
nurse 8	e	e	d	o	d	e	e	6	20
# nurses day-shift	3	2	3	3	1	3	3	**Total fitness**	T=160
# nurses evening-shift	4	3	2	1	2	2	3		
# nurses night-shift	1	1	2	2	4	1	2		
# nurses day off	0	2	1	2	1	2	0		
Total of nurses	8	8	8	8	8	8	8		

Table 4.29 Best solution for 200 cycles using four-point crossover

500 cycles

	Day 1	Day 2	Day 3	Day 4	Day 5	Day 6	Day 7	Days worked	Fitness
nurse 1	n	n	n	e	o	d	n	6	0
nurse 2	d	e	d	e	e	o	e	6	20
nurse 3	d	d	e	e	n	o	e	6	0
nurse 4	n	e	o	n	e	e	e	6	20
nurse 5	d	d	n	e	e	o	d	6	10
nurse 6	n	e	d	d	d	d	o	6	30
nurse 7	d	e	n	n	n	o	d	6	0
nurse 8	e	d	o	d	d	n	n	6	20
# nurses day-shift	4	3	2	2	2	2	2	**Total fitness**	T=110
# nurses evening-shift	1	4	1	4	3	1	3		
# nurses night-shift	3	1	3	2	2	1	2		
# nurses day off	0	0	2	0	1	4	1		
Total of nurses	8	8	8	8	8	8	8		

Table 4.30 Best solution for 500 cycles using four-point crossover

4.6.4 Results Analysis

It can be seen that in the GA using the two-point crossover (Table 4.25) and 100 cycles, the best solution meets all the hard constraints represented by equations 4.34 and 4.35. However, the soft constraint "after a night shift a nurse prefers not to have a day shift" is not satisfied in two cases (nurse 3 and nurse 5). The total fitness T that represents the entire schedule fitness is 200. When compared with the DM's aspiration level of 180 the difference is 11%. It is considered an ideal solution when the total fitness T is equal to 0, therefore the value of T can be considered as a measure of how far the solution found is from the ideal solution. In this case the solution's total fitness is 11% above the aspiration level.

After 200 cycles, it is evident that the solution has been improved from fitness T=200 to T=180 (Table 4.26). In other words, the final fitness has been reduced 10%. Nonetheless, the DM's aspiration level has been met, it can be seen that the soft constraint "after a night shift a nurse prefers not to have a day shift" is not satisfied in one case (nurse 7). Therefore, it is necessary to run the algorithm with a greater number of cycles.

The results reveal that when the algorithm has executed 500 cycles, it finds a feasible solution that not only meets the hard constraints, but also the soft constraints. Hence, when analysing Table 4.27 it can be seen that each nurse is working one shift a day and has one day off a week. The results clearly show that none of the nurses is working a day shift after having a

night shift. If the schedule's total fitness (140) is compared with the aspiration level (180), it shows an improvement of 22.22%. Consequently, it is clear that the schedule shown in Table 4.27 is not only a feasible solution but also a desirable solution.

Table 4.28 shows the solution using a four-point crossover and running the algorithm 100 cycles. It is possible to conclude that the results are very similar to those found using a two-point crossover. Again, all the hard constraints are met making the solution feasible. However, the soft constraint "after a night shift a nurse prefers not to have a day shift" is not satisfied for nurse 7. Comparing these results with those found using the two-point crossover (Table 4.25), it can be seen that there is a great improvement from one crossover to the other. The total fitness T in the two-point crossover is 200 while in the four-point crossover it is 150 yielding 25% improvement.

The results reveal that when the algorithm has completed 200 cycles (four-point crossover), it finds a feasible solution that not only meets the hard constraints, but also the soft constraints. Hence, in the analysis of Table 4.29 it can be seen that each nurse is working one shift a day and has one day off a week. Compared with the results found using the two-point crossover (T=180), the difference between the total fitness is more than 11%. The results clearly show that in the two-point crossover, one nurse (nurse 7) is working a day shift after having a night shift while in the four-point crossover none of them have to. Consequently, it is clear that the schedule shown in Table 4.29 is not only a feasible solution but also a desirable solution.

After 500 cycles, it is evident that the solution has been improved from an aspiration level T = 180 to a T = 110 (Table 4.30). In other words, the final fitness has been reduced almost 39%. When the two-point crossover results are compared to those of the four-point crossover, it is evident that both met the soft constraint "after a night shift a nurse prefers not to have a day shift". The main difference consists in the schedules' total fitness that in the two-point crossover it is 140 and in the four-point crossover it is 110. It can be concluded that in both cases the algorithm yields a desirable solution, although the performance of the four-point crossover is better.

The most important conclusion is that the four-point crossover general performance is better than the two-point crossover, yielding desirable solutions in a shorter period of time.

In order to improve the performance of the GA the penalty values (fitness values) assigned in Table 4.24 have been modified, resulting in the fitness values found in Table 4.31 as follows:

Days pattern	Assigned value
n d	120
n e	10
e d	40

Table 4.31 Assigned fitness values

The aspiration level of the entire schedule fitness T, subject to the constraints, is 340. In the first iteration the aspiration levels are equal to the goal levels.

Table 4.32 shows how the changes made in the penalty values affect the final solution. Although, the total fitness T increased (as was expected) the hard and soft constraints were satisfied.

500 cycles

	Day 1	Day 2	Day 3	Day 4	Day 5	Day 6	Day 7	Days worked	Fitness
nurse 1	n	o	e	n	e	d	d	6	50
nurse 2	n	n	e	o	d	n	n	6	10
nurse 3	o	d	d	d	n	e	e	6	10
nurse 4	e	e	d	e	e	e	o	6	40
nurse 5	d	d	o	d	e	n	n	6	0
nurse 6	d	n	o	n	e	e	d	6	50
nurse 7	d	e	e	d	d	d	o	6	40
nurse 8	o	n	n	e	n	e	n	6	20
# nurses day-shift	3	2	2	3	2	2	2	**Total fitness**	***T*=220**
# nurses evening-shift	1	2	3	2	4	4	1		
# nurses night-shift	2	3	1	2	2	2	3		
# nurses day off	2	1	2	1	0	0	2		
Total of nurses	8	8	8	8	8	8	8		

Table 4.32 Best solution for 500 cycles using two-point crossover

500 cycles

	day 1	day 2	day 3	day 4	day 5	day 6	day 7	Days worked	Fitness
nurse 1	n	n	n	e	o	d	n	6	10
nurse 2	d	e	d	e	e	o	e	6	40
nurse 3	d	d	e	e	n	o	e	6	0
nurse 4	n	e	o	n	e	e	e	6	20
nurse 5	d	d	n	e	e	o	d	6	10
nurse 6	n	e	d	d	d	d	o	6	50
nurse 7	d	e	n	n	n	o	d	6	0
nurse 8	e	d	o	d	d	n	n	6	40
# nurses day-shift	4	3	2	2	2	2	2	**Total fitness**	***T*=170**
# nurses evening-shift	1	4	1	4	3	1	3		
# nurses night-shift	3	1	3	2	2	1	2		
# nurses day off	0	0	2	0	1	4	1		
Total of nurses	8	8	8	8	8	8	8		

Table 4.33 Best solution for 500 cycles using four-point crossover

From Tables 4.27 and 4.32, it is evident that the final schedules using the values in Table 4.24 and the values in Table 4.31 are the same. Furthermore, from Tables 4.30 and 4.33, it is also evident that the final schedules using the values in Table 4.24 and the values in Table 4.31 are the same. To some extent, it would be fair to conclude that although the algorithm performs in the same way using different fitness assigned values, the results obtained using different crossover operators can differ considerably.

4.7 Summary

In this chapter the SEMOPS method has been presented as a tool used for the solution of multiple objective optimisation problems, considering the DM's preferences.

The main characteristics of SEMOPS are:

- It allows the DM to set aspiration levels and be able to modify them sequentially.
- It works with a surrogate function and a set of auxiliary problems.

In General, SEMOPS allows the direct programming of the GA due to the goal's transformation into the *d(x)* in the real positive set, using this as the fitness function.

Two problems were solved using SEMOPS and GAs: the Bow River Valley and the nurse-scheduling problem.

The Bow River Valley is a case study solved by the authors of the SEMOPS method, and for comparison purposes was solved using a GA instead of the cutting-plane programming technique. The results found by the GA are very close to the solutions found by Monarchi et al. (1973). The main conclusion that can be made is that SEMOPS is a robust method that allows in a direct way the interaction between the algorithm and the DM's preferences, giving it a major flexibility. The use of other recombination operators for real-valued GA, such as geometrical crossover and simplex crossover, constitute areas for further study.

Duenas and Mort (2001) proposed a modified multiobjective genetic optimiser using an FRBS instead of the DM. The results found by the GA are very close to the solutions found by the genetic optimiser. Nevertheless, using the FRBS will give the possibility of having self-sufficient systems. The application of this method for other Multiobjective Optimisation Problems constitutes an area for further study. Also, the use of other rules and the shape of the membership function could be changed. It is important to realise that in an MCDM problem the existence of the decision-maker necessarily introduces subjectivity that will be lost using FRBS. However, The FRBS does introduce flexibility, which is important in optimisation.

The nurse-scheduling problem was selected because is a scheduling problem that can be codified using binary coding and because it also involves the DM's preferences. This problem was solved using a hybrid model that includes SEMOPS and a GA. It was therefore shown that the use of a hybrid model gives the DM better control of the model's and the algorithms' outcomes. Moreover, the use of a GA makes the problem's solution more flexible and accurate, giving the programmer control of the genetic operators. The drastic changes occurred when the parameters of the problem were changed (assigned fitness values), showing again that the control of the actions to follow is now in the DM's hands. For future work, an area to explore could be the use of a FRBS to determine which is the best crossover operator to use, selected from a set of different operators, according to the parameters of the problem.

Finally, the most important conclusion is that the strongest point of SEMOPS is that it allows the programmer to manage goal levels and aspiration levels.

In the next chapter another interactive method that handles the DM's preferences is analysed. The main characteristic of this method is that it also considers risk and uncertainty in the solution of the problem.

CHAPTER 5

Probabilistic Trade-off Development Method

5.1 Introduction

Decisions in business, government, economics, engineering, and social matters are made every day. To make the right decision it is necessary to account for all the possible scenarios and to use all the data available. In the 1970's a variety of mathematical tools in the operations research area were developed to solve problems that involve these kinds of decisions (i.e. decision-making) (Goicoechea et al., 1982). Most of these tools work with a deterministic approach; to change this approach to a realistic one it is necessary to consider the impact of risk and uncertainty in the problem's solution. When problems exhibit significant uncertainty, which is generally quite difficult to deal with analytically, simulation is particularly useful. Over the years several techniques to solve problems through simulation have been developed.

Moreover most problems in practice consider the optimisation of several objectives simultaneously (Goicoechea et al., 1982), (Ignizio, 1982). As was presented in Chapter Three such problems are termed multiple objective optimisation problems (MOP) and one of their most important characteristics is that a large set of solutions is acceptable (these solutions are considered equivalent). Considering that, by definition MOPs are a subset of Multiple Criteria Decision-Making (MCDM) problems and that the solution methods for these problems were classified in three categories (deterministic, stochastic and fuzzy), it is interesting to analyse a method that involves a stochastic approach.

One of the objectives of this research is the solution of MCDM problems where the DM's preferences are considered as a pillar to yield a Pareto optimal solution[2]. In Chapter Four two MCDM problems were solved using a sequential method that involves the DM's preferences. These problems do not involve uncertainty or risk in their solutions because of the nature of the SEMOPS method. Then two questions arise: is it possible to find a method that incorporates both the DM's preferences and risk analysis? Is this method easy to implement using a GA?

The answer to both questions is yes and the method is called Probabilistic Trade-off Development method (PROTRADE). This method is a multiple objective stochastic method and was developed by Goicoechea et al. in 1979. Most importantly the PROTRADE method not only incorporates uncertainty and risk analysis in the solution of a problem but it also allows the decision-maker (DM) to introduce his preferences. Hence this is basically considered, as an interactive method that gives the information required by the GA to handle the fitness function. The PROTRADE method is presented and explained in section 5.2.

In this chapter a multiple use approach to land reclamation case study is solved using PROTRADE. The case study is described in section 5.3. The optimisation of this problem is carried out using a real-valued GA presented in section 5.5. Four models are developed to

---

[2] **Satisfactory solution**

handle the decision variables: one that considers neither risk nor probability of achievement, one that does not consider risk but does consider probability of achievement, one that considers risk but does not consider probability of achievement and a final one that considers both risk and probability of achievement. Section 5.4 contains the risk analysis performed to be able to develop the models. The results of the models are compared and discussed in section 5.6. Finally, the conclusions are presented in section 5.7.

5.2 Probabilistic Trade-off Development Method (PROTRADE)

Goicoechea et al. (1979) developed a multiobjective stochastic method called Probabilistic Trade-off Development (PROTRADE). This method is used basically to solve non-linear problems considering the DM's preferences (progressive articulation of preferences) and is capable of handling risk. The PROTRADE method consists of the formulation of surrogate and multiple attribute utility functions. On the one hand, the construction of the utility functions allows the inclusion of the DM's preferences in the problem's solution. On the other hand, the use of surrogate functions leads directly to the application of this method in GAs, where the surrogate functions are translated to the fitness function of the GA.

The DM is looking for the solution of a problem with several criteria although unconsciously visualises it as the optimisation of a unique function (Vincke, 1992). To represent this function there are some models and for the purposes of this research two models additive and multiplicative are discussed.

The additive model is one of the most used because of its simple form. The only constraint that this model imposes is the preferential independence of the criteria (Vincke, 1992).

$$U(a)=\sum_{i=1}^{n}U_i(f_i(a)) \tag{5.1}$$

where $f_i(a)$ is the ith criterion and a is an action that belongs to the space of possible actions to be followed.

The multiplicative model is represented by:

$$U(a)=\frac{\prod_{i=1}^{n}[1+k\,k_i U_i(a_i)]-1}{k} \tag{5.2}$$

where k and k_i are scaling constants.

The method applied in the solution of the case study uses a multiplicative model to incorporate the DM's preferences in an interactive way. It is important to mention that this model is not used to mathematically represent the function to optimise, but the DM's preferences. PROTRADE is a 12-step method defined as follows:

Step 1
A vector of objective functions is defined using the coefficients' expected values:

$$\begin{aligned}&\mathbf{z}(\mathbf{x})=[z_1(\mathbf{x}),z_2(\mathbf{x}),\ldots,z_p(\mathbf{x})],\\&g_q(\mathbf{x})\le 0 \quad \textit{where} \quad q\in I[1,Q]\\&\mathbf{x}>0,\\&z_i(\mathbf{x})=\sum_{j=1}^{n}c_{ij}x_j,\ z_i(\mathbf{x})=E[z_i(\mathbf{x})]\end{aligned} \tag{5.3}$$

where p is the number of objectives to optimise, q is the number of constraints, c_{ij} are the coefficients' expected values and n is the number of decision variables.

Step 2
Vectors $\mathbf{U}_1$ and $\mathbf{M}$ are defined as having the maximum and minimum values of the objective functions respectively. This means that vector $\mathbf{U}_1$ has the values of the maximisation of each objective separately (e.g. max $z_1(\mathbf{x})$, max $z_2(\mathbf{x})$, max $z_3(\mathbf{x})$, max $z_4(\mathbf{x})$, max $z_5(\mathbf{x})$), subject to constraints $g_q(\mathbf{x}) \leq 0$. In addition vector $\mathbf{M}$ has the minimum values found following the same procedure, in other words, minimising each objective separately subject to the constraints.

$$z_i(\mathbf{x}_i^*) = \max \quad z_i(\mathbf{x}), \qquad i \in I[1, p]$$

$$\mathbf{U}_1 = \begin{bmatrix} z_1(\mathbf{x}_1^*) \\ z_2(\mathbf{x}_2^*) \\ \vdots \\ z_p(\mathbf{x}_q^*) \end{bmatrix} \qquad (5.4)$$

$$\mathbf{M} = \begin{bmatrix} z_{1\min} \\ z_{2\min} \\ \vdots \\ z_{3\min} \end{bmatrix}$$

It is important to bear in mind that $\mathbf{M}$ may no exist in practice. Therefore, in such cases some other techniques have to be applied.

Step 3
An initial surrogate function is formulated:

$$F(\mathbf{x}) = \sum_{i=1}^{p} G_i(\mathbf{x}) \qquad (5.5)$$

where

$$G_i(\mathbf{x}) = \frac{z_i(\mathbf{x}) - z_{i\min}}{z_i(\mathbf{x}_i^*) - z_{i\min}} \qquad (5.6)$$

where $z_i(x)$ is the value of objective function i (i = 1,2,..,n); z_{imin} is the minimum value obtained when objective i is subjected to the constraints; and $z_i(x^*)$ is the maximum value obtained when objective i is subjected to the constraints. Hitherto, each objective has been normalised using equation 5.6 and the surrogate function $F(\mathbf{x})$ has been defined as the addition of the normalised objectives. What is more, each objective can be maximised or minimised. In $F(\mathbf{x})$ there will be given a + sign for maximisation and a - sign for minimisation.

Step 4
An initial solution $\mathbf{x}_1$ is obtained maximising $F(\mathbf{x})$, subject to the constraints $g_q(\mathbf{x}) \leq 0$. This solution is used to generate a goal vector $\mathbf{G}_1$:

$$\mathbf{G}_1 = \begin{bmatrix} G_1(\mathbf{x}_1) \\ G_2(\mathbf{x}_1) \\ \vdots \\ G_q(\mathbf{x}_1) \end{bmatrix} \qquad (5.7)$$

Step 5
A multidimensional utility function is defined and in this case Goicoechea et al. (1979) proposed a multiplicative form (see equation 5.2):

$$U(\mathbf{G}) = \frac{\prod_{i=1}^{p}[1 + k\,k_i U_i(G_i)] - 1}{k} \tag{5.8}$$

this function is used to reflect the DM's goal utility assessment, where k and k_i are constants which are determined by questions posed to the DM.

Step 6
A new surrogate objective function is defined:

$$S_1(\mathbf{x}) = \sum_{i=1}^{p} w_i G_i(\mathbf{x}) \tag{5.9}$$

where

$$w_i = 1 + \frac{r}{G_i(\mathbf{x}_1)} \frac{\partial u(\mathbf{G})}{\partial G_i}\bigg|_{G_i} \tag{5.10}$$

w_i are the weights that result from considering the DM's preferences in the solution of the problem and r is a scaling factor.

Step 7
An alternative solution is generated maximising the surrogate objective function S_1. The solution found $\mathbf{x}_2$ is used to generate vectors $\mathbf{G}_2$ and $\mathbf{U}_2$:

$$\mathbf{G}_2 = \begin{bmatrix} G_1(\mathbf{x}_2) \\ G_2(\mathbf{x}_2) \\ \vdots \\ G_p(\mathbf{x}_2) \end{bmatrix} \quad \mathbf{U}_2 = \begin{bmatrix} z_1(\mathbf{x}_2) \\ z_2(\mathbf{x}_2) \\ \vdots \\ z_p(\mathbf{x}_2) \end{bmatrix} \tag{5.11}$$

where $\mathbf{G}_2$ is a new goal vector and $\mathbf{U}_2$ contains the values of the maximisation of the surrogate function S_1.

Step 8
A vector $\mathbf{V}_1$ that expresses the trade-off between goal value and its probability of achievement is generated:

$$\mathbf{V}_1 = \begin{bmatrix} (G_1(\mathbf{x}_2), & 1-\alpha_1) \\ (G_2(\mathbf{x}_2), & 1-\alpha_2) \\ & \vdots \\ (G_p(\mathbf{x}_2), & 1-\alpha_p) \end{bmatrix} \tag{5.12}$$

where $1 - \alpha_i$ is such that,

$$prob[z_i(\mathbf{x}) \geq z_i(\mathbf{x}_2)] \geq 1 - \alpha_i \tag{5.13}$$

This step is very important because it allows the DM to define a probability of achievement for each goal.

Step9
The DM has to answer the following question: "Are all the $z_i(\mathbf{x}_2)$ values satisfactory?" (Goicoechea et al., 1982). If the answer is affirmative the vector $\mathbf{U}_2$ is a solution and if not go to step 10.

Step 10
The $z_k(\mathbf{x})$ with the least satisfactory pair of $(G_k(\mathbf{x}_2), 1\text{-}\alpha_k)$ is selected and the DM specifies a new probability for that pair:
$\varepsilon_k \in \boldsymbol{R}^+, 0 < \alpha_k^0 < 1$

$$prob[\underline{z}_k(\mathbf{x}) \geq \varepsilon_k] \geq 1 - \alpha_k^0 \tag{5.14}$$

Step 11
The solution space is redefined creating a new x-space:

$$g_q(\mathbf{x}) \leq 0 \quad q \in I[1, Q]$$

and a new constraint defined as follows:

$$\sum_{j=1}^{n} E(\underline{c}_{kj})\, x_j + K_{\alpha_k^0}[\mathbf{x}^t \mathbf{A} \mathbf{x}]^{1/2} \geq \varepsilon_k \tag{5.15}$$
$$\mathbf{x} > 0$$

where c_{kj} represent the coefficients' expected values and **A** is the variance-covariance matrix of the coefficients' expected values.

Step 12
A new surrogate objective function is generated and a sequential search for a satisfactory solution is performed going back to step 7 or step 6 as many times as necessary.

$$S_2(\mathbf{x}) = \sum_{i \neq k}^{p} w_i G_i(\mathbf{x}) \tag{5.16}$$

The sequence is repeated until a satisfactory vector $\mathbf{V}_2$ is found:

$$\mathbf{V}_2 = \begin{bmatrix} (\varepsilon_1, & 1-\alpha_1^0) \\ (\varepsilon_2, & 1-\alpha_2^0) \\ \vdots & \\ (\varepsilon_p, & 1-\alpha_p^0) \end{bmatrix} \tag{5.17}$$

5.3 Case Study

The Black Mesa Region problem was presented by Goicoechea et al. (1979). In Northern Arizona on the Navajo Nation lands there is an area of 5,700 hectares (ha) that will be strip-mined for coal in a 30-year period. The area has been used as rangeland and this activity has caused heavy overgrazing. This resulted in the development of a programme for designing and implementation of multiple land uses. This development programme can then be given to a management agency. The 30-year period is the time horizon to solve this problem although it is necessary to divide the analysis in sub-periods due to the variability of the water constraint. In other words, because the water will not remain constant over the 30-year period, it will be divided into 15 sub-periods each one 2 years long. This means that in each sub-period the total area to be strip-mined is 380 ha.
Five objectives are considered: 1. livestock production, 2. augmentation of water run-off, 3. farming of selected crops, 4. control of sedimentation rates, and 5. fish pond harvesting.

It is desired to maximise objectives 1,2,3 and 5 while objective 4 has to be minimised. The decision variables considered are twelve and are expressed in hectares of mined land:

1. No reclamation program by current management (x_1)
2. Contour furrowing livestock production good range conditions (x_2)
3. Contour furrowing livestock production poor range conditions (x_3)
4. Run-off augmentation compacted earth treatment (x_4)
5. Run-off augmentation compacting and salt treatment (x_5)
6. Run-off augmentation plastic cover and gravel (x_6)
7. Wheat production (x_7)
8. Corn production (x_8)
9. Alfalfa production (x_9)
10. Barley production (x_{10})
11. Sorghum production (x_{11})
12. Fish production pond base (x_{12})

The land allocation for a 2-year sub-period is represented in Figure 5.1.

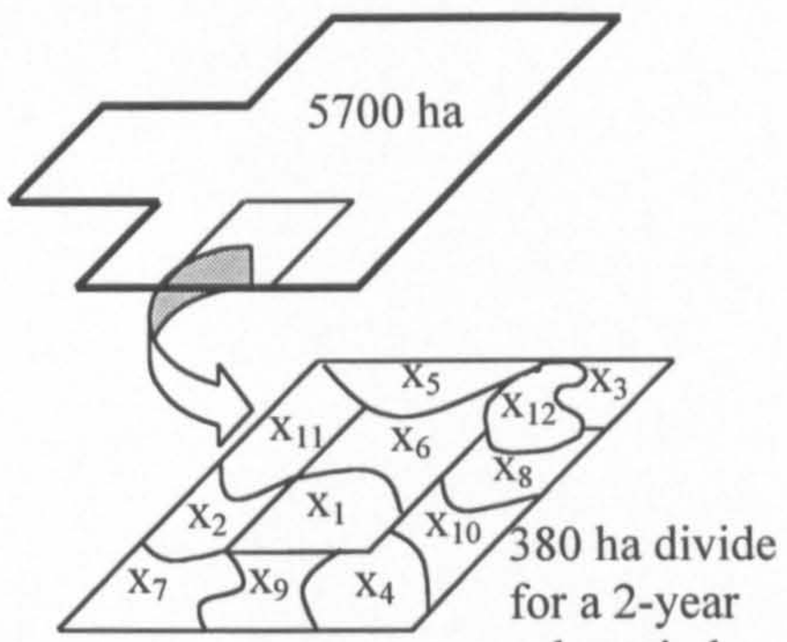


Figure 5.1 Land allocation for a 2-year sub-period
Adapted from Goicoechea et al. (1982)

The objectives are defined as follows:

Objective 1 Livestock production

$$f_1(\mathbf{x}) = \sum_{i=1}^{12} l_i x_i \qquad \text{animal units} \qquad (5.18)$$

where l_i is the number of livestock heads in animal units month per hectare of land (AUM/ha), and i is the number of decision variable applied.

Objective 2 Water run-off

$$f_2(\mathbf{x}) = \sum_{i=1}^{12} r_i x_i \qquad \text{m}^3 \qquad (5.19)$$

where r_i is the water run-off yield in cubic meters per hectare (m^3/ha), and i is the number of the decision variable applied.

Objective 3 Selected crops

$$f_3(\mathbf{x}) = \sum_{i=1}^{12} c_i x_i \qquad \text{kg} \qquad (5.20)$$

where c_i is the crop yield in kilograms per hectare (kg/ha), and i is the number of the decision variable applied.

Objective 4 Sediment

$$f_4(\mathbf{x}) = \sum_{i=1}^{12} s_i x_i \qquad \text{m}^3 \tag{5.21}$$

where s_i is the sediment yield in cubic meters per hectare (m^3/ha), and i is the number of the decision variable applied.

Objective 5 Fish yield

$$f_5(\mathbf{x}) = \sum_{i=1}^{12} p_i x_i \qquad \text{kg} \tag{5.22}$$

where p_i is the fish yield in kilograms per hectare (kg/ha), and i is the number of the decision variable applied.

There are three constraints to be considered:

Constraint 1 Land

$$\sum_{i=1}^{12} x_i = b_i \qquad \text{ha} \tag{5.23}$$

where b_i is the area to be strip-mined in a 2-year sub-period . If the total area to be strip-mined in a 30-year period is 5,700 ha, then $b_i = \frac{5700}{15} = 380$ hectares every two years.

Constraint 2 Capital

$$\sum_{i=1}^{12} q_i x_i = b_q \qquad \$ \tag{5.24}$$

where q_i is the cost of implementing the ith decision variable, and b_q is \$200,000. This is an estimated value and was modified from the original problem (\$35,000) (Goicoechea et al., 1979).

Constraint 3 Water

$$\sum_{i=1}^{12} w_i x_i = b_w \qquad \text{m}^3 \tag{5.25}$$

where w_i is the water consumption of the ith decision variable, and b_w is the water available for a 2-year subperiod through run-off practices and rainfall. Therefore, this value is a random variable since rainfall is unpredictable.The values of parameters l_i, r_i, c_i, s_i, p_i, q_i,and w_i can be found in Appendix B of this thesis.

5.4 Risk and Uncertainty Analysis and Decision-Maker's Preferences

As explained in Chapter Three, it is understood that risk and uncertainty are different. The main distinction between them consists in that in the risk models the outcomes can be described using a probability distribution whereas in uncertainty analysis this is not possible. Therefore, the following subsections are divided into risk analysis and uncertainty analysis.

5.4.1 Risk Analysis

In the no-risk model the parameter values for equation 5.18 to 5.22 are the expected values defined by Goicoechea et al. (1982) (Appendix B).
To introduce risk on the decision variables it is necessary to define a normal distribution as follows:

$$NORMDIST = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2} \tag{5.26}$$

where μ is the mean, σ is the standard deviation, and $-\infty < x < \infty$.

The standard deviation values used to calculate the normal distribution are those found in Appendix B. The mean value μ is equal to zero considering that the maximum value of the distribution is one, this normal distribution is multiplied by the expected values of each parameter.

5.4.2 Uncertainty Analysis

Understanding the water constraint as a random variable, since rainfall is unpredictable, in order to generate values for the water available for each 2-year sub-period, it is necessary to run a Monte Carlo simulation. This simulation is carried out for fifteen times and then these values are used as the values for b_w for the fifteen 2-year sub-periods. The Monte Carlo simulation was performed using @Risk setting the number of iterations to 1000. The values obtained are in Table 5.1:

No. of 2-year sub-period	b_w (m^3) (water for sub-period)
1	2,702,473
2	1,721,549
3	1,047,110
4	338,251
5	475,079
6	803,335
7	2,512,024
8	845,213
9	1,944,114
10	705,202
11	1,425,694
12	409,607
13	2,175,253
14	1,576,443
15	1,300,811

Table 5.1 Results of the Monte Carlo simulation run to obtain the water constraint values

Since the nature of the water constraint is unpredictable, uncertainty will be present in the throughout the development of the problem and its solution. It is clear that the values for the water constraint can vary every time the Monte Carlo simulation is run. For the purposes of this research the values considered are those of Table 5.1.

5.4.3 Decision-Maker's Preferences Model

The model for the DM's preferences was taken from Goicoechea et al. (1982). It consists of a set of pair-wise comparisons made by the DM to determine the importance of each goal compared to the others. The result of this procedure is as follows:

G_3, crops $\succ G_1$, livestock $\succ G_4$, sediment $\succ G_2$, run-off $\succ G_5$, fish
This means that G_3 is preferred to G_1, G_1 is preferred to G_4, G_4 is preferred to G_2, and G_2 is preferred to G_5.

Considering this goal hierarchy, the following step is to define the individual utility functions with a form $u_i(G_i) = c(1.0 - e^{b \cdot G_i})$. The individual utility functions are as follows:

$$u_1(G_1) = 1.788(1 - e^{-0.82G_1}) \quad (5.27)$$
$$u_2(G_2) = 0.819(e^{0.805G_2} - 1) \quad (5.28)$$
$$u_3(G_3) = 1.199(1 - e^{-1.8G_3}) \quad (5.29)$$
$$u_4(G_4) = G_4 \quad (5.30)$$
$$u_5(G_5) = 0.431(e^{1.21G_5} - 1) \quad (5.31)$$

Using equation 5.27 to 5.31 Goicoechea et al. (1982) solved for k and k_i yielding the following values:

$k_1 = 0.260$
$k_2 = 0.201$
$k_3 = 0.519$
$k_4 = 0.223$
$k_5 = 0.081$
$k = -0.534$

These values are used to integrate the DM's preferences with the GA.

5.5 Real-Valued GA

5.5.1 Real-Valued GA

A general GA structure will be described in this section. This GA structure will be applied every time that a GA is needed or run.
The initial population consists of i decision variables x_i. These are randomly generated and are expressed in real-valued vectors. The decision variables are in terms of land (hectares) and they can take values between 0 and 380 (the reason for these values is explained in further sections).
In the general GA model, the size of the population is 80 and the algorithm is run for 200 cycles performing the crossover and mutation operators 80 times for each cycle. It is important to have in mind that the normal structure of a GA (presented in Chapter Two) considers basically three operators: selection, crossover, and mutation.
The selection method used is tournament selection; initially the tournament size was two (binary tournament, Bäck et al. (2000)) but it was found that with a tournament size of three the results were more accurate. The crossover used is "arithmetic crossover" (Michalewicz, 1994) defined as follows: having two parents p_1 and p_2, the offspring ch, and a number α within [0,1]

$$ch_i = \alpha\, p_{1i} + (1 - \alpha)\, p_{2i} \quad (5.32)$$

where $i = 1, \ldots, n$

The mutation operator selected is that proposed by Michalewicz (1994) where the new child is a random value generated from a domain, in this case [0,380].

The crossover and mutation probabilities are 0.5 and 0.15 respectively. In other words, it is expected that 40 chromosomes out of 80 undergo crossover and the 15% of chromosomes undergo mutation. This mutation probability was selected in order to maintain diversity.
Once the GA is defined, the next step to follow is the definition of the four models to solve. The description of these models is presented in section 5.6.

5.6 PROTRADE Approach: Models Definition and Results

The objective functions are calculated by using equations 5.18 to 5.22, and the expected value of the coefficients found in Appendix B of this thesis. To be consistent with the problem definition of PROTRADE (Step 1 of the method) a vector of objective functions is defined as:

$$z(\mathbf{x}) = [z_1(\mathbf{x}), z_2(\mathbf{x}), z_3(\mathbf{x}), z_4(\mathbf{x}), z_5(\mathbf{x})] \tag{5.33}$$

where objective functions 1,2,3, and 5 are to be maximised and objective function 4 is to be minimised. Then the objective functions are defined as follows:

$$\begin{aligned} z_1(\mathbf{x}) &= f_1(\mathbf{x}) \\ z_2(\mathbf{x}) &= f_2(\mathbf{x}) \\ z_3(\mathbf{x}) &= f_3(\mathbf{x}) \\ z_4(\mathbf{x}) &= -f_4(\mathbf{x}) \\ z_5(\mathbf{x}) &= f_5(\mathbf{x}) \end{aligned} \tag{5.34}$$

Additionally the constraints are calculated by using equations 5.23 to 5.25, and using the coefficients' expected values found in Appendix B of this work. Therefore the three constraints are defined as follows:

Land constraint

$$x_1 + x_2 + \cdots + x_{12} \leq 380 \quad \text{ha} \tag{5.35}$$

Considering that 380 ha is the total area to be allocated in each 2-year sub-period.

Capital constraint

$$q_1 x_1 + q_2 x_2 + \cdots + q_{12} x_{12} \leq 200{,}000 \qquad \$ \tag{5.36}$$

Water constraint

$$w_1 x_1 + w_2 x_2 + \cdots + w_{12} x_{12} \leq b_w^* \qquad \text{m}^3 \tag{5.37}$$

* values in Table 5.1.

The next step (step 2) to follow is the optimisation of each objective separately in order to obtain vectors $\mathbf{U}_1$ and $\mathbf{M}$. Five GAs were programmed with the fitness functions represented in equations 5.38 to 5.42:

GA 1 $$F(\mathbf{x}) = z_1(\mathbf{x}) = l_2 x_2 + l_3 x_3 \tag{5.38}$$

GA 2 $$F(\mathbf{x}) = z_2(\mathbf{x}) = r_1 x_1 + r_2 x_2 + r_3 x_3 + r_4 x_4 + r_5 x_5 + r_6 x_6 \tag{5.39}$$

GA 3 $$F(\mathbf{x}) = z_3(\mathbf{x}) = c_7 x_7 + c_8 x_8 + c_9 x_9 + c_{10} x_{10} + c_{11} x_{11} \tag{5.40}$$

GA 4

$$F(\mathbf{x}) = z_4(\mathbf{x}) = s_1 x_1 + s_2 x_2 + s_3 x_3 + s_4 x_4 + s_5 x_5 + s_7 x_7 + s_8 x_8 + s_9 x_9 + s_{10} x_{10} + s_{11} x_{11} \tag{5.41}$$

GA 5 $$F(\mathbf{x}) = z_5(\mathbf{x}) = p_{12}\, x_{12} \tag{5.42}$$

Once the fitness functions are defined, each GA will be run for two different optimisation cases, maximisation and minimisation subject to land, capital, and water constraints. The results found will be used for the goals definition of each surrogate objective function. Thus vector $\mathbf{U}_1$ contains the results of the maximisation and vector $\mathbf{M}$ contains the results of the minimisation:

$$\mathbf{U}_1 = \begin{bmatrix} 39.45 \\ 720{,}558.21 \\ 1{,}808{,}033.08 \\ -9.35 \\ 70{,}552.44 \end{bmatrix} \qquad \mathbf{M} = \begin{bmatrix} 0.00 \\ 118.9756 \\ 72.106 \\ -6050.7 \\ 0.00 \end{bmatrix} \tag{5.43}$$

The values of vectors $\mathbf{U}_1$ and $\mathbf{M}$ are used to calculate the goals defined in equation 5.6. This equation is very important because the units of the objectives are of different dimensions, and therefore, they cannot be added directly. They have to be normalised to make them dimensionless quantities. For this reason each objective normalised will be called a goal.
It is important to bear in mind that the GAs used are real-valued and followed the structure defined in section 5.5.

Comparing this method with the SEMOPS method (Chapter Four) it is possible to see that both normalise their objectives. The main differences between them is that SEMOPS normalises the objectives using the aspiration levels defined by the DM while PROTRADE uses the values resultant from the optimisation of each objective individually.

Four models to handle the decision variables are developed: one that considers neither risk nor probability of achievement, one that does not consider risk but does consider probability of achievement, one that considers risk but does not consider probability of achievement, and a final one that considers both risk and probability of achievement. These models are discussed in the following sub-sections.

5.6.1 No-Risk No-Probability of Achievement Model

Once steps 1 and 2 are performed it is possible to proceed with the no-risk no probability of achievement model definition. As mentioned before, the coefficients of the objective function are the expected values found in Appendix B. Because this is a model that does not consider risk these coefficients will not change their values in the whole model's solution. The model is performed 15 times because each time represents a 2-year sub-period. That will make the total horizon time of 30 years.

The no-risk no-probability of achievement model proposed in this thesis is outlined in Figure 5.2. This model is used to find the results presented in this sub-section.

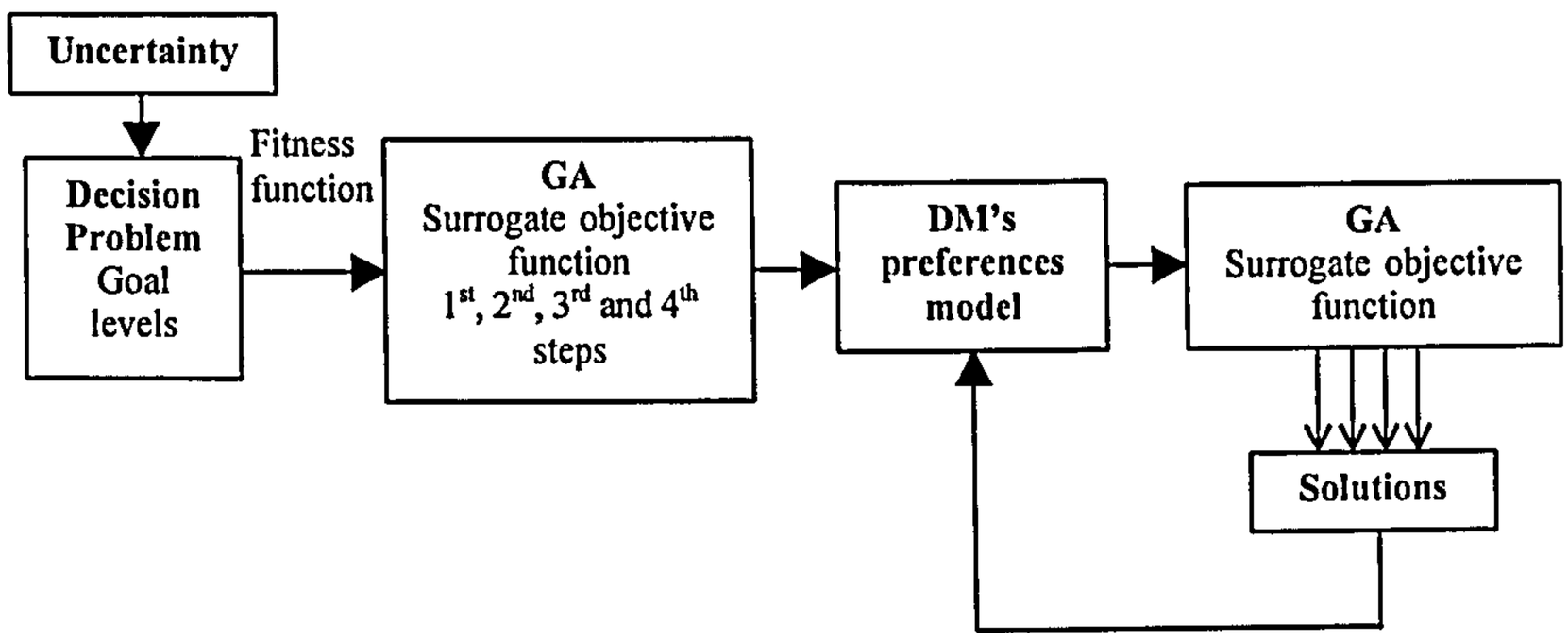


Figure 5.2 No-risk no-probability of achievement model

First sub-period

For this sub-period the value of b_w is 2,702,473 m^3. This means that the water constraint is defined as follows:

$$w_1 x_1 + w_2 x_2 + \cdots + w_{12} x_{12} \leq 2{,}702{,}473\ m^3 \tag{5.44}$$

Following Step 3 of PROTRADE the initial surrogate function is:

$$F(\mathbf{x}) = G_1(\mathbf{x}) + G_2(\mathbf{x}) + G_3(\mathbf{x}) + G_4(\mathbf{x}) + G_5(\mathbf{x}) \tag{5.45}$$

It is important to bear in mind that the surrogate function is dimensionless and that it will act as the GA's fitness function. The minus sign assigned to objective 4 is considered directly in the equation to calculate G_4.

Running the GA to maximise $F(\mathbf{x})$ using equation 5.45 subject to land, capital, and water constraints (Step 4), yields vector $\mathbf{x}_1 = [x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8, x_9, x_{10}, x_{11}, x_{12}]$ (ha):

$\mathbf{x}_1$	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_{10}	x_{11}	x_{12}
	194.11	2.67	0.47	3.93	3.23	51.20	1.95	1.69	89.22	10.27	2.59	18.16

Vector $\mathbf{x}_1$ contains the attempt land allocation for each decision variable.

Additionally, with the GA the values for the goal vector $\mathbf{G}_1$ are found:

$\mathbf{G}_1$	G_1	G_2	G_3	G_4	G_5
	0.009632	0.268126	0.391161	0.360843	0.962670

The goal values are dimensionless and are calculated using equation 5.6.

At this point the step 5 of the method is performed considering the DM's preferences using the ranking proposed by Goicoechea et al. (1982) and presented in sub-section 5.4.3 together with the k and k_i values.

Next the surrogate function has to be redefined, as is shown in equations 5.9 and 5.10 of the PROTRADE method (Step 6), using the results from Step 4 and 5.

a) Calculate $u(\mathbf{G}_1)$

$u_1(G_1) = u_1(0.009632) = 0.014066$
$u_2(G_2) = u_2(0.268126) = 0.197302$
$u_3(G_3) = u_3(0.391161) = 0.606024$
$u_4(G_4) = u_4(0.360843) = 0.360843$
$u_5(G_5) = u_5(0.962670) = 0.950519$

$u(\mathbf{G}_1) = 0.475801$

b) The DM decides on an incremental utility $\Delta u(\mathbf{G})$ between 0 and 1. The value decided is $\Delta u(\mathbf{G}) = 0.20$.

$$u(\mathbf{G}_1 + r \cdot \nabla u(\mathbf{G}_1)) = u(\mathbf{G}_1) + \Delta u(\mathbf{G}) = 0.475801 + 0.20 = 0.675801$$

$$\mathbf{G}_1 + r \cdot \nabla u(\mathbf{G}_1) = \begin{bmatrix} 0.009632 \\ 0.268126 \\ 0.391161 \\ 0.360843 \\ 0.962670 \end{bmatrix} + r \cdot \begin{bmatrix} 0.282662 \\ 0.125315 \\ 0.49662 \\ 0.173809 \\ 0.10533 \end{bmatrix}$$

Solving the equation $r = 0.6562$. Then the weights w_i are calculated using equation 5.10:

$w_1 = 20.25692$; $w_2 = 1.306691$; $w_3 = 1.833115$; $w_4 = 1.316075$; $w_5 = 1.071798$

Finally, the new surrogate function $S_1(\mathbf{x})$ is determined and the GA is run maximising it. The fitness function of the GA will be $S_1(\mathbf{x})$ and is defined as follows:

$$\max S_1(\mathbf{x}) = 20.2569\,G_1 + 1.30669\,G_2 + 1.833115\,G_3 + 1.316075\,G_4 + 1.071798\,G_5 \quad (5.46)$$

The optimal solution after running the GA is vector $\mathbf{x}_2 = [x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8, x_9, x_{10}, x_{11}, x_{12}]$ (ha):

$\mathbf{x}_2$	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_{10}	x_{11}	x_{12}
	0.14	234.36	2.78	2.65	1.06	4.75	1.62	1.23	110	3	1.41	16.16

Vector $\mathbf{x}_2$ contains the land allocation for each decision variable considering the DM's preferences.

Additionally, with the GA the values for the goal vector $\mathbf{G}_2$ are found:

$\mathbf{G}_2$	G_1	G_2	G_3	G_4	G_5
	0.819518	0.051039	0.460758	0.994283	0.856954

In this model it is considered that the DM is satisfied with the results obtained and hence, $\mathbf{x}_2$ is the definitive or final land allocation for the first 2-year sub-period.

Second sub-period

The procedure followed in the first sub-period is applied again. For this sub-period the value of b_w is 1,721,549 m^3. This means that the water constraint is defined as follows:

$$w_1 x_1 + w_2 x_2 + \cdots + w_{12} x_{12} \leq 1{,}721{,}549 \text{ m}^3 \quad (5.47)$$

Following Step 3 of PROTRADE the initial surrogate function is:

$$F(\mathbf{x}) = G_1(\mathbf{x}) + G_2(\mathbf{x}) + G_3(\mathbf{x}) + G_4(\mathbf{x}) + G_5(\mathbf{x}) \quad (5.48)$$

It is important to bear in mind that the surrogate function is dimensionless and that it will act as the GA's fitness function.

Running the GA to maximise $F(\mathbf{x})$ using equation 5.45 subject to land, capital, and water constraints (Step 4), yields vector $\mathbf{x}_1 = [x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8, x_9, x_{10}, x_{11}, x_{12}]$ (ha):

$\mathbf{x}_1$	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_{10}	x_{11}	x_{12}
	203.10	4.28	0.90	1.35	2.16	73.98	1.84	1.48	68.60	1.65	1.98	18.62

Vector $\mathbf{x}_1$ contains the attempt land allocation for each decision variable.

Additionally, with the GA the values for the goal vector $\mathbf{G}_1$ are found:

$\mathbf{G}_1$	G_1	G_2	G_3	G_4	G_5
	0.015716	0.330685	0.290479	0.332118	0.987189

The goal values are dimensionless and are calculated using equation 5.6.

At this point the step 5 of the method is performed considering the DM's preferences using the ranking proposed by Goicoechea et al. (1982) and presented in sub-section 5.4.3 together with the k and k_i values.

Next the surrogate function has to be redefined, as is shown in equations 5.9 and 5.10 of the PROTRADE method (Step 6), using the results from Step 4 and 5.

a) Calculate $u(\mathbf{G}_1)$

$$u_1(G_1) = u_1(0.0157) = 0.02287$$
$$u_2(G_2) = u_2(0.33068) = 0.249789$$
$$u_3(G_3) = u_3(0.29047) = 0.488196$$
$$u_4(G_4) = u_4(0.3321) = 0.3321$$
$$u_5(G_5) = u_5(1.06248) = 0.99212$$

$$u(\mathbf{G}_1) = 0.428674$$

b) The DM decides on an incremental utility $\Delta u(\mathbf{G})$ between 0 and 1. The value decided is $\Delta u(\mathbf{G}) = 0.20$.

$$u(\mathbf{G}_1 + r \cdot \nabla u(\mathbf{G}_1)) = u(\mathbf{G}_1) + \Delta u(\mathbf{G}) = 0.437518 + 0.20 = 0.628674$$

$$\mathbf{G}_1 + r \cdot \nabla u(\mathbf{G}_1) = \begin{bmatrix} 0.0157 \\ 0.33068 \\ 0.29047 \\ 0.3321 \\ 0.987189 \end{bmatrix} + r \cdot \begin{bmatrix} 0.291102 \\ 0.137023 \\ 0.592139 \\ 0.179033 \\ 0.112374 \end{bmatrix}$$

Solving the equation $r = 0.4994$. Then the weights w_i are calculated:

$w_1 = 10.2502$; $w_2 = 1.206931$; $w_3 = 2.018023$; $w_4 = 1.269209$; $w_5 = 1.056848$

Finally, the new surrogate function $S_1(\mathbf{x})$ is defined and the GA is run maximising it. The fitness function of the GA will be $S_1(\mathbf{x})$.

$$\max S_1(\mathbf{x}) = 10.2502\,G_1 + 1.206931\,G_2 + 2.01802\,G_3 + 1.269209\,G_4 + 1.056848\,G_5 \quad (5.49)$$

The optimal solution after running the GA is vector $\mathbf{x}_2 = [x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8, x_9, x_{10}, x_{11}, x_{12}]$ (ha):

$\mathbf{x}_2$	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_{10}	x_{11}	x_{12}
	0.48	236.79	0.54	2.72	2.69	19.47	2.98	1.45	83.31	9.96	1.81	16.43

Vector $\mathbf{x}_2$ contains the attempt land allocation for each decision variable considering the DM's preferences.

Additionally, with the GA the values for the goal vector $\mathbf{G}_2$ are found:

$\mathbf{G}_2$	G_1	G_2	G_3	G_4	G_5
	0.825856	0.095065	0.366847	0.993298	0.871379

In this model it is considered that the DM is satisfied with the results obtained and hence, $\mathbf{x}_2$ is the definitive or final land allocation for the second 2-year sub-period.

Third sub-period

The procedure followed in the first and second sub-periods is applied again. Obtaining the following results:
The water constraint is defined as follows:

$$w_1 x_1 + w_2 x_2 + \cdots + w_{12} x_{12} \le 1{,}047{,}110 \qquad \text{m}^3 \qquad (5.50)$$

Vector $\mathbf{x}_1$ is found by running the GA to maximise $F(\mathbf{x})$ subject to land, capital, and water constraints (Step 4) is:

$\mathbf{x}_1$	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_{10}	x_{11}	x_{12}
	216.54	0.27	0	2.44	6.03	76.34	2.75	0.91	51.82	2.42	1.08	18.89

Additionally, with the GA the values for the goal vector $\mathbf{G}_1$ are found:

$\mathbf{G}_1$	G_1	G_2	G_3	G_4	G_5
	0.001014	0.353566	0.223019	0.288196	1.00

The values of w_i are calculated to redefine the surrogate function $S_1(\mathbf{x})$ using $u(\mathbf{G}_1)$, and r:

$u(\mathbf{G}_1) = 0.378438$
$r = 0.4013$

$w_1 = 121.3013$; $w_2 = 1.164284$; $w_3 = 2.209341$; $w_4 = 1.256571$; $w_5 = 1.047441$

Finally, the new surrogate function $S_1(\mathbf{x})$ is defined and the GA is run maximising it:

$$\max S_1(\mathbf{x}) = 121.30\,G_1 + 1.16428\,G_2 + 2.20934\,G_3 + 1.25657\,G_4 + 1.047441\,G_5 \quad (5.51)$$

The optimal solution after running the GA is vector $\mathbf{x}_2$:

$\mathbf{x}_2$	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_{10}	x_{11}	x_{12}
	0.1	269.73	26.01	0.19	0.67	4.35	1.08	0.81	51.57	7.62	0.58	17.27

Additionally, with the GA the values for the goal vector $\mathbf{G}_2$ are found:

$\mathbf{G}_2$	G_1	G_2	G_3	G_4	G_5
	0.964259	0.053134	0.227507	0.994321	0.915582

In this model it is considered that the DM is satisfied with the results obtained and hence, $\mathbf{x}_2$ is the definitive or final land allocation for the third 2-year sub-period.

Fourth sub-period

The water constraint is defined as follows:

$$w_1 x_1 + w_2 x_2 + \cdots + w_{12} x_{12} \leq 338{,}251\ \mathrm{m}^3 \quad (5.52)$$

Vector $\mathbf{x}_1$ is found by running the GA to maximise $F(\mathbf{x})$ subject to land, capital, and water constraints (Step 4) is:

$\mathbf{x}_1$	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_{10}	x_{11}	x_{12}
	0.2	155.87	4.11	8.34	34.81	134.83	2.51	0.84	13.77	6.99	0.73	16.73

The goal vector $\mathbf{G}_1$ is:

$\mathbf{G}_1$	G_1	G_2	G_3	G_4	G_5
	0.547022	0.471849	0.074466	0.996016	0.887202

The values of w_i are calculated to redefine the surrogate function $S_1(\mathbf{x})$ using $u(\mathbf{G}_1)$, and r:

$u(\mathbf{G}_1) = 0.547341$
$r = 0.4079$

$w_1 = 1.14112$; $w_2 = 1.123554$; $w_3 = 4.96274$; $w_4 = 1.07333$; $w_5 = 1.042265$

Finally, the new surrogate function $S_1(\mathbf{x})$ is defined and the GA is run maximising it:

$$\max S_1(\mathbf{x}) = 1.14112\,G_1 + 1.123554\,G_2 + 4.96274\,G_3 + 1.07333\,G_4 + 1.042265\,G_5 \quad (5.53)$$

The optimal solution after running the GA is vector $\mathbf{x}_2$:

$\mathbf{x}_2$	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_{10}	x_{11}	x_{12}
	0	168.48	0.54	6.13	15.6	132.75	8.94	1.47	6.83	20.89	0.88	16.44

Additionally, with the GA the values for the goal vector $\mathbf{G}_2$ are found:

$\mathbf{G}_2$	G_1	G_2	G_3	G_4	G_5
	0.587833	0.426707	0.081980	0.996473	0.871952

In this model it is considered that the DM is satisfied with the results obtained and hence, $\mathbf{x}_2$ is the definitive or final land allocation for the fourth 2-year sub-period.

Fifth sub-period

The water constraint is defined as follows:

$$w_1 x_1 + w_2 x_2 + \cdots + w_{12} x_{12} \leq 475{,}079 \text{ m}^3 \quad (5.54)$$

Vector $\mathbf{x}_1$ is found by running the GA to maximise $F(\mathbf{x})$ subject to land, capital, and water constraints (Step 4) is:

$\mathbf{x}_1$	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_{10}	x_{11}	x_{12}
	0.11	197.92	5.59	5.3	43.15	82.61	1.62	0	23.22	1.68	0.45	17.24

The goal vector $\mathbf{G}_1$ is:

$\mathbf{G}_1$	G_1	G_2	G_3	G_4	G_5
	0.695057	0.346329	0.101192	0.995509	0.914268

The values of w_i are calculated to redefine the surrogate function $S_1(\mathbf{x})$ using $u(\mathbf{G}_1)$, and r:

$u(\mathbf{G}_1) = 0.0.57852$

$r = 0.4688$

$w_1 = 1.112637$; $w_2 = 1.168589$; $w_3 = 4.164026$; $w_4 = 1.082332$; $w_5 = 1.047785$

Finally, the new surrogate function $S_1(\mathbf{x})$ is defined and the GA is run maximising it:

$$\max S_1(\mathbf{x}) = 1.112637\,G_1 + 1.168589\,G_2 + 4.164026\,G_3 + 1.08233\,G_4 + 1.04778\,G_5 \quad (5.55)$$

The optimal solution after running the GA is vector $\mathbf{x}_2$:

$\mathbf{x}_2$	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_{10}	x_{11}	x_{12}
	0.27	143.91	0.81	10.22	13.76	151.15	5.83	1	15.32	18.66	1.58	16.13

Additionally, with the GA the values for the goal vector $\mathbf{G}_2$ are found:

$\mathbf{G}_2$	G_1	G_2	G_3	G_4	G_5
	0.502408	0.476134	0.108184	0.995943	0.855520

In this model it is considered that the DM is satisfied with the results obtained and hence, $\mathbf{x}_2$ is the definitive or final land allocation for the fifth 2-year sub-period.

Sixth sub-period

The water constraint is defined as follows:

$$w_1 x_1 + w_2 x_2 + \cdots + w_{12} x_{12} \leq 803{,}335 \text{ m}^3 \quad (5.56)$$

Vector $\mathbf{x}_1$ is found by running the GA to maximise $F(\mathbf{x})$ subject to land, capital, and water constraints (Step 4) is:

$\mathbf{x}_1$	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_{10}	x_{11}	x_{12}
	162.13	2.98	0.3	3.25	10.42	138.81	0.27	0	39.15	2.73	1.51	18.07

The goal vector $\mathbf{G}_1$ is:

$\mathbf{G}_1$	G_1	G_2	G_3	G_4	G_5
	0.010646	0.503038	0.167411	0.467187	0.958021

The values of w_i are calculated to redefine the surrogate function $S_1(\mathbf{x})$ using $u(\mathbf{G}_1)$, and r:

$u(\mathbf{G}_1) = 0.394163$
$r = 0.3622$

$w_1 = 11.17244$; $w_2 = 1.118125$; $w_3 = 2.549476$; $w_4 = 1.144538$; $w_5 = 1.041899$

Finally, the new surrogate function $S_1(\mathbf{x})$ is defined and the GA is run maximising it:

$$\max S_1(\mathbf{x}) = 11.1724\,G_1 + 1.118125\,G_2 + 2.54947\,G_3 + 1.14453\,G_4 + 1.041899\,G_5 \quad (5.57)$$

The optimal solution after running the GA is vector $\mathbf{x}_2$:

$\mathbf{x}_2$	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_{10}	x_{11}	x_{12}
	0.36	269.46	9.51	1.81	3.27	24.08	5.92	1.27	36.38	9.58	0.74	17.27

Additionally, with the GA the values for the goal vector $\mathbf{G}_2$ are found:

$\mathbf{G}_2$	G_1	G_2	G_3	G_4	G_5
	0.948035	0.112962	0.177553	0.993720	0.915898

In this model it is considered that the DM is satisfied with the results obtained and hence, $\mathbf{x}_2$ is the definitive or final land allocation for the sixth 2-year sub-period.

Seventh sub-period

The water constraint is defined as follows:

$$w_1 x_1 + w_2 x_2 + \cdots + w_{12} x_{12} \leq 2{,}512{,}024\ \mathrm{m}^3 \quad (5.58)$$

Vector $\mathbf{x}_1$ is found by running the GA to maximise $F(\mathbf{x})$ subject to land, capital, and water constraints (Step 4) is:

$\mathbf{x}_1$	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_{10}	x_{11}	x_{12}
	186.45	2.88	0.64	2.74	6.32	57.76	3.24	0	97.47	1.88	2.55	18.07

The goal vector $\mathbf{G}_1$ is:

$\mathbf{G}_1$	G_1	G_2	G_3	G_4	G_5
	0.0106464	0.286044	0.410809	0.3861505	0.958034

The values of w_i are calculated to redefine the surrogate function $S_1(\mathbf{x})$ using $u(\mathbf{G}_1)$, and r:

$u(\mathbf{G}_1) = 0.491798$

$r = 1.0886$

$w_1 = 29.55342$; $w_2 = 1.479077$; $w_3 = 2.264393$; $w_4 = 1.485906$; $w_5 = 1.117612$

Finally, the new surrogate function $S_1(\mathbf{x})$ is defined and the GA is run maximising it:

$$\max S_1(\mathbf{x}) = 29.5534\,G_1 + 1.479077\,G_2 + 2.26439\,G_3 + 1.485906\,G_4 + 1.11761\,G_5 \quad (5.59)$$

The optimal solution after running the GA is vector $\mathbf{x}_2$:

$\mathbf{x}_2$	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_{10}	x_{11}	x_{12}
	0.31	216	0.81	0.66	1.38	2.99	3.26	2.68	93.72	36.25	5.69	15.68

Additionally, with the GA the values for the goal vector $\mathbf{G}_2$ are found:

$\mathbf{G}_2$	G_1	G_2	G_3	G_4	G_5
	0.753612	0.041474	0.462565	0.992952	0.831172

In this model it is considered that the DM is satisfied with the results obtained and hence, $\mathbf{x}_2$ is the definitive or final land allocation for the seventh 2-year sub-period.

Eighth sub-period

The water constraint is defined as follows:

$$w_1 x_1 + w_2 x_2 + \cdots + w_{12} x_{12} \leq 845{,}213\ \mathrm{m}^3 \quad (5.60)$$

The values of vector $\mathbf{x}_1$ and the goal vector $\mathbf{G}_1$ are:

$\mathbf{x}_1$	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_{10}	x_{11}	x_{12}
	217.24	2.15	0.49	2.6	4.8	92.57	1.89	0.27	31.86	5.52	1.16	19.18

$\mathbf{G}_1$	G_1	G_2	G_3	G_4	G_5
	0.007858	0.396715	0.144952	0.286193	1.0

The values of $u(\mathbf{G}_1)$, r and w_i are:

$u(\mathbf{G}_1) = 0.330463$
$r = 0.3174$

$w_1 = 13.619$; $w_2 = 1.124276$; $w_3 = 2.684552$; $w_4 = 1.210859$; $w_5 = 1.038727$

Finally, the new surrogate function $S_1(\mathbf{x})$ is defined and the GA is run maximising it:

$$\max S_1(\mathbf{x}) = 13.619\,G_1 + 1.124276\,G_2 + 2.68455\,G_3 + 1.210859\,G_4 + 1.038727\,G_5 \quad (5.61)$$

The optimal solution after running the GA is vector $\mathbf{x}_2$ and the values for the goal vector $\mathbf{G}_2$ are:

$\mathbf{x}_2$	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_{10}	x_{11}	x_{12}
	0.31	269.44	8.1	2.1	2.55	21.91	3.57	2.09	36.71	14.26	1.7	17.16

$\mathbf{G}_2$	G_1	G_2	G_3	G_4	G_5
	0.946515	0.105807	0.185234	0.993743	0.909985

In this model it is considered that the DM is satisfied with the results obtained and hence, x_2 is the definitive or final land allocation for the eighth 2-year sub-period.

Ninth sub-period

The water constraint is defined as follows:

$$w_1 x_1 + w_2 x_2 + \cdots + w_{12} x_{12} \leq 1{,}944{,}114 \text{ m}^3 \quad (5.62)$$

The values of vector $\mathbf{x}_1$ and the goal vector $\mathbf{G}_1$ are:

$\mathbf{x}_1$	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_{10}	x_{11}	x_{12}
	207.58	6.07	0.72	1.62	1.89	47.25	2.44	1.6	86.71	3.73	1.47	18.55

$\mathbf{G}_1$	G_1	G_2	G_3	G_4	G_5
	0.021800	0.259967	0.368581	0.316928	0.983643

The values of $u(\mathbf{G}_1)$, r and w_i are:

$u(\mathbf{G}_1) = 0.461364$
$r = 0.6171$

$w_1 = 9.023497$; $w_2 = 1.298358$; $w_3 = 1.867802$; $w_4 = 1.340069$; $w_5 = 1.068589$

Finally, the new surrogate function $S_1(\mathbf{x})$ is defined and the GA is run maximising it:

$$\max S_1(\mathbf{x}) = 9.0235\,G_1 + 1.29836\,G_2 + 1.8678\,G_3 + 1.34007\,G_4 + 1.06859\,G_5 \quad (5.63)$$

The optimal solution after running the GA is vector $\mathbf{x}_2$ and the values for the goal vector $\mathbf{G}_2$ are:

$\mathbf{x}_2$	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_{10}	x_{11}	x_{12}
	0.55	218.7	1.89	3.42	4.65	25.45	4.2	2.59	97.41	2.86	1.76	16.19

$\mathbf{G}_2$	G_1	G_2	G_3	G_4	G_5
	0.764005	0.114005	0.415011	0.993157	0.858422

In this model it is considered that the DM is satisfied with the results obtained and hence, x_2 is the definitive or final land allocation for the ninth 2-year sub-period.

Tenth sub-period

The water constraint is defined as follows:

$$w_1 x_1 + w_2 x_2 + \cdots + w_{12} x_{12} \leq 705{,}202 \text{ m}^3 \quad (5.64)$$

The values of vector $\mathbf{x}_1$ and the goal vector $\mathbf{G}_1$ are:

$\mathbf{x}_1$	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_{10}	x_{11}	x_{12}
	170.37	0.54	0	3.3	3.92	145.14	1.51	0.71	30.69	3.09	2.55	18.1

$\mathbf{G}_1$	G_1	G_2	G_3	G_4	G_5
	0.001774	0.512317	0.137651	0.440325	0.959478

The values of $u(\mathbf{G}_1)$, r and w_i are:

$u(\mathbf{G}_1) = 0.366199$
$r = 0.3258$

$w_1 = 57.25695$; $w_2 = 1.10721$; $w_3 = 2.795601$; $w_4 = 1.140079$; $w_5 = 1.038414$

Finally, the new surrogate function $S_1(\mathbf{x})$ is defined and the GA is run maximising it:

$$\max S_1(\mathbf{x}) = 57.25695\,G_1 + 1.10721\,G_2 + 2.7956\,G_3 + 1.140079\,G_4 + 1.038414\,G_5 \quad (5.65)$$

The optimal solution after running the GA is vector $\mathbf{x}_2$ and the values for the goal vector $\mathbf{G}_2$ are:

$\mathbf{x}_2$	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_{10}	x_{11}	x_{12}
	0.35	269.73	27.85	1.54	2.05	13.77	3.3	1.93	32.04	8.09	0.85	17.54

$\mathbf{G}_2$	G_1	G_2	G_3	G_4	G_5
	0.965779	0.083912	0.153542	0.993531	0.930035

In this model it is considered that the DM is satisfied with the results obtained and hence, $\mathbf{x}_2$ is the definitive or final land allocation for the tenth 2-year sub-period.

Eleventh sub-period

The water constraint is defined as follows:

$$w_1 x_1 + w_2 x_2 + \cdots + w_{12} x_{12} \leq 1{,}425{,}694\ \mathrm{m}^3 \quad (5.66)$$

The values of vector $\mathbf{x}_1$ and the goal vector $\mathbf{G}_1$ are:

$\mathbf{x}_1$	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_{10}	x_{11}	x_{12}
	192.24	8.37	0	3.66	3.76	75.37	2.07	1.66	70.03	2.25	2.18	18.36

$\mathbf{G}_1$	G_1	G_2	G_3	G_4	G_5
	0.029151	0.334814	0.298235	0.367539	0.973370

The values of $u(\mathbf{G}_1)$, r and w_i are:

$u(\mathbf{G}_1) = 0.442454$
$r = 0.5201$

$w_1 = 6.101531$; $w_2 = 1.211605$; $w_3 = 2.011809$; $w_4 = 1.252037$; $w_5 = 1.058423$

Finally, the new surrogate function $S_1(\mathbf{x})$ is defined and the GA is run maximising it:

$$\max S_1(\mathbf{x}) = 6.101531\,G_1 + 1.211605\,G_2 + 2.011809\,G_3 + 1.252037\,G_4 + 1.058423\,G_5 \quad (5.67)$$

The optimal solution after running the GA is vector $\mathbf{x}_2$ and the values for the goal vector $\mathbf{G}_2$ are:

$\mathbf{x}_2$	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_{10}	x_{11}	x_{12}
	0.26	222.75	3.51	8.09	11.38	28.08	3.81	1.99	66.28	13.9	2.97	16.44

G_2	G_1	G_2	G_3	G_4	G_5
	0.779721	0.141366	0.307659	0.993814	0.871757

In this model it is considered that the DM is satisfied with the results obtained and hence, x_2 is the definitive or final land allocation for the eleventh 2-year sub-period.

Twelfth sub-period

The water constraint is defined as follows:

$$w_1 x_1 + w_2 x_2 + \cdots + w_{12} x_{12} \leq 409{,}607 \text{ m}^3 \quad (5.68)$$

The values of vector $\mathbf{x}_1$ and the goal vector $\mathbf{G}_1$ are:

x_1	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_{10}	x_{11}	x_{12}
	0	210.06	2.16	4.82	22.65	95.54	1.83	1.5	15.96	5.81	2.6	17

G_1	G_1	G_2	G_3	G_4	G_5
	0.734094	0.342292	0.083491	0.996079	0.901347

The values of $u(\mathbf{G}_1)$, r and w_i are:

$u(\mathbf{G}_1) = 0.571064$
$r = 0.4461$

$w_1 = 1.099172$; $w_2 = 1.16266$; $w_3 = 4.752914$; $w_4 = 1.078758$; $w_5 = 1.045566$

Finally, the new surrogate function $S_1(\mathbf{x})$ is defined and the GA is run maximising it:

$$\max S_1(\mathbf{x}) = 1.09917\,G_1 + 1.16266\,G_2 + 4.752914\,G_3 + 1.078758\,G_4 + 1.045566\,G_5 \quad (5.69)$$

The optimal solution after running the GA is vector $\mathbf{x}_2$ and the values for the goal vector $\mathbf{G}_2$ are:

x_2	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_{10}	x_{11}	x_{12}
	0.16	242.27	4.25	3.02	3.49	64.32	8.05	0.72	9.79	26.73	0	17.01

G_2	G_1	G_2	G_3	G_4	G_5
	0.848416	0.221219	0.100915	0.994842	0.901945

In this model it is considered that the DM is satisfied with the results obtained and hence, x_2 is the definitive or final land allocation for the twelfth 2-year sub-period.

Thirteenth sub-period

The water constraint is defined as follows:

$$w_1 x_1 + w_2 x_2 + \cdots + w_{12} x_{12} \leq 2{,}175{,}253 \text{ m}^3 \quad (5.70)$$

The values of vector $\mathbf{x}_1$ and the goal vector $\mathbf{G}_1$ are:

x_1	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_{10}	x_{11}	x_{12}
	192.51	0.27	0	2.18	4.38	54.52	2.19	0.89	90.48	12.51	2.05	18.03

G_1	G_1	G_2	G_3	G_4	G_5
	0.001014	0.275757	0.399154	0.366157	0.956109

The values of $u(\mathbf{G}_1)$, r and w_i are:

$u(\mathbf{G}_1) = 0.478498$
$r = 0.6666$

$w_1 = 187.451$; $w_2 = 1.304417$; $w_3 = 1.818255$; $w_4 = 1.316023$; $w_5 = 1.072679$

Finally, the new surrogate function $S_1(\mathbf{x})$ is defined and the GA is run maximising it:

$$\max S_1(\mathbf{x}) = 187.451\,G_1 + 1.304417\,G_2 + 1.81825\,G_3 + 1.31602\,G_4 + 1.072679\,G_5 \quad (5.71)$$

The optimal solution after running the GA is vector $\mathbf{x}_2$ and the values for the goal vector $\mathbf{G}_2$ are:

x_2	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_{10}	x_{11}	x_{12}
	0.09	269.73	37.8	0.74	1.03	6.32	7.09	4.97	30.21	1.52	0.73	17.39

G_2	G_1	G_2	G_3	G_4	G_5
	0.975158	0.061277	0.143359	0.994145	0.922139

In this model it is considered that the DM is satisfied with the results obtained and hence, x_2 is the definitive or final land allocation for the thirteenth 2-year sub-period.

Fourteenth sub-period

The water constraint is defined as follows:

$$w_1 x_1 + w_2 x_2 + \cdots + w_{12} x_{12} \leq 1{,}576{,}443\ \mathrm{m}^3 \quad (5.72)$$

The values of vector $\mathbf{x}_1$ and the goal vector $\mathbf{G}_1$ are:

x_1	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_{10}	x_{11}	x_{12}
	183.6	4.79	0.61	2.97	0.27	82.89	2.01	0.78	78.32	3.98	1.56	18.09

G_1	G_1	G_2	G_3	G_4	G_5
	0.017237	0.342153	0.333431	0.395984	0.958950

The values of $u(\mathbf{G}_1)$, r and w_i are:

$u(\mathbf{G}_1) = 0.463322$
$r = 0.5738$

$w_1 = 10.44903$; $w_2 = 1.226605$; $w_3 = 1.936436$; $w_4 = 1.255224$; $w_5 = 1.063286$

Finally, the new surrogate function $S_1(\mathbf{x})$ is defined and the GA is run maximising it:

$$\max S_1(\mathbf{x}) = 10.449\,G_1 + 1.226605\,G_2 + 1.936436\,G_3 + 1.255224\,G_4 + 1.063286\,G_5 \quad (5.73)$$

The optimal solution after running the GA is vector $\mathbf{x}_2$ and the values for the goal vector $\mathbf{G}_2$ are:

x_2	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_{10}	x_{11}	x_{12}
	0.27	243	2.43	2.8	4.67	17.01	5.69	2.62	75.65	5.29	2.78	16.53

G_2	G_1	G_2	G_3	G_4	G_5
	0.849176	0.093199	0.334276	0.993849	0.876341

In this model it is considered that the DM is satisfied with the results obtained and hence, x_2 is the definitive or final land allocation for the fourteenth 2-year sub-period.

Fifteenth sub-period

The water constraint is defined as follows:

$$w_1 x_1 + w_2 x_2 + \cdots + w_{12} x_{12} \leq 1{,}300{,}811\,\mathrm{m}^3 \quad (5.74)$$

The values of vector $\mathbf{x}_1$ and the goal vector $\mathbf{G}_1$ are:

x_1	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_{10}	x_{11}	x_{12}
	158.89	2.97	1.29	3.12	7.6	121.41	2.01	1.17	58.62	3.78	0.27	17.82

G_1	G_1	G_2	G_3	G_4	G_5
	0.011660	0.447705	0.251035	0.477464	0.944895

The values of $u(\mathbf{G}_1)$, r and w_i are:

$u(\mathbf{G}_1) = 0.441971$
$r = 0.4655$

$w_1 = 12.54348$; $w_2 = 1.156928$; $w_3 = 2.148699$; $w_4 = 1.176113$; $w_5 = 1.051948$

Finally, the new surrogate function $S_1(\mathbf{x})$ is defined and the GA is run maximising it:

$$\max S_1(\mathbf{x}) = 12.54348\,G_1 + 1.156928\,G_2 + 2.148699\,G_3 + 1.176113\,G_4 + 1.051948\,G_5 \quad (5.75)$$

The optimal solution after running the GA is vector $\mathbf{x}_2$ and the values for the goal vector $\mathbf{G}_2$ are:

x_2	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_{10}	x_{11}	x_{12}
	0.91	262.71	3.05	2.23	6.33	11.13	5.07	3.12	61.12	4.04	3.34	16.53

G_2	G_1	G_2	G_3	G_4	G_5
	0.918378	0.082633	0.272873	0.991639	0.876578

In this model it is considered that the DM is satisfied with the results obtained and hence, x_2 is the definitive or final land allocation for the fifteenth 2-year sub-period.

5.6.1.1 Result Analysis

Table 5.2 contains the decision variables values for the fifteen 2-year sub-periods. While $\mathbf{x}_1$ represents the vector without considering the DM's preferences, $\mathbf{x}_2$ is the vector that results from considering them, and wc is the value of the water constraint for each sub-period.

In Table 5.2, the results clearly show that the model without considering the DM's preferences does not always have the same behaviour. For example, in the first case where the

water constraint has a value of 2,702,473 m^3 (maximum value of wc) the x_1 value is 194 ha, whilst in the fourth case, where the water constraint has a value of 338,251 m^3 (minimum value of wc), the x_1 value is 0.2 ha. This represents a difference of more than 99%. If the cases where the water constraint tends to have a small value are analysed, it can be seen that the algorithm behaves in the same way. For instance, the x_1 value in the fifth sub-period (wc = 475,079) is 0.11 ha and the x_1 value in the twelfth sub-period (wc = 409,607) is 0 ha. Consequently, it is evident that the uncertainty reflected in the water constraint can significantly affect the GA initial results. In other words, since the water constraint is a random variable because rainfall is unpredictable, the land allocated to each activity the first time the algorithm is run, would be completely different for a small amount of water and for a big amount of water.

The decision variable x_1 was chosen for the comparison because it represents the no reclamation program by current management. This means that the practices without considering the DM's preferences are not expected to change considerably. In the case of a big amount of water, the first time the algorithm is run, it assigns more land to x_1 (no reclamation program by current management), x_6 (run-off augmentation plastic cover and gravel) and x_9 (Alfalfa production).

Sub-period		x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_{10}	x_{11}	x_{12}
First	x_1	194.1	2.67	0.47	3.93	3.23	51.2	1.95	1.69	89.22	10.27	2.59	18.16
wc=2,702,473	x_2	0.14	234.36	2.78	2.65	1.06	4.75	1.62	1.23	110	3	1.41	16.16
Second	x_1	203.1	4.28	0.9	1.35	2.16	73.98	1.84	1.48	68.6	1.65	1.98	18.62
wc=1,721,549	x_2	0.48	236.79	0.54	2.72	2.69	19.47	2.98	1.45	83.31	9.96	1.81	16.43
Third	x_1	216.5	0.27	0	2.44	6.03	76.34	2.75	0.91	51.82	2.42	1.08	18.89
wc=1,047,110	x_2	0.1	269.73	26.01	0.19	0.67	4.35	1.08	0.81	51.57	7.62	0.58	17.27
Fourth	x_1	0.2	155.87	4.11	8.34	34.81	134.83	2.51	0.84	13.77	6.99	0.73	16.73
wc=338,251	x_2	0	168.48	0.54	6.13	15.6	132.75	8.94	1.47	6.83	20.89	0.88	16.44
Fifth	x_1	0.11	197.92	5.59	5.3	43.15	82.61	1.62	0	23.22	1.68	0.45	17.24
wc=475,079	x_2	0.27	143.91	0.81	10.2	13.76	151.15	5.83	1	15.32	18.66	1.58	16.13
Sixth	x_1	162.1	2.98	0.3	3.25	10.42	138.81	0.27	0	39.15	2.73	1.51	18.07
wc=803,335	x_2	0.36	269.46	9.51	1.81	3.27	24.08	5.92	1.27	36.38	9.58	0.74	17.27
Seventh	x_1	186.5	2.88	0.64	2.74	6.32	57.76	3.24	0	97.47	1.88	2.55	18.07
wc=2,512,024	x_2	0.31	216	0.81	0.66	1.38	2.99	3.26	2.68	93.72	36.25	5.69	15.68
Eighth	x_1	217.2	2.15	0.49	2.6	4.8	92.57	1.89	0.27	31.86	5.52	1.16	19.18
wc=845,213	x_2	0.31	269.44	8.1	2.1	2.55	21.91	3.57	2.09	36.71	14.26	1.7	17.16
Ninth	x_1	207.6	6.07	0.72	1.62	1.89	47.25	2.44	1.6	86.71	3.73	1.47	18.55
wc=1,944,114	x_2	0.55	218.7	1.89	3.42	4.65	25.45	4.2	2.59	97.41	2.86	1.76	16.19
Tenth	x_1	170.4	0.54	0	3.3	3.92	145.14	1.51	0.71	30.69	3.09	2.55	18.1
wc=705,202	x_2	0.35	269.73	27.85	1.54	2.05	13.77	3.3	1.93	32.04	8.09	0.85	17.54
Eleventh	x_1	192.2	8.37	0	3.66	3.76	75.37	2.07	1.66	70.03	2.25	2.18	18.36
wc=1,425,694	x_2	0.26	222.75	3.51	8.09	11.38	28.08	3.81	1.99	66.28	13.9	2.97	16.44
Twelfth	x_1	0	210.06	2.16	4.82	22.65	95.54	1.83	1.5	15.96	5.81	2.6	17
wc=409,607	x_2	0.16	242.27	4.25	3.02	3.49	64.32	8.05	0.72	9.79	26.73	0	17.01
Thirteenth	x_1	192.5	0.27	0	2.18	4.38	54.52	2.19	0.89	90.48	12.51	2.05	18.03
wc=2,175,253	x_2	0.09	269.73	37.8	0.74	1.03	6.32	7.09	4.97	30.21	1.52	0.73	17.39
Fourteenth	x_1	183.6	4.79	0.61	2.97	0.27	82.89	2.01	0.78	78.32	3.98	1.56	18.09
wc=1,576,443	x_2	0.27	243	2.43	2.8	4.67	17.01	5.69	2.62	75.65	5.29	2.78	16.53
Fifteenth	x_1	158.9	2.97	1.29	3.12	7.6	121.41	2.01	1.17	58.62	3.78	0.27	17.82
wc=1,300,811	x_2	0.91	262.71	3.05	2.23	6.33	11.13	5.07	3.12	61.12	4.04	3.34	16.53

Table 5.2 Final decision variables without considering and considering the DM's preferences

Sub-period		G_1	G_2	G_3	G_4	G_5
First	$\mathbf{G_1}$	0.009632	0.268126	0.391161	0.360843	0.962670
	$\mathbf{G_2}$	0.819518	0.051039	0.460758	0.994283	0.856954
Second	$\mathbf{G_1}$	0.015716	0.330685	0.290479	0.332118	0.987189
	$\mathbf{G_2}$	0.825856	0.095065	0.366847	0.993298	0.871379
Third	$\mathbf{G_1}$	0.001014	0.353566	0.223019	0.288196	1.00
	$\mathbf{G_2}$	0.964259	0.053134	0.227507	0.994321	0.915582
Fourth	$\mathbf{G_1}$	0.547022	0.471849	0.074466	0.996016	0.887202
	$\mathbf{G_2}$	0.587833	0.426707	0.081980	0.996473	0.871952
Fifth	$\mathbf{G_1}$	0.695057	0.346329	0.101192	0.995509	0.914268
	$\mathbf{G_2}$	0.502408	0.476134	0.108184	0.995943	0.855520
Sixth	$\mathbf{G_1}$	0.010646	0.503038	0.167411	0.467187	0.958021
	$\mathbf{G_2}$	0.948035	0.112962	0.177553	0.993720	0.915898
Seventh	$\mathbf{G_1}$	0.0106464	0.286044	0.410809	0.3861505	0.958034
	$\mathbf{G_2}$	0.753612	0.041474	0.462565	0.992952	0.831172
Eighth	$\mathbf{G_1}$	0.007858	0.396715	0.144952	0.286193	1.0
	$\mathbf{G_2}$	0.946515	0.105807	0.185234	0.993743	0.909985
Ninth	$\mathbf{G_1}$	0.021800	0.259967	0.368581	0.316928	0.983643
	$\mathbf{G_2}$	0.764005	0.114005	0.415011	0.993157	0.858422
Tenth	$\mathbf{G_1}$	0.001774	0.512317	0.137651	0.440325	0.959478
	$\mathbf{G_2}$	0.965779	0.083912	0.153542	0.993531	0.930035
Eleventh	$\mathbf{G_1}$	0.029151	0.334814	0.298235	0.367539	0.973370
	$\mathbf{G_2}$	0.779721	0.141366	0.307659	0.993814	0.871757
Twelfth	$\mathbf{G_1}$	0.734094	0.342292	0.083491	0.996079	0.901347
	$\mathbf{G_2}$	0.848416	0.221219	0.100915	0.994842	0.901945
Thirteenth	$\mathbf{G_1}$	0.001014	0.275757	0.399154	0.366157	0.956109
	$\mathbf{G_2}$	0.975158	0.061277	0.143359	0.994145	0.922139
Fourteenth	$\mathbf{G_1}$	0.017237	0.342153	0.333431	0.395984	0.958950
	$\mathbf{G_2}$	0.849176	0.093199	0.334276	0.993849	0.876341
Fifteenth	$\mathbf{G_1}$	0.011660	0.447705	0.251035	0.477464	0.944895
	$\mathbf{G_2}$	0.918378	0.082633	0.272873	0.991639	0.876578

Table 5.3 Final goal values without considering and considering the DM's preferences

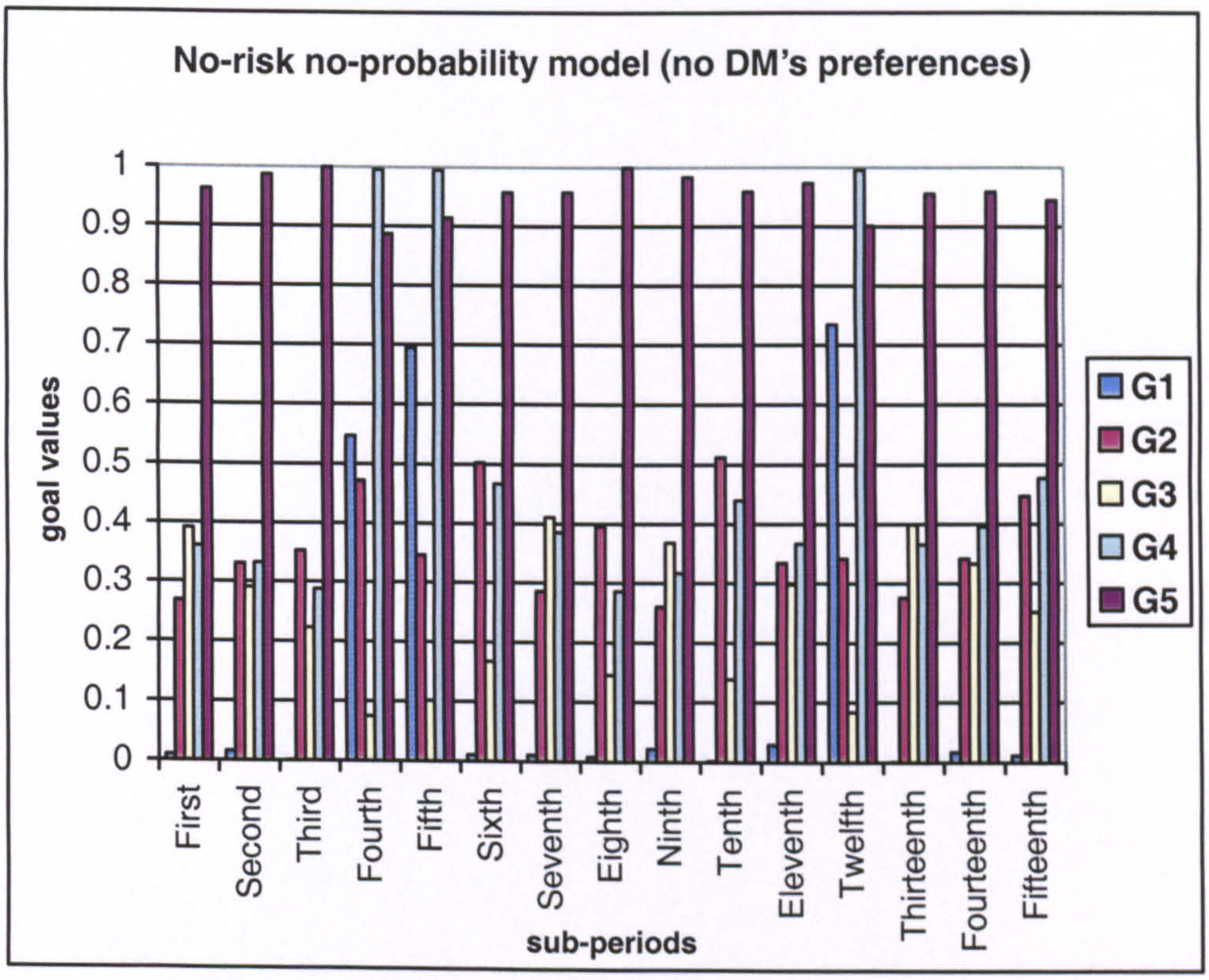


Figure 5.3 Goals' values for the fifteen sub-periods without the DM's preferences

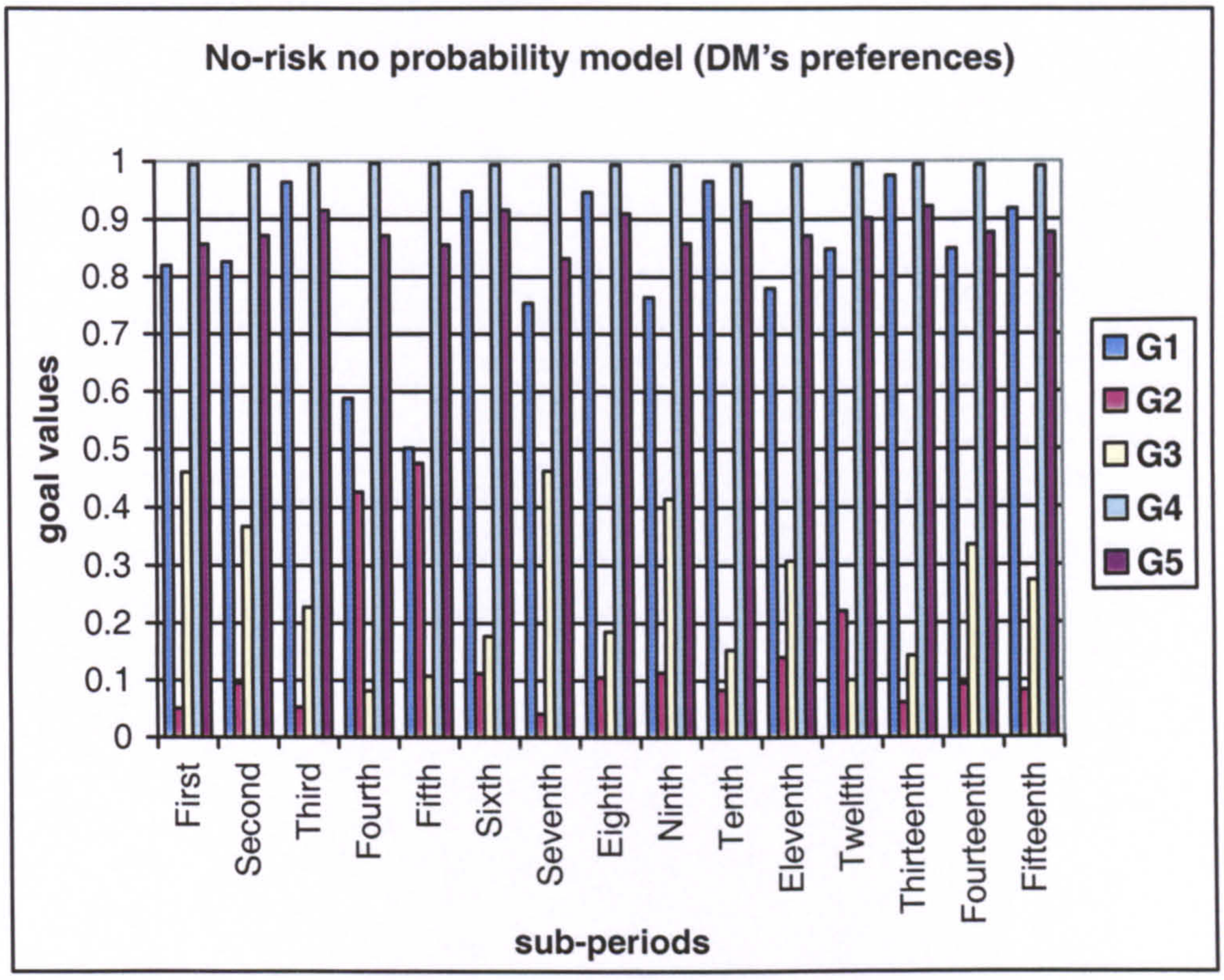


Figure 5.4 Goals' values for the fifteen sub-periods considering the DM's preferences

On the other hand, in the case of a small amount of water, the algorithm assigns more land to x_2 (contour furrowing livestock production good range conditions), x_5 (run-off augmentation compacting and salt treatment) and x_6 (run-off augmentation plastic cover and gravel). It is clear that the decision variables x_5 and x_6 are related to the improvement of the water use, corresponding to the fact that water constraint has a small value.

Table 5.3 presents the goals' values for the fifteen 2-year sub-periods. Where $\mathbf{G}_1$ represents the vector without considering the DM's preferences and $\mathbf{G}_2$ is the vector that results from considering them. In this problem five objectives have been defined: livestock production, water run-off, selected crops, sediment and fish yield. These objectives were normalised using equation 5.6, transforming them into five goals. The values each goal can take are in the range of [0,1].

Figure 5.3 represents the goals' values for the 15 sub-periods without considering the DM's preferences. From this figure, it can be seen that the goal that has the biggest values is G_5 (fish yield).

From Table 5.3, the results reveal that the value of G_5 in the fifteen sub-periods ranges from 0.831172 (smallest) to 1.0 (biggest). Thus, it is possible to conclude that G_5 is the goal that performs the best throughout the fifteen sub-periods. Analysing the remaining goals, G_1 (livestock production) has very small values in the cases where the water constraint has values bigger than 500,000 m^3. On the contrary, in the sub-periods where the water constraint is smaller the G_1 values are 0.547022 for the fourth sub-period, 0.695057 for the fifth sub-period and 0.734094 for the twelfth sub-period. Hence, as was expected, the value of the water constraint and the uncertainty related to it directly affect the goals' values.

Figure 5.4 represents the goals' values for the 15 sub-periods considering the DM's preferences. From this figure, it is evident that the goal that has the biggest values is G_4 (sediment). According to Table 5.3, all G_4 values are greater than 0.99, thus this goal is the one that performs best. Analysing the remaining goals, it can be seen that G_5 and G_1 also have a good performance, whilst G_2 and G_3 have values smaller than 0.5.

It would be interesting to compare the goal values with and without considering the DM's preferences. Figure 5.5 and 5.6 show this comparison for each of the fifteen sub-periods. On the one hand, these figures show that the algorithm's behaviour is very similar in the cases of the first, second, third, sixth, seventh, eighth, ninth, tenth, eleventh, thirteenth, fourteenth and fifteenth sub-periods. On the other hand, in the fourth, fifth and twelfth sub-periods, where the water constraint is small, the algorithm follows the same pattern yielding similar results.

The DM's preferences model defined in subsection 5.4.3, will be used in the results analysis as follows:

G_3, crops $\succ$ G_1, livestock $\succ$ G_4, sediment $\succ$ G_2, run-off $\succ$ G_5, fish

where G_3 is preferred to G_1, G_1 is preferred to G_4, G_4 is preferred to G_2, and G_2 is preferred to G_5.

For instance, in the first sub-period, the analysis is made in the order the DM set the preferences:

$G_3 \rightarrow G_1 \rightarrow G_4 \rightarrow G_2 \rightarrow G_5$

No-preferences:

$$G_3 = 0.391161 \rightarrow G_1 = 0.009632 \rightarrow G_4 = 0.360843 \rightarrow G_2 = 0.268126 \rightarrow G_5 = 0.96267$$

Preferences:

$$G_3 = 0.460758 \rightarrow G_1 = 0.819518 \rightarrow G_4 = 0.994283 \rightarrow G_2 = 0.051039 \rightarrow G_5 = 0.856954$$

In conclusion, once the DM's preference model is introduced into the algorithm, it is expected that the goals' values will be affected. Although, according to the DM, G_3 is preferred to G_1, the increment in G_1 is much bigger than that of G_3. This also occurs in G_4, whilst in G_2 a decrement is reflected. Essentially, this happens since it is not possible to affect one goal without affecting the others. In other words, the problem objectives are in conflict.

These results can also give the level of achievement for each goal. For example, the level of achievement in the no-preferences model for G_3 is 39.11%, G_1 is 0.96%, G_4 is 36.08%, G_2 is 26.81% and G_5 is 96.26%. In the preferences model, the level of achievement for G_3 is 46.07%, G_1 is 81.95%, G_4 is 99.42%, G_2 is 5.10% and G_5 is 85.69%. This comparison makes the difference between the no-preferences model and the preferences model clearer.

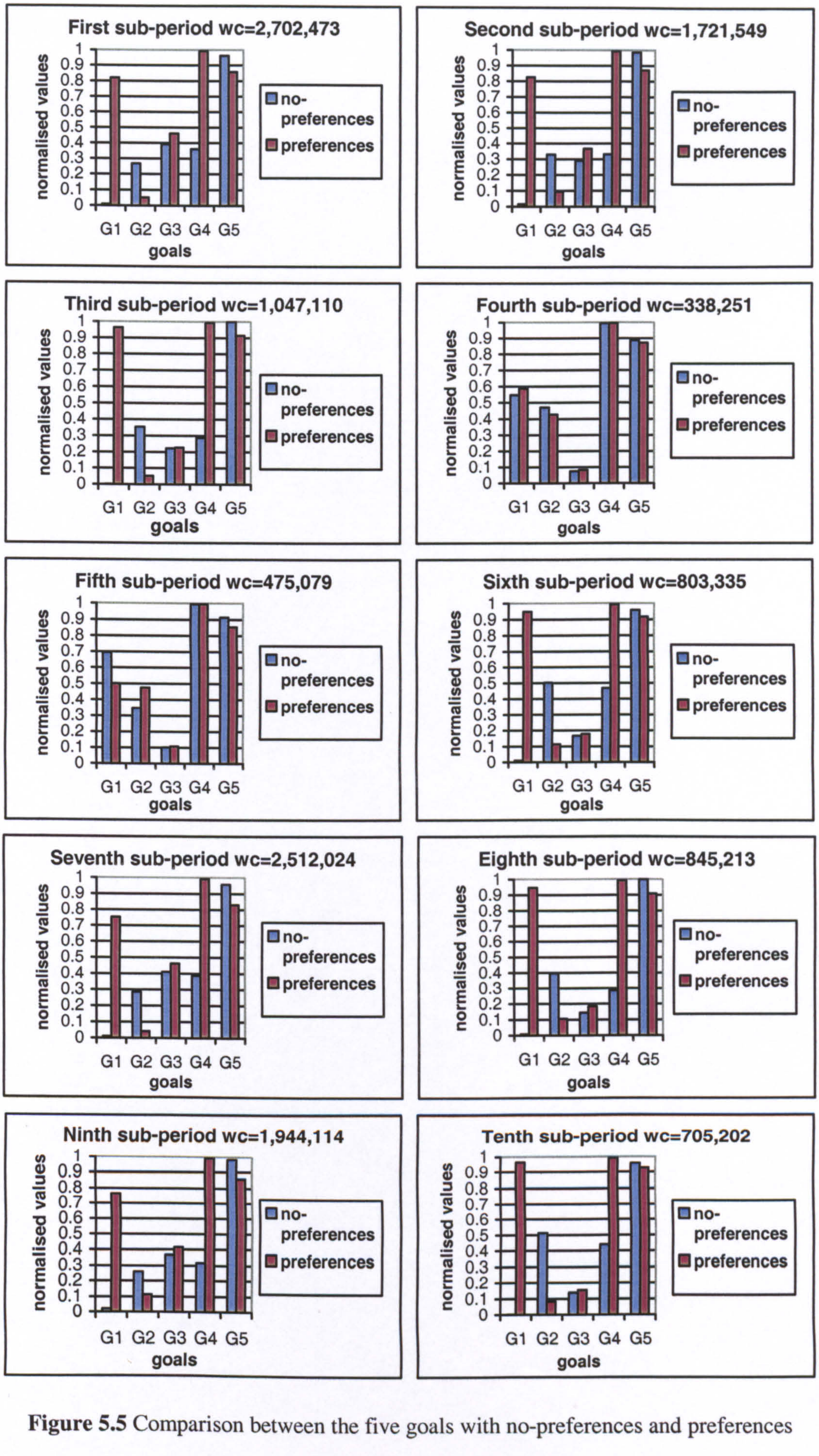


Figure 5.5 Comparison between the five goals with no-preferences and preferences

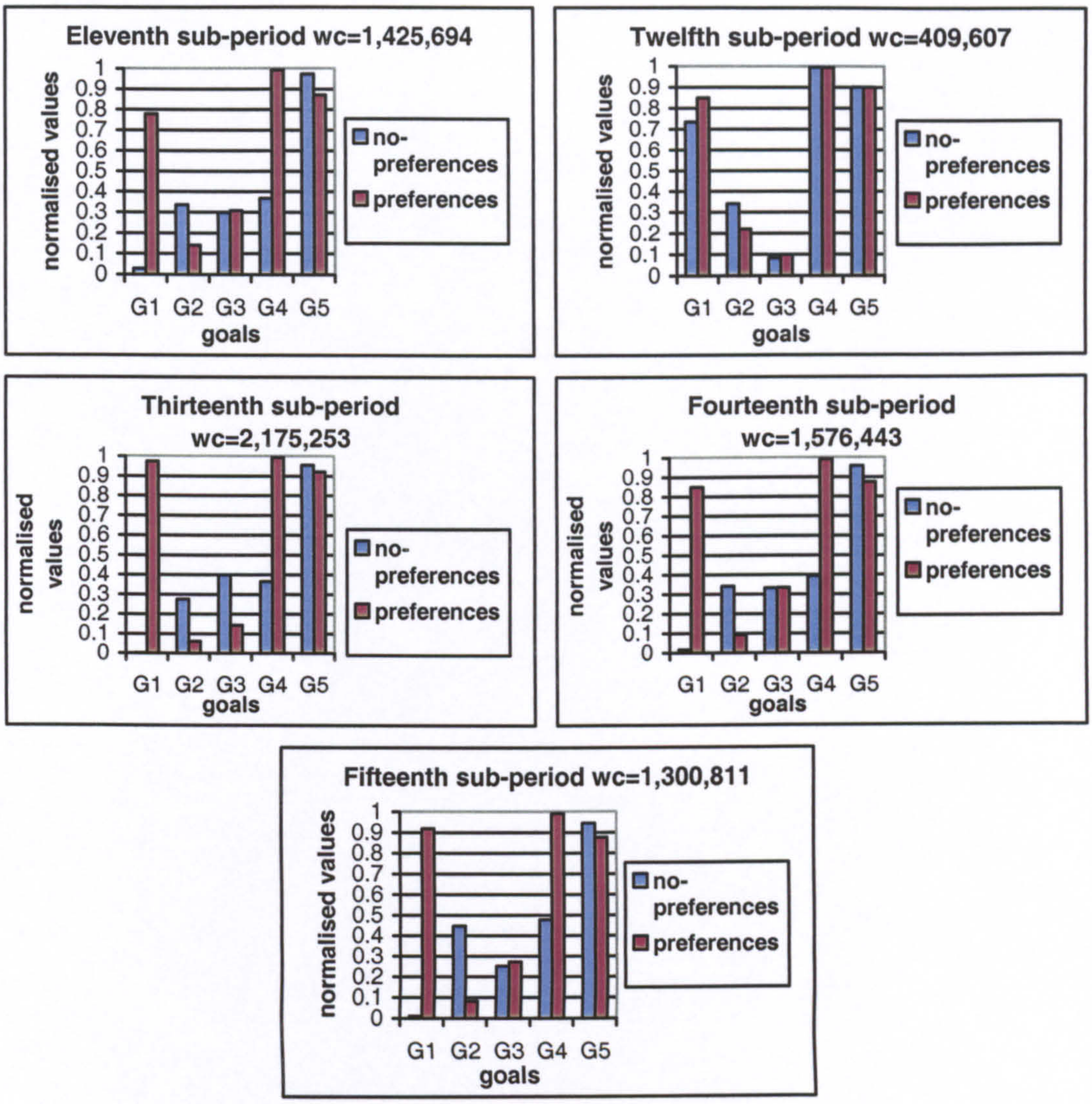


Figure 5.6 Comparison between the five goals with no-preferences and preferences

Analysing Figures 5.5 and 5.6, it can be seen that the changes in the five goals for the second, third, sixth, seventh, eighth, ninth, tenth, eleventh, thirteenth, fourteenth and fifteenth cases have the same tendency.

In the fourth, fifth and twelfth sub-periods the goals' values remain almost the same. For instance, in the fourth sub-period, the analysis is made in the order the DM set the preferences:

$G_3 \rightarrow G_1 \rightarrow G_4 \rightarrow G_2 \rightarrow G_5$

No-preferences:

$$G_3 = 0.074466 \rightarrow G_1 = 0.547022 \rightarrow G_4 = 0.996016 \rightarrow G_2 = 0.471849 \rightarrow G_5 = 0.887202$$

Preferences:

$$G_3 = 0.081980 \rightarrow G_1 = 0.587833 \rightarrow G_4 = 0.996473 \rightarrow G_2 = 0.426707 \rightarrow G_5 = 0.871952$$

It is possible to see how the goals G_3, G_1 and G_4 have been slightly incremented and the goals G_2 and G_5 have decreased slightly. The results clearly show that in this sub-period the DM's preferences model is completely followed by the algorithm.

Hitherto, it has been demonstrated how the DM's preferences can be introduced into the GA through the use of a method that translates the preferences into mathematical functions.

To validate the results obtained with the no-risk no-probability of achievement model, they are compared with those obtained using @Risk and RiskOptimiser in subsection 5.6.3.

5.6.2 No-Risk Probability of Achievement Model

Following the PROTRADE method definition, once the new goal vector $\mathbf{G_2}$ is obtained, the following step (step 8) is taken to generate a vector $\mathbf{V_1}$ that expresses the trade-off between the goals' values and their probability of achievement. The no-risk probability of achievement model proposed in this thesis is outlined in Figure 5.7.

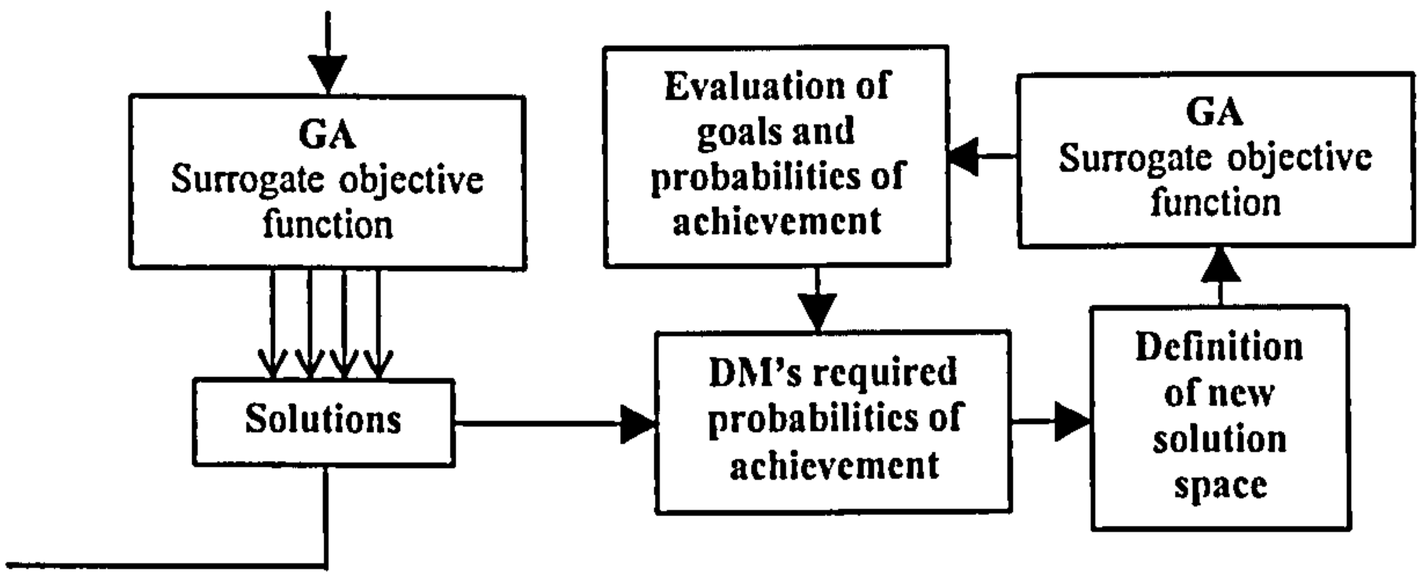


Figure 5.7 No-risk probability of achievement model

It has been decided that in order to analyse the impact of including the probability of achievement into the model only the first 2-year sub-period will be considered.

Firstly the DM has to analyse the vector $\mathbf{U_2}$ (equation 5.11). This vector has the objectives' values $[z_1(\mathbf{x_2}), z_2(\mathbf{x_2}), z_3(\mathbf{x_2}), z_4(\mathbf{x_2}), z_5(\mathbf{x_2})]$:

$$\mathbf{U_2} = \begin{bmatrix} 32.33 \\ 36776.78 \\ 833104.55 \\ -43.89 \\ 60460.23 \end{bmatrix}$$

If the DM is not satisfied with the results obtained in the no-risk no-probability of achievement model (sub-section 5.6.1), it is necessary to generate vector $\mathbf{V_1}$ using $\mathbf{G_2}$ goal values of the first 2-year sub-period. The DM defines an initial probability for each goal. Initially, it is supposed that the probability of achievement $(1-\alpha_i)$ is equal to 0.5 and that it is the same for the five objectives, so the vector $\mathbf{V_1}$ is defined as follows:

$$\mathbf{V}_1 = \begin{bmatrix} (0.819518, 0.5) \\ (0.051039, 0.5) \\ (0.460758, 0.5) \\ (0.994283, 0.5) \\ (0.856954, 0.5) \end{bmatrix}$$

From equation 5.13 the probability of achievement is read as follows:

$$prob[G_1 \geq 0.819518] \geq 0.5$$

In addition, the DM selects the pair $(G_i(\mathbf{x}_2), 1\text{-}\alpha_i)$ that is least satisfactory to him or her. According to the goal ranking proposed by the DM $G_3 > G_1 > G_4 > G_2 > G_5$, it is possible to see that G_3 is more important than G_1 and so on. Although the goal that is least likely to be achieved is G_2, the most important goal of all is G_3. Thus the DM has to specify new values for the pair $(G_3(\mathbf{x}_2), 1\text{-}\alpha_3)$. The DM wants an increment on G_3 from 0.460758 to 0.5 and with a probability of achievement of at least 60%. In order to define the new probability, the value from optimising (maximising) the individual objective function for goal three is needed. This value is obtained from equation 5.43 and is 1,808,033.08 kg. Then the probability is defined as follows:

$$prob[z_3(\mathbf{x}) \geq (0.5)(1{,}808{,}033.08)] \geq 0.6$$

The no-risk probability of achievement model is defined again using the coefficients' expected values (Appendix B) in the five objectives to be optimised, and because it does not consider risk, these will not change their values in the whole model's solution.

If the DM's requirements (defined above) are considered a new constraint has to be added, defining a new solution space $\mathbf{D}_2$ (Goicoechea et al., 1982):

$$x_1 + x_2 + \cdots + x_{12} \leq 380 \text{ ha}, \qquad \textit{Land}$$

$$q_1 x_1 + q_2 x_2 + \cdots + q_{12} x_{12} \leq 200{,}000 \text{ \$}, \qquad \textit{Capital}$$

$$w_1 x_1 + w_2 x_2 + \cdots + w_{12} x_{12} \leq 2{,}702{,}473 \text{ m}^3, \qquad \textit{Water}$$

$$\sum_{j=1}^{12} E(c_j) x_j + k_{\alpha 3} [\mathbf{x}' \mathbf{A} \mathbf{x}]^{1/2} \geq (0.5)(1{,}808{,}033.08) \text{ new constraint}$$

where c_j represent the crop yield in kilograms/hectares with treatment j and $\mathbf{A}$ is the variance-covariance matrix given by:

$$\mathbf{A} = \begin{bmatrix} \text{var}(c_1) & \text{cov}(c_1, c_2) & \cdots & \text{cov}(c_1, c_{11}) & \text{cov}(c_1, c_{12}) \\ \text{cov}(c_2, c_1) & \text{var}(c_2) & \cdots & \text{cov}(c_2, c_{11}) & \text{cov}(c_2, c_{12}) \\ \vdots & \vdots & \cdots & \vdots & \vdots \\ \text{cov}(c_{11}, c_1) & \text{cov}(c_{11}, c_2) & \cdots & \text{var}(c_{11}) & \text{cov}(c_{11}, c_{12}) \\ \text{cov}(c_{12}, c_1) & \text{cov}(c_{12}, c_2) & \cdots & \text{cov}(c_{12}, c_{11}) & \text{var}(c_{12}) \end{bmatrix}$$

If it is assumed that c_i are independent random variables, then $\text{cov}(c_i, c_j) = 0$ for all $i \neq j$ and the variance-covariance matrix is diagonal.

From Goicoechea et al. (1982), to find $k_{\alpha 3}$ the standard normal distribution function $\Phi_{normal}(k_{\alpha 3}) = 1 - 0.6$ is used. From standard normal distribution tables, it is found that the value for $k_{\alpha 3}$ = -0.255. The values of c_j and the variances are obtained from Table B.4 in Appendix B. Thus, redefining the new constraint:

$$\sum_{j=1}^{12} E(c_j)x_j - 0.255\left[\sum_{j=1}^{12} \text{var}(c_j)x_j^2\right]^{1/2} \geq (0.5)(1{,}808{,}033.08)$$

Then, $S_1(\mathbf{x})$ is maximised using $\mathbf{x} \in \mathbf{D}_2$ to find the vector $\mathbf{x}_3$. Where $S_1(\mathbf{x})$ is defined as in equation 5.46:

$$\max S_1(\mathbf{x}) = 20.2569\,G_1 + 1.30669\,G_2 + 1.833115\,G_3 + 1.316075\,G_4 + 1.071798\,G_5$$

In order to obtain vector $\mathbf{x}_3$, the GA is modified and run again using the new constraint:

$$\mathbf{x}_3 = \begin{bmatrix} 0 \\ 220 \\ 18 \\ 2.65 \\ 1.06 \\ 4.75 \\ 1.62 \\ 1.23 \\ 110 \\ 3 \\ 1.14 \\ 16.16 \end{bmatrix}$$

Once that $G_3 = 0.5$ has been achieved and with a minimum probability of 0.6, it is necessary to know the rest of the goals' values and their probabilities of achievement. The first goal to be analysed is G_1 (livestock) and the new constraint is defined as follows:

$$\sum_{j=1}^{12} E(l_j)x_j + k_{\alpha 1}\left[\sum_{j=1}^{12} \text{var}(l_j)x_j^2\right]^{1/2} \geq 39.45\,G_1$$

where l_j represent the number of livestock heads (animal units per hectare) with treatment j. The values of l_j and the variances are obtained from Table B.1 in Appendix B, the values for x_2 and x_3 are taken from the vector $\mathbf{x}_3$, yielding the following equation:

$$30.907 + 12.10\,k_{\alpha 1} \geq 39.45\,G_1$$

where $G_1 = 0.819518$ is taken from the vector $\mathbf{V}_1$, solving the equation $k_{\alpha 1} \geq 0.12$ and from normal distribution tables $\Phi(0.12) = 0.54776 = \alpha_1$, then 1- $\alpha_1 = 0.45224$.

The second goal to be analysed is G_2 (run-off) and the new constraint is defined as follows:

$$\sum_{j=1}^{12} E(r_j)x_j + k_{\alpha 2}\left[\sum_{j=1}^{12} \text{var}(r_j)x_j^2\right]^{1/2} \geq 720{,}439.2344\,G_2 + 118.9756$$

where r_j is the water run-off yield in cubic meters/hectare with treatment j. The values of r_j and the variances are obtained from Table B.3 in Appendix B, the values for x_j are taken from the vector $\mathbf{x}_3$, yielding the following equation:

$$36505 + 33501\,k_{\alpha 2} \geq 720{,}439.2344\,G_2 + 118.9756$$

where $G_2 = 0.051039$ is taken from the vector $\mathbf{V}_1$, solving the equation $k_{\alpha 2} \geq 0.01$ and from normal distribution tables $\Phi(0.01) = 0.50399 = \alpha_2$, then $1 - \alpha_2 = 0.49601$.

The third goal to be analysed is G_4 (sediment) and the new constraint is defined as follows:

$$-\sum_{j=1}^{12} E(s_j)x_j + k_{\alpha 4}\left[\sum_{j=1}^{12} \text{var}(s_j)x_j^2\right]^{1/2} \geq 6041.35\,G_4 - 6050.7$$

where s_j is the sediment yield in cubic meters/hectare with treatment j. The values of s_j and the variances are obtained from Table B.5 in Appendix B, the values for x_j are taken from the vector $\mathbf{x}_3$, yielding the following equation:

$$-47.13 + 29.71\,k_{\alpha 4} \geq 6041.45\,G_4 - 6050.7$$

where $G_4 = 0.994283$ is taken from the vector $\mathbf{V}_1$, solving the equation $k_{\alpha 4} \geq 0.13$ and from normal distribution tables $\Phi(0.13) = 0.55172 = \alpha_2$, then $1 - \alpha_2 = 0.448$.

The last goal to be analysed is G_5 (fish) and the new constraint is defined as follows:

$$\sum_{j=1}^{12} E(f_j)x_j + k_{\alpha 5}\left[\sum_{j=1}^{12} \text{var}(f_j)x_j^2\right]^{1/2} \geq 70{,}552\,G_5$$

where f_j is the fish yield in kilograms/hectare with treatment j. The values of f_j and the variances are obtained from Table B.5 in Appendix B, the value for x_{12} are taken from the vector $\mathbf{x}_3$, yielding the following equation:

$$4040.28 + 2160\,k_{\alpha 5} \geq 70{,}552\,G_5$$

where $G_5 = 0.856954$ is taken from the vector $\mathbf{V}_1$, solving the equation $k_{\alpha 5} \geq 0.0001$ and from normal distribution tables $\Phi(0.0001) = 0.5 = \alpha_2$, then $1 - \alpha_2 = 0.5$.

$$\mathbf{V}_2 = \begin{bmatrix} (0.819518, 0.45) \\ (0.051039, 0.49) \\ (0.5, 0.6) \\ (0.994283, 0.44) \\ (0.856954, 0.5) \end{bmatrix}$$

From these results, it can be seen that when the probability of achievement of goal three was modified, the other goals' probabilities of achievement were also modified.

5.6.3 Risk No-Probability of Achievement Model

The Risk no-probability of achievement model consists of including risk in the coefficients of the objective function as well as in the decision variables.

The risk no-probability of achievement model proposed in this thesis is shown in Figure 5.8. This model is used to find the results presented in this sub-section.

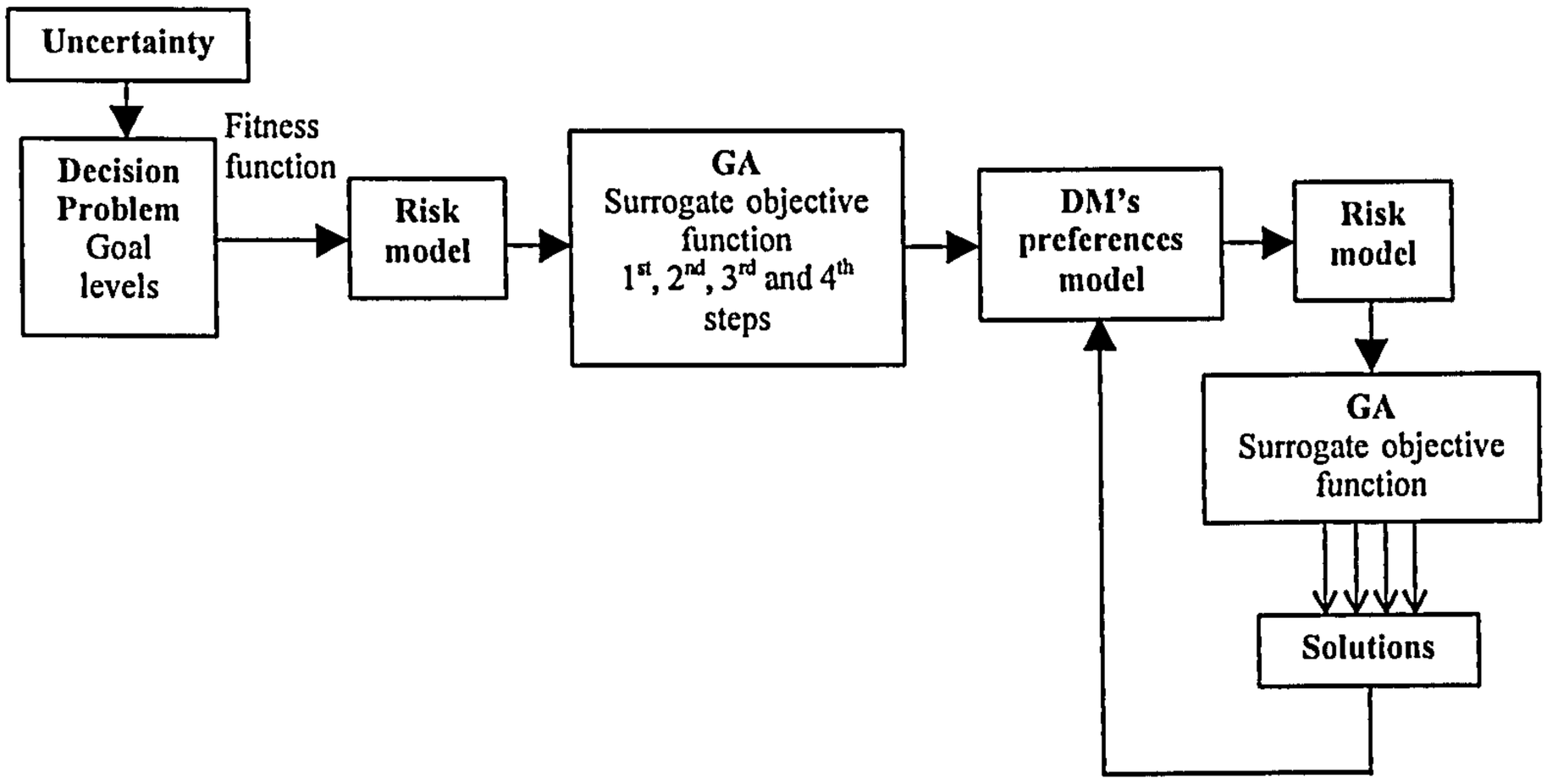


Figure 5.8 Risk no-probability of achievement model

The risk no-probability of achievement model was solved using the same algorithm (GA and PROTRADE) used in subsection 5.6.1, but including a normal distribution. This distribution will be used to generate random values for the decision variables' parameters. The GA has been defined in section 5.5 of this chapter. This problem was also solved by Duenas et al.(2002) using @Risk 4.0 and RiskOptimiser, in such a way as to compare the results.

@RISK is a decision and risk analysis programme based on the Monte Carlo technique. It allows DMs to explore the range of possible outcomes for any decision by using probability distributions. @RISK randomly samples from the probability distribution functions and records the resulting outcomes during a simulation. The result is a distribution of possible outcomes, and the probabilities of each outcome occurring. This not only demonstrates what could happen, but how likely it is to happen, and therefore assists the decision-maker in making his/her decision by helping them recognise that some outcomes are more likely to occur than others, and should therefore be given more weight in their evaluation.

RISKOptimiser achieves the optimisation of @RISK models. This is a stochastic optimisation add-in for Microsoft Excel, that combines genetic algorithm technology with the Monte Carlo simulation engine of @RISK to optimise models that include uncertain "stochastic" factors. It performs optimisation under uncertainty, finding the best combination of parameters while accounting for random, uncontrolled factors. RISKOptimiser runs multiple simulations, each time using genetic algorithms to find a better set of parameters to optimise simulation results.

As mentioned above the case study is solved using both a GA and RISKOptimiser. In both methods two models, each with variations, are constructed, one that does not take risk into account and another that does. This is done so that the outputs could be compared to see if the decision made based on the risk-free model would be the same as that made with the one that accounted for risk.

The water constraint used for both the GA and @Risk is 2,702,473 m^3, which is that, which corresponds with the maximum value that was obtained when the Monte Carlo simulation was performed.

Table 5.4 shows the results of the GA and PROTRADE algorithm considering risk and compares them with those found without risk (subsection 5.6.1). These results do not consider the DM's preferences and are rounded to the nearest integer. Figure 5.9 was built using these values.

Practices	**No-risk (in Hectares)**	**Risk (in Hectares)**
x_1	194	69
x_2	3	45
x_3	0	25
x_4	4	24
x_5	3	43
x_6	51	26
x_7	2	3
x_8	2	2
x_9	89	109
x_{10}	10	9
x_{11}	3	9
x_{12}	18	15

Table 5.4 Results of the land allocation using the GA and PROTRADE algorithm

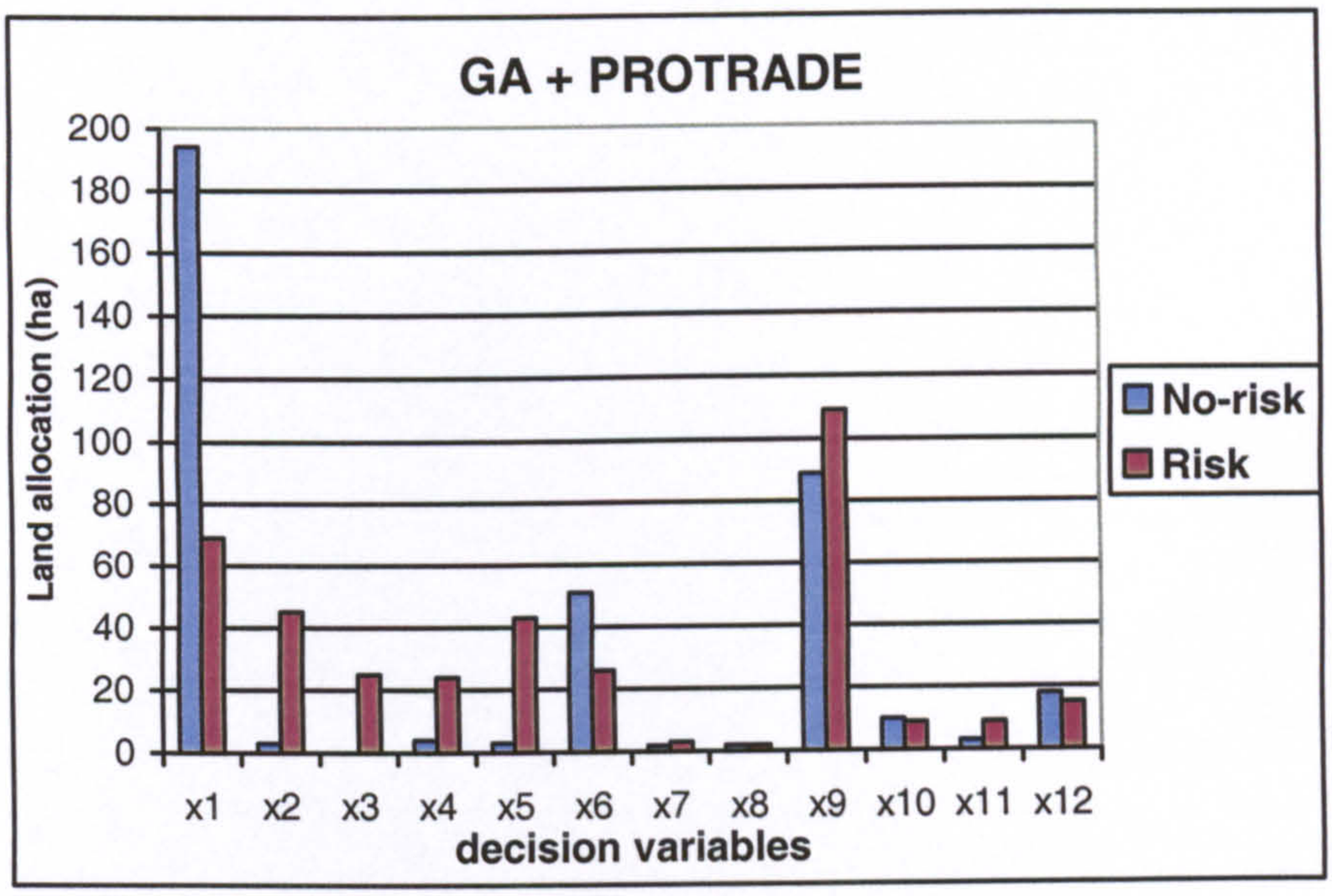


Figure 5.9 Land allocation comparison between no-risk and risk models no-preferences

In the risk model, land allocation tends to be more distributed between all the decision variables than in the no-risk model.

The maximum values for the objective function represented as the addition of the five objectives are 1.270746 (no risk) and 0.825715 (risk). In conclusion the results found considering uncertainty (risk) in the decision variables are significantly different to those found in the no risk model.

The values of the decision variables are used to generate the initial goal vector $\mathbf{G}_1$.

$$\mathrm{G}_1 = \begin{bmatrix} G_1 \\ G_2 \\ G_3 \\ G_4 \\ G_5 \end{bmatrix}$$

The goal vector is generated for both no-risk and risk models and the results are presented in Table 5.5. Figure 5.10 is constructed using these values and basically compares the results of the models.

Goals	No-risk	Risk
G_1	0.009632	0.171863
G_2	0.268126	0.195684
G_3	0.391161	0.474402
G_4	0.360843	0.819701
G_5	0.962670	0.803467

Table 5.5 Goal vector $\mathbf{G}_1$ for no-risk and risk models

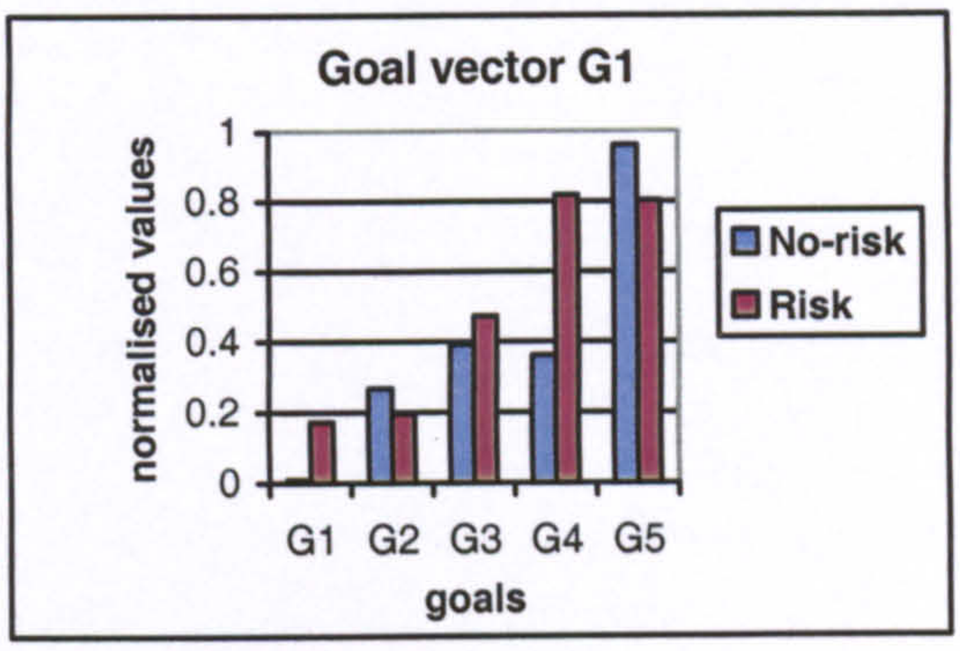


Figure 5.10 Comparison of the goal vector for no-risk and risk models (no-preferences)

From the goal vector it is possible to observe that in the no risk model the level of achievement for G_3 is 39.11%, G_1 is 0.96%, G_4 is 36.08%, G_2 is 26.81% and G_5 is 96.26% (Table 5.5). By contrast, in the risk model the level of achievement for G_3 is 47.44%, G_1 is 17.19%, G_4 is 81.97%, G_2 is 19.57% and G_5 is 80.34%. It is important to note that the goal vector $\mathbf{G}_1$ contains the goals' values without considering the DM's preferences.

At this point it is necessary to consider the DM preferences model presented in sub-section 5.4.3:

$$G_3,\text{crops} \succ G_1,\ \text{livestock} \succ G_4,\ \text{sediment} \succ G_2,\ \text{run-off} \succ G_5,\ \text{fish}$$

Maximising $S_1(\mathrm{x})$ using equation 5.9 yields the results shown in Table 5.6. $S_1(\mathrm{x})$ is the new surrogate function obtained after considering the DM's preferences.

Practices	**No-risk (in Hectares)**	**Risk (in Hectares)**
x_1	0	8
x_2	234	100
x_3	3	21
x_4	3	28
x_5	1	26
x_6	5	33
x_7	2	1
x_8	1	4
x_9	110	111
x_{10}	3	17
x_{11}	1	10
x_{12}	16	14

Table 5.6 Results of the land allocation maximising $S_1(x)$

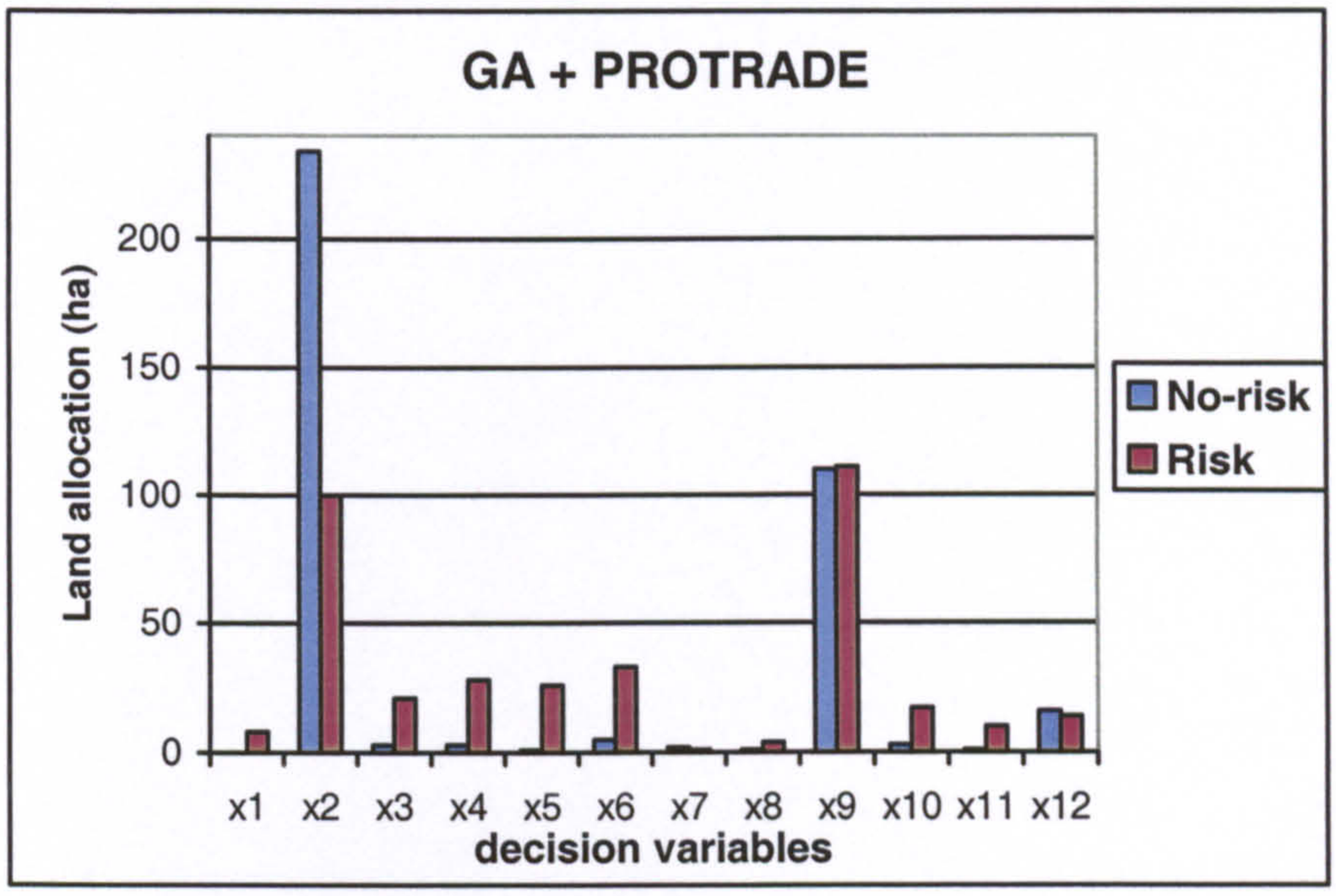


Figure 5.11 Land allocation comparison between no-risk and risk models (preferences)

The values of the decision variables are used to generate the goal vector $\mathbf{G_2}$.

$$\mathbf{G}_2 = \begin{bmatrix} G_1 \\ G_2 \\ G_3 \\ G_4 \\ G_5 \end{bmatrix}$$

The goal vector is generated for both no-risk and risk models and the results are presented in Table 5.7. Figure 5.12 is constructed using these values and compares the results of the models.

Goals	No-risk	Risk
G_1	0.819518	0.364766
G_2	0.051039	0.194940
G_3	0.460758	0.494646
G_4	0.994283	0.971271
G_5	0.856954	0.737199

Table 5.7 Goal vector $\mathbf{G_2}$ for no-risk and risk models

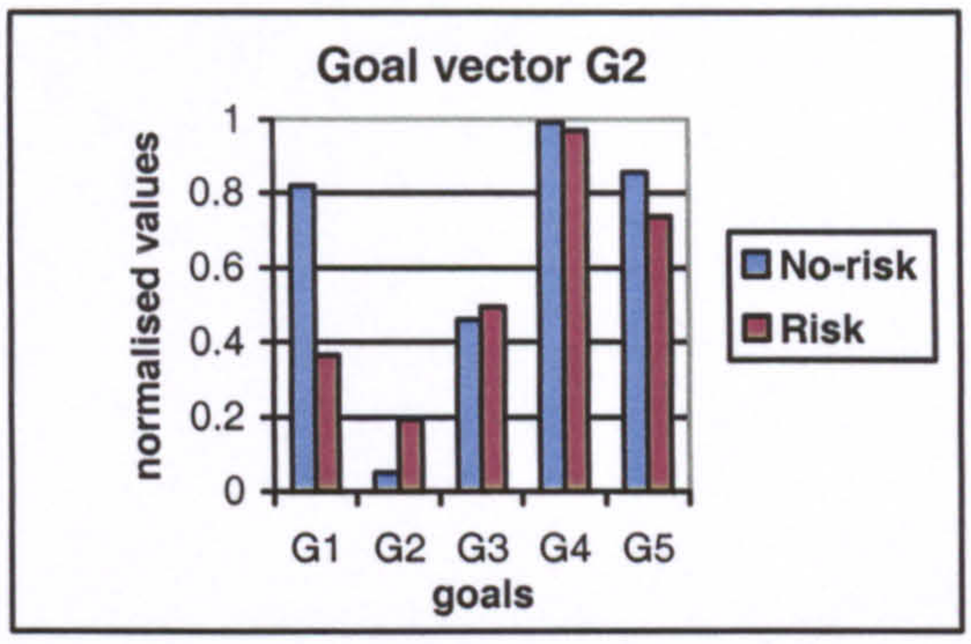


Figure 5.12 Comparison of the goal vector for no-risk and risk models (preferences)

From the goal vector it can be seen that in the no risk model the level of achievement for G_3 is 46.07%, G_1 is 81.95%, G_4 is 99.42%, G_2 is 5.10% and G_5 is 85.69% (Table 5.7). By contrast, in the risk model the level of achievement for G_3 is 49.46%, G_1 is 36.47%, G_4 is 97.12%, G_2 is 19.49% and G_5 is 73.71%.

After considering the DM's preferences the only goals that are increased are G_3 and G_2.

It is apparent that in the no risk model (Table 5.4), x_3 is the only practice that has no land allocated to it. The land allocated to x_5, x_7, x_8 and x_{11} decision variables is 3, 2, 2 and 3 hectares respectively. While x_1 (no reclamation programme) has the highest allocation, 194 hectares. In the risk model all the practices have land allocated and x_9 has the highest allocation of 109 hectares, and x_5, x_7, x_8 and x_{11} decision variables have 43, 3, 2 and 9 hectares respectively. This means for example that in the risk model the land allocation for practice x_5 (corn production) is increased more than 10 times. It is possible to conclude that when risk is considered, there is a drastic change in the allocations.

Comparing the goals' values in both no-risk and risk models with and without considering the DM's preferences yields Figures 5.13 and 5.14.

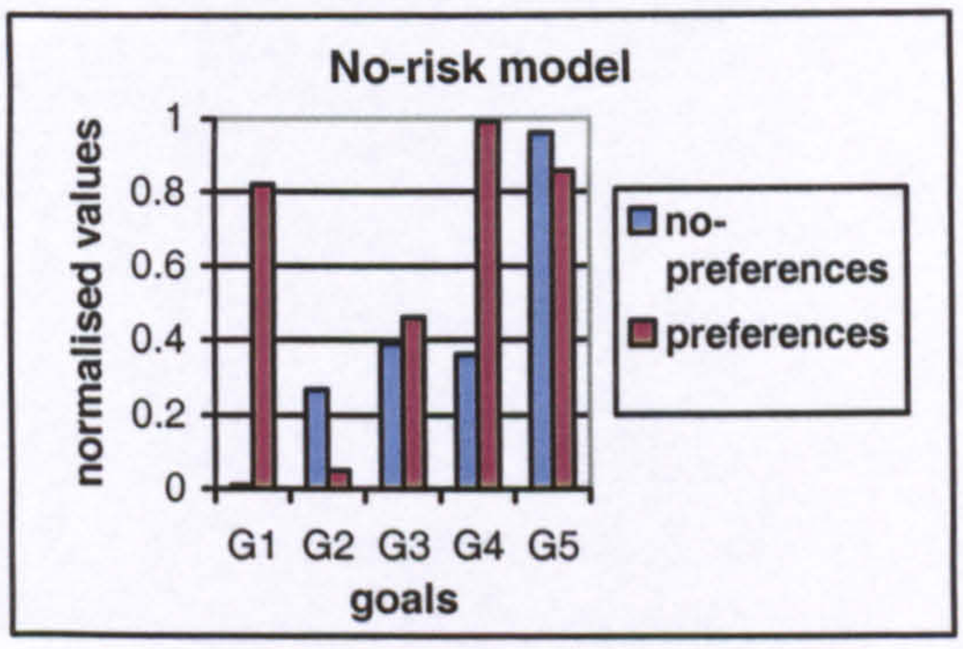


Figure 5.13 Comparison between $\mathbf{G_1}$ and $\mathbf{G_2}$ without risk

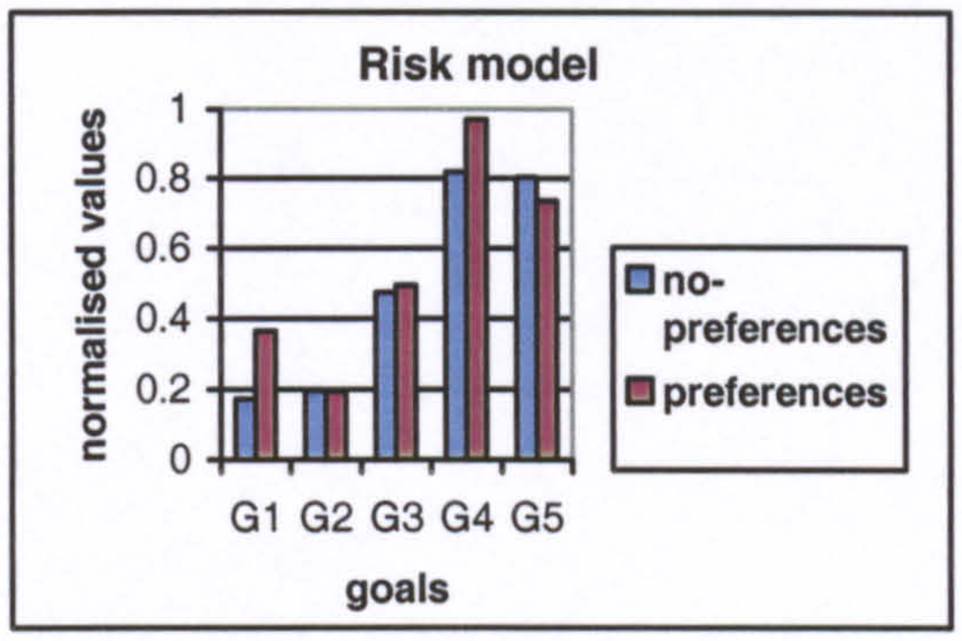


Figure 5.14 Comparison between $\mathbf{G_1}$ and $\mathbf{G_2}$ with risk

From Figure 5.13, the results indicate that the difference in the goals' values after considering the DM's preferences is considerably bigger, affecting the DM's decisions in a more dramatic way. While in Figure 5.14, it can be seen that the difference between the values without considering the DM's preferences is smoother, but it also affects the DM's decisions.

When the DM's preferences are considered, the levels of achievement also have a drastic change. For example, in the no risk model the level of achievement for goal two is 26.81% (Table 5.5), while, including the preferences, goal two's achievement level is 5.10% (Table 5.7). In the risk model with no-preferences goal one's achievement level is 17.18%, while including the preferences goal one's achievement level is 36.47%. With these comparisons it is possible to conclude firstly, that in terms of numerical results the differences between the no risk and risk model is considerable because the maximum value found for $F(\mathbf{x})$ in the no risk model is 1.270746 and in the risk model it is 0.825715. Secondly, in terms of level of achievement the PROTRADE method allows the DM to have a better control of the model and to know exactly what occurs in the algorithm. The inclusion of probabilities makes the model more dynamic in the sense that now it is possible to specify the probability of having a certain level of achievement in each goal according to the DM's requirements (preferences).

As mentioned in sub-section 5.6.1, in order to validate the results obtained by the GA and PROTRADE algorithm, it was decided to use @Risk and RiskOptimiser to solve the same case study.

The first set of optimisations is performed with the water constraint equal to 2,702,473 m^3 and the values of 0.5 and 0.15 are used for the crossover and mutation rates respectively. The decision variables were assigned probability distributions because they represent the area allocated to the different practices, which can vary from zero to three hundred and eighty hectares (0, 380 ha). These distributions were all Uniform Distributions (written as RiskUniform (min, max)), but with different ranges.

The population size was set to 80, being the same as that used in the GA. The results obtained are as follows:

Practices	**No risk (in Hectares)**	**Risk (in Hectares)**
x_1	165	99
x_2	91	32
x_3	0	16
x_4	7	16
x_5	0	19
x_6	32	57
x_7	2	1
x_8	0	9
x_9	61	62
x_{10}	5	0
x_{11}	0	63
x_{12}	18	7

Table 5.8 Results of the land allocation using RiskOptimiser

Practices	**GA-PROTRADE**				**RiskOptimiser**			
	No-risk		**Risk**		**No-risk**		**Risk**	
	(ha)	**%**	**(ha)**	**%**	**(ha)**	**%**	**(ha)**	**%**
x_1	194	51.05	69	18.16	165	43.42	99	26.05
x_2	3	0.789	45	11.84	91	23.95	32	8.42
x_3	0	0	25	6.579	0	0	16	4.21
x_4	4	1.052	24	6.316	7	1.842	16	4.21
x_5	3	0.789	43	11.32	0	0	19	5
x_6	51	13.42	26	6.842	32	8.42	57	15
x_7	2	0.526	3	0.789	2	0.526	1	0.263
x_8	2	0.526	2	0.526	0	0	9	2.368
x_9	89	23.42	109	28.68	61	16.05	62	16.32
x_{10}	10	2.631	9	2.368	5	1.316	0	0
x_{11}	3	0.789	9	2.368	0	0	63	16.58
x_{12}	18	4.737	15	3.947	18	4.737	7	1.842

Table 5.9 Results of the land allocation using GA-PROTRADE and RiskOptimiser

Table 5.9 contains the results of the GA-PROTRADE approach and compares them with the RiskOptimiser results. It also presents the percentile of land allocated to each decision variable with respect to the total land of 380 hectares for both no-risk and risk models. Additionally, Figures 5.15 and 5.16 are constructed using this table.

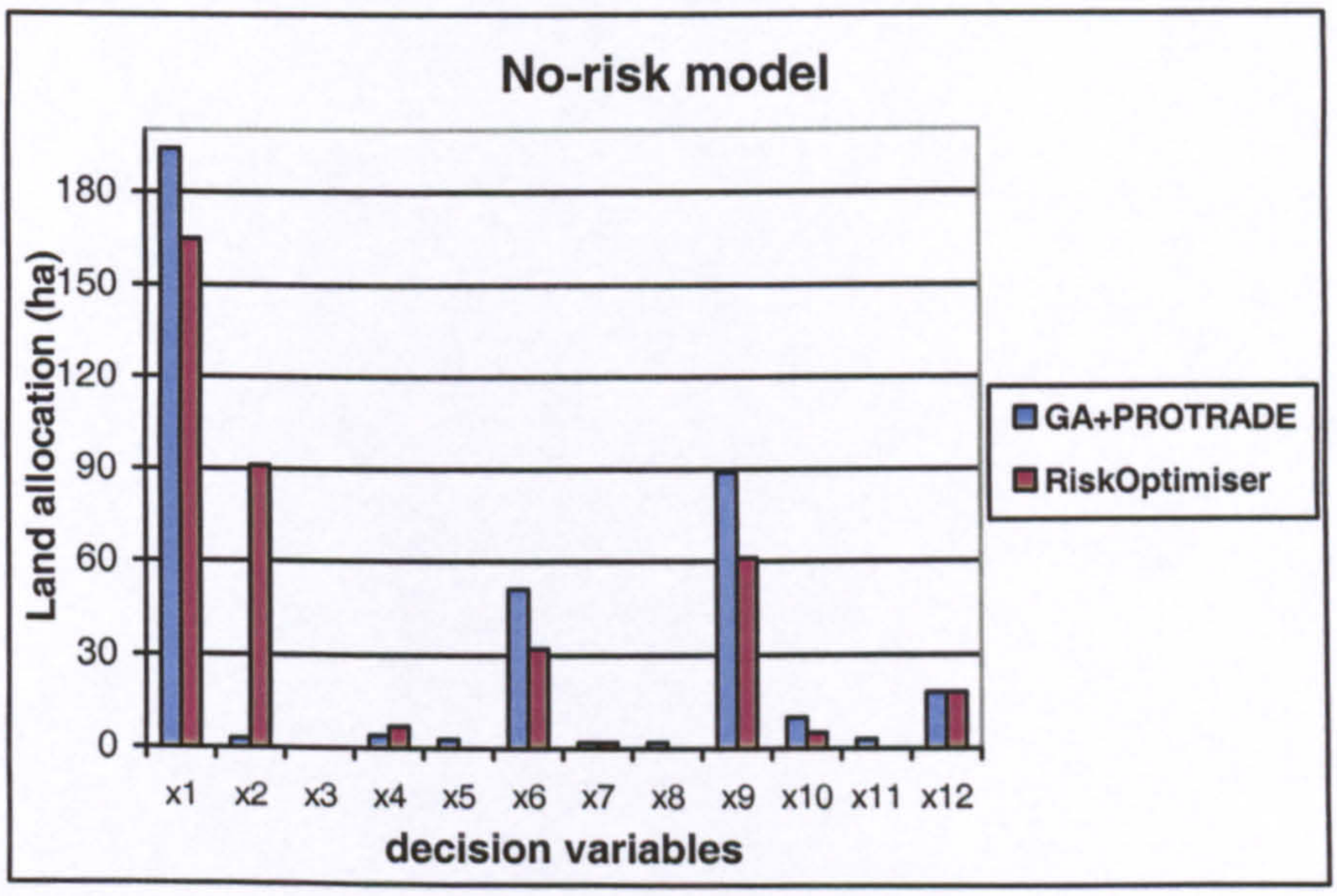


Figure 5.15 Land allocation comparison between GA-PROTRADE and RiskOptimiser

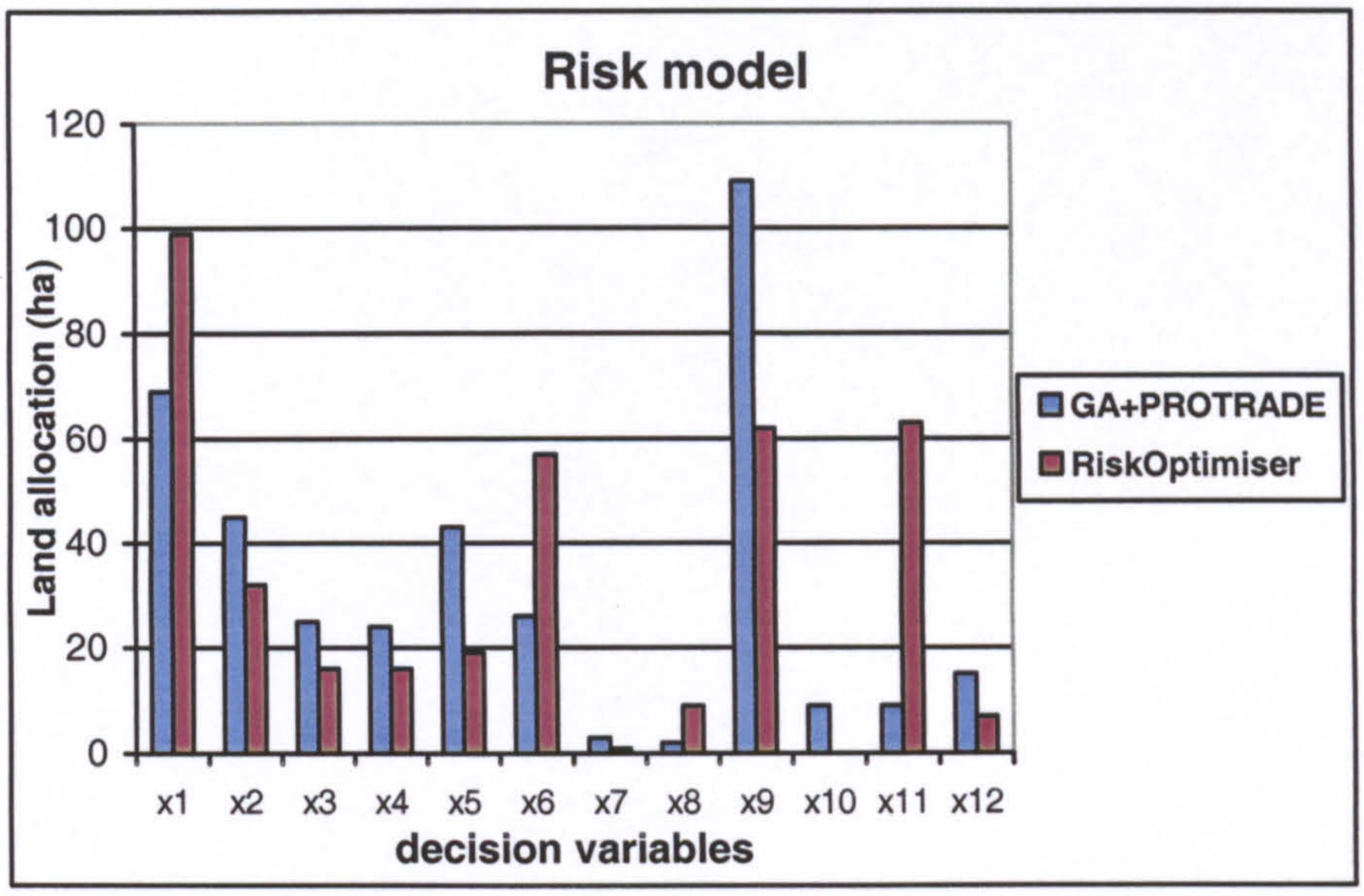


Figure 5.16 Land allocation comparison between GA-PROTRADE and RiskOptimiser

Table 5.9 and Figures 5.15 and 5.16 demonstrate that in the GA, when risk was not considered, x_3 is the only practice that has no land allocated to it for the first 2-year period. While in the @RISKOptimiser model, when risk was not considered, x_3, x_5, x_8 and x_{11} practices have no land allocated to them for the first 2-year period. The land allocated to x_5, x_8 and x_{11} practices in the GA is 3,2 and 3 hectares respectively; therefore it can be concluded that the GA-PROTRADE model allocates land along almost all decision variables. Moreover, x_1 (no reclamation programme) has the highest allocation in both models of 194 hectares in the GA and 165 hectares in the RISKOptimiser. In other words, the land allocated to x_1 in the no-risk model for the GA is 51% of the 380 hectares and for the RiskOptimiser there is 43.4% of the 380 hectares.

In the risk model, the GA-PROTRADE approach decision variable x_9 (Alfalfa production) has the highest allocation 28.68% of the 380 hectares, while in the RiskOptimiser the decision variable with the highest allocation is x_1 26%. In the GA all the decision variables have land allocated and in the RiskOptimiser just the decision variable x_{10} has no land allocated. In the RiskOptimiser no-risk model, the variables that have no land allocated are four (x_3, x_5, x_8 and x_{11}) while in the risk model there is one (x_{10}), then a conclusion can be made about the involvement of normal distribution in the model yield to more uniform allocation results. With these comparisons it is possible to conclude firstly, that although in terms of numerical results the GA and RiskOptimiser are different their tendency is similar when risk was not included. Secondly, when risk is considered not only there is a drastic change in some of the allocations but also the land allocation seems to be more uniform. This conclusion can be applied to both models. Therefore, generally when risk is considered the decisions taken will necessary be different.

In terms of allocation the results found with the GA-PROTRADE approach can be considered better from the point of view that all the decision variables are taken into account. Nevertheless, in terms of time performance RISKOptimiser is faster than the GA. Finally, the GA allows the programmer to have a better control of the model and to know exactly what occurs in the algorithm.

It is necessary to bear in mind that this kind of problem would be very difficult to solve without the aid of a GA because of its combinatorial nature, continuous values and the number of decision variables involved.

5.6.4 Risk Probability of Achievement Model

Following the PROTRADE method definition, once the new goal vector $\mathbf{G_2}$ is obtained, the following step (step 8) is taken to generate a vector $\mathbf{V_1}$ that expresses the trade-off between the goals' values and their probability of achievement. The risk probability of achievement model proposed in this thesis is outlined in Figure 5.17.

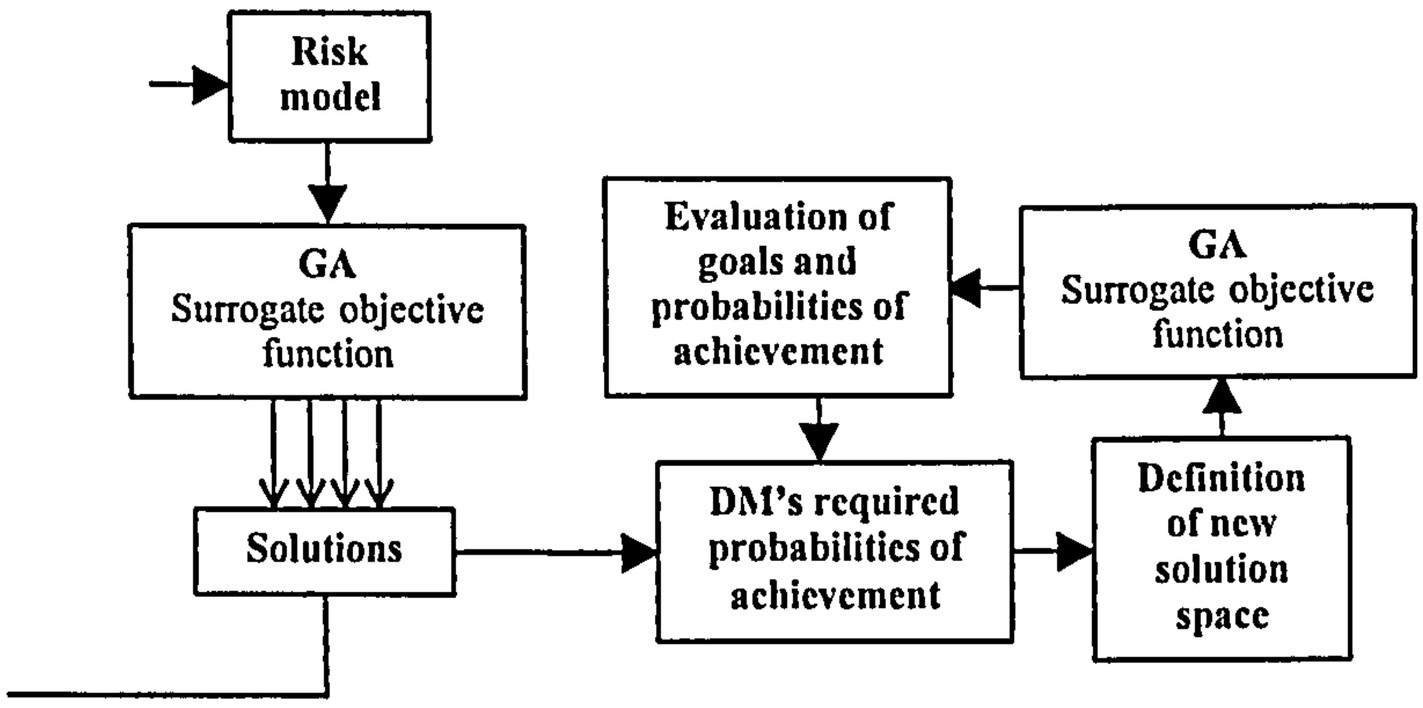


Figure 5.17 Risk probability of achievement model

It has been decided that in order to analyse the impact of including the probability of achievement into the model only the first 2-year sub-period will be considered.

Firstly the DM has to analyse the vector U_2 (equation 5.11). This vector has the objectives' values $[z_1(\mathbf{x}_2), z_2(\mathbf{x}_2), z_3(\mathbf{x}_2), z_4(\mathbf{x}_2), z_5(\mathbf{x}_2)]$:

$$\mathbf{U_2} = \begin{bmatrix} 14.39 \\ 140466.15 \\ 894373.45 \\ -182.91 \\ 52011.18 \end{bmatrix}$$

If the DM is not satisfied with the results obtained in the risk no-probability of achievement model (sub-section 5.6.3), it is necessary to generate vector $\mathbf{V_1}$ using $\mathbf{G_2}$ goal values of the first 2-year sub-period. The DM defines an initial probability for each goal. Initially, it is supposed that the probability of achievement $(1-\alpha_i)$ is equal to 0.5 and that it is the same for the five objectives, so the vector $\mathbf{V_1}$ is defined as follows:

$$\mathbf{V_1} = \begin{bmatrix} (0.364766, 0.5) \\ (0.194940, 0.5) \\ (0.494646, 0.5) \\ (0.971271, 0.5) \\ (0.737199, 0.5) \end{bmatrix}$$

From equation 5.13 the probability of achievement is read as follows:

$$prob[G_1 \geq 0.364766] \geq 0.5$$

In addition, the DM selects the pair $(G_i(\mathbf{x}_2), 1\text{-}\alpha_i)$ that is least satisfactory to him or her. According to the goal ranking proposed by the DM $G_3 > G_1 > G_4 > G_2 > G_5$, it is possible to see that G_3 is more important than G_1 and so on. Although the goal that is least likely to be achieved is G_2, the most important goal of all is G_3. Thus the DM has to specify new values for the pair $(G_3(\mathbf{x}_2), 1\text{-}\alpha_3)$. The DM wants an increment on G_3 from 0.494646 to 0.53 and with a probability of achievement of at least 60%. In order to define the new probability, the value from optimising (maximising) the individual objective function for goal three is needed. This value is obtained from equation 5.43 and is 1,808,033.08 kg. Then the probability is defined as follows:

$$prob[z_3(\mathbf{x}) \geq (0.53)(1{,}808{,}033.08)] \geq 0.60$$

The risk probability of achievement model is defined again using the coefficients' expected values (Appendix B) in the five objectives to be optimised.

If the DM's requirements (defined above) are considered, a new constraint has to be added, defining a new solution space $\mathbf{D}_2$ (Goicoechea et al., 1982):

$$x_1 + x_2 + \cdots + x_{12} \leq 380 \text{ ha}, \qquad \textit{Land}$$

$$q_1 x_1 + q_2 x_2 + \cdots + q_{12} x_{12} \leq 200{,}000 \text{ \$}, \qquad \textit{Capital}$$

$$w_1 x_1 + w_2 x_2 + \cdots + w_{12} x_{12} \leq 2{,}702{,}473 \text{ m}^3, \qquad \textit{Water}$$

$$\sum_{j=1}^{12} E(c_j) x_j + k_{\alpha 3}[\mathbf{x}^t \mathbf{A} \mathbf{x}]^{1/2} \geq (0.53)(1{,}808{,}033.08) \text{ new constraint}$$

where c_j represent the crop yield in kilograms/hectares with treatment j and $\mathbf{A}$ is the variance-covariance matrix given by:

$$\mathbf{A} = \begin{bmatrix} \text{var}(c_1) & \text{cov}(c_1, c_2) & \cdots & \text{cov}(c_1, c_{11}) & \text{cov}(c_1, c_{12}) \\ \text{cov}(c_2, c_1) & \text{var}(c_2) & \cdots & \text{cov}(c_2, c_{11}) & \text{cov}(c_2, c_{12}) \\ \vdots & \vdots & \cdots & \vdots & \vdots \\ \text{cov}(c_{11}, c_1) & \text{cov}(c_{11}, c_2) & \cdots & \text{var}(c_{11}) & \text{cov}(c_{11}, c_{12}) \\ \text{cov}(c_{12}, c_1) & \text{cov}(c_{12}, c_2) & \cdots & \text{cov}(c_{12}, c_{11}) & \text{var}(c_{12}) \end{bmatrix}$$

If it is assumed that c_i are independent random variables, then $\text{cov}(c_i, c_j) = 0$ for all $i \neq j$ and the variance-covariance matrix is diagonal.

From Goicoechea et al. (1982), to find $k_{\alpha 3}$ the standard normal distribution function $\Phi_{normal}(k_{\alpha 3}) = 1 - 0.60$ is used. From standard normal distribution tables, it is found that $k_{\alpha 3} =$ -0.255. The values of c_i and the variances are obtained from Table B.4 in Appendix B. Thus, redefining the new constraint:

$$\sum_{j=1}^{12} E(c_j) x_j - 0.255\left[\sum_{j=1}^{12} \text{var}(c_j) x_j^2\right]^{1/2} \geq (0.53)(1{,}808{,}033.08)$$

Then, $S_1(\mathbf{x})$ is maximised using $\mathbf{x} \in \mathbf{D}_2$ to find the vector $\mathbf{x}_3$. Where $S_1(\mathbf{x})$ is defined as in equation 5.46:

$$\max S_1(\mathbf{x}) = 2.372432\,G_1 + 1.554639\,G_2 + 1.856112\,G_3 + 1.207764\,G_4 + 1.098817\,G_5$$

In order to obtain vector $\mathbf{x}_3$, the GA is modified and run again using the new constraint:

$$\mathbf{x}_3 = \begin{bmatrix} 30 \\ 100 \\ 20 \\ 13 \\ 19 \\ 50 \\ 7 \\ 10 \\ 61 \\ 0 \\ 60 \\ 10 \end{bmatrix}$$

Once that $G_3 = 0.53$ has been achieved and with a minimum probability of 0.60, it is necessary to know the rest of the goals' values and their probabilities of achievement. The first goal to be analysed is G_1 (livestock) and the new constraint is defined as follows:

$$\sum_{j=1}^{12} E(l_j)\,x_j + k_{\alpha 1}\left[\sum_{j=1}^{12} \mathrm{var}(l_j)\,x_j^2\right]^{1/2} \geq 39.45\,G_1$$

where l_j represent the number of livestock heads (animal units per hectare) with treatment j. The values of l_j and the variances are obtained from Table B.1 in Appendix B, the values for x_2 and x_3 are taken from the vector $\mathbf{x}_3$, yielding the following equation:

$$13.75 + 5.506\,k_{\alpha 1} \geq 39.45\,G_1$$

where $G_1 = 0.364766$ is taken from the vector $\mathbf{V}_1$, solving the equation $k_{\alpha 1} \geq 0.11$ and from normal distribution tables $\Phi(0.11) = 0.54380 = \alpha_1$, then $1 - \alpha_1 = 0.4562$.

The second goal to be analysed is G_2 (run-off) and the new constraint is defined as follows:

$$\sum_{j=1}^{12} E(r_j)\,x_j + k_{\alpha 2}\left[\sum_{j=1}^{12} \mathrm{var}(r_j)\,x_j^2\right]^{1/2} \geq 720{,}439.2344\,G_2 + 118.9756$$

where r_j is the water run-off yield in cubic meters/hectare with treatment j. The values of r_j and the variances are obtained from Table B.3 in Appendix B, the values for x_j are taken from the vector $\mathbf{x}_3$, yielding the following equation:

$$132880 + 20727\,k_{\alpha 2} \geq 720{,}439.2344\,G_2 + 118.9756$$

where $G_2 = 0.194940$ is taken from the vector $\mathbf{V}_1$, solving the equation $k_{\alpha 2} \geq 0.37$ and from normal distribution tables $\Phi(0.37) = 0.64431 = \alpha_2$, then $1 - \alpha_2 = 0.3557$.

The third goal to be analysed is G_4 (sediment) and the new constraint is defined as follows:

$$-\sum_{j=1}^{12} E(s_j)x_j + k_{\alpha 4}\left[\sum_{j=1}^{12} \text{var}(s_j)x_j^2\right]^{1/2} \geq 6041.35\,G_4 - 6050.7$$

where s_j is the sediment yield in cubic meters/hectare with treatment j. The values of s_j and the variances are obtained from Table B.5 in Appendix B, the values for x_j are taken from the vector $\mathbf{x}_3$, yielding the following equation:

$$-650.63 + 619.71\,k_{\alpha 4} \leq 6041.45\,G_4 - 6050.7$$

where $G_4 = 0.971271$ is taken from the vector $\mathbf{V}_1$, solving the equation $k_{\alpha 4} \geq 0.75$ and from normal distribution tables $\Phi(0.75) = 0.77337 = \alpha_2$, then $1 - \alpha_2 = 0.2266$.

The last goal to be analysed is G_5 (fish) and the new constraint is defined as follows:

$$\sum_{j=1}^{12} E(f_j)x_j + k_{\alpha 5}\left[\sum_{j=1}^{12} \text{var}(f_j)x_j^2\right]^{1/2} \geq 70{,}552\,G_5$$

where f_j is the fish yield in kilograms/hectare with treatment j. The values of f_j and the variances are obtained from Table B.5 in Appendix B, the value for x_{12} are taken from the vector $\mathbf{x}_3$, yielding the following equation:

$$37410 + 20000\,k_{\alpha 5} \geq 70{,}552\,G_5$$

where $G_5 = 0.737199$ is taken from the vector $\mathbf{V}_1$, solving the equation $k_{\alpha 5} \geq 0.73$ and from normal distribution tables $\Phi(0.73) = 0.76730 = \alpha_2$, then $1 - \alpha_2 = 0.2327$.

$$\mathbf{V}_2 = \begin{bmatrix} (0.364766, 0.45) \\ (0.194940, 0.5) \\ (0.53, 0.6) \\ (0.971271, 0.23) \\ (0.737199, 0.23) \end{bmatrix}$$

From these results, it can be seen that when the probability of achievement of goal three was modified, the other goals' probabilities of achievement were also modified.

5.7 Summary

In this chapter the PROTRADE method has been presented as a tool used for the solution of multiple objective optimisation problems, considering the DM's preferences.

The main characteristics of PROTRADE are:

- It is an interactive multi-objective stochastic method
- It involves the DM's preferences.
- It handles risk and uncertainty.
- It is based on the formulation of surrogate functions.
- It allows the DM to visualise the goals function trade-off.
- It allows the DM's to control the goals' probability of achievement.

In General, PROTRADE allows the direct programming of the GA due to the goals' transformation into the F(**x**) and S(**x**) functions, using these as the fitness functions.

In this chapter a GA-PROTRADE approach was proposed creating a hybrid model that uses the strengths of GA and decision analysis techniques. Four original models were also developed, the aim of these models is to consider the DM's preferences in the problem solution as well as the DM's requirements related to the probability of achievement of each objective once it has been transformed into a goal. The four models were presented as follows:

1. No-risk no-probability of achievement.
2. No-risk probability of achievement.
3. Risk no-probability of achievement.
4. Risk probability of achievement.

Figure 5.18 shows the general model for the GA-PROTRADE approach.

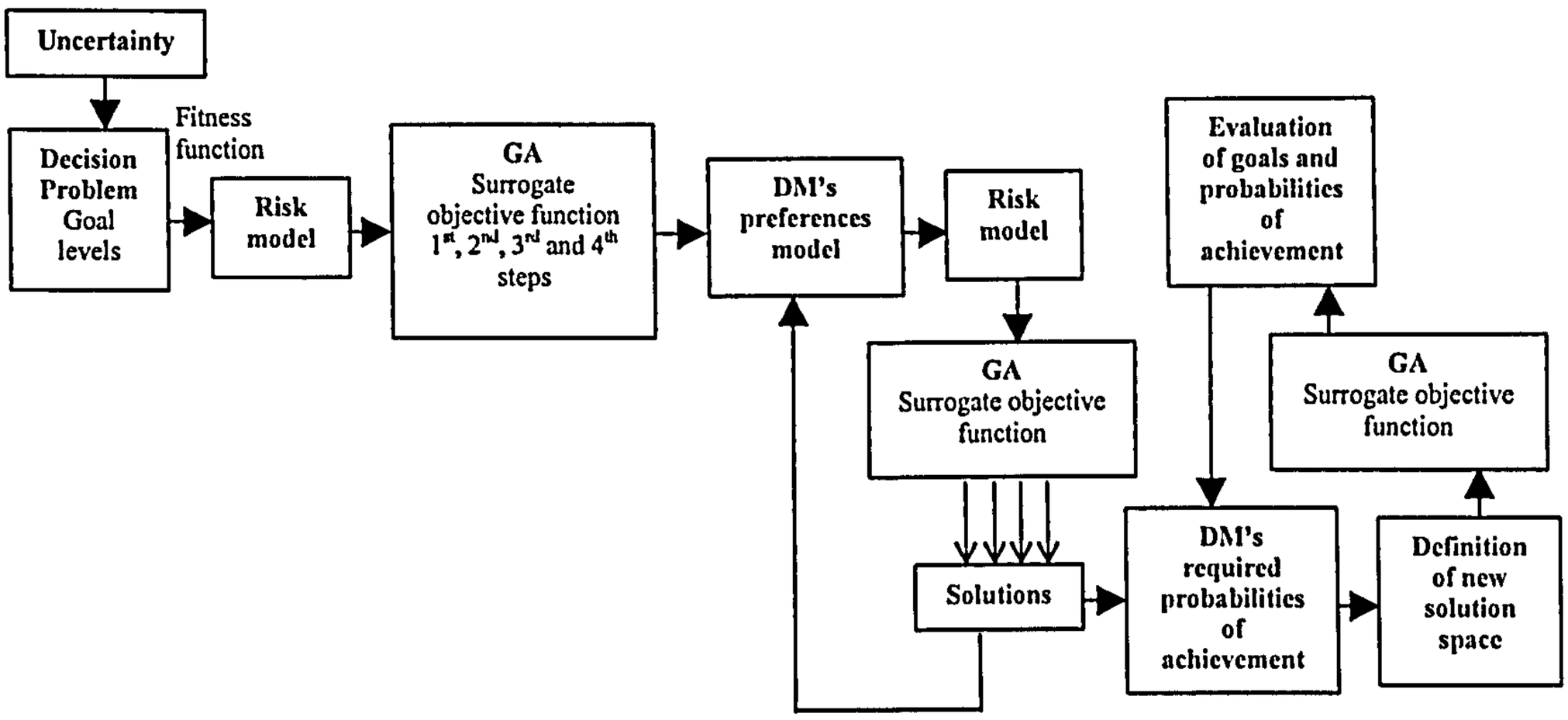


Figure 5.18 GA-PROTRADE approach

In order to demonstrate the applicability of the GA-PROTRADE approach the Black Mesa Region problem was solved. This problem is a case study solved by the authors of the PROTRADE method (Goicoechea et al., 1979). In the Northern Arizona on the Navajo Nation lands there was an area of 5,700 hectares to be strip-mined for coal in a 30 year-period, resulting in the development of a programme for designing and implementation of multiple land uses. The 30-year period was divided into 15 sub-periods each one 2 years long. Five objectives and twelve decision variables were considered. The initial definition of the problem also considered three constraints: land, capital and water. The water constraint consisted of two variables, the water consumption and the water available for each 2-year sub-period. The water available was determined through run-off practices and rainfall. Since rainfall is unpredictable, this value is a random variable, introducing uncertainty into the problem. Risk was also introduced on the decision variables through the use of a normal distribution. If the definition of this problem is analysed, it can be concluded that this case study is a very complex problem with a very complex solution.

This is a very interesting case study not only because it has 12 different decision variables but also because of the grade of uncertainty that is present and the risk analysis that can be performed. Two principal conclusions can be made; firstly that PROTRADE is a robust method that allows in a direct way the interaction between the algorithm and the DM preferences, giving it a major flexibility. Secondly, that without the GA the solution of this case study would have taken a very large amount of time. Due to the search capabilities of

GAs the solution time was decreased considerably showing the strengths of the GA-PROTRADE approach.

It was also shown that the use of a hybrid model gives the DM better control of the model's and the algorithms' outcomes. Moreover, the use of a GA makes the problem's solution more flexible and accurate, giving the programmer control of the genetic operators. The drastic changes occurred when the probability of achievement of the problem were changed, showing again that the control of the actions to follow is now in the DM's hands. Another conclusion that is evident is that the consideration of risk is very important to decision-making. The decisions made, based on models that did not include risk, were different to the decisions made when it was considered. It is also shown how the use of a real-valued GA can help in the direct implementation of the fitness function.

Finally, the most important conclusion is that the strongest point of the GA-PROTRADE approach is that it allows the programmer to manage goal levels, the DM's preferences and probabilities of achievement.

The application of this method for other Multiobjective Optimisation Problems constitutes an area for further study. An area to explore could be the use of a FRBS to automate the DM's preference model.

In the next chapter a method that combines the strengths of the GA-SEMOPS (Chapter Four) and the GA-PROTRADE approaches is proposed. This method is used for the solution of a resource allocation problem present in the Automatic Control and Systems Engineering Department at the University of Sheffield.

CHAPTER 6

Interactive Procedure for Multiple Objective Optimisation Problems (IPMOOP)

6.1 Introduction

In Chapters Four and Five, different approaches for the solution of MCDM problems were presented. The main objective of these models was the introduction of the DM's preferences in an interactive way. Primarily, the approaches consisted of a GA using an OR method for the solution of multiple objective optimisation problems. Some case studies were also solved in order to validate the good performance of the different proposals.

The objective of this chapter is to present a new model for the solution of MCDM problems which focuses on the decision-making procedure followed when a real life problem has to be solved rather than just the mathematical techniques used for the solution. The main objective of this model is to visualise the decision-making process as a continuous interaction between the DM and the analytic programmer (researcher). This model is named *interactive procedure for multiple objective optimisation problems* (IPMOOP) and is presented in section 6.2.

In order to determine the effectiveness of this proposal it was decided to solve a real life problem. The problem selected is a resource allocation problem present in the Automatic Control and Systems Engineering (ACSE) department at the University of Sheffield. The ACSE department was established in 1968 in response to industry needs. Nowadays, this department divides the academic staff activities into three areas: lecturing, research and administrative work. The overall objective of this problem is the optimisation of these activities. For instances it would be desirable to maximise the number of research activities and minimise the administrative work.

The solution of this problem is described in section 6.3. In subsections 6.3.1 and 6.3.2 respectively the initial problem is formulated and solved. In subsection 6.3.3 the data collection stage is described, in subsection 6.3.4 the final problem is formulated whilst in subsection 6.3.5 the final problem is solved.

The final problem's results are described in section 6.4. Finally, a chapter summary is presented in section 6.5.

6.2 Method Description

As stated by Churchman et al. (1957) there are two main actors in the solution of a decision problem: the DM and the researcher (analytic programmer). It is important to bear in mind that the DM could be one person or a group of people. The DM and the researcher have to be in continuous communication. Therefore, they will be considered as a unit called a *decision-making process group* (DMPG). The main purpose of this unit is to consider the DM and the

researcher as one entity that will work together throughout the solution of the problem (Figure 6.1).

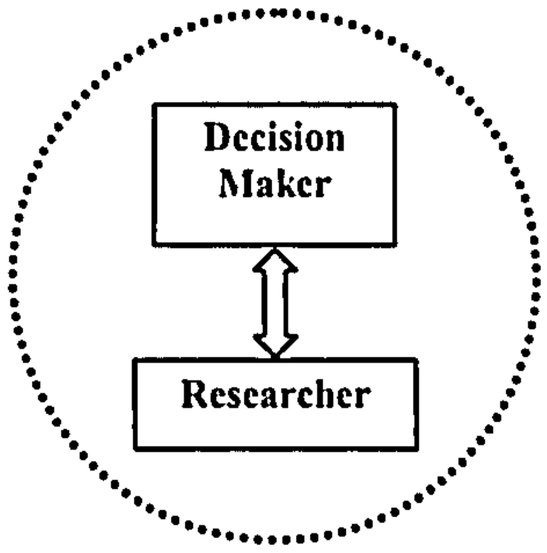


Figure 6.1 Decision-making process group (DMPG)

Saaty (1994) states that to make a decision it is necessary to have "knowledge, information and technical data" such as details about the problem, the people involved, objectives and policies, constraints and time horizons. Consequently, Figure 6.2 represents the general diagram of the IPMOOP proposed in this chapter. This procedure includes all the elements defined by Saaty (1994). From Figure 6.2, it can be seen that the first step is the DM identification. This is a vital aspect of the process because the DM is one of the most important elements in the application of this method. As mentioned before, this process is completely focused on the introduction of the DM's preferences and also the DM is part of the DMPG unit definition.

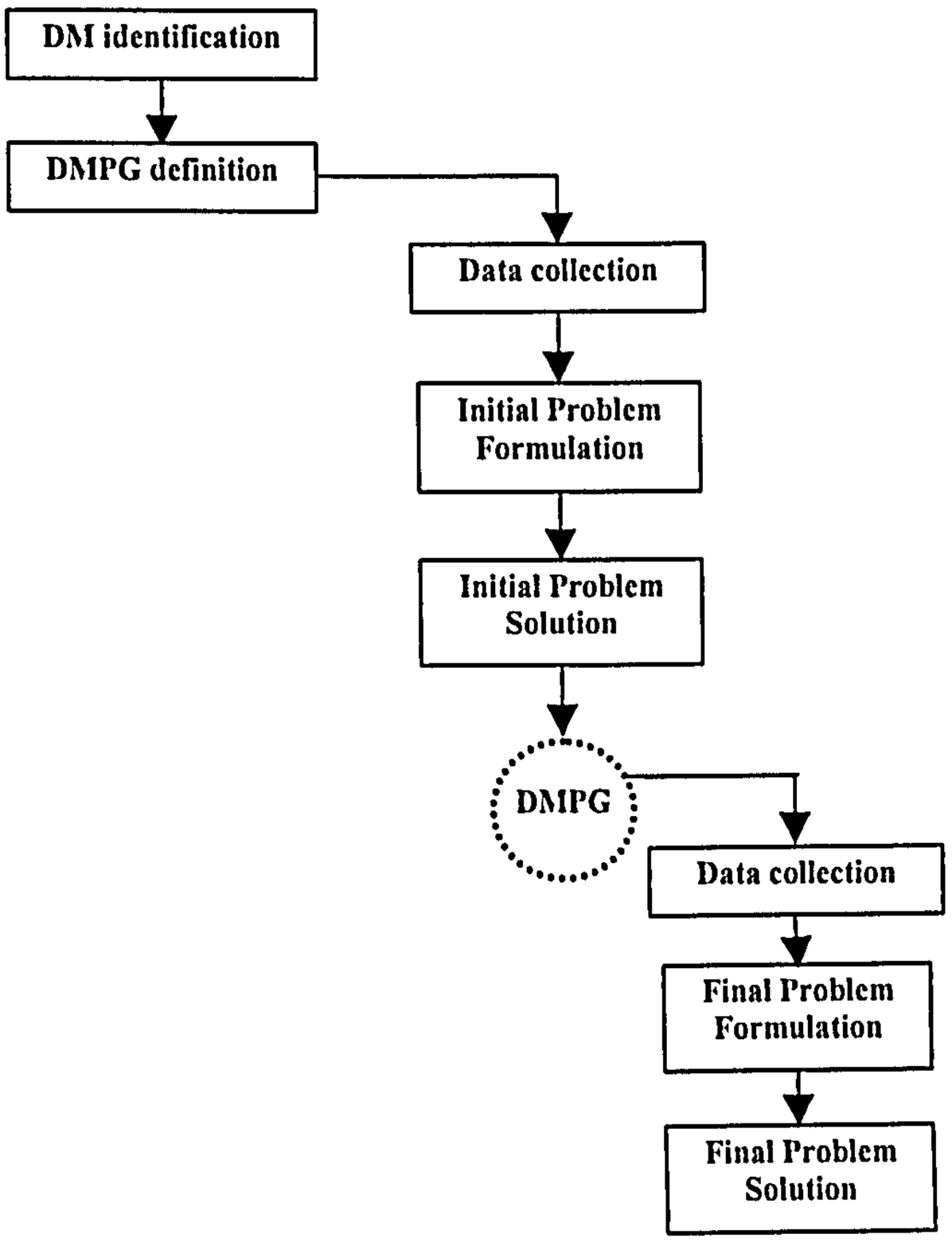


Figure 6.2 Interactive procedure for multiple objective optimisation problems (IPMOOP)

It is almost impossible to determine the nature of a problem as well as the final problem formulation from the first attempt. For this reason, it is necessary to have as a starting point an

initial problem formulation. Figure 6.3 outlines the elements of the initial problem formulation. From this figure, it can be seen that the model focuses mainly on quantitative data. It can be also seen that the DMPG unit appears twice during the process. The first time the unit is considered the aim is to define the objectives, the decision variables and the constraints of the initial problem. The second time the DMPG unit is considered, the researcher interviews the DM or the DM has to answer a questionnaire developed by the researcher. The main purpose of this questionnaire or interview is to gather as much quantitative information as possible from the DM to be used in the goals definition.

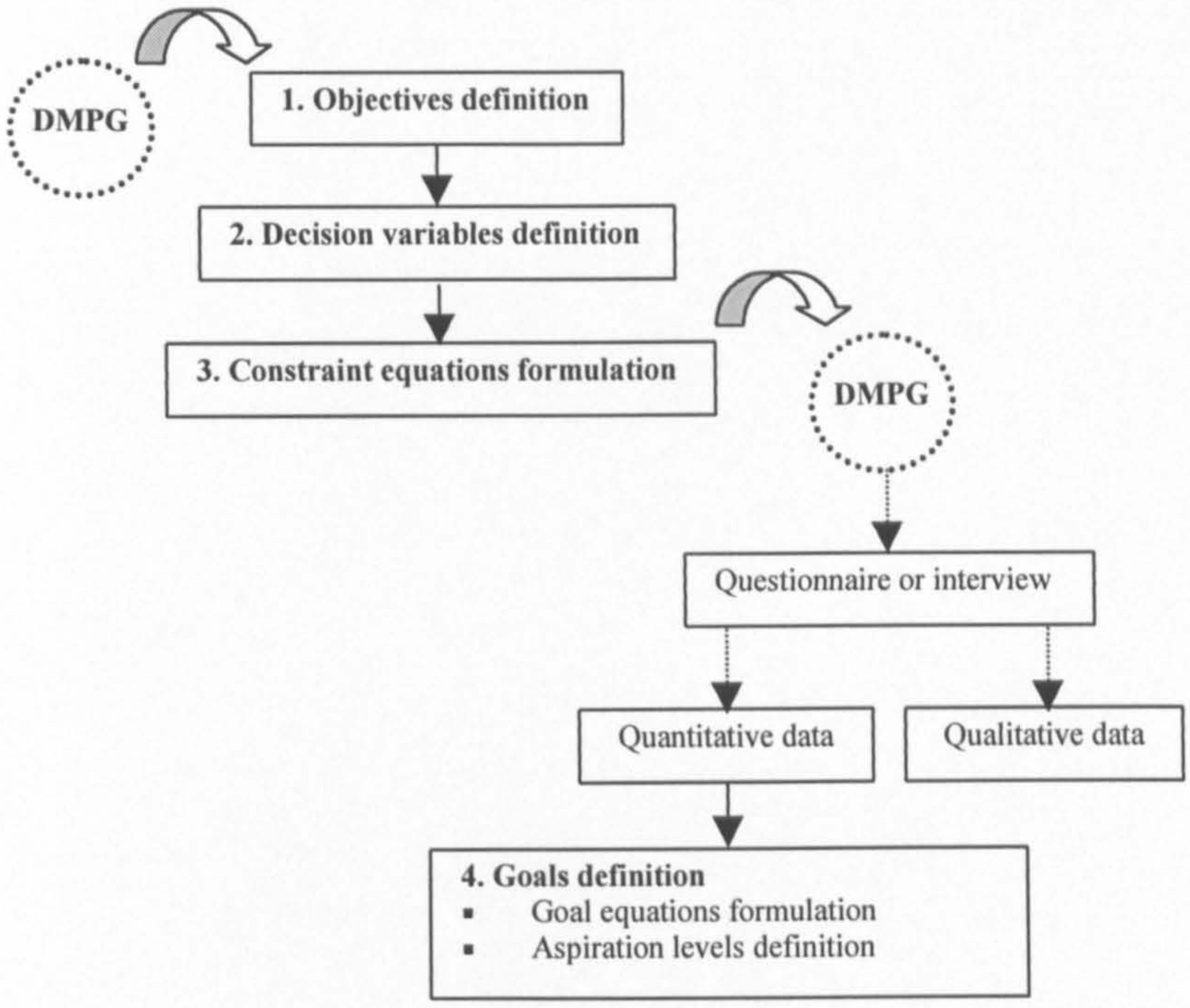


Figure 6.3 Initial Problem Formulation

Once the initial problem has been formulated, it has to be solved. The problem solution involves the determination of which analytic, numerical or simulation (Churchman et al. 1957) technique will be used. The selection of this technique depends on the problem's characteristics. It is desirable for this research purpose to use GA techniques for the solution of optimisation problems. Figure 6.4 is the general diagram for the problem solution. This process can be followed in both initial and final problem solutions.

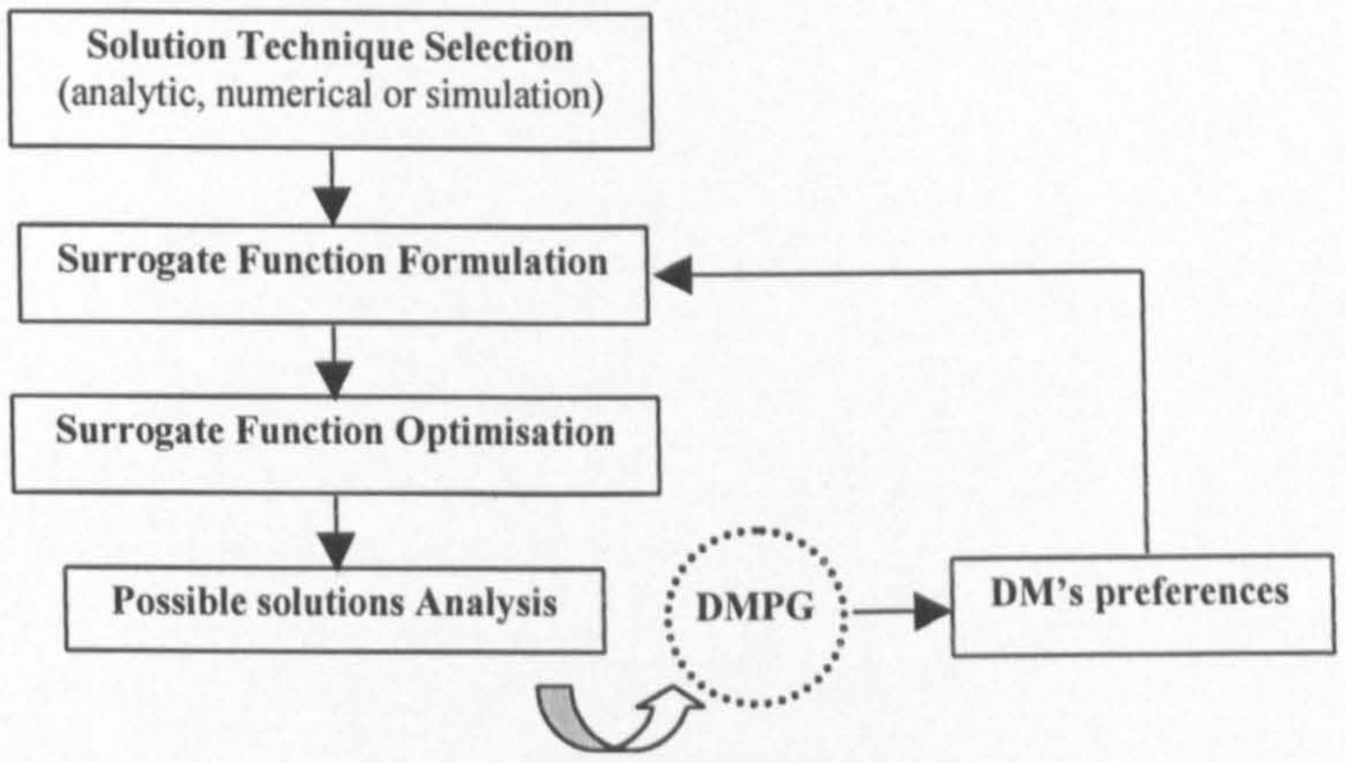


Figure 6.4 Problem Solution

In this research work two OR methods (SEMOPS and PROTRADE) have been combined with GAs. The common characteristic of both methods is that they use a surrogate function. Therefore, this model includes the use of a surrogate function that can be directly understood as the objective function and mapped as the GA's fitness function. Although the formulation of this surrogate function is different for each method, it is considered as a general step, called the *Surrogate Function Formulation* (Figure 6.4). The formulation of the surrogate function will be divided into two alternatives:

First alternative

This alternative is selected when sequential decisions are to be used. It is important to bear in mind that the ability to make sequential decisions is one of the strongest characteristics of the SEMOPS method. The surrogate function s used will be defined as in Chapter Four equation 4.5:

$$s = \sum_{i=1}^{P} d \tag{6.1}$$

where $d(\mathbf{x})$ function is defined after the goal levels $z_i(\mathbf{x})$ are compared to the aspiration levels AL_i having the following alternatives:

At most:

$$z_i(\mathbf{x}) \leq AL_i; \quad d_i = \frac{z_i(\mathbf{x})}{AL_i} \tag{6.2}$$

At least:

$$z_i(\mathbf{x}) \geq AL_i; \quad d_i = \frac{AL_i}{z_i(\mathbf{x})} \tag{6.3}$$

Equal:

$$z_i(\mathbf{x}) = AL_i; \quad d_i = \frac{1}{2}\left[\frac{AL_i}{z_i(\mathbf{x})} + \frac{z_i(\mathbf{x})}{AL_i}\right] \tag{6.4}$$

Within an interval:

$$AL_{iL} \leq z_i(\mathbf{x}) \leq AL_{iU} \tag{6.5}$$

$$d_i = \left[\frac{AL_{iU}}{AL_{iL} + AL_{iU}}\right]\left[\frac{AL_{iL}}{z_i(\mathbf{x})} + \frac{z_i(\mathbf{x})}{AL_{iU}}\right]$$

These equations are taken from the SEMOPS method (Monarchi et al., 1973) and were initially defined in Chapter Four equations 4.1 to 4.5.

Having defined the surrogate function, the next step is the surrogate function optimisation, in other words, its maximisation or minimisation.

Second alternative

The second alternative is selected when it is necessary to include risk in the solution of the problem. For this reason, it is necessary to calculate the objective coefficients' expected values.

A vector of objective functions is defined using the coefficients' expected values:

$$\begin{aligned} &\mathbf{z}(\mathbf{x}) = [z_1(\mathbf{x}), z_2(\mathbf{x}), \ldots, z_p(\mathbf{x})], \\ &g_q(\mathbf{x}) \le 0 \quad \textit{where} \quad q \in I[1, Q] \\ &\mathbf{x} > 0, \\ &z_i(\mathbf{x}) = \sum_{j=1}^{n} c_{ij} x_j, z_i(\mathbf{x}) = E[z_i(\mathbf{x})] \end{aligned} \tag{6.6}$$

where p is the number of objectives to optimise, q is the number of constraints, c_{ij} are the coefficients' expected values and n is the number of decision variables.

Then vectors $\mathbf{U}_1$ and $\mathbf{M}$ are defined as having the maximum and minimum values of the objective functions respectively. This means that vector $\mathbf{U}_1$ has the values of the maximisation of each objective separately (e.g. max $z_1(\mathbf{x})$, max $z_2(\mathbf{x})$, max $z_3(\mathbf{x})$, max $z_4(\mathbf{x})$, max $z_5(\mathbf{x})$), subject to constraints $g_q(\mathbf{x}) \le 0$. In addition vector $\mathbf{M}$ has the minimum values found following the same procedure, in other words, minimising each objective separately subject to the constraints.

$$\begin{aligned} &z_i(\mathbf{x}_i^*) = \max \quad z_i(\mathbf{x}), \qquad i \in I[1, p] \\ &\mathbf{U}_1 = \begin{bmatrix} z_1(\mathbf{x}_1^*) \\ z_2(\mathbf{x}_2^*) \\ \vdots \\ z_p(\mathbf{x}_q^*) \end{bmatrix} \\ &\mathbf{M} = \begin{bmatrix} z_{1\min} \\ z_{2\min} \\ \vdots \\ z_{3\min} \end{bmatrix} \end{aligned} \tag{6.7}$$

Finally, an initial surrogate function is formulated:

$$F(\mathbf{x}) = \sum_{i=1}^{p} G_i(\mathbf{x}) \tag{6.8}$$

where

$$G_i(\mathbf{x}) = \frac{z_i(\mathbf{x}) - z_{i\min}}{z_i(\mathbf{x}_i^*) - z_{i\min}} \tag{6.9}$$

where $z_i(x)$ is the value of objective function i (i = 1,2,..,n); z_{imin} is the minimum value obtained when objective i is subjected to the constraints; and $z_i(x^*)$ is the maximum value obtained when objective i is subjected to the constraints. Hitherto, each objective has been normalised using equation 6.9 and the surrogate function $F(\mathbf{x})$ has been defined as the addition of the normalised objectives. What is more, each objective can be maximised or minimised. In $F(\mathbf{x})$ there will be given a + sign for maximisation and a - sign for minimisation.

These equations are taken from the PROTRADE method (Goicoechea et al., 1979) and were initially defined in Chapter Five equations 5.3 to 5.6.

As defined in the first alternative, once the surrogate function is defined, the next step is its optimisation (maximisation or minimisation).

6.3 ACSE Problem Formulation

The ACSE department at the University of Sheffield was established in 1968 in response to industry needs. The ACSE department was graded 5* in the 2001 Research Assessment Exercise. Nowadays, this department provides a range of undergraduate and postgraduate degree courses, offering five separate courses, each of which can lead to either a Bachelor of Engineering (BEng) degree, or to a Master of Engineering (MEng) degree. The five courses are[3]:

1. Computer Systems Engineering. This course is intended for students who wish to gain the hardware and software skills and knowledge necessary to design and implement computer based engineering systems. These computer based systems range from simple domestic appliances to aero-engine controllers; from communications switching systems to process controllers on chemical plants; and from programmable logic controllers (PLCs) to transputer arrays.

2. Systems and Control Engineering. This course is intended for students who wish to gain a deep understanding of feedback control systems and their applications. Feedback control systems are routinely used in a wide range of engineering systems - electrical, electronic, mechanical, chemical, biotechnological and civil as well as in economics and in financial services.

3. Electronic, Control and Systems Engineering. This course is intended for students who wish to approach the analysis and design of sophisticated electronic systems from a systems philosophy rather than from an individual component level. This new strategy has lead to a growing demand for electronic, control and systems engineers from the telecommunications and electronics industry.

4. Mechanical Systems Engineering. This course is intended for students who wish to gain a degree in mechanical engineering, but this course is different to the others courses offered by mechanical engineering departments. The subject of mechanical engineering now embraces concepts such as advanced manufacturing systems, active control of vibration and robotics.

5. Medical Systems Engineering. The primary objective of the course is to provide students with the core skills of Systems Engineering and specialist skills in applications of engineering related to medical diagnostic and treatment procedures. The skills imparted to graduates from the course will give them the capability to take responsibility as Chartered Engineers for the creation, enhancement and maintenance of advanced modern engineering systems, and particularly those in medical applications.

In this department every semester or every year a resource allocation problem is solved. This problem consists of allocating different activities to the academic staff. The main purpose is to achieve the objectives of the university and the department in the best way possible. In this section, this problem using the IPMOOP is solved. It can be considered as an overall objective "to improve the activities allocation of the academic staff in the ACSE department".

Following the IPMOOP presented in Figure 6.2, the first step is the DM identification. For this problem the DM identified is the Assistant Head of Department of the ACSE. The second step of the procedure is the data collection. For this reason as much information as possible has been gathered about the ACSE department and its academic activities.

---

[3] **Taken from the University of Sheffield Web page**

The department counts with five professors, three readers, six senior lecturers and three lecturers. This makes a total of 17 members of the academic staff to be considered in the allocation. Each member of the academic staff has to work 37 hours a week.

The academic staff activities are divided into three areas:

(1) Research
(2) Lectures
(3) Administrative work

Lectures are divided into two areas: undergraduate and graduate. The graduate programmes are the MSc in Control Systems and the PhD research and taught programmes. It is important to bear in mind that the academic year is divided into two semesters: spring and autumn.

The next step to follow is the initial problem formulation (Figure 6.2), this will be described in the following subsection.

6.3.1 Initial Problem Formulation

In this subsection the process followed is the one outlined in Figure 6.3.

Objectives definition

The first objective is to maximise the overall number of research papers of the department. The second objective is to minimise the number of hours of administrative work for the department. It is important to bear in mind that this is just a first attempt for the problem formulation and it may appear to be very simplistic, but it will give both the DM and the researcher the information needed to define the final objectives.

Decision variables definition

Considering that there are three basic activities in the academic role (research, lecturing and administrative work), it is expected that each member of the academic staff will have a different activities allocation according to his or her skills and preferences. It is important to bear in mind that when the term lecturing is applied it refers to preparing the lecture, giving the lecture and assignments and exam marking. Although there are many ways of measuring the three different activities one direct way is as follows:

$$t_{ri} + t_{li} + t_{ai} = T_h \qquad \text{hours/week} \tag{6.10}$$

where $i = 1,\ldots, n$ and n is the number of members of academic staff, t_{ri} is the total of hours/week allocated to research activities to the i academic staff member, t_{li} is the total of hours/week allocated to lecturing activities to the i academic staff member, t_{ai} is the total of hours/week allocated to administrative activities to the i academic staff member and T_h is the total of hours/week that each academic staff has to work.

It has been decided to solve the problem considering the time that will take each academic staff to perform the task assigned on weekly basis.

Constraints equation formulation

The constraints are defined as follows:

1. Each academic staff member has to work a minimum of t_{lmin} hours lecturing.

2. Each academic staff member has to work a minimum of t_{rmin} hours doing research.
3. Each academic staff member has to work a minimum of t_{amin} hours of administrative work.

Interview applied to the DM:

1. How long does it take a member of the academic staff to write a paper?
2. How long does it take a member of the academic staff to prepare a lecture?
3. How long does it take a member of academic staff to mark assignments and exam papers?
4. How many administrative activities does a member of staff have?

Goals definition

Two kinds of goals are to be defined. Firstly, there are goals for each academic staff member, defined below:

1. To raise the number of hours/week doing research to at least 10.
2. To maintain the number of hours/week lecturing above 15.
3. To hold the number of hours/week doing administrative work below 2.

Secondly, there are goals for the total allocation, defined as follows:

1. To raise the number of hours/week doing research to at least 170.
2. To maintain the number of hours/week lecturing above 255.
3. To hold the number of hours/week doing administrative work below 34.

- *Goal equations formulation*

Goal 1 Hours of research ($z_1 \geq 170$)

$$z_1 = \sum_{i=1}^{n} t_{ri} \tag{6.11}$$

Goal 2 Hours of lecturing ($z_2 \geq 255$)

$$z_2 = \sum_{i=1}^{n} t_{li} \tag{6.12}$$

Goal 3 Hours of administrative work ($z_3 \leq 34$)

$$z_3 = \sum_{i=1}^{n} t_{ai} \tag{6.13}$$

- *Aspiration levels definition*

Two kinds of aspiration levels are to be used. Firstly, there are aspiration levels for each academic staff member. They are defined below:

$AL_1 = 20$ hrs/week (research)
$AL_2 = 25$ hrs/week (lecturing)
$AL_3 = 2$ hrs/week (administrative work)

It can be seen that if the aspiration levels are added the total hours/week that an academic staff member can have allocated is 47. This is because the aspiration levels will be changed every time the algorithm is performed and their values will try to converge in 37 hours/week.

Secondly, there are aspiration levels for the total allocation. They are defined as follows:

AL_{T1} = 340 hrs/week (research)
AL_{T2} = 425 hrs/week (lecturing)
AL_{T3} = 34 hrs/week (administrative work)

The following matrix will represent the results:

Academic staff member	Research	Lecturing	Administrative	Total
1	t_{r1}	t_{l1}	t_{a1}	T_h
2	t_{r2}	t_{l2}	t_{a2}	T_h
⋮	⋮	⋮	⋮	⋮
n	t_{rn}	t_{ln}	t_{an}	T_h
Total	T_{rh}	T_{lh}	T_{ah}	

Table 6.1 Weekly activities allocation

T_{rh} is the total number of hours/week allocated to research activities for the total number of academic staff, T_{lh} is the total number of hours/week allocated to lecturing activities for the total number of academic staff and T_{ah} is the total number of hours/week allocated to research activities for the total number of academic staff.

The next stage following the IPMOOP (Figure 6.2) is the initial problem solution. This will be described in subsection 6.3.2.

6.3.2 Initial Problem Solution

Solution Technique Selection

It has been decided to programme a binary GA with tournament selection and single-point crossover. The initial population size is 80 and the probabilities of crossover and mutation are 0.50 and 0.15 respectively.

Surrogate Function Formulation

To formulate the surrogate function two alternatives have been presented (section 6.2). For this problem solution, the alternative that handles sequential decisions has been selected. Following the formulations defined in equations 6.2 to 6.5, the function d is calculated as follows:

Goal 1 Hours of research ($z_1 \geq AL_{T1}$)

$$d_1 = \frac{AL_{T1}}{z_1} \tag{6.14}$$

Goal 2 Hours of lecturing ($z_2 \geq AL_{T2}$)

$$d_2 = \frac{AL_{T2}}{z_2} \tag{6.15}$$

Goal 3 Hours of administrative work ($z_3 \leq AL_{T3}$)

$$d_3 = \frac{z_3}{AL_{T3}} \tag{6.16}$$

The surrogate function is defined as follows:

$$s = d_1 + d_2 + d_3 \tag{6.17}$$

Surrogate Function Optimisation

After the surrogate function has been formulated, the GA will be performed minimising the surrogate function defined as follows:

$$\min s = d_1 + d_2 + d_3 \tag{6.18}$$

Possible Solutions Analysis

The solutions found are presented from Table 6.2 to Table 6.5:

Academic staff member	Research (hours)	Lecturing (hours)	Administrative work (hours)	Total Hours
1	20	15	2	37
2	19	17	1	37
3	18	17	2	37
4	10	26	1	37
5	15	20	2	37
6	15	20	2	37
7	15	21	1	37
8	15	21	1	37
9	16	19	2	37
10	21	15	1	37
11	20	16	1	37
12	11	25	1	37
13	20	16	1	37
14	17	19	1	37
15	16	20	1	37
16	16	20	1	37
17	10	26	1	37
	z_1	z_2	z_3	
Total hours	274	333	22	**Surrogate function *s***
	d_1	d_2	d_3	
	1.240876	1.276276	0.647059	3.164211

Table 6.2 Results with a surrogate function of 3.164211

Academic staff member	Research (hours)	Lecturing (hours)	Administrative work (hours)	Total hours
1	19	16	2	37
2	19	17	1	37
3	13	22	2	37
4	13	22	2	37
5	17	18	2	37
6	20	16	1	37
7	15	21	1	37
8	12	24	1	37
9	17	19	1	37
10	13	23	1	37
11	17	18	2	37
12	18	18	1	37
13	14	22	1	37
14	20	16	1	37
15	15	21	1	37
16	19	16	2	37
17	17	19	1	37
	z_1	z_2	z_3	
Total hours	278	328	23	
	d_1	d_2	d_3	
	1.223022	1.295732	0.676471	

Surrogate function s
3.195224

Table 6.3 Results with a surrogate function of 3.195224

Academic staff member	Research (hours)	Lecturing (hours)	Administrative work (hours)	Total hours
1	21	15	1	37
2	19	17	1	37
3	19	17	1	37
4	20	16	1	37
5	21	15	1	37
6	18	18	1	37
7	13	22	2	37
8	12	23	2	37
9	13	23	1	37
10	14	21	2	37
11	12	24	1	37
12	13	23	1	37
13	12	23	2	37
14	13	22	2	37
15	14	22	1	37
16	13	23	1	37
17	13	22	2	37
	z_1	z_2	z_3	
Total hours	260	346	23	
	d_1	d_2	d_3	
	1.307692	1.228324	0.676471	

Surrogate function s
3.212487

Table 6.4 Results with a surrogate function of 3.212487

Academic staff member	Research (hours)	Lecturing (hours)	Administrative work (hours)	Total hours
1	20	16	1	37
2	14	21	2	37
3	19	16	2	37
4	19	17	1	37
5	14	21	2	37
6	17	19	1	37
7	12	24	1	37
8	16	20	1	37
9	13	22	2	37
10	15	20	2	37
11	17	19	1	37
12	15	21	1	37
13	14	22	1	37
14	12	24	1	37
15	12	23	2	37
16	13	23	1	37
17	12	24	1	37
	z_1	z_2	z_3	
Total hours	254	352	23	Surrogate function s
	d_1	d_2	d_3	
	1.338583	1.207386	0.676471	3.222440

Table 6.5 Results with a surrogate function of 3.222440

From Tables 6.2 to 6.5, that the smallest surrogate function value found is 3.164211 (Table 6.2). This value corresponds to an allocation of 274 hours/week for research activities, 333 hours/week for lecturing activities and 22 hours/week for administrative work. This could be a good result if the DM decides that it is more important to allocate more hours to research than to lecturing. If the DM decides that the most important area is lecturing then the result that has a surrogate function value of 3.22244 (Table 6.5) is more appropriate, because the hours/week allocated to lecturing activities are 352 and the hours/week allocated to research activities are 254. Although the surrogate function value is the greater of the allocations presented. This is the point where the DM has to decide which solution satisfies his or her preferences.

It could be interesting to compare the results obtained with the goal values and the DM's aspiration levels. The results of this comparison are shown in Table 6.6.

The goal levels are:

$z_1 \geq 170$ hrs/week (research)
$z_2 \geq 255$ hrs/week (lecturing)
$z_3 \leq 34$ hrs/week (administrative work)

The aspiration levels are:

$AL_{T1} = 340$ hrs/week (research)
$AL_{T2} = 425$ hrs/week (lecturing)
$AL_{T3} = 34$ hrs/week (administrative work)

It is expedient not to consider the third goal (administrative work) because the objective is to minimise the number of hours/week allocated to administrative work activities and in all cases the objective is achieved.

Surrogate function value	Research z_1	Lecturing z_2	$AL_{T1} - z_1$	% from AL_{T1}	$AL_{T2} - z_2$	% from AL_{T2}
$s = 3.164211$	274	333	66	19.41	92	21.65
$s = 3.195224$	278	328	62	18.24	97	22.82
$s = 3.212487$	260	346	80	23.53	79	18.59
$s = 3.222440$	254	352	86	25.29	73	17.18

Table 6.6 Comparison between results found and aspiration levels

In Table 6.6 the percentage represents how far the value found is from the aspiration level. Consequently, it can be seen that the solution that maintains the research and lecturing allocation in a more uniform way is the one that has the minimum surrogate function value. It can be concluded that the objectives of the allocation are definitively in conflict because every time the hours/week allocated to research activities is modified it affects the hours/week allocated to lecturing activities.

As previously mentioned the decision-making process used for the solution of this problem is sequential, for this reason, the DMPG unit has to interact again. This unit will yield new aspiration levels defined by the DM and controlled by the researcher as follows:

Firstly, the aspiration levels for each academic staff member are:

$AL_1 = 18$ hrs/week (research)
$AL_2 = 25$ hrs/week (lecturing)
$AL_3 = 2$ hrs/week (administrative work)

Secondly, the aspiration levels for the total allocation are:

$AL_{T1} = 306$ hrs/week (research)
$AL_{T2} = 425$ hrs/week (lecturing)
$AL_{T3} = 34$ hrs/week (administrative work)

It can be seen that only one activity (research) has been modified. It is necessary to modify the GA and run it again. The solutions found are presented in Tables 6.7 to 6.10.

Academic staff member	Research (hours)	Lecturing (hours)	Administrative work (hours)	Total hours	
1	20	16	1	37	
2	11	25	1	37	
3	10	26	1	37	
4	16	20	1	37	
5	16	20	1	37	
6	12	24	1	37	
7	11	25	1	37	
8	12	23	2	37	
9	16	19	2	37	
10	12	24	1	37	
11	16	20	1	37	
12	11	25	1	37	
13	20	16	1	37	
14	13	23	1	37	
15	13	23	1	37	
16	20	15	2	37	
17	14	21	2	37	
	z_1	z_2	z_3		
Total hours	243	365	21		Surrogate function s
	d_1	d_2	d_3		
	1.259259	1.164384	0.617647		3.041290

Table 6.7 Results with a surrogate function of 3.041290

Academic staff member	Research (hours)	Lecturing (hours)	Administrative work (hours)	Total hours	
1	11	25	1	37	
2	16	20	1	37	
3	20	16	1	37	
4	18	18	1	37	
5	14	21	2	37	
6	14	21	2	37	
7	15	21	1	37	
8	20	16	1	37	
9	11	25	1	37	
10	12	24	1	37	
11	15	20	2	37	
12	13	22	2	37	
13	14	22	1	37	
14	17	19	1	37	
15	19	17	1	37	
16	20	16	1	37	
17	17	18	2	37	
	z_1	z_2	z_3		
Total hours	266	341	22		**Surrogate function s**
	d_1	d_2	d_3		
	1.150376	1.246334	0.647059		3.043769

Table 6.8 Results with a surrogate function of 3.043769

From Tables 6.7 and 6.8, it can be concluded that even though the surrogate function values are very close (3.041290 and 3.043769), the hours/week allocated to research and to lecturing vary considerably. In Table 6.7, the hours/week allocated to research and lecturing are 243 and 365 respectively while in Table 6.8 the hours/week allocated to research and lecturing are 266 and 341 respectively.

Academic staff member	Research (hours)	Lecturing (hours)	Administrative work (hours)	Total hours	
1	11	24	2	37	
2	12	24	1	37	
3	15	20	2	37	
4	15	20	2	37	
5	20	16	1	37	
6	20	16	1	37	
7	12	24	1	37	
8	17	18	2	37	
9	15	21	1	37	
10	10	26	1	37	
11	11	25	1	37	
12	13	23	1	37	
13	14	22	1	37	
14	11	25	1	37	
15	17	19	1	37	
16	11	25	1	37	
17	12	24	1	37	
	z_1	z_2	z_3		
Total hours	236	372	21		**Surrogate function s**
	d_1	d_2	d_3		
	1.296610	1.142473	0.617647		3.056730

Table 6.9 Results with a surrogate function of 3.056730

Academic staff member	Research (hours)	Lecturing (hours)	Administrative work (hours)	Total hours
1	18	18	1	37
2	14	22	1	37
3	20	15	2	37
4	16	20	1	37
5	16	20	1	37
6	16	20	1	37
7	20	16	1	37
8	19	16	2	37
9	15	20	2	37
10	15	21	1	37
11	12	24	1	37
12	18	18	1	37
13	17	18	2	37
14	12	23	2	37
15	16	19	2	37
16	20	16	1	37
17	20	16	1	37
	z_1	z_2	z_3	
Total hours	284	322	23	
	d_1	d_2	d_3	
	1.077465	1.319876	0.676471	

Surrogate function s
3.073811

Table 6.10 Results with a surrogate function of 3.073811

From the results presented in Tables 6.7 to 6.10, it can be seen that the values of d_3 function are smaller than 1 (0.617647, 0.647059, 0.617647 and 0.676471). In other words, the aspiration level for administrative work activities (AL_3) has a value much higher than those found for goal three (z_3). Consequently, the DMPG unit decided to change that aspiration level and run the programme once again.

Table 6.11 shows the comparison between the results found with an $AL_{T1} = 306$ hrs/week.

Surrogate function value	Research z_1	Lecturing z_2	$AL_{T1} - z_1$	% from AL_{T1}	$AL_{T2} - z_2$	% from AL_{T2}
$s = 3.041290$	243	365	63	20.59	60	14.12
$s = 3.043769$	266	341	40	13.07	84	19.76
$s = 3.056730$	236	372	70	22.88	53	12.47
$s = 3.073811$	284	322	22	7.19	103	24.23

Table 6.11 Comparison between results found and aspiration levels

As it was expected, the solution ($s = 3.073811$) that is closer to the research aspiration level (7.19%) is the one that is farthest to the lecturing aspiration level (24.23%). This occurs because it is not possible to allocate more than 37 hours a week.

The DMPG unit has decided to change the aspiration level for administrative work (AL_{T3}) from 34 to 25 hours/week. Therefore, the aspiration levels are set as follows:

$AL_{T1} = 306$ hrs/week (research)
$AL_{T2} = 425$ hrs/week (lecturing)
$AL_{T3} = 25$ hrs/week (administrative work)

The solutions found are presented in Tables 6.12 to 6.15.

Academic staff member	Research (hours)	Lecturing (hours)	Administrative work (hours)	Total hours
1	20	16	1	37
2	11	25	1	37
3	10	26	1	37
4	16	20	1	37
5	16	20	1	37
6	12	24	1	37
7	11	25	1	37
8	12	23	2	37
9	16	19	2	37
10	12	24	1	37
11	16	20	1	37
12	11	25	1	37
13	20	16	1	37
14	13	23	1	37
15	13	23	1	37
16	20	15	2	37
17	14	21	2	37
	z_1	z_2	z_3	
Total hours	243	365	21	**Surrogate function s**
	d_1	d_2	d_3	
	1.259259	1.164384	0.840000	3.263643

Table 6.12 Results with a surrogate function of 3.263643

The allocation found (Table 6.12) is the same allocation that was presented in Table 6.7 although the values of the surrogate function are different $s = 3.263643$ (Table 6.12) and $s = 3.041290$ (Table 6.7). The variation in the surrogate function values is due to the change of the aspiration level AL_{T3} from 34 to 25 hours a week.

Academic staff member	Research (hours)	Lecturing (hours)	Administrative work (hours)	Total hours
1	12	24	1	37
2	11	25	1	37
3	14	22	1	37
4	18	18	1	37
5	12	24	1	37
6	20	16	1	37
7	17	19	1	37
8	11	24	2	37
9	11	25	1	37
10	12	23	2	37
11	13	22	2	37
12	11	24	2	37
13	18	17	2	37
14	16	20	1	37
15	11	25	1	37
16	15	21	1	37
17	20	16	1	37
	z_1	z_2	z_3	
Total hours	242	365	22	**Surrogate function s**
	d_1	d_2	d_3	
	1.264463	1.164384	0.880000	3.308846

Table 6.13 Results with a surrogate function of 3.308846

Academic staff member	Research (hours)	Lecturing (hours)	Administrative work (hours)	Total hours
1	11	24	2	37
2	20	16	1	37
3	18	18	1	37
4	16	20	1	37
5	16	20	1	37
6	10	26	1	37
7	12	24	1	37
8	20	16	1	37
9	14	22	1	37
10	11	24	2	37
11	20	16	1	37
12	16	20	1	37
13	16	20	1	37
14	10	26	1	37
15	10	26	1	37
16	20	16	1	37
17	11	25	1	37
	z_1	z_2	z_3	
Total hours	251	359	19	
	d_1	d_2	d_3	
	1.219124	1.183844	0.760000	

Surrogate function s
3.162968

Table 6.14 Results with a surrogate function of 3.162968

Academic staff member	Research (hours)	Lecturing (hours)	Administrative work (hours)	Total hours
1	11	24	2	37
2	12	24	1	37
3	15	20	2	37
4	15	20	2	37
5	20	16	1	37
6	20	16	1	37
7	12	24	1	37
8	17	18	2	37
9	15	21	1	37
10	10	26	1	37
11	11	25	1	37
12	13	23	1	37
13	14	22	1	37
14	11	25	1	37
15	17	19	1	37
16	11	25	1	37
17	12	24	1	37
	z_1	z_2	z_3	
Total hours	236	372	21	
	d_1	d_2	d_3	
	1.296610	1.142473	0.840000	

Surrogate function s
3.279083

Table 6.15 Results with a surrogate function of 3.279083

The allocation found (Table 6.15) is the same allocation that was presented in Table 6.9 although the values of the surrogate function are different $s = 3.279083$ (Table 6.15) and $s = 3.056730$ (Table 6.9).

From these results, it can be concluded that the impact that goal three has in the overall allocation is very small due to the fact that the value allocated by the GA are 1 or 2 hours a week. Hence, the DMPG unit has decided to change the aspiration levels for research (AL_{T1}) and for lecturing (AL_{T2}). Therefore, the new aspiration levels are set as follows:

Firstly, the aspiration levels for each academic staff member are:

$AL_1 = 15$ hrs/week (research)
$AL_2 = 20$ hrs/week (lecturing)

Secondly, the aspiration levels for the total allocation are:

$AL_{T1} = 255$ hrs/week (research)
$AL_{T2} = 340$ hrs/week (lecturing)

The aspiration level for administrative work will remain the same $AL_{T3} = 25$ hrs/week.

The solutions found are shown in Tables 6.16 to 6.19.

Academic staff member	Research (hours)	Lecturing (hours)	Administrative work (hours)	Total hours	
1	11	24	2	37	
2	20	16	1	37	
3	18	18	1	37	
4	16	20	1	37	
5	16	20	1	37	
6	10	26	1	37	
7	12	24	1	37	
8	20	16	1	37	
9	14	22	1	37	
10	11	24	2	37	
11	20	16	1	37	
12	16	20	1	37	
13	16	20	1	37	
14	10	26	1	37	
15	10	26	1	37	
16	20	16	1	37	
17	11	25	1	37	
	z_1	z_2	z_3		
Total hours	251	359	19		Surrogate function s
	d_1	d_2	d_3		
	1.015936	0.947075	0.760000		2.723011

Table 6.16 Results with a surrogate function of 2.723011

Academic staff member	Research (hours)	Lecturing (hours)	Administrative work (hours)	Total hours
1	11	25	1	37
2	16	20	1	37
3	20	16	1	37
4	18	18	1	37
5	14	21	2	37
6	14	21	2	37
7	15	21	1	37
8	20	16	1	37
9	11	25	1	37
10	12	24	1	37
11	15	20	2	37
12	13	22	2	37
13	14	22	1	37
14	17	19	1	37
15	19	17	1	37
16	20	16	1	37
17	17	18	2	37
	z_1	z_2	z_3	
Total hours	266	341	22	**Surrogate function s**
	d_1	d_2	d_3	
	0.958647	0.997067	0.880000	2.835714

Table 6.17 Results with a surrogate function of 2.835714

Academic staff member	Research (hours)	Lecturing (hours)	Administrative work (hours)	Total hours
1	16	20	1	37
2	17	18	2	37
3	17	18	2	37
4	14	21	2	37
5	11	24	2	37
6	13	23	1	37
7	18	18	1	37
8	13	22	2	37
9	14	21	2	37
10	17	19	1	37
11	20	16	1	37
12	16	20	1	37
13	16	20	1	37
14	20	16	1	37
15	18	18	1	37
16	12	24	1	37
17	10	26	1	37
	z_1	z_2	z_3	
Total hours	262	344	23	**Surrogate function s**
	d_1	d_2	d_3	
	0.973282	0.988372	0.920000	2.881655

Table 6.18 Results with a surrogate function of 2.881655

Academic staff member	Research (hours)	Lecturing (hours)	Administrative work (hours)	Total hours
1	18	17	2	37
2	15	20	2	37
3	14	21	2	37
4	12	24	1	37
5	15	21	1	37
6	13	22	2	37
7	12	23	2	37
8	20	16	1	37
9	14	21	2	37
10	17	19	1	37
11	15	20	2	37
12	13	23	1	37
13	13	23	1	37
14	17	19	1	37
15	17	19	1	37
16	15	21	1	37
17	15	20	2	37
	z_1	z_2	z_3	
Total hours	255	349	25	
	d_1	d_2	d_3	
	1.000000	0.974212	1.000000	

Surrogate function s
2.974212

Table 6.19 Results with a surrogate function of 2.974212

From Tables 6.16 to 6.19, it can be seen that the allocation with the minimum surrogate function value (s = 2.723011) is the one presented in Table 6.16. However, this allocation does not meet all the aspiration levels, for instance, the number of hours/week allocated to research is 251 whilst the research aspiration level is 255 hrs/week. In the administrative work case, the allocated hours/week are 19 whilst the administrative work aspiration level is 25 hrs/week. Nevertheless, since the administrative work goal level is defined as $z_3 \leq 34$ hours a week, it can be seen that the allocation achieves the goal.

Table 6.20 shows the comparison between the allocation results found and the aspiration levels. It also outlines whether the aspiration levels are achieved or not.

Surrogate function value s	Research z_1	Lecturing z_2	Administrative work z_3	% from AL_{T1}	% from AL_{T2}	% from AL_{T3}
2.723011	251	359	19	1.57	-5.59	24
2.835714	266	341	22	-4.31	-0.29	12
2.881655	262	344	23	-2.75	-1.18	8
2.974212	255	349	25	0	-2.65	0

Table 6.20 Comparison between results found and aspiration levels

In Table 6.20 the percentage represents how far the value found is from the aspiration level, a + percentage signifies that the aspiration level was not achieved, a 0% signifies that it was achieved and a – percentage signifies that the aspiration level was achieved and moreover it was surpassed. From Table 6.20, it can be seen that the allocation that achieves the three aspiration levels (research, lecturing and administrative work) is the one that has a surrogate function value of 2.974212 (Table 6.19). Nevertheless, the solution when the surrogate function value is 2.835714 can be considered as satisfactory because its three d function values are less than one. This means that for the three activities the goals are achieved.

Finally, the DM has to decide which of the solutions found satisfies most of his or her preferences.

It can be concluded, that the IMOOP allows the DM to specify his or her preferences and have an assurance that they are considered for the solution of the problem. It also allows a complete interaction between the DM and the analytic programmer making possible to find high standard solutions.

As shown in Figure 6.2, once the initial problem has been solved the next steps to follow are data collection, final problem formulation and final problem solution. These are described in subsections 6.3.3, 6.3.4 and 6.3.5.

6.3.3 Data Collection

As previously stated, each academic staff member has to be considered as an individual with different preferences and skills. For this reason, it is necessary to formulate a final problem that considers other aspects of the problem outlined by the DM. The next step is data collection (Figure 6.2), where the objective is to gather as much information as possible. This information will be used to formulate the final problem.

The number of modules offered by the ACSE department a year is 30 for undergraduates and 16 for graduates, this makes a total of 46 modules. It is important to bear in mind that more than one lecturer gives some of these modules. Additionally, it is also important to take into consideration that the modules that involve the development of a project are not considered in the total number of modules because they require another kind of supervision rather than giving a lecture.

Table 6.21 presents the number of modules taught by each academic staff member. The modules are measured in terms of credits and one aspect to be aware of is that some modules do not have integer numbers. This occurs because some modules are given by more than one member of the academic staff and therefore the module's number of credits is divided by the number of academic staff members that are involved in it.

An interesting aspect to consider is the arithmetical mean of the total of credits because it can yield information about the average of credits taught in the ACSE department by each member of academic staff. The arithmetic mean in this case is 32.

In order to make the problem more understandable and easier to handle, the DMPG unit has decided to assign values from 1 to 3 to each member of academic staff to classify them in terms of the number of credits they taught.

The values assigned are as follows:

from 0 to 29 credits a value of 1 is assigned
from 30 to 39 credits a value of 2 is assigned
from 40 to 60 credits a value of 3 is assigned

In other words, an academic staff member who has been assigned a value of three is someone that dedicates most of his or her time on lecturing activities.

The total number of credits in the ACSE department is 545. This total number of credits can be used as a constraint in the final problem formulation

Academic staff member		Credits Semester 1	Credits Semester 2	TOTAL	Assigned values
1	Professor and Head of Department	5	5	10	1
2	Professor	24	15	39	2
3	Professor	24.17	8.75	32.92	2
4	Professor	30.83	20	50.83	3
5	Professor	0	0	0	1
6	Reader	22.5	10	32.5	2
7	Reader	3.33	50	53.33	3
8	Reader	14	23.75	37.75	2
9	Senior Lecturer	10	35	45	3
10	Senior Lecturer	32.5	0	32.5	2
11	Senior Lecturer	3.33	35.42	38.75	2
12	Senior Lecturer	24	25	49	3
13	Senior Lecturer	10.67	36.67	47.34	3
14	Senior Lecturer	20.67	13.75	34.42	2
15	Lecturer	0	20	20	1
16	Lecturer	5	6.67	11.67	1
17	Lecturer	10	0	10	1
	Total credits			**545**	

Table 6.21 Credits taught by each academic staff member

Once the information about the lecturing activities has been gathered, it is necessary to analyse the research activities. The term research activities will be understood as any activity related to research such as writing papers, attending conferences, research student supervision, and research projects.

The DM wants to allocate as many hours as possible to the research activities. For this reason, it has been decided to assign a value of 3 for the research activities of each academic staff member. This means, that each academic staff member is expected to have high levels in the research activity allocation in order to maintain high standards in the ACSE department research activities. The research activities will be measured in hours a week.

Table 6.22 presents the administrative roles of the ACSE department. From this table, it can be seen that the number of administrative work activities in the department is 38. The DM has been asked to evaluate each activity using a scale from 1 to 5, where 1 represents activities that require low performance time and 5 represent activities that the require high performance time. Consequently, the administrative work activities will be measure in points. The total of points assigned to the 38 different activities is 80.

Table 6.23 shows the total of points of administrative work activities assigned to each academic staff member. As in the research activities, the DM has been asked to assign a value from 1 to 3 to each academic staff member. The DMPG unit has decided to assign the values using the following rules:

from 0 to 3 points a value of 1 is assigned
from 4 to 6 points a value of 2 is assigned
from 7 to 15 points a value of 3 is assigned

Administrative roles	Evaluation
Chair, Health and Safety Committee	1
Chair, Research Committee	1
Demonstrator Co-ordinator	1
RTP Co-ordinator	1
Chair, Strategy Committee	1
Subject Review Co-ordinator	1
Chair, Student Affairs Committee	1
ERASMUS/SOCRATES Co-ordinator	1
Careers Co-ordinator	1
Assistant Chair, Teaching and Learning Committee	1
Library Co-ordinator	1
Deputy UUG Admissions Tutor	1
Schools Liaison Officer	1
Seminar Co-ordinator	1
Schools Liaison Officer	1
Chair, Executive Committee	2
PG(Taught) Admissions Tutor	2
Head of Web Team	2
Quality Assurance Co-ordinator	2
Chair, Publicity Committee	2
BEng Programme Leader	2
Chair, Teaching and Learning Committee	2
PGR Admissions Tutor	2
Industrial Liaison Co-ordinator	2
Assistant Examinations Officer	2
Chair, Policy Committee	3
Timetable Co-ordinator	3
Chair, Computing Committee	3
MTP/DLP Director	3
MSc Programme Leader	3
Project Co-ordinator (MSc & UG)	3
Examinations Officer	3
Aerospace Tutor/Admissions	3
PACT Director	3
MEng Programme Leader	4
UG Admissions Tutor	4
Assistant Head of Department	5
Head of Department	5
Total of points	**80**

Table 6.22 DM's evaluation of administrative work activities

Academic staff member		Total points	Assigned values
1	Professor and Head of Department	11	3
2	Professor	5	2
3	Professor	4	2
4	Professor	5	2
5	Professor	0	1
6	Reader	3	1
7	Reader	5	2
8	Reader	6	2
9	Senior Lecturer	4	2
10	Senior Lecturer	5	2
11	Senior Lecturer	5	2
12	Senior Lecturer	4	2
13	Senior Lecturer	9	3
14	Senior Lecturer	4	2
15	Lecturer	3	1
16	Lecturer	4	2
17	Lecturer	3	1

Table 6.23 Total of points assigned to each academic staff member (administrative work)

It is important to state that not all the information that has been gathered will be used for the formulation or solution of the problem.

6.3.4 Final Problem Formulation

In this subsection the process followed is the one outlined in Figure 6.3.

Objectives definition

The first objective is to maximise the overall number of research activities in the department. The second objective is to minimise the number of points of administrative work in the department. The third objective is to maximise the number of credits that each academic staff member gives. The overall objective of this problem is to maintain a high level of fairness in the allocation results for each academic staff member.

Decision variables definition

As in the initial problem formulation, three decision variables will be considered, each of them represent one of the three main activities (research, lecturing and administrative work) of the academic staff. The research activities are represented by the decision variable xr_i, measured in hours/week, where $i = 1, 2, 3,\ldots, n$ and n is the total number of academic staff members. The lecturing activities are represented by the decision variable xl_i, measured in credits/year, where $i = 1, 2, 3,\ldots, n$ and n is the total number of academic staff members. The administrative work activities are represented by the decision variable xa_i, measured in hours/week, where $i = 1, 2, 3,\ldots, n$ and n is the total number of academic staff members.

It is expected that each member of the academic staff will have a different activities allocation according to his or her skills and preferences.

Constraints equation formulation

The constraints are defined as follows:

1. Each academic staff member has to work a minimum of 10 hours a week doing research.
2. The total number of credits that the academic staff members have to teach is a minimum of 545.
3. The total number of points of administrative work for the academic staff members is a minimum of 80.

Goals definition

The goals defined are:

1. To raise the number of hours/week doing research to at least 170. This means at least 10 hours/week for each academic staff member.
2. To maintain the number of credits lecturing above 545.
3. To hold the number of points doing administrative work below 90.

- *Goal equations formulation*

Goal 1 Hours of research ($z_1 \geq 170$)

$$z_1 = \sum_{i=1}^{n} xr_i \tag{6.19}$$

Goal 2 Credits of lecturing ($z_2 \geq 545$)

$$z_2 = \sum_{i=1}^{n} xl_i \tag{6.20}$$

Goal 3 Points of administrative work ($z_3 \leq 90$)

$$z_3 = \sum_{i=1}^{n} xa_i \tag{6.21}$$

- *Aspiration levels definition*

The aspiration levels for each academic staff are defined as follows:

$$AL_1 = 20 \text{ hrs/week (research)}$$

In the lecturing case three aspiration levels have been assigned, each level correspond to the value assigned by the DMPG unit.

value = 1, $AL_2 \leq 19$ credits (lecturing)
value = 2, $20 \leq AL_2 \leq 39$ credits (lecturing)
value = 3, $AL_2 \geq 40$ credits (lecturing)

In the administrative work case three aspiration levels have been assigned, each level corresponds to the value assigned by the DMPG unit.

value = 1, $AL_3 \leq 3$ points (administrative work)
value = 2, $4 \leq AL_3 \leq 6$ points (administrative work)
value = 3, $AL_3 \geq 7$ points (administrative work)

6.3.5 Final Problem Solution

Table 6.24 presents the activity allocation in the ACSE department and the new activity allocation proposed by the DMPG unit in terms of assigned values form 1 to 3. The new activity allocation will be used as the programme's input data.

Academic staff Member	Assigned values (research)		Assigned values (lecturing)		Assigned values (administrative work)	
	Current	DMPG	Current	DMPG	Current	DMPG
1	3	3	1	1	3	3
2	3	3	2	2	2	2
3	3	3	2	2	2	2
4	3	3	3	3	2	1
5	3	3	1	2	1	1
6	3	3	2	2	1	1
7	3	3	3	3	2	1
8	3	3	2	2	2	2
9	3	3	3	3	2	1
10	3	3	2	2	2	2
11	3	3	2	2	2	2
12	3	3	3	3	2	1
13	3	3	3	2	3	3
14	3	3	2	2	2	2
15	3	3	1	3	1	1
16	3	3	1	1	2	2
17	3	3	1	2	1	2

Table 6.24 Activity allocation proposed by the DMPG unit

Solution Technique Selection

It has been decided to programme a binary GA with tournament selection and single-point crossover. The initial population size is 80 and the probabilities of crossover and mutation are 0.50 and 0.15 respectively.

Surrogate Function Formulation

To formulate the surrogate function two alternatives have been presented (section 6.2). For this problem solution, the alternative that handles sequential decisions has been selected. A surrogate function is defined for each academic staff member. Following the formulations defined in equations 6.2 to 6.5, the function d is calculated as follows:

Goal 1 Hours of research a week ($z_1 \geq AL_1$)

$$d_1 = \frac{AL_1}{z_1} = \frac{20}{z_1} \tag{6.22}$$

Goal 2 Credits of lecturing

$$\text{value}=1\ (z_2 \le AL_2);\ d_2=\frac{z_2}{AL_2}=\frac{z_2}{19}$$

$$\text{value}=2\ (AL_{L2} \le z_2 \le AL_{U2});\ d_2=\left[\frac{AL_{U2}}{AL_{L2}+AL_{U2}}\right]\left[\frac{AL_{L2}}{z_2}+\frac{z_2}{AL_{U2}}\right]=\left[\frac{39}{20+39}\right]\left[\frac{20}{z_2}+\frac{z_2}{39}\right]$$

$$\text{value}=3\ (z_2 \ge AL_2);\ d_2=\frac{AL_2}{z_2}=\frac{40}{z_2} \qquad (6.23)$$

Goal 3 Points of administrative work ($z_3 \le AL_{T3}$)

$$\text{value}=1\ (z_3 \le AL_3);\ d_3=\frac{z_3}{AL_3}=\frac{z_3}{3}$$

$$\text{value}=2\ (AL_{L3} \le z_3 \le AL_{U3});\ d_3=\left[\frac{AL_{U3}}{AL_{L3}+AL_{U3}}\right]\left[\frac{AL_{L3}}{z_3}+\frac{z_3}{AL_{U3}}\right]=\left[\frac{6}{4+6}\right]\left[\frac{4}{z_3}+\frac{z_3}{6}\right]$$

$$\text{value}=3\ (z_3 \ge AL_3);\ d_3=\frac{AL_3}{z_3}=\frac{40}{z_3} \qquad (6.24)$$

The surrogate function for each academic staff member is defined as follows:

$$s=d_1+d_2+d_3 \qquad (6.25)$$

Surrogate Function Optimisation

After the surrogate function has been formulated, the GA will be performed minimising the surrogate function defined as follows:

$$\min s=d_1+d_2+d_3 \qquad (6.26)$$

6.4 Experimental Results

In order to find the final problem's possible solutions, the GA has to be run minimising the surrogate function of each academic staff member. This is performed in such way to be able to handle each academic staff member individually. It is important to bear in mind that the goals are achieved if the values of the *d* functions are positive.

The solutions found after running the GA are presented in Table 6.25. From this table, it is possible to see that only in academic staff member number 1 the research aspiration level is not achieved. The rest of the allocation results are appropriate and achieve their correspondent aspiration levels. It can also be seen that the constraints such as "each academic staff member has to work a minimum of 10 hours/week doing research", " the total number of credits that the academic staff members have to teach is a minimum of 545" and "the total number of points of administrative work for the academic staff members is a minimum of 80" have been met. In other words, each academic staff member has been allocated with more than 10 hours/week for research activities, the total number of credits to teach is greater than 545 (598) and the number of points of administrative work are greater than 80 (82). Therefore, it could have been concluded that the allocation results are appropriate, however the fact that for

one of the academic staff members the research aspiration level is not achieved, makes the DMPG unit generate a new activity allocation.

Academic staff member	Research (hrs/week)	Lecturing (credits/year)	Administrative work (points)	d_1	d_2	d_3	s
1	19	7	15	1.052632	0.368421	0.466667	1.887719
2	20	29	5	1.000000	0.945942	0.980000	2.925942
3	20	36	4	1.000000	0.975897	1.000000	2.975897
4	20	49	2	1.000000	0.816327	0.666667	2.482993
5	22	27	1	0.909091	0.945812	0.333333	2.188236
6	22	30	2	0.909091	0.947692	0.666667	2.523450
7	20	59	2	1.000000	0.677966	0.666667	2.344633
8	23	27	4	0.869565	0.945812	1.000000	2.815377
9	20	57	1	1.000000	0.701754	0.333333	2.035088
10	24	37	6	0.833333	0.982911	1.000000	2.816244
11	20	28	6	1.000000	0.945275	1.000000	2.945275
12	24	57	3	0.833333	0.701754	1.000000	2.535088
13	24	37	14	0.833333	0.982911	0.500000	2.316244
14	22	27	4	0.909091	0.945812	1.000000	2.854903
15	26	52	3	0.769231	0.769231	1.000000	2.538462
16	23	8	4	0.869565	0.421053	1.000000	2.290618
17	27	31	6	0.740741	0.950422	1.000000	2.691163
Total	376	598	82				

Table 6.25 First allocation results

The DMPG unit has decided not to modify the whole allocation results; instead the allocation for member 1 has been calculated again, obtaining the following results:

Research activities 20 hrs/week
Lecturing activities 9 credits a year
Administrative work activities 13 points

It can be seen that in these allocation results all the individual aspiration levels are achieved. Therefore, the DMPG unit has decided to include it in the complete activity allocation. The results are presented in Table 6.26 and from them it can be seen that the constraints are met.

Academic staff member	Research (hrs/week)	Lecturing (credits/year)	Administrative work (points)	d_1	d_2	d_3	s
1	20	9	13	1.052632	0.368421	0.466667	1.887719
2	20	29	5	1.000000	0.945942	0.980000	2.925942
3	20	36	4	1.000000	0.975897	1.000000	2.975897
4	20	49	2	1.000000	0.816327	0.666667	2.482993
5	22	27	1	0.909091	0.945812	0.333333	2.188236
6	22	30	2	0.909091	0.947692	0.666667	2.523450
7	20	59	2	1.000000	0.677966	0.666667	2.344633
8	23	27	4	0.869565	0.945812	1.000000	2.815377
9	20	57	1	1.000000	0.701754	0.333333	2.035088
10	24	37	6	0.833333	0.982911	1.000000	2.816244
11	20	28	6	1.000000	0.945275	1.000000	2.945275
12	24	57	3	0.833333	0.701754	1.000000	2.535088
13	24	37	14	0.833333	0.982911	0.500000	2.316244
14	22	27	4	0.909091	0.945812	1.000000	2.854903
15	26	52	3	0.769231	0.769231	1.000000	2.538462
16	23	8	4	0.869565	0.421053	1.000000	2.290618
17	27	31	6	0.740741	0.950422	1.000000	2.691163
Total	377	600	80				

Table 6.26 Second allocation results

The DMPG unit decided to run the GA once more to see if it is possible to find an allocation that does not have to be modified. Table 6.27 presents the new allocation results.

Academic staff member	Research (hrs/week)	Lecturing (credits/year)	Administrative work (points)	d_1	d_2	d_3	s
1	23	8	11	0.869565	0.421053	0.636364	1.926981
2	27	30	5	0.740741	0.947692	0.980000	2.668433
3	23	27	5	0.869565	0.945812	0.980000	2.795377
4	23	47	3	0.869565	0.851064	1.000000	2.720629
5	20	29	2	1.000000	0.945942	0.666667	2.612608
6	23	27	2	0.869565	0.945812	0.666667	2.482044
7	23	47	2	0.869565	0.851064	0.666667	2.387296
8	20	33	5	1.000000	0.958462	0.980000	2.938462
9	22	47	1	0.909091	0.851064	0.333333	2.093488
10	20	33	5	1.000000	0.958462	0.980000	2.938462
11	26	31	6	0.769231	0.950422	1.000000	2.719653
12	24	57	3	0.833333	0.701754	1.000000	2.535088
13	26	31	13	0.769231	0.950422	0.538462	2.258114
14	20	38	5	1.000000	0.990445	0.980000	2.970445
15	26	52	3	0.769231	0.769231	1.000000	2.538462
16	27	12	6	0.740741	0.631579	1.000000	2.372320
17	22	30	5	0.909091	0.947692	0.980000	2.836783
Total	395	579	82				

Table 6.27 Third allocation results

From Table 6.27, it can be seen that all the allocation results are appropriate and achieve their correspondent aspiration level. It can also be concluded that the constraints such as "each academic staff member has to work a minimum of 10 hours/week doing research", " the total number of credits that the academic staff members have to teach is a minimum of 545" and "the total number of points of administrative work for the academic staff members is a minimum of 80" have been met. In other words, each academic staff member has been allocated with more than 10 hours/week for research activities, the total number of credits to teach is greater than 545 (579) and the number of points of administrative work are greater than 80 (82). Consequently, it can be concluded that the allocation results are appropriate.

Finally, the DM has to decide which of the solutions found satisfies most of his or her preferences.

The main conclusion, after applying the IMOOP for the solution of this activity allocation problem, is that this procedure is very effective in the solution of these kinds of optimisation problems and that the use of GAs allows the DMPG unit to find satisfactory solutions. It has been demonstrated that the IMOOP is capable of handling the interaction between the DM and the analytic programmer in a very smooth way. Finally, the use of hybrid approaches provides the necessary tools for the solution of multiple objective optimisation problems.

6.5 Summary

In this chapter the IPMOOP has been presented as a tool used for the solution of multiple objective optimisation problems. The main characteristics of this procedure are:

- The definition of a *decision-making process group* (DMPG) unit, where the DM and the analytic programmer actively interact throughout the solution of the problem.
- The formulation and solution of an initial problem that gives the (DMPG) information about the problem's nature and features.
- The use of surrogate objective functions to represent the different objectives.
- The DMPG unit is able to determine whether the decisions to be made are sequential or it is necessary to include risk analysis in the problem model.
- The use of different optimisation techniques in the solution of the problem.
- The formulation and solution of a final problem.

In General, IMOOP allows the direct programming of the GA due to the goals' transformation into the surrogate functions mapping them as the fitness functions.

In order to demonstrate the applicability of the IMOOP approach an activity allocation problem was solved. This problem is a resource allocation problem present in the Automatic Control and Systems Engineering (ACSE) department at the University of Sheffield. This problem consists of the allocation of academic staff members activities into three different areas: research, lecturing and administrative work.

An initial problem was formulated and solved. This problem consisted of the hours/week allocation for the three departmental activities for the 17 academic staff members of the ACSE department. The decisions were made sequentially, and the final solution was satisfactory to DM because it achieved all the goal and aspirations levels. It is important to state that all the constraints were met. The surrogate function optimisation was carried out using a binary GA with tournament selection and a single-point crossover. The GA's initial population was 80 and the probabilities of crossover and mutation were 0.50 and 0.15 respectively. As previously stated, the initial problem will give the DMPG unit information to pose a more general problem called final problem.

Once the initial problem has been solved, the next step is data collection to be used in the formulation and solution of the final problem. In this step, it is expected to gather as much information as possible.

The final problem formulation consisted of the definition of more general objectives. The allocation was divided into three areas: research, lecturing and administrative work activities. The units used to measure the goals were defined as follows:

Research (hours/week)
Lecturing (credits/year)
Administrative work (points)

Once again a surrogate function was defined in order to carry out the optimisation. This surrogate function allowed the GA to handle the different measure units. The solutions found were satisfactory from the DM's point of view and because they achieved all the goal and aspiration levels and met all the constraints.

After the solution of the ACSE resource allocation problem, it can be concluded that the IMOOP is effective in the solution of multiple objective optimisation problems. It was also demonstrated that this procedure (IMOOP) is capable of handling the interaction between the DM and the analytic programmer and that it allows the use of evolutionary computation techniques.

CHAPTER 7

Conclusions

7.1 Thesis Summary and Conclusions

The aim of this project was to modernise two interactive methods proposed in the 1970's and then refine them through contemporary areas of research such as genetic algorithms and fuzzy logic. These methods were selected because not only can they handle the DM's preferences in an interactive way but they also offer a direct way for the development of a hybrid approach using OR and evolutionary techniques. Furthermore, the introduction of risk in the problems' solution led to the development of an approach that can be applied in problems that involve uncertainty.

The initial chapters of this thesis gave an introduction to operations research (OR) as a process rather than a set of techniques to solve problems, and also presented the basic concepts of evolutionary computation. Moreover, the three areas of study of evolutionary computation: genetic algorithms, evolutionary programming and evolution strategies and their genetic operators were defined. As an example of the application of evolutionary computation to solve complex problems, the traveling salesman problem and the most important genetic operators to solve it were described. Additionally, decision theory was discussed as the basis for the solution of decision problems. Furthermore, in order to understand real life problems, where it is necessary to incorporate risk and uncertainty into the decision process, the risk analysis area was presented as well as some applications. It was also shown that several different methods for solving multiple criteria decision-making problems can be found in literature and that they differ by the type of problem they are designed to solve. Finally, the basic definitions of fuzzy sets, fuzzy rule-based systems, fuzzy controllers and defuzzyfication methods were discussed.

One of the objectives of this research was to demonstrate the use of GAs integrated with OR techniques. In order to achieve this goal, two methods with different features were selected. The first method is called *sequential multi-objective problem solving* (SEMOPS) and its main characteristics are that it allows the DM to set aspiration levels and be able to modify them sequentially. It also works with a surrogate function and a set of auxiliary problems. The second method is called *probabilistic trade-off development* (PROTRADE), its basic features are that it models the DM's preferences using mathematical functions such as multiple attribute utility functions, it is capable of handling risk and it is based on the formulation of surrogate functions. Although, these methods were proposed in 1973 and 1979 respectively, they were chosen because of the characteristics outlined above, but most importantly because both use progressive articulation of preferences information.

In Chapter Four two approaches were proposed, the first one (GA-SEMOPS) consisted of adding an evolutionary algorithm to the SEMOPS method. The principal objective of this approach was to handle the DM's preferences in an interactive way. The steps followed were: Firstly, the decision problem was defined and analysed, this step included the goals definition. Secondly, the DM defined the aspiration levels; these can change according to the DM. Thirdly, the surrogate objective function was formulated. Finally, the GA is applied. It is important to bear in mind that this process is performed $(n-1)$ times, where n is the number

of goals. It was interesting to work with this model because it is based on a surrogate objective function where different weights were assigned to each objective. The second approach was a fuzzy multiobjective genetic optimiser. This model is based on a fuzzy rule-based system (FRBS) in order to achieve an automated process emulating the DM. The kind of rules used were "IF <condition> THEN <action>" and the membership function had a linear shape within the interval [0,1].

The proposed GA-SEMOPS algorithm was applied in two case studies, namely Bow River Valley and nurse-scheduling problem.

From the results found in the Bow River Valley problem, it can be seen that this approach allows in a direct way the interaction between the algorithm and the DM's preferences. This problem had six goals, three decision variables and one constraint for the first iteration. Every time the algorithm was run the number of constraints was changed due to the DM's aspiration levels. This produced a reduction in the solution space making easier to find a satisfactory solution. It was decided to programme three real-valued GAs for solving this problem with the following features:

1. Tournament selection, arithmetic crossovers and a real-valued mutation operator.
2. Tournament selection, unfair crossover and a real-valued mutation operator.
3. Tournament selection, simulated binary crossover and a real-valued mutation operator.

The tournament size used was three. The results found by the three algorithms were compared and it was concluded that for the decision variables the values were almost the same. In other words, the algorithm was robust enough to yield the same results no matter which crossover operators were applied. Moreover, it was shown how the algorithm responds to the change of the DM's aspiration levels.

Additionally, the results were compared to those found by the authors of SEMOPS and it was concluded that they are very similar. Therefore, it was demonstrated that the SEMOPS by itself is good enough to work with any search technique.

As mentioned above, there were two approaches proposed in Chapter Four, the GA-SEMOPS and the fuzzy multiobjective genetic optimiser. In order to validate the second of these methods, it was decided to solve the Bow River Valley problem translating the DM's preferences as fuzzy rules. For each goal, two fuzzy sets were defined in terms of a deviation similar to the standard deviation but considering instead of the mean its aspiration level. The deviation was normalised in order to have a value between zero and one. Finally, the membership function was defined as linear for the six goals. Once the fuzzy sets and the membership functions were defined, it was possible to create the set of rules to be applied in the automated algorithm. The results from the fuzzy multiobjective genetic optimiser were 2% better than those found by the GA-SEMOPS approach.

In spite of these results, it was necessary to solve another case study to demonstrate that the GA-SEMOPS model can work with a different kind of optimisation problem. Hence, the multiple objective nurse-scheduling problem was considered. The objectives to optimise were the minimisation of hospital costs and the minimisation of the entire schedule fitness. The case study considered four hard and two soft constraints. The codification needed for the GA was binary and for this reason binary tournament selection was applied. For comparison reasons two crossovers were used: two-point and four-point. It was decided to run the programme for 100, 200 and 500 cycles in order to visualise the algorithm's convergence.

When considering the results found using the two-point crossover, in the 100 cycles instance all the hard constraints were met but the soft constraint "after a night shift a nurse prefers not to have a day shift" was not met for two nurses. In the 200 cycles instance, the hard

constraints were met again but the soft constraint " after a night shift a nurse prefers not to have a day shift" was not met for one nurse. Finally, in the 500 cycles instance, the hard and the soft constraints were met. Therefore, if the fitness values for the three cases are compared, it is possible to see that the performance of the algorithm depends on the number of cycles the programme has completed. Comparing the results found using the four-point crossover to those found using the two-point crossover, it was shown that in the 100 cycles instance the fitness value was considerably smaller to that found using the two-point crossover. It was also found that the soft constraint "after a night shift a nurse prefers not to have a day shift" was not met for one nurse while in the two-point crossover it was not met for two nurses. For the 200 cycles instance, the fitness value was smaller than that of the two-point crossover and all the constraints were met while in the two-point crossover the constraint was not met for one nurse. Finally, in the 500 cycles instance, all the constraints were met for both crossovers and the fitness value found using the four-point crossover was much smaller than the value found using the two-point crossover. Once again the results were very interesting, showing that although both algorithms had a good performance, the one that used the four-point crossover gave better results and performed faster.

Consequently, it has been demonstrated the effectiveness of combining the strengths of GA and OR techniques as they result in robust hybrid models for the solution of complex optimisation problems.

As discussed in Chapter Three, it is necessary to consider risk and uncertainty in the solution of most of optimisation problems. For this reason, it was decided to use the PROTRADE method to develop a new model for solving multiple criteria decision problems where risk and uncertainty are present. In Chapter Five, four models were developed that resulted in a general model called GA-PROTRADE. The principal objectives of this approach, besides handling the DM's preferences, including risk and uncertainty, were to model these preferences using a mathematic utility function and to set a probability of achievement for each goal.

In the first model (no-risk no-probability of achievement) the steps followed were:

1. Uncertainty analysis.
2. Goal levels and fitness function definition.
3. GA application.
4. DM's preference model definition.
5. GA application.
6. Solutions analysis.

For the second model (no-risk probability of achievement), it was necessary to have a possible solution where the probability of achievement for each goal is initially set with a value of 0.5. The steps followed were:

1. DM's required probabilities of achievement definition.
2. New solution space definition.
3. GA application.
4. Goals and probabilities of achievement evaluation.

The third model (risk no-probability of achievement) had the following steps:

1. Uncertainty analysis.
2. Goal levels and fitness function definition.
3. Risk model definition.
4. GA application.
5. DM's preference model definition.

6. Risk model definition.
7. GA application.
8. Solutions analysis.

For the fourth model (risk probability of achievement model) it was necessary to have a possible solution taken from the risk no probability of achievement model where the probability of achievement for each goal was initially set with a value of 0.5. The steps followed were:

1. DM's required probability of achievement definition.
2. New solution space definition.
3. GA application.
4. Goal and probabilities of achievement evaluation.

Hitherto, each model was described as an individual process although as mentioned above the objective was to generate a general model. This model (GA-PROTRADE) was defined making an integration of all four models, resulting on the following steps:

1. Uncertainty analysis.
2. Goal levels and fitness function definition.
3. Risk model definition.
4. GA application.
5. DM's preference model definition.
6. Risk model definition.
7. GA application.
8. Solution analysis.
9. DM's required probabilities of achievement definition.
10. New solution space definition.
11. GA application.
12. Goals and probabilities of achievement evaluation.

It is important to mention that the way the GA was programmed is modular in order to make it flexible to change each time it is applied. Every time the solution analysis step was applied it was expected that the DM and the programmer work together to determine whether the solution was acceptable or not.

The process proposed for the general model can be applied as many times as needed until the DM is satisfied with a solution. Once the model was defined, a case study was solved in order to validate it. The case study considered was a multiple use approach to land reclamation and management of the Black Mesa region in Arizona. This problem had twelve decision variables, five objectives and three constraints, showing the robustness of the GA-PROTRADE approach. It had a 30-year period divided into fifteen sub-periods each two years long. A real-valued GA was developed using tournament selection, with a tournament size of three, arithmetic crossover and real-valued mutation operators. It is important to bear in mind that many simulations were carried out to investigate the performance of the algorithm.

One of the most important conclusions was that the results vary in terms of uncertainty. In other words, as the water constraint depended on the amount of rain, an uncontrollable variable, the solutions would differ when the amount of water was less from those found when the amount of water was greater. It was also desirable to introduce risk in the solution of the case study; this was achieved introducing a normal probability distribution in the objective function coefficients as well as in the decision variables. From the results, it was concluded that when the normal distribution was considered the land allocation was more uniform. It was also concluded that the decisions made based on models that did not include

risk were different to the decisions made when risk was considered. Another aspect to discuss is that the probability of achievement can be modified for each goal, giving the DM even more control over the solution. This resulted in a new way of attacking a multiple objective optimisation problem because this approach (GA-PROTRADE) allows the researcher to handle goal levels, the DM' preferences and probabilities of achievement.

After analysing all the models proposed, it was decided to generate a new approach that combined the best characteristics of all of them. This model was called *interactive procedure for multiple objective optimisation problems* (IMOOP) and was focused on the solution of real life multiple objective optimisation problems and is presented in Chapter Six. The IMOOP's main characteristics are the definition of a unit called *decision-making process group* (DMPG), where the DM and the researcher continuously interact during the solution of the problem, the formulation and solution of an initial problem, the use of surrogate functions, the use of evolutionary computation in the solution of the problem and the formulation and solution of a final problem.

A real life problem was solved using the IMOOOP. This problem consisted of finding the best activity allocation for the academic staff members in the Automatic Control and Systems Engineering (ACSE) department at the University of Sheffield. The activities were divided into three areas: research, lecturing and administrative work. For the initial problem formulation, it was decided to measure the activities on weekly basis. Therefore, the results were presented in hours/week for each of the three departmental activities. A surrogate function was formulated with the features of a sequential decision process. A binary GA, validating once more the strengths and good performance of hybrid models, performed the optimisation. From the results, it can be concluded that although they were satisfactory for the DM, the problem was solved in a simplistic way and for this reason important information was left behind. Hence, the DMPG unit decided to gather as much information as possible related to the ACSE departmental activities. Using this information, it was possible to formulate and solve a final problem.

The final problem formulation consisted of the definition of more general objectives and the activities were measured as follows:

Research (hours/week)
Lecturing (credits/year)
Administrative work (points)

The surrogate function used a sequential decision process, allowing the DMPG unit to change the aspiration levels every time the algorithm was performed. The solutions found were satisfactory from the DM's point of view. It can be concluded that the formulation and solution of a final problem is one of the strongest characteristics of IMOOP, because in the ACSE department problem it gave more flexibility on the way the problem was treated. The allocation results found in the final problem allow the DM to visualise each academic staff member as an individual with different preferences and skills.

Finally, it is concluded that the IMOOP is effective in the solution of multiple objective optimisation problems because it is capable of handling the interaction between the DM and the researcher (DMPG unit). It considers that it is not always the first problem formulation that takes into account all the problem's characteristics, objectives and constraints. For this reason the use of a final problem makes the IMOOP a desirable tool for the solution of MCDM problems.

It was also noted that multiple criteria methods differ in the following ways: the timing of the decision-makers information, for example, prior versus progressive; the type of problem they are designed to solve, that is, whether multiple attribute or multiple objective; the type of

information required from the decision maker, for example, pairwise comparisons of alternatives; and the availability of supporting software.

7.2 Future work

In the GA-SEMOPS approach it would be interesting to analyse the behaviour of the GA using different real-valued crossovers such as geometrical crossover and simplex crossover.

In the case of the fuzzy multiobjective optimiser different membership functions could be implemented to develop the rules that represent the DM's preferences, although this may change the complexity of the FRBS. It would also be interesting to apply this model for the solution of other MCDM problems with progressive articulation of the DM's preferences, to confirm if the automation of the DM's preferences model is valid for other applications.

Another area of further study for the GA-PROTRADE approach is the development of a FRBS to implement the DM's preferences when the probability of achievement is considered, in other words, the creation of new rules that contain the DM's preferences related to the desired probability of achievement for each goal. The main purpose of this FRBS would be the automation of the goals' probabilities of achievement required by the DM.

Further research needs to be undertaken to find out whether the IPMOOP proposed could be used to solve different kinds of problems such as operations scheduling (e.g. energy conservation, fuel consumption in the delivering of electrical and thermal energy), water resource systems, planning and scheduling, robotics, engineering design, automotive and process control and automated highway systems. Another factor that can be considered is the inclusion of qualitative data into the problem formulation.

Finally, it is important to bear in mind that there are neither perfect methods nor perfect DMs. Therefore the solution of these kinds of problems involves the understanding of psychological, behavioural and cognitive aspects involved in real life decisions. Further research should be conducted to determine the influence of these aspects on the decision-making process.

APPENDIX A

Bow River Valley Case Study Results

A.1 Transfer coefficients and waste levels

	Bowville		Robin State Park		Plympton		State Line	
	C	N	C	N	C	N	C	N
Pierce-Hall Cannery	-5.86	—	-1.31	-3.15	-0.442	-0.771	-0.083	-0.073
Bowville	0	0	-2.18	-5.53	-0.764	-1.60	-0.0145	-0.162
Plympton	—	—	—	—	0	0	-3.49	-7.33

Table A.1 Carbonaceous (C) and nitrogenous (N) transfer coefficients
Adapted from Goicoechea (1982)

* All C and N values are time 10^{-5} (mg/l)(lb/day)

	Gross		After Preliminary Treatment	
	BOD_c	BOD_n	BOD_c	BOD_n
Pierce-Hall Cannery	40,000	28,000	28,000	19,000
Bowville	128,000	48,000	89,600	33,600
Plympton	95,700	35,700	67,000	25,000

Table A.2 BOD waste levels in pounds per day
Adapted from Goicoechea (1982)

A.2 Arithmetic crossover

A.2.1 Arithmetic crossover with final aspiration levels $z_1 = 6$, $z_2 = 4.9$, $z_3 = 6$, $z_4 = 6$, $z_5 = 1.8$, and $z_6 = 1.8$.

First generation, having the following aspiration levels:

A= (6.0, 6.0, 6.0, 6.5, 1.5, 1.5)

	Decision variables			Goals						const	Fitness
	x_1	x_2	x_3	z_1	z_2	z_3	z_4	z_5	z_6	q_4	S
Aspiration Levels				**≥6.0**	**≥6.0**	**≥6.0**	**≥6.5**	**≤1.5**	**≤1.5**		
Surrogate	0.782	0.609	0.841	5.845	3.513	5.595	6.729	0.375	1.809	3.547	6.229
Auxiliary 1	0.554	0.686	0.841	5.327	3.595	5.631	7.304	0.590	1.814	3.544	5.227
Auxiliary 2	0.739	0.581	0.853	5.748	3.346	5.542	6.905	0.316	1.977	3.631	4.597
Auxiliary 3	0.826	0.598	0.853	5.944	3.535	5.598	6.474	0.350	1.972	3.634	5.259
Auxiliary 4	0.734	0.569	0.851	5.736	3.291	5.524	6.924	0.292	1.947	3.615	5.448
Auxiliary 5	0.651	0.887	0.853	5.547	4.835	6.025	7.146	2.206	1.972	3.652	5.543
Auxiliary 6	0.835	0.632	0.883	5.965	3.689	5.649	6.407	0.430	2.500	3.872	4.996

Table A.3 Results using the goal values as aspiration levels

Second generation, having the following aspiration levels:

A= (6.0, 6.0, 6.0, 6.5, 1.5, 1.8)

	Decision variables			Goals					*const	const	Fitness
	x_1	x_2	x_3	z_1	z_2	z_3	z_4	z_5	z_6	q_4	S
Aspiration Levels				**≥6.0**	**≥6.0**	**≥6.0**	**≥6.5**	**≤1.5**	**≤1.8**		
Surrogate	0.822	0.676	0.836	5.935	3.852	5.703	6.503	0.555	1.748	3.519	4.990
Auxiliary 1	0.755	0.683	0.838	5.785	3.787	5.687	6.845	0.578	1.770	3.528	3.974
Auxiliary 2	0.758	0.362	0.838	5.791	2.597	5.288	6.834	0.041	1.778	3.519	3.149
Auxiliary 3	0.800	0.674	0.836	5.887	3.813	5.692	6.632	0.550	1.748	3.518	3.940
Auxiliary 4	0.958	0.584	0.836	6.246	3.794	5.663	4.081	0.321	1.748	3.524	3.816
Auxiliary 5	0.610	0.984	0.839	5.455	5.802	6.321	7.221	6.895	1.783	3.574	3.983

Table A.4 Results considering goal 6 as a constraint

* goal 6 acts as a constraint

Third generation, having the following aspiration levels:

A= (6.0, 4.9, 6.0, 6.5, 1.5, 1.8)

	Decision variables			Goal	*const	Goals			*const	const	Fitness
	x_1	x_2	x_3	z_1	z_2	z_3	z_4	z_5	z_6	q_4	S
Aspiration Levels				**≥6.0**	**≥4.9**	**≥6.0**	**≥6.5**	**≤1.5**	**≤1.8**		
Surrogate	0.870	0.859	0.834	6.046	4.927	6.044	6.077	1.757	1.727	3.528	4.226
Auxiliary 1	0.931	0.838	0.831	6.183	4.935	6.039	5.044	1.515	1.693	3.510	3.292
Auxiliary 2	0.879	0.856	0.834	6.064	4.926	6.043	5.983	1.725	1.722	3.525	3.226
Auxiliary 3	0.970	0.813	0.831	6.272	4.908	6.023	3.466	1.275	1.685	3.506	2.803
Auxiliary 4	0.736	0.982	0.832	5.741	5.905	6.351	6.916	6.638	1.702	3.533	2.930

Table A.5 Results considering goal 2 as a constraint

* goals 2 and 6 act as constraints

Fourth generation, having the following aspiration levels:

A= (6.0, 4.9, 6.0, 6.0, 1.5, 1.8)

	Decision variables			Goal	*const	Goal	*const	Goal	*const	const	Fitness
	x_1	x_2	x_3	z_1	z_2	z_3	z_4	z_5	z_6	q_4	S
Aspiration Levels				**≥6.0**	**≥4.9**	**≥6.0**	**≥6.0**	**≤1.5**	**≤1.8**		
Surrogate	0.803	0.881	0.833	5.892	4.974	6.064	6.620	2.100	1.718	3.522	3.408
Auxiliary 1	0.849	0.864	0.831	5.998	4.926	6.045	6.286	1.833	1.685	3.503	2.214
Auxiliary 2	0.861	0.861	0.833	6.024	4.922	6.043	6.180	1.785	1.718	3.523	2.186
Auxiliary 3	0.835	0.965	0.834	5.967	5.829	6.324	6.402	5.074	1.727	3.547	1.954

Table A.6 Results considering goal 4 as a constraint

* goals 2, 4 and 6 act as constraints

Fifth generation, having the following aspiration levels:

A= (6.0, 4.9, 6.0, 6.0, 1.8, 1.8)

	Decision variables			Goal	*const	Goal	*const	*const	*const	const	Fitness
	x_1	x_2	x_3	z_1	z_2	z_3	z_4	z_5	z_6	q_4	S
Aspiration Levels				**≥6.0**	**≥4.9**	**≥6.0**	**≥6.0**	**≤1.8**	**≤1.8**		
Surrogate	0.876	0.860	0.837	6.059	4.948	6.050	6.012	1.775	1.761	3.548	1.982
Auxiliary 1	0.876	0.861	0.835	6.059	4.958	6.053	6.012	1.794	1.744	3.538	0.991
Auxiliary 2	0.875	0.859	0.833	6.057	4.937	6.047	6.021	1.757	1.710	3.518	0.991

Table A.7 Results considering goal 5 as a constraint

* goals 2, 4, 5 and 6 act as constraints

A.2.2 Arithmetic crossover with final aspiration levels $z_1 = 5.9$, $z_2 = 5$, $z_3 = 6$, $z_4 = 6.5$, $z_5 = 2.3$, and $z_6 = 1.8$.

First generation, having the following aspiration levels:

A= (6.0, 6.0, 6.0, 6.5, 1.5, 1.5)

	Decision variables			Goals						const	Fitness
	x_1	x_2	x_3	z_1	z_2	z_3	z_4	z_5	z_6	q_4	S
Aspiration Levels				**≥6.0**	**≥6.0**	**≥6.0**	**≥6.5**	**≤1.5**	**≤1.5**		
Surrogate	0.782	0.609	0.841	5.845	3.513	5.595	6.729	0.375	1.809	3.547	6.229
Auxiliary 1	0.554	0.686	0.841	5.327	3.595	5.631	7.304	0.590	1.814	3.544	5.227
Auxiliary 2	0.739	0.581	0.853	5.748	3.346	5.542	6.905	0.316	1.977	3.631	4.597
Auxiliary 3	0.826	0.598	0.853	5.944	3.535	5.598	6.474	0.350	1.972	3.634	5.259
Auxiliary 4	0.734	0.569	0.851	5.736	3.291	5.524	6.924	0.292	1.947	3.615	5.448
Auxiliary 5	0.651	0.887	0.853	5.547	4.835	6.025	7.146	2.206	1.972	3.652	5.543
Auxiliary 6	0.835	0.632	0.883	5.965	3.689	5.649	6.407	0.430	2.500	3.872	4.996

Table A.8 Results using the goal values as aspiration levels

Second generation, having the following aspiration levels:

A= (6.0, 6.0, 6.0, 6.5, 1.5, 1.8)

	Decision variables			Goals					*const	const	Fitness
	x_1	x_2	x_3	z_1	z_2	z_3	z_4	z_5	z_6	q_4	S
Aspiration Levels				**≥6.0**	**≥6.0**	**≥6.0**	**≥6.5**	**≤1.5**	**≤1.8**		
Surrogate	0.822	0.676	0.836	5.935	3.852	5.703	6.503	0.555	1.748	3.519	4.990
Auxiliary 1	0.755	0.683	0.838	5.785	3.787	5.687	6.845	0.578	1.770	3.528	3.974
Auxiliary 2	0.758	0.362	0.838	5.791	2.597	5.288	6.834	0.041	1.778	3.519	3.149
Auxiliary 3	0.800	0.674	0.836	5.887	3.813	5.692	6.632	0.550	1.748	3.518	3.940
Auxiliary 4	0.958	0.584	0.836	6.246	3.794	5.663	4.081	0.321	1.748	3.524	3.816
Auxiliary 5	0.610	0.984	0.839	5.455	5.802	6.321	7.221	6.895	1.783	3.574	3.983

Table A.9 Results considering goal 6 as a constraint

* goal 6 acts as a constraint

Third generation, having the following aspiration levels:

A= (6.0, 5.0, 6.0, 6.5, 1.5, 1.8)

	Decision variables			Goal	*const	Goals			*const	const	Fitness
	x_1	x_2	x_3	z_1	z_2	z_3	z_4	z_5	z_6	q_4	S
Aspiration Levels				**≥6.0**	**≥5.0**	**≥6.0**	**≥6.5**	**≤1.5**	**≤1.8**		
Surrogate	0.869	0.874	0.834	6.044	5.032	6.077	6.089	1.973	1.727	3.530	4.363
Auxiliary 1	0.903	0.863	0.832	6.119	5.029	6.073	5.635	1.818	1.702	3.516	3.354
Auxiliary 2	0.854	0.884	0.832	6.009	5.085	6.095	6.245	2.155	1.697	3.514	3.476
Auxiliary 3	0.919	0.882	0.834	6.156	5.212	6.128	5.328	2.118	1.727	3.535	3.366
Auxiliary 4	0.712	0.984	0.836	5.685	5.906	6.352	6.995	6.895	1.752	3.561	2.929

Table A.10 Results considering goal 2 as a constraint

* goals 2 and 6 act as constraints

Fourth generation, having the following aspiration levels:

A= (6.0, 5.0, 6.0, 6.5, 1.5, 1.8)

	Decision variables			Goal	*const	Goal	*const	Goal	*const	const	Fitness
	x_1	x_2	x_3	z_1	z_2	z_3	z_4	z_5	z_6	q_4	S
Aspiration Levels				**≥6.0**	**≥5.0**	**≥6.0**	**≥6.5**	**≤1.5**	**≤1.8**		
Surrogate	0.819	0.905	0.837	5.930	5.189	6.130	6.517	2.572	1.761	3.551	3.705
Auxiliary 1	0.798	0.892	0.833	5.881	5.052	6.088	6.647	2.300	1.718	3.524	2.519
Auxiliary 2	0.808	0.886	0.835	5.905	5.018	6.077	6.587	2.180	1.744	3.538	2.470
Auxiliary 3	0.803	0.982	0.833	5.893	5.997	6.377	6.616	6.638	1.718	3.546	1.959

Table A.11 Results considering goal 4 as a constraint

* goals 2, 4 and 6 act as constraints

Fifth generation, having the following aspiration levels:

A= (6.0, 5.0, 6.0, 6.5, 2.3, 1.8)

	Decision variables			Goal	*const	Goal	*const	*const	*const	const	Fitness
	x_1	x_2	x_3	z_1	z_2	z_3	z_4	z_5	z_6	q_4	S
Aspiration Levels				**≥6.0**	**≥5.0**	**≥6.0**	**≥6.5**	**≤2.3**	**≤1.8**		
Surrogate	0.821	0.891	0.840	5.935	5.077	6.095	6.503	2.272	1.796	3.568	1.995
Auxiliary 1	0.814	0.891	0.839	5.917	5.071	6.093	6.553	2.286	1.787	3.563	0.985
Auxiliary 2	0.821	0.882	0.835	5.935	5.006	6.073	6.503	2.106	1.744	3.538	1.011

Table A.12 Results considering goal 5 as a constraint

* goals 2, 4, 5 and 6 act as constraints

A.3 Unfair crossover

A.3.1 Unfair crossover with final aspiration levels $z_1 = 6$, $z_2 = 4.9$, $z_3 = 6$, $z_4 = 6$, $z_5 = 1.8$, and $z_6 = 1.8$.

First generation, having the following aspiration levels:

A= (6.0, 6.0, 6.0, 6.5, 1.5, 1.5)

	Decision variables			Goals						const	Fitness
	x_1	x_2	x_3	z_1	z_2	z_3	z_4	z_5	z_6	q_4	s
Aspiration Levels				**≥6.0**	**≥6.0**	**≥6.0**	**≥6.5**	**≤1.5**	**≤1.5**		
Surrogate	0.745	0.548	0.837	5.761	3.228	5.502	6.884	0.256	1.767	3.515	6.279
Auxiliary 1	0.711	0.716	0.855	5.684	3.886	5.722	6.997	0.703	2.008	3.654	5.329
Auxiliary 2	0.716	0.471	0.861	5.695	2.912	5.398	6.982	0.145	2.106	3.688	4.597
Auxiliary 3	0.802	0.649	0.878	5.890	3.703	5.656	6.626	0.473	2.402	3.831	5.537
Auxiliary 4	0.999	0.719	0.853	6.337	4.534	5.895	0.579	0.712	1.977	6.659	5.081
Auxiliary 5	0.957	0.787	0.838	6.242	4.709	5.963	4.160	1.076	1.778	3.555	5.990
Auxiliary 6	0.814	0.667	0.886	5.918	3.800	5.687	6.551	0.526	2.566	3.899	4.991

Table A.13 Results using the goal values as aspiration levels

Second generation, having the following aspiration levels:

A= (6.0, 6.0, 6.0, 6.5, 1.5, 1.8)

	Decision variables			Goals					*const	const	Fitness
	x_1	x_2	x_3	z_1	z_2	z_3	z_4	z_5	z_6	q_4	s
Aspiration Levels				**≥6.0**	**≥6.0**	**≥6.0**	**≥6.5**	**≤1.5**	**≤1.8**		
Surrogate	0.781	0.672	0.836	5.842	3.774	5.681	6.735	0.544	1.748	3.517	5.000
Auxiliary 1	0.716	0.660	0.838	5.695	3.639	5.640	6.982	0.506	1.778	3.530	3.981
Auxiliary 2	0.768	0.331	0.840	5.814	2.509	5.257	6.791	0.019	1.796	3.528	3.143
Auxiliary 3	0.868	0.621	0.837	6.041	3.702	5.649	6.104	0.401	1.761	3.526	3.946
Auxiliary 4	0.999	0.674	0.840	6.339	4.336	5.830	0.462	0.550	1.796	3.561	3.726
Auxiliary 5	0.615	0.961	0.832	5.466	5.510	6.233	7.213	4.804	1.702	3.521	4.050

Table A.14 Results considering goal 6 as a constraint

* goal 6 acts as a constraint

Third generation, having the following aspiration levels:

A= (6.0, 4.9, 6.0, 6.5, 1.5, 1.8)

	Decision variables			Goal	*const	Goals			*const	const	Fitness
	x_1	x_2	x_3	z_1	z_2	z_3	z_4	z_5	z_6	q_4	s
Aspiration Levels				**≥6.0**	**≥4.9**	**≥6.0**	**≥6.5**	**≤1.5**	**≤1.8**		
Surrogate	0.921	0.838	0.837	6.161	4.903	6.031	5.282	1.508	1.761	3.548	4.205
Auxiliary 1	0.846	0.865	0.831	5.991	4.924	6.045	6.316	1.843	1.693	3.508	3.250
Auxiliary 2	0.907	0.866	0.833	6.128	5.059	6.082	5.568	1.858	1.710	3.522	3.385
Auxiliary 3	0.988	0.823	0.832	6.313	5.045	6.062	1.979	1.365	1.706	3.522	2.850
Auxiliary 4	0.753	0.976	0.838	5.779	5.850	6.334	6.855	6.030	1.770	3.570	2.934

Table A.15 Results considering goal 2 as a constraint

* goals 2 and 6 act as constraint

Fourth generation, having the following aspiration levels:

A= (6.0, 4.9, 6.0, 6.0, 1.5, 1.8)

	Decision variables			Goal	*const	Goal	*const	Goal	*const	const	Fitness
	x_1	x_2	x_3	z_1	z_2	z_3	z_4	z_5	z_6	q_4	s
Aspiration Levels				**≥6.0**	**≥4.9**	**≥6.0**	**≥6.0**	**≤1.5**	**≤1.8**		
Surrogate	0.821	0.877	0.832	5.935	4.975	6.063	6.503	2.035	1.702	3.513	3.357
Auxiliary 1	0.851	0.868	0.834	6.002	4.956	6.055	6.270	1.888	1.727	3.528	2.250
Auxiliary 2	0.853	0.883	0.835	6.006	5.072	6.091	6.254	2.130	1.735	3.535	2.419
Auxiliary 3	0.863	0.980	0.833	6.029	6.080	6.399	6.159	6.476	1.718	3.549	1.933

Table A.16 Results considering goal 4 as a constraint

* goals 2, 4 and 6 act as constraint

Fifth generation, having the following aspiration levels:

A= (6.0, 4.9, 6.0, 6.0, 1.8, 1.8)

	Decision variables			Goal	*const	Goal	*const	*const	*const	const	Fitness
	x_1	x_2	x_3	z_1	z_2	z_3	z_4	z_5	z_6	q_4	s
Aspiration Levels				**≥6.0**	**≥4.9**	**≥6.0**	**≥6.0**	**≤1.8**	**≤1.8**		
Surrogate	0.876	0.861	0.833	6.059	4.953	6.052	6.012	1.785	1.718	3.524	1.982
Auxiliary 1	0.876	0.861	0.831	6.059	4.953	6.052	6.012	1.785	1.685	3.504	0.991
Auxiliary 2	0.877	0.856	0.832	6.060	4.925	6.043	6.004	1.729	1.706	3.516	0.990

Table A.17 Results considering goal 5 as a constraint

* goals 2, 4, 5 and 6 act as constraint

A.3.2 Unfair crossover with final aspiration levels $z_1 = 5.9$, $z_2 = 5$, $z_3 = 6$, $z_4 = 6.5$, $z_5 = 2.3$, and $z_6 = 1.8$.

First generation, having the following aspiration levels:

A= (6.0, 6.0, 6.0, 6.5, 1.5, 1.5)

	Decision variables			Goals						const	Fitness
	x_1	x_2	x_3	z_1	z_2	z_3	z_4	z_5	z_6	q_4	S
Aspiration Levels				**≥6.0**	**≥6.0**	**≥6.0**	**≥6.5**	**≤1.5**	**≤1.5**		
Surrogate	0.745	0.548	0.837	5.761	3.228	5.502	6.884	0.256	1.761	3.515	6.279
Auxiliary 1	0.711	0.716	0.855	5.684	3.886	5.722	6.997	0.703	2.008	3.654	5.329
Auxiliary 2	0.716	0.471	0.861	5.695	2.912	5.398	6.982	0.145	2.106	3.688	4.597
Auxiliary 3	0.802	0.649	0.878	5.890	3.703	5.656	6.626	0.473	2.402	3.831	5.537
Auxiliary 4	0.999	0.719	0.853	6.337	4.534	5.895	0.579	0.712	1.977	3.659	5.081
Auxiliary 5	0.957	0.787	0.838	6.242	4.709	5.963	4.160	1.076	1.778	3.555	5.990
Auxiliary 6	0.814	0.667	0.886	5.918	3.800	5.687	6.551	0.526	2.566	3.899	4.991

Table A.18 Results using the goal values as aspiration levels

Second generation, having the following aspiration levels:

A= (6.0, 6.0, 6.0, 6.5, 1.5, 1.8)

	Decision variables			Goals					*const	const	Fitness
	x_1	x_2	x_3	z_1	z_2	z_3	z_4	z_5	z_6	q_4	S
Aspiration Levels				**≥6.0**	**≥6.0**	**≥6.0**	**≥6.5**	**≤1.5**	**≤1.8**		
Surrogate	0.781	0.672	0.836	5.842	3.774	5.681	6.735	0.544	1.748	3.517	5.000
Auxiliary 1	0.716	0.660	0.838	5.695	3.639	5.640	6.982	0.506	1.778	3.530	3.981
Auxiliary 2	0.768	0.331	0.840	5.814	2.509	5.257	6.791	0.019	1.796	3.528	3.143
Auxiliary 3	0.868	0.621	0.837	6.041	3.702	5.649	6.104	0.401	1.761	3.526	3.946
Auxiliary 4	0.999	0.674	0.840	6.339	4.336	5.830	0.462	0.550	1.796	3.561	3.726
Auxiliary 5	0.615	0.961	0.832	5.466	5.510	6.233	7.213	4.804	1.702	3.521	4.050

Table A.19 Results considering goal 6 as a constraint

* goal 6 acts as a constraint

Third generation, having the following aspiration levels:

A= (6.0, 5.0, 6.0, 6.5, 1.5, 1.8)

	Decision variables			Goal	*const	Goals			*const	const	Fitness
	x_1	x_2	x_3	z_1	z_2	z_3	z_4	z_5	z_6	q_4	S
Aspiration Levels				**≥6.0**	**≥5.0**	**≥6.0**	**≥6.5**	**≤1.5**	**≤1.8**		
Surrogate	0.829	0.894	0.831	5.952	5.115	6.106	6.451	2.335	1.685	3.507	4.555
Auxiliary 1	0.844	0.886	0.835	5.987	5.080	6.094	6.331	2.187	1.744	3.540	3.469
Auxiliary 2	0.857	0.889	0.832	6.016	5.124	6.107	6.215	2.232	1.702	3.517	3.531
Auxiliary 3	0.974	0.858	0.831	6.282	5.216	6.120	3.167	1.752	1.685	3.513	3.104
Auxiliary 4	0.736	0.982	0.832	5.741	5.905	6.351	6.916	6.638	1.702	3.533	2.930

Table A.20 Results considering goal 2 as a constraint

* goals 2 and 6 act as constraints

Fourth generation, having the following aspiration levels:

A= (6.0, 5.0, 6.0, 6.5, 1.5, 1.8)

	Decision variables			Goal	*const	Goal	*const	Goal	*const	const	Fitness
	x_1	x_2	x_3	z_1	z_2	z_3	z_4	z_5	z_6	q_4	S
Aspiration Levels				**≥6.0**	**≥5.0**	**≥6.0**	**≥6.5**	**≤1.5**	**≤1.8**		
Surrogate	0.660	0.909	0.835	5.569	5.021	6.083	7.125	2.665	1.744	3.534	3.840
Auxiliary 1	0.715	0.912	0.834	5.693	5.115	6.110	6.984	2.763	1.727	3.528	2.824
Auxiliary 2	0.764	0.895	0.833	5.804	5.027	6.082	6.809	2.356	1.714	3.520	2.604
Auxiliary 3	0.770	0.994	0.835	5.817	6.135	6.419	6.785	8.508	1.735	3.558	1.966

Table A.21 Results considering goal 4 as a constraint

* goals 2, 4 and 6 act as constraints

Fifth generation, having the following aspiration levels:

A= (6.0, 5.0, 6.0, 6.5, 2.3, 1.8)

	Decision variables			Goal	*const	Goal	*const	*const	*const	const	Fitness
	x_1	x_2	x_3	z_1	z_2	z_3	z_4	z_5	z_6	q_4	S
Aspiration Levels				**≥6.0**	**≥5.0**	**≥6.0**	**≥6.5**	**≤2.3**	**≤1.8**		
Surrogate	0.821	0.891	0.840	5.935	5.077	6.095	6.503	2.272	1.796	3.568	1.995
Auxiliary 1	0.814	0.891	0.839	5.917	5.071	6.093	6.553	2.286	1.787	3.563	0.985
Auxiliary 2	0.821	0.886	0.831	5.935	5.039	6.083	6.503	2.180	1.685	3.505	1.011

Table A.22 Results considering goal 5 as a constraint

* goals 2, 4, 5 and 6 act as constraints

A.4 Simulated binary crossover

A.4.1 Simulated binary crossover with final aspiration levels $z_1 = 6$, $z_2 = 4.9$, $z_3 = 6$, $z_4 = 6$, $z_5 = 1.8$, and $z_6 = 1.8$.

First generation, having the following aspiration levels:

A= (6.0, 6.0, 6.0, 6.5, 1.5, 1.5)

	Decision variables			Goals						const	Fitness
	x_1	x_2	x_3	z_1	z_2	z_3	z_4	z_5	z_6	q_4	s
Aspiration Levels				**≥6.0**	**≥6.0**	**≥6.0**	**≥6.5**	**≤1.5**	**≤1.5**		
Surrogate	0.819	0.670	0.845	5.928	3.823	5.694	6.521	0.537	1.860	3.580	6.230
Auxiliary 1	0.677	0.628	0.848	5.606	3.462	5.584	7.088	0.420	1.908	3.595	5.276
Auxiliary 2	0.848	0.390	0.855	5.995	2.828	5.358	6.298	0.064	2.008	3.645	4.534
Auxiliary 3	0.724	0.776	0.853	5.714	4.201	5.823	6.956	1.005	1.977	3.644	5.401
Auxiliary 4	0.800	0.709	0.836	5.887	3.967	5.743	6.632	0.672	1.752	3.523	5.192
Auxiliary 5	0.910	0.974	0.833	6.137	6.100	6.401	5.497	5.828	1.710	3.545	5.221
Auxiliary 6	0.822	0.704	0.996	5.936	3.979	5.745	6.498	0.653	10.464	5.152	4.999

Table A.23 Results using the goal values as aspiration levels

Second generation, having the following aspiration levels:

A= (6.0, 6.0, 6.0, 6.5, 1.5, 1.8)

	Decision variables			Goals					*const	const	Fitness
	x_1	x_2	x_3	z_1	z_2	z_3	z_4	z_5	z_6	q_4	s
Aspiration Levels				**≥6.0**	**≥6.0**	**≥6.0**	**≥6.5**	**≤1.5**	**≤1.8**		
Surrogate	0.847	0.623	0.836	5.992	3.670	5.641	6.310	0.406	1.752	3.520	5.001
Auxiliary 1	0.748	0.663	0.838	5.768	3.688	5.655	6.873	0.513	1.770	3.527	3.976
Auxiliary 2	0.638	0.362	0.837	5.518	2.462	5.249	7.171	0.041	1.761	3.503	3.164
Auxiliary 3	0.787	0.656	0.834	5.857	3.711	5.660	6.703	0.493	1.727	3.504	3.939
Auxiliary 4	0.999	0.489	0.838	6.339	3.611	5.589	0.462	0.168	1.778	3.542	3.794
Auxiliary 5	0.745	0.989	0.835	5.761	6.017	6.385	6.884	7.571	1.735	3.555	3.923

Table A.24 Results considering goal 6 as a constraint

* goal 6 acts as a constraint

Third generation, having the following aspiration levels:

A= (6.0, 4.9, 6.0, 6.5, 1.5, 1.8)

	Decision variables			Goal	*const	Goals			*const	const	Fitness
	x_1	x_2	x_3	z_1	z_2	z_3	z_4	z_5	z_6	q_4	s
Aspiration Levels				**≥6.0**	**≥4.9**	**≥6.0**	**≥6.5**	**≤1.5**	**≤1.8**		
Surrogate	0.930	0.838	0.832	6.181	4.928	6.037	5.063	1.508	1.702	3.515	4.254
Auxiliary 1	0.897	0.852	0.833	6.107	4.936	6.044	5.726	1.668	1.718	3.524	3.240
Auxiliary 2	0.934	0.832	0.832	6.191	4.904	6.029	4.949	1.450	1.702	3.515	3.249
Auxiliary 3	0.974	0.840	0.838	6.282	5.094	6.081	3.191	1.538	1.778	3.563	2.967
Auxiliary 4	0.756	0.996	0.838	5.785	6.138	6.420	6.844	8.776	1.778	3.582	2.921

Table A.25 Results considering goal 2 as a constraint

* goal 2 and 6 acts as a constraint

Fourth generation, having the following aspiration levels:

A= (6.0, 4.9, 6.0, 6.0, 1.5, 1.8)

	Decision variables			Goal	*const	Goal	*const	Goal	*const	const	Fitness
	x_1	x_2	x_3	z_1	z_2	z_3	z_4	z_5	z_6	q_4	s
Aspiration Levels				**≥6.0**	**≥4.9**	**≥6.0**	**≥6.0**	**≤1.5**	**≤1.8**		
Surrogate	0.730	0.892	0.840	5.728	4.962	6.063	6.935	2.300	1.796	3.564	3.570
Auxiliary 1	0.770	0.898	0.837	5.817	5.063	6.093	6.785	2.430	1.761	3.547	2.605
Auxiliary 2	0.866	0.871	0.836	6.037	5.008	6.070	6.123	1.935	1.752	3.544	2.284
Auxiliary 3	0.875	0.994	0.831	6.057	6.299	6.464	6.021	8.379	1.685	3.535	1.919

Table A.26 Results considering goal 4 as a constraint

* goal 2, 4 and 6 acts as a constraint

Fifth generation, having the following aspiration levels:

A= (6.0, 4.9, 6.0, 6.0, 1.8, 1.8)

	Decision variables			Goal	*const	Goal	*const	*const	*const	const	Fitness
	x_1	x_2	x_3	z_1	z_2	z_3	z_4	z_5	z_6	q_4	s
Aspiration Levels				**≥6.0**	**≥4.9**	**≥6.0**	**≥6.0**	**≤1.8**	**≤1.8**		
Surrogate	0.876	0.861	0.833	6.059	4.953	6.052	6.012	1.785	1.718	3.524	1.982
Auxiliary 1	0.876	0.861	0.831	6.059	4.953	6.052	6.012	1.785	1.685	3.504	0.991
Auxiliary 2	0.877	0.854	0.831	6.060	4.911	6.038	6.004	1.702	1.685	3.504	0.990

Table A.27 Results considering goal 5 as a constraint

* goal 2, 4, 5 and 6 acts as a constraint

A.4.1 Simulated binary crossover with final aspiration levels $z_1 = 6$, $z_2 = 5$, $z_3 = 6$, $z_4 = 6.5$, $z_5 = 2.3$, and $z_6 = 1.8$.

First generation, having the following aspiration levels:

A= (6.0, 6.0, 6.0, 6.5, 1.5, 1.5)

	Decision variables			Goals						const	Fitness
	x_1	x_2	x_3	z_1	z_2	z_3	z_4	z_5	z_6	q_4	s
Aspiration Levels				**≥6.0**	**≥6.0**	**≥6.0**	**≥6.5**	**≤1.5**	**≤1.5**		
Surrogate	0.819	0.670	0.845	5.928	3.823	5.694	6.521	0.537	1.860	3.580	6.230
Auxiliary 1	0.677	0.628	0.848	5.606	3.462	5.584	7.088	0.420	1.908	3.595	5.276
Auxiliary 2	0.848	0.390	0.855	5.995	2.828	5.358	6.298	0.064	2.008	3.645	4.534
Auxiliary 3	0.724	0.776	0.853	5.714	4.201	5.823	6.956	1.005	1.977	3.644	5.401
Auxiliary 4	0.800	0.709	0.836	5.887	3.967	5.743	6.632	0.672	1.752	3.523	5.192
Auxiliary 5	0.910	0.974	0.833	6.137	6.100	6.401	5.497	5.828	1.710	3.545	5.221
Auxiliary 6	0.822	0.704	0.996	5.936	3.979	5.745	6.498	0.653	10.464	5.152	4.999

Table A.28 Results using the goal values as aspiration levels

Second generation, having the following aspiration levels:

A= (6.0, 6.0, 6.0, 6.5, 1.5, 1.8)

	Decision variables			Goals					*const	const	Fitness
	x_1	x_2	x_3	z_1	z_2	z_3	z_4	z_5	z_6	q_4	s
Aspiration Levels				**≥6.0**	**≥6.0**	**≥6.0**	**≥6.5**	**≤1.5**	**≤1.8**		
Surrogate	0.847	0.623	0.836	5.992	3.670	5.641	6.310	0.406	1.752	3.520	5.001
Auxiliary 1	0.748	0.663	0.838	5.768	3.688	5.655	6.873	0.513	1.770	3.527	3.976
Auxiliary 2	0.638	0.362	0.837	5.518	2.462	5.249	7.171	0.041	1.761	3.503	3.164
Auxiliary 3	0.787	0.656	0.834	5.857	3.711	5.660	6.703	0.493	1.727	3.504	3.939
Auxiliary 4	0.999	0.489	0.838	6.339	3.611	5.589	0.462	0.168	1.778	3.542	3.794
Auxiliary 5	0.745	0.989	0.835	5.761	6.017	6.385	6.884	7.571	1.735	3.555	3.923

Table A.29 Results considering goal 6 as a constraint

* goal 6 acts as a constraint

Third generation, having the following aspiration levels:

A= (6.0, 5.0, 6.0, 6.5, 1.5, 1.8)

	Decision variables			Goal	*const	Goals			*const	const	Fitness
	x_1	x_2	x_3	z_1	z_2	z_3	z_4	z_5	z_6	q_4	s
Aspiration Levels				**≥6.0**	**≥5.0**	**≥6.0**	**≥6.5**	**≤1.5**	**≤1.8**		
Surrogate	0.860	0.874	0.835	6.022	5.016	6.073	6.187	1.979	1.744	3.539	4.354
Auxiliary 1	0.919	0.869	0.839	6.156	5.113	6.097	5.328	1.904	1.787	3.567	3.473
Auxiliary 2	0.943	0.847	0.832	6.212	5.032	6.068	4.671	1.617	1.702	3.518	3.436
Auxiliary 3	0.974	0.839	0.830	6.282	5.085	6.079	3.191	1.523	1.677	3.505	2.957
Auxiliary 4	0.632	0.994	0.834	5.504	5.978	6.374	7.183	8.508	1.727	3.547	2.936

Table A.30 Results considering goal 2 as a constraint

* goal 2 and 6 acts as a constraint

Fourth generation, having the following aspiration levels:

A= (6.0, 5.0, 6.0, 6.5, 1.5, 1.8)

	Decision variables			Goal	*const	Goal	*const	Goal	*const	const	Fitness
	x_1	x_2	x_3	z_1	z_2	z_3	z_4	z_5	Z_6	q_4	s
Aspiration Levels				**≥6.0**	**≥5.0**	**≥6.0**	**≥6.0**	**≤1.5**	**≤1.8**		
Surrogate	0.738	0.901	0.838	5.746	5.040	6.087	6.909	2.476	1.770	3.551	3.680
Auxiliary 1	0.734	0.901	0.838	5.736	5.035	6.085	6.924	2.476	1.770	3.551	2.636
Auxiliary 2	0.791	0.896	0.832	5.866	5.076	6.096	6.681	2.385	1.702	3.514	2.613
Auxiliary 3	0.813	0.973	0.836	5.916	5.898	6.346	6.557	5.763	1.752	3.563	1.960

Table A.31 Results considering goal 4 as a constraint

* goal 2, 4 and 6 acts as a constraint

Fifth generation, having the following aspiration levels:

A= (6.0, 5.0, 6.0, 6.0, 2.3, 1.8)

	Decision variables			Goal	*const	Goal	*const	*const	*const	const	Fitness
	x_1	x_2	x_3	z_1	z_2	z_3	z_4	z_5	z_6	q_4	s
Aspiration Levels				**≥6.0**	**≥5.0**	**≥6.0**	**≥6.0**	**≤2.3**	**≤1.8**		
Surrogate	0.820	0.889	0.836	5.932	5.059	6.089	6.512	2.232	1.752	3.544	1.997
Auxiliary 1	0.820	0.892	0.840	5.932	5.086	6.098	6.512	2.300	1.796	3.569	0.984
Auxiliary 2	0.821	0.886	0.831	5.935	5.039	6.083	6.503	2.180	1.685	3.505	1.011

Table A.32 Results considering goal 5 as a constraint

* goal 2, 4, 5 and 6 acts as a constraint

APPENDIX B

Parameter	Expected Value	Standard Deviation	Units
l_2	0.1375	0.0550	AU/ha
l_3	0.0365	0.0130	AU/ha
l_i, $i \neq 2, 3$	0	0	AU/ha
q_2	0.1000×10^3	NR*	$/ha
q_3	0.0750×10^3	NR*	$/ha
w_2	0.0019×10^3	NR*	cu m/ha
w_3	0.0005×10^3	NR*	cu m/ha

* NR = not required in the analysis

Table B.1 Livestock Production coefficients

Catchment Methods	Approximate Cost per ha (in $)	Efficiency (in %)	Estimated Life
Compacted earth	50.60	30-60	indefinite
Compacted earth sodium treated	85.20	40-70	indefinite
Graveled plastic	191.60	60-80	20-25 years

Table B.2 Soil Treatments for Water Runoff coefficients

Parameter	Expected Value	Standard Deviation	Units
r_1	0.428×10^3	0.223×10^3*	cu m/ha
r_2	0.098×10^3	0.152×10^3	cu m/ha
r_3	0.079×10^3	0.089×10^3	cu m/ha
r_4	0.990×10^3	0.223×10^3*	cu m/ha
r_5	1.410×10^3	0.223×10^3	cu m/ha
r_6	1.980×10^3	0.223×10^3*	cu m/ha
r_i, $i \neq [1,6]$	0	0	cu m/ha
q_1	0	0	$/ha
q_2	0.100×10^3	NR**	$/ha
q_3	0.075×10^3	NR**	$/ha
q_4	0.056×10^3	NR**	$/ha
q_5	0.085×10^3	NR**	$/ha
q_6	0.191×10^3	NR**	$/ha

* Assumed value

** NR = not required in the analysis

Table B.3 Water Runoff coefficients

Parameter	Expected Value	Standard Deviation	Units
c_7	3.024×10^3	0.505×10^3	kg/ha
c_8	1.568×10^3	0.249×10^3	kg/ha
c_9	7.392×10^3	1.037×10^3	kg/ha
c_{10}	3.169×10^3	0.102×10^3	kg/ha
c_{11}	2.576×10^3	0.249×10^3	kg/ha
q_7	0.232×10^3	NR*	$/ha
q_8	0.323×10^3	NR*	$/ha
q_9	0.262×10^3	NR*	$/ha
q_{10}	0.242×10^3	NR*	$/ha
q_{11}	0.278×10^3	NR*	$/ha
w_7	5.830×10^3	NR*	cu m/ha
w_8	5.010×10^3	NR*	cu m/ha
w_9	18.850×10^3	NR*	cu m/ha
w_{10}	6.390×10^3	NR*	cu m/ha
w_{11}	13.700×10^3	NR*	cu m/ha

* NR = not required in the analysis

Table B.4 Crop Model coefficients

Treatment i	K*	C**	P***	LS****	$E(s_i)$ (cu m/ha)	$[var(s_i)]^{1/2}$ (cu m/ha)
1	0.40	1.00	1.00	4.50	19.86	20.63
2	0.40	0.10	0.50	0.50	0.11	0.12
3	0.40	0.15	0.50	0.40	0.15	0.17
4	0.30	0.25	0.70	0.40	0.24	0.36
5	0.25	0.20	0.70	0.40	0.16	0.37
6	0	0	0	0	0	0
7	0.40	0.20	0.60	0.40	0.24	0.40
8	0.40	0.30	0.60	0.40	0.33	0.33
9	0.40	0.10	0.60	0.40	0.11	0.12
10	0.40	0.20	0.60	0.40	0.21	0.29
11	0.40	0.30	0.60	0.40	0.35	0.44
12	—	—	—	—	—	—

* K = soil-erodibility factor
** C = cropping-management factor
*** P = erosion-control factor
**** LS = slope length and gradient factor

Table B.5 Sediment coefficients

Cages (30 cages/ha) ($50/cage)	$1500/ha
Food pellets (30 cages/ha) (250 units/cage) (2.0 kg/unit) ($0.374/kg)	$5610/ha
Initial cost of stock (30 cages/ha) (500 units/cage) ($0.05/unit)	$750/ha
Digging of pond (reclamation program)	—
Transportation to and from power plant, ($10.15/man-hour) (100 man-hour/ha)	$1015/ha
$E(q_{12})$ Total	$8875/ha
$E(f_{12})$, expected yield (30 cages/ha) (250 units/cage) (0.5 kg/unit)	3741 kg/ha
$var(f_{12})$, yield variance, (assumed)	$(2000 \text{ kg/ha})^2$

Table B.6 Fish-Harvesting coefficients

References

Ackoff, R. (1978), *The art of problem solving*, John Wiley and Sons, Inc..

Anderson, D.R., Sweeney, D.J. and Williams, T.A. (1997), *An Introduction to Management Science: quantitative approaches to decision making*, eighth edition, West Publishing Company.

Applegate, D., Bixby, R., Chuatal, V. and Cook, W.: Overview [www] http://www.caam.rice.edu/keck/tsp/history.html

Baker, S., Ponniah, D. and Smith, S. (1999), "Survey of Risk Management in Major U.K. Companies," *Journal of Professional Issues in Engineering Education and Practice*, Vol. 125:3, pp. 94-102.

Bäck, T. (1996), *Evolutionary Algorithms in Theory and Practice, Evolution Strategies, Evolutionary Programming, Genetic Algorithms*. Oxford University Press, NY.

Bäck, T. (2000), "Introduction to evolutionary algorithms," *Evolutionary Computation 1*, Bäck, T., Fogel, D.B. and Michalewicz, Z. (Eds.), Chapter 7, Institute of Physics Publishing.

Bäck, T., Fogel, D. and Michalewicz, Z. (2000), "Preface," *Evolutionary Computation 1*, Bäck, T., Fogel, D.B. and Michalewicz, Z. (Eds.), Institute of Physics Publishing.

Bäck, T., Hammel, U. and Schwefel, H.-P. (1997), "Evolutionary Computation: Comments on the History and Current State," *IEEE Trans. Evolutionary Computation*, Vol. 1:1, pp. 3-17.

Bäck, T., Hoffmeister, F. and Schwefel, H.-P. (1991), "A survey of Evolution Strategies," *Proceedings of the 4th International Conference on Genetic Algorithms*, Belew, R.K. and Booker, L.B. (Eds.), Morgan Kaufmann Publishers, San Mateo, CA.

Bell, P.C (1999), *Management science/operations research: a strategic perspective*, South-Western College Publishing, Cincinnati, Ohio.

Berrada, I., Ferland, J.A. and Michelon, P. (1996), "A Multi-objective Approach to Nurse Scheduling with both Hard and Soft Constraints," *Socio-Economic Planning Sciences*, Vol. 30:3, pp. 183-193.

Blickle, T. (2000), "Tournament Selection," *Evolutionary Computation 1*, Bäck, T., Fogel, D.B. and Michalewicz, Z. (Eds.), Chapter 24, Institute of Physics Publishing.

Camponogara, E. and Talukdar, S.N. (1997), "A Genetic Algorithm for Constrained and Multiobjective Optimization," in Jarmo T. Alander, (Ed.), *3rd Nordic Workshop on Genetic Algorithms and Their Applications* (3NWGA), pp. 49-62, Vaasa, Finland, August, University of Vaasa.

Carlsson, C. (1984), "On the Relevance of Fuzzy Sets in Management Science Methodology," *Studies in the Management Science 20*, Zimmermann, H.-J, Zadeh, L.A. and Gaines, B.R., Elsevier Science Publishers B.V., pp. 11-28.

Chen, and Hwang, C-L. (1991), *Fuzzy Multiple Attribute Decision Making: Methods and Applications*, Springer-Verlag, Lecture Notes in Economics and Mathematical Systems No. 375.

Cheng, B.M.W., Lee, J.H.M. and Wu, J.C.K. (1996), "A Constraint-Based Nurse Rostering System Using a Redundant Modeling Approach," *IEEE 8th Internatinal Conference on Tools with Artificial Intelligence (TAI96)*, Toulouse, France.

Churchman, C.W., Ackoff, R.L. and Arnoff, E.L. (1957), *Introduction to Operations Research*, Wiley International Edition, John Wiley and Sons, Inc.

Clemen, R. (1991), *Making Hard Decisions: An Introduction to Decision Analysis*, International Student Edition, PWS-KENT Publishing Company, Boston.

Cleveland, G.A. and Smith, S.F. (1989), "Using Genetic Algorithm to Schedule Flow Shop Releases," *Proceedings of the Third International Conference on Genetic Algorithms*, Schaffer, D. (Ed.), Morgan Kaufmann.

Coello, C. (1998), "Using the Min-Max Method to Solve Multiobjective Optimization Problems with Genetic Algorithms," *Proceedings of the 6th Ibero-American Conference on IA, IBERAMIA 98*, H. Coelho (Ed.), Springer.

Coello, C. (2000), "Handling Preferences in Evolutionary Multiobjective Optimization: A Survey," Proceedings of the 2000 Congress on Evolutionary Computation, San Diego, USA.

Coello, C.A. and Christiansen, A.D. (1995), "An Approach to Multiobjective Optimization Using Genetic Algorithms," Dagli, C.H., Akay, M., Chen, C.L. P., Fernández, B.R. and Ghosh, J., (Eds.), *Intelligent Engineering Systems Through Artificial Neural Networks*, Vol. 5, Fuzzy Logic and Evolutionary Programming, pp. 411-416, ASME Press, St.Louis, Missouri, USA.

Cook, S.L. and Shutler, M.F. (1991), "The history of operations research," *Operations research in management*, Littlechild, S.C. and Shutler, M.F. (Eds.), Chapter 1, Prentice Hall International (UK) Ltd.

Curwin, J. and Slater, R. (1991), *Quantitative Methods for Business Decisions*, Third Edition, Chapman and Hall.

Deb, K. (2000), "Introduction to selection," *Evolutionary Computation 1*, Bäck, T., Fogel, D.B. and Michalewicz, Z. (Eds.), Chapter 22, Institute of Physics Publishing.

Deb, K. (2001), *Mutli-Objective Optimization using Evolutionary Algorithms*, John Wiley and Sons, Ltd.

Deb, K., Agrawal, S., Pratab, A. and Meyarivan, T. (2000), "A Fast Elitist Non-Dominated Sorting Genetic Algorithm for Multi-Objective Optimization: NSGA-II," *KanGAL report 200001*, Indian Institute of Technology, Kanpur, India.

Deisler, P.F (2002), "A Perspective: Risk Analysis as a Tool for Reducing the Risks of Terrorism," *Risk Analysis*, Vol. 22:3, pp. 405-413.

De Jong, K.A. (1975), "An analysis of the behaviour of a class of genetic adaptive systems," *Doctoral dissertation*, University of Michigan.

De Jong, K., Fogel, D. and Schwefel, H.-P. (2000), "A history of evolutionary computation," *Evolutionary Computation 1*, , Bäck, T., Fogel, D.B. and Michalewicz, Z. (Eds.), Chapter 6, Institute of Physics Publishing.

Dowsland, K. (1998), "Nurse scheduling with tabu search and strategic oscillation," *European Journal of Operations Research*, Vol. 106, pp. 393-407.

Duenas, A. and Mort, N. (2001), "A Genetic Algorithm for Multiobjective Optimisation using the interactive Sequential Multiobjective Problem Solving method," *Proceedings of the IC-AI'2001 International Conference on Artificial Intelligence*, June 25-28, Las Vegas, Nevada, USA, pp. 633-639.

Duenas, A. and Mort, N. (2001b), "A Multiobjective Optimiser with a Fuzzy Genetic Algorithm," *IASTED International Conference on Artificial Intelligence and Applications* (AIA2001), September 4-7, Marbella, Spain, pp. 191-197.

Duenas, A., Best K. and Mort, N. (2002), "Solving a Multiple Criteria Decision-Making Problem Under Uncertainty Using Protrade and @RISK 4.0," *Advances in Fuzzy Systems and Evolutionary Computation*, WSEAS Press.

Duenas, A. and Mort, N. (2002), "Solving a Multiple Criteria Decision-Making Problem Under Uncertainty," *IEEE Proccedings of International Symposium 'Intelligent Systems'. Methodology Models, Applications in Emerging Technologies*, September 10-12, Bulgaria, pp. 308-313.

Dzubera, J. and Whitley, D. (1994), "Advanced Correlation Analysis of Operators for the Traveling Salesman Problem," *Proceedings Parallel Problem Solving from Nature III*, Davidor, Y., Schwefel, H-P. and Männer, R. (Ed.), Springer-Verlag.

Eshelman, L.J., Caruna, R.A. and Schaffer, J.D. (1989), "Biases in the Crossover Landscape," Schaffer, J.D. (Ed.), *Proceedings of the 3rd International Conference on Genetic Algorithms and Their Applications*, Morgan Kaufmann Publishers, San Mateo, CA.

Evans, J.R. and Olson, D.L. (2002), *Introduction to Simulation and Risk Analysis*, Second Edition, Prentice Hall, USA.

Fishburn, P. (1970), U*tility Theory for Decision Making*, Publications in Operations Research No. 18, John Wiley and Sons, Inc.

Fogel, D.B. (1991), *System identification Through Simulated Evolution: A Machine Learning Approach to Modeling*, Ginn Press, Needham Heights, MA.

Fogel, D.B. (1994), "An Introduction to Simulated Evolutionary Optimization," *IEEE Trans. Neural Networks*, Vol. 5:1, pp. 3-14.

Fogel, D.B. (1995), *Evolutionary Computation: Toward a new Philosophy of Machine Intelligence*, IEEE PRESS, New York, NY.

Fogel, L.J., Owens, A.J. and Walsh, M.J. (1966), *Artificial Intelligence trough Simulated Evolution*, John Wiley & Sons, Inc.

Fonseca, C. M. and Fleming, P. J. (1993), *Genetic Algorithms for Multiobjective Optimization: Formulation, Discussion and Generalization*, In Stephanie Forrest, editor, Proceedings of the Fifth International Conference on Genetic Algorithms, pp. 416-423, San

Mateo, California, University of Illinois at Urbana-Champaign, Morgan Kauffmann Publishers.

Fonseca, C.M. and Fleming, P.J. (1995), *An overview of evolutionary algorithms in multiobjective optimization*, Evolutionary Computation, 3(1):1-16, Spring.

Forgionne, G.A. (1986), *Quantitative Decision Making*, Wadsworth Publishing Company, Belmont, California.

Forrest, S. (1991), *Emergent Computation: Self-organizing, Collective, and Cooperative Phenomena in Natural and Artificial Computing Networks*, MIT/ North-Holland.

French, S. (1982), *Sequencing and Scheduling: an introduction to the mathematics of Job-Shop*, Ellis Horwood Limited.

French, S. (1986), *Decision theory: an introduction to the mathematics of rationality*, Ellis Horwood series in mathematics and its applications.

Frey, H.C. and Patil, S.R. (2002), "Identification and Review of Sensitivity Analysis Methods," *Risk Analysis*, Vol. 22, No. 3, pp. 553-578.

Garey, M.R. (1979), *Computers and intractability: a guide to the theory of NP-completeness*, New York, W.H. Freeman.

Gero, J.S. and Louis, S.J. (1995), *Improving Pareto Optimal Designs Using Genetic Algorithms*, Microcomputers in Civil Engineering, 10(4), pp. 241-249.

Goicoechea, A., Duckstein, L. and Fogel, M. (1979), "Multiple Objectives Under Uncertainty: An Illustrative Application of Protrade," *Water Resources Research*, Vol. 15, No.2, April, pp. 203-210.

Goicoechea, A., Hansen, D.R. and Duckstein, L. (1982), *Multiobjective decision analysis with engineering and business applications*, John Wiley & Sons, Inc.

Goldberg, D.E. (1989), *Genetic Algorithms in Search, Optimisation and Machine Learning*, Addison-Wesley.

Goldberg, D.E. and Lingle, R. (1985), "Alleles, Loci and the Traveling Salesman Problem," *Proceedings of an International Conference on Genetic Algorithms and Their Applications*, Grefenstette, J.J. (Ed.), Lawrence Erlbaum, Hillsdale, NJ, pp. 154-159.

Goldberg, D.E. and Richardson, J. (1987), *Genetic Algorithms whit sharing for multimodal function optimization*, Genetic Algorithms and Their Applications: Proceedings of the Second International Conference on Genetic Algorithms, pp. 41-49.

Goldberg, D.E., Deb, K. and Horn, J. (1992), "Massive multimodality, deception, and genetic algorithms," *Parallel Problem Solving from Nature 2*, Amsterdam, North-Holland, pp. 37-46.

Goodwin, P. and Wright, G. (1998), *Decision Analysis for Management Judgment*, John Wiley and Sons Ltd, Chichester, England.

Gore, C., Murray, K. and Richardson, B. (1992), *Strategic Decision-Making*, Cassell, London.

Grefenstette, J. (2000), "Proportional selection and sampling algorithms," *Evolutionary Computation 1*, Bäck, T., Fogel, D.B. and Michalewicz, Z. (Eds.), Chapter 23, Institute of Physics Publishing.

Grefenstette, J. (2000b), "Rank-based selection," *Evolutionary Computation 1*, Bäck, T., Fogel, D.B. and Michalewicz, Z. (Eds.), Chapter 25, Institute of Physics Publishing.

Grefenstette, J., Gopal, R., Rosmaita, B. and Van Gucht, D. (1985), "Genetic Algorithms for the Traveling Salesman Problem," *Proceedings of an Intern. Conf. on Genetic Algorithms an Their Applications*, Grefenstette, J.J. (Ed.), Lawrence Erlbaum, Hillsdale, NJ, pp. 160-168.

Gregory, G. (1988), *Decision Analysis*, Pitman Publishing, London.

Haimes, Y.Y (1998), *Risk modelling, Assessment, and Management*, John Wiley and Sons Inc.

Harrison, E.F. (1975), *The Managerial Decision-making Process*, Houghton Mifflin Company.

Herdy, M. (1991), "Application of the Evolutionsstrategie to Discrete Optimization Problems," *Foundations of Genetic Algorithms*, Rawlins, G. (Ed.) Morgan Kaufman Publishers.

Herrera, F., Lozano, M. and Verdegay, J.L. (1998), "Tackling Real-Coded Genetic Algortihms: Operators and Tools for Behavioural Analysis," *Artificial Intelligence Review*, Kluwer Academic Publishers Group, Vol. 12:4, pp. 265-319.

Hertz, D. and Thomas, H. (1983), *Risk Analysis and its Applications*, John Wiley and Sons Ltd.

Hillier, F. and Lieberman, G. (2001), *Introduction to operations research*, McGraw-Hill Higher Education.

Hiroyasu, T., Miki, M., and Watanabe, S. (1999), "Divided Range Genetic Algorithms in Multiobjective Optimization Problems," *Proceedings of International Workshop on Emergent Synthesis*, pp. 57-66, Kobe, Japan, December.

Hoffmeister, F. and Bäck, T. (1990), "Genetic Algorithms and Evolution Strategies: Similarities and Differences," *Parallel Problem Solving from Nature*, H.-P Schwefel and R. Männer (Eds.), Springer-Verlag.

Horn, J. and Nafpliotis, N. (1993), *Multiobjective Optimization using the Niched Pareto Genetic Algorithm*, Technical Report IlliGAl Report 93005, University of Illinois at Urbana-Champaign, Urbana, Illinois, USA.

Hwang, C-L. and Masud, A. S. Md. (1979), *Multiple Objective Decision Making- Methods and Applications*, Lecture Notes in Economics and Mathematical Systems 164, Springer-Verlag.

Ignizio, J.P. (1982), *Linear Programming in single- and multiple-objective systems*, Englewood Cliffs, London: Prentice-Hall.

Jan, A., Yamamoto, M. and Ohuchi, A. (2000), "Evolutionary Algorithms for Nurse Scheduling Problem," *Proceedings of the 2000 Congress on Evolutionary Computation*, San Diego, USA.

Jaumard, B., Semet, F. and Vovor, T. (1998), "A generalized linear programming model for nurse scheduling," *European Journal of Operational Research*, Vol. 107, pp.1-18.

Katayama, K., Hirabayashi, H. and Narihisa, H. (1998), "Performance Analysis of a New Genetic Crossover for the Traveling Salesman Problem," *IEICE Trans. Fundamental*, Vol. E81-A, No. 5 May.

Keeney, R. (1974), "Multiplicative Utility Functions," *Operations Research*, Vol. 22:1, pp.22-34.

Keeney, R. and Raiffa, H. (1976), *Decisions with Multiple Objective: Preferences and Value Tradeoffs*, Series in probability and mathematical statistics, John Wiley and Sons.

Kickert, W. J.M. (1978), *Fuzzy theories on decision-making: A critical review*, Frontiers in systems research, Vol. 3, Martinus Nijhoff Social Sciences Division, Netherlands.

Knowles, J.D. and Corne, D.W. (2000), "Approximating the Nondominated Front Using the Pareto Archived Evolution Strategy," *Evolutionary Computation*, Vol. 8:2, pp. 149-172.

Langford, I.H. (2002), "An Existential Approach to Risk Perception," *Risk Analysis*, Vol. 22:1, pp. 101-120.

Laumanns, M., Zitzler, E. and Thiele, L. (2000), "A Unified Model for Multi-Objective Evolutionary Algorithms with Elitism," *Proceedings of the 2000 Congress on Evolutionary Computation*, San Diego, USA.

Lee, C.C. (1990), "Fuzzy Logic in Control Systems: Fuzzy Logic Controller, Part II," *IEEE Transactions on Systems, Man and Cybernetics*, Vol. 20, No. 2, March/April, pp. 419-435.

Lin, S. and Kernighan, B.W. (1973), "An Effective Heuristic Algorithm for the Traveling Salesman Problem," *Operations Research* 21.

Lis, J. and Eiben, A.E. (1997), "A Multi-sexual Genetic Algorithm for Multiobjective Optimization," *Proceedings IEEE International Conference on Evolutionary Computation*, ICEC'97.

Lohman, R. (1991), "Selforganisation by Evolution Strategy in Visual Systems," *Lecture Notes in Artificial Intelligence 1989, Subseries of Lecture Notes in Computer Science*, Springer-Verlag Berlin Heidelberg.

Mamdani, E.H. and Assilian, S. (1975), "An Experiment in Linguistic Synthesis with a Fuzzy Logic Controller," *International Journal of Man-Machine Studies*, Vol. 7, pp.1-13.

Mamdani, E.H. (1981), "Advances in the linguistic synthesis of fuzzy controller," *Fuzzy Reasoning and its Applications*, Mamdani, E.H. and Gaines, B.R. (Eds.), Academic Press Inc. (London) Ltd.

Mathias, K. and Whitley, D. (1992), "Genetic operators, the fitness landscape and the traveling salesman problem," *Parallel Problem Solving from Nature II*, Männer, R. and Manderick, B. (Eds.), Elsevier Science Publishers.

Michalewicz, Z. (1994), *Genetic Algorithms + Data Structures = Evolution Programs*. Springer-Verlag, New York.

Mitchell, M. (1996), *An introduction to Genetic Algorithms*, A Bradford Book, The MIT Press.

Monarchi, D., Kisiel, C. and Duckstein, L. (1973), "Interactive Multiobjective Programming in Water Resources: A Case Study," *Water Resources Research*, Vol. 9:4, pp. 837-850.

Moore, P.G. and Thomas, H. (1988), *The Anatomy of Decisions*, Second Edition, Penguin Business Books, London.

Mühlenbein, H. (1991), "Evolution in Time and Space – The parallel Genetic Algorithm," *Foundations of Genetic Algorithms*, Morgan Kaufmann.

Nomura, T. and Miyoshi, T. (1995), " Numerical Coding and Unfair Average Crossover in GA for Fuzzy Rule Extraction in Dynamic Environments," *Fuzzy Logic, Neural Networks and Evolutionary Computation, Lecture Notes in Artificial Intelligence No. 1152*, Furuhashi, T. and Uchikawa, Y. (Eds.), Springer, pp. 55-72.

Oka, T., Matsuda, H. and Kadono, Y. (2001), "Ecological Risk-Benefit Analysis of a Wetland Development Based o Risk Assessment Using "Expected Loss of Biodiversity"," *Risk Analysis*, Vol. 21:6, pp. 1011-1023.

Oliver, I.M., Smith, D.J. and Holland, J.R.C. (1987), "A study of permutation crossover operators on the Traveling Salesman Problem," *Proceedings of the Second International Conference on Genetic Algorithms*, Grefenstette, J.J (Ed.) Lawrence Erlbaum Associates, Hillsdale, NJ.

Pidd, M. (1991), "Operations research/management science method," *Operations research in management*, Littlechild, S.C. and Shutler, M.F. (Eds.), Chapter 2, Prentice Hall International (UK) Ltd.

Porto, W. (2000), "Evolutionary programming," *Evolutionary Computation 1*, Bäck, T., Fogel, D.B. and Michalewicz, Z. (Eds.), Institute of Physics Publishing.

Rechenberg, I. (1965), "Cybernetic solution path of an experimental problem," *Library Translation* 1122, Royal Aircraft Establishment, Farnborough, UK.

Reinelt,G. (1994), "The Traveling Salesman, Computational Solutions for TSP Applications," *Lecture Notes in Computer Sciences* Vol. 840, Springer-Verlag.

Romerio, F. (2002), "Which Paradigm for Managing the Risk of Ionizing Radiation?," *Risk Analysis*, Vol. 22:1, pp.59-66.

Roy, B. (1977), "A Conceptual Framework for a Prescriptive Theory of Decision-Aid," *Multiple Criteria Decision Making*, Starr, M. and Zeleny, M. (Eds.), North-Holland Publishing Company.

Saaty, T.L. (1977), "A Scaling Method for Priorities in Hierarchical Structures," *Journal of Mathematical Psychology*, Vol. 15:3, pp. 234-281.

Saaty T.L. (1994), *Fundamentals of Decision Making and Priority Theory with the Analytic Hierarchy Process*, RWS Publications.

Sarker, R. Liang, K-H. and Newton, C. (2002), "A new multiobjective evolutionary algorithm," *European Journal of Operational Research*, Vol. 140, pp. 12-23.

Schaffer, J.D. (1985), "Multiple objective optimisation with vector evaluated genetic algorithms," *Proceedings of the First International Conference on Genetic Algorithms*, Grefenstette, J.J.(Ed.), Lawrence Erlbaum.

Shaffer, C.A. (1998), *A Practical Introduction to Data Structures and Algorithm Analysis*, Prentice Hall.

Shaw, K.J. and Fleming, P.J. (1996), "Initial Study of Multi-Objective Genetic Algorithms for Scheduling the Production of Chilled Ready Meals," Research Report, Department of Automatic Control and Systems Engineering, University of Sheffield, Sheffield, U.K.

Simon, H.A. (1959) "Theories of Decision-Making in Economics," *The American Economic Review*, Vol.49, pp. 252-283.

Srinivas, N. and Deb, K. (1994), "Multiobjective Optimization Using Nondominated Sorting in Genetic Algorithms," *Evolutionary Computation*, Vol. 2:3, pp. 221-248.

Starkweather, T., McDaniel, S., Mathias, K. and Whitley, D. (1991), "A comparison of Genetic Operators," *Proceedings of the Fourth International Conference on Genetic Algorithms*, Morgan Kaufmann.

Syswerda, G. (1989), "Uniform Crossover in Genetic Algorithms," *Proceedings of the 3rd International Conference on Genetic Algorithms and Their Applications*, Schaffer, J.D. (Ed.), Morgan Kaufmann Publishers, San Mateo, CA.

Syswerda, G. (1991), "Schedule Optimization Using Genetic Algorithms," *Handbook of Genetic Algorithms*, Lawrence Davis (Ed.), New York, Van Nostrand Reinhold.

Taha, H.A. (1997), *Operations Research: an Introduction*, Sixth Edition, Prentice-Hall International, USA.

Tannenbaum, R. (1950), "Managerial Decision-Making", *The Journal of Business of the University of Chicago*, Vol. 23:1, pp. 22-39.

Tang, A. Y. and Leng, K. (1994), "A modified Edge Recombination operator for Traveling Salesman Problem," *Proceedings Parallel Problem Solving from Nature – PPSN III*, Davidor, Y. Schwefel, H-P. and Männer, R. (Eds.), Springer-Verlag.

Tao, G. and Michalewicz, Z. (1998), "Inver-over Operator for the TSP," *Proceedings Parallel Problem Solving from Nature – PPSN V*, Eiben, A., Bäck, T., Schoenauer, M. and Schwefel, H.-P. (Eds.), Springer.

Terano, T., Asai, K. and Sugeno M. (1992), *Fuzzy systems theory and its applications*, ACADEMIC PRESS, INC., USA.

TSPLIB, (1995), http://www.iwr.uni-heidelberg.de/iwr/comopt/software/TSPLIB95/

Ulder, N., Aarts, E., Bandelt, H., van Laarhoven, P. and Pesch, E. (1990), "Genetic Local Search Algorithms for the Traveling Salesman Problem," *Proceedings Parallel Problem Solving from Nature*, Schewefel,, H-P. and Männer, R. (Eds.), Springer-Verlag.

Valouxis, C. and Housos, E. (2000), "Hybrid optimization techniques for the workshift and rest assignment of nursing personnel," *Artificial Intelligence in Medicine*, Vol. 20, pp. 155-175.

Vincke, P. (1992), *Multicriteria decision-aid*, John Wiley and Sons, Chichester, England.

Voget, S. and Kolonko, M. (1998), "Multidimensional Optimization with a Fuzzy Genetic Algorithm," *Journal of Heuristics*, Vol. 4, Kluwer Academic Publishers, pp. 221-244.

Vose, M. (1999), *The Simple Genetic Algorithm*, A Bradford Book, The MIT Press, Cambridge, Massachusetts.

Whitley, D., Starkweather, T. and Fuquay, D'A. (1991), "Scheduling Problems and Traveling Salesman: The Genetic Edge Recombination Operator," *Handbook of Genetic Algorithms*, Davis, Lawrence (Ed.), New York, Van Nostrand Reinhold.

Whitley, D., Starkweather, T. and Shaner, D. (1991), "The Traveling Salesman and Sequence Scheduling: Quality Solutions Using Genetic Edge Recombination," *Handbook of genetic algorithms*, Davis, L. (Ed.), Van Nostrand Reinhold.

Yoon, K. and Hwang C-L. (1995), *Multiple attribute decision making: an introduction*, Sage University Paper series on Quantitative Applications in the Social Sciences, 07-104, Thousand Oaks, CA: Sage.

Zadeh, L.A. (1965) "Fuzzy Sets," *Information and Control*, Vol. 8, New York: Academic Press, pp. 338-353.

Zadeh, L.A. (1973) "Outline of a New Approach to the Analysis of Complex Systems and Decision Processes," *IEEE Trans. Systems, Man, and Cybernetics*, SMC-3, pp. 28-44.

Zeleny, M. (1982), *Multiple criteria decision making*, Series in quantitative methods for management, McGraw-Hill.

Zimmermann, H.-J. (1996), *Fuzzy Set Theory-and Its Applications*, Third Edition, Kluwer Academic Publishers.

Zitzler, E. and Thiele, L. (1999), "Multiobjective Evolutionary Algorithms: A Comparative Case Study and Strength Pareto Approach," *IEEE Transactions on Evoltuionary Computation*, Vol. 3:4, pp. 257-271.