1997

ADAPTIVE SEARCH AND THE PRELIMINARY DESIGN OF GAS TURBINE BLADE COOLING SYSTEMS

ROY, RAJKUMAR

http://hdl.handle.net/10026.1/2664

http://dx.doi.org/10.24382/3734
University of Plymouth

ADAPTIVE SEARCH AND THE PRELIMINARY DESIGN OF GAS TURBINE BLADE COOLING SYSTEMS

by

RAJKUMAR ROY

A thesis submitted to the University of Plymouth
in partial fulfilment for the degree of

DOCTOR OF PHILOSOPHY

School of Computing
Faculty of Technology

In collaboration with
Rolls Royce plc., Bristol (UK)

January 1997

Copyright Statement

Rajkumar Roy

(RAJKUMAR ROY)

Dated: 7/5/97

ADAPTIVE SEARCH AND THE PRELIMINARY DESIGN OF GAS TURBINE BLADE COOLING SYSTEMS

by

RAJKUMAR ROY

A thesis submitted to the University of Plymouth
in partial fulfilment for the degree of

DOCTOR OF PHILOSOPHY

School of Computing
Faculty of Technology

In collaboration with
Rolls Royce plc., Bristol (UK)

January 1997

Adaptive Search and The Preliminary Design of Gas Turbine Blade Cooling Systems

Rajkumar Roy

ABSTRACT

This research concerns the integration of Adaptive Search (AS) technique such as the Genetic Algorithms (GA) with knowledge based software to develop a research prototype of an Adaptive Search Manager (ASM). The developed approach allows to utilise both quantitative and qualitative information in engineering design decision making. A Fuzzy Expert System manipulates AS software within the design environment concerning the preliminary design of gas turbine blade cooling systems. Steady state cooling hole geometry models have been developed for the project in collaboration with Rolls Royce plc. The research prototype of ASM uses a hybrid of Adaptive Restricted Tournament Selection (ARTS) and Knowledge Based Hill Climbing (KBHC) to identify multiple "good" design solutions as potential design options. ARTS is a GA technique that is particularly suitable for real world problems having multiple sub-optima. KBHC uses information gathered during the ARTS search as well as information from the designer to perform a deterministic hill climbing. Finally, a local stochastic hill climbing fine tunes the "good" designs. Design solution sensitivity, design variable sensitivities and constraint sensitivities are calculated following Taguchi's methodology, which extracts sensitivity information with a very small number of model evaluations. Each potential design option is then qualitatively evaluated separately for manufacturability, choice of materials and some designer's special preferences using the knowledge of domain experts. In order to guarantee that the qualitative evaluation module can evaluate any design solution from the entire design space with a reasonably small number of rules, a novel knowledge representation technique is developed. The knowledge is first separated in three categories: inter-variable knowledge, intra-variable knowledge and heuristics. Inter-variable knowledge and intra-variable knowledge are then integrated using a concept of compromise. Information about the "good" design solutions is presented to the designer through a designer's interface for decision support.

LIST OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ACKNOWLEDGEMENTS

I am very delighted to acknowledge contributions of the following people:

- My 'director of studies', Dr. Ian C. Parmee, for his technical guidance, valuable discussions, motivation concerning this research. Without his support it would not have been possible to complete the research within a two year period.
- Graham Purchase and Dr. Rob Randall from Rolls Royce. I am grateful to them for their assistance in the model development, continuous technical support, critical assessment of the research and the outcomes during the entire period of the research.
- My other two supervisors, Prof. M. Denham (School of Computing) and Prof. R. Burns (School of Manufacturing, Materials and Mechanical Engineering), for their guidance and critical assessment of the research.
- My family and friends who has sacrificed for my academic pursuits.
- My friends, Dr. Raju Bapi, Harish Vekeria and Stephen Harris, for their assistance in preparing the thesis.
- George Harik and Dr. Samir Mahfoud from Illinois Genetic Algorithms Laboratory (IlliGAL), for their valuable technical discussion over internet.
- All the members of Plymouth Engineering Design Centre, for interesting discussions, for their assistance, and moral support over the last two years. My special thanks go to Andy Phippen for his assistance with UNIX systems and some coding problems. I am specially grateful to Joanne Levers for her continuous assistance and secretarial support.
- Members of School of Computing for their occasional assistance and advice.

The research has used information available over internet. I am grateful to everybody whoever has taken the initiative to promote the internet. The research is jointly funded by University of Plymouth, Rolls Royce and EPSRC (UK). I am very grateful to the above mentioned authorities for their financial assistance.

AUTHOR'S DECLARATION

At no time during the registration for the degree of Doctor of Philosophy has the author been registered for any other University award.

This research was financed with the aid of a studentship from the University of Plymouth, partly funded by, and in collaboration with Rolls Royce plc.

The research has been presented by the author at four relevant technical workshops and conferences; Rolls Royce has been visited regularly for technical discussions and consultation purposes, and several papers prepared for publication. The research used information available over the internet and publications have been made available through on-line internet workshops.

Publications:

Roy, R. and Parmee, I. C.; 1995; 'Adaptive Restricted Tournament Selection for multi-modal function optimisation'; Research Report No.: PEDC-03-95; Plymouth Engineering Design Centre, University of Plymouth, Plymouth (UK).

Roy, R. and Parmee, I. C.; 1996; 'Adaptive Restricted Tournament Selection for the Identification of Multiple Sub-Optima in a Multi-Modal Function'; Lecture Notes in Computer Science (Evolutionary Computing); vol. 1143; Springer - Verlag; pp. 236-256.

Roy, R., Parmee, I. C. and Purchase, G.; 1995a; 'Qualitative evaluation of engineering designs using fuzzy logic'; Research Report No.: PEDC-04-95; Plymouth Engineering Design Centre, University of Plymouth, Plymouth (UK).

Roy, R., Parmee, I. C. and Purchase, G.; 1995b; 'Sensitivity analysis of engineering designs using Taguchi's methodology'; Research Report No.: PEDC-05-95; Plymouth Engineering Design Centre, University of Plymouth, Plymouth (UK).

Roy, R., Parmee, I. C. and Purchase, G.; 1996a; 'Integrating the genetic algorithms with the preliminary design of gas turbine blade cooling systems'; Proceedings of ACEDC'96 Conference; 26-28 March, Plymouth (UK); pp. 228-235.

Roy, R., Parmee, I. C. and Purchase, G.; 1996b; 'Adaptive search manager: an engineering design decision support approach'; Applications of Artificial Intelligence for Technological and Business Processes; vol. 2; University of Plymouth (UK); pp. 46 - 49.

Roy, R., Parmee, I.C. and Purchase, G.; 1996c; 'Sensitivity analysis of engineering designs using Taguchi's methodology'; 96-DETC/DAC-1455; CD-Rom Proceedings of the ASME DETC-Design Automation Conference, 18-22 August, Irvine, CA.

Roy, R., Parmee, I. C. and Purchase, G.; 1996d; 'Qualitative evaluation of engineering designs using fuzzy logic'; 96-DETC/DAC-1449; CD-Rom Proceedings of the ASME DETC-Design Automation Conference, 18-22 August, Irvine, CA.

Rajkumar Roy

(RAJKUMAR ROY)

Dated: 7/5/97

CHAPTER - 1

1. Introduction

1.1 Decision Making in Engineering Design

Decision making is the principle task in engineering design [Starkey (1992)]. The advent of new technologies, especially computer based tools, has helped designers to design a product more efficiently. The new technologies are mostly useful in automating routine tasks involved in the design process. Decision making is still very much left to the designers. The ever growing competition in the market place and increasing expectation of the users are adding many more dimensions to the design decision making process. Thus decision making is becoming increasingly complex. With the advancement in technology the demography of the work force is also changing. Designers with many years of experience in one area are becoming an extinct species. When facing the realities of increasing complexity, some designers with relatively less experience find decision making difficult. The designers often face a hard dead line in which to produce an efficient design that has improved functionality and reduced costs. The nature of the challenge varies according to the stage of a design. A design process generally starts at the conceptual level, and that stage is known as *conceptual* design. Conceptual design is very abstract and approximate, but determines a framework for the design. This stage of design process involves knowledge from different aspects of a design, and can be considered the most innovative stage in the design process. Once the general framework is identified, the next stage is *preliminary* design. Preliminary design is less abstract and more detailed than the conceptual stage. As a result of the preliminary design, an approximate design solution is selected, and subsequently fine tuned during the next *detailed* design stage. Detailed design involves rigorous design analysis that fine tunes the preliminary design.

At every stage of a design process, the designer has to select one solution from a number of alternatives [Smith and Browne (1993)], and thus an initial design is optimised. The design process can be described as a *divergent-convergent phenomenon*. At the initial divergence stage of a design many alternative solutions are generated. The designer then converges to (selects) only one solution, and this stage of the design is known as the convergence stage. The designer's decision in one stage of a design can significantly influence the outcome of the next stage of the design. A wrong decision at one stage of a design can eventually produce a final design solution with low performance [Sherwin (1982)]. With the increasing complexity in the marketplace, design decision making is becoming much more difficult. Designers are often expected to evaluate a design from many different considerations and then select the best suited solution. These criteria may be contradictory to each other. Some of the criteria can be *quantitative* whereas others can be *qualitative* in nature. Time available for the decision making is continuously reducing due to market competition. Thus, designers often have to deal with a vast amount of information for decision making within a short period of time which may cause cognitive overload.

Decisions made by a designer during the design process can be divided into three categories: *Fundamental*, *Intermediates* and *Minors* [Starkey (1992)]. Fundamental decisions are the most important decisions among the three. This category of decisions is absolutely crucial for the success of the design project. The fundamental decisions determine the principal components of a design which form a foundation. Other non fundamental decisions are developed from this foundation to fine tune the design. The intermediate and minor types of decisions are less important than the fundamental decisions. Minors are relatively unimportant decisions that have little effect upon the design performance. The minor decisions are most often concerned with design details.

Design can also be considered to represent a process that begins with a recognition of the need and the conception of an idea to meet this need [Balachandran (1993)]. Thus, in design decision making the main aim of the designer is to find a design solution that meets or closely meets the performance requirements of the design, while satisfying all the constraints. That defines a concept of 'optimum design' as a design that is feasible and also superior to a number of other feasible alternative designs. There are two ways to obtain an optimum design: through an iterative process or by solving an optimisation problem. The iterative process improves a design by repeated modifications. The design variables are changed one at a time. Designers often use their previous experience to decide changes in the design variables. They may easily improve a design involving few variables. If the design involves many variables this can pose a great challenge to the human designer, especially if he or she needs to consider variable interaction. If the designer does not have prior knowledge about the design the iterative process can simply become a trial and error exercise. Thus the iterative approach can be very time consuming and tedious. On the other hand, the second approach (i.e. solving an optimisation problem) can simultaneously determine all the design variables so as to satisfy a set of constraints and optimise a set of objectives. To solve an optimisation problem a computable design model is required. Many aspects of a design process can be represented by a formal model and are thus computable. On the other hand, some of the required designer's knowledge can be very abstract and complex, and thus can not be formalised. A design therefore can involve computable or quantitative formal knowledge as well as qualitative or abstract knowledge. In the absence of a formal model of the design process or at least a partial model, the iterative approach may often become the only choice.

Designers typically require much information for design decision making. Information is collected from the laws of physics, previous experiences, available literature, logical deductions and designers' intuition. Some of the information may be imprecise and ambiguous, whereas some may be precise and structured. The designer often faces a challenge to manipulate this combination of precise and imprecise information in order to reach a decision. To achieve good decisions, the designers must be able to take an overview of the possible alternative actions at any point in the design process. The designers can then predict the results of more than one selected course of action. The predictions can be heavily influenced by various other industrial factors and also the market environment. For example, predictions about a design action can be affected by the impact of that decision on the manufacturing organisation responsible for implementing that decision and on the end user (that is the customer). The impact of the decision on the overall market (that is the market environment within which the industry operates) also needs to be assessed. With the dynamic nature of the industrial and market environment in many cases it becomes almost impossible to predict the outcome of a decision very precisely. Design decisions that use precise information from historical data, scientific evidence, etc. can be said to be virtually certain. The decisions that involve designers' knowledge, intuition and judgement involve a certain degree of uncertainty. Uncertainty can also be caused due to the complex dynamic interactions within the industry, between the industry and the market environment, imprecision involved in the designers' knowledge and vagueness involved in the designers' language. It is observed that designers often use their higher level knowledge and intelligence to perform the decision making even in the presence of high uncertainty [Balachandran (1993), Tong and Sriram (1992), Suh (1990), Green (1992), Coyne et. al.

(1990)]. The research reported in this thesis tries to address some of the issues involved in design decision making. The adaptive search manager is a systematic approach to provide relevant information to the designer so that the decision making can be facilitated and cognitive overload can be minimised.

The research reported in this thesis is intended to provide a framework for the development of a design decision support system for the preliminary design stage of a gas turbine blade cooling system. The system is developed to provide relevant information concerning alternative design solutions to the designer. The information is utilised by the designer to select one design solution for the cooling system. The preliminary design stage involves a coarse model of the cooling system, so the selected design is approximate and would need fine tuning in the detailed design stage. The objective of this exercise is to rapidly identify the most interesting design direction [Parmee (1993), Parmee(1994)] that is then utilised in the detailed design stage.

1.2 Engineering Design Decision Support

Chandrasekaran (1990) describes a design problem as a search problem in a large space for objects that satisfy multiple constraints. An object in the design space is equivalent to an acceptable value of a design variable. Only a very small number of objects in this space constitute satisfying, not to mention optimal, solutions. In order to make design decisions, practical strategies that radically shrink the search space are needed. A good design decision support tool can assist a designer in the search space reduction. The first step towards the search space reduction is to separate the information required for a design into two categories: *formal* and *non-formal*. The information obtained from the laws of physics, design catalogues, and design archives is structured and probably computable. Thus the information can be considered as contributing towards formal knowledge. The designer's experience, intuition and judgement can be very abstract, unstructured and incomplete, thus they constitute the non-formal knowledge.

It is observed that engineering designers can often handle formal and non-formal knowledge separately. There are many numerical optimisation techniques [Goldberg (1989), Fonseca and Fleming (1995), Srinivas and Deb (1995), Pham and Yang (1993a) and (1993b)] that can be used for design decision making. Numerical optimisation techniques consider formal knowledge only. Yang and Sen (1994) describe an interactive multiple objective decision making procedure. The process describes a multiobjective preliminary design problem as a non-linear vector maximisation problem. The technique defines the design model using some computable functions. The methodology is a learning-oriented interactive technique that supports the designer in easily searching for preferred solutions following an adaptive approach. The technique allows designer's preferences to be progressively articulated with the generation of efficient design solutions. Through designer interaction the technique also makes sure that no unacceptable solution is selected as a preferred design. Numerical optimisation methods can provide the designer with multiple preferred solutions and thus reduce the search space for the designer. Design decision making with non-formal knowledge can be a very difficult task. Many attempts have been made to represent non-formal knowledge as production rules [Balachandran (1993), Coyne et. al. (1990), Green (1992), Tong and Sriram (1992)]. Production rules can then be used with a Knowledge Based System to provide support in design decision making. Balachandran (1993) identified the following major advantages of knowledge based systems:

a. Knowledge based systems provide a flexible environment which can incorporate designers' knowledge, heuristics and rules of thumb;
b. Knowledge based systems allow symbolic as well as numeric manipulation of information; and

c. They have the ability to reason using the knowledge explicitly incorporated within them.

Knowledge based systems model a design problem using qualitative knowledge. Thus the system is capable of providing a qualitative evaluation of a design. The designer then uses only the evaluation information in decision making and thus faces minimal cognitive overload. Designs often require both qualitative (that can be considered as knowledge based) and quantitative (that is numerical) computation. Thus a collaboration among different types of programs (knowledge based, algorithmic, symbolic and numerical) written in different languages is essential for effective design decision support [Balachandran (1993)].

Knowledge based systems attempt to represent the qualitative knowledge involved in a design process. Fuzzy Expert Systems [Durkin (1994)] have made the task easier by modelling the knowledge using a language closer to that of the designer. Quality of the decision support provided by a knowledge based system depends on the quality of knowledge embedded in the system. Knowledge is formalised from expert designers using a knowledge elicitation technique. It is observed that there is always a gap between the designers' knowledge and the knowledge extracted from the designers using a knowledge elicitation technique. The reason is that the designers think differently when they try to express the strategy followed during a previous design decision [Dreyfus and Dreyfus (1986), Bapi and Denham (1996), Bapi et. al. (1996)]. There is always a mismatch between implicit thinking (when a decision is taken) and explicit thinking (when the designer tries to express the reasoning behind the decision). Thus knowledge based systems can never capture the complete knowledge. A knowledge based system along with other numerical optimisation techniques can support a design decision making process, but the final decision needs to be taken by a human designer.

The research presented in this thesis initially uses a numerical technique such as a genetic algorithm and a hill climbing hybrid to identify multiple "good" design solutions for the turbine blade problem. The hybrid system starts with many randomly generated possible design solutions and this can be viewed as the divergence stage of the design process. Then the search converges to multiple "good" design solutions. The sensitivity of each of the "good" designs is calculated. A fuzzy expert system qualitatively evaluates these designs considering the manufacturability, choice of materials and designer's special preferences as three different criteria. The multiple design options along with the relevant quantitative and qualitative information are presented to the designer for the final selection. Thus the divergent-convergent design process is completed with the designer's participation.

1.3 The Adaptive Search Manager

Engineering design often involves several objectives. True engineering design solutions are not necessarily the global optimum as described by some mathematical simulation with respect to one criterion [Parmee (1994), Parmee and Denham (1994)]. Often designer interaction is required to take many different criteria into account. In the case of multimodal design problems there may be quite different design solutions that perform quantitatively similar, but have large differences in their degree of multi-criteria satisfaction. Criteria may include manufacturability, choice of materials, maintainability, reliability, specific customer requirement, designer's special preferences, etc., many aspects of which can be qualitative in nature. Integrating all of these criteria into one comprehensive evaluation function is difficult and at times misleading. If the criteria are quantitative in nature a multiobjective genetic algorithm can be utilised [Goldberg (1989), Fonseca and Fleming (1995), Srinivas and Deb (1995)]. The research reported in this thesis attempts to obtain multiple "good" design solutions based on the most important quantitative criteria. Sensitivities of each of these design solutions are then calculated. The "good" design solutions are qualitatively evaluated for other less well defined criteria. The final decisions are left to the designer.

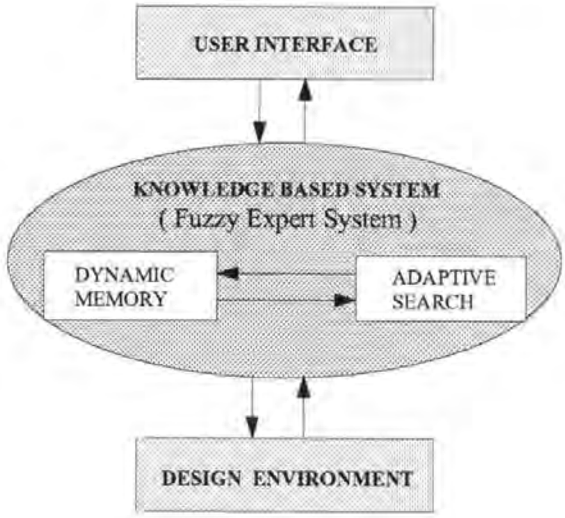


Figure 1.1: The Adaptive Search Manager. The figure exhibits different components of the system and how they interact with each other.

An Adaptive Search Manager (ASM) (Figure 1.1) is developed by integrating a Genetic Algorithm (GA) [Goldberg (1989)] , an Adaptive Search technique, with Knowledge Based Software. The ASM comprises of a fuzzy expert system manipulating GA software within the design environment of the preliminary design of gas turbine blade cooling systems. A steady state cooling hole geometry design model has been developed for the research in collaboration with Rolls Royce PLC. The model can evaluate a cooling system design solution quantitatively. ASM is an integrated system which consists of a fuzzy expert system manipulating the adaptive search technique and interacting with a dynamic memory. ASM extracts the following information from the search process for the turbine blade problem, which is then processed and presented to the designer:

i) multiple "good" design solutions

ii) design solution sensitivity

iii) design variable sensitivities

iv) constraints sensitivity

v) qualitative ratings of the "good" design solutions

ASM uses a hybrid system of an Adaptive Restricted Tournament Selection (ARTS) [Roy and Parmee (1995) and (1996), and Roy et. al. (1996a)] based GA and a knowledge based local hill climbing technique to identify multiple "good" design solutions (multiple sub-optima) with respect to the amount of coolant mass flow. By identifying multiple "good" designs the novel hybrid search technique considerably reduces the quantitative design search space for the designer. Sensitivity information concerning the neighbourhoods of the "good" designs is obtained using Taguchi's methodology. The method is capable of providing nearly accurate sensitivity information about the neighbourhoods provided that no interaction between variables can be assumed within local regions. A local region is defined by the tolerance on each dimension and Taguchi's orthogonal matrix. The methodology provides a computationally inexpensive way of calculating the sensitivities. The designs are then qualitatively evaluated using a fuzzy expert system to ascertain qualitative ratings in terms of manufacturability, choice of materials and designer's special preferences. The developed qualitative evaluation system utilises domain knowledge concerning inter-variable preferences, intra-variable preferences and heuristics. Inter-variable preferences are combined with intra-variable preferences using a concept of compromise [Roy et. al. (1995a)]. The concept of compromise has been defined as "reducing the severity of the negative effect of one variable on the final qualitative rating". This novel knowledge representation technique has helped to cover the entire design space with a small number of rules.

One part of the memory is *static* i.e. it holds the expert knowledge regarding several qualitative aspects of the design thereby providing a qualitative model of the design problem. The other part is *dynamic* retrieving information during the adaptive search design process. The system interacts with the models of the design environment that evaluate every single design solution both quantitatively and qualitatively. An Adaptive Search Manager interface has been developed using the Xview facility in the UNIX system. The interface provides flexibility to change the boundaries of the design variables and that of the constraints at the beginning of a search process. The design manager is used as a decision support tool where the final selection of a design option is left to the designer.

Information about the "good" design solutions is then presented to the designer. The overall objective is to provide as much relevant information as possible to the designer for the decision support. The decision support utilises the knowledge of many experts and at the same time can enhance the knowledge of some inexperienced designers. The presentation of relevant information concerning the "good" designs also helps in minimising any cognitive overload on the designer. The approach developed in this thesis is expected to result in the achievement of optimal *engineering* solutions [Parmee and Denham (1994), Parmee et. al. (1994)] at the preliminary design stage.

1.4 Overview of the Thesis

The thesis is divided into eight chapters. This chapter introduces principal issues in engineering design decision making. Then, Chapter 2 narrates the development of a preliminary design model of a gas turbine blade cooling system. The physics and the domain knowledge involved in the development are also elaborated. The model has been developed in collaboration with Rolls Royce plc. The chapter describes all the terminology used in the model development, the inputs and outputs of the model and finally the constraints on the model. A step-by-step description of the model development describes the physics and the iterative process involved in the design. Some parts of the model reflect design practice (not necessarily Rolls Royce's current practice) present in the industry. The chapter concludes with some insight into the nature of the model in unconstrained and constrained situations.

The adaptive search manager uses an adaptive search technique to partially represent the divergent-convergent phenomenon in the design. The adaptive search technique is a hybrid comprising of a Genetic Algorithm based search and a knowledge based local hill climbing method. The type of the genetic algorithm used is known as a multimodal genetic algorithm. Chapter 3 introduces multimodal genetic algorithms. The chapter starts with a brief description of genetic algorithms including the basic principles and the theory. Then the chapter describes how a variant of the genetic algorithm can be used to locate multiple sub-optima in a multimodal function. The chronological development of multimodal genetic algorithm is discussed. The discussion identifies the limitation of existing multimodal genetic algorithms in the case of real life problems. Characteristics of real life problems are discussed and the challenge presented by real life problems is defined.

Chapter 4, describes a novel multimodal genetic algorithm that is suitable for real life problems. The developed technique is known as adaptive restricted tournament selection. The chapter describes the algorithm and the different issues involved in the technique. A comparison of the technique is performed with two other recent multimodal genetic algorithms. The comparison is performed on test functions and the results are presented and discussed. A further analysis of the developed technique is performed to understand the effects of a critical parameter on the performance of the technique. Results from the analysis are presented and discussed. Next, the adaptive restricted tournament selection technique is applied to the turbine blade cooling system design problem in order to identify multiple "good" design solutions. A design is considered "good" if the performance of the design is better than similar (that is closer in terms of the design variables) designs. The chapter describes in detail the steps involved in the application. The characteristics of the technique that help to handle the issues involved with real life problems are discussed. Some improvements that are adopted in the search technique to reduce the total design time are also explained. Knowledge gathered during the search process and the designers' prior knowledge concerning the design variables are utilised by a Knowledge Based Hill Climber to fine tune the important design variables of the "good" designs. The chapter describes the rationale behind using such a hill climbing technique along with the multimodal genetic algorithm based search. The principle and the methodology behind the hill climbing technique are presented. This chapter explains how the hybrid of the multimodal genetic algorithm based search and the hill climbing works for the cooling system design problem. Once the hybrid search technique identifies several "good" designs, further fine tuning of the designs are performed using a stochastic local hill climbing technique. The stochastic hill climbing algorithm is also presented in the chapter.

The "good" designs are next analysed for design sensitivity information. Chapter 5 describes the sensitivity analysis method developed for this research. The analysis is performed in a neighbourhood of a design solution. Taguchi's orthogonal matrix and the tolerances on the design variables define the neighbourhood of a design solution. It is assumed that the neighbourhood can be approximated as a small region where there is no or very little interaction among the design variables. Taguchi's methodology, a technique for experimental design, is followed to calculate three categories of sensitivity information: *design solution sensitivity*, *design variable sensitivity* and *constraint sensitivity*. The use of Taguchi's methodology enables the calculation of sensitivity information with a very small number of the cooling system model evaluations. The chapter starts with a brief introduction to design of experiments and Taguchi's technique. The principle behind Taguchi's orthogonal matrix is discussed. The chapter then describes the development of an orthogonal matrix that is suitable for the design problem. The use of this orthogonal matrix to define different categories of sensitivity information are presented in the next section. The neighbourhood of each design solution is checked for interaction. The sensitivity calculations are accepted only if there is no significant interaction between the design variables within the region. The sensitivity information is close to reality if the minimal interaction assumption is correct. In order to validate this notion, Taguchi's methodology based sensitivity calculation result is compared with the sensitivity analysis using an exhaustive search. The comparison results are presented and discussed.

Chapter 6 presents the qualitative evaluation of the design solutions. The "good" designs are evaluated for different qualitative criteria: manufacturability, choice of material and designer's special preferences. The evaluation technique uses a fuzzy expert system to obtain three qualitative ratings (that is three crisp numbers) for the three criteria. The chapter introduces the concepts of fuzzy logic and fuzzy expert systems. Different components of a fuzzy expert system are also discussed. A description of the Qualitative Evaluation System developed for the design problem is also given. The chapter explains different components of the system and discusses the principal issues involved. A novel knowledge representation technique is developed that guarantees the evaluation of any possible design solution with a reasonably small number of rules. Knowledge is first separated into several categories and then integrated using *a concept of compromise*. The chapter provides a detailed description of the knowledge representation technique and also discusses the motivation behind the approach. The qualitative evaluation system uses FuzzyCLIPS, a fuzzy logic version of CLIPS (developed by NASA). FuzzyCLIPS is a fuzzy expert system shell from National Research Council, Canada. The terminology and syntax used in the examples follow FuzzyCLIPS standards. The chapter also discusses how to integrate the FuzzyCLIPS based qualitative evaluation system with the adaptive search technique mentioned before. Finally, the chapter is concluded with the description of the validation procedure for the qualitative evaluation system.

The adaptive search manager identifies several "good" design solutions, and then retrieves some additional quantitative and qualitative information about the designs. The multiple "good" designs along with the additional information are presented to the designer through an adaptive search manager interface. This information supports the designer in design decision making. The adaptive search manager is executed with different conditions (that is unconstrained and constrained) and with different settings for the design variable ranges and the constraints. Representative results from these experiments are reported in Chapter 7. It is difficult to validate a system involving real life problems. The results from the experiments are validated by an expert and a user from Rolls Royce. A questionnaire (Appendix I) is prepared to assist in the validation. The chapter concludes with a description of the evaluation procedure adopted for the adaptive search manager.

The final chapter, Chapter 8, provides a detailed discussion on the results reported in the previous chapter and also on the techniques developed in this thesis. The chapter also presents the conclusions from the research and the scope of future research.

The thesis assumes that the reader has some preliminary background in engineering design, genetic algorithms and fuzzy expert systems. An attempt has been made to briefly introduce engineering design decision making, genetic algorithms, Taguchi's methodology and fuzzy expert systems before or in the relevant chapters. For a detailed study on any one of these topics a comprehensive list of references is provided in the thesis.

CHAPTER - 2

2. The Development of a Preliminary Design Model of a Turbine Blade Cooling System

2.1 Introduction

In order to maximise gas turbine engine performance and efficiency, turbine blades need to operate in an environment where the gas temperature is as high as possible. This temperature often exceeds the operational limits of the turbine blade materials. To ensure component integrity whilst operating at high gas temperatures blade materials are cooled to safe operating temperature levels by passing relatively cool air through them and, in more extreme cases, over them in the form of films. A small portion of the compressor exit airflow is utilised to cool the blades (Figure 2.1). The temperature of this cooling air depends on the compressor pressure ratio and on the flight Mach number and temperature. The sacrifices for the blade cooling include loss of work (and some loss of efficiency) due to the portion of the air taken from the compressor exit. Thus one of the objectives of the Adaptive Search Manager (ASM) is to try to minimise the amount of airflow (hence referred to as coolant flow) required for the blade cooling. In general, however, these losses are much smaller than the gains associated with operating the engine at much higher turbine inlet temperature than would be possible without the blade cooling.

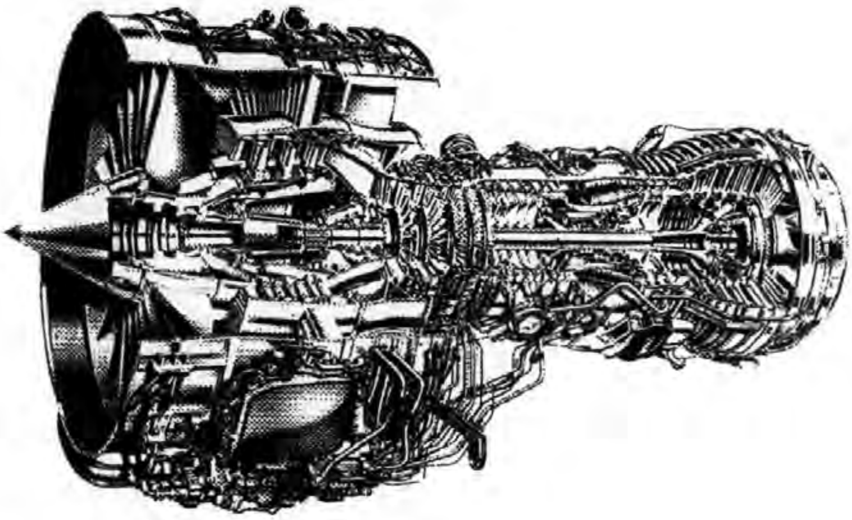


Figure 2.1(a): At-a-glance: a large twin-spool turbofan engine [Cohen et. al. (1987)].

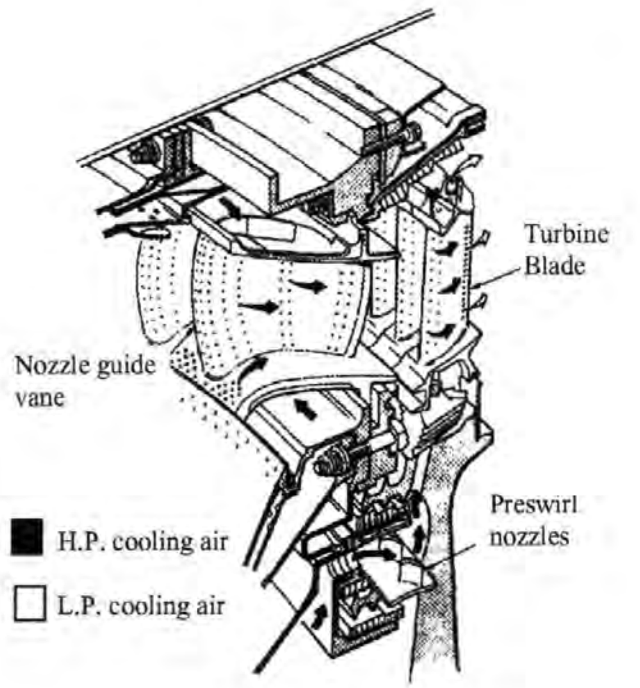


Figure 2.1(b): A section showing the cooled high-pressure turbine stage [Hill and Peterson (1992)].

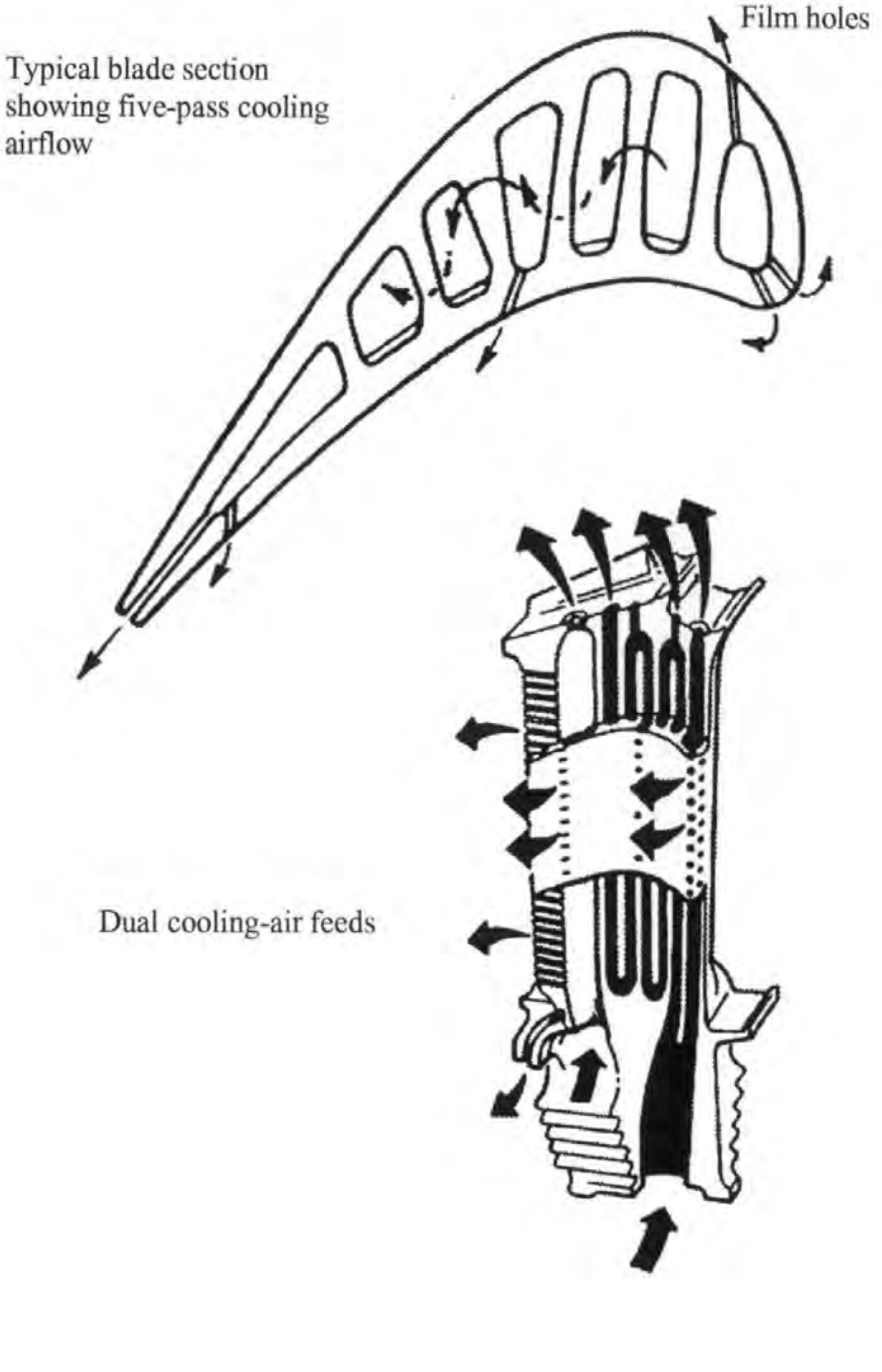


Figure 2.2: The general arrangement of five-pass cooling of a turbine rotor blade [Hill and Peterson (1992)].

A preliminary design model of the cooling system has been developed in collaboration with Rolls Royce plc. (Bristol, UK) and Plymouth Engineering Design Centre. The model is developed considering one dimensional, single pass coolant flow. This represents a computationally inexpensive mathematical model of the blade cooling system. The model includes a film cooling mechanism (Figure 2.2) and involves twelve design variables. This Turbine Blade COoling system Model (TBCOM) also uses several constants known as *design parameters*. The values of the constants have been set by the design experts from Rolls Royce plc., but may not represent the current practice in the company. TBCOM also includes three non-linear constraints. ASM utilises the model to provide quantitative evaluation of the cooling system performance.

This chapter explains the terminologies used in the model development, describes step-by-step development of the model, and finally gives some light on the nature of the model in unconstrained and constrained situations.

2.2 Nomenclatures used in the Model Development

The list of nomenclature used in the model development is presented below. Some of the symbols are standard engineering terms, but others are specific to this thesis. Please refer to Figure 2.3 for the general arrangement of the coolant flow with film cooling.

A: Cross sectional area of passage

C_d: Coefficient of discharge

C_p: Specific heat at constant pressure

C_v: Specific heat at constant volume

d: Hydraulic diameter

dth: Wall thickness

h: Heat transfer coefficient

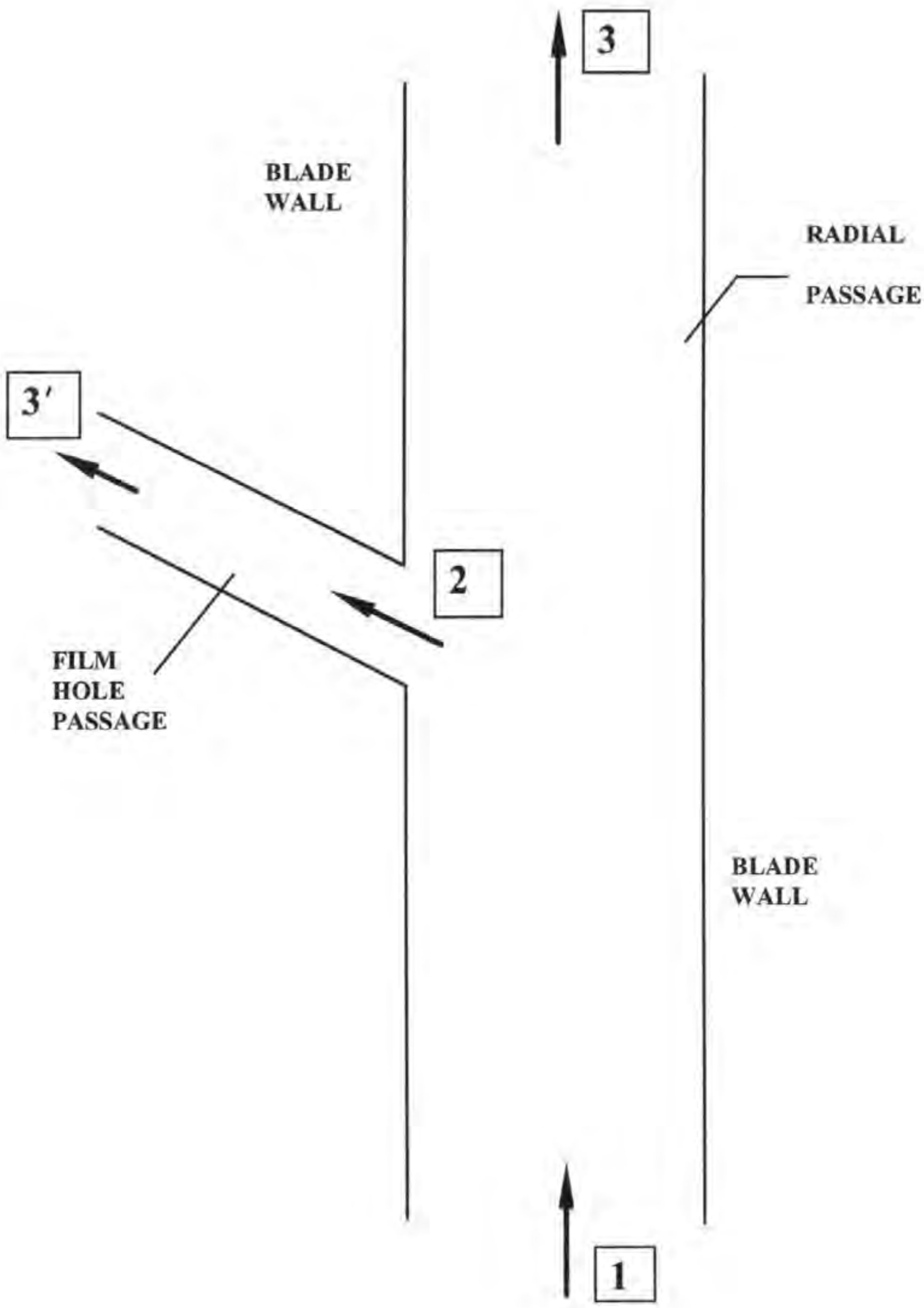


Figure 2.3: A schematic diagram showing the general arrangement of the coolant flow through a turbine blade with a film cooling mechanism. Where, 1: cooling air inlet, 2: film cooling passage inlet, 3: cooling air exit and 3′: film cooling hole exit.

H1: Parameter group for heat balance equation

H2: Parameter group for heat balance equation

H3: Parameter group for heat balance equation

k: Thermal conductivity

l: passage length

M: Mach number

N: Number of..

Pc: Cooling air pressure

R: Gas constant

S_c: Cooling side perimeter

S_g: Gas side effective perimeter

Tc: Cooling air temperature

W: Mass flow

X_F: Distance from film cooling hole exit / Effective slot width of film

γ: Ratio of specific heats

μ: Dynamic viscosity

Subscript:

1: cooling air inlet

2: film cooling passage inlet

3: cooling air exit

3′: film cooling hole exit

b: blade

c: coolant

f: film

g: gas

hpc: high pressure compressor

r: radial passage

w: wall

2.3 Constants used in the Model:

Like many other design models TBCOM involves several constants known *as design parameters*. The design parameters are selected by experts from Rolls Royce from their experience and knowledge in the area. This helps to limit the complexity of the model by fixing the values of some variables. One such example is the number of film holes (designated by N_f). The design parameter values with their respective nomenclature are:

1. Heat transfer coeff. (gas side), $h_g = 3000.0\ W/m^2\ K$
2. Gas side temperature, $T_g = 1500.0$ K
3. Ratio of specific heats, $\gamma = 1.36$
4. Mass flow (high pressure compressor), $W_{hpc} = 84.85\ Kg/s$
5. Radial cooling hole exit pressure, $Pc_3 = 460000.0\ N/m^2$
6. Number of blades, $N_b = 78$
7. Wall temperature (gas side) for initial calculations, $T_{wg} = 1250.0$ K
8. Radial passage length, $l_r = 0.0406$ m
9. Specific heat at constant pressure, $C_p = 993.0$
10. One of two factors for heat transfer coefficient, $F = 0.01855$
11. Gas constant, $R = 287.0$
12. Distance from film cooling hole exit/Effective slot width of film, $X_F = 10$
13. Mach Number, Mach = 0.6
14. Number of film holes, $N_f = 30$
15. Initial outside temperature, $T_{wg1} = 1500.0$ K

16. Maximum radial passage area, $A_{cr} \leq 2.75E\text{-}05\ m^2$

17. Bounds on radial coolant flow heat transfer coefficient,

$$100.0W/m^2K \langle h_{cr} \langle 4000.0W/m^2K$$

18. Check on metal temperature, $1000.0K \langle T_{wg} \langle 1500.0K$

19. For the film cooling section, heat transfer coefficients are the same for the film side and the gas side, that is:

$$h_f = h_g$$

20. For the film cooling section, the perimeter ratio, $R_{sf} = 1.0$

2.4 Nomenclature for the Model Input and Output

Twelve design variables are input to TBCOM and there are four outputs. The principle objective is to minimise mass flow (designated by W_{cr}) through the radial passage of the blade. Constraints are set on the other three outputs, that is each output should lie within a predefined range of values. The nomenclature for the inputs and outputs are as follows:

INPUTS:

1. Type of geometry, Geom

   There are three discrete types of geometry involved: plane, ribbed and pedestal.

2. Coefficient of discharge (radial passage), C_{dr}

   The value of C_{dr} varies within a range according to the type of geometry.

3. Heat transfer coefficient factor (radial passage), Fhc

   The value of Fhc varies within a range according to the type of geometry.

4. Inlet temperature, Tc_1 (K)

5. Wall thickness, dth (m)

6. Thermal conductivity of the blade material, k_w (wK/m^3)

7. Pressure ratio (between inlet and outlet of radial passage), R_p

where $R_p = Pc_1 / Pc_3$.

8. Perimeter ratio (radial passage), R_s

   where $R_s = S_{gr} / S_{cr}$.

9. Film hole diameter, df (m)

10. Coefficient of discharge (film hole), C_{df}

11. Heat transfer coefficient factor (film hole), F_f

12. Pressure ratio (film), R_{pf}

    where $R_{pf} = (Pc_1 - Pc_2) / (Pc_1 - Pc_3)$.

OUTPUTS:

1. Coolant mass flow (radial passage), W_{cr} (Kg/s)

2. Coolant mass flow (film hole), W_{cf} (Kg/s)

3. Metal temperature (gas side), T_{wg} (K)

4. Metal temperature (film side), T_{wf}(K)

2.5 Model Development

The model is developed considering coolant flow through the radial passage of a turbine blade and the flow through film holes. The coolant air passes through the film holes and spreads over the blade as a thin film of cooler air, and thus provides additional cooling to the blade. The model development uses the basic principles of physics, but some of the design parameters are set from domain knowledge. This section describes the step-by-step procedure followed to establish a relation between the input variables and the outputs.

2.5.1 Calculation of the relationship between the Mass Flow and the Pressure Ratio

The first task in the model development is to establish a general relation between a fluid mass flow (that is the coolant flow in this case) and the pressure differential that drives the fluid. The relationship between mass flow and pressure ratio for an idealised, steady, one dimensional, compressible flow can be calculated as follows:

The steady flow energy equation (SFEE) may be expressed as:

$$Q' + h_1' + \frac{V_1^2}{2.0} + z_1 = W' + h_2' + \frac{V_2^2}{2.0} + z_2 \qquad ..... (2.1)$$

where,

Q ' = heat transfer

W ' = work done

h_1 ' and h_2 ' = enthalpy

V_1 and V_2 = velocity

z_1 and z_2 = energy due to elevation

If the flow is brought to rest isentropically over an infinitesimally small distance then,

dQ ' = dW ' = 0

dz = negligible

$V_2 = 0$ (thus $dV = V_1$)

dh ' = $C_p$dt

where C_p = specific heat at constant pressure

t = static temperature

Hence the SFEE (equation (2.1)) reduces to:

$$C_p dt + d\left(\frac{V^2}{2}\right) = 0 \qquad .....(2.2)$$

Integrating equation (2.2) gives:

$$C_p(t_1 - t_2) + \frac{V_1^2}{2} = 0 \qquad .....(2.3)$$

For an adiabatic process $T_1 = T_2 = t_2$, where T_1 and T_2 are stagnation temperatures (that is the summation of static and dynamic temperatures).

Substituting T_1 for t_2 in equation (2.3) and rearranging gives:

$$V_1 = \left[2C_p\left(T_1 - t_1\right)\right]^{0.5} \qquad .....(2.4)$$

An expression for velocity can also be obtained from mass flow continuity as:

$$W = \rho A V \qquad .....(2.5)$$

where, ρ = density of the coolant

A = flow cross sectional area

And from the perfect gas relationship:

$$\rho = \frac{p}{Rt} \qquad .....(2.6)$$

where, p = static pressure

R = universal gas constant

From equations (2.5) and (2.6):

$$V = \frac{WRt}{Ap} \qquad .....(2.7)$$

Equating equations (2.4) and (2.7), and generalising V_1, T_1 and t_1 by V, T and t in equation (2.4):

$$\frac{WRt}{Ap} = \left[2C_p(T - t)\right]^{0.5} \qquad .....(2.8)$$

Rearranging the equation (2.8) in terms of temperature ratio:

$$\frac{WR\sqrt{t}}{Ap} = \left[2C_p\left(\frac{T}{t} - 1\right)\right]^{0.5} \qquad .....(2.9)$$

C_p is a function of the universal gas constant, R; where R can be expressed as:

$$R = C_p - C_v$$

Substituting for the ratio of specific heats, $\gamma = C_p/C_v$, gives:

$$C_p = \frac{R\gamma}{\gamma - 1} \qquad .....(2.10)$$

Substituting for C_p from equation (2.10) in equation (2.9) gives:

$$\frac{WR\sqrt{t}}{Ap}=\left[\frac{2R\gamma}{\gamma-1}\left(\frac{T}{t}-1\right)\right]^{0.5}$$

Rearranging the above equation gives:

$$\frac{W\sqrt{T}}{Ap}=\left[\frac{2\gamma}{R(\gamma-1)}\left(\frac{T}{t}\right)\left(\frac{T}{t}-1\right)\right]^{0.5} \qquad .....(2.11)$$

Further, for an adiabatic process:

$$\frac{p}{\rho}\gamma=const.$$

Therefore, using the perfect gas relationship from equation (2.6), gives:

$$\frac{T}{t}=\left(\frac{P}{p}\right)^{\frac{\gamma-1}{\gamma}} \qquad .....(2.12)$$

where, P = stagnation pressure

Substituting for T/t in equation (2.11) gives:

$$\frac{W\sqrt{T}}{Ap}=\left[\frac{2\gamma}{R(\gamma-1)}\left(\frac{P}{p}\right)^{\frac{\gamma-1}{\gamma}}\left(\left(\frac{P}{p}\right)^{\frac{\gamma-1}{\gamma}}-1\right)\right]^{0.5} \qquad .....(2.13)$$

Further,

$$\frac{W\sqrt{T}}{AP}=\frac{W\sqrt{T}}{Ap}\left(\frac{P}{p}\right)^{-1}$$

$$=\left[\frac{2\gamma}{R(\gamma-1)}\left(\frac{P}{p}\right)^{\frac{\gamma-1}{\gamma}}\left(\left(\frac{P}{p}\right)^{\frac{\gamma-1}{\gamma}}-1\right)\right]^{0.5}\left(\frac{P}{p}\right)^{-1}$$

Hence,

$$\frac{W\sqrt{T}}{AP}=\left[\frac{2\gamma}{R(\gamma-1)}\left(1-\left(\frac{P}{p}\right)^{\frac{1-\gamma}{\gamma}}\right)\left(\frac{P}{p}\right)^{-\frac{2}{\gamma}}\right]^{0.5} \qquad .....(2.14)$$

This ideal relationship can form the basis for a more general one which may be expressed in terms of two arbitrary stations as follows:

$$\frac{W\sqrt{T}}{AC_dP_1} = \left[\frac{2\gamma}{R(\gamma - 1)} \left(1 - \left(\frac{P_1}{P_2} \right)^{\frac{1-\gamma}{\gamma}} \right) \left(\frac{P_1}{P_2} \right)^{-\frac{2}{\gamma}} \right]^{0.5} \quad .....(2.15)$$

where, P_1/P_2 is the pressure ratio which controls the system mass flow, W.

2.5.2 Calculation of Blade Temperature Considering Radial Coolant Flow

The basic equations that represent blade heat transfer and coolant flow are derived from a 'steady-state' heat balance and from momentum and continuity considerations. Consider the heat flow to and from an elemental length δl of a blade a distance l from the root of the blade. As the coolant passes up the blade it increases in temperature which reduces the cooling effectiveness, so that the blade temperature increases from the root to the tip. There is some conduction of heat along the blade to and from the small element δl due to this temperature gradient along the blade. Turbine blades are generally made of low thermal conductivity alloys thus the conduction term would be small and is therefore neglected here [Cohen et. al. (1987), and Hill and Peterson (1992)]. The heat balance equation for the radial passage that also includes the effect of materials is given by:

$$h_g S_{gr} l_r (T_g - T_{wg}) = \frac{k_w}{dth}\left(\frac{S_{gr} + S_{cr}}{2}\right) l_r (T_{wg} - T_{wc}) = h_{cr} S_{cr} l_r (T_{wc} - Tc) = W_{cr} C_p (Tc_3 - Tc_1) \quad .....(2.16)$$

An initial value of h_{cr} can be calculated from:

$$h_{cr} = h_g \left(\frac{S_{gr}}{S_{cr}}\right) \frac{(T_g - T_{wg})}{(T_{wc} - Tc)} \quad .....(2.17)$$

And an initial value of W_{cr} can be calculated as:

$$W_{cr} = 0.003 \times \frac{W_{hpc}}{N_b} \quad .....(2.18)$$

This enables the flow cross sectional area (radial) to be calculated as follows:

$$A_{cr} = \left(FF \times \left(\frac{k}{\mu^{0.8}} \right) \left(\frac{W_{cr}^{0.8}}{h_{cr}} \right) \right)^{\frac{1}{0.9}} \qquad .....(2.19)$$

where,

$$FF = F \times Fhc$$

$$k = \frac{2.978E-03 \times Tc^{0.5}}{1 + \left(\frac{240.0}{Tc} \right)} \qquad .....(2.20)$$

$$\mu = \frac{1.488E-06 \times Tc^{1.5}}{Tc + 110.4} \qquad .....(2.21)$$

(for initial value assume Tc = Tc_1)

Using the equation (2.15) and the driving pressure ratio in the radial passage, coolant mass flow, W_{cr}, can be recalculated:

$$W_{cr} = \frac{A_{cr} C_d Pc_1}{\sqrt{Tc_1}} \left(\frac{2\gamma}{R(\gamma - 1)} \left(\left(\frac{Pc_1}{Pc_2} \right)^{-\frac{2}{\gamma}} - \left(\frac{Pc_1}{Pc_2} \right)^{-\frac{1+\gamma}{\gamma}} \right) \right)^{0.5} \qquad .....(2.22)$$

And hence h_{cr} is recalculated as:

$$h_{cr} = FF \times \left(\frac{k}{\mu^{0.8}} \right) \times \left(\frac{W_{cr}^{0.8}}{A_{cr}^{0.9}} \right) \qquad .....(2.23)$$

Equation (2.23) lead to the calculation of metal temperature (gas side), T_{wg}. Rearranging the equation (2.16):

$$T_{wg} = \frac{\left(1 + H2 - \frac{H1 \times H2}{H1 + H3} \right) T_g + \left(H1 - \frac{H1^2}{H1 + H3} \right) Tc_1}{1 + H2 - \frac{H1 \times H2}{H1 + H3} + \frac{H1 \times H3}{H1 + H3}} \qquad .....(2.24)$$

where,

$$H1 = \frac{h_{cr}}{h_g \times \left(S_{gr} / S_{cr} \right)} \qquad .....(2.25)$$

Step Number	Task	Equation	Comment
Step 1	estimate W_{cr}	2.18	based on the limiting value of flow off-take from the engine compressor.
Step 2	estimate T_{wg}		based on material property limitation, suggested 1500.0 K.
Step 3	calculate h_{cr}	2.17	
Step 4	calculate A_{cr}	2.19	check the value, if within the limiting value of A_{cr}, go to Step 5. If not within the limiting value of A_{cr}, then $W_{cr} = W_{cr} * 0.99$ and go back to Step 4.
Step 5	calculate W_{cr}	2.22	
Step 6	calculate h_{cr}	2.23	compare h_{cr} value from Step 6 with Step 3, if within tolerance then proceed to check whether h_{cr} lies within the acceptable range, if yes then proceed to Step 7 otherwise reset the T_{wg} and h_{cr} values and go to Step 4. If the wall temperature calculation reaches a steady state then only accept, if not equal then go back to Step 4.
Step 7	calculate T_{wg}	2.24	check the value, if within the acceptable limit then accept. If not within the limit and if W_{cr} has not been changed previously, change W_{cr}: $W_{cr} = W_{cr} * 1.01$.
Step 8	calculate Tc	2.30	
Step 9	recalculate k	2.20	
Step 10	recalculate μ	2.21	reset T_{wg} and h_{cr} values and go to Step 4. If the wall temperature calculation reaches a steady state then only accept.

Table 2.1: The cooling system design procedure used in TBCOM.

$$H2 = \frac{h_{cr} S_{cr} l_r}{2 W_{cr} C_p} \qquad .....(2.26)$$

$$S_{cr} = 3.545 \times \sqrt{A_{cr}} \qquad .....(2.27)$$

$$l_r = 0.0406 (\text{the value in meter}) \qquad .....(2.28)$$

$$H3 = \frac{k_w}{dth \times h_g} \times \frac{1}{2} \left[1 + \frac{1}{\frac{S_{gr}}{S_{cr}}} \right] \qquad .....(2.29)$$

From equations (2.16), (2.25) and (2.26) Tc can be recalculated as:

$$Tc = \frac{H2}{H1} \left(T_g - T_{wg} \right) + Tc_1 \qquad .....(2.30)$$

where the temperature balance along the radial passage length is approximated as:

$$Tc_3 - Tc_1 = 2(Tc - Tc_1)$$

i.e. the approximation assumes the temperature rise in the second half of passage length is equal to that in the first half.

The values of W_{cr} and T_{wg} are calculated following an iterative design process. The cooling system design procedure used in the TBCOM is described in Table 2.1.

2.5.3 The Introduction of a Film Cooling Mechanism to the Model

A film cooling mechanism is used in order to achieve a more effective cooling in the turbine blade. A portion of the coolant passing through the radial passage is bled through film holes and provides a film of the coolant over the blade. This film is cooler and thus enhances the cooling effect.

The coolant temperature (Tc), as calculated from the previous section, provides the film hole entry temperature of the coolant, thus:

$$Tc_2 = Tc \qquad .....(2.31)$$

The total film cross sectional area is calculated from:

$$A_f = N_f \frac{1}{4}\pi(df)^2 \qquad .....(2.32)$$

And the pressure ratio across a film hole:

$$\frac{Pc_2}{Pc_{3'}} = \frac{Pc_2}{Pc_3} = \frac{Pc_1}{Pc_3} \Big/ \frac{Pc_1}{Pc_2} \qquad .....(2.33)$$

Hence, referring to equation (2.22), the coolant flow through the film holes, W_{cf} can be calculated as:

$$W_{cf} = \frac{A_f C_{df} Pc_2}{\sqrt{Tc_2}}\left(\frac{2\gamma}{R(\gamma-1)}\left(\left(\frac{Pc_2}{Pc_3}\right)^{-\frac{2}{\gamma}} - \left(\frac{Pc_2}{Pc_3}\right)^{-\frac{1+\gamma}{\gamma}}\right)\right)^{0.5} \qquad .....(2.34)$$

The cooling side heat transfer coefficient, h_{cf} is determined as:

$$h_{cf} = FF' \times \left(\frac{k}{\mu^{0.8}}\right)\left(\frac{W_{cf}^{\ 0.8}}{A_f^{\ 0.9}}\right) \qquad .....(2.35)$$

where, k and μ are evaluated from equations (2.20) and (2.21) with $Tc = (Tc_2 + Tc_{3'})/2.0$, and for the initial calculation $Tc_{3'} = Tc_2$,

FF′ is a constant and $FF' = F_f \times F$.

Then, intermediate metal temperature (along the film hole), T_{wfm} is calculated:

$$T_{wfm} = \frac{\left(1 + H2 - \frac{H1 \times H2}{H1 + H3}\right)T_g + \left(H1 - \frac{H1^2}{H1 + H3}\right)Tc_2}{1 + H2 - \frac{H1 \times H2}{H1 + H3} + \frac{H1 \times H3}{H1 + H3}} \qquad .....(2.36)$$

where,

$$H1 = \frac{h_{cf}}{h_g \times \left(S_{gr}/S_{cr}\right)} \qquad .....(2.37)$$

$$H2 = \frac{h_{cf} S_{cf} l_f}{2 W_{cf} C_p} \qquad .....(2.38)$$

$$S_{cf} = N_f \times \pi \times df \qquad .....(2.39)$$

$$l_f = 5.0 \times df \qquad .....(2.40)$$

$$dth_f = dth/2.0 \qquad .....(2.41)$$

$$H3 = \frac{k_w}{dth_f \times h_g} \times \frac{1}{2}\left[1 + \frac{1}{\dfrac{S_{gr}}{S_{cr}}}\right] \qquad .....(2.42)$$

This enables the temperature of the coolant at the film cooling exit to be calculated:

$$Tc_{3'} = \frac{2H_2}{H_1}(T_g - T_{wfm}) + Tc_2 \qquad .....(2.43)$$

An iterative calculation follows to determine the final value of $Tc_{3'}$. In order to determine the blade wall temperature on the film-cooled side, T_{wf}, film cooling effectiveness, ε_f is calculated as:

$$\varepsilon_f = 0.66 - 0.0092 \times \left(RWA \times \frac{A_{cf}C_{df}}{W_{cf}} \times X_F\right)^{0.8} \times \left(\frac{T_g}{Tc_{3'}}\right)^{0.6} \qquad .....(2.44)$$

where,

$$RWA = Mach \times Pc_3\left(\frac{\gamma}{R \times t_g}\right)^{0.5} \qquad .....(2.45)$$

$$t_g = T_g \Big/ \left[1.0 + \frac{\gamma - 1}{2.0} \times Mach^2\right] \qquad .....(2.46)$$

This allows the calculation of the film temperature, T_f, as follows:

$$T_f = T_g - \varepsilon_f(T_g - Tc_{3'}) \qquad .....(2.47)$$

And finally, the blade wall temperature downstream of the film, T_{wf} is calculated from:

$$T_{wf} = \frac{\left(1 + H2 - \dfrac{H1 \times H2}{H1 + H3}\right)T_f + \left(H1 - \dfrac{H1^2}{H1 + H3}\right)Tc_1}{1 + H2 - \dfrac{H1 \times H2}{H1 + H3} + \dfrac{H1 \times H3}{H1 + H3}} \qquad .....(2.48)$$

where,

$$H1 = \frac{h_{cr}}{h_g} \qquad .....(2.49)$$

$$H3 = \frac{k_w}{dth \times h_g} \qquad .....(2.50)$$

it is assumed that $S_{gf}/S_{cf} = 1.0$ and H2, S_{cr} and l_r are calculated as per equations (2.26), (2.27) and (2.28) respectively.

2.5.4 Design Model Constraints

There are three constraints involved in the design model primarily to ensure the designs do not cross material and flow limits. In order for a design to be acceptable the following constraints must be satisfied:

1. The blade wall temperature on the gas side, $1200.0K \langle T_{wg} \langle 1300.0K$.
2. The blade wall temperature on the film side, $T_{wf} < 1300.0$ K.
3. The flow ratio, $W_{cr}/W_{cf} \geq 0.8$.

2.6 Nature of the Model in Unconstrained and Constrained Situations

TBCOM is a computer model of a multidimensional real life design problem. Although some of the design parameters are set by intuition and experience there is little prior knowledge concerning the nature of the problem. In order to better understand the problem domain and to aid engineering judgement concerning the results achieved it is desirable to have some idea of the nature of the problem. This also helps to define the search methodology to be used with the model.

In an attempt to obtain some information regarding the shape and the nature of complexity involved in the model, a few designs or points are selected from different regions of the total design space. The design space is defined by all possible combinations of the design variables. Investigation into the model is performed by passing hyperplanes through the

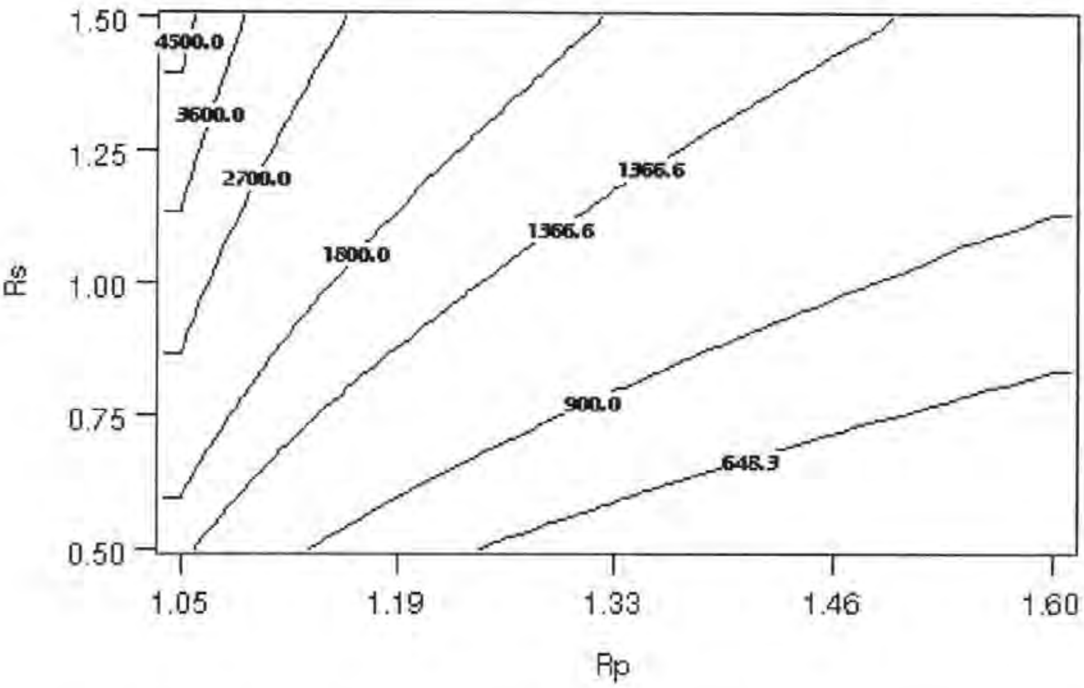


Figure 2.4(a): A contour plot of the unconstrained fitness from the hyperplane-1 through TBCOM; the values of Rp and Rs are varied within their ranges whereas other variables remain constant. The other variables are : (Geom: 3, Cdr: 0.23, Fhc: 3.2, Tc1: 781.0, dth: 0.00082, kw: 28.0, df: 0.0003, Cdf: 0.62, Ff: 1.5, Rpf: 0.25).

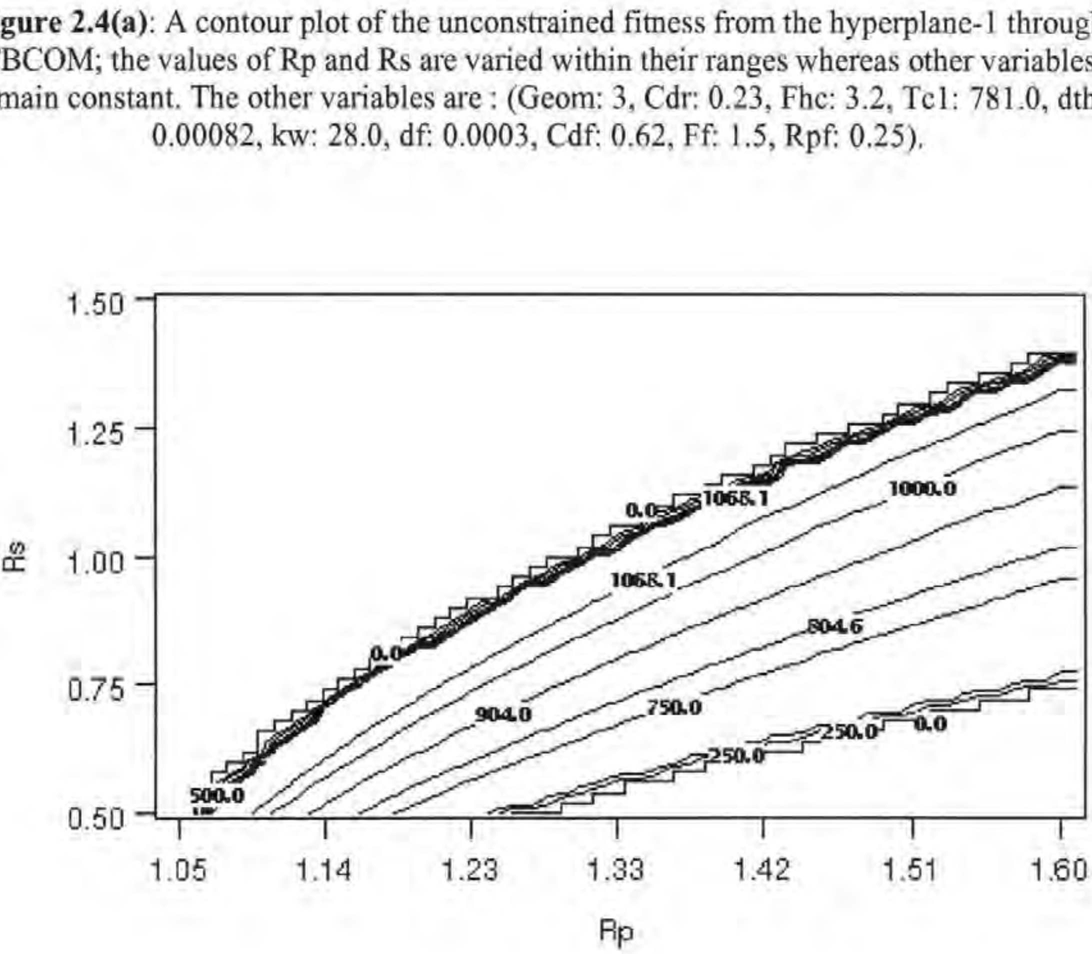


Figure 2.4(b): A contour plot of the constrained fitness (where the fitness is set to 0.0 in case any constraint is violated) from the hyperplane mentioned above; the values of Rp and Rs are varied within their ranges whereas other variables remain constant.

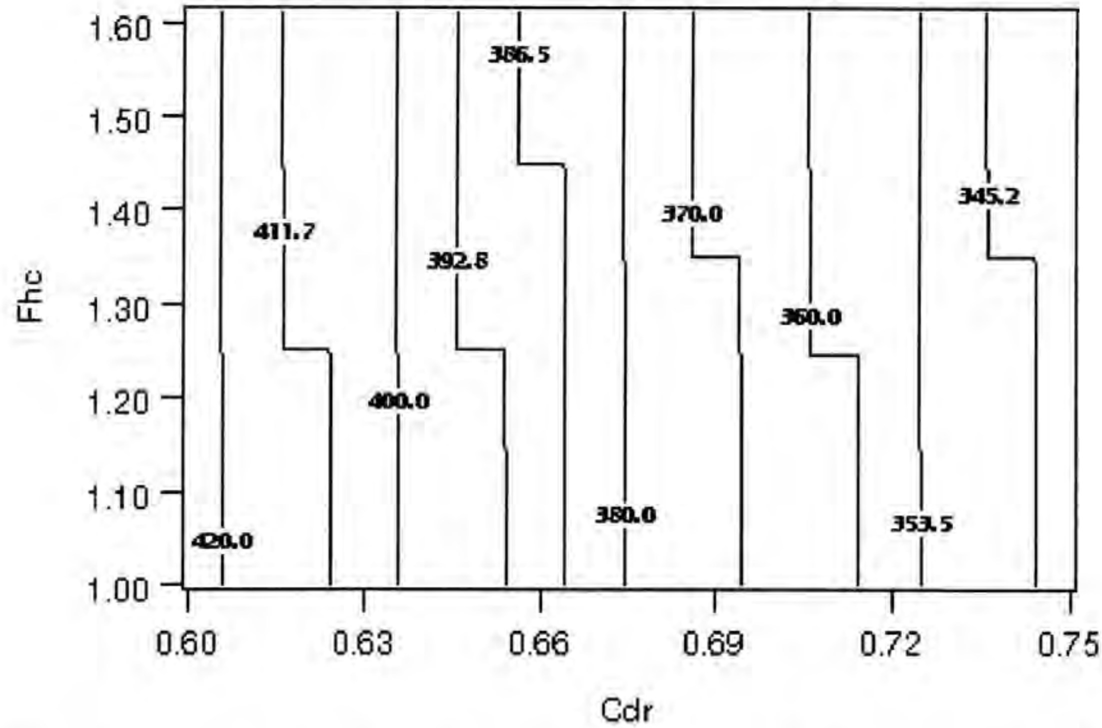


Figure 2.5(a): A contour plot of the unconstrained fitness from the hyperplane-2 through TBCOM; the values of Cdr and Fhc are varied within their ranges whereas other variables remain constant. The other variables are : (Geom: 1, Tc1: 793.0, dth: 0.002340, kw: 24.0, Rp: 1.26, Rs: 0.86, df: 0.00025, Cdf: 0.75, Ff: 1.2, Rpf: 0.28).

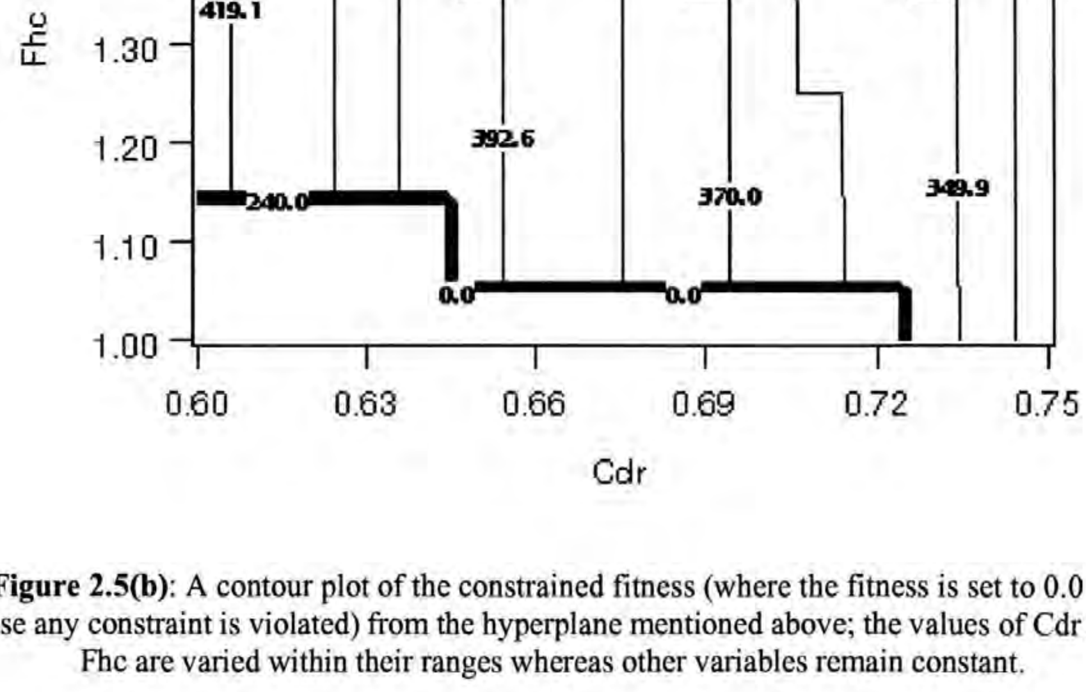


Figure 2.5(b): A contour plot of the constrained fitness (where the fitness is set to 0.0 in case any constraint is violated) from the hyperplane mentioned above; the values of Cdr and Fhc are varied within their ranges whereas other variables remain constant.

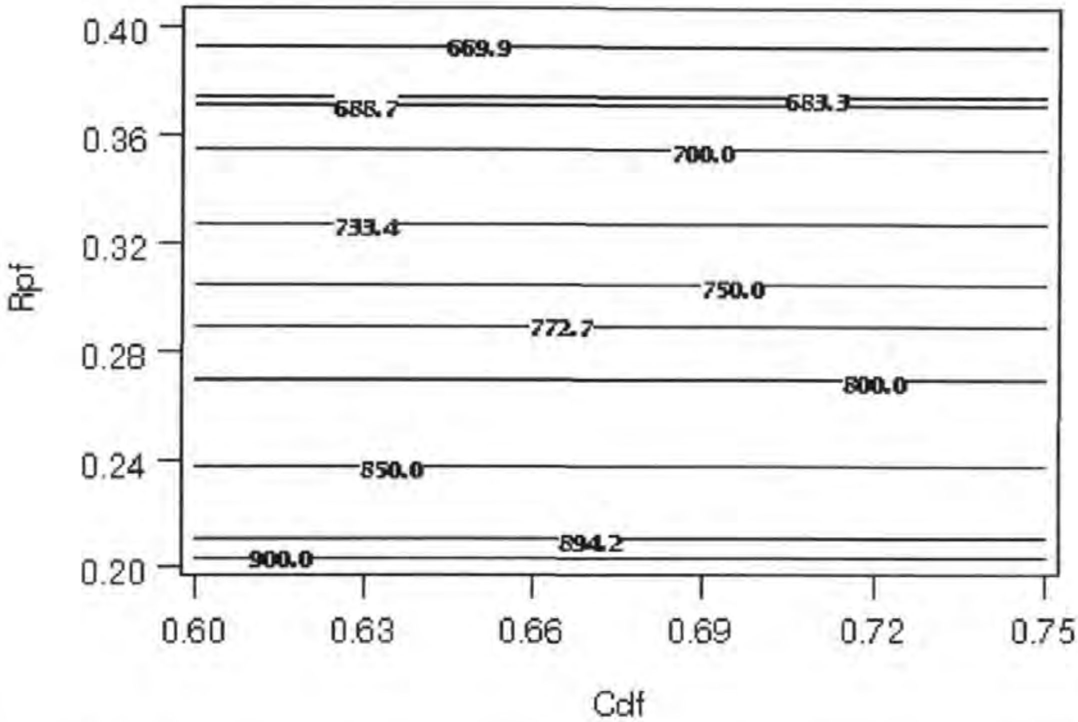


Figure 2.6(a): A contour plot of the unconstrained fitness from the hyperplane-3 through TBCOM; the values of Cdf and Rpf are varied within their ranges whereas other variables remain constant. The other variables are : (Geom: 2, Cdr: 0.44, Fhc: 2.0, Tc1: 744.0, dth: 0.001420, kw: 19.0, Rp: 1.28, Rs: 1.34, df: 0.0002, Ff: 1.0).

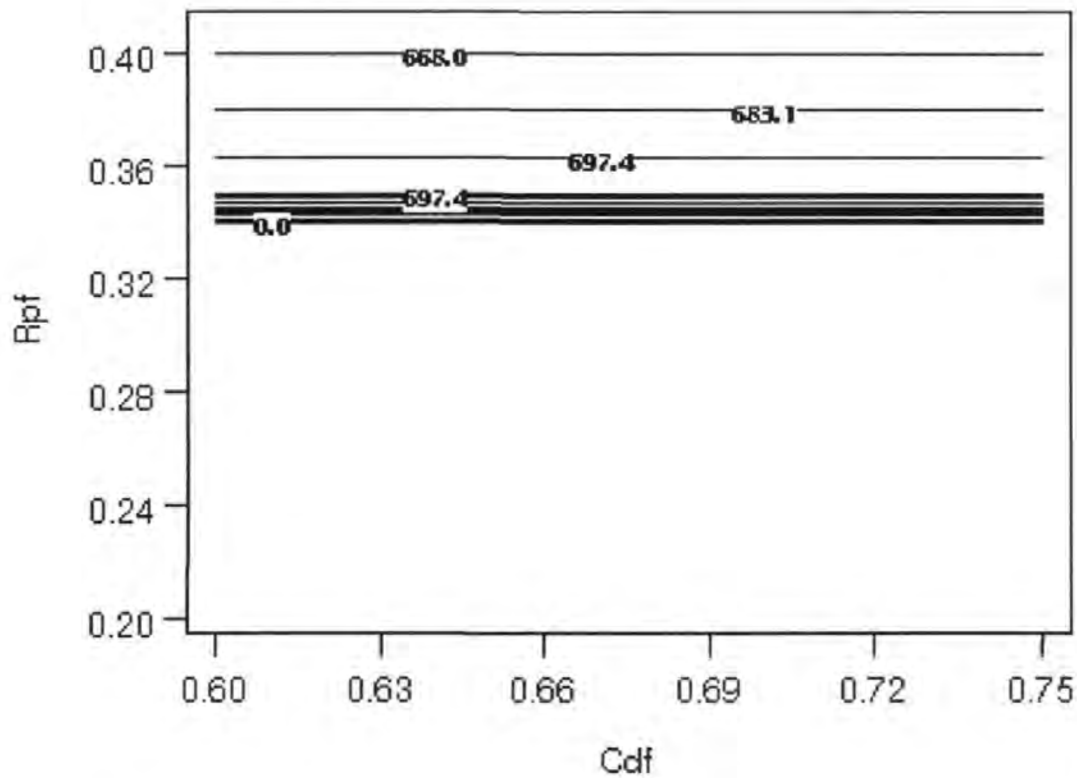


Figure 2.6(b): A contour plot of the constrained fitness (where the fitness is set to 0.0 in case any constraint is violated) from the hyperplane mentioned above; the values of Cdf and Rpf are varied within their ranges whereas other variables remain constant.

points. In each case two design variables are varied within their acceptable ranges while keeping other design variables constant. Figures 2.4 to 2.6 exhibit the contour plots from three different hyperplanes. The figures show both unconstrained and constrained fitness situations, where the fitness is defined as the inverse of the coolant mass flow. The constrained fitness is implemented using a penalty function (defined in the next chapter). The hyperplanes can only provide some insight into the multi-dimensional problem. It is observed that the shape of the constrained fitness plots can be different from the unconstrained one. This is mainly due to the use of the penalty function. The type of geometry (Geom) introduces discreteness in the design space, apart from that the presence of non-linearity is also observed.

2.7 Verification of the model

TBCOM is verified by an expert and a user from Roll Royce. The checking mainly concentrates on the equations derived from the laws of physics. The model also involves certain amount of designers' experience as values of some design parameters. In order to verify whether the design parameter values are representative several design solutions are verified by the expert and the user. They check whether the combination of design variables (the combination represents a design solution) and the fitness (that is the inverse of coolant mass flow) correspond to their understanding about the problem. The design parameter values are changed to fine tune the model till the expert and user are fully satisfied of the results of TBCOM.

The next chapter introduces several existing techniques to obtain multiple solutions from a multimodal fitness landscape. The developed technique, a hybrid of a GA based search and a hill climber, which addresses some of the issues with real life problem optimisation and search is described in chapter 4.

CHAPTER - 3

3. Identification of 'good' solutions using Genetic Algorithms

3.1 Genetic Algorithms

Genetic Algorithms (GAs) [Goldberg (1989)] are adaptive computation methodologies which may be applied to solve search and optimisation problems. They are based upon genetic and evolutionary principles of biological organisms. Biological organisms maintain their presence in the world over many generations by 'evolving' or reproducing new members while some from the existing population die to make room for the younger. This natural selection is performed with a very simple rule of nature, 'survival of the fittest'. Charles Darwin and Alfred Wallace in 1858 independently presented an idea of natural selection. The idea was simple, elegant, and offered a scientific explanation for the complexity, diversity and rules of nature.

Darwin observed that living organisms generally reproduce many offspring but the population tends to remain constant rather than growing exponentially. He noticed the diversity of the organisms present in a population and concluded that despite the presence of natural forces such as resource limitations, disease and predation, some organisms perish. Only the organisms best suited for the environment can survive and proceed to the next generation. These fitter organisms reproduce or 'evolve' new members and thus pass on their 'good' characteristics (i.e. those that helped them to survive) to the next generation. This natural phenomenon helps the organisms to adapt with the change in environment and survive. This also helps to produce, over generations, the best suited offspring for an environment. The evolution by natural selection works by the accumulation of small positive changes in the population.

Later research in genetics has shown that DNA stores all the 'instructions' that define different characteristics of an organism. Thus, there is a mapping between the organism's genetic materials (genotype) and its physical characteristics (phenotype). Physical characteristics of an organism can also be influenced by the environment. Sometimes the relationship between the genotype and phenotype can be very complex. The part of DNA that produces a characteristic is called a 'gene' and the possible alternatives that can occur in the section are known as the gene's 'alleles'. For example, there is a gene for hair colour with black, brown and white alleles. A number of DNA strings are stored in a 'chromosome' within any living cell.

Parts of the parents' DNA combine to form new DNA for their children. Thus characteristics are passed from parents to children. 'Good' features of parents can be brought together in a single individual by this 'crossover' of genetic material through sexual reproduction. The opposite phenomenon is also true: 'bad' features can come together while the 'good' features are not transmitted. However, the 'survival of the fittest' rule of nature favours the survival of children with the 'good' characteristics and enables them to reproduce, thus passing on the combined 'good' characteristics. Children can also have unique characteristics that are totally different from their parents. These unique characteristics can come from a sudden change in the child's DNA. The reasons for this phenomenon can vary from some errors in the natural process, to environmental effects. This process of sudden change is termed 'mutation'.

The very brief introduction to natural evolution offered above highlights features of natural selection and genetics that are a direct motivation for evolutionary computation. In reality, nature is very complicated and many things are still unknown. The intention is to glean some ideas from nature and utilise them to solve search and optimisation problems. One such attempt are genetic algorithms (GAs). Genetic algorithms use a direct analogy of natural phenomenon. GAs work with a 'population' of organisms or 'individuals', each representing a probable solution to a given problem. The problem that needs to be solved serves as the environment. To apply GAs to a problem, two things are essential: a genetic coding for the problem variables and a measure of fitness implemented by a mathematical model of the system called 'fitness function'. The fitness function assigns a numeric value to each solution according to its performance. All possible solutions to the problem describe a ‘search space’ that has to be investigated by GAs. Fitter individuals (represented by parameter sets) are allowed to survive and reproduce into the next generation by 'crossover' and 'mutation' allows the introduction of random change. New individuals (children) of the next generation share some features taken from each 'parent'. The new generation contains a higher proportion of the characteristics possessed by the good members of the previous generation whilst lower performance individuals have a lesser probability of survival. As a result, over many generations, good characteristics are spread throughout the population, being mixed and interchanged with other good characteristics as they go. By favouring the fitter individuals, the most promising or interesting areas of the 'search space' are explored. This 'exploitation' of the good features results in increasingly fit individuals. It is also observed that bad features can combine to produce good features. Good features can also be created by random mutation of the parameter sets which allows the discovery of previously unknown good features. Thus, at least during the initial stages of a search the GA goes through an 'exploration' phase. An efficient GA will converge to an optimal or near optimal solution to the problem. If the problem is multimodal (i.e. more than one optimal solution exists), the GA can be modified to identify the most, if not all, of these solutions. The 'power' of the GA comes from its dynamics which result in robust behaviour. In general GAs can quickly identify solutions close to the global optimum. On the other hand, some specialised techniques suitable for particular problems can out-perform GAs by identifying the global optimum quicker. For example, a classical hill climbing algorithm may identify the optimum quicker than GAs on a unimodal or monotonic search space. Often a hybrid of GAs and a classical search or optimisation algorithm may perform better than either working alone.

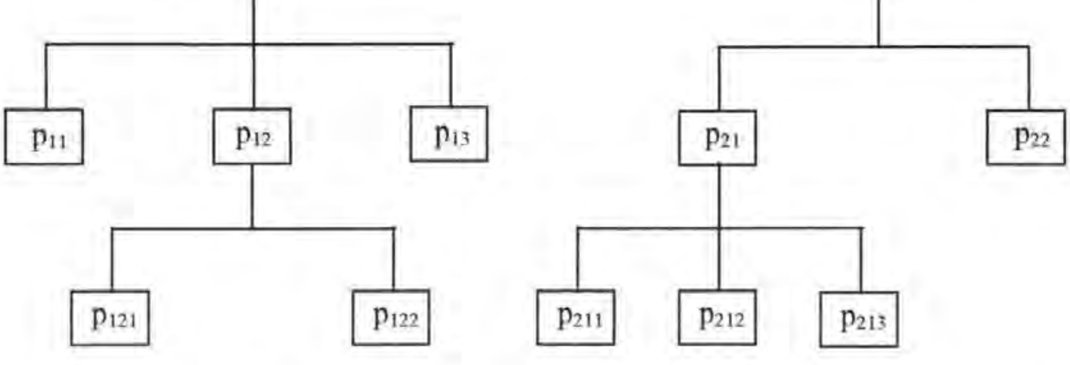
The next section discusses the basic principles of GAs and gives a brief summary of the present theoretical understanding of the process. GAs have been applied to many different areas of engineering, science and economics.

3.1.1 Basic Principles

GAs are used to find the optimum solution (or solutions) to a problem. There are many types of genetic algorithms, each suitable for a separate category of problems. The most commonly used simple GA can be represented as shown in Figure 3.1. The simple GA starts by randomly selecting an initial population of probable solutions. The GA iterates for a fixed number of generations or until it satisfies a stopping criterion. During each generation, the simple GA performs a fitness proportionate selection. The selection mechanism follows the 'survival of the fittest' law to determine which of the chromosomes of the current population are represented in the following population. The next operation is 'crossover', generally the principle genetic operation of the GA. The crossover operator combines the genetic information of a pair of *parent* chromosomes to produce a pair of offspring

```
start GA
   t := 0;                        /* start with an initial time*/
   initpopulation P (t);          /* initialize a usually random population of individuals*/
   evaluate P (t);                /* evaluate fitness of all initial individuals of population*/
   while not done do              /* test for termination criterion (time, fitness, etc.)*/
      t := t + 1;                 /* increase the time counter*/
      P' := selectparents P (t);  /* select a sub-population for offspring production*/
      crossover P' (t);           /* recombine the "genes" of selected parents*/
      mutate P' (t);              /* perturb the mated population stochastically [optional]*/
      invert P' (t);              /* invert the mated population stochastically [optional]*/
      evaluate P' (t);            /* evaluate it's new fitness*/
      P := survive P,P' (t);      /* select the survivors from actual fitness*/
   do
end GA.
```

Figure 3.1: A general description of a simple Genetic Algorithm using pseudo code.



Thus, stC = (p_1, p_2, p_{11}, p_{12}, p_{13}, p_{21}, p_{22}, p_{121}, p_{122}, p_{211}, p_{212}, p_{213})

Figure 3.2: An example of the hierarchical structure in a structured chromosome.

chromosomes (*children*). The proportion of the population selected for 'crossover' is known as the crossover rate or crossover probability. Mutation is the second genetic operator of the GA. The mutation operator acts upon single chromosomes chosen at random from the population. The operator randomly selects a position within the chromosome, and the allele value of the gene at the position is altered. The proportion of the total number of genes in the population selected for mutation is known as the mutation rate or mutation probability. Mutation probability is generally kept much smaller than the crossover probability. Also there are two prerequisites for a GA application: defining a suitable 'coding' (representation) and a fitness function for the problem. The principle issues involved in a GA operation are described as follows.

3.1.1.1 Coding or Problem Representation

GAs are expected to identify the best possible solution or solutions to a problem. It is assumed that a potential solution to the problem can be represented as a set of parameters or problem variables. These parameters represent genes and are combined to form a string of values which represents a *chromosome* and describes a probable solution to the problem. Most GA applications use fixed-length, fixed-order bit strings to encode a probable solution. The use of a binary alphabet for the string is most common for a number of reasons. The first reason is 'historical', GA research started with the binary representation and later others followed the same path. Many people are also comfortable in using the binary representation simply because much of the GA theory and research finding are based on the representation [Mitchell (1996)]. Other possibilities include vectors of real numbers [Davis (1991)], or using an alphabet of many characters. The research reported in this thesis uses a fixed-length binary chromosome; but variable-length chromosomes are appropriate for many problems [Goldberg et. al. (1993)].

For some problems, a simple parametric representation may not be sufficiently flexible to fully describe a possible solution. For example, consider the problem of designing the most cost effective design for a transport system. The solution to this problem may be bus services, rail services or flights. The representation of each of these transport systems requires a different set of parameters, and thus a simple parametric description is not applicable for this design problem. The *structured GA* (stGA) [Dasgupta and McGregor (1991)] utilises redundancy within the chromosome to allow search in such problem domains. The chromosomes of the stGA represent hierarchical structures from which the parameter sets are derived. The hierarchical structure of the chromosome can handle a combination of discrete and continuous variables. High level genes are mostly responsible for discrete design decisions, activating or deactivating lower level genes accordingly. The lower level genes can represent another discrete variable or a continuous variable. The leaf nodes of the hierarchical structure provide a parametric description for each of the design solutions. Thus, generally the higher level genes determine the overall description of the solution whilst the lower level genes determine the parameter set that describes a particular example of the overall structure. For example, in the above transport system design problem, a single high level gene could determine which of the transport systems the chromosome would describe. A set of lower level genes would describe relevant parameter set for the selected transport system. The hierarchical structure shown in Figure 3.2, for example, can be encoded by the chromosomal structure, stC = (p_1, p_2, p_{11}, p_{12}, p_{13}, p_{21}, p_{22}, p_{121}, p_{122}, p_{211}, p_{212}, p_{213}). The two highest level genes (p_1, p_2) determine which of the second level genes are active and contribute to the final parameter set. Similarly, the second level genes determine which parts of the third level genes are active. The turbine blade problem is encoded using a structured chromosome and thus it uses the stGA approach. If the hierarchy is complex and multi-level, there can be very high amount of redundancy in a structured chromosome. The high redundancy can hinder the efficiency of stGA search [Parmee (1996)].

The representation of a problem in the chromosome is referred to as the 'genotype'. A fitness function evaluates the information contained in the chromosome and provides a fitness rating, referred to as the 'phenotype' of the problem. The mapping between the genotype and the phenotype is crucial for the success of a GA.

3.1.1.2 Fitness Function

Application of GAs to a search or optimisation problem requires that a fitness function be used to evaluate the individual solutions. The fitness function can be considered as a model of the problem. The fitness function may involve just one criterion or a combination of many criteria. GAs that handle multicriteria problems are termed as 'multiobjective GAs'. In this case several fitness functions each defining one criteria can also be used with a multiobjective GA [Goldberg (1989)]. Many search or optimisation problem domains involve constraints. If a possible solution to the problem violates any constraint (non-feasible), the fitness of the solution is degraded according to a penalty function. The use of a penalty function helps the GA search to concentrate in the regions of the search space that satisfy the constraints (feasible regions). On the other hand, the application of the penalty function changes the shape of the fitness landscape (Figure 3.3). Thus, selecting an appropriate penalty function is very important for constrained optimisation or search problems.

Some knowledge about the nature of the fitness function can help in designing the GA. Often the information is lacking in real life multidimensional problems. Presence of highly

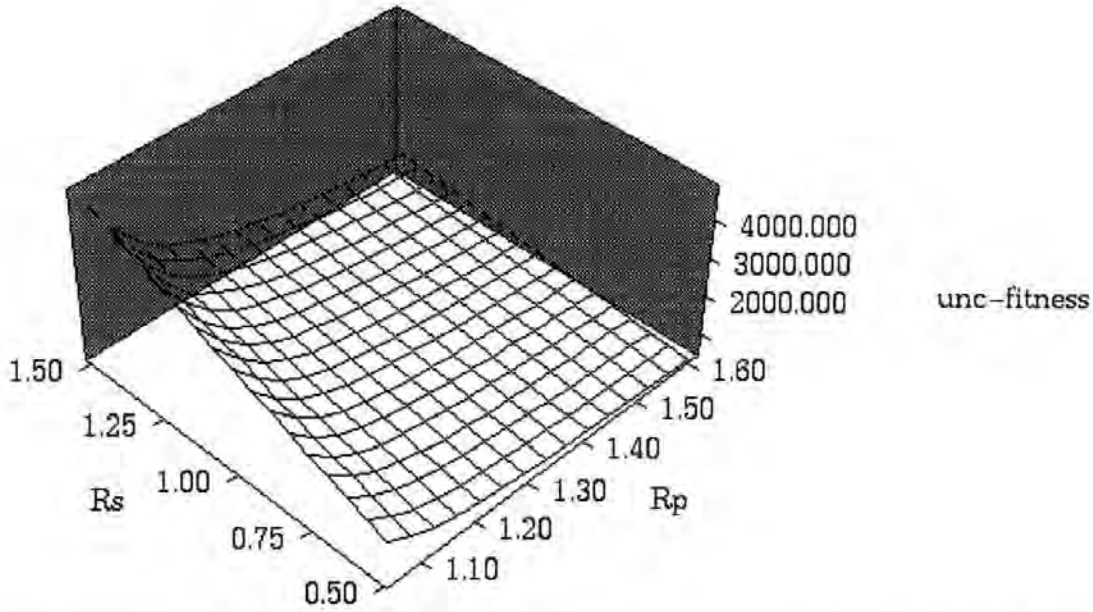


Figure 3.3(a): A plot of the unconstrained fitness from a hyperplane through the Turbine Blade problem (TBCOM), where only two variables Rp and Rs are varied keeping others constant. The other variables are :: (Geom: 3, Cdr: 0.23, Fhc: 3.2, Tc1: 781.0, dth: 0.00082, kw: 28.0, df: 0.0003, Cdf: 0.62, Ff: 1.5, Rpf: 0.25).

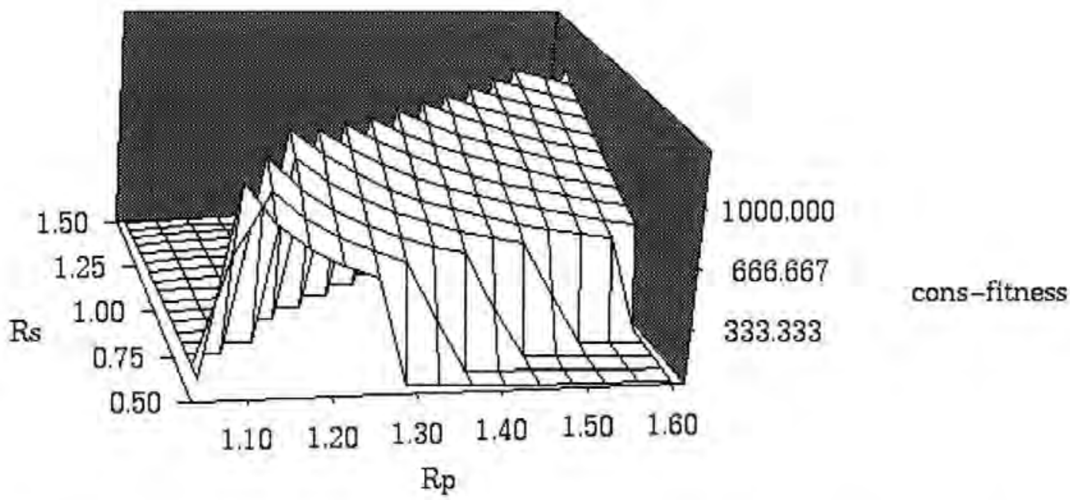


Figure 3.3(b): A plot of the constrained fitness from the above mentioned hyperplane.

non-linear constraints implemented using the penalty functions along with many independent variables, and complex relations between them make the problem very difficult to comprehend. This poses a great challenge to the GA search. In case of real life problems the GA has hardly any prior knowledge concerning the nature of the fitness landscape. In these cases, it is also difficult to validate the best solution(s) achieved by the GA. Research reported herein concentrates on establishing a more confident approach to handle the search or optimisation task for real life problems.

3.1.1.3 The Mechanics of Selective Reproduction

The selection mechanism determines which of the chromosomes of the present population are represented in the following population. Typically, the selection process follows the 'survival of the fittest' rule. Thus, those chromosomes of high fitness prosper at the expense of those chromosomes with low fitness. The simplest and most common type of the fitness proportionate selection is known as *roulette-wheel selection* [Goldberg (1989)]. In case of a fixed size population of n number of solutions (say), the fitness proportionate selection assigns each solution, i, a probability of selection p_{s_i}. The probability is determined according to the fitness of the solution and the total fitness of the population:

$$p_{s_i} = \frac{f_i}{\sum_{j=1}^{n} f_j}$$

The selection scheme chooses a total of n number of solutions or individuals for reproduction, according to the probability distribution (p_{s_i}). The method selects solutions through n number of simulated spins of a roulette wheel. The wheel contains n slots, one each for the solutions. The width of each slot is directly proportional to its respective p_{s_i}. Thus the individuals with higher fitness values are likely to be selected more than those with lower fitnesses. There are many alternatives to this selection strategy. Two popular alternative methodologies are 'tournament selection' and 'stochastic remainder selection'. Tournament selection [Brindle (1981), Goldberg and Deb (1991)] is sensitive to the relative rather than the absolute fitnesses. There are different types of tournament selection. Generally, the tournament selection holds *n* number of tournaments, where *n* is the population size, to select *n* individuals. The tournament selection randomly chooses two (or may be more) individuals for the tournament and the fittest one is selected. This type of selection mechanism is found to be more effective for multimodal fitness function optimisation [Harik (1995), Roy and Parmee (1996)]. Stochastic remainder selection [Brindle (1981), Booker (1982)] is a variant of the roulette wheel selection algorithm which guarantees that a chromosome will receive at least the integer part of its expected number of offspring, and the population is sorted according to the fractional parts of the expected number of offspring. The remainder of the strings needed to fill the population are drawn from the top of the sorted list.

Elitism [De Jong (1975)] is a concept that complements the selection technique used by the GA. Elitism ensures that the best individual present in one generation is passed on to the next generation. The concept is implemented as follows:

> *Let $A^*(t)$ be the best individual generated up to time t. If $a^*(t+1)$ be the best individual present in a population at time $t+1$, and $a^*(t+1)$ is worse than $A^*(t)$, then $A^*(t)$ replaces one of the chromosomes of the new population - either the worst or a randomly selected chromosome.*

Thus the GA never loses the previously found fit individual. This concept is generic and any standard selection method can be changed to be elitist. The new population as produced by a selection mechanism is then used in the reproductive phase.

The reproductive phase starts by randomly pairing the individuals present in the new population. For each couple (*parents*), crossover is determined by a fixed probability p_C , known as the *crossover probability*. Crossover produces two new individuals, known as *children*. The children then proceed to the mutation stage. The parents directly proceed to the mutation stage if they are not crossed (that is with a probability of $1-p_c$). This allows each individual a chance of passing on its genes without the disruption due to crossover. There are many varieties of crossover mechanisms for example, single-point crossover [Goldberg (1989)], two-point crossover [Cavicchio (1970), Goldberg (1989)], uniform crossover [Syswerda (1989)] and order based crossover [Goldberg (1985), Syswerda (1991a) and Davis (1991)]. Single-point crossover is the simplest of all. For a fixed size chromosome of length l, in single-point crossover one of l-1 possible crossing sites is randomly selected. The crossing sites are between a chromosome's neighbouring bits. This produces two 'head' segments and two 'tail' segments. The tail segments are swapped between the parents to produce two new individuals or children (Figure 3.4). The following pseudo code demonstrates an implementation of the single point crossover operation.

```
procedure single_point_crossover
begin
        P1 := the first parent chromosome;
        P2 := the second parent chromosome;
        cross_point = random( 0, chromosome_length - 1);
        for i := 0 to cross_point - 1 do
        begin
                child1[i] = P1[i];
                child2[i] = P2[i];
        end
```

HEAD TAIL

Parent 1: 0 0 1 0 1 1 | **0 1 0** $\Rightarrow$ **Child 1**: 0 0 1 0 1 1 **1 0 1**

Parent 2: 1 1 1 0 0 1 | 1 0 1 $\Rightarrow$ **Child 2**: 1 1 1 0 0 1 **0 1 0**

Figure 3.4: An example of one-point crossover. The children are produced by randomly dividing the parents at the positions denoted by the vertical lines and exchanging the 'tail' parts of the parental genetic material.

Individual 1: 1 0 1 1 **0** 1 0 0 0 1 0 $\Rightarrow$ **New Individual 1**: 1 0 1 1 **1** 1 0 0 0 1 0

↑

Figure 3.5: An example of mutation operation. One individual produces a new individual by flipping the bit at the arrow position (selected randomly).

Individual 1: 1 0 1 | **1 1 0 0** | 1 0 1 1 $\Rightarrow$ **New Individual 1**: 1 0 1 **0 0 1 1** 1 0 1 1

Figure 3.6: An example of inversion operation. One individual produces a new individual by reversing the order of the bits between the two randomly selected positions as denoted by the vertical lines.

```
        for i := cross_point to chromosome_length - 1 do
        begin
                child1[i] = P2[i];
                child2[i] = P1[i];
        end
end
```

In two-point crossover, two crossing sites are randomly selected, and the parents exchange the segment in between the two crossing sites. Uniform crossover is radically different from the previous two types of crossover. Each child is created by randomly copying some bits or genes from one parent and filling the remaining positions from the other parent. Therefore children contain a mixture of genes from each parent. The number of effective crossing points is not fixed, but averages to $l/2$ (where l is the length of a chromosome). In case of order based crossover, it is not the values of the genes that are exchanged, but the order in which they appear. The children have genes that inherit ordering information from each parent. This avoids the generation of children that violate the problem constraints.

The second genetic operator is *mutation*. Unlike crossover, mutation acts upon single chromosomes chosen at random from the population. For every individual undergoing mutation, a random bit position or locus is selected, and the allele value of the gene at that locus is altered (Figure 3.5). The following pseudo code explains an implementation of the mutation operation using binary representation:

```
procedure binary_mutation
begin
```

```
        i := random (0, population_size - 1);
        j := random (0, chromosome_length - 1);
        Ci := the i 'th chromosome of the population;
        Ci[j] = 1 - Ci[j];
end
```

Sometimes another genetic operator known as *inversion* is introduced after mutation [Holland (1975)] (Figure 3.6). Inversion is a reordering operator inspired by a similar operator in biology. Inversion works by reversing the order of genes between two randomly chosen positions within the chromosome. The technique has been applied with some success to 'ordering problems' such as the DNA fragement-assembly problem [Parsons et. al. (1995)]. However, the benefits of inversion to GAs are not very clear yet and therefore needs more systematic experimentations and theoretical studies [Mitchell (1996)].

The most widely used reproduction strategies used in standard GA replace the entire population at once, and are known as 'generational reproduction strategies'. Steady state reproduction [Whitley (1989), Syswerda (1991b)] is a significant departure from the standard GA. In the 'steady state' GA, children enter the parent population immediately after they are produced and are available for reproduction at once. There is therefore the opportunity to exploit a promising chromosome immediately. Syswerda (1991b) compared reproduction in 'generational' and 'steady-state' genetic algorithms. It is observed that, in many cases the 'steady-state' GA converges more rapidly than the 'generational' GA. The standard generation of selection, crossover, mutation and inversion is replaced and a pair of chromosomes are randomly chosen from the population crossed over, mutated and inverted with some probability condition, and put back into the population often replacing the worst chromosomes. The following pseudo code expresses the general structure of a 'steady state' GA.

> **procedure** steady_state_GA
>
> **begin**
>
>   t := 0;
>
>   initialise_population POP(t);
>
>   evaluate POP(t);
>
>   **while** (not stopping_condition) **do**
>
>   **begin**
>
>   i := random (0, population_size - 1);
>
>   j := random (0, population_size - 1);
>
> C_1 := ith member of POP(t);
>
> C_2 := jth member of POP(t);
>
>   **if** (random (0, 1) <= $prob_{crossover}$) **then** crossover C_1 and C_2 ;
>
>   **if** (random (0, 1) <= $prob_{mutation}$) **then** mutate C_1 [optional];
>
>   **if** (random (0, 1) <= $prob_{mutation}$) **then** mutate C_2 [optional];
>
>   **if** (random (0, 1) <= $prob_{inversion}$) **then** invert C_1 [optional];
>
>   **if** (random (0, 1) <= $prob_{inversion}$) **then** invert C_2 [optional];
>
>   copy C_1 to the worst member present in POP(t);
>
>   copy C_2 to the second worst member present in POP(t);
>
>   **end**
>
> **end**

Users of the 'generational' GAs often provide a guarantee that the best member in the current population will be present in the next. This is not necessary with the 'steady-state'

GAs, since often (that is depending on the strategy often used for the replacement) they automatically grant elitist status to *all* good members of the population. Research work presented in this thesis uses the concept of a steady state GA to develop a genetic algorithm suitable for multimodal function optimisation.

3.1.2 Theory

Genetic Algorithm applications have been developed with both binary and non-binary representations. The effectiveness of the representation is very problem specific. That means, some problems are suited for binary representation, whilst others are suited to non-binary representation [Wolpert and Macready (1995)]. The behaviour of the GA has been described in terms of binary representation. The theory, known as the *Building Block Hypothesis* and the *Schemata Theorem*, describes the working of the GA as the processing of several binary templates or schemata. In an attempt to describe the GA with binary or non-binary representation, a *Multary Theory* of GA has been proposed [Field (1996)]. The theory introduces a concept of *key schemata* and extends the present binary operators to multary equivalents. The theory is very recent and needs more investigation.

3.1.2.1 Building Block Hypothesis and Schemata Theorem

The schemata theorem concerns the GA processing of schemata, binary templates that match a set of chromosomes. A schema is a binary string of total length l defined over three alphabets {0,1,#}, where # is a wildcard equivalent to either 0 or 1. For example, the schema #00#100 may represent {0000100, 0001100, 1000100, 1001100}. The schema is also characterised by its *order* and *defining length*. The number of non wildcard characters (that is 0 and 1) present in a schema defines the order of the schema. The order and length of a schema determine the number of chromosomes the schema can match. Defining length is the distance between a schema's outermost, non wildcard character positions. For example, the schema mentioned above is 7 bits long, is of the order 5, written as $o(\#00\#100) = 5$, and has a defining length of 5, written as $\delta(\#00\#100) = 5$. Fitness of a schema is defined as the average fitness of the chromosomes that it represents.

Goldberg (1989) suggested that some schemata are interesting and would help in the GA search. These schemata represent characteristics of a particular problem and are known as *Building Blocks*. The building blocks are low order, short defining length, and highly fit schemata. The survival of the fittest strategy for the selection helps to propagate chromosomes that are members of highly fit schemata. Also the shorter defining length schemata are less disrupted due to crossover and the low order schemata are less likely to be destroyed due to mutation. Thus, the building blocks can survive from generation to generation and are processed by GAs. Holland (1992) estimated that while a GA processes n number of chromosomes in a generation, it actually processes on the order of n^3 building blocks or useful schemata. This phenomenon is described as *implicit parallelism*.

The schemata theorem provides a measure of how many chromosomes of a schema H can survive in the next generation (represented as $m(H, t+1)$) given the distribution of the present generation (given as $m(H, t)$). The following equation determines the value of $m(H, t+1)$:

$$m(H,t+1) \geq m(H,t).\left(\frac{f(H)}{f'}\right).(1 - p_c.\frac{\delta(H)}{l-1} - o(H).p_m) \qquad \ldots (3.1)$$

where $f(H)$ is the fitness of H in generation t, f' is the mean fitness of the chromosomes in generation t, and p_c and p_m are crossover and mutation probabilities. This inequality is known as the Schema Theorem [Holland (1975)]. The theorem describes the expected variation in the number of samples of a given ma from one generation to the next, given its fitness, defining length, and order.

Thus according to the schema theorem, short, low order, or highly fit schemata is expected to survive and prosper within the populations of the GA, whilst long, high order, or poorly fit schemata does not. Goldberg (1983 and 1989) defined these short, low order and highly fit schema as building blocks, and stated his building block hypothesis as "building blocks combine to form better strings". That means during the GA search building blocks recombine to produce fitter building blocks that lead to the fittest solution. The theorem also states that by decreasing either p_c or p_m, an increased use or *exploitation* of the better schemata can be achieved. And by increasing either p_c or p_m, an increased sampling or *exploration* of the search space is achieved. As a rule the GA is expected to maintain a delicate balance between exploitation and exploration. But some time a trade off can be influenced by the nature of a particular problem to which the GA is being applied.

3.2 Identification of multiple sub-optima using multimodal genetic algorithms

3.2.1 Diversity versus useful diversity

Maintaining the population diversity is a major issue in GA search. Early convergence in a GA search can lead to a local sub-optimum, and thus attempts have been made in the past to stop quick convergence of the GA. A diverse search by the GA allows exploration of larger part of the search space in order to converge on a better, single solution. While doing a diverse search, the GA also explores different sub-optima. The three main reasons for a quick convergence of the GA are: *selection pressure, selection noise* and *operator* *disruption*. In case of a finite population GA, use of the 'survival of the fittest' promotes high fitness individuals in the population. This introduces a *selection pressure* towards higher fitness individuals. In case of identically fit individuals the GA randomly selects one, thus there is a variance in the selection process. These variance results in *selection noise*, by which some fitter individuals are randomly thrown out of the population. The use of crossover, mutation and inversion can sometimes destroy the building blocks for higher fitness individuals this is known as *operator disruption*.

One method of increasing the exploration by the GA is to reduce selection pressure and increase operator disruption. Operator disruption can be increased either by appropriate tuning or the introduction of more disruptive operations. This type of exploration is not necessarily useful, for example a very high mutation rate can lead to a random search. A useful diversity should explore the good building blocks [Goldberg and Richardson (1987)]. An exploration can be called *useful* if it exploits the genotypic information present in the population to search through the interesting areas of the search space. The *useful exploration* should be goal directed.

Diversity is utilised in search either to achieve the global optimum or to maintain multiple sub-optima in the final population. In case of multimodal functions these two goals can be dependent on each other. An exploratory GA search that tries to identify the global best in a multimodal function often encounters many local optima. Similarly, a GA search that tries to maintain many sub-optima is likely to do a useful exploration in the search space and thus also likely to find the global optimum in a multimodal function. The GA suitable for multimodal function optimisation is called the *multimodal GA*. Techniques used to achieve the useful exploration for the multimodal GA are generally termed as the *niching methods*.

This thesis concentrates on developing a multimodal GA technique that can maintain diverse individuals in a finite population. The GA is also expected to be suitable for real life problems. When applied to the turbine blade problem (TBCOM), the technique is expected to identify a number 'good' designs. The good designs provide a choice to the designer and thus can help in the design decision making. The next section discusses the chronological development of different techniques used for the maintenance of diversity in a GA search.

3.2.2 Chronological development of multimodal genetic algorithms: a survey of literature

Getting multiple sub-optima or "good" solutions from a genetic search falls in the realm of maintaining diversity in population. The earliest work reported on maintenance of population diversity is Cavicchio's dissertation [Cavicchio (1970)]. As a method of preserving population diversity or variance he introduced a number of *preselection schemes*. The best selection scheme says: if a child is better (in terms of fitness) than the worse parent then replace the parent by the child for the next generation. Cavicchio assumed a parent as the closest member in the population to its child. This assumption may not be valid in case of many multimodal functions. Thus, the *preselection scheme* as described by Cavicchio suffers from high replacement error [Mahfoud (1992)].

De Jong's dissertation [De Jong (1975)] presented his model of multimodal function optimisation based on what is called the *crowding factor* or simply the *crowding model*. The crowding model was inspired from the ecological phenomenon that similar species compete with each other for survival whilst sharing a limited amount of resource. Different species live in different groups or niches, and thus dissimilar species do not compete among each other. The competition for survival to the next generation is local rather than global. The model implements the above phenomenon by using only a fraction of the population (termed as the *generation gap*) to reproduce for the next generation. The same fraction of the population die to accommodate the newly produced individuals due to a finite population size. Preferably the most similar individuals (according to the Hamming distance) are replaced. The replaced individuals are selected from a small sample randomly taken from the population, where the size of the sample is defined by the *crowding factor*. The more similar an individual becomes to other individuals in the population, the more it experiences a heavier selection pressure [De Jong (1975)]. This early work is limited to maintaining diversity of species present in the initial population; however it cannot discover new species or niches. The model also suffers due to stochastic errors introduced in case of low crowding factor.

Application of parallel sub-populations to evolve multiple solutions from a genetic algorithm was attempted by Grosso (1985). In his study he used some degree of communication between sub-populations to allow good building blocks to spread, but that caused reduced diversity and eventual convergence on one global peak. Without such communication the technique becomes equivalent to running a GA several times with a smaller population. Elo (1994) presents a genetic algorithm with a dynamic division mechanism conceived on the Connection Machine-2 for multimodal function optimisation problems. The technique dynamically divides the population into an increasing number of sub-populations to allow specialisation on different maxima as discovered during the search process. This method allows the GA search to adapt to the topology of different multimodal optimisation problems. Without defining the control parameters explicitly, the dynamic nature of the algorithm enables divisions to occur appropriately when the maxima are discovered during the search process. Thus the method is flexible and requires very little knowledge about the fitness landscape. The use of parallel genetic algorithms to obtain multiple sub-optima from a multimodal function is a very promising area of research.

Goldberg and Richardson (1987) introduced what they called as the *sharing method.* In the sharing scheme, fitness is shared as a single resource among similar individuals. Fitness of an individual element of population is derated due to the presence of similar elements in the population. The concept of sharing is implemented by defining a sharing function, *share(d)* as shown below, where d is a measure of dissimilarity between two elements of the population :

$$share(d) = 1 - \left(\frac{d}{s_{share}} \right)^{\alpha}, \text{ when } d \leq s_{share} \qquad \ldots (3.2)$$

$$= 0 \qquad d > s_{share}$$

where, s_{share} is defined as the dissimilarity threshold and α is a constant to determine the shape of the sharing function. An individual is compared with each member of the population to calculate the sharing function values. Summation of all the values due to individual members of the population defines the total sharing function value for the individual. The fitness of an individual is degraded by the total sharing function value, and the new fitness, F ', can be described as follows :

$$F' = F \bigg/ \sum_{i=1}^{N} share(d)_i \text{ , where N = population size} \quad \ldots (3.3)$$

Goldberg et. al.(1992) have discussed the strengths and weaknesses of the above fitness sharing mechanism for optimisation of multimodal functions. Performance of the sharing scheme is very much dependent on the value of s_{share}. Determination of an appropriate value for s_{share} is a difficult task and is dependent on prior knowledge concerning the nature of the problem. Further work has been performed in the same direction by Oei et. al. (1991), where they use tournament selection with a continuously updated sharing technique. The method updates the fitness (or calculates the shared fitness) with respect to the new population distribution as it is being developed. The technique claims to promote and maintain multiple sub-populations over many generations. But the technique is also dependent on prior knowledge regarding the fitness landscape. In an attempt to handle multimodal deceptive functions, Goldberg et. al. (1992) used fitness scaling and the new fitness sharing scheme. Yin and Germay (1993) presented their implementation of a faster genetic algorithm with the sharing scheme using a clustering technique. The clustering method is used to identify different niches present in the population. Niche count (that is the number of elements present in a niche) is used to degrade fitness of individuals present in the niche; thus sharing is local within one niche. Performance of the technique depends on the clustering method used. Setting of parameters for the clustering algorithm needs some trials and prior knowledge. The clustering algorithm also enforces an artificial shape (in this case spherical) to the niches, that may not necessarily be the natural shape for some niches. Jelasity and Dombi (1995) described a niching technique called GAS. The technique dynamically creates a sub-population structure (they call it taxonomic chart) using a *radius function* instead of a single radius value, and a 'cooling' method similar to simulated annealing. The GAS algorithm uses a steady state GA and a high-level algorithm responsible for creating and maintaining the taxonomic chart. The technique allows the population to grow up to a limit and then to die off to reduce the population size to the starting level. The technique introduces a new function called *speed* of a species, that determines the *radius function*. It is not very clear how the technique would perform in case of multidimensional problems. The paper also does not elaborate on the computational complexity of the technique.

In an attempt to model naturally occurring Niche and Species formation, Davidor (1991) developed a GA model called ECO GA, which uses a steady-state GA and is based on local and computationally inexpensive operators. In ECO-GA, the population of strings is held on a 2-D grid having its opposite edges connected together in such a way that each grid element has 8 adjacent elements. Initially individuals are placed at random, one on each grid point. ECO-GA randomly selects one grid element, and defines an 8-element sub-population around it, thus defining a sub-population of 9 elements. This definition implements implicitly parallel and overlapping sub-populations. A steady-state GA is applied with the population size of 9. Two individuals are selected probabilistically from the sub-population according to their relative fitnesses, and genetic operators are applied on them to produce two new individuals. The newly created individuals are probabilistically put back to the same grid positions depending on the relative fitnesses of the opponents (that is the already existing individuals at the two grid points). That means the children are more likely to stay in the vicinity of their parents. The smallness of the size of the sub-population helps the GA to converge very quickly. The technique works based on local convergence which is quick, and assumes that the global optimum can be obtained by the interaction of locally optimised individuals. It is not clear how the search is restricted due to the exploitation of only locally 'good' schema. The implicitly parallel overlapping sub-populations evolve locally but information migrates from one grid to adjacent grid elements because of the overlap. The technique intends to explore the search space in order to identify the global optimum in a multimodal function. The paper has presented some results with a standard one dimensional problem, but it is not clear how the technique would perform in higher dimensions. Further investigation is necessary for a better understanding of the strengths and weaknesses of the technique.

Mahfoud (1992) performed a detailed study on the different niching techniques, especially the crowding methods. Outcome of the study was an improved variant of the crowding technique called the *Deterministic Crowding* (DC) [Mahfoud (1994) and (1995a)]. During his experiments with different crowding methods, Mahfoud found that by choosing members randomly for reproduction, and then providing the selection pressure by only replacing a parent with a fitter child better performance can be achieved. To determine which of the possible parent-child pairing should be used in comparing the parents to their children (that is either (parent1-child1 and parent2-child2) or (parent1-child2 and parent2-child1)), the total of the parent-child similarities (in terms of the Euclidean distance) for each of the two possible combinations are determined. The parents-children pairing that has the highest total similarity is used to determine if the child should replace the parent. The replacement is only possible if the child is fitter than the parent. Deterministic crowding has been applied on two-class and multi-class test problems. In case of multi-class problems it is apparent some peaks dominate over others. Due to crossover interactions among niches some peaks also assist each other to migrate to other peaks. It is observed that the number of population elements present in one class is proportional to the sum of the width of the base of its peak and the widths of the bases of all peaks it dominates. Dominated peaks disappear after some generations unless their assisting peaks are removed beforehand. Although the method performs better than crowding, it is not clear if multiple solutions can be maintained for many generations using this method. The loss of some dominated peaks is a major limitation in case of real life multimodal problems, because there is always a possibility of losing some interesting peaks that are dominated by few others. Another limitation of DC is that it does not guarantee that the final population shall be distributed only among the peaks. This also limits the application of DC in real life problems, because in that case it is not clear whether what is returned from the algorithm is at least a sub-peak or not. Cedeno et. al. (1995) developed the concept of multiniche crowding (MNC) in a genetic algorithm that permits one to simultaneously find several peaks of a multimodal function. In MNC both the selection and replacement steps are modified with a concept of crowding. The idea is to remove the selection pressure due to the fitness proportionate selection (FPR) whilst maintaining the diversity in the population. The method works with local mating and replacement strategy while allowing for some competition for population slots among the niches. In multiniche crowding the FPR is replaced by a *crowding selection*, where each member of the population has equal chance to mate in the next generation. First, an individual is selected either sequentially or at random. The partner for mating is selected from a random sample taken from the population (the size of the sample is defined by the *crowding selection group size* (C_s)). The MNC uses a replacement policy called *worst among the most similar*. In order to select an individual from the population for replacement by a child, crowding factor groups (the number of groups are defined by the *crowding factor* (C_f)) are defined by randomly selecting *s* (called as the *crowding factor group size*) number of individuals from the population per group. Next, one individual from each group is identified that is phenotypically the most similar to the child; and this constitutes a list of individuals ready for the replacement. The child replaces the lowest fit individual in the list. It is worth noting that the child could possibly have a lower fitness than the individual being replaced. The technique is applied on several test functions and also to determine the sequence of all nucleotide in a DNA molecule, from restriction-fragment data. The method works well for the test functions using the given set of crowding parameters. The paper does not comment concerning the quality of the solutions achieved. The parameters are set by trial and error and the paper also does not mention possible effects of the crowding parameters' values on the search. In a recent work, Miller and Shaw (1996) have introduced the *Dynamic Niche Sharing* for multimodal function optimisation. The technique is developed to be faster than the previous sharing method. The dynamic niching uses a greedy approach to identify peaks present in the population in every generation. Individuals are categorised according to the peak it belongs to (that is if within the σ_{sh} radius of the peak). If an individual does not belong to any peak, it is categorised as 'non-peak'. Thus every individual belongs to a niche (or category), and the fitness of the individual is degraded by the size of its niche (*niche count*). Thus every individual within a dynamic niche has their raw fitness degraded equally. This means that there is no incentive to maintain distance between individuals within a dynamic niche. This allows the dynamic niching to explore the regions around the peaks of the niches more thoroughly than standard sharing. The overall performance of the technique is found to be better than the sharing technique and DC on a test function. It is not clear how efficient the technique would be for multidimensional problems. Setting a value for the σ_{sh} would require prior knowledge about the problem, and that also restricts the use of the technique for real life problems.

In real life problems, some time the model evaluation can be very expensive, and thus a smaller population size is used. All the techniques mentioned above try to maintain multiple peaks in one population. That means, in case of fixed sized population the identification of a number of peaks is restricted by the size of the population. An alternative approach called the *Sequential Niche Technique*, was proposed by Beasly et. al. (1993) where peaks are identified one at a time. This generalised technique allows unimodal function optimisation methods to be extended to identify all optima and sub-optima of multimodal problems. The research implements the concept with a standard genetic algorithm. The method involves multiple runs of a GA but uses knowledge obtained from previous runs to avoid re-searching the regions of the problem space where peaks (optima or sub-optima) have already been identified. Whenever one peak is located, in subsequent runs, region around the peak (defined by a *niche radius*) is depressed by applying a fitness derating function. That helps the search in concentrating in other interesting areas and thus identifying multiple peaks. The algorithm is dependent on the right selection of the niche radius. The use of the niche radius imposes a shape to the niches (in this case spherical). In case of problems where the maxima are not evenly distributed, the fixed size of the niche radius would underestimate the size of some niches whereas overestimating the size of others. An inappropriate selection of the niche radius can introduce false peaks, and that can misguide the search. Sequential niching can also offset a peak's location as a consequence of the fitness deration. The artificial shape may not match with the natural shapes of some niches. Prior knowledge concerning the problem would be helpful in determining a workable niche radius. This is a similar limitation as with the fitness sharing technique. In the fitness sharing method fitness landscape is modified every time an individual is evaluated, whereas in the sequential niche technique the fitness landscape remains static during one run. Thus the sequential niche technique overcomes the problem of exponential scaling of its fitness landscape. Another major limitation of the technique is that it does not allow transfer of the building block information to find one solution from another. This can restrict the GA's search capability in some applications. Mahfoud (1995b) compared other niching techniques with the sequential niching. The paper supports the above mentioned weaknesses of the sequential niching. It is also shown that, fitness sharing or DC performs better than the sequential niching over a wide range of functions.

The immune system model for pattern matching was first developed by Stadnyk (1987). The model could achieve niching by lowering the number of antigens used in computing the fitness of each population element. Smith et. al. (1993) implemented an immune system model along with a GA in order to develop a GA which can search for diverse and co-operative populations. It is observed that the model exhibits an implicit fitness sharing which can be useful for multimodal function optimisation. The area of research is relatively new and needs further investigation before it can be useful for multidimensional multimodal real life problems. In a very recent work Darwen and Yao (1996) compared the fitness sharing technique with the above mentioned implicit sharing. The authors used a realistic letter classification problem for the comparison. It is observed that the implicit fitness sharing searches the optima more comprehensively even when those optima belong to smaller hills, and also when the population is not large enough to form the species at each optima. In case of implicit sharing the individual closest to a peak is rewarded even if it's not particularly close to it and when another individual is almost as close. That means in case of implicit sharing there is greater relative selection pressure for the nearer individual and that helps in the better exploration. Whereas in case of fitness sharing the niching radius σ_{sh} means the closest individual to a peak shares its payoff with all other individuals that are almost as close. In the case of small population the tendency of comprehensive peak coverage degrades the performance of the implicit sharing more than the fitness sharing.

Parmee et. al. (1994) and Parmee (1996) describe a method of maintaining diversity and reinforcing the natural clustering (niching) tendencies of the GA by appropriate tuning of crossover and mutation probabilities. A shared near neighbour clustering algorithm is used after some pre set number of generations to further define the naturally occurring clusters present in the population. The clustering method does not impose any artificial shape on the niches present in a population. The method is suitable for rapid identification of 'good' regions in a problem space as opposed to the identification of individual optima. In this respect the technique is being developed to provide information to the engineer concerning high-performance regions of a complex, multidimensional search space [Parmee (1995)].

The technique requires no prior knowledge concerning the modality of the fitness landscape.

An improved tournament selection method for multimodal functions called the *Restricted Tournament Selection* (RTS) is developed by Harik (1994) and (1995). The technique is based on the principle of local competition, that is a tournament among similar individuals (according to a distance metric). The method creates a new population as in a steady state GA [Syswerda (1991b)]. Before an individual is allowed to the next generation it is placed into tournament with the closest (according to the distance metric) individual present within a random sample of the population. The size of the sample is kept fixed and is termed as the *window size*. This form of tournament selection should restrict an entering individual from competing with others, which are too different from it. For an individual, if the closest sub-optimum is selected in the random sample, the individual competes with the sub-optimum and fails to replace it. Thus, if the window size is big enough the replacement error is reduced. Therefore after the peaks are identified, the underlying distribution of the population is expected not to change for a long time. The procedure is dependent on the probability of a peak present in the sample taken from the population. This restricts the number of peaks the algorithm can maintain depending on the size of the window. That means the size of the window is determined using prior knowledge concerning the modality of the fitness landscape. RTS has been successfully applied to some multimodal test functions. The presence of a *dominance factor* in RTS is demonstrated in the next chapter. It is observed that in a prolonged run some peaks start dominating others. Thus RTS can not achieve a steady state of distribution and it carries the risk of losing some peaks. RTS can delay complete dominance of some peaks over others. But because of the presence of the dominance factor, distribution of individuals on several peaks changes. A steady distribution can be achieved by using a very large window size. The dominance factor becomes prominent when some dominating individuals start occupying a major part of the population. In case of real life problems, without any prior knowledge concerning the location and the number of peaks present, it becomes almost impossible to determine when to stop the GA so that the population is distributed among the peaks. Stopping early may mean converging to individuals which are not peaks. But delayed stopping can also lose some peaks because of the dominance factor.

3.2.3 Limitations of the previous research for real life problems

Real life problems can pose some additional challenge than test functions. Test functions can be made very complex, but as a test function is developed with a goal in mind (say one wants to develop a multimodal two dimensional test function), it is easier to get some idea about the nature of the problem. Real life problems are difficult mainly because of the lack of prior knowledge. The techniques mentioned in the previous section are mostly tested on test functions. The main reason is that it is easier to visualise and measure the performance of an algorithm on test functions. Most of the techniques determine the search parameters assuming prior knowledge concerning the search space. Performance of the techniques is measured in terms of population distributions on known peaks. Only a few techniques are applied to real life problems, where the validation of the techniques is extremely difficult. A real life problem may be considered to have the following characteristics:

a) There is not much prior knowledge regarding the shape of the search space.

b) No prior knowledge regarding the performance and location of the optimum and sub-optimum points in the search space.

The lack of prior knowledge invites some difficulties for a multimodal GA search, such as:

a) The determination of search parameter values becomes extremely difficult in the absence of prior information regarding the modality of the search space.

b) It is very difficult to identify the state at which the GA distributes the population on the peaks.

c) The validation of the results obtained from the GA search becomes quite difficult because of the lack of knowledge concerning the quality and location of the peaks.

The next chapter describes the *Adaptive Restricted Tournament Selection*, a multimodal GA technique suitable for real life problems. The technique is compared with RTS and DC using some test functions. A hybrid of the multimodal GA technique and a local hill climber is used to identify multiple 'good' designs for the turbine blade design problem (TBCOM).

CHAPTER - 4

4. Adaptive Restricted Tournament Selection

4.1 Introduction

Genetic Algorithms (GA) and other adaptive search techniques such as simulated annealing and tabu search [Reeves (1993)] have been successfully applied to many optimisation problems where the aim is to identify the global optimum solution. Many real life problems require the identification of several "good" solutions (that is multiple sub-optima) in addition to the global optimum. Multimodal GAs identify several sub-optima present in a problem space. Research presented in this chapter attempts to add another methodology to the list of the multimodal GA techniques.

Engineering design often involves several objectives. A true engineering solution is not necessarily the global optimum with respect to one criterion [Parmee (1994), Parmee and Denham (1994)]. Often the final design needs to be selected by the designer considering many different criteria. In the case of multimodal design problems there may be quite different design solutions that perform similarly with respect to one criterion but these designs can have large differences in the degree of satisfaction of other criteria. Both the quantitative and qualitative aspects of criteria related to say, manufacturability, cost, maintainability, robustness and customer preferences should be taken into consideration. Integrating all of these criteria into one comprehensive evaluation function is difficult and may prove misleading. If the criteria are quantitative in nature one way of handling the situation is to use multiobjective genetic algorithms with *pareto* optimality [Goldberg (1989)]. An attempt is made here to identify multiple "good" design solutions in terms of the most important and quantitative criterion and then to evaluate them qualitatively in terms of other criteria. The research has developed a multimodal GA technique called 'Adaptive Restricted Tournament Selection'.

Adaptive Restricted Tournament Selection (ARTS) [Roy and Parmee (1995) and (1996)] identifies multiple sub-optima in a multimodal fitness landscape, where each sub-optimum represents a design option. The technique is an improvement over Restricted Tournament Selection (RTS) [Harik (1994) and (1995)]. In RTS a window (that is a fixed size sample) is defined to identify the closest point from a newly generated individual. A tournament is performed between the newly generated individual and the closest point before one of them can enter the next generation. The size of the window limits how many peaks or sub-optima may be represented in the final population. Without knowing how many peaks are present in the fitness landscape it is difficult to decide the size of the window. Thus RTS requires *prior* knowledge about the problem. In real life problems information about the modality of the fitness landscape is not available. In order to handle real life problems, ARTS uses a shared near neighbour clustering method [Jarvis and Patrick (1973)] to define the closest point for a newly generated individual. For every generation this method identifies clusters of points present in the population. For each newly generated individual the closest point in the generation is determined by finding the closest point of the closest cluster present in the population. Thus the necessity for a fixed size window and prior knowledge about the problem (as in case of RTS) are eliminated in ARTS.

ARTS is compared with two recent multimodal GA techniques, RTS and Deterministic Crowding (DC) [Mahfoud (1992) and (1994)]. This chapter presents and discusses the results. A study on the effects of the clustering parameters on the performance of ARTS is also presented. A hybrid of an ARTS based GA and a local knowledge based hill climbing technique is used in the Adaptive Search Manager (ASM) [Roy et. al. (1996a)]. Finally a stochastic local hill climber algorithm is used to fine tune the designs selected by the ASM. The chapter also describes both the hill climbing techniques.

4.2 The Shared Near Neighbour Clustering Method

The shared near neighbour clustering method [Jarvis and Patrick (1973)] is a nonparametric clustering technique incorporating the concept of similarity based on the sharing of near neighbours. The technique is simple to implement and computationally inexpensive (except in case of very high dimensional problem). The clustering methodology is applicable to a wide class of practical problems involving large sample size and high dimensionality [Jarvis and Patrick (1973)]. The method is particularly suitable as an analysis tool when little *prior* knowledge about the problem space is available.

4.2.1 The Similarity by Sharing of Near Neighbours

Let $\{x_1, x_2, ..., x_n\}$ be a set of parametric data vectors in an L dimensional Euclidean vector space. The task is to divide these n data points into M number of clusters (where M is unknown), where each group can be considered as a cluster of points. Two data points are considered *similar* if their respective K number of nearest neighbours match. The value of K defines the size of a nearest neighbour list for each point. The similarity measurement is valid only if the tested points themselves also belong to the common neighbourhood. This avoids the possibility of clustering a small and relatively isolated number of points with a high density group. The similarity measure has its own built-in automatic scaling. This means that where points are widely spread, the neighbourhood (that is the volume containing K nearest neighbours) expands. If the points are tightly positioned the neighbourhood shrinks. Thus the clustering technique does not depend on a globally fixed distance threshold. There is possible interactive control of the clustering by specifying K and the number of shared neighbours that is regarded as sufficient (KT) for the clustering. KT is known as the *similarity threshold*.

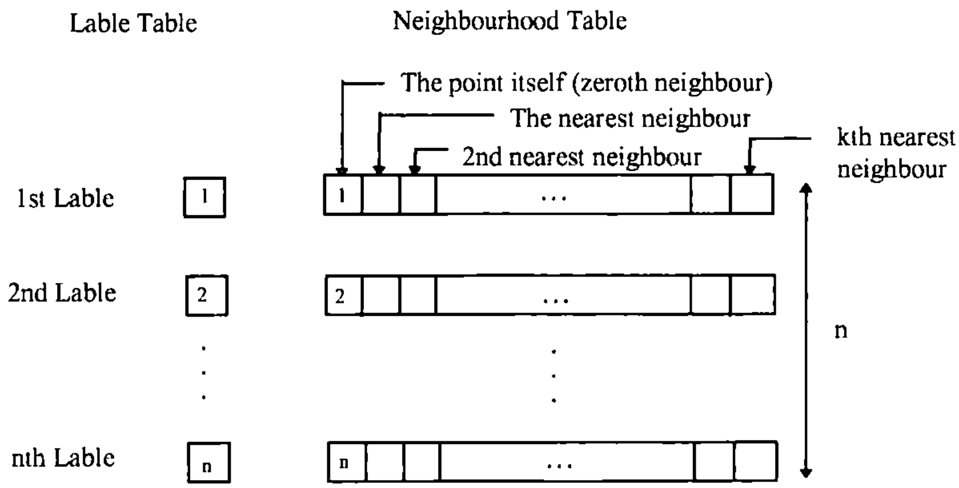


Figure 4.1: The near neighbour and the lable table. All the entries are integer numbers.

4.2.2 The Clustering Algorithm

The clustering algorithm using the above mentioned concept of similarity can be described as follows:

Step 1: For each point of the data set $\{x_1, x_2, ..., x_n\}$, K nearest neighbours (in this research they are defined using the Euclidean distance) are listed in an increasing order of the distance. The data point is regarded as its own zeroth neighbour.

Step 2: An integer label table of length n, with each entry initially set to the first entry of the corresponding neighbourhood list is developed (Figure 4.1).

Step 3: All possible pairs of the neighbourhood lists are tested as follows: replace both label entries by the smaller of the two existing entries if both zeroth neighbours (that are the points being tested) are found in both the neighbourhood lists, and at least the KT neighbour matches exist between the two lists. Also, all appearances of the higher label (throughout the entire label table) are replaced with the lower label if the above test is positive.

Step 4: The clusters with the K and KT values are now indicated by identical labelling of the points belonging to the clusters.

Step 5: Recalculation of the clusters with new values of K and KT can be carried out simply by returning to step 2 until a desired criterion is satisfied. The first selection of K should be the largest the clustering would ever require so that the original vector data need not be recalled.

Thus by setting K and KT equal one can achieve the tightest clustering possible. Although Euclidean distance is mentioned in Step 1, the method is by no means restricted to this measure and any suitable measure can be used. In general the clustering does not impose a shape to the clusters, but with a relatively large value of K the clustering will tend to produce globular bias. The computational complexity of calculating the near neighbourhood table is of the order of $(n)^2L + C(K)$ operations, where C is a relatively small factor to allow for the extra overhead of testing for all K near neighbours for each point. With little improvement in the algorithm, only $n(n-1)/2$ distance measures are necessary for the clustering. The clustering algorithm is integrated with the ARTS based GA technique.

4.3 Adaptive Restricted Tournament Selection

4.3.1 The Algorithm

Adaptive Restricted Tournament Selection (ARTS) is an improved multimodal GA algorithm. ARTS identifies a number of sub-optimum points in a search space without any prior knowledge concerning the modality of the fitness landscape. Thus ARTS is suitable for real life problems. The sub-optimum solutions can be considered as "good" solutions. A formal definition of a "good" solution for this thesis is given below:

Let us assume a search space S and an objective function f (that assigns a real number to any member of S).

$$f: S \rightarrow R^n$$

Without loss of generality, let us assume that the goal is to maximise with respect to f. A neighbourhood of an element i of the search space S is defined by the resolution on each dimension. For any $i \in S$, $N(i) \subseteq S$ is the neighbourhood of i in S. Where i can be considered a "good" solution or a sub-optimum member of the search space S if:

$$f(i) \geq f(j) \text{ for all } j \in N(i)$$

The algorithm is used with a steady state GA [Syswerda (1991b)]. In every generation, there are n (where, n = *population size*) number of iterations and in every iteration two individuals are selected at random (they are termed as *parents*). Two new individuals, *children*, are created by crossover between the parents. The population is clustered every generation using the shared near neighbour clustering technique [Jarvis and Patrick (1973)]. The clustering is performed with respect to the Euclidean space (that is the parameter space), clustering time is therefore independent of the model evaluation time. The clustering is controlled by the two parameters, K and KT. The tightest possible clustering is achieved if the values of K and KT are set equal for the clustering. The clusters are considered as niches present in the population. For a newly generated individual (a child) the closest element in the population is found by finding the closest element of the closest cluster present. The closest cluster is identified according to the Euclidean distances between a child and the cluster centroids. With a relatively large value of K (in this case $K > L$) the shape of the clusters can be given some globular bias, that is necessary to make the cluster centroid calculations more meaningful. Each child competes with the closest individual found in the population. The number of individuals present in the closest cluster is equivalent to the window size in RTS, but here the number of elements is determined adaptively according to the distribution of elements in the population. Thus ARTS does not need any prior knowledge concerning how many peaks are present in the problem space. The algorithm can be described as follows:

Step 1 : Initialise population. gen = 0

Step 2 : Cluster population. Find the centroids of the clusters, num = 0

Step 3 : Randomly select two individuals (say, P1 and P2)

Apply the GA operators (Crossover and may be Mutation) on P1 and P2 to generate C1 and C2

For C1:

Select the closest cluster (according to the Euclid. dist. between C1 and the cluster centroids)

Find the closest individual (say, $C1'$) from the closest cluster

If fitness(C1) $\geq$ fitness($C1'$) then replace $C1'$ by C1 in the population

For C2 :

Select the closest cluster (according to the Euclid. dist. between C2 and the cluster centroids)

Find the closest individual (say, $C2'$) from the closest cluster

If fitness(C2) $\geq$ fitness($C2'$) then replace $C2'$ by C2 in the population

num = num + 1

If num < POPSIZE go to Step 3

Step 4 : gen = gen + 1

If gen < MAXGEN go to step 2.

The technique is applied here on a number of one dimensional multimodal test functions. ARTS identifies and maintains all the peaks present in those functions. ARTS is also successfully applied to the turbine blade problem within the Adaptive Search Manager.

4.3.2 ARTS and the GA Search

The principle behind ARTS is local competition while using the pool of building blocks present in the population. It is observed during the empirical trials with different multimodal test functions that ARTS exploits schema information at its initial stages of a run (i.e. the first few hundred generations). Once the population elements are distributed among the peaks a steady state is achieved where the competition is entirely local. During the initial stages of a run when the population is quite diverse the clustering algorithm tends to form wider clusters thus introducing some replacement errors in the ARTS search (*clustering error*). This causes a delayed convergence on the peaks. At the steady state of distribution, when the population is distributed among the peaks the clustering algorithm identifies the niches correctly. This helps to restrict the tournament within each niche and thus eliminates the dominance problem (that is discussed in the previous chapter) as seen in the case of RTS. A simple genetic algorithm (SGA) [Goldberg (1989)] converges to a global optimum, whereas ARTS can maintain multiple peaks. ARTS also continues to search (even in later generations) a larger space by crossover between different niches present at the steady state of population distribution.

4.4 A Comparative Study of ARTS, RTS and DC

ARTS, RTS and DC have been tested on four test functions, among which two are sine functions (termed as F1 and F2) as used by Harik (1994) and (1995), and the other two are class functions (termed as CF1 and CF2) as used by Mahfoud (1994). Same test parameters are used for all the three experiments. The test parameters are as follows:

Population size = 100

Maximum generation = 500

Crossover type = One point crossover

Crossover probability = 1.0

Mutation probability = 0.0

Window size for RTS = 20

K for shared near neighbour clust. in ARTS = 15

KT for shared near neighbour clust. in ARTS = 15

Tests were performed on a Sun Sparc 10 computer with the same seed value for the random number generator. ARTS, RTS and DC have been tested for the distribution of population elements on the peaks. An individual (i.e. a population element) having a fitness of at least 99% of a peak value is considered to be on the peak.

4.4.1 The Two Dimensional Test Functions

The four two dimensional test functions used for the tests are described below:

Function F1

This is a sine function that has five equally spaced peaks of equal height within a range [0,1]. The function is defined as $f(x) = \sin^6(5\pi x)$ (Figure 4.2). The five peaks have equal height of 1.0 at x = 0.1, 0.3, 0.5, 0.7, and 0.9.

Function F2

This sine function is defined on [0,1], having five unevenly spaced unequal peaks. The function is defined as $f(x) = e^{-2\ln 2((x-0.1)/0.8)^{**2}}\sin^6(5\pi(x^{3/4} - 0.05))$ (Figure 4.3). This function is

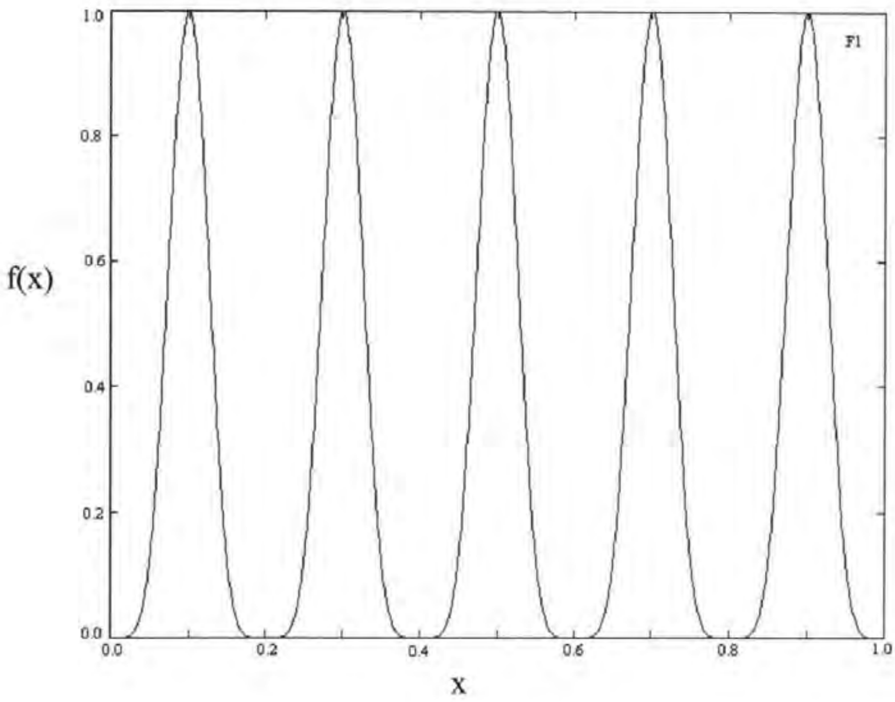


Figure 4.2: Function F1

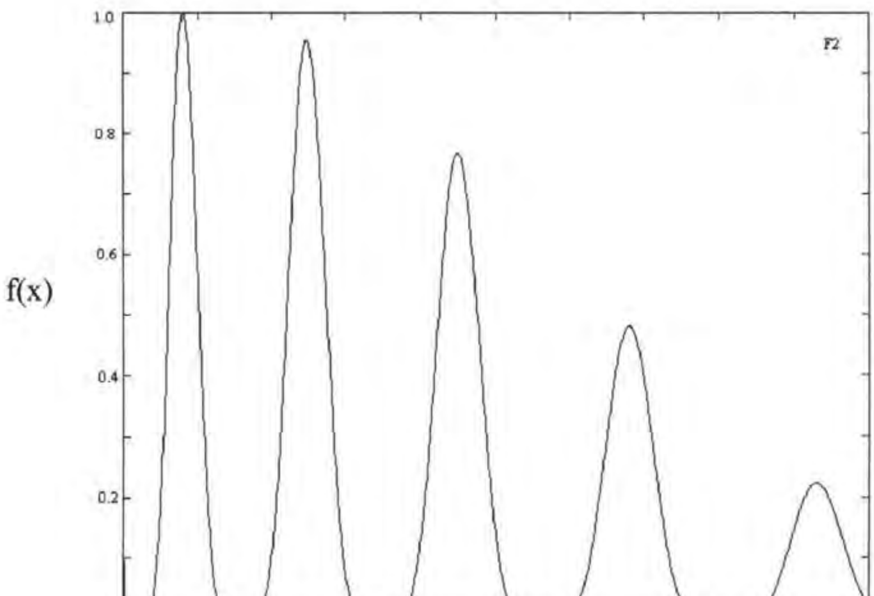


Figure 4.3: Function F2

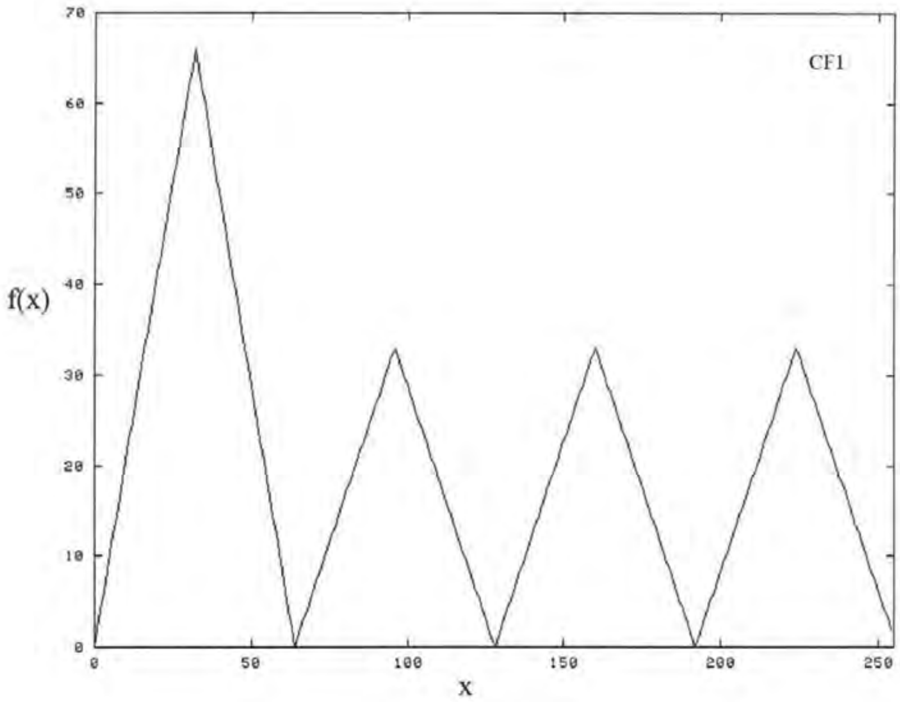


Figure 4.4: Function CF1

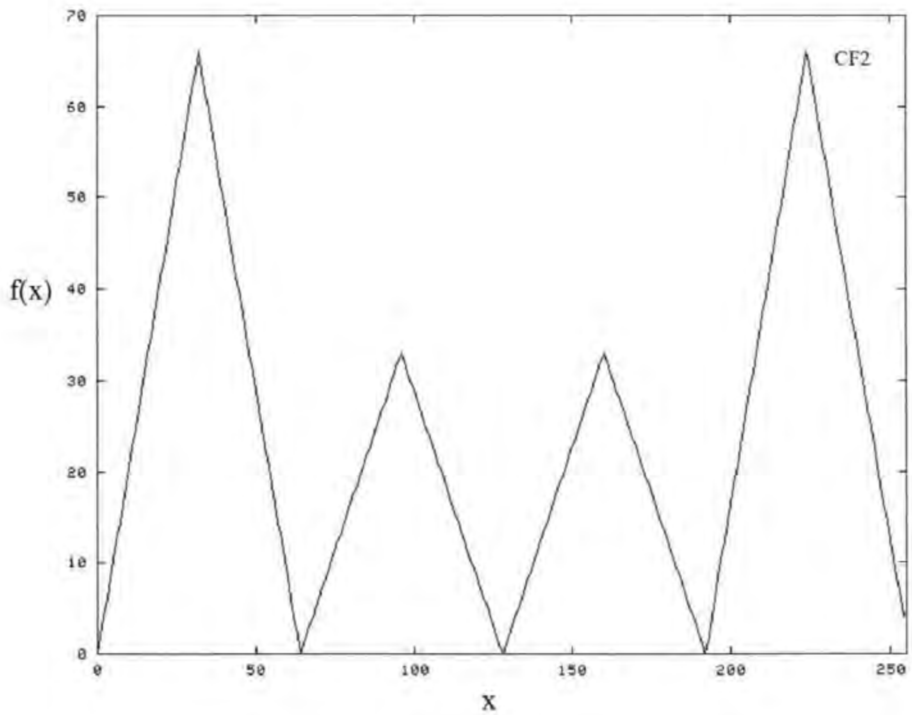


Figure 4.5: Function CF2

used in testing the ability of a multimodal GA to distribute its final population on different sub-optima.

Function CF1

This is a class function where the first class is twice as fit as the other three classes (Figure 4.4). The classes are equally spaced. DC has been observed to have the dominance problem with this function [Mahfoud (1994)]. In this case the peak belonging to the first class is called the dominating peak, and that dominates its less fit neighbour the second peak.

Function CF2

CF2 (Figure 4.5) is a modification of the class function CF1 where the fourth class has also been made to be dominating. The first and fourth classes are equally fit but twice as fit as the second and the third. It is observed that when DC is applied to this problem one weaker class assists another weaker class for migration. In absence of the assistance (that is when one class is completely migrated) the weaker class is no longer dominated.

4.4.2 The Comparison Results

Results of the experiments are shown in Figures 4.6 to Figure 4.9. In the case of function F1 (Figure 4.6), ARTS can maintain all the five peaks. The population is distributed among the peaks upon reaching a steady state of population distribution. ARTS takes some time to attain the steady state. This can be attributed to the clustering error involved at the initial stages of the run. On the other hand RTS shows the dominance effect by losing the third peak at around 400 generations. A steady state is only maintained over a few generations. DC achieves a steady state in its population distribution after some generations, but it is observed that the final population is not distributed among the peaks only. A consequent trial with F2 also exhibits similar performances of ARTS, RTS and DC. On the class

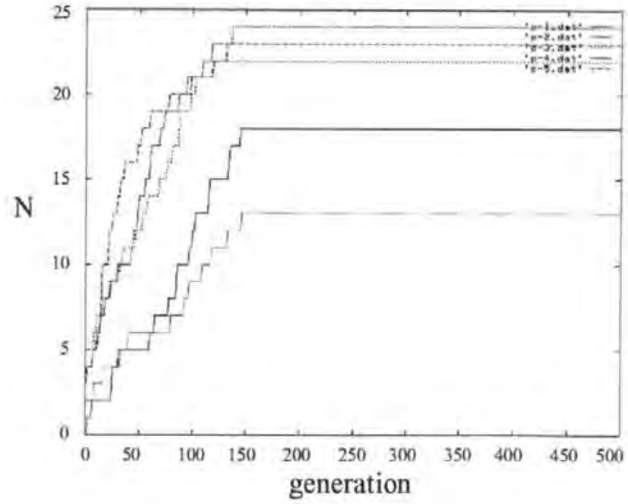


Figure 4.6(a): ARTS on F1, where N is the number of elements on each peak.

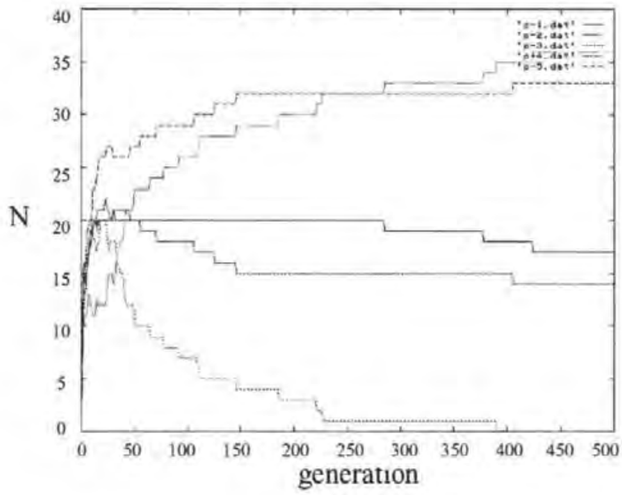


Figure 4.6(b): RTS on F1, where N is the number of elements on each peak.

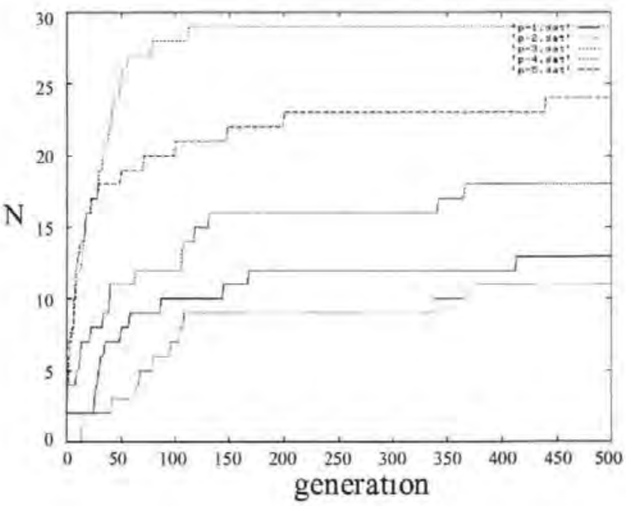


Figure 4.6(c): DC on F1, where N is the number of elements on each peak.

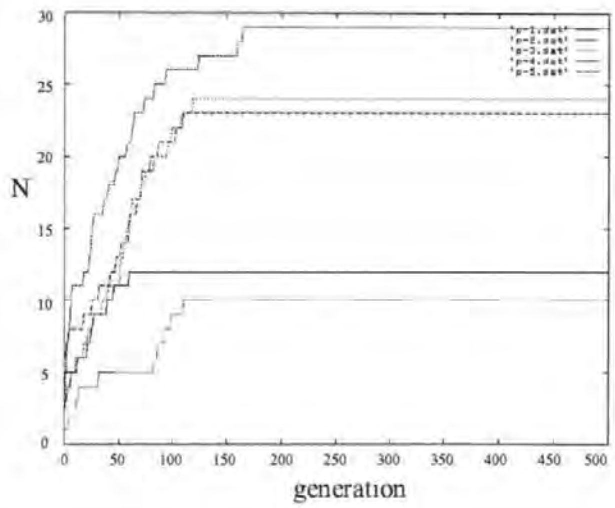


Figure 4.7(a): ARTS on F2, where N is the number of elements on each peak.

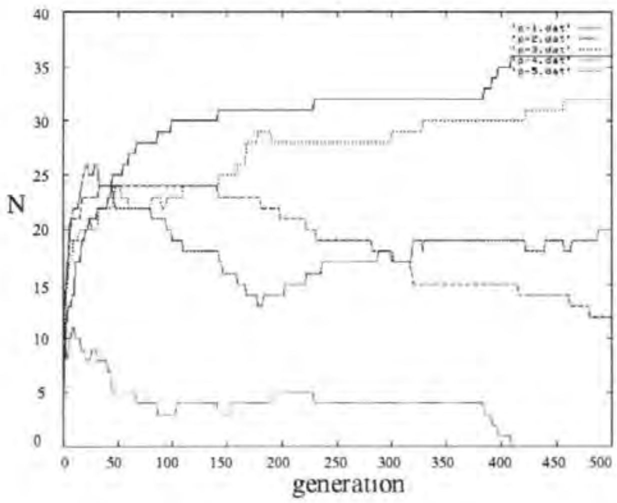


Figure 4.7(b): RTS on F2, where N is the number of elements on each peak.

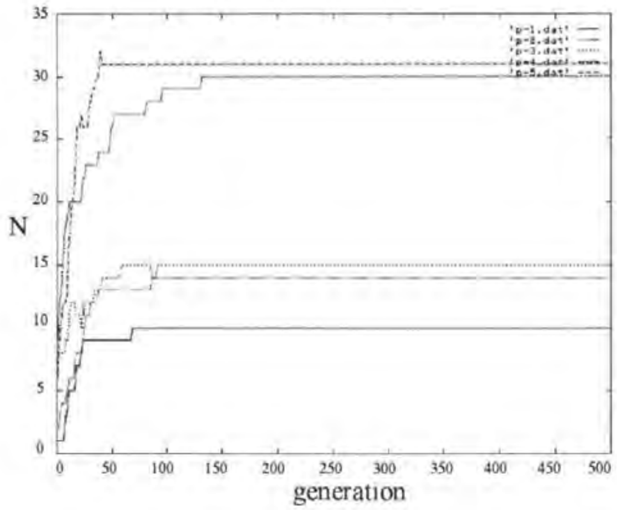


Figure 4.7(c): DC on F2, where N is the number of elements on each peak.

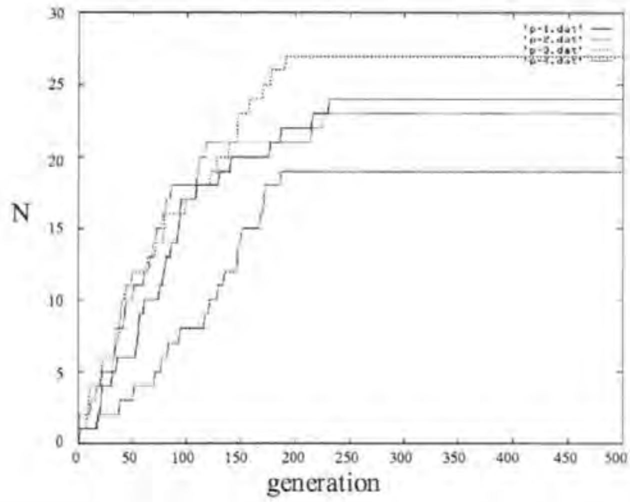


Figure 4.8(a): ARTS on CF1, where N is the number of elements on each peak.

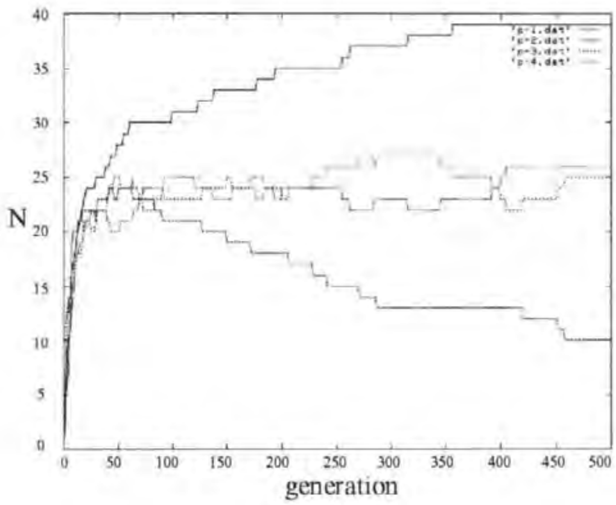


Figure 4.8(b): RTS on CF1, where N is the number of elements on each peak.

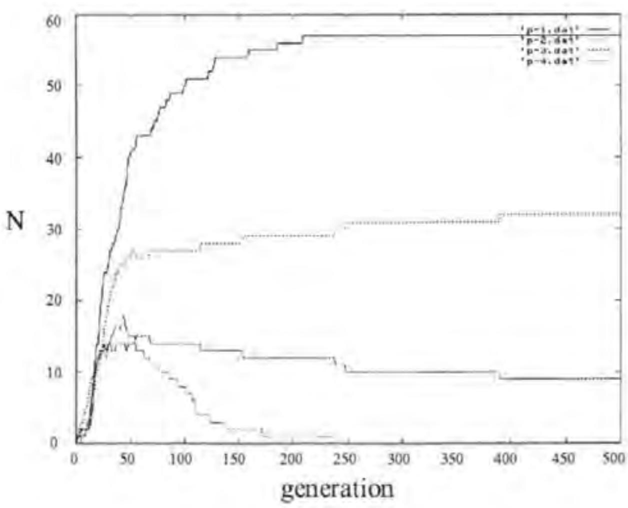


Figure 4.8(c): DC on CF1, where N is the number of elements on each peak.

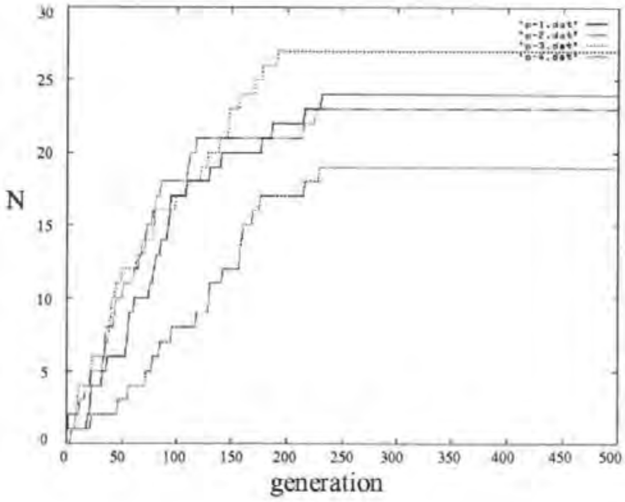


Figure 4.9(a): ARTS on CF2, where N is the number of elements on each peak.

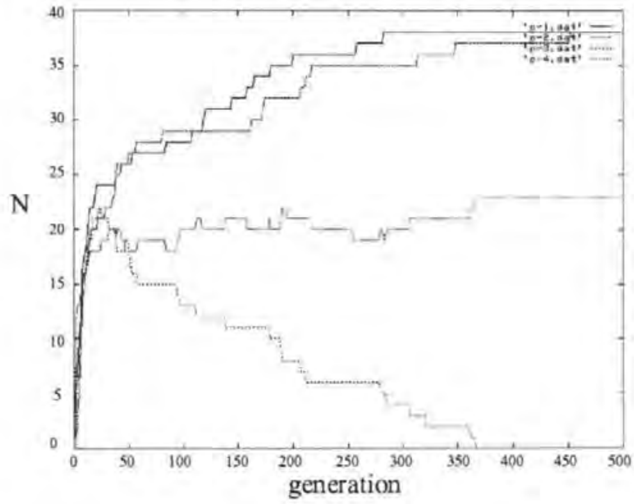


Figure 4.9(b): RTS on CF2, where N is the number of elements on each peak.

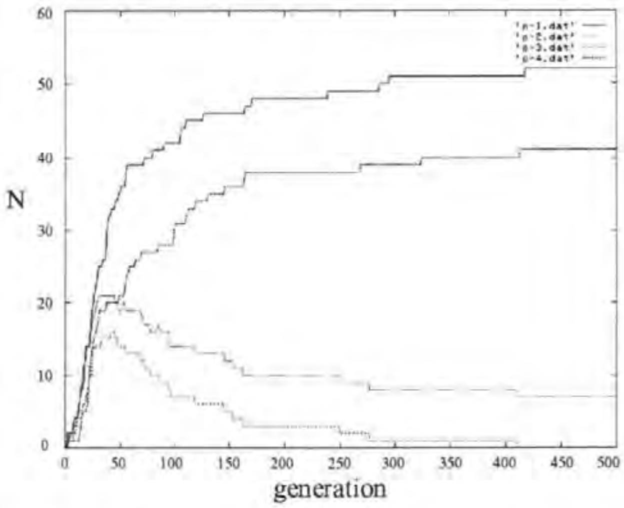


Figure 4.9(c): DC on CF2, where N is the number of elements on each peak.

function CF1, DC and RTS also exhibit the dominance problem. RTS never achieves a steady distribution among the peaks whereas, ARTS takes about 250 generations to achieve the steady state. However, once the distribution is achieved all the members of the population are distributed on the peaks. In the case of function CF2, DC performs as expected [Mahfoud (1994)], that is, two peaks dominate the other two. RTS also exhibits the dominance factor on this function, i.e. the two fitter peaks dominate the other two peaks; whilst ARTS performs consistently well as before. From these experiments it is evident that, ARTS has avoided the problem of dominance and can distribute its population among the peaks once it reaches the steady state. ARTS achieves this without any prior knowledge about the modality of the search space.

To analyse further, RTS has also been tested on functions F1 and F2 with three different window sizes 15, 20 and 25. In each case ten random runs are performed. The variance of the number of elements on each peak is presented in Figures 4.10 and 4.11. The figures show that, RTS cannot attain a steady state of population distribution on the peaks and in a few cases peaks are totally lost after some generations. Figure 4.10 exhibits that for function F1 the search is less robust with a smaller window size (i.e. there is a higher variance). On the other hand a larger window size of 25 introduces more stability to the search (i.e. smaller variance). Figure 4.11 also shows that for function F2 the performance of the search is improved using a higher window size. The larger window size of 25 helps to maintain all the peaks for a longer period.

4.5 A Study on the Effects of the Clustering Parameters, K and KT, on ARTS

The shared near neighbour clustering technique is controlled by the two parameters K and KT. It is important to understand the effect(s) of the two parameters on the ARTS based

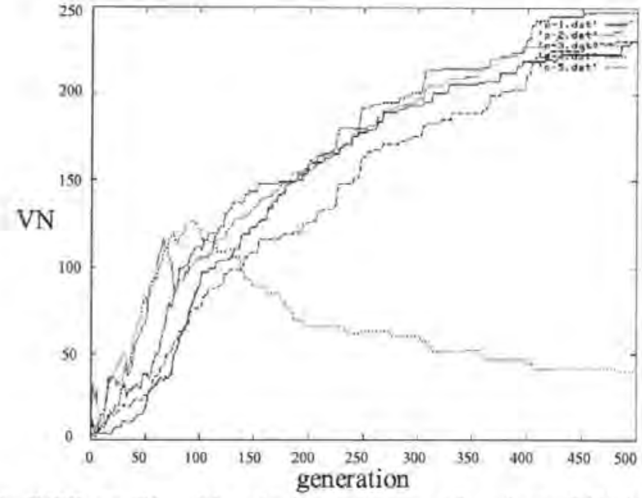


Figure 4.10(a): RTS on F1 with window_size = 15, where VN is the variance of the number of elements on each peak over ten random runs.

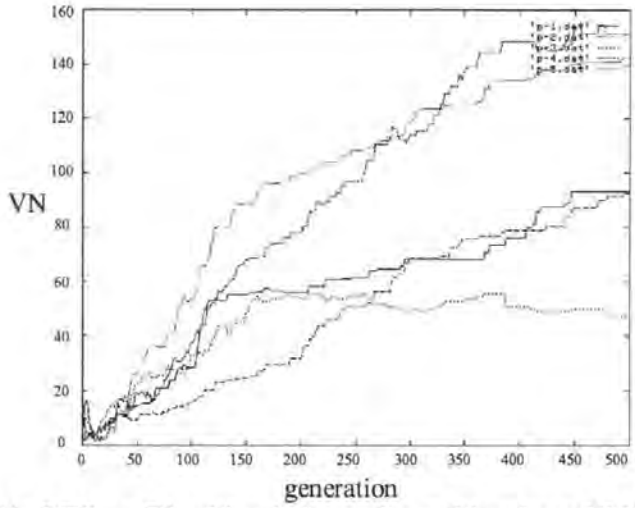


Figure 4.10(b): RTS on F1 with window_size = 20, where VN is the variance of the number of elements on each peak over ten random runs.

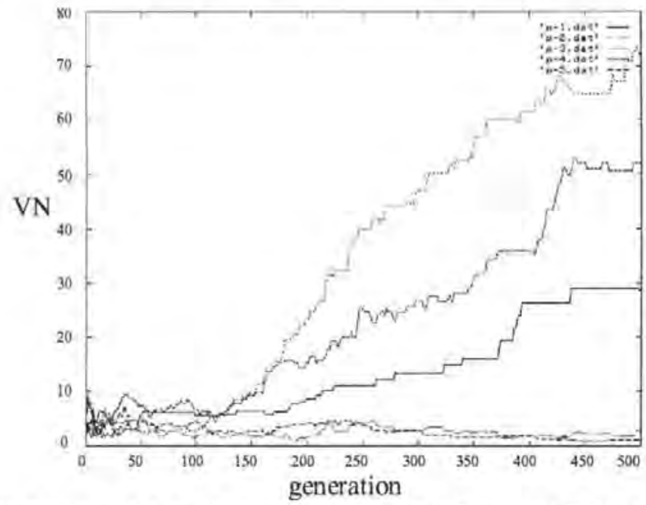


Figure 4.10(c): RTS on F1 with window_size = 25, where VN is the variance of the number of elements on each peak over ten random runs.

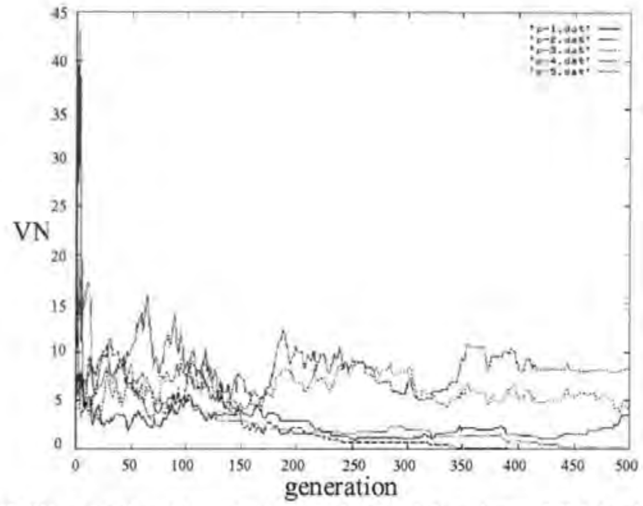


Figure 4.11(a): RTS on F2 with window_size = 15, where VN is the variance of the number of elements on each peak over ten random runs.

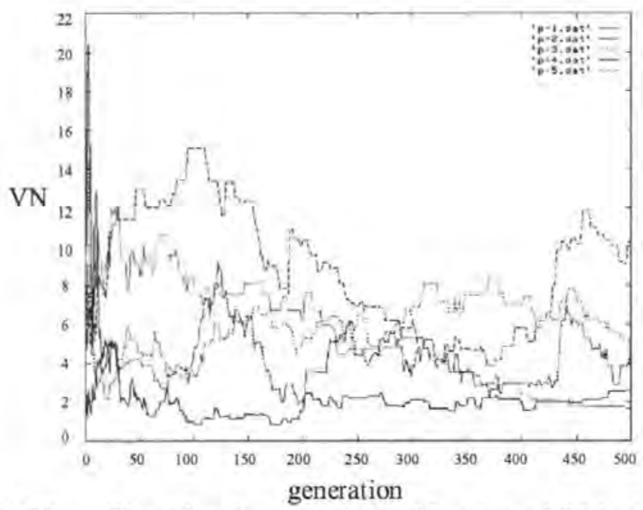


Figure 4.11(b): RTS on F2 with window_size = 20, where VN is the variance of the number of elements on each peak over ten random runs.

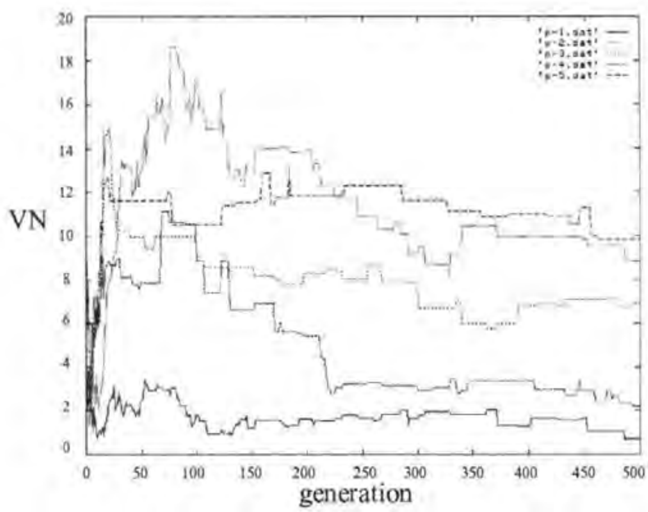


Figure 4.11(c): RTS on F2 with window_size = 25, where VN is the variance of the number of elements on each peak over ten random runs.

GA search. In order to study the effect, the ARTS based search is performed with different values of KT whilst keeping K constant. The value of KT is varied from 15 to 8, while K is kept at 15. The study is performed on both F1 and F2 functions. When K and KT are the same, that is when they are both 15, the tightest possible clusters are produced. Reducing the value of KT from 15 results in less tight clusters. Ten random runs are performed for each combination of K and KT. The average and variance of the number of elements present on each peak with KT equal to 15 and 8 only are plotted in Figures 4.12 to 4.15. The experiments show that in all the cases ARTS is found to have achieved a steady state of population distribution and the performances are similar. The value of K does not affect the clustering significantly, and generally K is fixed at 15 with a population size of 100. The value of K is suitable to provide the necessary globular bias to the clustering.

4.5.1 Chi-square-like performance test for different values of KT

ARTS is tested with different values of KT for the clustering. The value of K is kept the same. The final population distributions on the peaks of F1 and F2 with the tightest clustering (that is K and KT are set equal) are used as the benchmarks. For each function, KT is varied from 15 to 8 and the population distribution is noted for 500 generations, while the value of K is kept fixed at 15 only. The experiments use the GA parameters as mentioned in the section 4.4. The final population distributions (that is at generation 500) with KT from 14 to 8 are compared with the benchmark distributions. The chi-square-like performance statistic is used to determine how far the final population distributions (that is with KT from 14 to 8) differ from the respective benchmark distribution. This measures the effect of different values of KT on the performance of ARTS based GA search. The chi-square-like measure [Deb and Goldberg (1989), Miller and Shaw (1996)], given below, returns a positive number that decreases as the two distributions become closer; it returns 0 if the distributions are identical.

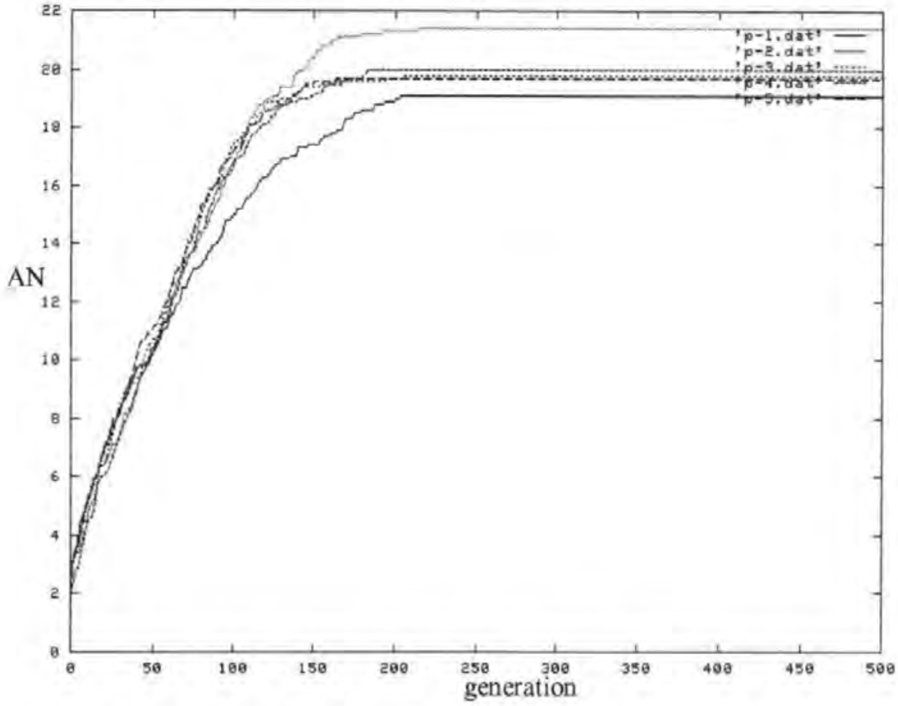


Figure 4.12(a): ARTS on F1 with KT = 15, where AN is the average number of elements on each peak over ten random runs.

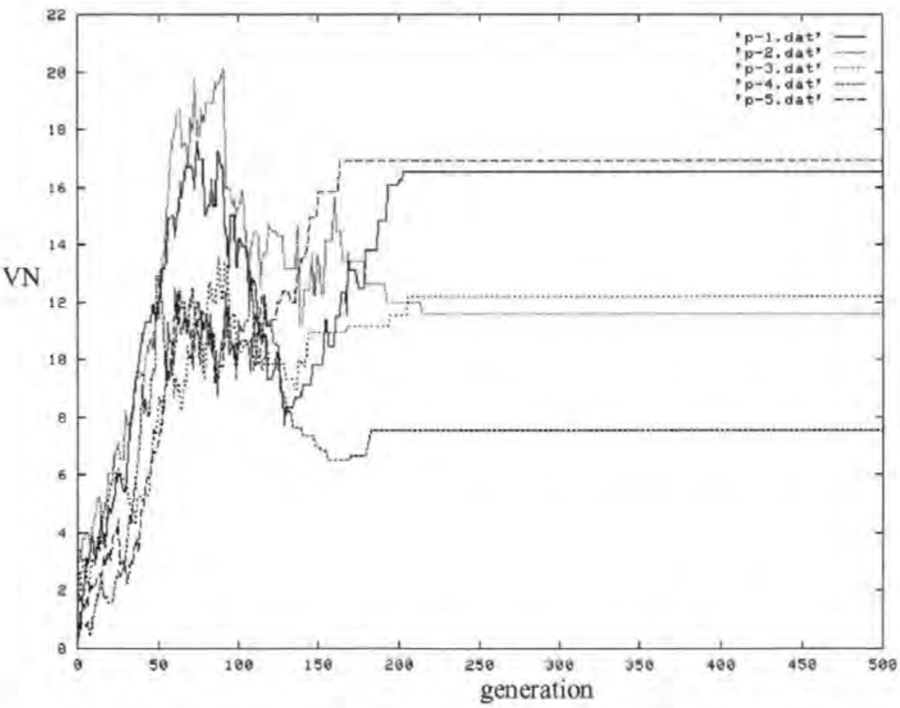


Figure 4.12(b): ARTS on F1 with KT = 15, where VN is the variance of the number of elements on each peak over ten random runs.

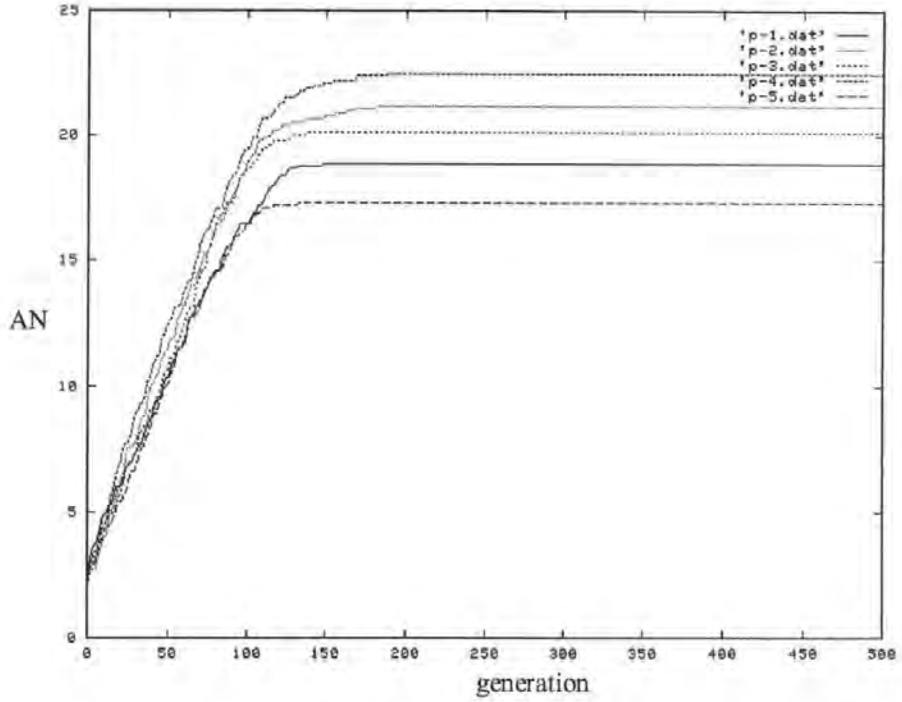


Figure 4.13(a): ARTS on F1 with KT = 8, where AN is the average number of elements on each peak over ten random runs.

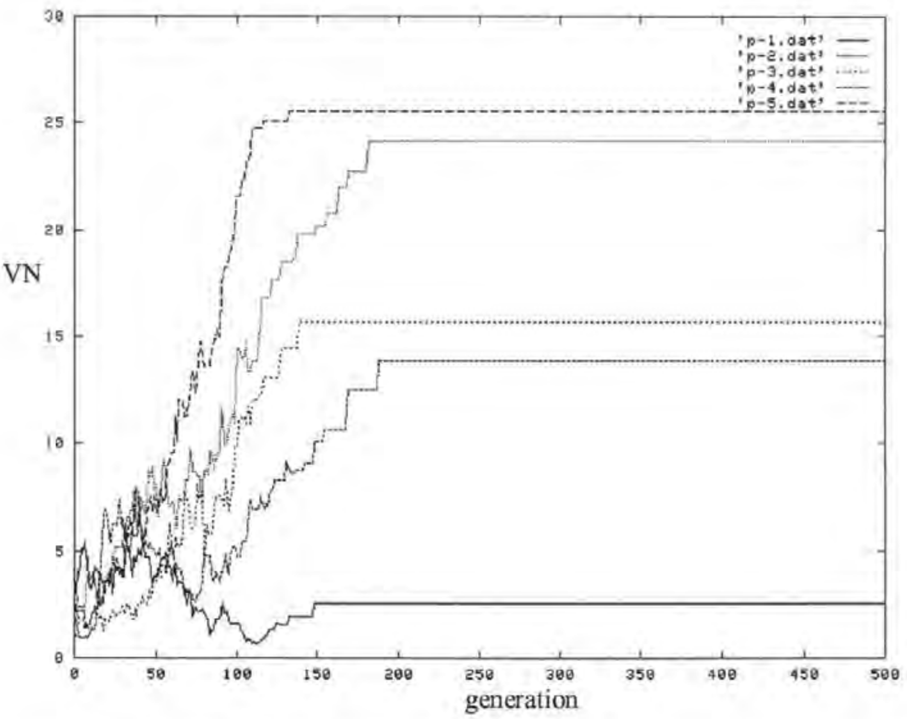


Figure 4.13(b): ARTS on F1 with KT = 8, where VN is the variance of the number of elements on each peak over ten random runs.

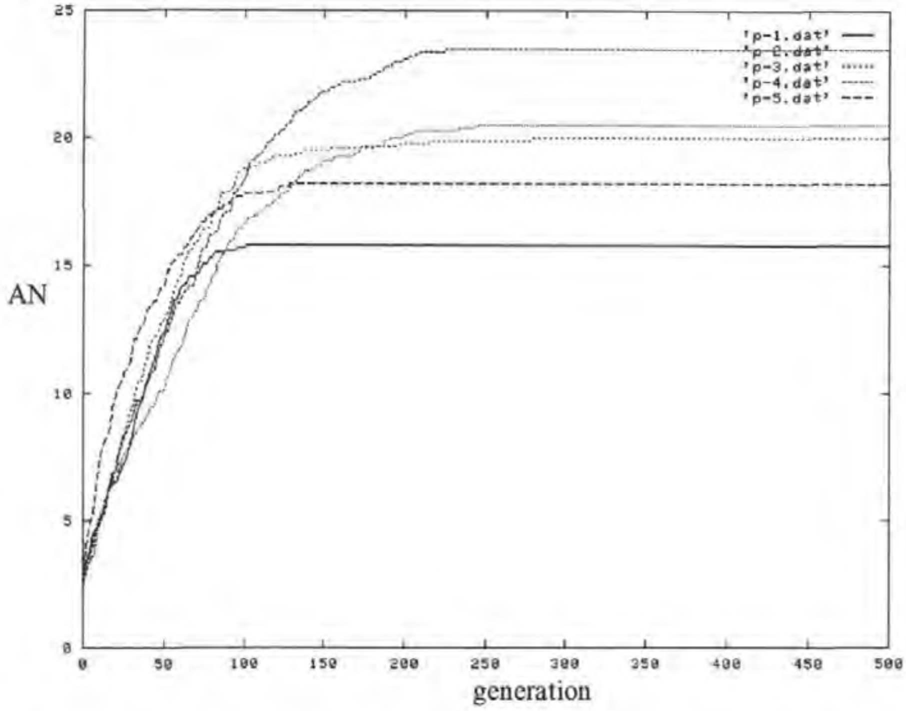


Figure 4.14(a): ARTS on F2 with KT = 15, where AN is the average number of elements on each peak over ten random runs.

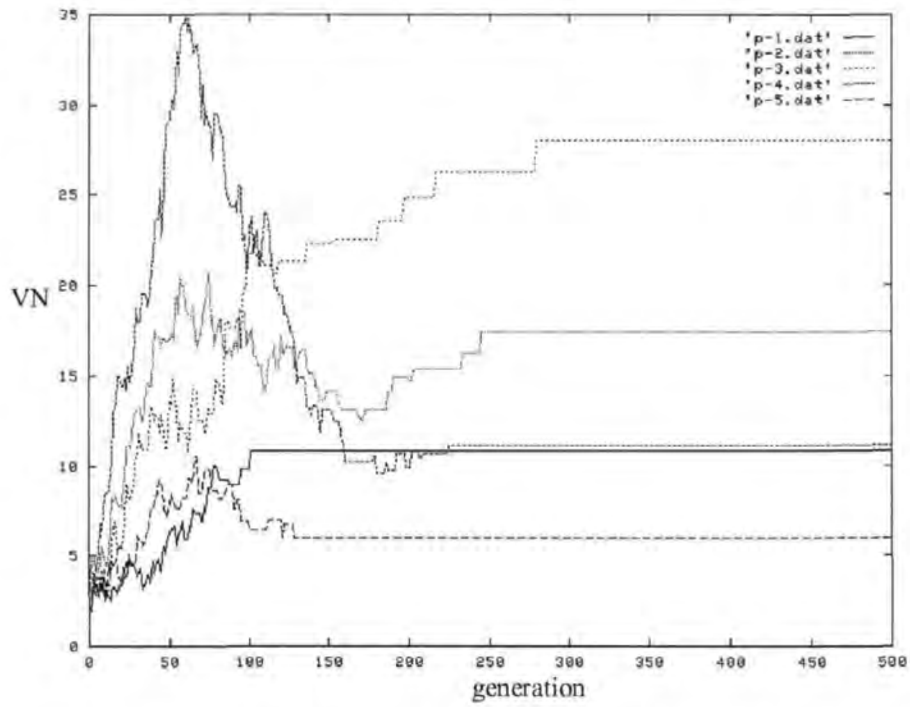


Figure 4.14(b): ARTS on F2 with KT = 15, where VN is the variance of the number of elements on each peak over ten random runs.

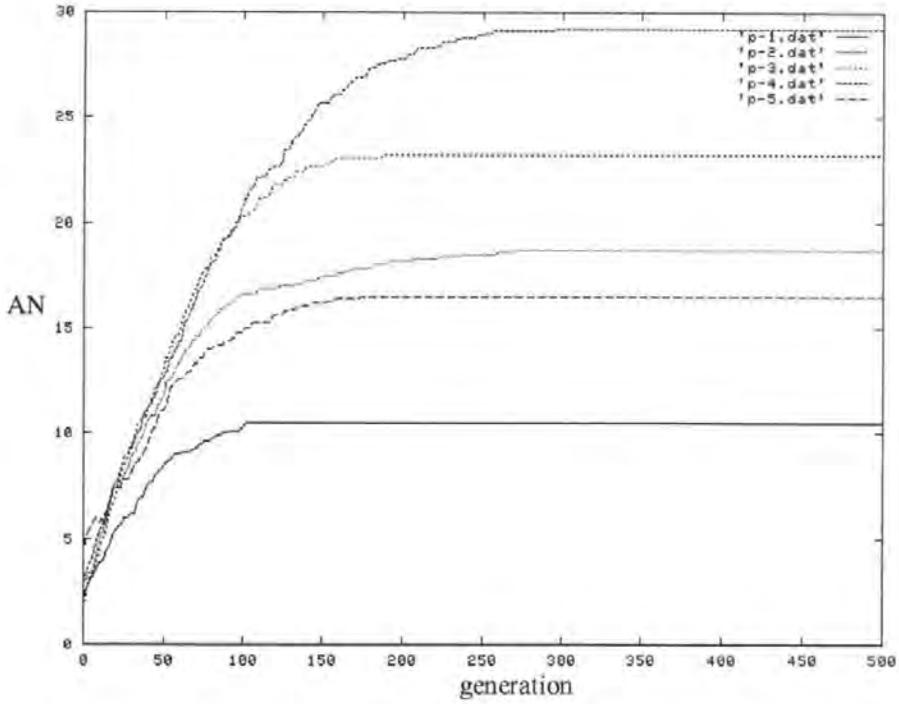


Figure 4.15(a): ARTS on F2 with KT = 8, where AN is the average number of elements on each peak over ten random runs.

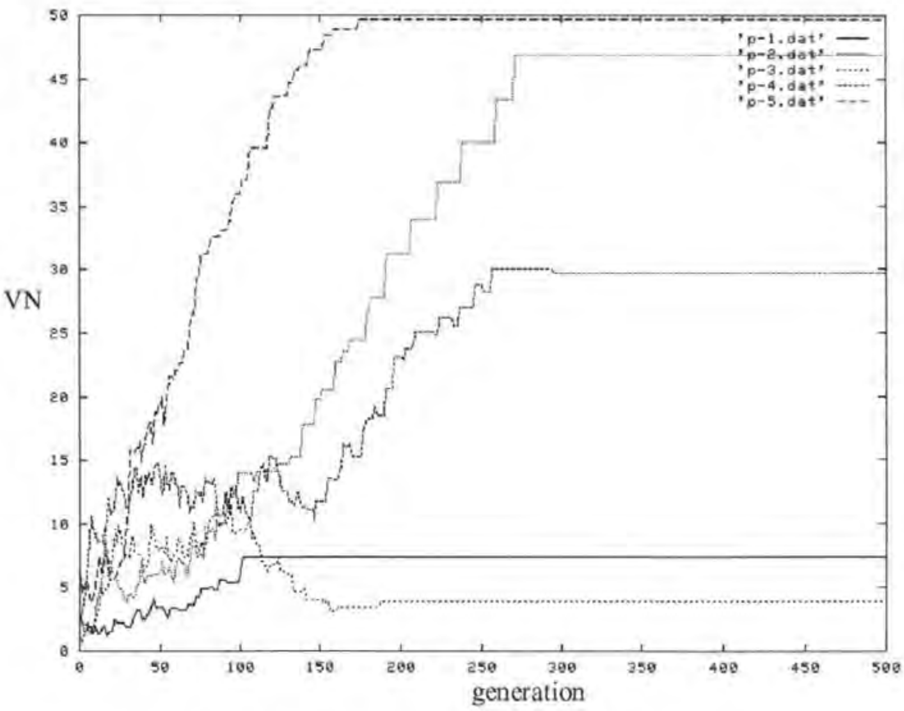


Figure 4.15(b): ARTS on F2 with KT = 8, where VN is the variance of the number of elements on each peak over ten random runs.

$$Chi\text{-}square\text{-}like\ performance\ measure = \sqrt{\sum_{i=1}^{q}\left(\frac{X_i - \mu_i}{\sigma_i}\right)^2} \qquad ...(4.1)$$

The chi-square-like performance metric measures the deviation of the actual distributions of individuals, X_i, from the benchmark distribution mean μ_i on all the i peaks (there are total q number of peaks). The variable X_i represents the actual number of individuals on the peak i. The average and the standard deviation of the number of individuals on the ith peak in the benchmark distribution are denoted by μ_i and σ_i respectively. The smaller the chi-square-like performance measure the closer are the two distributions. Tables 4.1 and 4.2 present the benchmark distributions for the functions F1 and F2 respectively.

Peak Number	μ_i	σ_i
1	19.1	4.07
2	21.4	3.40
3	19.8	3.49
4	20.0	2.75
5	19.7	4.11

Table 4.1: The benchmark population distribution on the peaks of F1 (where the peaks are counted from the left in figure 4.2). Here K and KT are kept equal at 15.

Peak Number	μ_i	σ_i
1	15.8	3.29
2	20.5	4.17
3	20.0	5.29
4	23.5	3.34
5	18.2	2.44

Table 4.2: The benchmark population distribution on the peaks of F2 (where the peaks are counted from the left in figure 4.3). Here K and KT are kept equal at 15.

Ten random runs are performed for every combination of K and KT. Experiments are performed for both the functions, F1 and F2. The average and standard deviation (SD) of the chi-square-like statistic over the ten runs for each combination of K and KT are presented in Tables 4.3 and 4.4. It is clear from Table 4.3 that in all the seven cases for the function F1, the average chi-square-like measures are quite small and of similar value. The corresponding standard deviations are also reasonably small. These show that for function F1, in terms of attaining the steady state ARTS search is robust to changes in KT. Also Table 4.4 exhibits a similar trend for the function F2. It is observed that in Table 4.4 the average and standard deviation tend to increase with decreasing values of KT, but they are still quite low. Very loose clustering is performed when KT is set to 8. This introduces more clustering error in the search. Though the search attains steady state of distribution, the final distribution can vary from the benchmark. This is observed from the fact that the standard deviation of the chi-square-like measure is higher. Thus, in terms of attaining the steady state the performance of ARTS based GA can be considered as reasonably robust with different values of KT, while K remains constant.

K	**KT**	**Chi-Square-Like measure**	
		Average	**SD**
15	14	2.2331	0.9888
15	13	2.4290	0.9404
15	12	2.5381	0.4660
15	11	3.1692	1.2363
15	10	3.1449	1.3300
15	9	2.3134	0.8427
15	8	2.4611	1.2041

Table 4.3: The chi-square test results for the function F1 with different values of KT.

K	KT	Chi-Square-Like measure	
		Average	SD
15	14	2.0436	0.5623
15	13	2.0472	0.9137
15	12	3.3653	1.0135
15	11	2.8944	1.0590
15	10	4.2240	1.5466
15	9	4.0833	1.4650
15	8	3.9397	2.1321

Table 4.4: The chi-square test results for the function F2 with different values of KT.

4.6 The Identification of "Good" Design Solutions using ARTS

The developed technique, ARTS, is applied on the twelve dimensional turbine blade cooling system design problem. The problem involves three non-linear constraints. The objective is to identify several sub-optima or in other words multiple "good" design solutions present in the constrained design space.

4.6.1 Genetic Encoding of the Design Variables

The turbine blade problem includes three types of geometry for the cooling passage. Types of geometry determine the ranges for the coefficient of discharge (Cdf) and the factor for heat transfer coefficient (Fhc). A structured chromosome approach [Dasgupta and McGregor (1991)] is implemented using binary encoding. The structure of the chromosome is shown in Figure 4.16. Every variable is defined by a maximum value, a minimum value, a resolution and a design tolerance. Every variable is represented by an eight bits long string.

4.6.2 The Constrained Optimisation

The turbine blade cooling system design involves three non-linear constraints. Any new technique should be able to handle this constrained optimisation task. Michalewicz (1995) has listed several techniques for constrained optimisation. The most popular technique uses penalty functions, where the fitness (that is the inverse of coolant mass flow through the radial passage) of a solution is degraded if it violates any constraint. The problem uses three linear penalty functions for the three constraints. The penalty functions (Figure 4.17) help the GA to concentrate search in the feasible regions of the search space.

4.6.3 ARTS for the Design Problem

ARTS is applied to the turbine blade design problem to identify multiple "good" design solutions. The solutions are presented to the designer by ASM for design decision support. ARTS uses the shared near neighbour clustering technique to cluster the elements or design solutions present in every population. The clustering time depends on the total number of elements to be clustered. As an ARTS based GA run progresses, some duplicate solutions are produced. In order to reduce the clustering time a clustering list is developed by eliminating the duplicate designs from every generation. Thus the clustering list changes its size and becomes smaller as the run progresses. The clustering list is used to identify smaller clusters present in the population. In an initial attempt [Roy and Parmee (1996)], the two control parameters of the clustering technique, K and KT, were set equal but proportional to the size of the clustering list. That helped to achieve the tightest clustering possible. In a later development, an attempt has been made to integrate a knowledge based hill climbing technique (KBHC) with ARTS. KBHC is discussed in detail in the next section. KBHC works on every generation and tries to improve the "good" designs (that is the best design of each cluster) utilising designers' prior knowledge and information extracted from the clusters. Designers' prior knowledge represents a heuristic concerning the contribution of

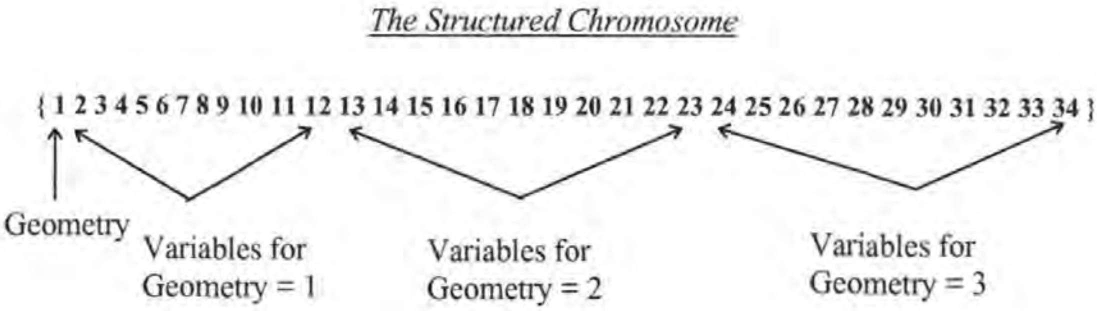


Figure 4.16: The structured chromosome that is used for the design problem, where 1, 2, 3, 4 ... are gene (each gene represents one design variable and is represented by a binary string) numbers.

```
cons1 = 1.0            ;; penalty factor for constraint one
cons2 = 1.0            ;; penalty factor for constraint two
cons3 = 1.0            ;; penalty factor for constraint three

If violation1 > 150.0, cons1 = 0.0
Else  cons1 = 1.0 - (1.0/150.0)*violation1

If violation2 > 150.0, cons2 = 0.0
Else cons2 = 1.0 - (1.0/150.0)*violation2

If violation3 > 0.4, cons3 = 0.0
Else cons3 = 1.0 - (1.0/0.4)*violation3

constrained_fitness = unconstrained_fitness*cons1*cons2*cons3 / 10.0
```

Figure 4.17: The penalty functions used for the problem, where violation1, violation2 and violation3 represent the amounts of constraint violation for the three constraints.

individual variables to the fitness. Data within a cluster also provides some information about the relative contribution of the design variables towards fitness within the neighbourhood. If a cluster contains very few designs, some designs are randomly generated around the best design of the cluster. These designs are then used to obtain the cluster information. For every cluster, the method learns about the best design using the cluster information, uses designer's existing knowledge and identifies the most contributing variables. The technique assumes that variables are independent. At first, real-number hill climbing is performed only on the three most contributing variables. If KBHC cannot improve the designs for some generation, the next set of three most contributing variables are hill climbed. Then if KBHC does not improve the designs for some generations the hill climbing is stopped.

Every design belongs to the hill of a local peak and KBHC tries to climb up to that peak. Thus KBHC is a local hill climbing technique. The method only searches in limited directions thus it cannot guarantee to identify the local peak, but it can climb up the hill deterministically. The technique is very quick, and may improve the best design in each cluster. Thus it is acceptable to apply the technique every generation. Whenever KBHC is successful the improved design replaces the best design of the cluster and its duplicates in the population. This improved population is then reclustered to provide information for the next generation.

It is important that the search attains a steady state to distribute the design solutions on different sub-optima. In the initial attempt, whether the ARTS based GA has attained a steady state was determined by checking the average fitness of the population every generation. If the average fitness remained unchanged for a certain number of generations it was assumed that the GA has attained a steady state [Roy and Parmee (1996)]. This steady state criterion is found to delay the ASM run and does not contribute to the search at the later stage of a run. Thus the criterion is not very suitable for industrial applications. In order to achieve "good" designs within a reasonable time, the steady state criterion has been changed. The ASM now maintains three lists of "good" designs for the three geometry types. The lists are of fixed size and the sizes are determined heuristically. A list is updated under the following circumstances:

a) If the list is not of full size: the list is updated until the list attains the full size.

b) The lists are updated every generation only if a better design is found outside the neighbourhoods of the designs in the lists, but within the same geometry type. The neighbourhood of a design is defined by the tolerances on each dimension. The better design replaces the worst design in the list.

c) If a better design lies in the neighbourhood of a design from the list, the better design replaces the design in the list.

If all the three lists are not updated for some generations it is assumed that the search has reached steady state. Thus, the objective of the search is redefined as '*only five best designs are required from each geometry type*' (that is a total of 15 designs using an initial population of 120). KT is assigned 90% of the value of K. This smaller value of KT (that is smaller than K) provides bigger clusters. The clusters provide information for the KBHC search. Once the GA reaches a steady state, the best solution in each cluster is considered as a potential "good" solution. In an attempt to reduce the run time of ASM for the turbine blade problem following improvements are also introduced:

A. *An Effective Crossover Technique*: The structured chromosome used to represent the problem results in a large amount of redundancy in a chromosome (in this case there is about 66% redundancy present in the chromosome). If the one-point crossover position is selected within the redundant areas of the parent chromosomes, the children produced would not be different from their parents. This makes the crossover ineffective on some occasions and thus prolongs the search. An effective crossover technique is developed that prevents crossover in the redundant regions of the parent chromosomes. The crossover position is selected at least within the active regions of a parent chromosome. This improves the effectiveness of the crossover technique, and thus ASM run time is reduced [Wade et. al. (1994)].

B. During an ARTS based GA search some duplicate solutions are produced in the population. Thus, randomly selecting two individuals from the population may mean selecting duplicate chromosomes as parents. Mating of identical chromosomes cannot produce any new schema; and as a result the effectiveness of the reproductive stage is reduced. In order to avoid the selection of two similar chromosomes as parents, they are selected from the cluster list, whilst the cluster list is developed from the population after eliminating the duplicate designs [Eshelman and Schaffer (1991)].

The potential "good" design solutions were validated by randomly checking the fitness of many solutions from the neighbourhood. It was observed that, although the fitnesses looked very promising most of them were actually not local optima. The solutions achieved were found to be close to the local optima. This difficulty can be attributed to the inefficiency of the GA and KBHC hybrid to exactly locate a sub-optimum, specially if the problem is complex and multidimensional. At the end a stochastic local hill climbing algorithm (described in section 4.8) is also applied on each potential "good" solution to ensure that the local peak is attained.

4.7 The Knowledge Based Hill Climbing Technique

A Knowledge Based Hill Climbing (KBHC) technique is developed to be used with the Adaptive Search Manager (ASM) [Roy et. al. (1996a)]. The KBHC technique learns from the cluster information gathered by the ASM, uses prior knowledge of experts, and deterministically performs a limited hill climbing. The objective is to improve upon a design with a very small number of trials. The technique assumes that the clusters represent the neighbourhoods of the design solutions, and that there is very little interaction between the design variables. KBHC works with the principle of Bayes' Theorem. The theorem provides a learning framework that identifies the interesting variables to hill climb. The hill climbing is limited within a type of the geometry. The technique is applied every generation on the best design of each cluster. KBHC is stopped if it cannot improve the designs for a few generations.

4.7.1 Learning from a Single Data Set using Bayes' Theorem

It is assumed that a designer considers a finite list of models for the design task; where each model represents one variable, $\{ M_1, M_2, ..., M_k \}$, to constitute an exclusive and exhaustive set of possible probability models for the problem. It is further assumed that, before any data is obtained, the designer assigns *prior probabilities*, $\{P(M_1), P(M_2), ..., P(M_k)\}$, (prior probability represents designer's heuristic knowledge about the problem and is represented as the degree of belief) to these models, where $0 \leq P(M_i) \leq 1$; $i = 1, 2, ..., k$, k is the number of variables and

$$P(M_1) + P(M_2) + ... + P(M_k) = 1. \qquad ....(4.1)$$

Each probability model defines a probability distribution over the possible data that may be obtained. In particular, if the acquired data set is denoted by D, the probabilities of the data as defined by each of the alternative models are given by the conditional probabilities

$$\{ P(D/M_1), P(D/M_2), ..., P(D/M_k) \}.$$

Considering in terms of the $\{ M_1, M_2, ..., M_k \}$, for a given D, the above quantities are often referred to as the *likelihood* of the M_i's given D.

After considering an exclusive and exhaustive set of probability models the designer specifies a set of prior probabilities. Assuming that the design variables are independent, for an actually obtained data D, univariate linear regression analysis coefficient, *b*, can provide a measure of the likelihood. Thus:

$$P(D/M_i) = \frac{b_i}{\sum_{j=1}^{k} b_j} \qquad ....(4.2)$$

The designer may now wish to revise the prior probabilities in the light of the information provided by the data. Expressed mathematically, the designer would wish to calculate the probabilities for the alternative models, conditional now on having the observed data D:

$$\{P(M_1/D), P(M_2/D), ..., P(M_k/D)\},$$

The mathematical result that expresses these *posterior probabilities* in terms of the prior probabilities and the likelihood is defined by *Bayes' Theorem*. The theorem for the situation under consideration can be stated as follows:

BAYES' THEOREM (in the discrete form): If $\{M_1, M_2, ..., M_k\}$ are an exclusive and exhaustive set of probability models, and the prior probabilities $\{P(M_1), P(M_2), ..., P(M_k)\}$ and likelihood $\{P(D/M_1), P(D/M_2), ..., P(D/M_k)\}$ are specified such that $P(D) > 0$, then the posterior probabilities are given by [Lloyd (1984)]:

$$P(M_i/D) = \frac{P(D/M_i)P(M_i)}{P(D)} \quad \text{where, } i = 1, 2, ..., k . \qquad ....(4.3)$$

and,

$$P(D) = P(D/M_1).P(M_1) + P(D/M_2).P(M_2) + ... + P(D/M_k).P(M_k) \qquad ....(4.4)$$

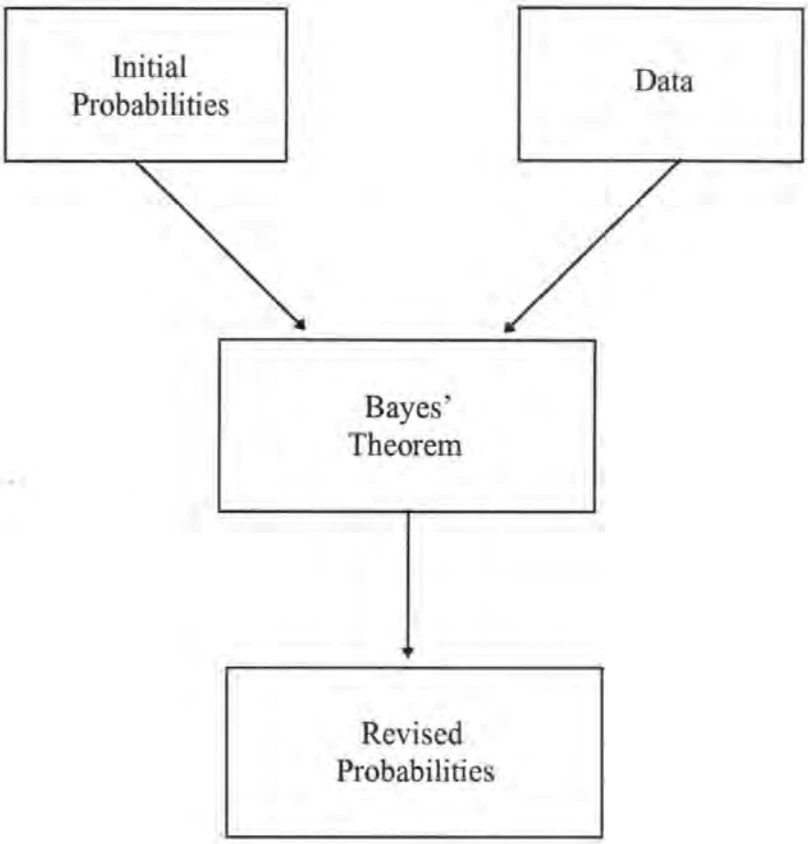


Figure 4.18: The fundamental principle of the Bayesian paradigm.

What distinguishes the Bayes' theorem from other statistical approaches is that, *prior to obtaining the data, the statistician considers his degrees of belief for the possible models and represents them in the form of probabilities* [Lloyd (1984)]. Once the data is obtained, the theorem enables the statistician to calculate a new set of probabilities, which represent revised degrees of belief in the possible models, taking into account the new information provided by the data. For a given set of possible models, the fundamental process underlying Bayesian approach is summarised schematically in Figure 4.18.

4.7.2 The Methodology

The hill climbing methodology can be described as follows:

1. ASM produces clusters of design solutions each generation, and KBHC obtains information from these clusters. The clusters need to have a minimum number of designs in order to provide meaningful information. If a cluster does not have a minimum number of designs then some solutions are generated randomly in the neighbourhood of the best solution, where the neighbourhood is defined by the resolution on each variable.

2. *Univariate Regression Analysis* :

To find the univariate regression coefficient:

$$\bar{y}_i = a_i + b_i \bar{x}_i \qquad \text{....(4.5)}$$

where, i = 1, 2, ..., n; n being the number of variables.

$$b_i = \frac{\sum_{j=1}^{m} x_{ij} y_{ij} - \sum_{j=1}^{m} x_{ij} \sum_{j=1}^{m} y_{ij} \Big/ m}{\sum_{j=1}^{m} x_j^2 - \left(\sum_{j=1}^{m} x_j\right)^2 \Big/ m} \qquad \text{....(4.6)}$$

$$a_i = \bar{y}_i - b_i \bar{x}_i \qquad \text{....(4.7)}$$

and, m = number of data.

Residual (error) sum of squares = SSR_i = $\sum_{j=1}^{m} (y_{ij} - \hat{y}_{ij})^2$ ....(4.8)

where, $\hat{y}_{i1}, \hat{y}_{i2}, \ldots, \hat{y}_{im}$ are obtained by substituting the x value for each observation into the least-squares lines: $\hat{y}_{i1} = a_i + b_i x_{i1}, \ldots, \hat{y}_{im} = a_i + b_i x_{im}$.

Estimated Standard Deviation = Se_i = $\sqrt{\frac{SSR_i}{m-2}}$ ....(4.9)

Estimated standard deviation of the statistic $b_i = Sb_i = \dfrac{Se_i}{\sqrt{\sum_{j=1}^{m} x_{ij}^2 - \left(\sum_{j=1}^{m} x_{ij}\right)^2 \Big/ m}}$

....(4.10)

The probability distribution of the standardised variable:

$$t_i = \frac{b_i - \beta}{Sb_i} \qquad \text{....(4.11)}$$

is the t distribution with m-2 degrees of freedom.

For each variable the calculation is performed in the following sequence:

a) Calculate b_i

b) Calculate t_i

c) Get the *critical* value of t_i from standard table for 95% confidence interval and (m-2) degrees of freedom.

d) In order to conclude that there is a linear relation between a design variable and the fitness, the converse of this *research hypothesis*, the *null hypothesis*, needs to be rejected. The logic is similar to the mathematical method of proof by contradiction. Thus if the null hypothesis (H0: $\beta = 0$) is rejected the b_i value can be used as the measure of likelihood. Otherwise, the model does not appear to satisfy a useful way of predicting the dependent variable. The null hypothesis is rejected if $t_i > t_{critical}$ or $t_i < -t_{critical}$.

3. *The Designers' Knowledge*: The pre-probability represents designers' heuristic knowledge about the contribution of individual variables to the fitness. The probability is represented as the designers' degree of belief.

The knowledge based hill climbing learns from the designs in a cluster and also uses some prior knowledge from the designer. The probability of individual variable to be the most contributor for the fitness is initially given by the designer from his experience. Following assumptions are made:

a) The designer has some prior knowledge about the important variables. This information is not necessarily true in local regions.

b) The cluster data alone cannot provide enough information about the neighbourhood of the design because they are too small in number. There is uncertainty involved regarding any information retrieved from the data. This can be considered as a degree of disbelief.

c) Information gathered from the cluster data and the designers' prior knowledge can provide a more realistic assessment of a local region in the search space.

4) The values of the posterior probability, $P(M_i/D)$, are used to identify the six most contributing variables.

5) To start with, a deterministic real number hill climbing is performed on the first three most contributing variables. The hill climbing starts in the best direction and then climbs other directions in the order.

6) After few generations KBHC becomes less effective because the three variables achieve their optimum value. If KBHC is not successful for a few generations, then the second set of three of the six most probable variables are hill climbed. KBHC is then stopped if it fails

to improve the designs for few generations. After the hybrid of ARTS based GA and KBHC attains a steady state of population distribution, the GA search is stopped. Finally, a stochastic local hill climbing is applied on the "good" designs for fine tuning.

4.8 The Stochastic Local Hill Climbing Technique

A stochastic local hill climbing algorithm is used to identify the sub-optimum solution present on the hill of a probable "good" design solution. The local search is again limited to a type of geometry of a potential "good" design solution. The search is performed on the constrained fitness (that is the inverse of the coolant mass flow through the radial passage) landscape. The neighbourhood of a design solution is defined by the resolutions on the design variables. The hill climbing algorithm is a local random walk technique. The algorithm can be described as follows:

```
For every best individual in the final cluster (CB):
        count = 0
        Best item = CB
        DO
                Randomly generate one individual (N) from the neighbourhood
                        of the Best item
                If (Fitness(N) > Fitness(Best item))  THEN
                        Best item = N
                        count = 0
                Else count = count + 1
        Until count = MAXcount
```

The algorithm tries to climb up the hill of a design. The algorithm stops searching if it cannot find a suitable solution within MAXcount number of trials. Thus it is not guaranteed that the algorithm will locate the sub-optimum. The hill climbing is stochastic in nature and thus may involve many model evaluations. It is observed that, the algorithm does improve a potential design solution and thus make it at least closer to the sub-optimum as defined by the model.

4.9 Validation of the hybrid search

The effectiveness of the hybrid search (ARTS + local hill climbing) is also validated with TBCOM. Results from several runs of the search are presented to an expert and a user from Rolls Royce. They checked whether the search mechanism can identify multiple 'good' design solutions (from different areas of the design space) within reasonable time. The first steady state criterion (section 4.6.3) was changed following the feedback from the validation.

4.10 Summary

This chapter discusses the developments of ARTS based GA technique for real life problems. The chapter also presents a knowledge based hill climbing and a stochastic local hill climbing technique, that are used in conjunction with ARTS for the turbine blade problem. ARTS is compared with RTS and DC, and the results are presented and discussed. Experiments are performed to analyse the effects of KT, a control parameter, on ARTS. A hybrid of ARTS and the knowledge based hill climbing is applied to the turbine blade problem to identify multiple "good" designs. The stochastic local hill climbing technique helps to fine tune the "good" designs. Modifications and enhancements to suit the hybrid algorithm to the turbine blade problem are also described. The next chapter presents how sensitivity information concerning the "good" designs is obtained using Taguchi's methodology.

CHAPTER - 5

5. Sensitivity Analysis of Engineering Designs

5.1 Introduction

Information concerning sensitivity of engineering designs can be essential for engineering decision making. Sensitivity analysis provides the information on the performance of a design when there is some minor change in the values of the design variables. Sensitivities of a design can be defined in terms of design solution sensitivity, design variable sensitivity and constraint sensitivity. The design solution sensitivity means sensitivity of a design solution performance within a defined neighbourhood. The design variable sensitivity is the effect of each design variable on the design solution performance within a defined neighbourhood. Violations of constraints within the neighbourhood of a design define the constraint sensitivity of the design. The study described here is performed with the steady state twelve dimensional computer model of the Rolls Royce turbine blade problem (TBCOM). The sensitivity analysis module is an integral part of the Adaptive Search Manager (ASM). Once an ARTS based GA search identifies multiple 'good' design solutions the sensitivity analysis is performed on each of these designs. The sensitivity information is presented to the designer in order to assist in the design decision making. The chapter defines a sensitivity index for the design solutions, a measure of design variable sensitivities and different categories of constraint sensitivity.

The study of the effect of varying independent variables (in this case the design variables) on a dependent variable (that is the coolant mass flow from TBCOM) requires the relationship between the dependent and independent variables to be known. An empirical method, known as *design of experiments*, is some times used to establish such relationship. For an empirical study all possible combinations of the values of the independent variables (also known as *factors*) are required to define the relationship using a statistical technique. This method of exhaustive trials is known as *full factorial experiments*. In many cases, it is too expensive to run a full factorial experiment, for example a multidimensional real life design problem. In this situation, a *fractional factorial experiment* can be performed where a fraction of the full factorial experiments is considered. The price of running a fractional factorial experiment is the loss of some information regarding the independent variables and their relation to the dependent variable. Taguchi advocates a systematic approach and has developed several standard *orthogonal matrices* to define the fractional factorial experiments [Phadke (1989), Roy (1993)]. The use of the orthogonal matrices involves the least amount of information loss, especially if the variables do not interact with each other. In order to avoid an exhaustive search for the sensitivity analysis, Taguchi's orthogonal matrix and the tolerances on the design variables have been utilised to define the neighbourhood of a design solution [Roy et. al. (1995b) and (1996b)]. This neighbourhood is called the tolerance space. Considering the worst case variability [Emch and Parkinson (1993)], the worst combinations of the design variables within the tolerances to satisfy the design constraints are expected. The sensitivity calculations are performed within the tolerance space of a design solution. Taguchi's methodology is followed to calculate the effect of each variable on the performance of a design solution (the performance here is measured by coolant mass flow x 10^3, as calculated by the model). The designs are also tested for constraint criticality [Sundaresan et. al. (1993)]. Depending upon the extent of constraint satisfaction within the neighbourhood of a design solution, different categories of constraint sensitivity can be defined as constraint satisfied, statistically active constraint, quasi-active constraint, peak-active constraint and constraint not satisfied. The definitions of these categories are given in section 5.5. The three types of sensitivity information are essential before one design is selected out of many design solutions.

Taguchi's methodology assumes no interaction between variables. Thus the analysis can be very reliable provided there is no or very little interaction among the design variables in the neighbourhoods of the design solutions. One way of checking for the presence of interactions is to validate the additivity principle in the region. The additivity principle assumes that the result of each experiment is the superposition of the single factor effects plus the error due to this assumption and any repetition of the tests. A comparison between the technique and an exhaustive search based sensitivity analysis is presented with more than 100 design solutions where the neighbourhoods of the design solutions maintain the additivity principle.

The research presented in this chapter demonstrates the applicability of Taguchi's methodology for an approximate sensitivity analysis. The methodology needs a very small number of model evaluations (experiments) and is thus suitable for multi dimensional real life problems. The technique is also suitable for performing in the integrated environment of ASM.

5.2 Sensitivity Analysis

The sensitivity analysis of the turbine blade cooling hole system design includes three components: calculating the design solution sensitivity, the design variable sensitivities and the constraint sensitivity. The sensitivities are calculated in the neighbourhood of a design solution. The neighbourhood is defined by a suitable orthogonal matrix and the tolerances of the design variables. The orthogonal matrix is expected to provide reliable information about the neighbourhood provided there is no or very little interaction among the design variables [Phadke (1989), Roy (1993), Roy et al. (1994)]. The three types of sensitivities are described below:

Design Solution Sensitivity:

A measure of the variation of design fitness in the neighbourhood of a design solution is defined as the design solution sensitivity. In this case the design fitness is determined by coolant mass flow x 10^3 through the radial passage.

Design Variable Sensitivity:

This is defined as the effect of a design variable on the design fitness within a neighbourhood of the design. The effects due to the interaction (if any) between variables are not considered.

Constraint Sensitivity:

The constraint sensitivity can be described as criticality of constraints (violations) in the neighbourhood of a design solution. According to the criticality five categories of constraint sensitivity have been defined: *constraint satisfied, statistically active constraint, quasi-active constraint, peak-active constraint* and *constraint not satisfied* (section 5.5).

5.3 Taguchi's Orthogonal Matrix

Taguchi's orthogonal matrix comes from the concept of *Latin Squares* that has been known in mathematics for thousands of years [Phadke (1989)]. Recently it has become popular as a tool for design of experiments [Phadke (1989), Sundaresan et. al. (1993), Roy et. al. (1994), Roy and Cave (1996a) and (1996b)]. Variable levels are orthogonal, i.e. they are represented proportionately (equal number of times) in any two columns of the matrix. Where the levels of a variable are defined as possible values of the variable, they are discrete, and there can be two or more levels of a variable. For example, inlet temperature can have three levels: *high*, *medium* and *low*. The smallest orthogonal matrix (that is designated as L_4) designs four experiments for three variables (these are the factors) with two levels each (Table 5.1). The scheme combines all factor levels with the same number of other factor levels. For example, in Table 5.1, B2 (i.e. the second level of the design variable B) is tested together with A1 and C2 in row 2 and variable settings A2 and C1 in row 4. The average of the corresponding test results R2 (that is the result of the second experiment, the second row) and R4 is different from the overall mean μ of all test results. The difference is due to the influence of B2, known as the "factor effect" (b2). Taguchi's methodology depends on the principle of additivity. If a neighbourhood maintains the additivity principle, factor effects due to the different levels of a design variable should nullify each other [Phadke (1989)].

Experiment	**Variables**			**Result**
	A	***B***	***C***	
1	1	1	1	R1
2	1	2	2	R2
3	2	1	2	R3
4	2	2	1	R4
				mean μ

Table 5.1: Standard L_4 Orthogonal Matrix. The matrix consists of three variables (A, B and C) with two levels each (denoted by 1 and 2).

For example, considering the L_4 orthogonal matrix as shown in Table 5.1, the result R2 for the second experiment can be expressed as :

$$R2 = \mu + a1 + b2 + c2 + e$$

where, a1 is the difference between the overall mean μ and the average result of all the tests which involved A1. Similarly b2 and c2 are also defined. e is the associated error.

The additivity principle assumes that the effects of different levels of a variable should cancel each other. Thus assuming the additivity principle holds in the neighbourhood:

$$a1 + a2 = b1 + b2 = c1 + c2 = 0$$

Using the above assumption it is possible to calculate the effects of individual levels for each design variable as follows:

Average result of all tests which involved A1,

$$m(A1) = \tfrac{1}{2} \{ (\mu + a1 + b1 + c1 + e1) + (\mu + a1 + b2 + c2 + e2) \}$$

(from the 1st and 2nd experiments in Table 5.1)

where, e1 and e2 are errors associated with the 1st and 2nd experiments in Table 5.1.

$$= \tfrac{1}{2} (2\mu + 2a1) + \tfrac{1}{2} (b1 + b2) + \tfrac{1}{2} (c1 + c2) + \tfrac{1}{2} (e1 + e2)$$

$$= (\mu + a1) + \tfrac{1}{2} (e1 + e2) \qquad \text{(from the equations above)}$$

Ignoring the error part, the effect of A1 can be expressed as :

$$a1 = m(A1) - \mu$$

Similar procedures are followed to calculate the effect of other levels for all the variables. This model can only work efficiently if the additivity principle holds and there is no interaction between variables. For detailed discussion about different types of interactions refer to Phadke (1989) or Roy (1993). Often the additivity requirement limits the use of the orthogonal matrix for designing experiments. If there is an interaction between two variables, the resulting deviation from the mean μ will be falsely added to another variable, which can affect the conclusions considerably.

The neighbourhood of a design solution is defined using tolerances on design variables and Taguchi's orthogonal matrix. The tolerance space of any design solution is ideally defined as all worst combinations of design variables (considering the worst-case variability). Taguchi's orthogonal matrix is a fractional factorial strategy so that fewer experiments are required to perform an approximate calculation of the sensitivities. Thus the tolerance space is defined using the orthogonal matrix that is then used as the basis for the sensitivity calculations. Calculation of the factor effects following Taguchi's methodology provides the design variable sensitivity information.

5.4 Developing the Taguchi's Orthogonal Matrix for the Problem

Developing an orthogonal matrix for a problem requires some knowledge about the nature of the problem. There are some standard orthogonal matrices defined by Taguchi [Taguchi (1986)]. Often a standard orthogonal matrix can be modified to work with the real life problem. In order to select a standard orthogonal matrix and then modify it the following information about the problem is required:

a) *Number of Factors* (i.e. the number of design variables) to be studied .

For the turbine blade problem the sensitivity calculation is limited to one geometry only, thus

Total no. of variables = 11

b) *Levels associated with each design variable*:

L (LOW), M (MEDIUM), H (HIGH)

The levels are defined by defining the tolerance on each design variable as:

$L = M - \Delta$

$H = M + \Delta$

where, the variable can be expressed as $M^{\pm\Delta}$.

Experts use domain knowledge to determine the tolerances on the design variables.

c) Interaction between variables is not considered, because it is assumed that a small neighbourhood of a design solution can be approximated with an additive model.

d) *Ranking of the design variables* according to the ease of changing their levels is determined heuristically in the decreasing order as follows:

Rs, kw, Cdf, dth, Cdr, Fhc, Rp, df, Ff, Rpf and Tc1.

5.4.1 Degrees of Freedom Calculation

The first step in constructing an orthogonal matrix to fit the turbine blade problem is to count the total degrees of freedom. It tells the minimum number of experiments (in this case the model evaluations) that must be performed to study all the chosen factors. To begin with, one degree of freedom is associated with the overall mean regardless of the number of design variables to be studied. The number of degrees of freedom associated with a factor is equal to one less than the number of levels for that factor. This is because only two comparisons are required in case of a 3-level design variable. Thus the total degrees of freedom for the problem can be estimated as :

Factor (Design Variable)	*Degrees of Freedom*
Overall mean	1
All variables (11)	11 x (3 - 1) = 22

	Total: 23

That means, at least 23 experiments (model evaluations) are required to estimate the effect of each factor.

Exp. No.	**Column**												
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**
1	1	1	1	1	1	1	1	1	1	1	1	1	1
2	1	1	1	1	2	2	2	2	2	2	2	2	2
3	1	1	1	1	3	3	3	3	3	3	3	3	3
4	1	2	2	2	1	1	1	2	2	2	3	3	3
5	1	2	2	2	2	2	2	3	3	3	1	1	1
6	1	2	2	2	3	3	3	1	1	1	2	2	2
7	1	3	3	3	1	1	1	3	3	3	2	2	2
8	1	3	3	3	2	2	2	1	1	1	3	3	3
9	1	3	3	3	3	3	3	2	2	2	1	1	1
10	2	1	2	3	1	2	3	1	2	3	1	2	3
11	2	1	2	3	2	3	1	2	3	1	2	3	1
12	2	1	2	3	3	1	2	3	1	2	3	1	2
13	2	2	3	1	1	2	3	2	3	1	3	1	2
14	2	2	3	1	2	3	1	3	1	2	1	2	3
15	2	2	3	1	3	1	2	1	2	3	2	3	1
16	2	3	1	2	1	2	3	3	1	2	2	3	1
17	2	3	1	2	2	3	1	1	2	3	3	1	2
18	2	3	1	2	3	1	2	2	3	1	1	2	3
19	3	1	3	2	1	3	2	1	3	2	1	3	2
20	3	1	3	2	2	1	3	2	1	3	2	1	3
21	3	1	3	2	3	2	1	3	2	1	3	2	1
22	3	2	1	3	1	3	2	2	1	3	3	2	1
23	3	2	1	3	2	1	3	3	2	1	1	3	2
24	3	2	1	3	3	2	1	1	3	2	2	1	3
25	3	3	2	1	1	3	2	3	2	1	2	1	3
26	3	3	2	1	2	1	3	1	3	2	3	2	1
27	3	3	2	1	3	2	1	2	1	3	1	3	2

Table 5.2: $L_{27}(3^{13})$, a standard orthogonal matrix. The matrix consists of 13 variables with 3 levels each. The matrix suggests 27 experiments in total.

5.4.2 Selecting a Standard Orthogonal Matrix

Taguchi has tabulated 18 basic orthogonal matrices [Taguchi (1986)], which are called standard orthogonal matrices. The most common technique is to select one of these standard matrices and then modify it to suit the problem. The selected matrix should have the number of rows at least equal to the degrees of freedom required for the problem. The number of columns of a matrix represents the maximum number of factors that can be

Exp. No.	**Column**												
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**
1	G_L	E_L	I_L	D_L	A_L	B_L	F_L	H_L	J_L	K_L	C_L		
2	G_L	E_L	I_L	D_L	A_M	B_M	F_M	H_M	J_M	K_M	C_M		
3	G_L	E_L	I_L	D_L	A_H	B_H	F_H	H_H	J_H	K_H	C_H		
4	G_L	E_M	I_M	D_M	A_L	B_L	F_L	H_M	J_M	K_M	C_H		
5	G_L	E_M	I_M	D_M	A_M	B_M	F_M	H_H	J_H	K_H	C_L		
6	G_L	E_M	I_M	D_M	A_H	B_H	F_H	H_L	J_L	K_L	C_M		
7	G_L	E_H	I_H	D_H	A_L	B_L	F_L	H_H	J_H	K_H	C_M		
8	G_L	E_H	I_H	D_H	A_M	B_M	F_M	H_L	J_L	K_L	C_H		
9	G_L	E_H	I_H	D_H	A_H	B_H	F_H	H_M	J_M	K_M	C_L		
10	G_M	E_L	I_M	D_H	A_L	B_M	F_H	H_L	J_M	K_H	C_L		
11	G_M	E_L	I_M	D_H	A_M	B_H	F_L	H_M	J_H	K_L	C_M		
12	G_M	E_L	I_M	D_H	A_H	B_L	F_M	H_H	J_L	K_M	C_H		
13	G_M	E_M	I_H	D_L	A_L	B_M	F_H	H_M	J_H	K_L	C_H		
14	G_M	E_M	I_H	D_L	A_M	B_H	F_L	H_H	J_L	K_M	C_L		
15	G_M	E_M	I_H	D_L	A_H	B_L	F_M	H_L	J_M	K_H	C_M		
16	G_M	E_H	I_L	D_M	A_L	B_M	F_H	H_H	J_L	K_M	C_M		
17	G_M	E_H	I_L	D_M	A_M	B_H	F_L	H_L	J_M	K_H	C_H		
18	G_M	E_H	I_L	D_M	A_H	B_L	F_M	H_M	J_H	K_L	C_L		
19	G_H	E_L	I_H	D_M	A_L	B_H	F_M	H_L	J_H	K_M	C_L		
20	G_H	E_L	I_H	D_M	A_M	B_L	F_H	H_M	J_L	K_H	C_M		
21	G_H	E_L	I_H	D_M	A_H	B_M	F_L	H_H	J_M	K_L	C_H		
22	G_H	E_M	I_L	D_H	A_L	B_H	F_M	H_M	J_L	K_H	C_H		
23	G_H	E_M	I_L	D_H	A_M	B_L	F_H	H_H	J_M	K_L	C_L		
24	G_H	E_M	I_L	D_H	A_H	B_M	F_L	H_L	J_H	K_M	C_M		
25	G_H	E_H	I_M	D_L	A_L	B_H	F_M	H_H	J_M	K_L	C_M		
26	G_H	E_H	I_M	D_L	A_M	B_L	F_H	H_L	J_H	K_M	C_H		
27	G_H	E_H	I_M	D_L	A_H	B_M	F_L	H_M	J_L	K_H	C_L		

Table 5.3: The orthogonal matrix used for the sensitivity analysis, where A = Cdr, B = Fhc, C = Tc1, D = dth, E = kw, F = Rp, G = Rs, H = df, I = Cdf, J = Ff, K = Rpf. The three subscripts L, M, and H mean Low, Medium and High levels.

studied using that matrix. In addition, in order to use a standard orthogonal matrix directly, one must be able to match the number of levels of the factors with the numbers of levels in the columns of the matrix. The smallest possible matrix is selected to save the number of model evaluations.

Considering the minimum number of experiments required, the number of variables and the number of levels per variable, a standard matrix $L_{27}(3^{13})$ (Table 5.2) has been selected for the problem. The matrix defines 27 experiments, it has 13 columns and the factors (that is the design variable) have three levels each. As there are only 11 design variables in the problem, the 12th and the 13th column of the matrix are left empty (Table 5.3). This does not destroy the orthogonality of the matrix. The design variables are placed in the columns according to the ranking of the design variables considering the ease of changing their levels.

5.5 Use of Taguchi's Orthogonal Matrix

***Definition 5.1*: Tolerance Space**

The Tolerance Space (TS) around a design solution can be defined as a set of points where each point represents a possible combination of the design variables with the tolerances associated with them. The points are selected using the Taguchi's orthogonal matrix (OM). Each design variable of a design solution can have an upper and a lower value defined by its tolerance. Thus the three levels of each variable can be represented as g (the variable value), gu (the upper level, that is g + tol.) and gl (the lower level, that is g - tol.). In Figures 5.1-5.5 dashed rectangles represent the tolerance space in 2 dimensions (as in case of full factorial, that is all possible combinations). The design solution lies at the centre of the dashed rectangle and is marked by a larger circle. It is assumed that each design solution is expressed as:

$$d_g = [\ x_{1g}\ ,\ x_{2g}\ ,\ x_{3g},\ \ldots\ldots\ldots,\ x_{mg}\ ]\ \in DS \qquad \ldots.(5.1)$$

where:

x : design variable

d : design solution as vector of m design variables

DS : the design space

m : no. of design variables

and the tolerance space associated with it can be expressed as:

$$TS(d_g) = \{\ d : d\ \in\ DS\ |\ |\ d - d_g\ |\ \leq\ \Delta\ d\ \}^{OM}$$

where:

tolerances on each variable = $\Delta\ d = [\ \Delta\ x_1,\ \Delta\ x_2, \ldots\ \Delta\ x_m\ ]^{TS}$

....(5.2)

Definition 5.2 **: Vertex Space**

Vertex Space (VS) consists of all possible design solutions or options (all worst case combinations) of the tolerance space (TS) except the design solution (d_g). Thus VS can be formally represented as:

$$VS(d_g) = TS - \{\ d_g\ \} \qquad \ldots.(5.3)$$

Definition 5.3 **: Design Solution Sensitivity**

Once the tolerance space (the neighbourhood) is defined, in order to measure sensitivity of the design solution a *Sensitivity Index* (SI) is defined as follows:

$$SI = \frac{1}{\eta} \qquad \text{where, } \eta = \text{signal to noise ratio [Phadke (1989)]} \qquad \ldots.(5.4)$$

Considering the task of the optimisation is to reduce coolant mass flow rate, the problem can be considered as a "Nominal-the-Best" type [Phadke (1989)]. Thus the Signal to Noise ratio is defined as:

$$\eta = 10 \log_{10} \frac{\mu^2}{\sigma^2} \qquad ....(5.5)$$

where:

$$\mu = \frac{1}{n}\sum_{i=1}^{n} y_i$$

and $$\sigma^2 = \frac{1}{n-1}\sum_{i=1}^{n}(y_i - \mu)^2$$

$y_i = y_1, y_2, ..., y_n$

= coolant flow for different designs.

n = no. of designs considered by the orthogonal matrix

***Definition 5.4* : Design Variable Sensitivity**

Taguchi's methodology is followed to calculate the effects of the different levels (on the coolant mass flow) for each design variable in the tolerance space. Assuming that the additivity principle is valid in the tolerance space of the design solution, summation of the three factor level effects for each design variable gives the error. The error calculated for an individual variable is subtracted from its level effects and then the absolute values are considered. The maximum of these three new effects defines the design variable sensitivity. For example, considering the orthogonal matrix shown in Table 5.1, sensitivity of the variable C can be mathematically expressed as:

Sensitivity of the variable C = max {|c1-E|, |c2-E|, |c3-E|} ....(5.6)

where, E = error = (c1 + c2 + c3)/3.0

Similarly effects of the other variables are calculated. Critical design variables can be identified by ranking them according to this design variable sensitivity.

Assumption 5.1 :

Constraints are assumed to be monotonic with respect to all design variables in the tolerance space. That means the maximum constraint value will occur at one of the corner points [Sundaresan et. al. (1993)].

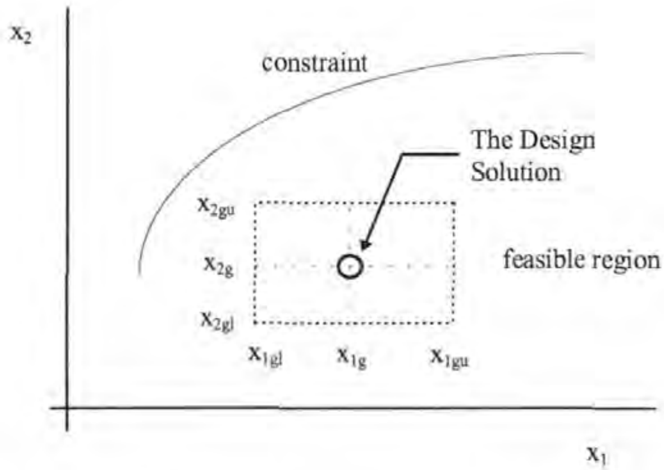


Figure 5.1: Constraint Satisfied.

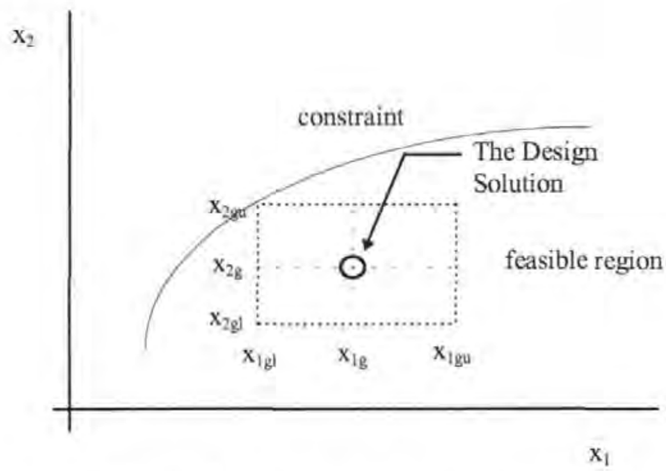


Figure 5.2: Statistically Active Constraint.

Definition 5.5 **: Constraint Satisfied**

An *i*th inequality constraint (C_i) is considered to be satisfied when the value of the constraint is negative at all worst case combinations (VS) as well as at the design solution (Figure 5.1).

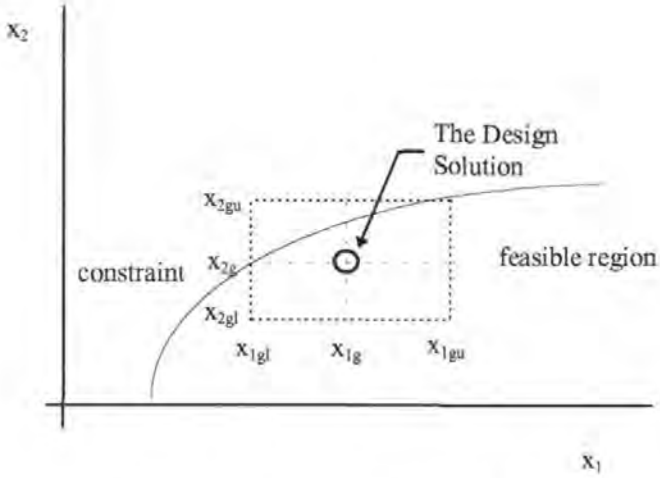


Figure 5.3: Quasi - Active Constraint.

For $d_g \in$ DS, C_i constraint is satisfied if

1) $C_i(d_g) < 0.0$

2) $\forall d \in TS(d_g)\, s.t. C_i(d) < 0.0$ ....(5.7)

Definition 5.6 **: Statistically Active Constraint**

An *i*th inequality constraint (C_i) is considered to be statistically active when the value of the constraint is zero (that is the point is on the constraint boundary) at least at one worst combination (VS) of design variables and negative at the design solution as well as at the remaining worst combinations (VS) of design variables (Figure 5.2).

So, for $d_g \in$ DS, C_i constraint is statistically active if

1) $\exists d \in VS(d_g) s.t. C_i(d) = 0.0$

2) $C_i(d_g) < 0.0$

3) $\forall d^{'} \in VS(d_g) - \{d\} s.t. C_i(d^{'}) < 0.0$ ....(5.8)

Definition 5.7 **: Quasi - Active Constraint**

An *i*th inequality constraint (C_i) is considered to be quasi - active when the value of the constraint is positive (that is the constraint is violated) at least at one worst combination

(VS) of design variables and negative at the design solution. At the remaining worst combinations (VS) of design variables, value of constraint can be either zero or negative (Figure 5.3).

So, for $d_g \in DS$, C_i constraint is quasi - active if

1) $\exists d \in VS(d_g) s.t. C_i(d) > 0.0$

2) $C_i(d_g) < 0.0$

3) $\forall d' \in VS(d_g) - \{d\} s.t. C_i(d') \leq 0.0$ ....(5.9)

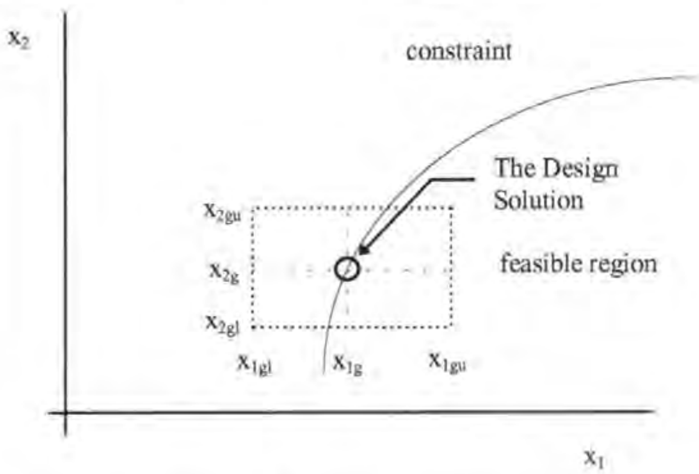


Figure 5.4: Peak - Active Constraint.

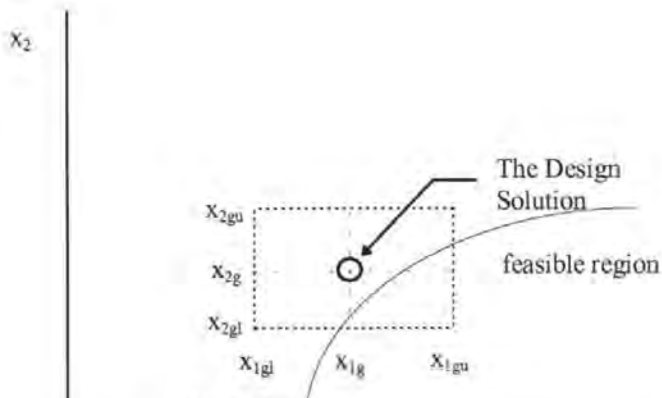


Figure 5.5: Constraint Not Satisfied.

Definition 5.8 **: Peak - Active Constraint**

An *i*th inequality constraint (C_i) is considered to be peak - active when value of the constraint is zero at the design solution and positive at least at one worst case combination (VS) of design variables (Figure 5.4).

So, for $d_g \in DS$, C_i constraint is peak - active if

1) $C_i(d_g) = 0.0$

2) $\exists d \in VS(d_g) s.t. C_i(d) > 0.0$ ....(5.10)

Definition 5.9**: Constraint Not Satisfied**

An *i*th inequality constraint (C_i) is considered to be not satisfied when value of the constraint is positive at the design solution. The neighbourhood is not checked for this case (Figure 5.5).

So, for $d_g \in DS$, C_i constraint is not satisfied if

1) $C_i(d_g) > 0.0$ ....(5.11)

5.6 Applying the Technique to the Turbine Blade Problem

The sensitivity analysis technique based on Taguchi's methodology has been successfully applied to the real life turbine blade cooling system design problem. The sensitivity analysis works as an integral part of ASM. The methodology is based on the assumption that there is very little or no interaction among the variables in the tolerance space (TS) of the design solutions. The turbine blade cooling system model development assumed no interaction between the design variables, so the technique is expected to be effective for the application. The sensitivity analysis only considers the tolerance space around a design solution, so it is more probable that the small region (the neighbourhood) can be approximated using an additive model. A priori knowledge about the interactions is not available in the majority of the real life problems. An attempt has been made to check for the interactions in the tolerance spaces of the design solutions. One way of checking for the interactions is to validate the additivity principle. Validity of the additivity principle in a tolerance space increases the confidence on the sensitivity calculations for that region.

The basis of the sensitivity calculations, the orthogonal matrix, allows the examination of only a small number of design solutions (in this case 27 model evaluations only) rather than all 3^{11} possible evaluations (that is in case of 11 design variables with 3 levels each) within the neighbourhood of a design solution for the sensitivity calculation. Though in this case the technique is very reliable, the sensitivity analysis results should only be used to compare two design solutions rather than to define their absolute sensitivity values. The use of the signal-to-noise ratio to calculate the design solution sensitivity index is a measure of the robustness of the solution within its neighbourhood. The information concerning individual design variable sensitivity is very useful for engineering design decision support, because it determines the criticality of the different variables in the tolerance space of the design solution. A designer often selects design solutions that satisfy constraints, but that may not be enough, as the criticality of constraints in the neighbourhood also plays a major role in engineering decision making.

5.6.1 Checking for the Additivity Principle

The additivity principle defines the output of a design solution as the summation of the different variable level effects and the mean output of its neighbourhood. It is assumed that, a variable A has three levels 1, 2 and 3; the effect of the level 1 in the tolerance space of A is a'1 and can be defined as:

$$a'1 = a1 - ER \quad \text{....(5.12)}$$

where, a1 = factor effect of level 1 of the variable A, calculated as described in section 5.5.

ER = error = (a1 + a2 + a3)/3.0

(this is because if the additivity principle is satisfied a1 + a2 + a3 should be equal to zero)

Once the effects of individual variable levels are calculated, the most contributing level for each variable is defined as the level having the largest absolute effect. An experiment or a design solution is defined consisting of all the most contributing design variable levels. The output of this experiment is predicted by adding the variable level effects with the mean output in the tolerance space. A validation experiment is conducted by calling the turbine blade cooling system model with the set of design variable level values as the input. If the difference between the predicted output and the validation experiment output is within an acceptable range (that is the difference is less than or equal to 5%), the additivity model is considered to be a good approximation of the reality in the tolerance space, that is the additivity principle is valid in the region. Every "good" design solution is first tested for the

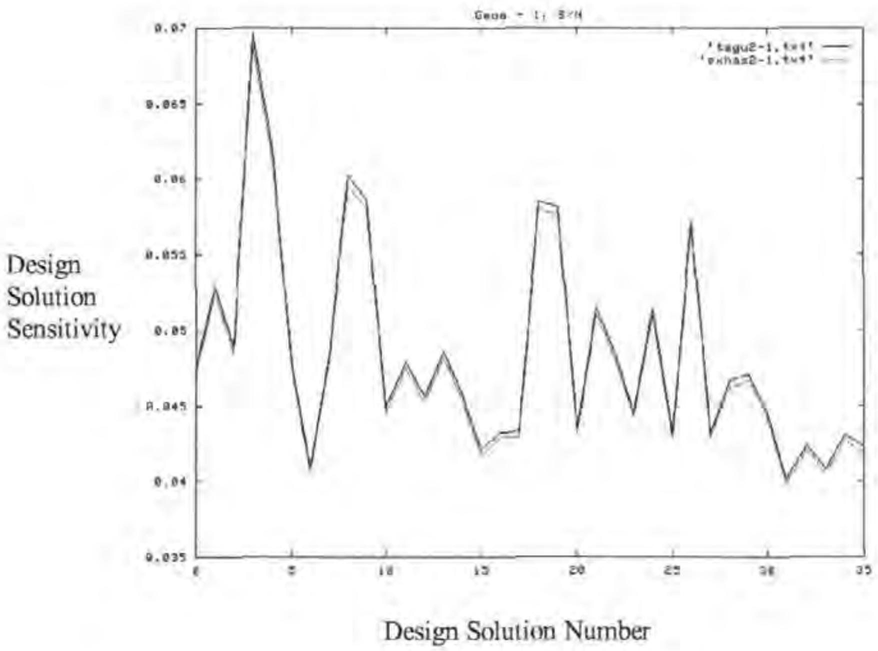


Figure 5.6: The comparison between Taguchi's methodology based sensitivity analysis and the sensitivity analysis based on an exhaustive search for Geometry 1.

additivity principle. If the additivity principle is valid in its tolerance space the sensitivity calculation results are considered as reliable, otherwise, one option is to perform an exhaustive search in the area, and then calculate the sensitivities.

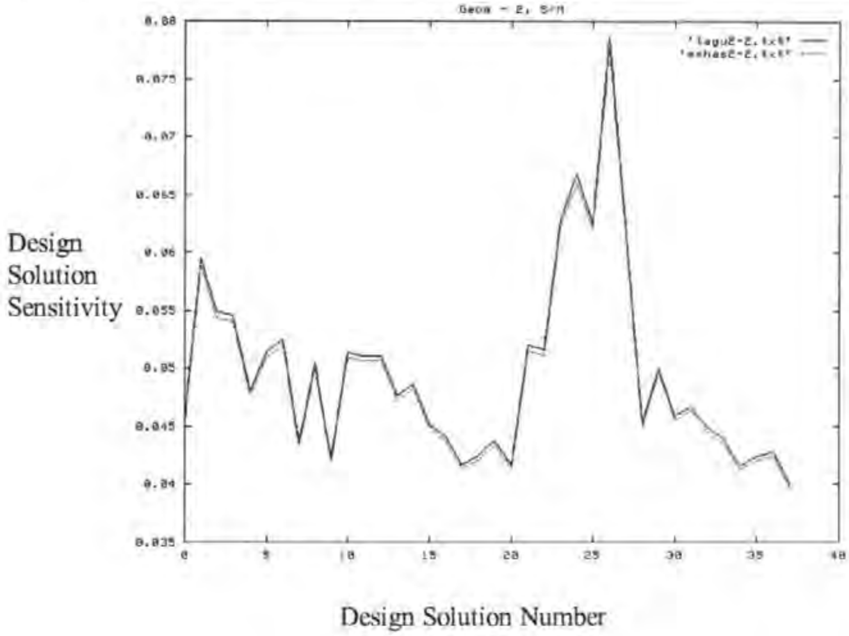


Figure 5.7: The comparison between Taguchi's methodology based sensitivity analysis and the sensitivity analysis based on an exhaustive search for Geometry 2.

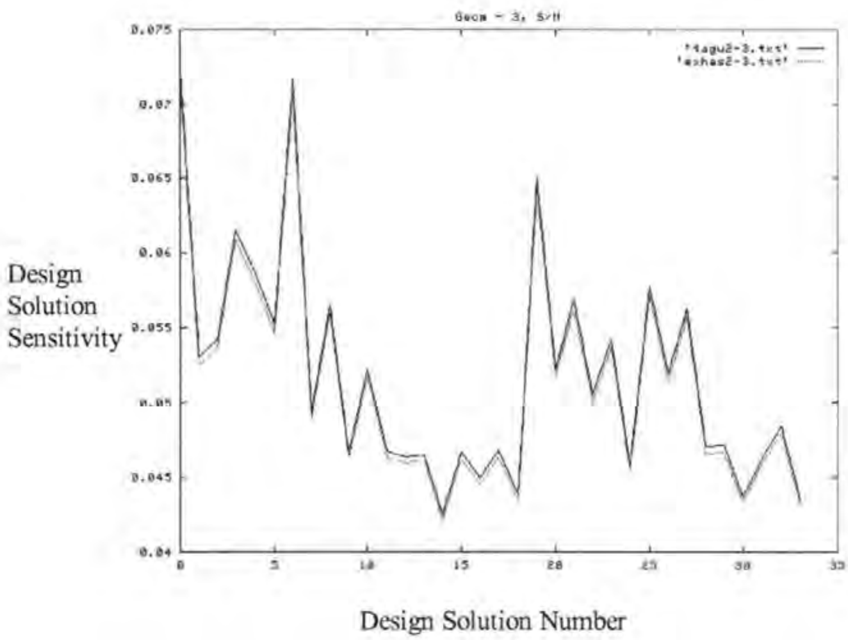


Figure 5.8: The comparison between Taguchi's methodology based sensitivity analysis and the sensitivity analysis based on an exhaustive search for Geometry 3.

5.6.2 Comparing Taguchi's Methodology and an Exhaustive Search Based Sensitivity Analysis

The sensitivity calculation results are considered to be reliable if there is no or very little interaction among design variables in the tolerance space of a design solution. In order to validate this notion, Taguchi's methodology based sensitivity calculation result is compared with the sensitivity analysis using an exhaustive search. More than 100 design solutions whose tolerance spaces satisfy the additivity principle are used for the comparison. The design solution sensitivities calculated from the two methods are plotted separately for each geometry (Figures 5.6-5.8). The Figures 5.6-5.8 show a very high level of correlation between the two sets of design solution sensitivities. Thus the figures confirm the hypothesis that Taguchi's methodology is very effective (that is comparable to an exhaustive search based method) if there is no interaction or very little interaction among design variables. The figures also show within one geometry some design solutions can have very high design solution sensitivities with respect to the others.

5.7 Validation of the sensitivity analysis

The sensitivity analysis is validated by the expert and the user from Rolls Royce. Results from several individual analysis are presented to the expert and the user. Every 'good' solution identified by the hybrid search method is tested for the additivity principle before the analysis is performed. It is observed that the study described in the previous section increased confidence of the expert and the user on the results of the analysis. They also check whether the design variable sensitivities correspond to their general understanding about the problem. If a result does not correspond to their understanding, the solution is further analysed, and that may improve the general understanding of the problem.

5.8 Summary

The research presented in the chapter describes a method of obtaining sensitivity information concerning a design solution. The sensitivity is calculated within the neighbourhood of the design. Three types of sensitivity information are defined: design solution sensitivity, design variable sensitivity and constraint sensitivity. Taguchi's methodology is introduced to perform the sensitivity analysis with a small number of model evaluations. Results from the analysis are useful provided there is no serious interaction between the design variables within the neighbourhood of the design. The sensitivity information facilitates comparison of two designs, and thus helps in design selection. The next chapter discusses a method of measuring (qualitatively) the effectivenesses of a design with respect to three different qualitative criteria.

CHAPTER - 6

6. Qualitative Evaluation of Engineering Designs

6.1 Introduction

This chapter discusses a method of qualitative evaluation of the designs as obtained from the hybrid search described in chapter 4. The design solutions are qualitatively evaluated using a fuzzy expert system to find out the qualitative ratings of a design in terms of manufacturability, choice of materials and designers' special preferences. The fuzzy expert system is implemented using a fuzzy logic version of CLIPS (developed by NASA) called FuzzyCLIPS [FuzzyCLIPS User's Guide (1994)]. Some aspects of manufacturability, choice of material and special preferences (of the customer or the designer) can be qualitative in nature. Qualitative ratings that represent these criteria are calculated using fuzzy logic. The qualitative evaluation system is integrated within the 'Adaptive Search Manager' (ASM). The tasks of ASM are to identify different "good" design solutions (chapter 4), perform the sensitivity analysis (chapter 5), and qualitatively evaluate the designs. This method of qualitative evaluation realised through fuzzy logic involves fuzzy modelling or linguistic modelling [Sugeno and Yasukawa (1993)]. In order to develop the integrated system it is necessary to ensure that the qualitative evaluation system can evaluate any design from the search space. A novel knowledge representation technique [Roy et. al. (1995a) and (1996c)] is developed where the domain knowledge is first defined in terms of inter-variable preferences, intra-variable preferences and heuristics. The inter-variable preferences are combined with the intra-variable preferences using a concept of compromise. The concept of compromise is defined as "reducing the severity of the negative effect of one variable on the final qualitative rating". This method of knowledge separation and then integration has helped to cover the entire design space utilising a small number of rules.

This chapter briefly introduces fuzzy logic and fuzzy expert systems before discussing the development of the qualitative evaluation system. The novel knowledge representation technique is discussed in detail. The chapter concludes with the validation of the system.

6.2 Fuzzy Logic

Traditional set theory requires the arbitrary placement of some sort of threshold, at which an object abruptly changes from belonging to one set to its complement. In early 60's, Lotfi Zadeh published a paper outlining a 'fuzzy set theory' [Zadeh (1965)]. He proposed graded memberships in sets, which is to say that an element could be, say, 20% element of set A and 80% element of A´ (i.e. complement of A). For example, if *ambient temperature* is 25 °C, for many people it is *medium* temperature whereas others may consider that as *hot*. Thus, the temperature can be represented as having *memberships* of 60% in the *medium* set and 40% in the *hot* set. The terms, *medium* and *hot*, are referred to as *fuzzy terms*. The logic tool for representing and manipulating fuzzy terms is called *fuzzy logic*. At that time the concept was very radical, and even many did not accept the idea at all. The idea of fuzzy logic showed first indications of success only after almost two decades of research. Now there are many applications of fuzzy logic in engineering problems [Mendel (1995)]. Expert systems [Durkin (1994)] have been the most obvious recipients of the benefits of fuzzy logic, since their application domain is often inherently fuzzy. Expert systems that utilise fuzzy logic concepts are termed *fuzzy expert systems*. The research presented in this thesis uses a fuzzy expert system to develop the qualitative evaluation system.

It is seen that engineering designers often evaluate a design heuristically utilising their past knowledge. On many occasions they do not think in terms of precise values of the design attributes. The knowledge they use in many cases represents their intuition and feeling [Oksala (1994)]. Thus the designers very often use somewhat "fuzzy" concepts about the design task to evaluate a design; and often when they are asked to express their knowledge they use several "fuzzy" terms to define different design attributes. A design attribute or a design variable that is expressed through some "fuzzy" terms is known as *linguistic variable* (e.g. ambient temperature), whereas the "fuzzy" terms are called the variable's *linguistic values* (e.g. medium and hot). The range of possible values of a linguistic variable is called the variable's *universe of discourse*. A production rule [Durkin (1994)] is termed *fuzzy rule* if it uses linguistic variables and fuzzy terms. It is worth noting that the 'fuzziness' introduced above does not necessarily arise from errors or uncertainties. Even if the experts are all equally reputable, it does not follow that they will always agree. This fuzziness is an important aspect of the problem that cannot be modelled using ordinary sets, but they can be modelled using *fuzzy sets*. The description of a fuzzy set is presented in the next section.

6.2.1 Fuzzy Sets and the Representation

A fuzzy set assigns membership values between 0 and 1 that reflect more naturally a member's association with the set. The mapping between elements of the fuzzy set and values in the interval [0, 1] defines a membership function for the fuzzy set. To represent a fuzzy set for a problem, it is required to define the membership function. The membership function actually represents peoples' intuition and opinion concerning the fuzzy set. Multiple opinions, which often can be contradictory, can be accommodated by taking an average of the opinions, and that can be represented in the fuzzy set. For example, Figure

6.1 describes the fuzzy variable 'height' by three fuzzy sets called *short*, *medium* and *tall*. When multiple fuzzy sets are defined on the same universe of discourse, the fuzzy literature

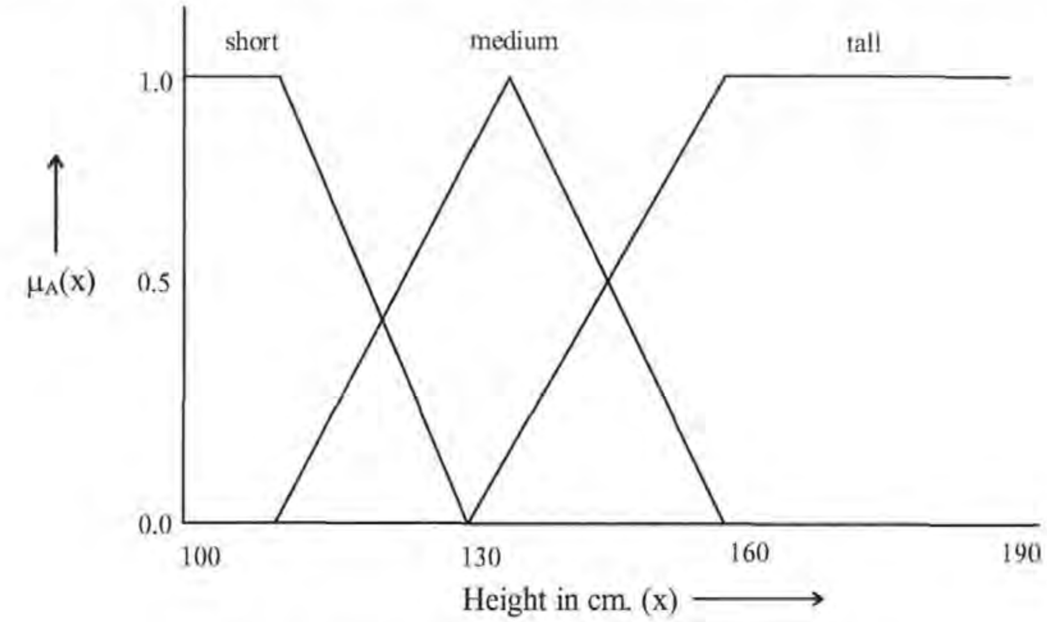


Figure 6.1: Fuzzy sets used for the fuzzy variable "height".

often refers to them as *fuzzy subsets*. This thesis uses only 'fuzzy set' in order to represent both single and multiple fuzzy sets, and to reduce confusion. Figure 6.1 shows a person of a height of 115 cm. is a member of *short* persons' group with a membership value of 0.7, and at the same time a member of *medium* persons' group with a value of 0.2. Thus, a single object is considered a *partial* member of multiple sets. This property can model ambiguities in human thinking, and thus utilised in fuzzy expert systems. A formal representation of a fuzzy set [Durkin (1994)] can be described as follows:

> *Let U be the universe of discourse and A is a fuzzy set defined on it. Further assume there is a discrete set of U elements $\{x_1, x_2, \ldots\ldots, x_n\}$. The fuzzy set A defines a membership function $\mu_A(x)$ that maps the elements x_i of U to the degree of memberships in [0,1]. For a discrete set of elements, a convenient way of representing a fuzzy set is through the use of a set of ordered pairs:*
>
> $$A = \{(x_1, a_1), (x_2, a_2), \ldots\ldots\ldots, (x_n, a_n)\}$$
>
> *where,* $a_i = \mu_A(x_i)$

During conversation, humans often add more vagueness to a statement by using adverbs such as *very*, *slightly*, or *somewhat*. For example, it could be said: 'Harry is *very* tall'. Here, *very* is the adverb used with the fuzzy term *tall*. This extra adverb is called a *modifier* or a *hedge*. A hedge mathematically modifies an existing fuzzy set account for some added adverb. For example, if the membership function values for the fuzzy term *tall* is represented by y, then the values for *very tall* can be expressed as y^2.

6.3 Fuzzy Expert Systems

In simple terms, a fuzzy expert system is an expert system that uses fuzzy logic instead of Boolean logic. In other words, a fuzzy expert system is a collection of membership functions and fuzzy rules that are used to reason about data. Unlike conventional expert systems, which are mainly symbolic reasoning engines, fuzzy expert systems are oriented towards numerical processing.

The rules in a fuzzy expert system are usually of a form similar to the following:

> **if** *x is high and y is low*
>
> **then** *z is very high*

where x and y are input variables (names for known data values), and z is an output variable (a name for a data value to be computed). The linguistic terms used with the variables are *high*, *low* and *very high*. The adverb *very* is a hedge and is used to modify the *high* fuzzy set of the variable z. Most tools for working with fuzzy expert systems allow more than one conclusion per rule. The research described in the thesis uses FuzzyCLIPS (a fuzzy expert system development tool). Thus, from now onwards discussions and examples are presented in FuzzyCLIPS terminology. FuzzyCLIPS can handle inexact concepts, fuzziness and uncertainty. Uncertainty occurs when one is not totally certain about a piece of information.

This uncertainty is generally expressed as a degree of certainty using a *certainty factor*. Uncertainties associated with a fact and any fuzzy rule can be represented in FuzzyCLIPS.

6.3.1 Fuzzy Inference

The fuzzy inference technique is used to evaluate a set of relevant fuzzy rules given some information. The already existing information is stored as fuzzy sets in a facts list. The rule evaluation depends on a number of factors, such as whether or not fuzzy variables are found in the antecedent or consequent part of a rule, whether a rule contains multiple antecedents or consequents, and whether a fuzzy fact being asserted has the same fuzzy variable as an already known fuzzy fact. If a fuzzy rule has only one antecedent then the rule is termed 'simple'. Whereas, if a rule contains more than one antecedent it is called a 'complex' rule. Fuzzy rules are stored as fuzzy associations [Durkin (1994)]. *Fuzzy inference* attempts to establish a degree of belief in a rule's consequent given available evidence on the rule's antecedent. The task is to map the antecedent fuzzy set information to the consequent set information. The inference technique establishes a modified fuzzy set from information about a related fuzzy set. The methodology to establish the relation categorises the inference technique mostly into: max-min inference and max-product inference. The research described in this thesis uses the max-min type inferencing. If the antecedent part of a fuzzy rule matches or partially matches already existing information, the rule is fired and the conclusion (a fuzzy fact) is asserted in the facts list. In case an asserted fuzzy fact has the same fuzzy variable as an already present fuzzy fact in the fact list, the asserted fact is modified according to a principle of *global contribution* [Durkin (1994)]. The global contribution becomes very useful when a fuzzy expert system works with many rules deciding only about one fuzzy variable. Each fuzzy rule can assert a fuzzy fact about the variable and finally all of them unite together to give a final conclusion.

6.3.2 The Max-Min Fuzzy Inference Technique

The max-min inferencing technique can be easily described in case of a simple fuzzy rule. The general structure of a simple fuzzy rule can be shown as follows:

If A Then C CF_r

A' CF_f

C' CF_c

where:

A: the antecedent of the rule

A': the matching fact in the fact list

C: the consequent of the rule

C': the actual consequent calculated

CF_r: the certainty factor of the rule

CF_f: the certainty factor of the fact

CF_c: the certainty factor of the conclusion

If A and C are two fuzzy sets, then A' must be a fuzzy fact with the same fuzzy variable as specified in A in order for a match to occur and the rule to be placed on the agenda. Rules are sequentially executed or fired from the agenda. In addition, while values of the fuzzy variables A and A' represented by their respective fuzzy sets (say F_α and F'_α) do not have to be equal, they must overlap. In case the fuzzy fact and the antecedent of the rule match, it is shown in Zadeh (1973) that the antecedent and the consequent of such a rule are connected by the fuzzy relation:

$$R = F_\alpha * F_c \qquad \text{....(6.1)}$$

where:

F_α: a fuzzy set denoting the value of the fuzzy antecedent pattern

F_c: a fuzzy set denoting the value of the fuzzy consequent

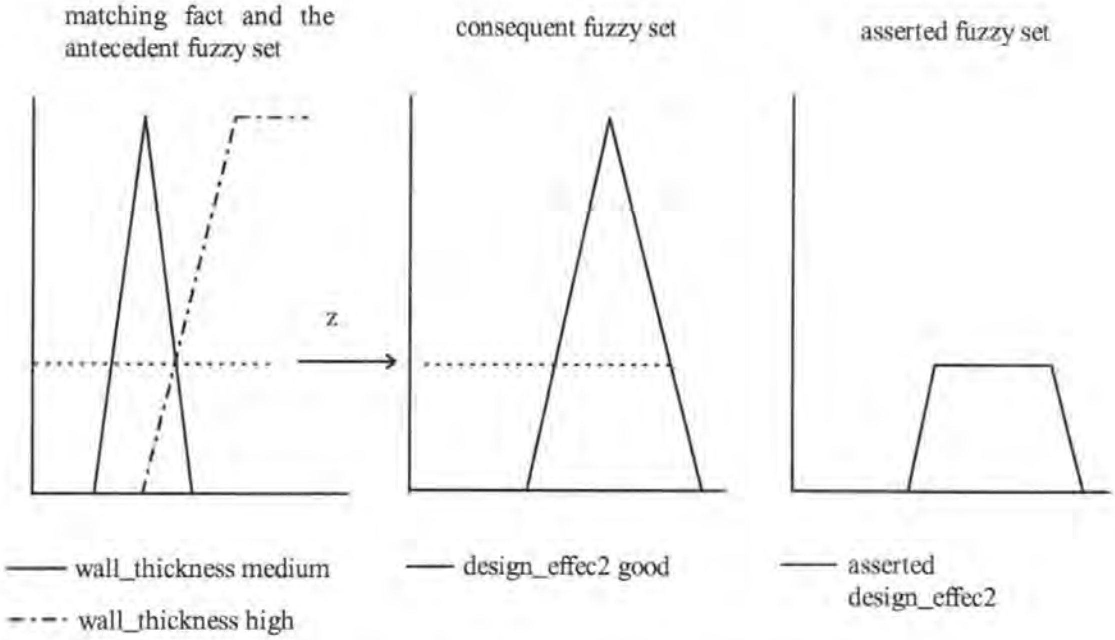


Figure 6.2: The Max-Min compositional rule of fuzzy inference.

The research described in this thesis calculates the membership function of the relation R as follows:

$$\mu_R(u,v) = \min(\mu_{F_\alpha}(u), \mu_{F_c}(v)), \qquad \forall (u,v) \in U \times V \qquad ....(6.2)$$

The calculation of the conclusion depends upon the compositional rule of inference [Zadeh (1973)], which is described as follows:

$$F_c' = F_\alpha' \circ R \qquad ....(6.3)$$

where, F_c' is a fuzzy set denoting the value of the fuzzy object of the consequent. The membership function of F_c' is calculated as follows:

$$\mu_{F_c'}(v) = \max_{u \in U}(\min(\mu_{F_\alpha}(u), \mu_R(u,v))) \qquad ....(6.4)$$

which can be simplified to:

$$\mu_{F_c'}(v) = \min(z, \mu_{F_c}(v)) \qquad ....(6.5)$$

$$\text{where, } z = \max(\min(\mu_{F_\alpha'}(u), \mu_{F_\alpha}(u)))$$

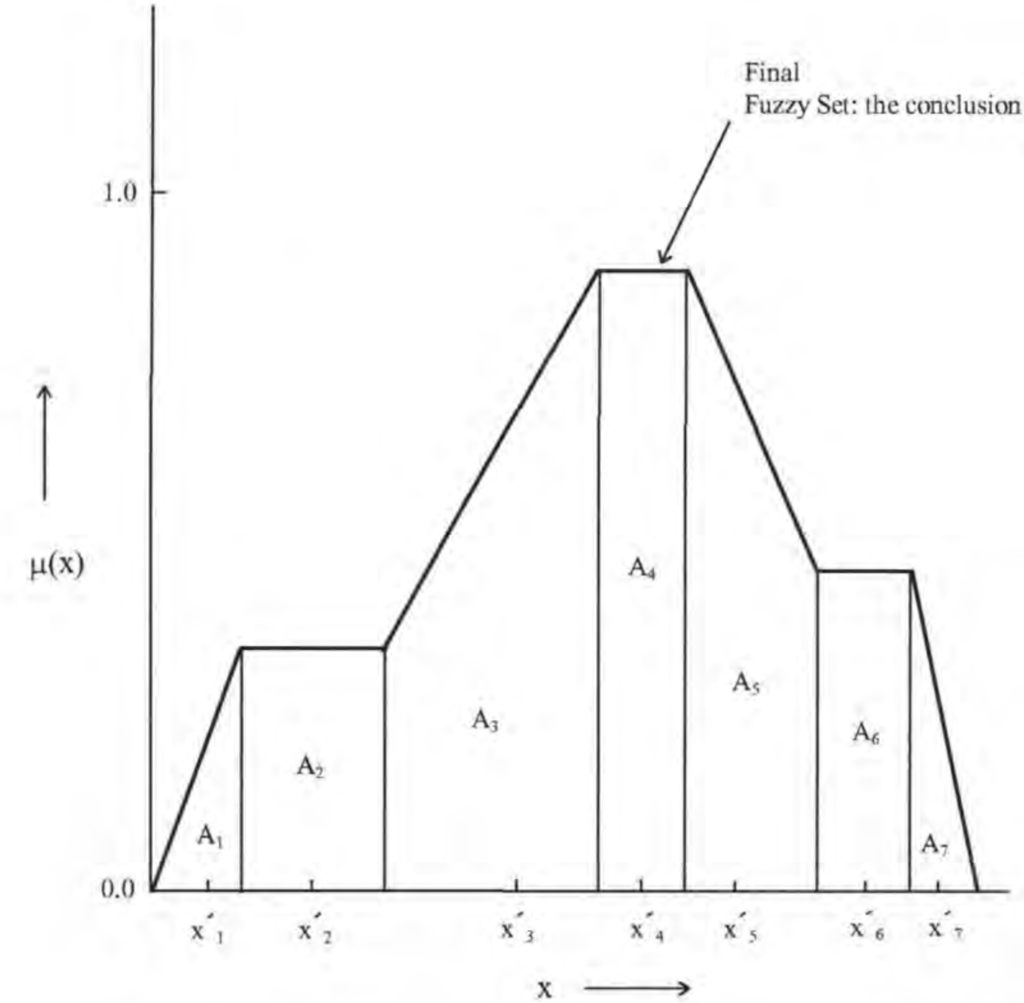
The certainty factor of the conclusion is a product of the certainty factor of the rule, the matching fact from the fact list, and the asserted fact. The certainty factor can be represented as: $CF_c = CF_r * CF_f * CF_{af}$, where CF_{af} is the certainty factor associated with the asserted fact (that is in the consequent). The max-min inferencing technique is presented in the Figure 6.2, and the figure shows how simply the conclusion is clipped off or truncated at the z value. The research reported in this thesis also uses complex rules with multiple antecedents. In such case, conclusions due to each antecedent and the corresponding fact are united to form the conclusion for the entire rule. The confidence factor for the conclusion is defined as the product of the rule confidence factor and the minimum confidence factor among the antecedents.

6.3.3 Defuzzification

The result of a fuzzy inference process is a fuzzy fact, specifying a fuzzy distribution of a conclusion. However, in some applications such as in the research elaborated in this thesis, only a single crisp value is required as the conclusion. So a single point that represents the fuzzy distribution needs to be selected. The process of representing a fuzzy distribution or fuzzy set by a crisp value or a single point is known as *defuzzification*. The most popular method of defuzzification is known as the *centre of gravity* method. The method takes the centre of gravity of the whole set as the representative single point of the set. The centre of gravity method has the advantage of smoothly varying output. Another method is known as *mean of maxima*, which concentrates on the values where the possibility distribution reaches a maximum. In a real life application this method may produce less smooth output, but the method is quicker due to fewer floating point calculations. The research presented in this thesis uses the centre of gravity type defuzzification technique, and that can be formally expressed as:

$$x' = \frac{\int_{(x \in U)} (x.f(x)).dx}{\int_{(x \in U)} f(x).dx} \qquad ....(6.6)$$

where, x´ is the recommended, crisp value, and U is the universe of discourse.



For each shaded subsection above, the area and centre of gravity is calculated according to the shape identified (i.e. triangular, rectangular or trapezoidal). The centre of gravity of the whole set is then determined as follows:

$$x' = \frac{A_1.x_1' + A_2.x_2' + A_3.x_3' + A_4.x_4' + A_5.x_5' + A_6.x_6' + A_7.x_7'}{A_1 + A_2 + A_3 + A_4 + A_5 + A_6 + A_7}$$

Figure 6.3: The "centre of gravity" type defuzzification process.

The qualitative evaluation system developed in this thesis using FuzzyCLIPS defines a fuzzy set by a set of points which are considered to be connected by straight line segments. Thus the integral becomes a simple summation:

$$x' = \frac{\sum_{i=1}^{n} x_i'.A_i}{\sum_{i=1}^{n} A_i} \qquad \text{....(6.7)}$$

where, the whole distribution is divided into n number of parts, each having A_i amount of area and the centre of gravity at x'_i.

Figure 6.3 illustrates the principle of the centre of gravity type defuzzification method. It is shown in future sections that the method is further used in implementing a concept of compromise for the turbine blade design problem.

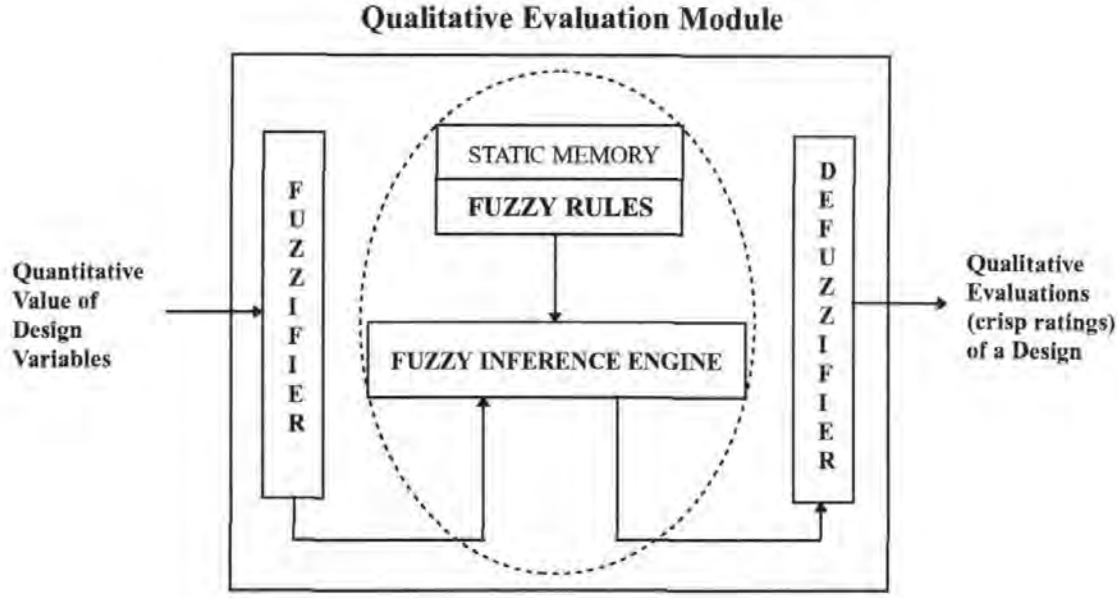


Figure 6.4: The Qualitative Evaluation System (QES).

6.4 The Qualitative Evaluation System

The qualitative evaluation system (QES) (Figure 6.4) is a *fuzzy* expert system developed using FuzzyCLIPS. QES evaluates the turbine blade cooling system design solutions with respect to three different qualitative criteria. The criteria are manufacturability, cost of material and designers' special preferences. QES takes variable values of each design solution as inputs and outputs three qualitative ratings for the design, that is an individual rating for each criterion. The fuzzy expert system has three components, the fuzzifier, the fuzzy inference engine and the defuzzifier. The knowledge is stored in a fuzzy rulebase, which is used by QES. Three sets of rules are used for the three criteria. QES considers the rulebase as the static memory. The system transforms crisp values of each design variable into a fuzzy representation. These representations are then processed using the fuzzy rules (domain specific) and a fuzzy inference engine (domain independent) to determine the qualitative ratings of the design solution. Initially the qualitative ratings are expressed using a fuzzy set. A crisp value for the rating is obtained through the centre-of-gravity type defuzzification. The tasks involved in the system development are described below.

6.4.1 Fuzzification of Design Variables

The fuzzification of the design variables is performed in two stages. During the first stage of the fuzzification each variable range is divided into five sub ranges, and expressed using some linguistic terms (also known as primary terms). The variable range is the universe of discourse for the variable. The linguistic terms used for the design variables are determined through an interview and discussions with the representatives from Rolls Royce. The fuzzy

```
(deftemplate kw   ;definition of fuzzy variable 'kw'
        18 33 wK/m^3
        (
                (VERY_LOW (18 1) (20 1) (22 0))
                (LOW (20 0) (22 1) (24 0))
                (MEDIUM (23 0) (26 1) (28 0))
                (HIGH (27 0) (29 1) (31 0))
                (VERY_HIGH (30 0) (33 1))
        )
)
```

Figure 6.5(a): The Deftemplate Construct for Thermal Conductivity (kw).

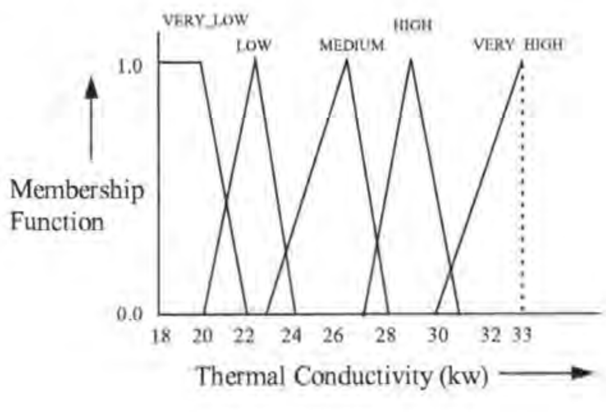


Figure 6.5(b): Fuzzy Deftemplate for Thermal Conductivity (kw).

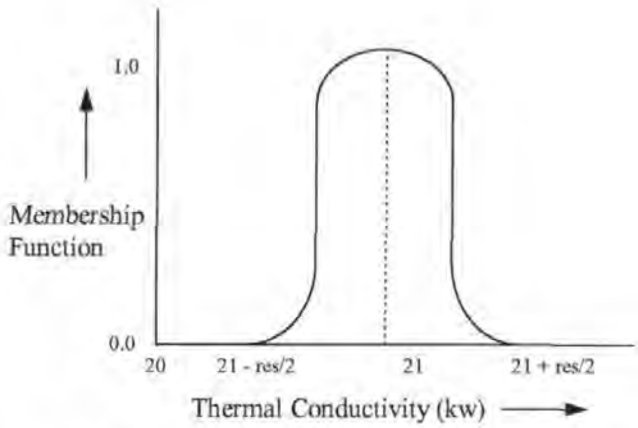


Figure 6.6: Fuzzification of Thermal Conductivity (kw).

representation of the design variables and the qualitative rating (or the effectiveness) are then expressed in FuzzyCLIPS syntax called *Deftemplate Construct* [FuzzyCLIPS User Guide (1994)]. The Deftemplate representation for the thermal conductivity (kw) is shown in Figure 6.5. Several linguistic expressions are used to represent the variable. In Figure 6.5(a) the thermal conductivity (kw) in wK/m^3 is expressed with the primary terms like MEDIUM, LOW, HIGH etc. Each of these linguistic terms is then expressed as a triangular or trapezoidal shaped fuzzy set. The triangular or trapezoidal shaped fuzzy sets are easier to understand by the designers. It is also observed that designers can easily relate the simple triangular and trapezoidal representations to their understanding of the problem, and thus can modify them easily. For example, MEDIUM has been expressed as a list of (23 0) (26 1) (28 0), where the left value in each pair of brackets is the value of kw and the right value is the corresponding value of the membership function (Figure 6.5(b)). The next stage in the fuzzification transforms the crisp value of each design variable (as obtained from the hybrid search method described in chapter 4) into a fuzzy set using the FuzzyCLIPS defined functions (in this case S and Z functions) as shown in Figure 6.6. The spread of the fuzzy set is defined by the resolution on the design variable.

6.4.2 The Knowledge Representation

Knowledge is the essence of any expert system. The knowledge base or the rulebase for the system is developed using fuzzy rules and facts. The fuzzy rules are nothing but production rules [Durkin (1994)] integrated with fuzzy set concepts. The knowledge embodies qualitative aspects of the design problem in terms of manufacturability, choice of materials and designers' special preferences. An integration of QES with the decision support tool (that is ASM) demands qualitative evaluation of any design solution retrieved during the first stage of ASM's operation. This means that it is necessary to guarantee that the knowledge base can cover the entire design space. To develop such a system with a minimum number of rules, a novel knowledge representation is adopted. The knowledge is separated into three categories: *Inter-variable Knowledge, Intra-variable Knowledge* and *Heuristics*. Inter-variable and intra-variable knowledge is integrated by *a concept of compromise*. This approach allows the development of a knowledge base that can cover the entire problem space with a few rules. Three sets of rules concerning the intra-variable knowledge for the three different criteria are developed. The present system includes 38 fuzzy rules in total and a function that asserts fuzzy facts to represent the inter-variable knowledge (Appendix - II).

6.4.2.1 The Design Thinking Process

The integration of an adaptive search and a fuzzy expert system with the ASM has posed a challenge in terms of knowledge representation. The task is to develop a complete but small knowledge base for the 12 dimensional problem. In order to find the right knowledge representation a deeper understanding concerning the design thinking process is necessary. The design thinking process involved in the qualitative evaluation of a design solution is complex [Smith and Browne (1993), Oksala (1994)]. Designers generally respond to complexity by decomposing the whole system into parts. During the discussion with the designers on the turbine blade problem the following observations have been made for any one qualitative criterion:

a) Designers can confidently give a rating for a design solution based upon their past experience.

b) Designers' knowledge is not complete and they find it difficult to define a rating for an unseen design solution. The designer then starts decomposing the problem into smaller dimensions even to a single dimension.

c) The designer decomposes the problem using some knowledge concerning the relative importance of different design variables for the criterion.

d) It is much easier for the designer to provide a qualitative rating for a design solution which consists of a small number of design variables.

e) The designer tries to obtain an overall rating by considering some idea about interaction between variables. Often the information concerning interaction between design variables is simplified to relative importance of the variables for the criterion.

The better understanding of the underlying cognitive process for the design evaluation task provides the motivation for the knowledge representation technique. Here the knowledge is represented concerning the individual variables for each criterion. The knowledge is separated into three categories: inter-variable and intra-variable knowledge and heuristics. The designer's idea about the interaction among design variables is implemented in the knowledge integration process using a concept of compromise. The concept of compromise incorporates the notion of relative importance of design variables and an idea of *condoning the negative evaluations from a less important design variable*. The decomposition of the evaluation task and then the integration of the evaluations into an overall rating has helped to develop a complete but small knowledge base for the task. Heuristic rules are used to incorporate the designers past experiences.

6.4.2.2 The Inter-variable Knowledge

Considerations like manufacturability, choice of materials and some special preferences for a design solution dictate the relative importance of a variable. While evaluating a design for a criterion, the important variables contribute to the conclusion, whereas the least important variables are ignored. For example, if a turbine blade cooling system design is evaluated for cost of material the wall thickness (dth) becomes the most important variable. Whereas, Cdr, Fhc, Tc1 etc. are the least important variables for the criterion. A design is evaluated as bad or not good, if the design does not conform to requirements of an important design variable (later defined as intra-variable knowledge). This negative effect on the design evaluation is condoned or compromised in case of less important variables. Thus, the degree of compromise depends on the relative importance of a variable. The most important variable cannot be compromised. This inter-variable knowledge is represented by ranking each design variable between 0 and 1 for each criterion. If ranking is high that means less compromise is allowed and alternatively low ranking means higher degree of compromise is possible. The function that implements the inter-variable knowledge is shown below:

```
(deffunction Inter_Var_Preferences ()

        ;; COST OF MANUFACTURING ;;
        (assert (Geom_pref-1 (0.8 0) (0.8 1) (0.8 0)) CF 0.8)
        (assert (Cdr_pref-1 (0.3 0) (0.3 1) (0.3 0)) CF 0.8)
        (assert (Fhc_pref-1 (0.3 0) (0.3 1) (0.3 0)) CF 0.6)
        (assert (Tc1_pref-1 (0.3 0) (0.3 1) (0.3 0)) CF 0.6)
        (assert (dth_pref-1 (0.8 0) (0.8 1) (0.8 0)) CF 0.8)
        (assert (kw_pref-1 (0.4 0) (0.4 1) (0.4 0)) CF 0.9)
        (assert (Rp_pref-1 (0.3 0) (0.3 1) (0.3 0)) CF 0.8)
        (assert (Rs_pref-1 (0.3 0) (0.3 1) (0.3 0)) CF 0.6)
```

```
(assert (df_pref-1 (0.8 0) (0.8 1) (0.8 0)) CF 0.8)
(assert (Cdf_pref-1 (0.4 0) (0.4 1) (0.4 0)) CF 0.6)
(assert (Ff_pref-1 (0.4 0) (0.4 1) (0.4 0)) CF 0.6)
(assert (Rpf_pref-1 (0.3 0) (0.3 1) (0.3 0)) CF 0.8)

        ;; COST OF MATERIAL ;;
(assert (Geom_pref-2 (0.5 0) (0.5 1) (0.5 0)) CF 0.8)
(assert (Cdr_pref-2 (0.3 0) (0.3 1) (0.3 0)) CF 0.8)
(assert (Fhc_pref-2 (0.3 0) (0.3 1) (0.3 0)) CF 0.6)
(assert (Tc1_pref-2 (0.3 0) (0.3 1) (0.3 0)) CF 0.8)
(assert (dth_pref-2 (0.9 0) (0.9 1) (0.9 0)) CF 0.9)
(assert (kw_pref-2 (0.7 0) (0.7 1) (0.7 0)) CF 0.8)
(assert (Rp_pref-2 (0.5 0) (0.5 1) (0.5 0)) CF 0.8)
(assert (Rs_pref-2 (0.3 0) (0.3 1) (0.3 0)) CF 0.9)
(assert (df_pref-2 (0.3 0) (0.3 1) (0.3 0)) CF 0.8)
(assert (Cdf_pref-2 (0.4 0) (0.4 1) (0.4 0)) CF 0.6)
(assert (Ff_pref-2 (0.4 0) (0.4 1) (0.4 0)) CF 0.6)
(assert (Rpf_pref-2 (0.5 0) (0.5 1) (0.5 0)) CF 0.8)

        ;; DESIGNER'S SPECIAL PREFERENCE ;;
(assert (Geom_pref-3 (0.8 0) (0.8 1) (0.8 0)) CF 0.8)
(assert (Cdr_pref-3 (0.5 0) (0.5 1) (0.5 0)) CF 0.7)
(assert (Fhc_pref-3 (0.8 0) (0.8 1) (0.8 0)) CF 0.8)
(assert (Tc1_pref-3 (0.7 0) (0.7 1) (0.7 0)) CF 0.7)
(assert (dth_pref-3 (0.9 0) (0.9 1) (0.9 0)) CF 0.9)
(assert (kw_pref-3 (0.8 0) (0.8 1) (0.8 0)) CF 0.8)
(assert (Rp_pref-3 (0.6 0) (0.6 1) (0.6 0)) CF 0.8)
(assert (Rs_pref-3 (0.7 0) (0.7 1) (0.7 0)) CF 0.8)
(assert (df_pref-3 (0.9 0) (0.9 1) (0.9 0)) CF 0.8)
(assert (Cdf_pref-3 (0.5 0) (0.5 1) (0.5 0)) CF 0.7)
(assert (Ff_pref-3 (0.8 0) (0.8 1) (0.8 0)) CF 0.8)
(assert (Rpf_pref-3 (0.6 0) (0.6 1) (0.6 0)) CF 0.8)
)
```

The rankings are obtained from the designers. For each criterion, the designers are asked to rank the variables and also include the confidence factors for their decisions. The information presented in the thesis reflects designers' knowledge and experience with the design problem. Once the rankings are decided, the function shown above asserts fuzzy facts in the FuzzyCLIPS facts list. Three different sets of facts are introduced in the fact list for the three criteria. Each fact contains a confidence factor that represents the confidence on the variable ranking. These facts are utilised for the knowledge integration.

6.4.2.3 The Intra-variable Knowledge

According to different qualitative criteria each variable has a preferred value, say for example, the blade wall thickness (that is the variable *dth*) should be very low to reduce the cost of material but this may not be suitable from manufacturability considerations. Thus from a cost of material point of view, preference for the wall thickness (dth) is VERY_LOW but the preference is VERY_HIGH from manufacturability point of view. In a design solution, whenever the wall thickness falls within the VERY_HIGH range (previously mentioned as requirements of the variable) the design is qualitatively rated as VERY_GOOD from the manufacturability consideration. If the wall thickness is "not" VERY_HIGH then the qualitative rating is determined by compromising BAD according to the inter-variable preference of the wall thickness from manufacturability consideration. The expression "not" is the hedge or the modifier used in conjunction with the fuzzy term VERY_HIGH. This knowledge is also retrieved from the designers through interviews and detailed discussions. Conflicts in opinion between the designers are resolved by dialogue. Rules 9 and 10 as shown below exhibit the intra-variable knowledge representation for the wall thickness (dth) from the manufacturability consideration, where QR-1 is the qualitative rating.

```
(defrule rule-9
        (declare (salience -50) (CF 0.7))
        (HEURISTICS-1 NO)
        (dth VERY_HIGH)
        ?fa <- (dth_pref-1 ?)
        =>
        (bind ?cf (get-cf ?fa))
```

```
        (assert (QR-1 VERY_GOOD) CF 1.0)
)

(defrule rule-10
        (declare (salience -50) (CF 0.7))
        (HEURISTICS-1 NO)
        (dth not VERY_HIGH)
        ?fa <- (dth_pref-1 ?)
        =>
        (bind ?cf (get-cf ?fa))
        (bind ?pref (get-fs-x ?fa 0))
        (if (> ?pref 0.9)
                then
                (assert (QR-1 BAD) CF ?cf))
        (if (and (<= ?pref 0.9) (> ?pref 0.8))
                then
                (assert (QR-1 slightly_compromise BAD) CF ?cf))
        (if (and (<= ?pref 0.8) (> ?pref 0.6))
                then
                (assert (QR-1 less_compromise BAD) CF ?cf))
        (if (and (<= ?pref 0.6) (> ?pref 0.4))
                then
                (assert (QR-1 compromise BAD) CF ?cf))
        (if (<= ?pref 0.4)
                then
                (assert (QR-1 more_compromise BAD) CF ?cf))
)
```

6.4.2.4 The Concept of Compromise

The literal meaning of compromise is "to settle (a dispute) by making concessions". The same concept of concession is implemented as "reducing the severity of the negative effect of one individual variable on the final qualitative rating" (Figure 6.7). The inter-variable knowledge determines the degree of compromise possible on every variable. Different degrees of compromise are described as *slightly_compromise*, *less_compromise*, *compromise* and *more_compromise* (Figure 6.8). If the inter-variable rating for a variable is more than 0.9 that variable is not compromised. QES uses the min-max type fuzzy

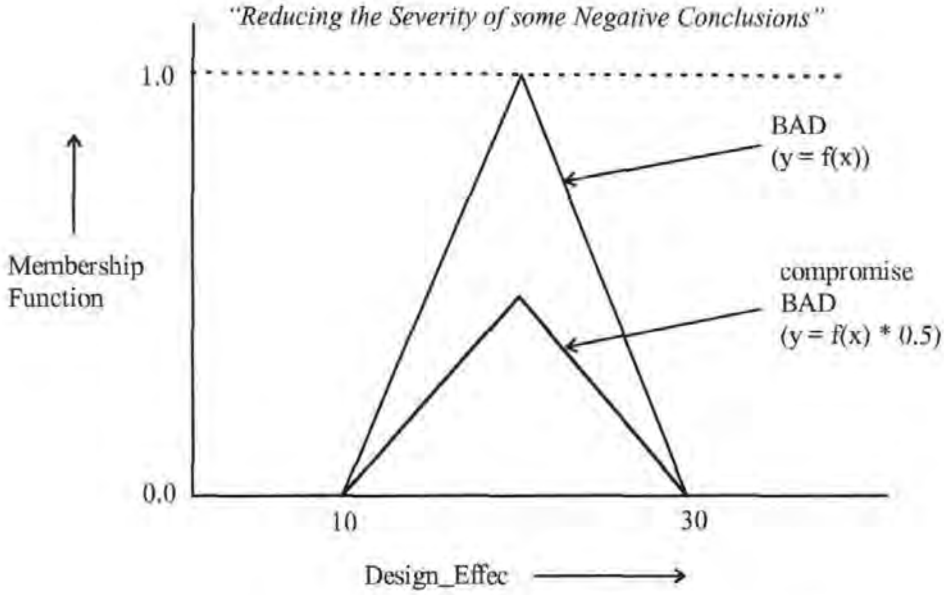


Figure 6.7: The Concept of Compromise.

Modifier Name	**Modifier Description**
slightly_compromise	*y * 0.9*
less_compromise	*y * 0.7*
compromise	*y * 0.5*
more_compromise	*y * 0.3*

Figure 6.8: The descriptions of different compromise modifier used, where y denotes the corresponding membership function.

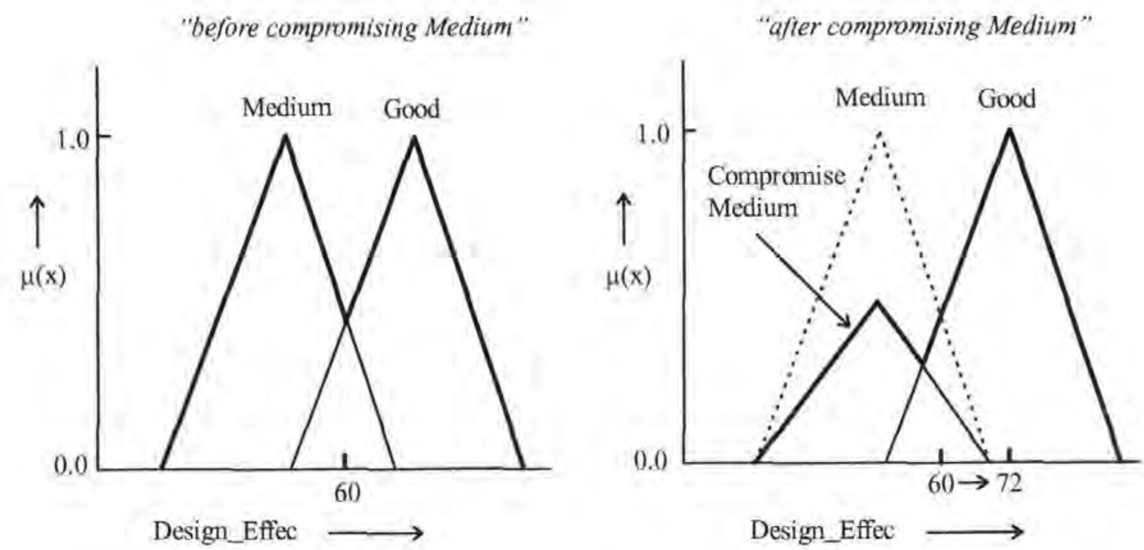


Figure 6.9: The effect of compromise: the design effectiveness is increased by compromising the "Medium" (relatively negative conclusion) fuzzy set.

inference mechanism as used in FuzzyCLIPS. The fuzzy inference mechanism works on the principle of global contribution to the facts list. That means when a new fact is asserted about a variable any previous fact about the variable gets replaced by a different fact that is a union of the two facts. Once all relevant rules are fired fuzzy representations of the qualitative ratings are obtained. The centre of gravity type defuzzification method is used to obtain crisp values for the three qualitative ratings. In the defuzzification process the crisp value is obtained depending on the area covered by the final fuzzy fact. The concept of compromise is implemented by reducing the area due to the negative conclusions (Figure 6.7). The method uses a multiplication modifier. Reducing the area at one end pushes the crisp value towards the other end. Thus by reducing the area covered due to the negative conclusions the crisp value obtained for the qualitative rating can be pushed towards the positive conclusion end (that is a higher qualitative rating can be obtained) (Figure 6.9). In other words the severity of the negative conclusions is reduced. Thus the corresponding variable is compromised (to a different degree). This novel but simple method of knowledge integration has helped to develop a knowledge base that covers the entire design space but uses a small number of fuzzy rules. Three different qualitative ratings are obtained for the three criteria. The present system includes 38 fuzzy rules and a function that asserts fuzzy facts to represent the inter-variable knowledge (Appendix - II).

6.4.2.5 Heuristic Rules

Heuristics are the most common way of expressing domain knowledge. The inter-variable and intra-variable knowledge and the method by which they are integrated can evaluate any solution in the design space. Even then, some definite heuristics from the designer are included. These are mostly about some specific cases where the designer is definite about the conclusion. If any design solution matches with any one of the heuristic rules, previous conclusions are discarded and only the heuristic is used for the conclusion. Three different sets of heuristic rules are used for the three criteria (i.e. manufacturability, cost of material and designers' special preferences). One sample heuristic rule for the designers' special preference criterion is shown below.

```
(defrule rule-37                              ;; heuristics-6
        (declare (salience -450) (CF 0.7))
        (Cdr VERY_LOW)
        (dth HIGH)
        (Rp VERY_HIGH)
        (Rs VERY_LOW)
        (df BIG)
        (Rpf VERY_HIGH)
        ?fa <- (QR-3 ?)
        ?he <- (HEURISTICS-3 NO)
        =>
        (printout t crlf)
        (printout t " The design has satisfied Heuristic-6 (Designer's Special Preference) :: " crlf)
        (printout t "                              Cdr  : VERY_LOW" crlf)
        (printout t "                              dth  : HIGH" crlf)
        (printout t "                              Rp   : VERY_HIGH" crlf)
        (printout t "                              Rs   : VERY_LOW" crlf)
        (printout t "                              df   : BIG" crlf)
        (printout t "                              Rpf  : VERY_HIGH" crlf)
        (printout t crlf)
        (retract ?fa)
        (retract ?he)
        (assert (HEURISTICS-3 YES))
        (assert (QR-3 GOOD) CF 1.0)
)
```

6.4.3 Building the System

Now that the fuzzy sets and rules are defined, the next task is to build the system. This task involves coding of the fuzzy sets, and rules and selection of proper fuzzy logic procedures. Max-min type fuzzy inferencing and centre of gravity type defuzzification are selected for the development. An additional task involves the integration of the ARTS based hybrid GA search with QES in the Adaptive Search Manager environment. Coding involved in QES development is developed following FuzzyCLIPS nomenclature and the procedures used are selected from the available FuzzyCLIPS functions. The ARTS based hybrid GA search technique is implemented in "C" language, and involves many structures to define the principle components of the search technique. Thus the next task of integrating the search technique with QES demands data handling from and to "C" structures by the Adaptive Search Manager. The methodology involved in mirroring "C" structures in FuzzyCLIPS for the integration can be summarised as follows:

1. Classes and instances in FuzzyCLIPS are defined to mirror "C" structures using CLIPS Object Oriented Language (COOL).
2. "C" functions are written to extract data from FuzzyCLIPS instances.
3. "C" functions are written to be used in FuzzyCLIPS to extract data from "C" structures.
4. UserFunctions(), a FuzzyCLIPS function, is modified to accommodate the change and then FuzzyCLIPS is recompiled.

QES evaluations are presented along with other details of the selected designs through an *adaptive search manager (ASM) interface*. The interface provides an access to the linguistic term definitions used for the design variables and the fuzzy knowledge base. This flexibility helps the designer to adapt the system quickly to any new situation and also to add any new criterion if necessary.

6.4.4 Validating the System

Any expert system needs to be validated in order to check the validity of the conclusions. Validation of QES has been performed by experts from Rolls Royce plc. Several cases are verified to check whether the conclusions conform to the expectations of the experts [Satre and Massey (1991), Massey et. al. (1991)]. In an early attempt only one qualitative rating was obtained for all the three criteria [Roy et. al. (1995a) and (1996c)]. During the validation separate ratings are given for each qualitative criterion, i.e. manufacturability, cost of material and designers' special preferences. As a part of the validation process, the knowledge base is verified in detail and accordingly few rules are modified by the experts. The linguistic terms used with the variables are kept unaltered. Often contradictory opinions are resolved amicably through discussions. This validation process requires few iterations. It is observed that, while defining the intra-variable knowledge that some variables can have an indirect effect on a criterion. The fuzzy rules are developed using only the direct effect consideration. It is generally agreed that the validation process also contributes towards a better understanding of the problem.

6.5 Summary

The chapter presents a method of evaluating engineering designs with respect to a qualitative criterion. The methodology uses a fuzzy expert system that provides three crisp ratings for a design solution considering three different qualitative criteria. The chapter briefly introduces the concepts of fuzzy logic and fuzzy expert systems. The research describes the development of a qualitative evaluation system using a novel knowledge representation technique. The technique helps to integrate the system with the hybrid search technique (ARTS based GA search and hill climbing). The system is capable of evaluating any possible design using a small number of rules. Finally the validation procedure for the qualitative evaluation system is described. The next two chapters present the results obtained using ASM, discussion on the results and finally the conclusions.

CHAPTER - 7

7. Results and Evaluation

7.1 ASM and Design Decision Support

ASM is used as a decision support tool for the turbine blade cooling system design problem. ASM presents multiple "good" design solutions for the problem to the designer. It is necessary that the information provided by ASM is evaluated before the system can be used in practice. The evaluation process can be divided into two categories: verification of the results and validation of the approach adopted in ASM. The objective of the verification stage [Satre and Massey (1991)] is to check the quality of the results achieved using ASM. It is important to verify whether the quantitative and qualitative information concerning the design solutions conforms to the understanding of the expert designers. The real life design problem (TBCOM) does not provide prior knowledge concerning possible "good" designs: how good can they be and where are they located? In absence of such knowledge, it is very difficult to judge the quality of the results automatically (that is using a computerised verification method). The validation of the design decision support approach is also essential to identify how well the system can support a designer in his or her needs. A numerical measurement of the overall success of ASM is impossible. Thus ASM needs to be evaluated qualitatively. The evaluation is performed by an expert designer, and a user. Many runs of ASM are performed to obtain information at different conditions, and the results are used in the evaluation process. The next section describes different parameters used in the search process and presents some representative results from the runs.

7.2 Results

Some representative results are reported from typical ASM runs. Results obtained from ASM at different conditions are used to evaluate the system. The hybrid of ARTS based search and the knowledge based hill climbing stops if the search attains a steady state. The research defines two different definitions of the steady state:

1. A steady state is achieved if the average fitness (that is the inverse of the coolant mass flow through the radial passage) of the population per generation remains unchanged for 100 generations.

2. Alternatively, the search is considered to have attained a steady state if the search cannot find any new sub-optimum for 100 generations. In this case, a list of sub-optima is maintained. The list is of a fixed size and is updated every generation. If the search fails to update the list for 100 generations then it is assumed the search has reached a steady state.

It is worth noting that, the first steady state condition was used in Roy and Parmee (1995) and (1996). Later, in order to reduce the run time of ASM the second steady state definition is used. The results using the second definition are used for the evaluation purpose and are presented in this section. The design solutions in the designs' list are considered "good" designs as identified by ASM. Many runs of ASM are performed with different ranges for the design variables and the constraints, and the results are presented to the expert and the user to evaluate the system. A sample of the results is presented in this section. Each design solution is represented by twelve values for the design variables (i.e. Geom, Cdr, etc.). The quantitative information about the "good" designs includes the fitness of a design and the sensitivity information. The fitness is the inverse of the coolant mass flow through the radial passage. The sensitivity analysis is limited within the geometry type (Geom) of a design solution. The design solution sensitivity shows the variation of the design solution performance within the neighbourhood. The higher the sensitivity value the more sensitive is the design. In this case the performance of a design solution is the amount of coolant mass flow through the radial passage. The design variable sensitivity values are rounded to the nearest integer values. The eleven values for the design variable sensitivities (that is for the eleven design variables except the geometry) exhibit how critical each design variable is within the neighbourhood of a design solution. The higher the sensitivity value the more sensitive is the design variable. The constraint sensitivity is determined for the three constraints individually. The qualitative information contains three qualitative ratings considering the three criteria: manufacturability, choice of materials and designers' special preferences. The higher the rating, the more effective the design solution is from the qualitative criterion point of view. The quantitative and qualitative information is presented to the designer as decision support. The information helps the designer to compare between the designs, as a result the designer selects the most appropriate design.

ARTS based GA search uses two control parameters, K and KT, for the clustering. The value of K is set to be one fifth of the cluster list size. The value of KT is set to be 90% of the value of K. In order to maximise the diversity of the initial population, the design space is divided into 24 equal hyper spaces, then individuals are produced randomly from each hyper space in equal numbers. The total number of individuals in a population is 120. The binary string for one individual is 272 bits long. The search process uses one point crossover with probability 1.0 but no mutation or inversion. All the trial runs use the same seed for the random number generator. The runs are performed under identical computing conditions. Each design variable is defined by an upper bound (*top*), a lower bound (*bottom*), a resolution (*res*), a tolerance (*tol*) and a pre-probability (*pre_prob*) for the knowledge based hill climbing. There are three possible types of internal geometry for the cooling passage: *plane*, *ribbed* and *pedestal*, and geometry type determines the ranges for the design variables: Cdr and Fhc. The three non-linear inequality constraints are also defined with a maximum and a minimum limit. The ranges for each design variable or each constraint can be altered by a designer at the beginning of an ASM run. The results shown in this section use the default settings for the design variables and the constraints. The default values of the design variables and the constraints are presented below. The design variables and the constraints assume the same nomenclature and units as mentioned in chapter 2.

THE DEFAULT SETTINGS OF THE DESIGN VARIABLES:

Design Variable: Geom
top = 3
bottom = 1
res = 0.5
tol = 1.0
pre_prob = 0.00

Design Variable: Cdr-1 /* Cdr for *plane* type geometry */
top = 0.75
bottom = 0.60
res = 0.01
tol = 0.02
pre_prob = 0.1

Design Variable: Cdr-2 /* Cdr for *ribbed* type geometry */
top = 0.6
bottom = 0.4
res = 0.01
tol = 0.02
pre_prob = 0.1

Design Variable: Cdr-3 /* Cdr for *pedestal* type geometry */
top = 0.4
bottom = 0.2
res = 0.01
tol = 0.02
pre_prob = 0.1

Design Variable: Fhc-1 /* Fhc for *plane* type geometry */
top = 1.6
bottom = 1.0
res = 0.1
tol = 0.1
pre_prob = 0.1

Design Variable: Fhc-2 /* Fhc for *ribbed* type geometry */
top = 3.0

bottom = 1.3
res = 0.1
tol = 0.2
pre_prob = 0.1

Design Variable: Fhc-3 /* Fhc for *pedestal* type geometry */
top = 3.2
bottom = 1.8
res = 0.1
tol = 0.2
pre_prob = 0.1

Design Variable: Tc1
top = 800
bottom = 700
res = 1
tol = 2.0
pre_prob = 0.08

Design Variable: dth
top = 0.0025
bottom = 0.00075
res = 0.00001
tol = 0.00005
pre_prob = 0.16

Design Variable: kw
top = 33
bottom = 18
res = 1.0
tol = 2.0
pre_prob = 0.1

Design Variable: Rp
top = 1.6
bottom = 1.05
res = 0.01
tol = 0.03
pre_prob = 0.2

Design Variable: Rs
top = 1.50
bottom = 0.50
res = 0.01
tol = 0.05
pre_prob = 0.12

Design Variable: df
top = 0.0004
bottom = 0.0001
res = 0.00005
tol = 0.00005
pre_prob = 0.04

Design Variable: Cdf
top = 0.75
bottom = 0.6
res = 0.01
tol = 0.02
pre_prob = 0.04

Design Variable: Ff
top = 1.6
bottom = 1.0
res = 0.1
tol = 0.1
pre_prob = 0.02

Design Variable: Rpf
top = 0.4
bottom = 0.2
res = 0.01
tol = 0.02
pre_prob = 0.04

THE DEFAULT SETTINGS OF THE CONSTRAINTS:

Constraint - 1:
C1MAX = 1300.0
C1MIN = 1200.0

Constraint - 2:
C2MAX = 1300.0
C2MIN = 0.0

Constraint - 3:
C3MAX = 100000.0
C3MIN = 0.8

7.2.1 Results: unconstrained search

The unconstrained search is performed by setting the constraints' upper limits to very large numbers and lower limits to zero. The alterations are performed through an interactive session at the beginning of the ASM run. The run uses the second steady state condition. The results from a typical ASM run are presented. The ASM run is completed after 203 generations. The run produces 5 "good" designs from geometry one, five from geometry two and five from geometry three. The fifteen "good" designs are then presented to the

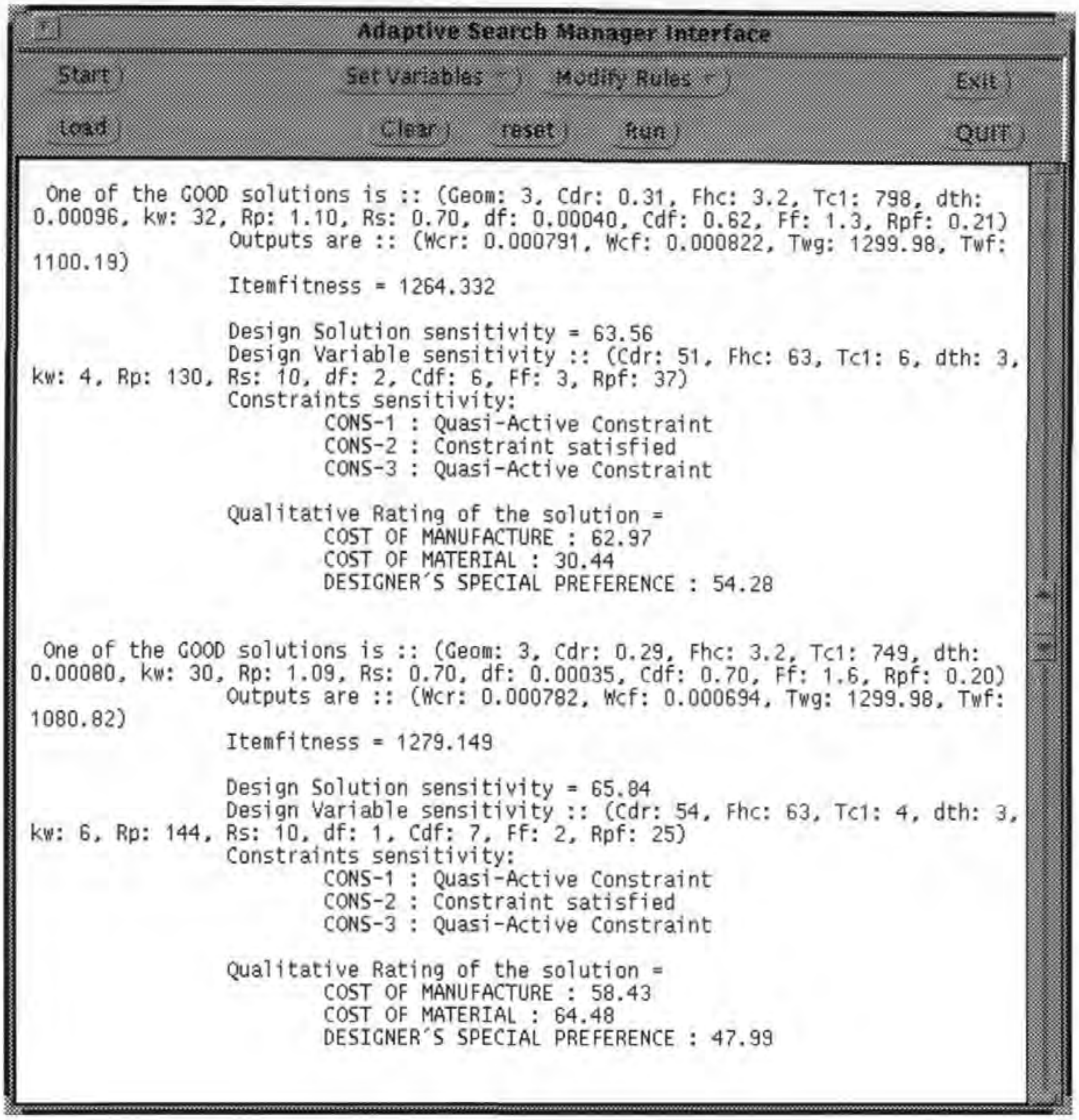


Figure 7.1: The Adaptive Search Manager Interface.

designer through the ASM interface (Figure 7.1) for the final decision. The "good" designs are shown below:

DESIGN: 1

One of the GOOD solutions is :: (Geom: **1**, Cdr: 0.61, Fhc: 1.4, Tc1: 776, dth: 0.00250, kw: 31, Rp: 1.05, Rs: 1.50, df: 0.00035, Cdf: 0.67, Ff: 1.5, Rpf: 0.22)

Quantitative Information:

Fitness = **1990.185**

Design Solution sensitivity = **56.72**

Design Variable sensitivity :: (Cdr: **18**, Fhc: 14, Tc1: 3, dth: 1, kw: 1, Rp: **94**, Rs: 4, df: 0, Cdf: 1, Ff: 2, Rpf: **25**)

Constraints sensitivity:

CONS-1 : Constraint satisfied

CONS-2 : Constraint satisfied

CONS-3 : Constraint satisfied

Qualitative Information:

Qualitative Rating of the solution:

COST OF MANUFACTURE: **64.74**

COST OF MATERIAL: **55.26**
DESIGNER'S SPECIAL PREFERENCE: **31.53**

DESIGN: 2
One of the GOOD solutions is :: (Geom: **1**, Cdr: 0.60, Fhc: 1.0, Tc1: 800, dth: 0.00250, kw: 18, Rp: 1.05, Rs: 1.50, df: 0.00010, Cdf: 0.74, Ff: 1.3, Rpf: 0.20)

Quantitative Information:

Fitness = **2283.391**
Design Solution sensitivity = **55.57**
Design Variable sensitivity :: (Cdr: 11, Fhc: **12**, Tc1: 2, dth: 1, kw: 1, Rp: **84**, Rs: 1, df: 0, Cdf: 1, Ff: 1, Rpf: **16**)
Constraints sensitivity:
CONS-1 : Constraint satisfied
CONS-2 : Constraint satisfied
CONS-3 : Constraint satisfied

Qualitative Information:

Qualitative Rating of the solution:
COST OF MANUFACTURE: **63.24**
COST OF MATERIAL: **55.26**
DESIGNER'S SPECIAL PREFERENCE: **15.81**

DESIGN: 3
One of the GOOD solutions is :: (Geom: **1**, Cdr: 0.60, Fhc: 1.1, Tc1: 797, dth: 0.00250, kw: 28, Rp: 1.05, Rs: 1.50, df: 0.00010, Cdf: 0.74, Ff: 1.1, Rpf: 0.20)

Quantitative Information:

Fitness = **2258.755**
Design Solution sensitivity = **55.63**
Design Variable sensitivity :: (Cdr: 11, Fhc: **13**, Tc1: 2, dth: 1, kw: 1, Rp: **85**, Rs: 3, df: 0, Cdf: 1, Ff: 1, Rpf: **16**)
Constraints sensitivity:
CONS-1 : Constraint satisfied
CONS-2 : Constraint satisfied
CONS-3 : Constraint satisfied

Qualitative Information:

Qualitative Rating of the solution:
COST OF MANUFACTURE: **63.24**
COST OF MATERIAL: **55.26**
DESIGNER'S SPECIAL PREFERENCE: **15.81**

DESIGN: 4
One of the GOOD solutions is :: (Geom: 1, Cdr: 0.60, Fhc: 1.1, Tc1: 800, dth: 0.00250, kw: 28, Rp: 1.05, Rs: 1.50, df: 0.00010, Cdf: 0.74, Ff: 1.1, Rpf: 0.20)

Quantitative Information:

Fitness = **2277.903**
Design Solution sensitivity = **55.57**
Design Variable sensitivity :: (Cdr: 11, Fhc: **12**, Tc1: 2, dth: 1, kw: 1, Rp: **84**, Rs: 1, df: 0, Cdf: 1, Ff: 1, Rpf: **16**)
Constraints sensitivity:
CONS-1 : Constraint satisfied
CONS-2 : Constraint satisfied
CONS-3 : Constraint satisfied

Qualitative Information:

Qualitative Rating of the solution:
COST OF MANUFACTURE: **63.24**
COST OF MATERIAL: **55.26**
DESIGNER'S SPECIAL PREFERENCE: **15.81**

DESIGN: 5
One of the GOOD solutions is :: (Geom: **1**, Cdr: 0.60, Fhc: 1.1, Tc1: 800, dth: 0.00250, kw: 18, Rp: 1.05, Rs: 1.50, df: 0.00010, Cdf: 0.74, Ff: 1.1, Rpf: 0.20)

Quantitative Information:

Fitness = **2278.954**

Design Solution sensitivity = **55.56**
Design Variable sensitivity :: (Cdr: 11, Fhc: **12**, Tc1: 2, dth: 1, kw: 1, Rp: **84**, Rs: 1, df: 0, Cdf: 1, Ff: 1, Rpf: **16**)
Constraints sensitivity:
CONS-1 : Constraint satisfied
CONS-2 : Constraint satisfied
CONS-3 : Constraint satisfied
Qualitative Information:
Qualitative Rating of the solution:
COST OF MANUFACTURE: **63.24**
COST OF MATERIAL: **55.26**
DESIGNER'S SPECIAL PREFERENCE: **15.81**

DESIGN: 6
One of the GOOD solutions is :: (Geom: **2**, Cdr: 0.40, Fhc: 1.8, Tc1: 800, dth: 0.00250, kw: 18, Rp: 1.05, Rs: 1.50, df: 0.00020, Cdf: 0.61, Ff: 1.5, Rpf: 0.20)
Quantitative Information:
Fitness = **3366.266**
Design Solution sensitivity = **55.73**
Design Variable sensitivity :: (Cdr: **11**, Fhc: 8, Tc1: 1, dth: 1, kw: 1, Rp: **57**, Rs: 1, df: 0, Cdf: 1, Ff: 1, Rpf: **11**)
Constraints sensitivity:
CONS-1 : Constraint satisfied
CONS-2 : Constraint satisfied
CONS-3 : Constraint satisfied
Qualitative Information:
Qualitative Rating of the solution:
COST OF MANUFACTURE: **56.65**
COST OF MATERIAL: **15.81**
DESIGNER'S SPECIAL PREFERENCE: **55.25**

DESIGN: 7
One of the GOOD solutions is :: (Geom: **2**, Cdr: 0.40, Fhc: 1.8, Tc1: 800, dth: 0.00250, kw: 20, Rp: 1.05, Rs: 1.50, df: 0.00020, Cdf: 0.72, Ff: 1.0, Rpf: 0.20)
Quantitative Information:
Fitness = **3365.623**
Design Solution sensitivity = **55.73**
Design Variable sensitivity :: (Cdr: **11**, Fhc: 8, Tc1: 1, dth: 1, kw: 1, Rp: **57**, Rs: 1, df: 0, Cdf: 1, Ff: 1, Rpf: **11**)
Constraints sensitivity:
CONS-1 : Constraint satisfied
CONS-2 : Constraint satisfied
CONS-3 : Constraint satisfied
Qualitative Information:
Qualitative Rating of the solution:
COST OF MANUFACTURE: **56.65**
COST OF MATERIAL: **15.81**
DESIGNER'S SPECIAL PREFERENCE: **27.64**

DESIGN: 8
One of the GOOD solutions is :: (Geom: **2**, Cdr: 0.40, Fhc: 1.7, Tc1: 800, dth: 0.00250, kw: 18, Rp: 1.05, Rs: 1.50, df: 0.00010, Cdf: 0.66, Ff: 1.5, Rpf: 0.20)
Quantitative Information:
Fitness = **3371.235**
Design Solution sensitivity = **55.74**
Design Variable sensitivity :: (Cdr: **11**, Fhc: 8, Tc1: 1, dth: 1, kw: 1, Rp: **57**, Rs: 1, df: 0, Cdf: 1, Ff: 1, Rpf: **11**)
Constraints sensitivity:
CONS-1 : Constraint satisfied
CONS-2 : Constraint satisfied
CONS-3 : Constraint satisfied
Qualitative Information:

Qualitative Rating of the solution:

COST OF MANUFACTURE: **56.65**

COST OF MATERIAL: **15.81**

DESIGNER'S SPECIAL PREFERENCE: **18.25**

DESIGN: 9

One of the GOOD solutions is :: (Geom: **2**, Cdr: 0.40, Fhc: 1.3, Tc1: 800, dth: 0.00250, kw: 18, Rp: 1.05, Rs: 1.50, df: 0.00020, Cdf: 0.72, Ff: 1.0, Rpf: 0.20)

Quantitatice Information:

Fitness = **3394.035**

Design Solution sensitivity = **55.78**

Design Variable sensitivity :: (Cdr: **11**, Fhc: 8, Tc1: 1, dth: 1, kw: 1, Rp: **57**, Rs: 1, df: 0, Cdf: 1, Ff: 1, Rpf: **11**)

Constraints sensitivity:

CONS-1 : Constraint satisfied

CONS-2 : Constraint satisfied

CONS-3 : Constraint satisfied

Qualitative Information:

Qualitative Rating of the solution:

COST OF MANUFACTURE: **56.65**

COST OF MATERIAL: **15.81**

DESIGNER'S SPECIAL PREFERENCE: **27.64**

DESIGN: 10

One of the GOOD solutions is :: (Geom: **2**, Cdr: 0.40, Fhc: 1.8, Tc1: 800, dth: 0.00250, kw: 28, Rp: 1.05, Rs: 1.50, df: 0.00025, Cdf: 0.68, Ff: 1.3, Rpf: 0.20)

Quantitative Information:

Fitness = **3363.944**

Design Solution sensitivity = **55.74**

Design Variable sensitivity :: (Cdr: **11**, Fhc: 8, Tc1: 1, dth: 1, kw: 1, Rp: **57**, Rs: 1, df: 0, Cdf: 1, Ff: 1, Rpf: **11**)

Constraints sensitivity:

CONS-1 : Constraint satisfied

CONS-2 : Constraint satisfied

CONS-3 : Constraint satisfied

Qualitative Information:

Qualitative Rating of the solution:

COST OF MANUFACTURE: **56.65**

COST OF MATERIAL: **15.81**

DESIGNER'S SPECIAL PREFERENCE: **37.81**

DESIGN: 11

One of the GOOD solutions is :: (Geom: **3**, Cdr: 0.20, Fhc: 1.8, Tc1: 799, dth: 0.00250, kw: 20, Rp: 1.05, Rs: 1.50, df: 0.00015, Cdf: 0.74, Ff: 1.2, Rpf: 0.20)

Quantitative Information:

Fitness = **6664.488**

Design Solution sensitivity = **57.12**

Design Variable sensitivity :: (Cdr: **11**, Fhc: 4, Tc1: 1, dth: 0, kw: 1, Rp: **29**, Rs: 1, df: 0, Cdf: 0, Ff: 0, Rpf: **5**)

Constraints sensitivity:

CONS-1 : Constraint satisfied

CONS-2 : Constraint satisfied

CONS-3 : Constraint satisfied

Qualitative Information:

The design has satisfied Heuristic-3 (Cost of Material):

Geom: THREE

Tc1: VERY_HIGH

dth: VERY_HIGH

kw: VERY_HIGH or VERY_LOW

Rs: VERY_HIGH

Qualitative Rating of the solution:

COST OF MANUFACTURE: **58.46**

COST OF MATERIAL: **8.61**
DESIGNER'S SPECIAL PREFERENCE: **54.70**

DESIGN: 12
One of the GOOD solutions is :: (Geom: **3**, Cdr: 0.20, Fhc: 1.8, Tc1: 800, dth: 0.00250, kw: 26, Rp: 1.05, Rs: 1.50, df: 0.00015, Cdf: 0.74, Ff: 1.3, Rpf: 0.20)
Quantitative Information:
Fitness = **6681.801**
Design Solution sensitivity = **57.04**
Design Variable sensitivity :: (Cdr: **11**, Fhc: 4, Tc1: 1, dth: 0, kw: 1, Rp: **29**, Rs: 1, df: 0, Cdf: 0, Ff: 0, Rpf: **5**)
Constraints sensitivity:
CONS-1 : Constraint satisfied
CONS-2 : Constraint satisfied
CONS-3 : Constraint satisfied
Qualitative Information:
Qualitative Rating of the solution:
COST OF MANUFACTURE: **58.46**
COST OF MATERIAL: **15.81**
DESIGNER'S SPECIAL PREFERENCE: **54.70**

DESIGN: 13
One of the GOOD solutions is :: (Geom: **3**, Cdr: 0.20, Fhc: 1.8, Tc1: 799, dth: 0.00250, kw: 19, Rp: 1.05, Rs: 1.50, df: 0.00035, Cdf: 0.74, Ff: 1.2, Rpf: 0.20)
Quantitative Information:
Fitness = **6664.893**
Design Solution sensitivity = **57.12**
Design Variable sensitivity :: (Cdr: **11**, Fhc: 4, Tc1: 1, dth: 0, kw: 1, Rp: **29**, Rs: 1, df: 0, Cdf: 0, Ff: 0, Rpf: **5**)
Constraints sensitivity:
CONS-1 : Constraint satisfied
CONS-2 : Constraint satisfied
CONS-3 : Constraint satisfied
Qualitative Information:
The design has satisfied Heuristic-3 (Cost of Material) :
Geom: THREE
Tc1: VERY_HIGH
dth: VERY_HIGH
kw: VERY_HIGH or VERY_LOW
Rs: VERY_HIGH
Qualitative Rating of the solution:
COST OF MANUFACTURE: **60.00**
COST OF MATERIAL: **8.61**
DESIGNER'S SPECIAL PREFERENCE: **53.44**

DESIGN: 14
One of the GOOD solutions is :: (Geom: **3**, Cdr: 0.20, Fhc: 1.8, Tc1: 800, dth: 0.00250, kw: 18, Rp: 1.05, Rs: 1.50, df: 0.00040, Cdf: 0.69, Ff: 1.6, Rpf: 0.20)
Quantitative Information:
Fitness = **6684.441**
Design Solution sensitivity = **57.04**
Design Variable sensitivity :: (Cdr: **11**, Fhc: 4, Tc1: 1, dth: 0, kw: 1, Rp: **29**, Rs: 1, df: 0, Cdf: 0, Ff: 0, Rpf: **5**)
Constraints sensitivity:
CONS-1 : Constraint satisfied
CONS-2 : Constraint satisfied
CONS-3 : Constraint satisfied
Qualitative Information:
The design has satisfied Heuristic-3 (Cost of Material):
Geom: THREE
Tc1: VERY_HIGH
dth: VERY_HIGH

kw: VERY_HIGH or VERY_LOW
Rs: VERY_HIGH
Qualitative Rating of the solution:
COST OF MANUFACTURE: **62.97**
COST OF MATERIAL: **8.61**
DESIGNER'S SPECIAL PREFERENCE: **52.30**

DESIGN: 15
One of the GOOD solutions is :: (Geom: **3**, Cdr: 0.20, Fhc: 1.8, Tc1: 799, dth: 0.00250, kw: 26, Rp: 1.05, Rs: 1.50, df: 0.00030, Cdf: 0.74, Ff: 1.2, Rpf: 0.20)
Quantitative Information:
Fitness = **6662.703**
Design Solution sensitivity = **57.13**
Design Variable sensitivity :: (Cdr: **11**, Fhc: 4, Tc1: 1, dth: 0, kw: 1, Rp: **29**, Rs: 1, df: 0, Cdf: 0, Ff: 0, Rpf: **5**)
Constraints sensitivity:
CONS-1 : Constraint satisfied
CONS-2 : Constraint satisfied
CONS-3 : Constraint satisfied
Qualitative Information:
Qualitative Rating of the solution:
COST OF MANUFACTURE: **58.46**
COST OF MATERIAL: **15.81**
DESIGNER'S SPECIAL PREFERENCE: **56.98**

7.2.2 Results: constrained search

The constrained search is performed with the default settings of the three constraints. The constraints are implemented using the penalty functions described in chapter 4. The run uses the second steady state condition. The results from a typical run of ASM are presented. The ASM run is completed after 383 generations. The run produces 5 "good" designs from geometry one, five from geometry two and five from geometry three. All these fifteen "good" designs are then presented to the designer through the ASM interface for the final decision. The results are shown below:

DESIGN: 1
One of the GOOD solutions is :: (Geom: **1**, Cdr: 0.61, Fhc: 1.6, Tc1: 768, dth: 0.00135, kw: 27, Rp: 1.05, Rs: 0.68, df: 0.00025, Cdf: 0.60, Ff: 1.3, Rpf: 0.37)
Quantitative Information:
Fitness = **636.269**
Design Solution sensitivity = **59.10**
Design Variable sensitivity :: (Cdr: **57**, Fhc: **142**, Tc1: 12, dth: 4, kw: 4, Rp: **287**, Rs: 13, df: 2, Cdf: 11, Ff: 3, Rpf: 46)
Constraints sensitivity:
CONS-1 : Quasi-Active Constraint
CONS-2 : Constraint satisfied
CONS-3 : Constraint satisfied
Qualitative Information:
Qualitative Rating of the solution:
COST OF MANUFACTURE: **56.97**
COST OF MATERIAL: **55.26**
DESIGNER'S SPECIAL PREFERENCE: **60.25**

DESIGN: 2

One of the GOOD solutions is :: (Geom: **1**, Cdr: 0.61, Fhc: 1.6, Tc1: 702, dth: 0.00150, kw: 28, Rp: 1.07, Rs: 0.74, df: 0.00040, Cdf: 0.70, Ff: 1.2, Rpf: 0.22)

Quantitative Information:

Fitness = **639.949**

Design Solution sensitivity = **73.32**

Design Variable sensitivity :: (Cdr: 51, Fhc: **117**, Tc1: 15, dth: 3, kw: 5, Rp: **368**, Rs: 15, df: 3, Cdf: 17, Ff: 10, Rpf: **69**)

Constraints sensitivity:

CONS-1 : Quasi-Active Constraint

CONS-2 : Constraint satisfied

CONS-3 : Constraint satisfied

Qualitative Information:

Qualitative Rating of the solution:

COST OF MANUFACTURE: **63.00**

COST OF MATERIAL: **55.26**

DESIGNER'S SPECIAL PREFERENCE: **57.45**

DESIGN: 3

One of the GOOD solutions is :: (Geom: **1**, Cdr: 0.61, Fhc: 1.6, Tc1: 702, dth: 0.00150, kw: 30, Rp: 1.06, Rs: 0.69, df: 0.00035, Cdf: 0.60, Ff: 1.2, Rpf: 0.22)

Quantitative Information:

Fitness = **642.226**

Design Solution sensitivity = **81.07**

Design Variable sensitivity :: (Cdr: 51, Fhc: **124**, Tc1: 17, dth: 3, kw: 6, Rp: **433**, Rs: 16, df: 3, Cdf: 21, Ff: 11, Rpf: **69**)

Constraints sensitivity:

CONS-1 : Quasi-Active Constraint

CONS-2 : Constraint satisfied

CONS-3 : Constraint satisfied

Qualitative Information:

Qualitative Rating of the solution:

COST OF MANUFACTURE: **58.68**

COST OF MATERIAL: **55.26**

DESIGNER'S SPECIAL PREFERENCE: **56.71**

DESIGN: 4

One of the GOOD solutions is :: (Geom: **1**, Cdr: 0.61, Fhc: 1.6, Tc1: 705, dth: 0.00150, kw: 30, Rp: 1.06, Rs: 0.69, df: 0.00035, Cdf: 0.61, Ff: 1.4, Rpf: 0.22)

Quantitative Information:

Fitness = **646.875**

Design Solution sensitivity = **81.07**

Design Variable sensitivity :: (Cdr: 51, Fhc: **123**, Tc1: 17, dth: 3, kw: 6, Rp: **430**, Rs: 16, df: 3, Cdf: 21, Ff: 11, Rpf: **68**)

Constraints sensitivity:

CONS-1 : Quasi-Active Constraint

CONS-2 : Constraint satisfied

CONS-3 : Constraint satisfied

Qualitative Information:

Qualitative Rating of the solution:

COST OF MANUFACTURE: **58.68**

COST OF MATERIAL: **55.26**

DESIGNER'S SPECIAL PREFERENCE: **56.71**

DESIGN: 5

One of the GOOD solutions is :: (Geom: **1**, Cdr: 0.60, Fhc: 1.6, Tc1: 705, dth: 0.00148, kw: 32, Rp: 1.06, Rs: 0.67, df: 0.00040, Cdf: 0.63, Ff: 1.3, Rpf: 0.21)

Quantitative Information:

Fitness = **651.386**

Design Solution sensitivity = **81.06**

Design Variable sensitivity :: (Cdr: 34, Fhc: **128**, Tc1: 19, dth: 2, kw: 6, Rp: **432**, Rs: 11, df: 3, Cdf: 22, Ff: 12, Rpf: **72**)

Constraints sensitivity:

CONS-1 : Quasi-Active Constraint

CONS-2 : Constraint satisfied

CONS-3 : Constraint satisfied

Qualitative Information:

Qualitative Rating of the solution:

COST OF MANUFACTURE: **62.96**

COST OF MATERIAL: **55.26**

DESIGNER'S SPECIAL PREFERENCE: **56.35**

DESIGN: 6

One of the GOOD solutions is :: (Geom: **2**, Cdr: 0.41, Fhc: 3.0, Tc1: 766, dth: 0.00136, kw: 33, Rp: 1.07, Rs: 0.82, df: 0.00010, Cdf: 0.65, Ff: 1.2, Rpf: 0.25)

Quantitative Information:

Fitness = **1156.894**

Design Solution sensitivity = **72.97**

Design Variable sensitivity :: (Cdr: **42**, Fhc: **58**, Tc1: 7, dth: 2, kw: 5, Rp: **202**, Rs: 11, df: 1, Cdf: 8, Ff: 4, Rpf: 33)

Constraints sensitivity:

CONS-1 : Quasi-Active Constraint

CONS-2 : Constraint satisfied

CONS-3 : Constraint satisfied

Qualitative Information:

Qualitative Rating of the solution:

COST OF MANUFACTURE: **19.04**

COST OF MATERIAL: **15.81**

DESIGNER'S SPECIAL PREFERENCE: **57.27**

DESIGN: 7

One of the GOOD solutions is :: (Geom: **2**, Cdr: 0.40, Fhc: 3.0, Tc1: 779, dth: 0.00125, kw: 31, Rp: 1.07, Rs: 0.73, df: 0.00025, Cdf: 0.66, Ff: 1.2, Rpf: 0.21)

Quantitative Information:

Fitness = **1178.030**

Design Solution sensitivity = **73.19**

Design Variable sensitivity :: (Cdr: 28, Fhc: **65**, Tc1: 9, dth: 2, kw: 4, Rp: **203**, Rs: 8, df: 2, Cdf: 9, Ff: 5, Rpf: **40**)

Constraints sensitivity:

CONS-1 : Quasi-Active Constraint

CONS-2 : Constraint satisfied

CONS-3 : Constraint satisfied

Qualitative Information:

Qualitative Rating of the solution

COST OF MANUFACTURE: **19.04**

COST OF MATERIAL: **15.81**

DESIGNER'S SPECIAL PREFERENCE: **59.90**

DESIGN: 8

One of the GOOD solutions is :: (Geom: **2**, Cdr: 0.41, Fhc: 3.0, Tc1: 767, dth: 0.00135, kw: 33, Rp: 1.07, Rs: 0.83, df: 0.00010, Cdf: 0.72, Ff: 1.1, Rpf: 0.26)

Quantitative Information:

Fitness = **1153.228**

Design Solution sensitivity = **72.88**

Design Variable sensitivity :: (Cdr: **42**, Fhc: **57**, Tc1: 7, dth: 2, kw: 5, Rp: **203**, Rs: 11, df: 1, Cdf: 8, Ff: 4, Rpf: 32)

Constraints sensitivity:

CONS-1 : Quasi-Active Constraint

CONS-2 : Constraint satisfied

CONS-3 : Constraint satisfied

Qualitative Information:

Qualitative Rating of the solution:
COST OF MANUFACTURE: **19.04**
COST OF MATERIAL: **15.81**
DESIGNER'S SPECIAL PREFERENCE: **57.35**

DESIGN: 9

One of the GOOD solutions is :: (Geom: **2**, Cdr: 0.41, Fhc: 3.0, Tc1: 767, dth: 0.00135, kw: 33, Rp: 1.09, Rs: 0.83, df: 0.00015, Cdf: 0.65, Ff: 1.5, Rpf: 0.20)

Quantitative Information:

Fitness = **1148.681**
Design Solution sensitivity = **64.01**
Design Variable sensitivity :: (Cdr: **43**, Fhc: **59**, Tc1: 5, dth: 3, kw: 5, Rp: **161**, Rs: 9, df: 1, Cdf: 6, Ff: 2, Rpf: 27)
Constraints sensitivity:
CONS-1 : Quasi-Active Constraint
CONS-2 : Constraint satisfied
CONS-3 : Constraint satisfied

Qualitative Information:

Qualitative Rating of the solution:
COST OF MANUFACTURE: **19.04**
COST OF MATERIAL: **15.81**
DESIGNER'S SPECIAL PREFERENCE: **57.06**

DESIGN: 10

One of the GOOD solutions is :: (Geom: **2**, Cdr: 0.41, Fhc: 3.0, Tc1: 767, dth: 0.00133, kw: 33, Rp: 1.07, Rs: 0.83, df: 0.00010, Cdf: 0.61, Ff: 1.5, Rpf: 0.26)

Quantitative Information:

Fitness = **1153.209**
Design Solution sensitivity = **72.88**
Design Variable sensitivity :: (Cdr: **42**, Fhc: **57**, Tc1: 7, dth: 2, kw: 5, Rp: **203**, Rs: 11, df: 1, Cdf: 8, Ff: 4, Rpf: 32)
Constraints sensitivity:
CONS-1 : Quasi-Active Constraint
CONS-2 : Constraint satisfied
CONS-3 : Constraint satisfied

Qualitative Information:

Qualitative Rating of the solution:
COST OF MANUFACTURE: **19.04**
COST OF MATERIAL: **15.81**
DESIGNER'S SPECIAL PREFERENCE: **57.35**

DESIGN: 11

One of the GOOD solutions is :: (Geom: **3**, Cdr: 0.27, Fhc: 3.2, Tc1: 748, dth: 0.00118, kw: 32, Rp: 1.10, Rs: 0.70, df: 0.00025, Cdf: 0.69, Ff: 1.5, Rpf: 0.21)

Quantitative Information:

Fitness = **1265.239**
Design Solution sensitivity = **64.09**
Design Variable sensitivity :: (Cdr: **58**, Fhc: **62**, Tc1: 6, dth: 4, kw: 5, Rp: **129**, Rs: 10, df: 2, Cdf: 6, Ff: 3, Rpf: 37)
Constraints sensitivity:
CONS-1 : Quasi-Active Constraint
CONS-2 : Constraint satisfied
CONS-3 : Constraint satisfied

Qualitative Information:

Qualitative Rating of the solution:
COST OF MANUFACTURE: **19.04**
COST OF MATERIAL: **15.81**
DESIGNER'S SPECIAL PREFERENCE: **56.23**

DESIGN: 12

One of the GOOD solutions is :: (Geom: **3**, Cdr: 0.26, Fhc: 3.2, Tc1: 728, dth: 0.00118, kw: 32, Rp: 1.10, Rs: 0.71, df: 0.00035, Cdf: 0.60, Ff: 1.3, Rpf: 0.21)

Quantitative Information:

Fitness = **1266.621**

Design Solution sensitivity = **64.18**

Design Variable sensitivity :: (Cdr: **60**, Fhc: **61**, Tc1: 5, dth: 4, kw: 5, Rp: **129**, Rs: 10, df: 2, Cdf: 6, Ff: 3, Rpf: 36)

Constraints sensitivity:

CONS-1 : Quasi-Active Constraint
CONS-2 : Constraint satisfied
CONS-3 : Constraint satisfied

Qualitative Information:

Qualitative Rating of the solution:

COST OF MANUFACTURE: **58.43**
COST OF MATERIAL: **15.81**
DESIGNER'S SPECIAL PREFERENCE: **56.91**

DESIGN: 13

One of the GOOD solutions is :: (Geom: **3**, Cdr: 0.22, Fhc: 3.2, Tc1: 733, dth: 0.00123, kw: 33, Rp: 1.05, Rs: 0.50, df: 0.00035, Cdf: 0.68, Ff: 1.1, Rpf: 0.26)

Quantitative Information:

Fitness = **1336.613**

Design Solution sensitivity = **61.46**

Design Variable sensitivity :: (Cdr: **71**, Fhc: **53**, Tc1: 7, dth: 3, kw: 6, Rp: **132**, Rs: 10, df: 1, Cdf: 4, Ff: 2, Rpf: 30)

Constraints sensitivity:

CONS-1 : Quasi-Active Constraint
CONS-2 : Constraint satisfied
CONS-3 : Constraint satisfied

Qualitative Information:

The design has satisfied Heuristic-1 (cost of manufacture):

Geom: THREE
Cdr: VERY_LOW
dth: MEDIUM
Rs: VERY_LOW
df: BIG

Qualitative Rating of the solution:

COST OF MANUFACTURE: **70.00**
COST OF MATERIAL: **15.81**
DESIGNER'S SPECIAL PREFERENCE: **57.74**

DESIGN: 14

One of the GOOD solutions is :: (Geom: **3**, Cdr: 0.27, Fhc: 3.2, Tc1: 748, dth: 0.00096, kw: 32, Rp: 1.10, Rs: 0.70, df: 0.00025, Cdf: 0.69, Ff: 1.5, Rpf: 0.21)

Quantitative Information:

Fitness = **1265.009**

Design Solution sensitivity = **64.09**

Design Variable sensitivity :: (Cdr: **58**, Fhc: **62**, Tc1: 6, dth: 4, kw: 5, Rp: **129**, Rs: 10, df: 2, Cdf: 6, Ff: 3, Rpf: 37)

Constraints sensitivity:

CONS-1 : Quasi-Active Constraint
CONS-2 : Constraint satisfied
CONS-3 : Constraint satisfied

Qualitative Information:

Qualitative Rating of the solution:

COST OF MANUFACTURE: **19.04**
COST OF MATERIAL: **30.44**
DESIGNER'S SPECIAL PREFERENCE: **52.27**

DESIGN: 15

One of the GOOD solutions is :: (Geom: **3**, Cdr: 0.20, Fhc: 3.2, Tc1: 724, dth: 0.00116, kw: 32, Rp: 1.07, Rs: 0.50, df: 0.00010, Cdf: 0.74, Ff: 1.1, Rpf: 0.21)

Quantitative Information:

Fitness = **1339.368**

Design Solution sensitivity = 0.00
Design Variable sensitivity :: (Cdr: 0, Fhc: 0, Tc1: 0, dth: 0, kw: 0, Rp: 0, Rs: 0, df: 0, Cdf: 0, Ff: 0, Rpf: 0)
Constraints sensitivity: none

Qualitative Information:

Qualitative Rating of the solution:
COST OF MANUFACTURE: **39.47**
COST OF MATERIAL: **15.81**
DESIGNER'S SPECIAL PREFERENCE: **54.70**

7.3 The Evaluation of ASM

The ASM approach is developed for a real life design problem. Presently the system works with the turbine blade cooling system design problem, but the approach is generic and can be applied to other similar applications with small changes. Without prior knowledge concerning the nature of a real life problem, it is virtually impossible to automate the evaluation process for such a system like ASM. Due to the limited resources, ASM is evaluated by one expert (that is an expert designer) and a user (that is a designer). The expert and the user evaluate the system based on their experience, judgement, and personal satisfaction. It is almost impossible to obtain a crisp number representing the degree of success of ASM. The evaluation approach tries to express the overall feeling of the expert and the user using a set of linguistic expressions or statements. The evaluation approach involves two stages: verification of the results and validation of the approach. The next two sections describe the principal issues involved in the verification and the validation processes.

7.3.1 Verification of the Results

The objective for the verification of the results is to check whether the quantitative and qualitative information concerning the design solutions conforms to the understanding of the expert designers. The expert and the designer take part in every stage of the verification. The verification process can be divided into three stages:

a) *Verification of the design solutions*

This stage of the verification first involves manual checking of the mathematical model for the preliminary design of the cooling system (TBCOM). The checking mainly concentrates on the equations derived from the laws of physics. The model also involves certain amount of designers' experience. The expert decides the values of the design parameters. In order to verify whether the design parameter values are representative several design solutions are verified by the expert and the user. They check whether the combination of design variables (the combination represents a design solution) and the fitness correspond to their understanding about the problem.

b) *Verification of Sensitivity Information*

The "good" design solutions identified by the hybrid search method are tested for the additivity principle. The sensitivity values of a design are accepted only if the additivity principle is valid in the neighbourhood of the design. The study presented in chapter 5 shows that in the case where the additivity principle is valid in a neighbourhood, the sensitivity calculations are very close to exhaustive search based results. If the additivity principle is not valid in a neighbourhood the sensitivity values are set to zero. The expert and the user are found to be confident on the sensitivity results provided the additivity principle is valid in the neighbourhoods. They also check whether the design variable sensitivities correspond to the general understanding about the problem. In some cases the information does not correspond to the designer's understanding, this is mainly due to their lack of knowledge concerning the part of the design space. In such cases a through analysis is performed on the results.

c) *Verification of the Qualitative Information*

The qualitative information involves three ratings for the three qualitative criteria: manufacturability, choice of materials, and designer's special preferences. The qualitative information is verified by the expert and the user: first by checking the fuzzy rules for individual criterion. For each design, they check whether the qualitative ratings correspond to their understanding of the problem. If not, the designers perform an investigation to identify any possible reason behind the results. In case the designers are not confident on the results they suggest modifications to the fuzzy rule base.

7.3.2 Validation of the approach adopted in ASM

ASM represents an approach towards design decision support in the preliminary design stage. It is necessary to evaluate the effectiveness of such an approach in engineering design decision making. The validation of ASM has involved group and individual meetings and a final validation by a questionnaire. A thorough understanding about the design environment is necessary to develop the questionnaire. In the context of this thesis, a questionnaire is defined as a set of questions [Bradburn and Sudman (1979)] developed to qualitatively evaluate the different components of ASM and the overall approach. Once the questions are answered the feedback is discussed with the respondents (the expert and the user). Any conflict is resolved through mutual discussion. The discussion developes a statement expressing the qualitative evaluation by consensus.

To develop the questionnaire the nature of the problem, the organisation and the existing work practice are researched using a series of *pilot* interviews [Oppenheim (1992)] with the expert and the user at Rolls Royce. Before the questions are worded the following decisions are made about the questionnaire [Oppenheim (1992)]:

a) *The method of approach to respondents*

The two respondents are selected by Rolls Royce. They have contributed during the development of ASM with their specific technical knowledge. The Rolls Royce expertise is utilised in the model development as well as for the development of the knowledge base for the qualitative evaluation. Preliminary discussions with the expert and the user also contributed towards a broad requirement specification for the design decision support. The expert and the user could easily validate a research prototype of ASM due to their familiarity with the problem. The development of the questionnaire addresses the issue of any possible biases (due to the familiarity) by broadening the aspects covered by the questions. The purposes of the questionnaire and of each module of the questionnaire are explained at the beginning of each section (Appendix - I). The questions are also set to fit in the available time frame in an industrial environment. ASM is a research prototype in the present form. The respondents are requested to evaluate the system from a research prototype point of view. The questionnaire is answered separately by the two respondents. Later the feedback is discussed with them. As the evaluation is performed by only two respondents, the issues of confidentiality and anonymity do not arise very much. The feedback from the questionnaire is used for academic purposes only.

b) *The build up of the question sequence*

The sequence of the questions is very important for the evaluation. The questionnaire is divided into five modules. The first module is about the general issues covering the design environment and the work practice, and the module contains ten questions. The second set of questions is about the design model, and has only four questions. The next module is intended to understand the overall feeling of the respondents concerning the performance of ASM and the results achieved. There are eleven questions in this module. The fourth module asks for any general remarks. The final and the fifth module is optional, and the respondents are asked to assess their personal strengths and weaknesses. The purpose of this module is to gather some information about the respondents so that their comments can be evaluated in right perspective. Every module starts with a broad question, which is then gradually made more specific in subsequent questions. The process is called *funnelling*, which is a standard practice in similar applications. The broad questions at the beginning of a module prepare the ground for the subsequent questions.

c) *The type of question*

The questionnaire uses both the 'closed' or pre-coded answer and 'open' or free-response type of questions. A 'closed' question is one in which the respondents are offered a choice of alternative replies. They are requested to tick the chosen answer(s) in the written questionnaire. The pilot interviews help to develop the 'close' type questions and answers. The type of questions allows less freedom of expression, and thus sometime can be less informative. On the other hand, the questions are easy to answer and the answers can also be compared easily. Often the pre-coded answers can help the respondents to think in the required direction.

'Open' or free-response type questions are not followed by any kind of choice, and the answers have to be written in full. The amount of space left in the questionnaire limits the size of the answers. The main advantage of the 'open' questions is the freedom it gives to the respondents. Once they have understood the intent of the question they can express themselves freely. Thus their ideas and feelings can be recorded in their own language. On the other hand, the answers can also involve some personal emotion and biases. Free-response questions are often easy to ask, difficult to answer, and still more difficult to analyse. The questionnaire developed for ASM evaluation uses some 'open' type questions. The open questions can put forward the true feelings of the respondents. The responses are then discussed in a meeting before any form of conclusive remark is developed.

Many iterations are required to finalise the content and the wordings of the questions for the questionnaire. A number of meetings with the expert and the user help to construct the questions. Once the questionnaire is answered by both the expert and the user, the feedback is discussed with the designers, misunderstandings if any are cleared. The discussion develops a statement to express the feelings of the expert and the user concerning the effectiveness of ASM as the design decision support tool.

7.3.3 Summary of the Feedback using the Questionnaire

A research prototype of ASM is validated in Rolls Royce by the expert and the user. The designers (i.e. the expert and the user) are requested to run the system as they like with different settings for the variables and the constraints. Then they evaluate the results and the effectiveness of ASM as a design decision support tool for the problem. After few weeks of trials with ASM, the designers are asked to provide their feedback individually through the questionnaire (mentioned in section 7.3.2). The expert is directly involved in turbine blade cooling system design activities in Rolls Royce. The other evaluator, the user, is a user of CAD systems and is knowledgeable in artificial intelligence applications in design. It is observed that the responses reflect the backgrounds of the designers. The questionnaire responses are then discussed with the designers.

It is observed that both the designers agree the ASM approach is effective for the design decision support. It is mentioned that there is a scope to expand the preliminary design model to include other design variables considering the overall turbine design and economics. It is expected that the model will grow with experience. The user is found to have some difficulty in utilising the information provided by ASM to select a design. During the discussion, it is identified a user of ASM would require some domain specific knowledge and experience to use the system effectively. The feedback also mentions possible expansion of the fuzzy-rule base to include more cases from past experiences of other experts in the area. In general the expert and the user has expressed their intention to include ASM in their design activities as a decision support tool. It is mentioned that the interface of ASM needs improvement in terms of presentation and flexibility before the system can be used in a production environment. Considering the feedback and after the discussion the following statement is developed as a representative of the designers' perception on the effectiveness of the ASM approach in design decision support:

> *The approach developed in ASM is effective for design decision support, especially in the case of a preliminary design of a turbine blade cooling system. The approach sometimes helps the designer to select a design outside the existing limits of the design problem. Although effective use of the system needs some domain knowledge and experience, ASM reduces the cognitive overload of a designer. Implementation of ASM in a production environment requires improvement of the interface in terms of presentation and flexibility.*

CHAPTER - 8

8. Discussion and Conclusions

8.1 Discussion

The thesis presents the development and application of ASM to the turbine blade cooling system design problem. The attempt uses an adaptive search technique and a knowledge based system to provide relevant information for design decision support. The developed approach in the thesis allows to utilise both quantitative and qualitative information in the design decision making. The hybrid of ARTS based GA search and local hill climbing and the sensitivity analysis provides the quantitative information. The qualitative information for the decision making is provided by the qualitative evaluation of the 'good' designs. During the evaluation of ASM by Rolls Royce experts it is observed that successful utilisation of the quantitative and qualitative information requires some domain specific knowledge and experience. On the other hand, it is also noticed that the information presented by ASM reduces the cognitive overload of a designer. The ASM interface allows the designer to change the existing boundaries of the design problem, and thus ASM sometimes may suggest novel 'good' designs.

The design model, TBCOM is developed for the preliminary design of the cooling system. The model is a coarse representation of the design problem. From the study mentioned in chapter 2 and also the experience from running ASM several times, it is observed that in an unconstrained situation TBCOM is most probably monotonic in nature. The presence of discontinuities is also observed in the unconstrained design space. In a constrained situation the space changes considerably, and becomes quite complex for the search technique. As mentioned in chapter 3, the use of penalty functions also introduces some additional complexity to the design space. It is also observed that the design space involves some nonlinearity (chapter 3). TBCOM is a computational model of the real life design problem. Thus the model does not provide any prior knowledge concerning the quality and location of the sub-optimum peaks. The absence of prior information in TBCOM poses some challenge for an adaptive search technique.

Chapter 4 shows that an ARTS based GA search successfully identifies and maintains all the peaks in the case of the test functions. The search technique distributes the population on the peaks and attains a steady state of distribution. ARTS does not need prior knowledge concerning the modality of the fitness landscape. When compared with RTS and DC, ARTS performed better in terms of maintaining the population on the peaks. It is observed that, ARTS takes a little longer to distribute the population on the peaks than RTS and DC. This can be attributed to the clustering error at the initial stages of a run. Further study on the effect of KT on ARTS shows that a change in KT can delay the attainment of the steady state distribution. A chi-square-like measure test is performed with seven different values of KT for the functions F1 and F2. The test exhibits that, for the function F1 the average measures are quite similar. The corresponding standard deviations are also reasonably small. That means for the function F1 the final population distributions are similar. It is observed that on the function F2, the average and the standard deviation tend to increase with decreasing values of KT, but they are still very low. A hybrid of ARTS and a knowledge based hill climbing is next applied to the cooling system design problem. The design problem is encoded using a structured chromosome. The representation helps to accommodate the discontinuities due to the three types of coolant hole internal geometries.

On the other hand, the chromosome contains a large amount of redundancy. In the case of a large multidimensional problem the redundancy can become a hindrance to the search process. The constraints in TBCOM are implemented using three linear penalty functions. The type of and the values in the penalty functions are selected as a preference after a number of trials with other penalty functions. ARTS produces some duplicate solutions in the population and maintains them. The cluster list is developed after eliminating the duplicate solutions. This helps to reduce the cluster list size as the search progresses and thus reducing the clustering time for every generation. The application of the clustering technique along with the elimination of duplicate solutions helps in avoiding the dominance problem, and thus assist in attaining a steady state of population distribution. The search uses an effective crossover technique that prevents crossover in the redundant regions of the parent chromosomes. The technique improves the effectiveness of the crossover and thus reduces ASM run time. In a further attempt to reduce ASM run time, the parent chromosomes are selected from the cluster list. This avoids selecting duplicate solutions as parents, and thus increases the effectiveness of the reproductive stage of ARTS. A knowledge based hill climbing tries to improve the best design in each cluster for every generation. The search is stopped after it satisfies a steady state criterion. The first steady state criterion is satisfied if the average fitness of the population remains unchanged for 100 generations. It takes longer to satisfy this criterion. Thus a second criterion is defined, that stops the search quicker while achieving satisfactory results. According to this criterion, the search is stopped if it cannot find better sub-optimum solutions for 100 generations. A fixed size list of "good" design is maintained every generation for each geometry. The size of the list is selected purely from individual preference.

The knowledge based hill climbing is developed to exploit information from the clusters produced every generation. The hill climbing is performed on selected dimensions. The dimensions are selected by utilising information from the clusters along with designers' prior knowledge about the problem. The designers' prior knowledge is global in nature, thus not necessarily true in certain local regions. On the other hand, information from clusters involves some uncertainty, because the quality of the information depends on the sampling within a cluster. The two levels of uncertain information are combined using Bayes' theorem. It is expected that the new information is more certain than either one of the two. The information from the clusters is retrieved by a series of univariate linear regression analyses. Thus the search assumes that the region within a cluster can be approximated by a linear and additive model. Depending on the size of a cluster this assumption may become very strong, and thus can reduce the effectiveness of the hill climbing. The method involves deterministic hill climbing, and thus uses a small number of model evaluations. On the other hand, because of the limited search the method cannot guarantee to reach the peak of a hill. KBHC reaches close to a peak with small number of model evaluations.

A stochastic hill climbing tries to fine tune the designs obtained from the hybrid search towards the sub-optima. The technique improves designs but requires many model evaluations. Due to the stochastic nature of the search it cannot be guaranteed that the search would achieve the sub-optima within a reasonable number of model evaluations.

The "good" design solutions are then analysed for sensitivity information. The thesis presents (chapter 5) an approximate sensitivity analysis for the turbine blade problem. Taguchi's methodology based on the orthogonal matrix can provide a maximum amount of information about the neighbourhood of a design solution. The technique is effective if there is no or very little interaction between the design variables. The turbine blade cooling system assumed no interaction between the design variables (unconstrained), so the technique is expected to be effective for the application. The sensitivity analysis only considers the tolerance space around a design solution, so it is more probable that the small region can be approximated using an additive model. The methodology to check the validity of the additivity principle in the tolerance space of a design solution adds more confidence to the results. Figures 5.6-5.8 confirm the hypothesis that Taguchi's methodology is very effective (comparable to an exhaustive search based method) if there is no interaction or very little interaction among the design variables. The figures also show within one geometry some design solutions can have very high design solution sensitivities with respect to others. The analysis is performed within the neighbourhood of a design. The basis of the sensitivity calculations, the orthogonal matrix, requires the examination of only a small number of design solutions. In this case only 27 model evaluations are required rather than all 3^{11} possible evaluations (that is in this case of 11 design variables with 3 levels each). Though in this case the technique is very reliable, the sensitivity analysis results should only be used to compare two design solutions rather than as absolute sensitivity values. The use of the signal-to-noise ratio to calculate the design solution sensitivity index is a measure of the robustness of the solution within its neighbourhood. The information concerning individual design variable sensitivity is also very useful for engineering design decision support. The design variable sensitivity determines criticality of the different variables in the tolerance space of the design solution. A designer often selects design solutions that satisfy constraints. But that may not be enough, the criticality of constraints in its neighbourhood also plays a major role in the decision making. The constraint sensitivity provides an overall idea concerning the constraint violations in the neighbourhood of a design.

ASM also retrieves qualitative information concerning the designs. Each design is evaluated to obtain qualitative ratings for three criteria: manufacturability, choice of material and designer's special preferences. The criteria were selected by the expert designer from Rolls Royce. Any other similar criteria can be easily included. The results presented in the previous chapter exhibit that the design fitness (quantitative) and the qualitative ratings are independent of each other. Once again, the qualitative ratings are suitable for comparison between two designs. The knowledge separation and then the knowledge integration using the concept of compromise guarantee that ASM can evaluate any design solution from the entire design space. ASM uses a reasonably small number of fuzzy rules. On the other hand, the knowledge representation technique is restricted in representing any inter variable interaction. Some definitive knowledge about interactions can be incorporated as heuristics. A more generalised structure of knowledge representation would be necessary for highly interactive design problems. Another limitation of ASM is that the fuzzy expert system does not have any explanation facility. It would be very useful for the designers to know why certain decisions are made. It is noticed that any fuzzy expert system presently lacks this capability. This is still an open area for further research. The crisp rating obtained for a criterion is not the best way of representing qualitative information about any design solution. Use of appropriate linguistic terms (that is using some linguistic approximation method or similar approach) to express the information can be more effective for the designers. This particular area needs further investigation.

The results presented in chapter 7 are the representative of the results obtained from the ASM. During experimentation it is observed that, the best solution identified by ARTS and the knowledge based hill climber hybrid system is always at least equal or better than a simple GA application on the problem. The new technique identifies multiple "good" design solutions from TBCOM design space. The search does not require prior knowledge concerning the modality of the design space. The "good" solutions can be further screened by setting up a threshold on the fitness of the solutions. Section 7.2.1 presents results from unconstrained search. The fifteen designs as suggested by ASM are "good" designs and are from different regions in the design space. The design solution sensitivity is similar in all the designs. This corresponds to the fact that the unconstrained design space is very likely to be monotonic. It is observed that Rp is the most sensitive variable in the neighbourhoods of the designs. The other sensitive variables are: Rpf, Fhc and Cdr. Some variations in the qualitative ratings are also observed. The designer considers all the information to select one design from the list. The constrained search (results are presented in section 7.2.2) is more complex: more generations are required to attain the steady state. According to the expert and the user the designs identified by ASM are "good" and representative of the constrained design space. The designs are from different positions in the space, and thus provide several design options. The fitness (that is the inverse of the coolant mass flow through the radial passage) of designs from plane type geometry varies considerably from that of pedestal or ribbed type geometries. The designs from the pedestal type geometry (Geom: 3) have the highest fitness. Within the plane geometry the design solution sensitivity varies from 55.26 to 81.07. That shows although the designs can be "good" in terms of coolant mass flow criterion (i.e. less coolant flow), performance can differ considerably in terms of the sensitivity. Less sensitive designs are preferred by the designers. The most sensitive variables for plane type geometry are: Rp, Fhc, Rpf and Cdr. The designs from the plane type geometry have the same qualitative rating for the cost of material. Some differences are observed in the ratings for cost of manufacture and designer's special preference. The designs from the second type of geometry, the ribbed type, are similar in fitness but the design solution sensitivity varies from 64.01 to 73.19. The major contributing variables are again Rp, Fhc, Cdr and Rpf. The qualitative ratings are found to be similar. So the designer mainly uses the sensitivity information to compare designs. The designs are generally less sensitive than that of the plane geometry type. These designs are less suitable than the plane geometry designs in terms of cost of manufacture and cost of material. The first constraint on the metal temperature seems very hard and it is violated in the neighbourhoods of the designs. Fitnesses of the designs with the pedestal type geometry have higher fitness than the others. The design solution sensitivity varies from 61.46 to 64.18, i.e. less sensitive than the designs with plane or ribbed type of geometry. Once again Rp, Cdr, Fhc and Rpf are the most sensitive design variables. The first constraint is also violated in the neighbourhoods of the designs. It is observed that the ratings for the qualitative criteria largely vary among the designs. Thus the qualitative information can play a significant role to compare between two designs. The neighbourhood of the fifteenth design violates the additivity principle, thus the sensitivity calculation is not performed and the values are set to zero. That means there is a significant interaction between the design variables within the neighbourhood. Designers are often not interested in a design from a highly interactive region of the design space. The mix of quantitative and qualitative information provides support to the designers for the design decision making.

ASM works with a real life problem, where there is less prior information concerning the nature of the problem. Also the evaluation of ASM should represent the *feelings* of the designers rather than definitive conclusions. Thus, during the evaluation of ASM an expert and a user are requested to verify and validate different components of ASM. A questionnaire helps the designers (i.e. the expert and the user) to validate the system. The use of many open type questions helps the designers to express themselves better. Due to limited available human resources ASM is evaluated only by an expert and a user. This number is very small for any evaluation procedure. In order to minimise the effect of this small number of evaluators, the feedback from the questionnaire is discussed with the designers. An effort is made to obtain a consensus on the statement that expresses the views of the designers in terms of the effectiveness of ASM as a design decision support approach.

8.2 Conclusions

The feedback from the expert and the user suggests that the approach developed in ASM can successfully support the design decision making in the preliminary design stage. Thus the approach is effective for the real life multiobjective design problem. The approach developed in ASM has provided a methodology to utilise both quantitative and qualitative information in engineering design decision making. The approach can reduce the cognitive overload of a designer. The final design decision is taken by the designer, and thus ASM also provides the opportunity to utilise the value system of the designer in the design process. There are three main components of ASM: the ARTS based GA and hill climbing hybrid search technique, sensitivity analysis using Taguchi's methodology and qualitative evaluation using fuzzy expert system. The ARTS based GA and the knowledge based hill climbing hybrid has added another tool to the list of multimodal GAs. The objective of the search is to maintain peaks in the final population. The hybrid search method is suitable for real life problems. The research presented in the thesis highlights some of the characteristics of real life optimisation problems. The study has enhanced the understanding concerning the issues involved in such optimisation. The limitation of the search technique is that there is no guarantee that the search has visited all sub-optima in the search space. This can sometimes severely damage the confidence of the designer in the decision support system. The sensitivity analysis uses the well-established Taguchi's methodology. The application is novel and can be very useful for multidimensional real life problems. The limitation of the application is that it assumes the neighbourhoods can be approximated by an additive model. That may not be the case in many applications. The qualitative evaluation of designs is performed using a fuzzy expert system. The fuzzy expert system utilises knowledge from experts in the field. The adaptive search technique (i.e. the hybrid search) and the sensitivity analysis modules are integrated with this fuzzy expert system to develop the integrated ASM system. The knowledge representation technique developed in this thesis has made the integration possible. The technique represents the design thinking process, and guarantees the evaluation of any possible design. Due to the novel knowledge representation technique, the qualitative evaluation module requires a small number of fuzzy rules. The major limitation of the module is that it cannot fully address the interaction between design variables. Some definitive knowledge about the interaction can be represented as heuristics. A generalised knowledge representation that can handle at least a limited amount of design variable interaction (i.e. interaction among a small number of variables) is required for industrial problems. Lack of an explanation facility is another limitation of the evaluation system. The evaluation approach adopted for ASM is suitable for real life problems. It is observed that in the industrial environment, and especially in life critical and sensitive industries, it is important that the final validation is performed by human experts. Instead of a crisp rating for the effectiveness of ASM, a statement is developed that expresses the feelings of the designers. It is observed that such approach is more acceptable and effective in evaluating a real life design decision support system.

8.3 Future Research Directions

The research reported in this thesis has also contributed to open new issues for research. This section provides an outline to possible future research directions. There are several aspects of the design decision support that needs further investigation. ASM uses a very simple interface, and thus a major investigation is required to develop a suitable interface for the decision support. The questions of human computer interaction need to be addressed in that research.

One of the major issues that decide the acceptability of a design decision support system is the confidence of the designers in the system. There is a need for better understanding of the causes that can increase the confidence of the designer. The decision support system should address the issues to enhance designers' confidence. One example can be developing a search algorithm that can produce results with confidence within an acceptable time limit. The penalty function, when used for the constrained optimisation, modifies the design fitness landscape. A multimodal GA algorithm that handles constraints without penalty functions would be very useful.

The knowledge representation technique developed in the thesis can only address very limited interaction between design variables (i.e. using heuristic rules). A further investigation is necessary to develop a more generalised knowledge representation that can efficiently handle interaction between the variables. One way of representing such interaction is to use meta rules. It is observed that the designers face difficulty while expressing their knowledge concerning the interactions. Research is necessary to extract the interaction information also from other sources, such as past designs, physical modelling, etc. Further development is required in fuzzy expert system research to develop the explanation facility. The results from the qualitative evaluation should ideally be expressed using linguistic expressions, but the expression needs to be short enough so that the designer can comprehend the meaning. Further research in this area should develop approximation techniques that can produce short but representative expressions.

APPENDIX - I

VALIDATION OF
THE ADAPTIVE SEARCH MANAGER (ASM)

Questionnaire

Category of the Software: *Research Prototype*

Version of the ASM: ....

Dated: .... Time: ....

Serial Number: ....

Name: ....

Organisation: ....

Address: ....

....

....

Contact Telephone Number: ....

Contact FAX Number: ....

Email: ....

The information provided will only be used for academic and research purposes. If you agree please tick the box: ☐

Validation conducted by:

....

PURPOSE OF THE VALIDATION:

The Adaptive Search Manager has been developed to solve real life design problems. ASM is a design decision support tool suitable in the preliminary design stage for turbine blade cooling systems. Validation of the tool is essential to assess its effectiveness and to provide feed back for further development. ASM needs to be validated as a research prototype. The questionnaire provides a structured procedure for the validation. If you do not want to answer a question from the choice of answers given in the questionnaire please circle 'other' and answer in your own words.

MODULE 1

General issues

This section of the questionnaire tries to understand the general design practice involved in your company, the work environment and your opinions concerning some general recent issues in design activities. Please try to answer the questions considering your own experience by circling only one answer (or more than one answers if specified in the question):

Q. 1: Could you please briefly describe the nature of your involvement in the design projects of the company?

A. 1:

Q. 2: How much time (in days) in average do you spend on design related activities per week?

A. 2:

Q. 3: When you work in design related activities, which one of the following closely describes the type of environment you work with?

A. 3:

a. performing a design task on your own
b. performing a design sub-task as part of a small group of 4-5 people.
c. performing a small part of an overall design task as a member of a design team, where there is one team leader.
d. other, please specify:

Q. 4: Would it be possible to categorise the general nature of your design activities?

A. 4:

a. detailed design
b. preliminary design
c. creative or innovative design
d. design analysis
e. design evaluation
f. design activity management and co-ordination
g. developing tools that can be useful in design activities
h. no, it can not be categorised, because:

----

i. other, please specify:

Q. 5: In your day to day design activities which one of the following methods do you normally follow?

A. 5:

a. doing routine designs using previous designs from the archive
b. designing fresh from the first principles of physics involved in the design problem
c. developing a design specification (that defines the task) first and then carrying out step-by-step procedures for the design
d. developing a design specification and then distributing the task among the group members
e. perform any design analysis task and pass the results to your group leader
f. evaluate a design and give feedback directly to the designer
g. other, please specify:

Q. 6: How often in average you are given a new (that is when you have to start from the first principle) design task?

A. 6:

a: once a month
b. once in three months
d. once in six months
e. once a year
f. never
g. that is totally random
h. other, please specify:

Q. 7: What are the different tools you use for your day to day design activities? Please circle more than one answer if you wish.

A. 7:

a. drafting board and pencil
b. pen and pencil for calculations and free hand drawings
c. digitiser
d. computer aided drafting package
e. computer aided design and analysis package
f. simple spreadsheet for calculations
g. project management software
h. other, please specify:

Q. 8: How would you generally improve the design that you are working on?

A. 8:

a: blind trial and error
b. many iterations of educated guesses using previous knowledge
c. using a conventional optimisation algorithm
d. using expert opinion
f. using any optimisation package available in computer integrated design tools
g. other, please specify:

Q. 9: In your day to day design activities which of the following would you consider to be useful or can be useful for the activities. Please circle more than one answer if you wish.

A. 9:

a. design handbooks and different component catalogues
b. a computer database with component details
c. a computer system that advises you as an expert
d. a computer system that advises you of different possible solutions to a problem
e. guidance of an expert in the area
f. a novice designer, who can provide some fresh ideas
g. discussion with a small group of colleagues
h. a computer system that can provide relevant information concerning several possible design solutions
h. discussion with a designer from a rival company
i. none of the above
j other, please specify:

Q. 10: Some companies have recently started to use computer systems that are expected to assist their designers in their design decision making. The outcome of the implementation *in terms of design improvements or cost saving is not very clear yet. Considering a future, more competitive market, do you think such a system should be implemented in your company? Please describe your opinion and reasons behind it:*

A. 10:

MODULE 2

The design model

This section is more specific. Here we discuss several issues involved in the turbine blade cooling system design model development. The adaptive search manager, a computer system that assists in design decision making, uses a preliminary design model that is a good mathematical approximation of the gas turbine blade cooling system. The model is developed considering one dimensional and single pass coolant flow. The model includes film cooling mechanisms and is limited to twelve design variables. The design model also uses several constant design parameters (some of them are determined from domain experience). There are four outputs from the model. The principle task is to minimise the coolant mass flow through the radial hole passage of a blade (that is one of the outputs). The other three outputs constrain the design process. Please look at the designs achieved from several runs of ASM and answer the following questions by circling only one answer (or more than one answer if specified in the question):

Q. 11: Though the model is developed to represent the design problem, it needs to be validated. Could you please comment on whether the results achieved from ASM runs correspond to your engineering understanding about the design problem?

A. 11:

Q. 12: One of the issues in developing an engineering design decision support tool is flexibility. That is how easy is it to adapt the tool to search in different areas of the design space? If you wanted to search different areas of the design space by setting different ranges for the design variables, do you think you can do that easily with ASM? Please give your comments.

A. 12:

Q. 13: While experimenting with ASM in different regions of the design space, have you observed any infeasible solution in the results?

A. 13:

a. YES

please give the reason(s) why you think the design is infeasible:

----

b. NO

Q. 14. A design process may be constrained by some criteria. The developed design model is constrained by three output variables. The constraints are implemented by setting up a range on each of these variables. If the outputs of a model evaluation goes beyond any one of those ranges the design is considered to have violated the constraint. Changing the ranges for the constraints may help to achieve different design goals. During your experiments with ASM did you make changes with the ranges for the constraints?

A. 14:

a. YES

please specify the reason for your changes:

----

please give your comment(s) on the ease of changing the ranges:

----

b. NO

MODULE 3

Performance of the ASM and the results achieved

This section of the questionnaire deals with general issues involved in ASM. The questions are developed to understand the effectiveness of ASM as a design decision support tool. It is important that users of ASM gain a good understanding of the performance of the system through extensive use. The ASM is a research prototype, so it should be assessed according to its merit in terms of the techniques developed, and the quality of the results. Please run

ASM several times with different random number seeds, examine the results and then answer the following questions:

Q. 15: How do you think a computer based decision support system can help you in design? Please categorise major areas of help that can be provided by such tool.

A. 15:

Q. 16: One way a design decision support tool can be useful is to identify the best design in the entire design space. Is that approach acceptable for your design practice?

A. 16:

a. YES

b. NO

please explain the approach you think is more useful to you instead:

Q. 17: Often when you are designing a product, design lead time is very important in terms of design cost. That is why computer based design tools are being introduced to reduce the lead time. The success of ASM depends on how it fits within time constraints of the designers. In your opinion, how does the run time necessary for ASM fit into the overall time constraint for the turbine blade cooling system design?

A. 17:

Q. 18: ASM has been developed to identify multiple 'good' solutions in terms of coolant mass flow. It is important to validate ASM in terms of this capability. Is ASM capable of identifying meaningfully different 'good' design solutions? Please give your opinion based on observations and personal judgement.

A. 18:

Q. 19: Assume you are given the task of designing the cooling system for a turbine blade. It is general practice to look for designs with a minimum amount of coolant mass flow. Is that information enough for you to decide the values of different design variables involved in the design process (that is selecting a design)?

A. 19:

a. YES

b. NO

please mention the other categories of information you would like to have in order to make the right selection:

Q. 20: ASM is a research prototype, and thus its successful development to a complete decision support tool very much depends on your feed back. ASM identifies several 'good' design solutions and then calculates sensitivity information for each design. A qualitative rating is also obtained in order to assess how a design qualifies with respect to certain qualitative criteria. Considering your observations and expectations, could you please answer the following questions:

Q. 20(a): What is your opinion regarding the usefulness of the extra information in helping you to design the cooling system?

A. 20(a):

Q. 20(b): The sensitivity information for a design describes three categories of sensitivity information: design solution sensitivity, design variable sensitivity and constraint sensitivity. From your experience in design, what is your general opinion regarding the utility of the three categories of sensitivity information for the design task?

A. 20(b):

Q. 20(c): Do you think the qualitative ratings for a design conform to your understanding about the details and functionalities of the design?

A. 20(c):

a. YES
b. NO
c. I am not sure
d. other, please describe your opinion:

Q. 21: It is possible that you are already using some computer based design tools or tools that help you in design. If you wanted to enhance your capability in terms of design flexibility and efficiency, would you consider using a system such as ASM in addition to your existing methods?

A. 21:

a. Yes, I would like to add ASM to my tool kit
b. I would prefer to use ASM instead of some of my present tools.
Please specify the tools you would like to replace:

----

c. No.
Please specify the reason:

----

d. other, please specify:

Q. 22: In case you decide to include the use of ASM in your regular design practice (otherwise please mention N/A in the answer), how easily you think you can integrate the system with your existing practices? Please write a couple of sentences to describe your view.

A. 22:

Q. 23: Please consider any one set of results, and select the best design. If you can not select one from the information presented or you are not happy about the quality of the designs then please circle NO, otherwise YES. Please attach the results.

A. 23:

a. YES

Please mention your choice: -----------------------------------

Please explain the reason for the selection:

----

b. NO

Please explain the reason:

Q. 24: Please mention your confidence level on ASM as a rating between 0 and 100. A confidence level of 100 means absolute confidence.

A. 24:

Q. 25: If you are requested to validate future versions of ASM would you like to take part in the validation?

A. 25:

a. YES
b. NO
c. I am not sure at the moment
d. other, please specify:

MODULE 4

General Remarks

Please write any general remarks you wish to make, and mention if you have any suggestion for further development of the system. Also describe the aspect(s) of ASM you like and/or dislike the most.

MODULE 5

Self assessment of the users

This section of the questionnaire is optional. The sole purpose of this module is to gather some information about you so that your comments may be evaluated in the right perspective. In case you do not feel comfortable in answering any part of this module please ignore it. If you are happy to answer a question, please tick the appropriate box.

Q. 1: Design of turbine blade cooling system:

A. 1:

	Best				Worst
Knowledge	1	2	3	4	5
Experience	1	2	3	4	5

Q. 2: Assessment of prototype system:

A. 2:

	Best				Worst
Knowledge	1	2	3	4	5
Experience	1	2	3	4	5

Q. 3: Use of computers for design decision support:

A. 3:

	Best				Worst
Knowledge	1	2	3	4	5
Experience	1	2	3	4	5

Q. 4: Use of CAD systems:

A. 4:

	Best				Worst
Knowledge	1	2	3	4	5
Experience	1	2	3	4	5

Q. 5: Artificial Intelligence techniques:

A. 5:

	Best				Worst
Knowledge	1	2	3	4	5
Experience	1	2	3	4	5

<u>many thanks for your contribution</u>

APPENDIX - II

FUZZY RULES USED IN THE ADAPTIVE SEARCH MANAGER

```
;; ASM_know.clp ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;
;       ADAPTIVE SEARCH AND THE PRELIMINARY DESIGN OF
;               GAS TURBINE BLADE COOLING SYSTEM
;
;       This is the main rule base for the ADAPTIVE SEARCH MANAGER
;
;               This is  modified rule base after VALIDATION
;
;               This file is used with FuzzyCLIPS 6.02A
;
;                               Rajkumar Roy
;                       Plymouth Engineering design Centre
;                           University of Plymouth
;                           Plymouth,  PL4 8AA, UK
;
;                       Tel. : +44 (0)1752 233508
;                       FAX. : +44 (0)1752 233505
;                       Email : rroy@plymouth.ac.uk  or
;                               r.roy@ieee.org
;
;               DIRECTOR OF STUDIES : DR. IAN PARMEE, PEDC
;               INDUSTRIAL COLLABORATOR : ROLLS ROYCE PLC.
;
;;; Rajkumar Roy (c) 1996, Uni Plym ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

                        ;; Global variables ;;
( defglobal

        ?*cluster_number* = 0     ;;Global variable to store no of clusters
)


;
;; Deffunction PIfuzzify ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;
;; Inputs are : ?fztemplate  - name of a fuzzy deftemplate
;;              ?delta              - precision of the value
;;              ?value              - float value to be fuzzified
;;              ?cf                 - confidence factor of the newly asserted fuzzy fact
;;
;;      Asserts a fuzzy fact for the fuzzy deftemplate. The fuzzy set is
;;      a standard PI (as defined in FuzzyCLIPS) type (which is almost like
;;      a normal distribution) centered on the value provided with zero
;;      possibility at value+delta and value-delta. Note that it checks bounds
;;      of the universe of discourse to generate a fuzzy set that does not
;;      have values outside of the universe range.
;;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

(deffunction PIfuzzify (?fztemplate ?delta ?value ?cf)
        (bind ?low (get-u-from ?fztemplate))
        (bind ?high (get-u-to ?fztemplate))
        (if (< ?value (+ ?low ?delta))
```

```
                then
                 (if (<= ?value ?low)
                   then
                        (assert-string
                         (format nil "(%s (Z %g %g)) CF %g" ?fztemplate ?low (+ ?low ?delta) ?cf))
                 else
                     (assert-string
                         (format nil "(%s (Z %g %g)) CF %g" ?fztemplate ?value (+ ?value ?delta) ?cf)))
        else
                (if (> ?value (- ?high ?delta))
                        then
                         (if (>= ?value ?high)
                          then
                                (assert-string
                                 (format nil "(%s (S %g %g)) CF %g" ?fztemplate (- ?high ?delta)
?high ?cf))
                         else
                                (assert-string
                                 (format nil "(%s (S %g %g)) CF %g" ?fztemplate (- ?high ?delta)
?value ?cf)))
                else
                        (assert-string
                                (format nil "(%s (PI %g %g)) CF %g" ?fztemplate ?delta ?value ?cf))
                )
        )
)

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;
; This is the section to define QUALITATIVE RATING (QR) of selected designs
;
;       QR-1 : Manufacturability
;       QR-2 : Choice of material
;       QR-3 : Designer's special preference
;
;       Design variables are defined according to domain knowledge
;
;       A set of rules are defined to calculate the effectiveness
;
;               This forms a part of STATIC MEMORY
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;


;;;;;;;;;;;;;;; defining rules for qualitative assessment ;;;;;;;;;;;;;;;;;;

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;
;       Top level rules showing preferences about each variable.
;
;               Total knowledge has been represented in a
;               hierarchical manner to help interpolation
;               or extrapolation with few rules. Heuristics
;               has also been added. The three types of
;               knowledge as represented :
;                - preferences about each variable (deffacts of preferences)
;                - intra variable preferences (rules)
;                - heuristics (rules)
;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

                        ;;;;;; inter variable preferences ;;;;;;
```

```
(deffunction Inter_Var_Preferences ()

            ;;MANUFACTURABILITY ;;

        (assert (Geom_pref-1 (0.8 0) (0.8 1) (0.8 0)) CF 0.8)
        (assert (Cdr_pref-1 (0.3 0) (0.3 1) (0.3 0)) CF 0.8)
        (assert (Fhc_pref-1 (0.3 0) (0.3 1) (0.3 0)) CF 0.6)
        (assert (Tc1_pref-1 (0.3 0) (0.3 1) (0.3 0)) CF 0.6)
        (assert (dth_pref-1 (0.8 0) (0.8 1) (0.8 0)) CF 0.8)
        (assert (kw_pref-1 (0.4 0) (0.4 1) (0.4 0)) CF 0.9)
        (assert (Rp_pref-1 (0.3 0) (0.3 1) (0.3 0)) CF 0.8)
        (assert (Rs_pref-1 (0.3 0) (0.3 1) (0.3 0)) CF 0.6)
        (assert (df_pref-1 (0.8 0) (0.8 1) (0.8 0)) CF 0.8)
        (assert (Cdf_pref-1 (0.4 0) (0.4 1) (0.4 0)) CF 0.6)
        (assert (Ff_pref-1 (0.4 0) (0.4 1) (0.4 0)) CF 0.6)
        (assert (Rpf_pref-1 (0.3 0) (0.3 1) (0.3 0)) CF 0.8)

            ;; CHOICE OF MATERIAL ;;

        (assert (Geom_pref-2 (0.5 0) (0.5 1) (0.5 0)) CF 0.8)
        (assert (Cdr_pref-2 (0.3 0) (0.3 1) (0.3 0)) CF 0.8)
        (assert (Fhc_pref-2 (0.3 0) (0.3 1) (0.3 0)) CF 0.6)
        (assert (Tc1_pref-2 (0.3 0) (0.3 1) (0.3 0)) CF 0.8)
        (assert (dth_pref-2 (0.9 0) (0.9 1) (0.9 0)) CF 0.9)
        (assert (kw_pref-2 (0.7 0) (0.7 1) (0.7 0)) CF 0.8)
        (assert (Rp_pref-2 (0.5 0) (0.5 1) (0.5 0)) CF 0.8)
        (assert (Rs_pref-2 (0.3 0) (0.3 1) (0.3 0)) CF 0.9)
        (assert (df_pref-2 (0.3 0) (0.3 1) (0.3 0)) CF 0.8)
        (assert (Cdf_pref-2 (0.4 0) (0.4 1) (0.4 0)) CF 0.6)
        (assert (Ff_pref-2 (0.4 0) (0.4 1) (0.4 0)) CF 0.6)
        (assert (Rpf_pref-2 (0.5 0) (0.5 1) (0.5 0)) CF 0.8)

            ;; DESIGNER'S SPECIAL PREFERENCE ;;

        (assert (Geom_pref-3 (0.8 0) (0.8 1) (0.8 0)) CF 0.8)
        (assert (Cdr_pref-3 (0.5 0) (0.5 1) (0.5 0)) CF 0.7)
        (assert (Fhc_pref-3 (0.8 0) (0.8 1) (0.8 0)) CF 0.8)
        (assert (Tc1_pref-3 (0.7 0) (0.7 1) (0.7 0)) CF 0.7)
        (assert (dth_pref-3 (0.9 0) (0.9 1) (0.9 0)) CF 0.9)
        (assert (kw_pref-3 (0.8 0) (0.8 1) (0.8 0)) CF 0.8)
        (assert (Rp_pref-3 (0.6 0) (0.6 1) (0.6 0)) CF 0.8)
        (assert (Rs_pref-3 (0.7 0) (0.7 1) (0.7 0)) CF 0.8)
        (assert (df_pref-3 (0.9 0) (0.9 1) (0.9 0)) CF 0.8)
        (assert (Cdf_pref-3 (0.5 0) (0.5 1) (0.5 0)) CF 0.7)
        (assert (Ff_pref-3 (0.8 0) (0.8 1) (0.8 0)) CF 0.8)
        (assert (Rpf_pref-3 (0.6 0) (0.6 1) (0.6 0)) CF 0.8)


)

                        ;;;;;;; intra variable preferences ;;;;;;;

                        ;;;;;;;;;;;;;;; Geom ;;;;;;;;;;;;;;;;;;

        ;;;;;; MANUFACTURABILITY criteria ;;;;;;;

(defrule rule-1
        (declare (salience -10) (CF 0.8))
        (Geom ONE)
        ?fa <- (Geom_pref-1 ?)
        ;;?eff <- (QR-1 ?)
```

```
        =>

        (bind ?cf (get-cf ?fa))
        ;;(plot-fuzzy-value t + nil nil ?eff)
        (assert (QR-1 VERY_GOOD) CF 1.0)
        (assert (HEURISTICS-1 NO))
)

(defrule rule-2
        (declare (salience -10) (CF 0.8))
        (HEURISTICS-1 NO)
        (Geom not ONE)
        ?fa <- (Geom_pref-1 ?)
        ;;?eff <- (QR-1 ?)

        =>

        (bind ?cf (get-cf ?fa))
        (bind ?pref (get-fs-x ?fa 0))
        ;;(plot-fuzzy-value t + nil nil ?eff)

        (if (> ?pref 0.9)
                then
                (assert (QR-1 BAD) CF ?cf))

        (if (and (<= ?pref 0.9) (> ?pref 0.8))
                then
                (assert (QR-1 slightly_compromise BAD) CF ?cf))

        (if (and (<= ?pref 0.8) (> ?pref 0.6))
                then
                (assert (QR-1 less_compromise BAD) CF ?cf))

        (if (and (<= ?pref 0.6) (> ?pref 0.4))
                then
                (assert (QR-1 compromise BAD) CF ?cf))

        (if (<= ?pref 0.4)
                then
                (assert (QR-1 more_compromise BAD) CF ?cf))

)

        ;;;;;; CHOICE OF MATERIAL criteria ;;;;;;

(defrule rule-3
        (declare (salience -20) (CF 0.6))
        (Geom ONE)
        ?fa <- (Geom_pref-2 ?)
        ;;?eff <- (QR-2 ?)

        =>

        (bind ?cf (get-cf ?fa))
        ;;(plot-fuzzy-value t + nil nil ?eff)
        (assert (QR-2 VERY_GOOD) CF 1.0)
        (assert (HEURISTICS-2 NO))
)

(defrule rule-4
        (declare (salience -20) (CF 0.6))
        (HEURISTICS-2 NO)
```

```
        (Geom not ONE)
        ?fa <- (Geom_pref-2 ?)
        ;;?eff <- (QR-2 ?)

        =>

        (bind ?cf (get-cf ?fa))
        (bind ?pref (get-fs-x ?fa 0))
        ;;(plot-fuzzy-value t + nil nil ?eff)

        (if (> ?pref 0.9)
                then
                (assert (QR-2 NOT_VERY_GOOD) CF ?cf))

        (if (and (<= ?pref 0.9) (> ?pref 0.8))
                then
                (assert (QR-2 slightly_compromise NOT_VERY_GOOD) CF ?cf))

        (if (and (<= ?pref 0.8) (> ?pref 0.6))
                then
                (assert (QR-2 less_compromise NOT_VERY_GOOD) CF ?cf))

        (if (and (<= ?pref 0.6) (> ?pref 0.4))
                then
                (assert (QR-2 compromise NOT_VERY_GOOD) CF ?cf))

        (if (<= ?pref 0.4)
                then
                (assert (QR-2 more_compromise NOT_VERY_GOOD) CF ?cf))

)

        ;;;;;; SPECIAL PREFERENCES criteria ;;;;;;

                ;; None ;;

                        ;;;;;;;;;;;;;;;; Cdr ;;;;;;;;;;;;;;;;;;;


        ;;;;;;;;; MANUFACTURABILITY criteria ;;;;;;;;;

(defrule rule-5
        (declare (salience -30) (CF 0.8))
        (HEURISTICS-1 NO)
        (Cdr VERY_HIGH)
        ?fa <- (Cdr_pref-1 ?)
        ;;?eff <- (QR-1 ?)

        =>

        (bind ?cf (get-cf ?fa))
        ;;(plot-fuzzy-value t + nil nil ?eff)
        (assert (QR-1 GOOD) CF 1.0)
)

(defrule rule-6
        (declare (salience -30) (CF 0.8))
        (HEURISTICS-1 NO)
        (Cdr not VERY_HIGH)
        ?fa <- (Cdr_pref-1 ?)
        ;;?eff <- (QR-1 ?)
```

```
        =>

        (bind ?cf (get-cf ?fa))
        (bind ?pref (get-fs-x ?fa 0))
        ;;(plot-fuzzy-value t + nil nil ?eff)

        (if (> ?pref 0.9)
                then
                (assert (QR-1 BAD) CF ?cf))

        (if (and (<= ?pref 0.9) (> ?pref 0.8))
                then
                (assert (QR-1 slightly_compromise BAD) CF ?cf))

        (if (and (<= ?pref 0.8) (> ?pref 0.6))
                then
                (assert (QR-1 less_compromise BAD) CF ?cf))

        (if (and (<= ?pref 0.6) (> ?pref 0.4))
                then
                (assert (QR-1 compromise BAD) CF ?cf))

        (if (<= ?pref 0.4)
                then
                (assert (QR-1 more_compromise BAD) CF ?cf))

)

        ;;;;;;;;; CHOICE OF MATERIAL criteria ;;;;;;;;;

                ;; NONE ;;

        ;;;;;;;;; SPECIAL PREFERENCES criteria ;;;;;;;;;

(defrule rule-7
        (declare (salience -40) (CF 0.9))
        (HEURISTICS-3 NO)
        (Cdr VERY_LOW)
        ?fa <- (Cdr_pref-3 ?)
        ;;?eff <- (QR-3 ?)

        =>

        (bind ?cf (get-cf ?fa))
        ;;(plot-fuzzy-value t + nil nil ?eff)
        (assert (QR-3 VERY_GOOD) CF 1.0)
        (assert (HEURISTICS-3 NO))
)

(defrule rule-8
        (declare (salience -40) (CF 0.9))
        (HEURISTICS-3 NO)
        (Cdr not VERY_LOW)
        ?fa <- (Cdr_pref-3 ?)
        ;;?eff <- (QR-3 ?)

        =>

        (bind ?cf (get-cf ?fa))
        (bind ?pref (get-fs-x ?fa 0))
        ;;(plot-fuzzy-value t + nil nil ?eff)
```

```
        (if (> ?pref 0.9)
                then
                (assert (QR-3 NOT_VERY_GOOD) CF ?cf))

        (if (and (<= ?pref 0.9) (> ?pref 0.8))
                then
                (assert (QR-3 slightly_compromise NOT_VERY_GOOD) CF ?cf))

        (if (and (<= ?pref 0.8) (> ?pref 0.6))
                then
                (assert (QR-3 less_compromise NOT_VERY_GOOD) CF ?cf))

        (if (and (<= ?pref 0.6) (> ?pref 0.4))
                then
                (assert (QR-3 compromise NOT_VERY_GOOD) CF ?cf))

        (if (<= ?pref 0.4)
                then
                (assert (QR-3 more_compromise NOT_VERY_GOOD) CF ?cf))

)

                        ;;;;;;;;;;;;;;;; Fhc ;;;;;;;;;;;;;;;;;;

        ;;;;;;;; MANUFACTURABILITY criteria ;;;;;;;;;

                ;;;;;; No effect ;;;;;

        ;;;;;;; CHOICE OF MATERIAL criteria ;;;;;;;;;;

                ;;;;;; No effect ;;;;;;

        ;;;;;;;;; SPECIAL PREFERENCES criteria ;;;;;;;;;;

                ;;;;;; No effect ;;;;;;

                        ;;;;;;;;;;;;;; Tc1 ;;;;;;;;;;;;;;;;

        ;;;;;;;;;;;; MANUFACTURABILITY criteria ;;;;;;;;;;

                ;;;;;;;; No effect ;;;;;;

        ;;;;;;;;;;;; CHOICE OF MATERIAL criteria ;;;;;;;;;;;;;

                ;;;;;;; No effect ;;;;;;

        ;;;;;;;;;;;; SPECIAL PREFERENCES criteria ;;;;;;;;;;;;;

                ;;;;;;;; No effect ;;;;;;

                        ;;;;;;;;;;;;;;; dth ;;;;;;;;;;;;;;;;;;;

        ;;;;;;;;;; MANUFACTURABILITY criteria ;;;;;;;;;;;;;;

(defrule rule-9
        (declare (salience -50) (CF 0.7))
        (HEURISTICS-1 NO)
        (dth VERY_HIGH)
        ?fa <- (dth_pref-1 ?)
        ;;?eff <- (QR-1 ?)
```

```
	=>

	(bind ?cf (get-cf ?fa))
	;;(plot-fuzzy-value t + nil nil ?eff)
	(assert (QR-1 VERY_GOOD) CF 1.0)
)

(defrule rule-10
	(declare (salience -50) (CF 0.7))
	(HEURISTICS-1 NO)
	(dth not VERY_HIGH)
	?fa <- (dth_pref-1 ?)
	;;?eff <- (QR-1 ?)

	=>

	(bind ?cf (get-cf ?fa))
	(bind ?pref (get-fs-x ?fa 0))
	;;(plot-fuzzy-value t + nil nil ?eff)

	(if (> ?pref 0.9)
		then
		(assert (QR-1 BAD) CF ?cf))

	(if (and (<= ?pref 0.9) (> ?pref 0.8))
		then
		(assert (QR-1 slightly_compromise BAD) CF ?cf))

	(if (and (<= ?pref 0.8) (> ?pref 0.6))
		then
		(assert (QR-1 less_compromise BAD) CF ?cf))

	(if (and (<= ?pref 0.6) (> ?pref 0.4))
		then
		(assert (QR-1 compromise BAD) CF ?cf))

	(if (<= ?pref 0.4)
		then
		(assert (QR-1 more_compromise BAD) CF ?cf))

)

	;;;;;;;;;;;; CHOICE OF MATERIAL criteria ;;;;;;;;;;;;;;

(defrule rule-11
	(declare (salience -60) (CF 0.8))
	(HEURISTICS-2 NO)
	(dth VERY_LOW)
	?fa <- (dth_pref-2 ?)
	;;?eff <- (QR-2 ?)

	=>

	(bind ?cf (get-cf ?fa))
	;;(plot-fuzzy-value t + nil nil ?eff)
	(assert (QR-2 VERY_GOOD) CF 1.0)
)

(defrule rule-12
	(declare (salience -60) (CF 0.8))
	(HEURISTICS-2 NO)
```

```
        (dth not VERY_LOW)
        ?fa <- (dth_pref-2 ?)
        ;;?eff <- (QR-2 ?)

        =>

        (bind ?cf (get-cf ?fa))
        (bind ?pref (get-fs-x ?fa 0))
        ;;(plot-fuzzy-value t + nil nil ?eff)

        (if (> ?pref 0.9)
                then
                (assert (QR-2 BAD) CF ?cf))

        (if (and (<= ?pref 0.9) (> ?pref 0.8))
                then
                (assert (QR-2 slightly_compromise BAD) CF ?cf))

        (if (and (<= ?pref 0.8) (> ?pref 0.6))
                then
                (assert (QR-2 less_compromise BAD) CF ?cf))

        (if (and (<= ?pref 0.6) (> ?pref 0.4))
                then
                (assert (QR-2 compromise BAD) CF ?cf))

        (if (<= ?pref 0.4)
                then
                (assert (QR-2 more_compromise BAD) CF ?cf))

)

        ;;;;;;;;;;;; SPECIAL PREFERENCES criteria ;;;;;;;;;;;;;;

(defrule rule-13
        (declare (salience -70) (CF 0.9))
        (HEURISTICS-3 NO)
        (dth MEDIUM)
        ?fa <- (dth_pref-3 ?)
        ;;?eff <- (QR-3 ?)

        =>

        (bind ?cf (get-cf ?fa))
        ;;(plot-fuzzy-value t + nil nil ?eff)
        (assert (QR-3 VERY_GOOD) CF 1.0)
)

(defrule rule-14
        (declare (salience -70) (CF 0.9))
        (HEURISTICS-3 NO)
        (dth not MEDIUM)
        ?fa <- (dth_pref-3 ?)
        ;;?eff <- (QR-3 ?)

        =>

        (bind ?cf (get-cf ?fa))
        (bind ?pref (get-fs-x ?fa 0))
        ;;(plot-fuzzy-value t + nil nil ?eff)

        (if (> ?pref 0.9)
```

```
            then
            (assert (QR-3 BAD) CF ?cf))

        (if (and (<= ?pref 0.9) (> ?pref 0.8))
            then
            (assert (QR-3 slightly_compromise BAD) CF ?cf))

        (if (and (<= ?pref 0.8) (> ?pref 0.6))
            then
            (assert (QR-3 less_compromise BAD) CF ?cf))

        (if (and (<= ?pref 0.6) (> ?pref 0.4))
            then
            (assert (QR-3 compromise BAD) CF ?cf))

        (if (<= ?pref 0.4)
            then
            (assert (QR-3 more_compromise BAD) CF ?cf))

)

                    ;;;;;;;;;;;;;;;;;; kw ;;;;;;;;;;;;;;;;;;

        ;;;;;;;;;;; MANUFACTURABILITY criteria ;;;;;;;;;;;;;

            ;;;;;; No effect ;;;;;;

        ;;;;;;;;;;; CHOICE OF MATERIAL criteria ;;;;;;;;;;;;;

            ;;;;;; No effect ;;;;;;;

        ;;;;;;;;; SPECIAL PREFERENCES criteria ;;;;;;;;;;;;;;

            ;;;;;;;;; No effect ;;;;;;;;;


                    ;;;;;;;;;;;;;;;;;; Rp ;;;;;;;;;;;;;;;;;;;;

        ;;;;;;;;;;;; MANUFACTURABILITY criteria ;;;;;;;;;;;;;;

            ;;;;;;;;; No effect ;;;;;;;;

        ;;;;;;;;;;; CHOICE OF MATERIAL criteria ;;;;;;;;;;;;

            ;;;;;;;; No effect ;;;;;;;;

        ;;;;;;;;;;;; SPECIAL PREFERENCES criteria ;;;;;;;;;;;;;;;;

(defrule rule-15
        (declare (salience -80) (CF 0.5))
        (HEURISTICS-3 NO)
        (or (Rp LOW) (Rp above LOW))
        ?fa <- (Rp_pref-3 ?)
        ;;?eff <- (QR-3 ?)

        =>

        (bind ?cf (get-cf ?fa))
        ;;(plot-fuzzy-value t + nil nil ?eff)
        (assert (QR-3 GOOD) CF 1.0)
)
```

```
(defrule rule-16
	(declare (salience -80) (CF 0.5))
	(HEURISTICS-3 NO)
	(Rp below LOW)
	?fa <- (Rp_pref-3 ?)
	;;?eff <- (QR-3 ?)

	=>

	(bind ?cf (get-cf ?fa))
	(bind ?pref (get-fs-x ?fa 0))
	;;(plot-fuzzy-value t + nil nil ?eff)

	(if (> ?pref 0.9)
		then
		(assert (QR-3 BAD) CF ?cf))

	(if (and (<= ?pref 0.9) (> ?pref 0.8))
		then
		(assert (QR-3 slightly_compromise BAD) CF ?cf))

	(if (and (<= ?pref 0.8) (> ?pref 0.6))
		then
		(assert (QR-3 less_compromise BAD) CF ?cf))

	(if (and (<= ?pref 0.6) (> ?pref 0.4))
		then
		(assert (QR-3 compromise BAD) CF ?cf))

	(if (<= ?pref 0.4)
		then
		(assert (QR-3 more_compromise BAD) CF ?cf))

)

				;;;;;;;;;;;;;;;;;; Rs ;;;;;;;;;;;;;;;;;;;;

	;;;;;;;;;;;;; MANUFACTURABILITY criteria ;;;;;;;;;;;;;;;;;;

		;;;;;; No effect ;;;;;;

	;;;;;;;;;;;;; CHOICE OF MATERIAL criteria ;;;;;;;;;;;;;;;;;

		;;;;;;;; No effect ;;;;;;

	;;;;;;;;;;;; SPECIAL PREFERENCES criteria ;;;;;;;;;;;;;;;;

(defrule rule-17
	(declare (salience -90) (CF 0.8))
	(HEURISTICS-3 NO)
	(Rs VERY_LOW)
	?fa <- (Rs_pref-3 ?)
	;;?eff <- (QR-3 ?)

	=>

	(bind ?cf (get-cf ?fa))
	;;(plot-fuzzy-value t + nil nil ?eff)
	(assert (QR-3 VERY_GOOD) CF 1.0)
)

(defrule rule-18
```

```
	(declare (salience -90) (CF 0.8))
	(HEURISTICS-3 NO)
	(Rs LOW)
	?fa <- (Rs_pref-3 ?)
	;;?eff <- (QR-3 ?)

	=>

	(bind ?cf (get-cf ?fa))
	;;(plot-fuzzy-value t + nil nil ?eff)
	(assert (QR-3 GOOD) CF 1.0)
)

(defrule rule-19
	(declare (salience -90) (CF 0.8))
	(HEURISTICS-3 NO)
	(Rs MEDIUM)
	?fa <- (Rs_pref-3 ?)
	;;?eff <- (QR-3 ?)

	=>

	(bind ?cf (get-cf ?fa))
	(bind ?pref (get-fs-x ?fa 0))
	;;(plot-fuzzy-value t + nil nil ?eff)

	(if (> ?pref 0.9)
		then
		(assert (QR-3 NOT_VERY_GOOD) CF ?cf))

	(if (and (<= ?pref 0.9) (> ?pref 0.8))
		then
		(assert (QR-3 slightly_compromise NOT_VERY_GOOD) CF ?cf))

	(if (and (<= ?pref 0.8) (> ?pref 0.6))
		then
		(assert (QR-3 less_compromise NOT_VERY_GOOD) CF ?cf))

	(if (and (<= ?pref 0.6) (> ?pref 0.4))
		then
		(assert (QR-3 compromise NOT_VERY_GOOD) CF ?cf))

	(if (<= ?pref 0.4)
		then
		(assert (QR-3 more_compromise NOT_VERY_GOOD) CF ?cf))
)

(defrule rule-20
	(declare (salience -90) (CF 0.8))
	(HEURISTICS-3 NO)
	(Rs below MEDIUM)
	?fa <- (Rs_pref-3 ?)
	;;?eff <- (QR-3 ?)

	=>

	(bind ?cf (get-cf ?fa))
	(bind ?pref (get-fs-x ?fa 0))
	;;(plot-fuzzy-value t + nil nil ?eff)

	(if (> ?pref 0.9)
		then
```

```
                (assert (QR-3 BAD) CF ?cf))

        (if (and (<= ?pref 0.9) (> ?pref 0.8))
                then
                (assert (QR-3 slightly_compromise BAD) CF ?cf))

        (if (and (<= ?pref 0.8) (> ?pref 0.6))
                then
                (assert (QR-3 less_compromise BAD) CF ?cf))

        (if (and (<= ?pref 0.6) (> ?pref 0.4))
                then
                (assert (QR-3 compromise BAD) CF ?cf))

        (if (<= ?pref 0.4)
                then
                (assert (QR-3 more_compromise BAD) CF ?cf))
)


                        ;;;;;;;;;;;;;;;; df ;;;;;;;;;;;;;;;;;;

        ;;;;;;;;;;;;;; MANUFACTURABILITY criteria ;;;;;;;;;;;;;;;

(defrule rule-21
        (declare (salience -100) (CF 0.9))
        (HEURISTICS-1 NO)
        (df BIG)
        ?fa <- (df_pref-1 ?)
        ;;?eff <- (QR-1 ?)

        =>

        (bind ?cf (get-cf ?fa))
        ;;(plot-fuzzy-value t + nil nil ?eff)
        (assert (QR-1 VERY_GOOD) CF 1.0)
)

(defrule rule-22
        (declare (salience -100) (CF 0.9))
        (HEURISTICS-1 NO)
        (df not BIG)
        ?fa <- (df_pref-1 ?)
        ;;?eff <- (QR-1 ?)

        =>

        (bind ?cf (get-cf ?fa))
        (bind ?pref (get-fs-x ?fa 0))
        ;;(plot-fuzzy-value t + nil nil ?eff)

        (if (> ?pref 0.9)
                then
                (assert (QR-1 NOT_VERY_GOOD) CF ?cf))

        (if (and (<= ?pref 0.9) (> ?pref 0.8))
                then
                (assert (QR-1 slightly_compromise NOT_VERY_GOOD) CF ?cf))

        (if (and (<= ?pref 0.8) (> ?pref 0.6))
                then
                (assert (QR-1 less_compromise NOT_VERY_GOOD) CF ?cf))
```

```
        (if (and (<= ?pref 0.6) (> ?pref 0.4))
                then
                (assert (QR-1 compromise NOT_VERY_GOOD) CF ?cf))

        (if (<= ?pref 0.4)
                then
                (assert (QR-1 more_compromise NOT_VERY_GOOD) CF ?cf))

)

        ;;;;;;;;;;;;;; CHOICE OF MATERIAL criteria ;;;;;;;;;;;;;;;;

                ;;;;;;;; No effect ;;;;;;;;;

        ;;;;;;;;;;;;; SPECIAL PREFERENCES criteria ;;;;;;;;;;;;;;;;

(defrule rule-23
        (declare (salience -110) (CF 0.7))
        (HEURISTICS-3 NO)
        (df MEDIUM)
        ?fa <- (df_pref-3 ?)
        ;;?eff <- (QR-3 ?)

        =>

        (bind ?cf (get-cf ?fa))
        ;;(plot-fuzzy-value t + nil nil ?eff)
        (assert (QR-3 GOOD) CF 1.0)
)

(defrule rule-24
        (declare (salience -110) (CF 0.7))
        (HEURISTICS-3 NO)
        (df BIG)
        ?fa <- (df_pref-3 ?)
        ;;?eff <- (QR-3 ?)

        =>

        (bind ?cf (get-cf ?fa))
        (bind ?pref (get-fs-x ?fa 0))
        ;;(plot-fuzzy-value t + nil nil ?eff)

        (if (> ?pref 0.9)
                then
                (assert (QR-3 NOT_VERY_GOOD) CF ?cf))

        (if (and (<= ?pref 0.9) (> ?pref 0.8))
                then
                (assert (QR-3 slightly_compromise NOT_VERY_GOOD) CF ?cf))

        (if (and (<= ?pref 0.8) (> ?pref 0.6))
                then
                (assert (QR-3 less_compromise NOT_VERY_GOOD) CF ?cf))

        (if (and (<= ?pref 0.6) (> ?pref 0.4))
                then
                (assert (QR-3 compromise NOT_VERY_GOOD) CF ?cf))

        (if (<= ?pref 0.4)
                then
```

```
		(assert (QR-3 more_compromise NOT_VERY_GOOD) CF ?cf))
)

(defrule rule-25
	(declare (salience -110) (CF 0.7))
	(HEURISTICS-3 NO)
	(df SMALL)
	?fa <- (df_pref-3 ?)
	;;?eff <- (QR-3 ?)

	=>

	(bind ?cf (get-cf ?fa))
	(bind ?pref (get-fs-x ?fa 0))
	;;(plot-fuzzy-value t + nil nil ?eff)

	(if (> ?pref 0.9)
		then
		(assert (QR-3 BAD) CF ?cf))

	(if (and (<= ?pref 0.9) (> ?pref 0.8))
		then
		(assert (QR-3 slightly_compromise BAD) CF ?cf))

	(if (and (<= ?pref 0.8) (> ?pref 0.6))
		then
		(assert (QR-3 less_compromise BAD) CF ?cf))

	(if (and (<= ?pref 0.6) (> ?pref 0.4))
		then
		(assert (QR-3 compromise BAD) CF ?cf))

	(if (<= ?pref 0.4)
		then
		(assert (QR-3 more_compromise BAD) CF ?cf))

)

			;;;;;;;;;;;;;;;;;; Cdf ;;;;;;;;;;;;;;;;;;;;

	;;;;;;;;;;;;;; MANUFACTURABILITY criteria ;;;;;;;;;;;;;;;;

(defrule rule-26
	(declare (salience -120) (CF 0.8))
	(HEURISTICS-1 NO)
	(Cdf VERY_HIGH)
	?fa <- (Cdf_pref-1 ?)
	;;?eff <- (QR-1 ?)

	=>

	(bind ?cf (get-cf ?fa))
	;;(plot-fuzzy-value t + nil nil ?eff)
	(assert (QR-1 GOOD) CF 1.0)
)

(defrule rule-27
	(declare (salience -120) (CF 0.8))
	(HEURISTICS-1 NO)
	(Cdf not VERY_HIGH)
	?fa <- (Cdf_pref-1 ?)
	;;?eff <- (QR-1 ?)
```

```
        =>

        (bind ?cf (get-cf ?fa))
        (bind ?pref (get-fs-x ?fa 0))
        ;;(plot-fuzzy-value t + nil nil ?eff)

        (if (> ?pref 0.9)
                then
                (assert (QR-1 BAD) CF ?cf))

        (if (and (<= ?pref 0.9) (> ?pref 0.8))
                then
                (assert (QR-1 slightly_compromise BAD) CF ?cf))

        (if (and (<= ?pref 0.8) (> ?pref 0.6))
                then
                (assert (QR-1 less_compromise BAD) CF ?cf))

        (if (and (<= ?pref 0.6) (> ?pref 0.4))
                then
                (assert (QR-1 compromise BAD) CF ?cf))

        (if (<= ?pref 0.4)
                then
                (assert (QR-1 more_compromise BAD) CF ?cf))
)

        ;;;;;;;; CHOICE OF MATERIAL criteria ;;;;;;;;;

                ;;;;;; No effect ;;;;;;

        ;;;;;;;;;; SPECIAL PREFERENCES criteria ;;;;;;;;;;

(defrule rule-28
        (declare (salience -130) (CF 0.8))
        (HEURISTICS-3 NO)
        (Cdf VERY_LOW)
        ?fa <- (Cdf_pref-3 ?)
        ;;?eff <- (QR-3 ?)

        =>

        (bind ?cf (get-cf ?fa))
        ;;(plot-fuzzy-value t + nil nil ?eff)
        (assert (QR-3 VERY_GOOD) CF 1.0)
)

(defrule rule-29
        (declare (salience -130) (CF 0.8))
        (HEURISTICS-3 NO)
        (Cdf not VERY_LOW)
        ?fa <- (Cdf_pref-3 ?)
        ;;?eff <- (QR-3 ?)

        =>

        (bind ?cf (get-cf ?fa))
        (bind ?pref (get-fs-x ?fa 0))
        ;;(plot-fuzzy-value t + nil nil ?eff)

        (if (> ?pref 0.9)
```

```
			then
			(assert (QR-3 NOT_VERY_GOOD) CF ?cf))

		(if (and (<= ?pref 0.9) (> ?pref 0.8))
			then
			(assert (QR-3 slightly_compromise NOT_VERY_GOOD) CF ?cf))

		(if (and (<= ?pref 0.8) (> ?pref 0.6))
			then
			(assert (QR_3 less_compromise NOT_VERY_GOOD) CF ?cf))

		(if (and (<= ?pref 0.6) (> ?pref 0.4))
			then
			(assert (QR-3 compromise NOT_VERY_GOOD) CF ?cf))

		(if (<= ?pref 0.4)
			then
			(assert (QR-3 more_compromise NOT_VERY_GOOD) CF ?cf))
)

				;;;;;;;;;;;;;;;;; Ff ;;;;;;;;;;;;;;;;;;;

		;;;;;;;;; MANUFACTURABILITY criteria ;;;;;;;;;;

			;;;;;; No effect ;;;;;;;

		;;;;;;;; CHOICE OF MATERIAL criteria ;;;;;;;;;;;

			;;;;;; No effect ;;;;;;;

		;;;;;;;;; SPECIAL PREFERENCES criteria ;;;;;;;;;;;

			;;;;;;;; No effect ;;;;;;


				;;;;;;;;;;;;;;;;;; Rpf ;;;;;;;;;;;;;;;;;;;

		;;;;;;;;;; MANUFACTURABILITY criteria ;;;;;;;;;;;

			;;;;;;;; No effect ;;;;;;

		;;;;;;;;;; CHOICE OF MATERIAL criteria ;;;;;;;;;;;;;

			;;;;;; No effect ;;;;;;

		;;;;;;;;;; SPECIAL PREFERENCES criteria ;;;;;;;;;;;;;;

(defrule rule-30
		(declare (salience -140) (CF 0.5))
		(HEURISTICS-3 NO)
		(or (Rpf LOW) (Rpf above LOW))
		?fa <- (Rpf_pref-3 ?)
		;;?eff <- (QR-3 ?)

		=>

		(bind ?cf (get-cf ?fa))
		;;(plot-fuzzy-value t + nil nil ?eff)
		(assert (QR-3 GOOD) CF 1.0)
)

(defrule rule-31
```

```
        (declare (salience -140) (CF 0.7))
        (HEURISTICS-3 NO)
        (Rpf below LOW)
        ?fa <- (Rpf_pref-3 ?)
        ;;?eff <- (QR-3 ?)

        =>

        (bind ?cf (get-cf ?fa))
        (bind ?pref (get-fs-x ?fa 0))
        ;;(plot-fuzzy-value t + nil nil ?eff)

        (if (> ?pref 0.9)
                then
                (assert (QR-3 BAD) CF ?cf))

        (if (and (<= ?pref 0.9) (> ?pref 0.8))
                then
                (assert (QR-3 slightly_compromise BAD) CF ?cf))

        (if (and (<= ?pref 0.8) (> ?pref 0.6))
                then
                (assert (QR-3 less_compromise BAD) CF ?cf))

        (if (and (<= ?pref 0.6) (> ?pref 0.4))
                then
                (assert (QR-3 compromise BAD) CF ?cf))

        (if (<= ?pref 0.4)
                then
                (assert (QR-3 more_compromise BAD) CF ?cf))

)

                ;;;;;;;;;;;;; some HEURISTICS ;;;;;;;;;;;;;;;;

                        ;;;; MANUFACTURABILITY ;;;;

(defrule rule-32                        ;; heuristics-1
        (declare (salience -400) (CF 0.9))
        (Geom THREE)
        (Cdr VERY_LOW)
        (dth MEDIUM)
        (Rs VERY_LOW)
        (df BIG)
        ?fa <- (QR-1 ?any)
        ?he <- (HEURISTICS-1 NO)

        =>

        (printout t crlf)
        (printout t "           The design has satisfied Heuristic-1 (Cost of Manufacture) ::" crlf)
        (printout t "                       Geom : THREE" crlf)
        (printout t "                       Cdr  : VERY_LOW" crlf)
        (printout t "                       dth  : MEDIUM" crlf)
        (printout t "                       Rs   : VERY_LOW" crlf)
        (printout t "                       df   : BIG" crlf)
        (printout t crlf)
        (retract ?fa)
        (retract ?he)
        (assert (HEURISTICS-1 YES))
        (assert (QR-1 GOOD) CF 1.0)
```

```
)

(defrule rule-33                          ;; heuristics-2
        (declare (salience -410) (CF 0.8))
        (Geom THREE)
        (dth VERY_LOW)
        (kw VERY_LOW)
        (Rs VERY_HIGH)
        (df SMALL)
        ?fa <- (QR-1 ?)
        ?he <- (HEURISTICS-1 NO)

        =>

        (printout t crlf)
        (printout t "                 The design has satisfied Heuristic-2 (Cost of Manufacture) ::" crlf)
        (printout t "                              Geom : THREE" crlf)
        (printout t "                              dth  : VERY_LOW" crlf)
        (printout t "                              kw   : VERY_LOW" crlf)
        (printout t "                              Rs   : VERY_HIGH" crlf)
        (printout t "                              df   : SMALL" crlf)
        (printout t crlf)

        (retract ?fa)
        (retract ?he)
        (assert (HEURISTICS-1 YES))
        (assert (QR-1 BAD) CF 1.0)
)

              ;;;;;;; CHOICE OF MATERIAL ;;;;;;

(defrule rule-34                          ;; heuristics-3
        (declare (salience -420) (CF 0.8))
        (Geom THREE)
        (Tc1 VERY_HIGH)
        (dth VERY_HIGH)
        (or (kw VERY_HIGH) (kw VERY_LOW))
        (Rs VERY_HIGH)
        ?fa <- (QR-2 ?)
        ?he <- (HEURISTICS-2 NO)

        =>

        (printout t crlf)
        (printout t "                      The design has satisfied Heuristic-3 (Cost of Material) ::" crlf)
        (printout t "                              Geom : THREE" crlf)
        (printout t "                              Tc1  : VERY_HIGH" crlf)
        (printout t "                              dth  : VERY_HIGH" crlf)
        (printout t "                              kw   : VERY_HIGH or VERY_LOW" crlf)
        (printout t "                              Rs   : VERY_HIGH" crlf)
        (printout t crlf)
        (retract ?fa)
        (retract ?he)
        (assert (HEURISTICS-2 YES))
        (assert (QR-2 BAD) CF 1.0)
)

(defrule rule-35                          ;; heuristics-4
        (declare (salience -430) (CF 0.8))
        (dth VERY_LOW)
        (kw MEDIUM)
        ?fa <- (QR-2 ?)
```

```
        ?he <- (HEURISTICS-2 NO)

        =>

        (printout t crlf)
        (printout t "                    The design has satisfied Heuristic-4 (Cost of Material) ::" crlf)
        (printout t "                              dth  : VERY_LOW" crlf)
        (printout t "                              kw   : MEDIUM" crlf)
        (printout t crlf)
        (retract ?fa)
        (retract ?he)
        (assert (HEURISTICS-2 YES))
        (assert (QR-2 GOOD) CF 1.0)
)

                ;;;;;;;;; DESIGNER'S SPECIAL PREFERENCE ;;;;;;;;;;

(defrule rule-36                    ;; heuristics-5
        (declare (salience -440) (CF 0.7))
        (Tc1 VERY_HIGH)
        (dth VERY_LOW)
        (Rp VERY_HIGH)
        (df SMALL)
        ?fa <- (QR-3 ?)
        ?he <- (HEURISTICS-3 NO)

        =>

        (printout t crlf)
        (printout t "       The design has satisfied Heuristic-5 (Designer's Special Preference) ::" crlf)
        (printout t "                              Tc1  : VERY_HIGH" crlf)
        (printout t "                              dth  : VERY_LOW" crlf)
        (printout t "                              Rp   : VERY_HIGH" crlf)
        (printout t "                              df   : SMALL" crlf)
        (printout t crlf)
        (retract ?fa)
        (retract ?he)
        (assert (HEURISTICS-3 YES))
        (assert (QR-3 NOT_VERY_GOOD) CF 1.0)
)

(defrule rule-37                    ;; heuristics-6
        (declare (salience -450) (CF 0.7))
        (Cdr VERY_LOW)
        (dth HIGH)
        (Rp VERY_HIGH)
        (Rs VERY_LOW)
        (df BIG)
        (Rpf VERY_HIGH)
        ?fa <- (QR-3 ?)
        ?he <- (HEURISTICS-3 NO)

        =>

        (printout t crlf)
        (printout t "       The design has satisfied Heuristic-6 (Designer's Special Preference) ::" crlf)
        (printout t "                              Cdr  : VERY_LOW" crlf)
        (printout t "                              dth  : HIGH" crlf)
        (printout t "                              Rp   : VERY_HIGH" crlf)
        (printout t "                              Rs   : VERY_LOW" crlf)
        (printout t "                              df   : BIG" crlf)
        (printout t "                              Rpf  : VERY_HIGH" crlf)
```

```
(printout t crlf)
(retract ?fa)
(retract ?he)
(assert (HEURISTICS-3 YES))
(assert (QR-3 GOOD) CF 1.0)
)

;;;;;;;;;;;;;;;; rule to defuzzify QUALITATIVE RATINGS ;;;;;;;;;;;;;;;;;

(defrule rule-38                    ;; defuzzify Qualitative Ratings
    (declare (salience -500))
    (QR-1 ?all-1)
    (QR-2 ?all-2)
    (QR-3 ?all-3)

    =>

  (bind ?clnum (send [CL] get-clnum))
  (bind ?qvalue-1 (moment-defuzzify ?all-1))      ;; Cost of Manufacture
  (bind ?qvalue-2 (moment-defuzzify ?all-2))      ;; Cost of Material
  (bind ?qvalue-3 (moment-defuzzify ?all-3))      ;; Designer's Special Preference
  (printout t crlf)
  (printout t "         Qualitative Rating of the solution = " crlf)
  (format t "                  COST OF MANUFACTURE : %3.2f" ?qvalue-1)
  (printout t crlf)
  (format t "                  COST OF MATERIAL : %3.2f" ?qvalue-2)
  (printout t crlf)
  (format t "                  DESIGNER'S SPECIAL PREFERENCE : %3.2f" ?qvalue-3)
  (printout t crlf crlf crlf)

  (if (<= ?*cluster_number* ?clnum) then

    (retract *)
    (assert (HEURISTICS-1 NO))
    (assert (HEURISTICS-2 NO))
    (assert (HEURISTICS-3 NO))

    (Inter_Var_Preferences)

    (bind ?bestitem_inputs (send (send (nth$ ?*cluster_number* (send [CL] get-cldetails)) get-
bestitem) get-inputs))
    (bind ?bestitem_outputs (send (send (nth$ ?*cluster_number* (send [CL] get-cldetails)) get-
bestitem) get-outputs))

    (printout t " One of the GOOD solutions is :: (")
    (bind ?l 1)
    (while ( <= ?l ?*clPA*) do

        (if (= ?l 1) then
            (format t "Geom: %01d, " (nth$ ?l ?bestitem_inputs)))
        (if (= ?l 2) then
            (format t "Cdr: %03.2f, " (nth$ ?l ?bestitem_inputs)))
        (if (= ?l 3) then
            (format t "Fhc: %02.1f, " (nth$ ?l ?bestitem_inputs)))
        (if (= ?l 4) then
            (format t "Tc1: %03d, " (nth$ ?l ?bestitem_inputs)))
        (if (= ?l 5) then
            (format t "dth: %06.5f, " (nth$ ?l ?bestitem_inputs)))
        (if (= ?l 6) then
            (format t "kw: %02d, " (nth$ ?l ?bestitem_inputs)))
        (if (= ?l 7) then
            (format t "Rp: %03.2f, " (nth$ ?l ?bestitem_inputs)))
```

```
                (if (= ?l 8) then
                        (format t "Rs: %03.2f, " (nth$ ?l ?bestitem_inputs)))
                (if (= ?l 9) then
                        (format t "df: %06.5f, " (nth$ ?l ?bestitem_inputs)))
                (if (= ?l 10) then
                        (format t "Cdf: %03.2f, " (nth$ ?l ?bestitem_inputs)))
                (if (= ?l 11) then
                        (format t "Ff: %02.1f, " (nth$ ?l ?bestitem_inputs)))
                (if (= ?l 12) then
                        (format t "Rpf: %03.2f)" (nth$ ?l ?bestitem_inputs)))

                (bind ?l (+ ?l 1))
        )
        (printout t crlf)

        (printout t "                    Outputs are :: (")
        (bind ?l 1)
        (while ( <= ?l ?*clNO*) do

                (if (= ?l 1) then
                        (format t "Wcr: %07.6f, " (nth$ ?l ?bestitem_outputs)))
                (if (= ?l 2) then
                        (format t "Wcf: %07.6f, " (nth$ ?l ?bestitem_outputs)))
                (if (= ?l 3) then
                        (format t "Twg: %05.2f, " (nth$ ?l ?bestitem_outputs)))
                (if (= ?l 4) then
                        (format t "Twf: %05.2f)" (nth$ ?l ?bestitem_outputs)))
                (bind ?l (+ ?l 1))
        )
        (printout t crlf)
        (printout t "                    Itemfitness = " )
        (format t "%5.3f" (send (send (nth$ ?*cluster_number* (send [CL] get-cldetails)) get-bestitem) get-
itemfitness))
        (printout t crlf)
        (bind ?bestitem_constraints (send (send (nth$ ?*cluster_number* (send [CL] get-cldetails)) get-
bestitem) get-cons_sensi))
        (bind ?cons1 (nth$ 1 ?bestitem_constraints))
        (bind ?cons2 (nth$ 2 ?bestitem_constraints))
        (bind ?cons3 (nth$ 3 ?bestitem_constraints))

        (if (= ?cons1 10000)
          then
                (printout t crlf "              CONSTRAINTS NOT SATISFIED" crlf crlf)
                (printout t "                  Sensitivity Analysis is not performed to this design...." crlf)
          else
                (printout t crlf)
                (format t "                    Design Solution sensitivity = %03.2f" (send (send (nth$
?*cluster_number* (send [CL] get-cldetails)) get-bestitem) get-sensitivity))
                (printout t crlf)
                (bind ?bestitem_var_sensi (send (send (nth$ ?*cluster_number* (send [CL] get-cldetails))
get-bestitem) get-var_sensi))
                (printout t "                    Design Variable sensitivity :: (")
                (bind ?l 1)
                (while ( < ?l ?*clPA*) do
                        (if (= ?l 1) then
                                (format t "Cdr: %01d, " (nth$ ?l ?bestitem_var_sensi)))
                        (if (= ?l 2) then
                                (format t "Fhc: %01d, " (nth$ ?l ?bestitem_var_sensi)))
                        (if (= ?l 3) then
                                (format t "Tc1: %01d, " (nth$ ?l ?bestitem_var_sensi)))
                        (if (= ?l 4) then
                                (format t "dth: %01d, " (nth$ ?l ?bestitem_var_sensi)))
```

```
                (if (= ?l 5) then
                        (format t "kw: %01d, " (nth$ ?l ?bestitem_var_sensi)))
                (if (= ?l 6) then
                        (format t "Rp: %01d, " (nth$ ?l ?bestitem_var_sensi)))
                (if (= ?l 7) then
                        (format t "Rs: %01d, " (nth$ ?l ?bestitem_var_sensi)))
                (if (= ?l 8) then
                        (format t "df: %01d, " (nth$ ?l ?bestitem_var_sensi)))
                (if (= ?l 9) then
                        (format t "Cdf: %01d, " (nth$ ?l ?bestitem_var_sensi)))
                (if (= ?l 10) then
                        (format t "Ff: %01d, " (nth$ ?l ?bestitem_var_sensi)))
                (if (= ?l 11) then
                        (format t "Rpf: %01d)" (nth$ ?l ?bestitem_var_sensi)))
                (bind ?l (+ ?l 1))
        )
        (printout t crlf)
        (printout t "                    Constraints sensitivity: " crlf)
                        ;; for the constraint ONE
        (if (= ?cons1 1) then
                (printout t "                CONS-1 : Constraint satisfied" crlf)
        )
        (if (= ?cons1 2) then
                (printout t "                CONS-1 : Statistically Active Constraint" crlf)
        )
        (if (= ?cons1 3) then
                (printout t "                CONS-1 : Quasi-Active Constraint" crlf)
        )
        (if (= ?cons1 4) then
                (printout t "                CONS-1 : Peak-Active Constraint" crlf)
        )
                        ;; for the constraint TWO

        (if (= ?cons2 1) then
                (printout t "                CONS-2 : Constraint satisfied" crlf)
        )
        (if (= ?cons2 2) then
                (printout t "                CONS-2 : Statistically Active Constraint" crlf)
        )
        (if (= ?cons2 3) then
                (printout t "                CONS-2 : Quasi-Active Constraint" crlf)
        )
        (if (= ?cons2 4) then
                (printout t "                CONS-2 : Peak-Active Constraint" crlf)
        )
                        ;; for the constraint THREE

        (if (= ?cons3 1) then
                (printout t "                CONS-3 : Constraint satisfied" crlf)
        )
        (if (= ?cons3 2) then
                (printout t "                CONS-3 : Statistically Active Constraint" crlf)
        )
        (if (= ?cons3 3) then
                (printout t "                CONS-3 : Quasi-Active Constraint" crlf)
        )
        (if (= ?cons3 4) then
                (printout t "                CONS-3 : Peak-Active Constraint" crlf)
        )
)
(PIfuzzify Geom 0.25 (nth$ 1 ?bestitem_inputs) 1.0)
(PIfuzzify Cdr 0.005 (nth$ 2 ?bestitem_inputs) 1.0)
```

```
        (PIfuzzify Fhc 0.05 (nth$ 3 ?bestitem_inputs) 1.0)
        (PIfuzzify Tc1 0.5 (nth$ 4 ?bestitem_inputs) 1.0)
        (PIfuzzify dth 0.000005 (nth$ 5 ?bestitem_inputs) 1.0)
        (PIfuzzify kw 0.5 (nth$ 6 ?bestitem_inputs) 1.0)
        (PIfuzzify Rp 0.005 (nth$ 7 ?bestitem_inputs) 1.0)
        (PIfuzzify Rs 0.005 (nth$ 8 ?bestitem_inputs) 1.0)
        (PIfuzzify df 0.000025 (nth$ 9 ?bestitem_inputs) 1.0)
        (PIfuzzify Cdf 0.005 (nth$ 10 ?bestitem_inputs) 1.0)
        (PIfuzzify Ff 0.05 (nth$ 11 ?bestitem_inputs) 1.0)
        (PIfuzzify Rpf 0.005 (nth$ 12 ?bestitem_inputs) 1.0)
        (bind ?*cluster_number* (+ ?*cluster_number* 1))
    )
)
```

REFERENCES

Balachandran, M.; 1993; 'Knowledge-based optimum design'; Computational Mechanics Publications, Southampton, UK.

Bapi, R. S. and Denham, M. J.; 1996; 'Computational intelligence: a synergistic viewpoint'; International Journal of Computational Intelligence and Organizations'; vol. 1, no. 1; Lawrence Erlbaum Associates, Inc.; pp. 2-9.

Bapi, R. S., McCabe, S. L. and Roy, R.; 1996; Book Review on 'Artificial intelligence and neural networks: steps toward principled integration'; Neural Networks; vol. 9, no. 3; Elsevier Science Ltd.; pp. 545-548.

Beasley, D., Bull, D. R. and Martin, R. R.; 1993; 'A sequential niche technique for multimodal function optimization'; Evolutionary Computation; MIT Press; vol. 1, no. 2; pp. 101-125.

Booker, L. B.; 1982; 'Intelligent behaviour as an adaptation to the task environment'; Doctoral dissertation, Technical report no. 243; University of Michigan (Logic of Computers Group), Ann Arbour; Dissertations Abstract International, 43(2), 496B, (University Microfilms no. 8214966).

Bradburn, N. M. and Sudman, S.; 1979; Improving Interview Method and Questionnaire Design; Jossey-Bass Publishers, London.

Brindle, A.; 1981; 'Genetic algorithms for function optimization'; Unpublished doctoral dissertation; University of Alberta, Edmonton.

Cavicchio, D. J., Jr.; 1970; 'Adaptive search using simulated evolution'; Unpublished doctoral dissertation; University of Michigan, Ann Arbor.

Cedeno, W., Vemuri, V. R., and Slezak, T.; 1995; 'Multiniche crowding in genetic algorithms and its application to the assembly of DNA restriction-fragments'; Evolutionary Computation; MIT Press, Cambridge; vol. 2, no. 4; pp. 321-345.

Chandrasekaran, B.; 1990; 'Design problem solving: a task analysis'; AI Magazine; AAAI; pp. 59-71.

Cohen, H., Rogers, G. F. C. and Saravanamuttoo, H. I. H.; 1987; 'Gas Turbine Theory'; Third Edition; Longman Group UK Ltd., England.

Coyne, R. D., Rosenman, M. A., Radford, A. D., Balachandran, M., and Gero, J. S.; 1990; 'Knowledge-based design systems'; Addison-Wesley Publishing Company.

Darwen, P. and Yao Xin; 1996; 'Every niching method has its niche: fitness sharing and implicit sharing compared'; Proceedings of PPSN IV Conference; 22-27 September, Berlin.

Dasgupta, D. and McGregor, 1991; 'A structured genetic algorithms: the model and first results'; Research Report No. IKBS-2-91; University of Strathclyde, UK.

Davidor, Y.; 1991; 'A naturally occurring niche and species phenomenon: The model and first results'; Proceedings of the Fourth International Conference on Genetic Algorithms; pp. 257 - 263.

Davis, L. (Ed.); 1991; Handbook of genetic algorithms; Van Nostrand Reinhold, New York.

De Jong, K. A.; 1975; 'An analysis of the behavior of a class of genetic adaptive systems'; (Doctoral Dissertation, University of Michigan) Dissertation Abstracts International; 36(10), 5140B, University Microfilms No. 76-9381.

Deb, K. and Goldberg, D. E.; 1989; 'An investigation of niche species formation in genetic function optimization'; In J. D. Schaffer (Ed.), Proceedings of the Third International Conference on Genetic Algorithms; Morgan Kaufmann, San Mateo, CA; pp. 42-50.

Dreyfus, H. L. and Dreyfus, S.; 1986; Mind over machine; Free Press, New York.

Durkin, J.; 1994; 'Expert systems: design and development'; Prentice-Hall International Inc.

Elo, S.; 1994; 'A parallel genetic algorithm on the CM-2 for multi-modal optimization'; Proceedings of the First IEEE Conference on Evolutionary Computation; 27-29 June, Florida; pp. 818-822.

Emch, G. and Parkinson, A.; 1993; 'Using engineering models to control variability: Feasibility robustness for worst-case tolerances'; DE-Vol. 65-1, Advances in Design Automation; vol. 1; ASME; pp. 411-418.

Eshelman, L. J. and Schaffer, J. D.; 1991; 'Preventing premature convergence in genetic algorithms by preventing incest'; In R. K. Belew and L. B. Booker, Eds., Proceedings of the Fourth International Conference on Genetic Algorithms; Morgan Kaufmann, pp. 115 - 122.

Field, P.; 1996; 'A multary theory for genetic algorithms: unifying binary and nonbinary problem representations'; Doctoral thesis; University of London.

Fonseca, C. and Fleming, P. J.; 1995; 'An overview of evolutionary algorithms in multiobjective optimization'; Evolutionary Computation; vol. 3, no. 1; The MIT Press; pp. 1-16.

FuzzyCLIPS User's Guide, version 6.02A; 1994; Knowledge Systems Laboratory, National Research Council, Canada.

Goldberg, D. E.; 1983; 'Computer-aided gas pipeline operation using genetic algorithms and rule learning; (Doctoral dissertation, University of Michigan) Dissertation Abstracts International; 44(10), 3174B, University Microfilms No. 8402282.

Goldberg, D. E.; 1985; 'Alleles, loci, and the TSP'; In J. J. Grefenstette (Ed.), Proceedings of the First International Conference on genetic Algorithms; Lawrence Erlbaum Associates; pp. 154-159.

Goldberg, D. E.; 1989; Genetic Algorithms in search, optimization, and machine learning; Addison-Wesley, Reading, MA.

Goldberg, D. E. and Deb, K.; 1991; 'A comparative analysis of selection schemes used in genetic algorithms'; In G. J. E. Rawlins (Ed.), Foundations of Genetic Algorithms; Morgan Kaufmann, San Mateo; pp. 69-93.

Goldberg, D. E., Deb, K., and Horn, J.; 1992; 'Massive multimodality, deception, and genetic algorithms'; In R. Manner and B. Manderick (Eds.), Parallel Problem Solving from Nature, 2; Elsevier Science Pub. B. V.; pp. 37-46.

Goldberg, D. E., Deb, K., Kargupta, H. and Harik, G.; 1993; 'Rapid, accurate optimization of difficult problems using fast messy genetic algorithms'; Proceedings of the Fifth International Conference on Genetic Algorithms; Morgan Kauffman, San Mateo, CA; pp. 56-64.

Goldberg, D. E. and Richardson, J. J.; 1987; 'Genetic algorithms with sharing for multi-modal function optimization'; Genetic algorithms and their applications: Proceedings of the Second International Conference on Genetic Algorithms; pp. 41-49.

Green, M. (Ed.); 1992; 'Knowledge aided design'; Academic Press.

Grosso, P. B.; 1985; 'Computer simulation of genetic adaptation: Parallel subcomponent interaction in a multilocus model'; PhD Dissertation, University of Michigan; University Microfilms No. 8520908.

Harik, G.; 1994; 'Finding multiple solutions in problems of bounded difficulty'; IlliGAL Report No. 94002; University of Illinois, Urbana-Champaign.

Harik, G.; 1995; 'Finding multimodal solutions using Restricted Tournament Selection'; Proceedings of the Sixth ICGA Conference; 15-20 July, Pittsburgh; pp. 24-31.

Hill, P. G. and Peterson, C. R.; 1992; 'Mechanics and Thermodynamics of Propulsion'; Second Edition; Addison-Wesley Publishing Company, England.

Holland, J. H.; 1975; Adaptation in Natural and Artificial Systems; MIT Press, Cambridge, MA.

Holland, J. H.; 1992; Adaptation in natural and artificial systems; MIT Press; Cambridge, MA.

Jarvis, R. A. and Patrick, E. A.; 1973; 'Clustering using a similarity measure based on shared near neighbors'; IEEE Transactions on Computers; vol. 22, no. 11; pp. 388-397.

Jelasity, M. and Dombi, J.; 1995; 'GAS, an approach to a solution of the niche radius problem'; Proceedings of GALISIA'95 Conference; 12-14 September, Sheffield, UK; pp. 424-429.

Lloyd, E. (Ed.); 1984; Handbook of applicable mathematics, part B; John Wiley & Sons.

Mahfoud, S. W.; 1992; 'Crowding and preselection revisited'; In R. Manner and B. Manderick (Eds.), Parallel Problem Solving from Nature, 2; Elsevier Science Publishers B. V.; pp. 27-36.

Mahfoud S. W.; 1994; 'Crossover interactions among niches'; Proceedings of the First IEEE Conference on Evolutionary Computation, World Congress on Computational Intelligence; pp. 188-193.

Mahfoud, S. W.; 1995a; 'Niching Methods for genetic Algorithms'; Doctoral thesis; University of Illinois, Urbana-Champaign.

Mahfoud, S. W.; 1995b; 'A comparison of parallel and sequential niching methods'; Proceedings of the Sixth International Conference on Genetic Algorithms; 15-19 July, Pittsburgh; Morgan Kaufmann Pub. Inc.; pp. 136-143.

Massey, J. G., Satre, T. W. and Ray, C. D.; 1991; 'Expert systems verification and validation Part II: Implementing the concepts'; Proceedings of Application of Artificial Intelligence in Engineering VI; G. Rzevski and R. A. Adey (Eds.); Computational Mechanics Publications, Southampton, Boston; pp. 873-883.

Mendel, M. J.; 1995; 'Fuzzy logic systems for engineering : a tutorial'; Proceedings of the IEEE; vol. 83, no. 3; pp. 345-376.

Michalewicz, Z.; 1995; 'A survey of constraint handling techniques in evolutionary computation methods'; Proceedings of the Evolutionary Programming IV; 1-3 March, CA; MIT Press; pp. 135-155.

Miller, B. L. and Shaw, M. J.; 1996; 'Genetic algorithms with dynamic niche sharing for multimodal function optimization'; Proceedings of 1996 IEEE International Conference on Evolutionary Computation; 20-22 May, Nagoya, Japan; pp. 786-791.

Mitchell, M.; 1996; An introduction to Genetic Algorithms; The MIT Press, Cambridge, Massachusetts.

Oei, C. K., Goldberg, D. E., and Chang, S. J.; 1991; 'Tournament Selection, Niching, and the preservation of Diversity'; IlliGAL Report No. 91011; University of Illinois, Urbana-Champaign.

Oksala, T.; 1994; 'Design intelligence: Evolutionary and cognitive aspects'; Automation in Construction; Elsevier; vol. 3; pp. 105-111.

Oppenheim, A. N.; 1992; Questionnaire design, interviewing and attitude measurement; New Edition; Pinter Publishers, London.

Parmee, I. C.; 1993; 'The concrete arch dam: an evolutionary model of the design process'; Proceedings of the International Conference on Neural Nets and Genetic Algorithms; Innsbruck, Austria; Springer-Verlag, Wien; pp. 544-551.

Parmee, I. C.; 1994; 'The implementation of adaptive search tools to promote global search in engineering design'; Proceedings of 3rd IFIP Working Conference on Optimisation-based Computer-aided Modelling and Design; 24-26 May, UTIA, Prague.

Parmee, I. C.; 1995; 'Towards an optimal design process using adaptive search'; Proceedings of the International Conference on Engineering Design; 22-24 Aug., Praha; vol. 1; pp. 809-815.

Parmee, I. C.; 1996; 'The maintenance of search diversity for effective design space decomposition using cluster-oriented genetic algorithms (COGAs) and multi-agent strategies (GAANT)'; Proceedings of the ACEDC'96 Conference; University of Plymouth, UK; pp. 128-138.

Parmee, I. C. and Denham, M. J.; 1994; 'The integration of adaptive search techniques with current engineering design'; Proceedings of ACEDC'94 Conference; 21-22 September, Plymouth (UK); pp. 1-13.

Parmee, I. C., Johnson, M. and Burt, S.; 1994; 'Techniques to aid global search in engineering design'; Proceedings of the Seventh International Conference on Industrial and Engineering Applications of Artificial Intelligence and Expert Systems; 31 May-3 June, Austin, Texas; pp. 377-385.

Parsons, R., Forrest, S. and Burks, C.; 1995; 'Genetic operators for the DNA fragement-assembly problem'; Machine Learning; Kulwer Academic Publication; vol. 21:1/2, pp. 11-33.

Phadke, M. S.; 1989; 'Quality engineering using robust design'; Prentice-Hall International Inc.

Pham, D. T. and Yang, Y.; 1993a; 'Optimization of multi-modal discrete functions using genetic algorithms'; Proceedings of Institution of Mechanical Engineers; vol. 207; IMechE; pp. 53-59.

Pham, D. T. and Yang, Y.; 1993b; 'A genetic algorithm based preliminary design system'; Proceedings of Institution of Mechanical Engineers; vol. 207; IMechE; pp. 127-133.

Reeves, C. R. (Ed.); 1993; 'Modern Heuristic Techniques for Combinatorial Problems'; Blackwell Scientific Publications.

Roy, R.; 1993; 'Artificial neural networks and Taguchi's methodology to model a complex non-linear system'; M.Sc. (Intelligent Systems) Dissertation, University of Plymouth (UK).

Roy, R. and Cave, P.; 1996a; 'Non linear system modelling: A hybrid approach'; Proceedings of the ACEDC'96 Conference; 26-28 March, University of Plymouth (UK); pp. 304-306.

Roy, R. and Cave, P.; 1996b; 'Application of neural networks, Taguchi's methodology and genetic algorithms for system modelling'; Proceedings of EUFIT'96 Conference; 2-5 Sept., Aachen (Germany); pp. 1929-1933.

Roy, R., Cave, P. and Parmee, I. C.; 1994; 'Artificial neural networks and Taguchi's methodology to model a complex non-linear system'; Proceedings of the ACEDC'94 Conference; 21-22 Sept., University of Plymouth (UK); pp. 114-116.

Roy, R. and Parmee, I. C.; 1995; 'Adaptive Restricted Tournament Selection for multi-modal function optimisation'; Research Report No.: PEDC-03-95; Plymouth Engineering Design Centre, University of Plymouth, Plymouth (UK).

Roy, R. and Parmee, I. C.; 1996; 'Adaptive Restricted Tournament Selection for the Identification of Multiple Sub-Optima in a Multi-Modal Function'; Lecture Notes in Computer Science (Evolutionary Computing); vol. 1143; Springer - Verlag; pp. 236-256.

Roy, R., Parmee, I. C. and Purchase, G.; 1995a; 'Qualitative evaluation of engineering designs using fuzzy logic'; Research Report No.: PEDC-04-95; Plymouth Engineering Design Centre, University of Plymouth, Plymouth (UK).

Roy, R., Parmee, I. C. and Purchase, G.; 1995b; 'Sensitivity analysis of engineering designs using Taguchi's methodology'; Research Report No.: PEDC-05-95; Plymouth Engineering Design Centre, University of Plymouth, Plymouth (UK).

Roy, R., Parmee, I. C. and Purchase, G.; 1996a; 'Integrating the genetic algorithms with the preliminary design of gas turbine blade cooling systems'; Proceedings of ACEDC'96 Conference; 26-28 March, Plymouth (UK); pp. 228-235.

Roy, R., Parmee, I.C. and Purchase, G.; 1996b; 'Sensitivity analysis of engineering designs using Taguchi's methodology'; 96-DETC/DAC-1455; CD-Rom Proceedings of the ASME DETC-Design Automation Conference, 18-22 August, Irvine, CA.

Roy, R., Parmee, I. C. and Purchase, G.; 1996c; 'Qualitative evaluation of engineering designs using fuzzy logic'; 96-DETC/DAC-1449; CD-Rom Proceedings of the ASME DETC-Design Automation Conference, 18-22 August, Irvine, CA.

Satre, T. W. and Massey, J. G.; 1991; 'Expert systems verification and validation Part I: Defining the concepts'; Proceedings of Application of Artificial Intelligence in Engineering VI; G. Rzevski and R. A. Adey (Eds.); Computational Mechanics Publications, Southampton, Boston; pp. 859-872.

Sherwin, K.; 1982; 'Engineering design for performance'; Ellis Horwood Limited, Chichester, UK.

Smith, G. F. and Browne, G. J.; 1993; 'Conceptual foundations of design problem solving'; IEEE Transactions on Systems, Man, and Cybernetics; vol. 23, no. 5; pp. 1209-1219.

Smith, R. E., Forrest, S. and Perelson, A. S.; 1993; 'Searching for diverse, co-operative populations with genetic algorithms'; Evolutionary Computation; MIT Press; vol. 1, no. 2; pp. 127-149.

Srinivas, N. and Deb, K.; 1995; 'Multiobjective optimization using nondominated sorting in genetic algorithms'; Evolutionary Computation; vol. 2, no. 3; The MIT Press; pp. 221-248.

Stadnyk, I.; 1987; 'Schema recombination in a pattern recognition problem'; Genetic Algorithms and their Applications: Proceedings of the Second International Conference on Genetic Algorithms; J. J. Grefenstette (Ed.); Lawrence Erlbaum Associates; pp. 27 - 35.

Starkey, C. V.; 1992; 'Engineering design decisions'; Edward Arnold, London.

Sugeno, M. and Yasukawa, T.; 1993; 'A fuzzy-logic-based approach to qualitative modelling'; IEEE Transactions on Fuzzy Systems; vol. 1, no. 1; pp. 7-31.

Suh, N. P.; 1990; 'The principles of design'; Oxford University Press, New York.

Sundaresan, S., Ishii, K. and Houser, D.; 1993; 'A robust optimisation procedure with variations on design variables and constraints'; DE-Vol. 65-1, Advances in Design Automation; vol. 1; ASME; pp. 379-386.

Syswerda, G.; 1989; 'Uniform crossover in genetic algorithms'; Proceedings of the Third International Conference on Genetic Algorithms; Morgan Kauffman, San Mateo, CA; pp. 2-9.

Syswerda, G; 1991a; 'Schedule optimization using genetic algorithms'; In L. Davis (Ed.), Handbook of genetic algorithms; Van Nostrand Reinhold; Chapter 21, pp. 332-349.

Syswerda, G.; 1991b; 'A study of reproduction in generational and steady-state genetic algorithms'; In G. J. E. Rawlins (Ed.), Foundations of Genetic Algorithms; Morgan Kaufmann Pub.; pp. 94-101.

Taguchi, G.; 1986; 'Introduction to quality engineering'; Asian Productivity Organisation.

Tong, C. and Sriram, D. (Eds); 1992; 'Artificial intelligence in engineering design'; vol. II; Academic Press, Inc., London.

Wade, G., Roberts, A., and Williams, G.; 1994; 'Multiplier-less FIR filter design using a genetic algorithm'; IEE Proceedings on Vis. Image Signal Process; vol. 141, no. 3; pp. 175-180.

Whitley, D.; 1989; 'The GENITOR algorithm and selection pressure: why rank-based allocation of reproductive trials is best'; Proceedings of the Third International Conference on Genetic Algorithms; Morgan Kaufmann, San Mateo, CA; pp. 116-121.

Wolpert, D. H. and Macready, W. G.; 1995; 'No Free Lunch theorems for search'; Technical Report No. SFI-TR-95-02-010; The Santa Fe Institute, Santa Fe, NM.

Yang, J. B. and Sen, P.; 1994; 'Multiple objective design optimization by estimating local utility functions'; DE-Vol. 69-2, Advances in Design Automation; vol. 2; ASME; pp. 135-145.

Yin, X. and Germay, N.; 1993; 'A fast genetic algorithm with sharing scheme using cluster analysis methods in multimodal function optimization'; In ; R. F. Albrecht, C. R. Reeves, and N. C. Steele (Eds.), Proceedings of the International Conference on Artificial Neural Nets and Genetic Algorithms; Springer-Verlag, Berlin; pp. 450 - 457.

Zadeh, L. A.; 1965; 'Fuzzy sets'; Information and Control; vol. 8.

Zadeh, L. A.; 1973; 'The concept of a linguistic variable and its application to approximate reasoning'; Information Science; vol. 8; pp. 395-460.