Biologically Inspired Optimisation Algorithms for Transparent Knowledge Extraction Allied to Engineering Materials Processing

Jun Chen

Thesis submitted in partial fulfillment of the requirements for
the degree of Doctor of Philosophy in
the Department of Automatic Control and Systems Engineering of
the University of Sheffield

November 2009

To my wonderful wife: Jiajia

Abstract

Traditionally, modelling tasks involve the building of mathematical equations which can best describe the underlying process. Such a modelling practice normally requires a deep understanding of the systems under investigation, hence the reason why it is often referred to as *knowledge-driven modelling*. On the contrary, knowledge extraction from data (or *data-driven modelling*), inspired principally from artificial intelligence techniques, is based on limited knowledge of the modelling process and relies on the data describing the input and output mappings. Such a process is able to make abstractions and generalisations of the process and plays often a complementary role to *knowledge-driven modelling*.

The Fuzzy Rule-Based System (FRBS) has been found more appealing for such a knowledge extraction process, compared to other 'black-box' modelling techniques, due to its ability of providing human understandable knowledge. However, such interpretability is only semi-inherent in the FRBS. Without a special caution one can easily end up with a FRBS with equally good predictions as those given by the 'black-box' modelling methods, while on the other hand with equally bad interpretability. Hence, extracting a transparent (interpretative) FRBS is reckoned to be of a multi-objective nature with often conflicting outcomes, which gives the rationale of using bio-inspired optimisation paradigms, more specifically, Artificial Immune Systems, in this research project. In a bid to further improve the overall predictive performance, especially for the scatter and uncertain data set, an error correction scheme is proposed so that one can compensate the original predictive model via the predicted error.

The proposed immune optimisation framework was tested extensively using several benchmark problems and was compared with other salient techniques. Consistent better performances were obtained. The immune based modelling approach was tested using a set of benchmark problems, and was further applied to different real data sets, viz. Tensile Strength (TS), Elongation and Reduction of Area (ROA), taken from the steel industry, which are all featured by high dimensional, nonlinear and sparse data spaces. Results show that the

proposed modelling approach is capable of eliciting not only accurate but also transparent FRBSs. Such a transparent FRBS establishes the required predictions of the mechanical properties of materials, which on the one hand can help metallurgists to further understand the underlying mechanisms of alloys processing, and on the other hand will automate and simplify their design. Charpy toughness (impact energy) as a special data set featured by scatters and uncertainties was used to validate the proposed error correction mechanism and proved its validity.

The project is part of the research activities which are currently conducted in the Institute for Microstructural and Mechanical Process Engineering: The University of Sheffield (IMMPETUS).

Acknowledgements

I would like to thank the following people for their assistance in the completion of this thesis:

First of all, my gratitude goes to Professor Mahdi Mahfouf, not only for his help and advice throughout this project but also for his encouragement (and pressure!) that I bore in mind all the time and guided me through the duration of my Ph.D. study. Without his help, I could never imagine completing my doctorate .

I would like to thank my parents in China. Although thousands of miles away, my father's support and caring, my mother's loving watch and blessing from heaven are always with me. Their love gives me hope and courage in times of hardship and solitude.

I want to thank my wife Jiajia, her support in every possible way during the past four years made our lives in the UK a much less stressful and much more fruitful one. However, I have to say that we have agreed that we will never study for a degree at the same time again.

My thanks also go to the Department of Automatic Control and Systems Engineering for the academic and financial support that I received.

Last but not least, this thesis is for the memory of my grandfather who passed away during my 3rd year of the programme. He has been an idol of my life time and has inspired me to face difficulties in a positive way.

Jun Chen
Sheffield, United Kingdom, 4-Nov. 2009

Related Publications

During the course of Ph. D. study, a number of papers represented my research are produced. The followings are the list of them.

Book Chapter/Conference Papers:

Chen, J., Mahfouf, M. (2009): An Artificial Immune Systems based Predictive Modelling Approach for the Multi-objective Elicitation of Mamdani Fuzzy Rules-A Special Application to Modelling Alloys. *IEEE International Conference on Systems, Mans, and Cybernetics*, 2009.

Chen, J., Mahfouf, M. (2008): Artificial Immune Systems as a Bio-inspired Optimisation Technique and Its Engineering Applications. *in Hongwei Mo (Eds.): Artificial Immune Systems and Natural Computing: Applying complex Adaptive Technologies,* College of Automation, Harbin Engineering University, Harbin, China.

Chen, J., Mahfouf, M. (2008): An Immune Algorithm Based Fuzzy Predictive Modelling Mechanism using Variable Length Coding and Multi-objective Optimisation Allied to Engineering Materials Processing. *In proceedings of the 2008 IEEE International Conference on Granule Computation (GrC 2008), pp. 26-28, August 2008, China*.

Chen, J., Mahfouf, M. (2006): A Population Adaptive Based Immune Algorithm for Solving Multi-objective Optimisation Problems. *In H. Bersini and J. Carneiro (ed.): ICARIS 2006, LNCS 4163, pp. 280-293, Springer Berlin/Heidelberg*.

Colloquium Presentations:

Chen, J., Mahfouf, M. (2009): An artificial immune systems based predictive modelling approach for the multi-objective elicitation of Mamdani Fuzzy Rules. *In IMMPETUS Colloquium 2009. Sheffield.*

Chen, J., Mahfouf, M. (2009):Improving the prediction accuracy of fuzzy rule-based systems via error corrections. *In IMMPETUS Colloquium 2009. Sheffield.*

Chen, J., Mahfouf, M. (2008): An Immune Algorithm Based Fuzzy Predictive Modelling Mechanism using Variable Length Coding and Multi-objective Optimisation Allied to Engineering Materials Processing. *In IMMPETUS Colloquium 2008. Sheffield.*

Contents

1 Introduction

2 Bio-Inspired Computing

3 A Population Adaptive Based Immune Algorithm

4 An Evolutionary Based Clustering Algorithm

5 An Immune Inspired Multi-Objective Fuzzy Modelling (IMOFM)

List of Figures

List of Tables

Table Page

Abbreviations

Abs	**A**nti**b**odie**s**
AFRBS	**A**IS **F**uzzy **R**ule **B**ased **S**ystems
Ags	**A**nti**g**en**s**
AIS	**A**rtificial **I**mmune **S**ystems
ANN	**A**rtificial **N**eural **N**etworks
BEP	**B**ack-**E**rror **P**ropagation
BF	**B**last **F**urnace
BOF	**B**asic **O**xygen **F**urnace
CFSL	**C**onventional **F**eedback-based **S**upervised **L**earning
DDM	**D**ata-**D**riven **Modelling**
EAF	**E**lectric **A**rc **F**urnace
EAs	**E**volutionary **A**lgorithm**s**
EC	**E**volutionary **C**omputing
ECS	**E**rror **C**orrection **S**cheme
EPF	**E**rror **P**redictive **F**RBS
ES	**E**volution **S**trategies
FCM	**F**uzzy ***C***-**M**eans
FM-HCMO	**F**uzzy **Modelling** approach with a **H**ierarchical **C**lustering algorithm and a **M**ulti- objective **O**ptimisation mechanism
FRBS	**F**uzzy **R**ule **B**ased **S**ystems
GA	**G**enetic **A**lgorithm
GD	**G**enerational **D**istance
G3PCX	**Generalis**ed **G**eneration **G**ap model and the **PCX**
G3Kmeans	Hybridisation of **G3**PCX and **K-means**
IMOFM_S/M	An **I**mmune inspired **M**ulti-**O**bjective **F**uzzy **Modelling** mechanism for **S**ingleton/**M**amdani FRBS
KB	**K**nowledge **B**ase
MCP	**M**c**C**ulloch-**P**itts
MCR	**M**ean-**C**entric **R**ecombination
MLP	**M**ulti-**l**ayer **P**erceptron
MO	**M**ulti-objective **O**ptimisation

MOP	**MO P**roblems
M-PAIA2	A Multi-stage optimisation procedure based on PAIA2
MSE	**M**ean **S**quared **E**rror
NFS	**N**euro-**F**uzzy **S**ystems
NSGA-II	**N**on-dominated **S**orting **G**enetic **A**lgorithm **II**
OPF	**O**riginal **P**redictive **F**RBS
PAES	**P**areto **A**rchived **E**volutionary **S**trategy
PAIA	A **P**opulation **A**daptive based **I**mmune **A**lgorithms
PAIA2	An improved version of PAIA
PCX	**P**arent-**C**entric recombination
PNIA	**P**areto-optimal **N**eighbour **I**mmune **A**lgorithm
PSO	**P**article **S**warm **O**ptimisation
RBF	**R**adial **B**asis **F**unction
RMSE	**R**oot **Mean S**quared **E**rror
ROA	**R**eduction **o**f **A**rea
SBX	**S**imulated **B**inary **C**rossover
SOP	**S**ingle-objective **O**ptimisation **P**roblems
SPEA2	**S**trength **P**areto **E**volutionary **A**lgorithm
SPR	**S**imilarity of **R**ule **P**remise
TSK	**T**akagi-**S**ugeno-**K**an
UTS	**U**ltimate **T**ensile **S**trength
VIS	**V**ector **I**mmune **A**lgorithm
VLC	**V**ariable **L**ength **C**oding

Chapter 1

Introduction

"All men by nature desire knowledge".

Aristotle, Metaphysics, 384BC-322BC

1.1 General Background

In this research project, the main emphasis will be on how to allow Artificial Immune Systems (AIS) to cooperate with fuzzy rule based systems (FRBS), artificial neural networks and clustering methods in order to solve engineering problems, especially those associated with optimisation, knowledge extraction, modelling and control. With the characteristics of recognition of foreign agents, reinforcement learning, associative memory, distributed or parallel processing capability, self-adaptive and self-organization inherent in the human immune systems it is believed that AIS, as a metaphor, can accomplish the aforementioned tasks in a more efficient and transparent way. Unlike other evolutionary computing paradigms, which can be thought of as natural optimisers, AIS has been applied to broad application areas ranging from data analysis, computer security to optimisation. Hence, AIS provides a more extendable platform on which AIS Fuzzy Rule-Based System (AFRBS), for instance, can be built. The whole project can be divided into two subsequent stages, viz. optimisation and modelling.

1.2 Project Description

In the first phase of this project, AIS will be extended to the area of multi-objective optimisation problems (MOP) by realising that real-world problems are inherently of a multi-objective nature with often conflicting issues. Hence, the best way to deal with such problems is to provide a set of trade-off solutions. The rationality of adopting AIS, or broadly speaking, Evolutionary Algorithms (EAs), as *the* search engine is based on its ability to exploit the accumulated information about an initially unknown search space in order to bias subsequent searches into useful subspaces (Bodenhofer *et al.*, 1997). The advantages of AIS over classical search methods which are based on derivative information or random search methods are manifold: on the one hand, it is a derivative-free and global search method and thus offers a valid approach to tackle the problems which are not differentiable and have many local optima; on the other hand, it effectively uses previous search experiences to guide the following search rather than random search, which is recognised as the main scheme responsible for its efficiency, particularly in large, complex, and poorly understood search spaces; thirdly, its ability of simultaneously manipulating an adaptive antibody population makes it a very suitable way to handle MOP.

There have been several attempts to address the applications of AIS to MOP in the literature (Yoo *et al.*, 1999; Cruz Cortes *et al.*, 2003; Coello Coello *et al.*, 2005; Wang and Mahfouf, 2005; Jiao *et al.*, 2005; Freschi, 2006) but none of these presented a formal systematic framework for doing so. Further, all of the previous attempts are based on just a small part of the mechanism within the whole immune system. We intend to propose a more formal framework combining more immune metaphors, e.g. combining immune network theory with a clonal selection principle, and to identify the main differences between AIS and other EAs in terms of their structures, robustness, parameter settings and efficiency. Some hybridisations with other Evolutionary Algorithms (EAs), e.g. Genetic Algorithm (GA), will also be considered in this phase to improve the existing accuracy and efficiency of AIS.

In the broad sense, knowledge extraction can also be viewed as an optimisation process. The task of extracting an appropriate knowledge base (KB) is equivalent to parameterizing this KB, and to finding a set of optimal parameter values with respect to some design criteria (Cordon *et al.*, 2004). With a good and reliable optimisation algorithm developed via the first stage, the project can proceed to the second stage, which will consider a ‘hybrid’ form of AIS and fuzzy rule-based systems together in order to extract transparent knowledge purely from data for complex systems’ modelling. Some other soft computing techniques, such as clustering and neural networks will also be considered for inclusion at this stage in order to facilitate this process of knowledge extraction.

From the system identification viewpoint, KBs can also be viewed as identified models of the systems under investigation. There are mainly three methodologies to solve identification tasks, viz. white-box modelling, black-box modelling and grey-box modelling. In the line of the white box modelling everything is considered to be known as *a priori* from physics and the produced model (knowledge) is normally in the form of a mathematical equation following related physical laws without the need for any other measurements. However, in a real-world modelling situation, one can never have complete process knowledge, and uncertain factors always affect the system, which only have chance to be revealed (or partially revealed) through experiments. Black-box modelling responds to these requirements and is designed entirely from measured data without assumption of knowing any physical or verbal insight at all. The drawbacks of black-box modelling are twofold: first, once the model has been built one can only obtain a projection from inputs to outputs and nothing more, which means that no deep understanding about the process itself can be obtained through the modelling procedure; second, the thrust of the principle in the modelling field is to only estimate what is still unknown, however, black-box modelling breaks this law to some extent by employing a sufficiently flexible model family (Hellendoorn *et al.*, 1997). To overcome these problems, the need for grey-box modelling is pressing, which combines both human knowledge and black box estimation to account for complex systems' knowledge acquisition.

Fuzzy rule-based system is the one that falls into the third category with an additional ability to integrate human expert knowledge in the form of vague or imprecise statements rather than crisp mathematics, for many real-world systems' knowledge can only be described by experts using natural language. Previous research on fuzzy rule-based system has been mainly concerned with how to synthesis a rule-base with domain dependent knowledge from human experts, such as operators, and render the task of optimising the parameters associated with the antecedent and consequent parts to some estimation methods, e.g. recursive least squares or gradient based methods (Takagi *et al.*, 1985). However, this paradigm gives rise to three limitations:

1) More often than not, expert knowledge is lacking or is limited due to the newly discovered unknown complex system, or the narrow and partial knowledge gathered from a single expert.
2) It is very hard to handle problems with considerable amount of data to be processed and analysed (Cordon *et al.*, 2004).

3) The way to design such a fuzzy system is not domain-independent and thus no systematic design procedure can be followed.

In all these cases, a sole knowledge extraction from would-be experts will undoubtedly fail to provide a satisfactory solution, while it must be stressed that discovering knowledge from data can help in overcoming aforementioned limitations by augmenting a fuzzy rule-based system with an additional learning ability provided by some machine learning approaches.

In the past two decades, many successes have been witnessed in the hybridisation of neural networks and fuzzy systems. A well-known representative is Adaptive Network-based Fuzzy Inference Systems (ANFIS) (Jang, 1993). Although neuro-fuzzy systems may contribute successfully in overcoming the lack of the linguistic representation and transparency associated with the neural networks, the designer will still need to decide on major design parameters such as universe granulation, rule antecedent aggregation operators, rule semantics, rule base aggregation operators and defuzzification methods (Cordon *et al.*, 2004). Almost at the same time, attempts of hybridising clustering methods with fuzzy systems were carried out and gave very promising results (Gomez-Skameta *et al.*, 1999). The aim of this type of hybridisation is to automatically infer rules from large collections of learning data. However, designers following this line of research still face the problem of setting an appropriate cluster number as *a priori*, and the clustering depends highly on the chosen starting point.

As pointed out by Cordon *et al.* (2004), contrary to neural networks and clustering, GAs provide a means to encode and evolve everything involved in the design of the rule base. Despite the prospective promising future, hitherto, no systematic design procedure has been put forward regarding this new line of research, although many successful attempts have been made in the past. Clearly, the main difficulty is that if everything is encoded and evolved using GAs the search space becomes prohibitively large. Hence, one has to reach a compromise on what level GAs are to be involved to learn to cover, e.g. data base tuning, rule base learning or the whole knowledge base extraction. For this reason, many possibilities exist, which hinder the formation of a generic systematic design procedure. Having said this, AIS architectures have shown great capabilities in dealing with high dimensional optimisation problems and exhibited great flexibility. It is believed that with their global search capability and their encoding schemes being similar to those of GA, and with their learning and data analysis capabilities being similar to those of neural networks, AIS can offer a better and a more integrated route to solving complex knowledge extraction problems. At this stage of this project, the emphasis is on how to automatically generate fuzzy rule-bases from numerical data only and on how to select rules in the generated rule-base to remove or merge redundant and similar rules to obtain a compact and transparent knowledge base. Clustering and neural networks will be considered to either be incorporated into or compared with AFRBS.

The most attractive property of fuzzy systems lies in its capability of processing linguistic expression and providing human understandable knowledge. However, sometimes this property is only compromised in order to produce more accurate results, which can be achieved through either a mathematical function of the consequent part, or an increased number of complex rules, both routes deviating from the original intention of FRBS. Thus, in this project, more attention will be focused on the Singleton FRBS (Takagi *et al.*, 1985) and the Mamdani FRBS (Mamdani *et al.*, 1975) rather than the Takagi-Sugeno-Kang (TSK) FRBS with linear functions as its consequents (Takagi *et al.*, 1985). To improve the interpretability of the fuzzy model, MOP will be incorporated as a substitute for the traditional estimation methods by accepting the fact that the increased interpretability is often a contradictory goal against the objective of the accuracy. Using MOP algorithm developed in the first phase of this project, one can simultaneously deal with several, usually cofilicting, objectives in a consistent fashion.

Although FRBS can deal with imprecise data and incomplete knowledge, collected data, especially in ‘dirty’ environments, such as the steel industry, may often consist of severe stochastic activity which cannot be modelled easily. In order to further improve the generalisation ability of the model elicited via a data-driven modelling method in such a scenario, a special case of ‘Stacked Generalisation’ (Wolpert, 1992) is investigated, which relates to the case of when the first layer contains only one generaliser. In such a case, ‘Stacked Generalisation’ is reduced to a scheme for estimating the error of the model in the first layer. An error correction scheme (ECS) is thus proposed based on such a special case. The basic idea of ECS is to build an Error Predictive FRBS (EPF) apart from the Original Predictive FRBS (OPF) so that one can predict the errors associated with the OPF, given the inputs of OPF. When a new scenario is encountered, the EPF will be able to predict the potential error and thus the predicted error can be used to compensate for the predicted output produced by the OPF. An improved predictive accuracy in terms of not only the learning but also in terms of the generalisation is expected via the ECS.

All the proposed methods are tested with benchmark problems and with a ‘real world’ engineering application associated with the mechanical property prediction for hot-rolled steels. Specialist heat treatments are used to develop the required mechanical properties in a range of alloy steels. The mechanical properties of the alloy steels depend on several factors of which the followings are believed to be the major ones: tempering temperature, quench type, chemical compositions of the steel, geometry of the bar, test sample location on the bar, batch distribution in the furnace, measurement tolerances and variations in the process equipment and operators (Tenner, 1999). Traditionally, a heat treatment metallurgist would attempt to balance these factors using their metallurgical knowledge and experience in a bid to obtain the desired mechanical properties. However, due to the increasing complexity of the underlying system, this may still prove difficult even for the metallurgists to tune these parameters. Given the lack of mathematical models which can account for these complex systems and a large amount of available industrial process data associated with the systems, data-driven modelling becomes more and more vital for assisting the metallurgist to predict the mechanical test results without actually doing it. Based on these models, further optimisations of the heat treatment process can also be developed, which is envisaged to be able to automate the steel design process and reduce the experimental costs.

All in all, this research aims at proposing a systematic and integrated knowledge extraction framework with considerable transparency using AFRBS to automate and simplify the design of the alloy steels. This particular application work is currently being carried-out as part of a project within the EPSRC sponsored Institue for Microstructural and Mechanical Process Engineering: The University of Sheffield (IMMPETUS) project. IMMPETUS is a multi-disciplinary research centre dedicated to integrating metallurgical, mechanical and thermal considerations in developing soundly based models for process planning and control to achieve target microstructures and product properties with increasingly fine tolerances and greater efficiency. The core of the approach is the concept of black, grey and white box modelling, and this requires a wide range of techniques and interdisciplinary knowledge. As control engineers involved in this project, the concerns are more related to AIS, fuzzy logic and hybrid modelling methodology, i.e. using an intelligent optimisation approach to design metals in a ‘right-first-time’ fashion.

1.3 Original Contributions

The original contributions of this thesis are of the followings:

- A population adaptive based immune algorithm (PAIA) has been proposed for solving MOP.
- A Multi-stage optimisation procedure is proposed, where a single-objective optimisation method is utilised in the first stage to obtain any global optimum resting on the Pareto front, and PAIA is then invoked in the second stage as *the* post-processing algorithm to approximate the rest solutions along the Pareto front.
- The differences and the extra strength of immune based optimisation algorithms, as compared to other EAs, have been identified and summarised.
- An evolutionary algorithm based clustering method has been proposed, which is the product of hybridisation of a real-coded GA and K-means algorithm. The proposed algorithm can avoid local optima during the search of proper cluster centres and, therefore, can find cluster centres as close to the real ones as possible.
- A systematic Immune inspired Multi-objective Fuzzy Modelling (IMOFM) is proposed which can be regarded as a three-stage modelling procedure. The first stage is used to extract the prior knowledge from the available data using an evolutionary clustering algorithm. The second stage is used to refine such an initial model using a modified back-error propagation algorithm which can deal with both Singleton and Mamdani FRBSs. In the third and final modelling stage, and in order to tackle the problem of simultaneously optimising the rule-base structure and parameters, a variable length coding scheme (VLC) is adopted, and a new distance index is proposed to cope with the variable-length individuals.
- The proposed modelling mechanism has been tested with benchmark problems and has been applied to the prediction of mechanical properties of alloy steels, such as Ultimate Tensile Strength (UTS), Reduction of Area (ROA), elongation and impact energy. The results are promising.
- In order to further improve the generalisation ability of the models elicited via a data-driven method, a special case of ‘Stacked Generalisation’, namely the error correction scheme (ECS), is proposed. An improved predictive accuracy in terms of not only the learning but also the generalisation is observed.

1.4 Outline of the Thesis

The thesis is organized as follows:

Chapter 2 gives a basic introduction to the relevant aspects of bio-inspired computing, such as evolutionary computing, Artificial Immune Systems (AIS), particle swarm optimisation (PSO), artificial neural networks (ANNs) and fuzzy rule based systems (FRBS). The emphasis is then given to the general aspects of bio-inspired optimisation, which is not restricted to a particular computing paradigm. The common features embedded in the modern heuristic based optimisation algorithms are also explored.

Chapter 3 presents the development of the proposed PAIA and its improved version, viz. PAIA2. A multi-stage optimisation procedure (M-PAIA2) is also proposed, which aims as speeding up the optimisation process. All these algorithms are tested via benchmark problems, such as the well-known ZDT and DTLZ test suites, and compared with other well-known algorithms, such as the Non-dominated Sorting Genetic Algorithm II (NSGA-II), the Strength Pareto Evolutionary Algorithm (SPEA2) and the Vector Immune algorithm (VIS). The differences and the extra strength of the immune based optimisation algorithms, compared to other EAs, are also elucidated.

Chapter 4 provides the details which describe the proposed evolutionary based clustering algorithm. Experimental studies on the proposed clustering algorithm are carried out in order to justify such hybridisation. The relationship between *unsupervised learning* and *supervised learning* is further expanded so that one can easily generalise it to the relationship between data clustering and the elicitation of FRBS.

Chapter 5 discusses the implementation of an immune inspired multi-objective fuzzy modelling (IMOFM) mechanism, which can be used to elicit not only Singleton FRBS but also Mamdani FRBS. To this end, a modified Mamdani fuzzy inference system is proposed with a carefully chosen output membership functions, the inference and the defuzzification methods. Such modifications ensure the efficiency and the differentiability of the developed Mamdani FRBS, which also leads to a set of new back-error propagation (BEP) updating formulas for refining Mamdani FRBS. Some important factors, such as the variable length coding scheme and the rule alignment, are also discussed.

Chapter 6 presents the results of using IMOFM method for the modelling of two benchmark problems and for the predictions of three mechanical properties of alloy steels, viz. UTS, ROA and elongation. Such results are also compared with other already established modelling methods, such as the Fuzzy Modelling Approach with a Hierarchical Clustering Algorithm and a Multi-objective Optimisation Mechanism (FM-HCMO). The empirical differences between Singleton FRBS and Mamdani FRBS are also discussed.

Chapter 7 describes an Error Correction Scheme (ECS) which improves predictive accuracy in terms of not only the learning but also the generalisation. Experimental results on the impact energy are presented.

Finally, concluding remarks, new perspectives and future research directions are given in **Chapter 8**.

Chapter 2

Bio-Inspired Computing

"The designs found in nature are nothing short of brilliant, but the process of design that generates them is utterly lacking in intelligence of its own".

Daniel Dennett, NY Times, 2005

Bio-Inspired Computing lies within the realm of Natural Computing, a field of research that is concerned with both the use of biology as an inspiration for solving computational problems and the use of the natural world experiences to solve 'real world' problems. On the one hand, the increasing interest in this field lies in the fact that nowadays the world is facing more and more complex, large, distributed and ill-structured systems, while on the other hand, people notice that the apparently simple structures and organizations in nature are capable of dealing with most complex systems and tasks with relative ease.

The most successful and visible work belongs to the realm of Evolutionary Computing (EC) through the simulation of biological evolution. Within this realm, three independently developed methodologies exist, viz. Genetic Algorithms (GA) (Holland, 1975), Evolution Strategies (ES) (Back, 1996) and Genetic Programming (GP) (Koza, 1999). Apart from EC, Artificial Immune Systems (AIS) (Farmer *et al.*, 1986) and Particle Swarm Optimisation (PSO) (Eberhart & Kennedy 1995; Kennedy & Eberhart 1995) represent alternative lines of this type of research through the simulation of vertebrate immune mechanisms and of the migration of the bird flock. In the following Sections, GA and ES, as two most widely used methods in the EC field, and AIS, as the core of this project, will be briefly reviewed. PSO, as an alternative line of this type of research will also be reviewed. Special concerns will then be given to the so called bio-inspired optimisation which includes solving Single-Objective Optimisation Problems (SOP) and Multi-Objective Optimisation Problems (MOP) with the aforementioned three methods. Artificial Neural Networks (ANN) and Fuzzy Rule Based

Systems (FRBS), as the relevant subjects and techniques that will be used in Chapter 5~ 7, will also be introduced. The review in this chapter is meant to be comprehensive but non-exclusive.

2.1 Evolutionary Based Computing (EC)

2.1.1 Genetic Algorithms

GAs are general purpose search algorithms which use principles inspired by natural genetics to evolve solutions to problems. The basic idea is to maintain a pool of chromosomes that evolves over time through a process of competition elitism and controlled variation (Cordon *et al.*, 2004; Goldberg, 1989). GAs were initially designed using binary coding scheme to solve many problems, such as gene alignment, combination optimisation and continuous optimisation. Due to GA's binary coding scheme and population-based search concept, it is very easy to be adapted to different application scenarios. Although there are many variants of GAs, it is widely accepted that a GA should have the following five components:

- A genetic representation of the solutions to the problem;
- A way to generate the initial population;
- A way to evaluate fitness of each solution (chromosome);
- Two genetic operators, viz. crossover and mutation, to alter the genetic composition of offspring during reproduction;
- A selection mechanism to introduce competition and pressure to individuals.

Standard GAs use a binary coding scheme to represent the genetics of solutions. The crossover operator under the binary coding scheme is designed by randomly picking two strings and exchanging some portion of the strings. Among all possible implementations of this kind of crossover operators, single-point and two-point crossover (Goldberg, 1989) are the most used ones. However, the feasibility of such a simple crossover scheme, which is valid in the binary case, does not hold in the real-valued case. The stagnation of finding a feasible crossover operator in the real-valued situation is the main reason for the limitation of GAs, such as slow convergence, low accuracy etc., when they are applied to continuous optimisation problems. In recent years, research on real-valued GAs made a great breakthrough. Crossovers, such as Simulated Binary Crossover (SBX) (Deb *et al.*, 1994), were proposed. In those works, crossovers are not naive real-valued crossovers (Goldberg, 1989) anymore; they are very much similar to mutation operators in ES or PSO in the way that they alter the compositions of offspring. Both the speed of the convergence and accuracy are improved to be at least as good as those of newly developed methods.

2.1.2 Evolutionary Strategies

Unlike GAs, ES-based algorithms were originally designed via real values for coding and is still marked with this. ES is a joint development of Bienert, Rechenberg and Schwefel, who built the idea and foundation for this field in the 1960s at the Technical University of Berlin (Back, 1996). It is first implemented as an experimental procedure to deal with hydrodynamical problems such as shape optimisation of a bended pipe or structure optimisation of a two-phase flashing nozzle. The strategy used is very simple, which is based on random small changes of experimental setups following observations from nature that smaller mutations occur more often than larger ones. If the new construction happened to be better than its predecessor, it will replace the old one and serves as the basis for the next trial. It was Schwefel (Schwefel, 1981) who first simulated a *two membered* ES on the first available computer which now commonly has the name of (1+1)-ES.

To incorporate the principle of a population, $(\mu + \lambda) - ES$ and $(\mu, \lambda) - ES$ were introduced by Schwefel (Schwefel, 1981). In the first case, the best μ individuals out of the union of parents and offspring survive while in the later case only the best μ offspring individuals from the next parent generation survive. It is argued that the selection schemes of *multi-membered* ES are somewhat similar to two main implementations of selection in AIS, which are discussed in more detail in the next Section and in Section 3.2.1 (refer to Section 3.2.1 for 'Clonal Selection' and 'Reselection').

2.2 Artificial Immune Systems (AIS)

AIS is relatively a new research area which can be traced back to Farmer *et al.*'s paper published in 1986 (Farmer & Packard, 1986). In this pioneering paper the author proposed a dynamical model for the immune systems based on the Clonal Selection Principle (Burnet, 1959) and Network Hypothesis (Jerne, 1974; Perelson, 1989). However, there were only a few developments since then until 1996 when the first international conference based on artificial immune systems was held in Japan. Following this event, the increasing number of researchers involved in this field indicated the emergence of a new research field: Artificial Immune Systems (AIS). But hitherto, no new formal framework based on AIS has since been proposed.

There are three main application domains which AIS research effort has focused on, viz. fault diagnosis, computer security, and data analysis. The reason behind this is that it is relatively easy to create a direct link between the real immune system and the aforementioned three application areas, e.g. in the applications of data analysis, clusters to be recognised are easily related to Antigens (Ags), and the set of solutions to distinguish between these clusters is linked to Antibodies[2.1] (Abs). Recently, a few attempts to extend AIS to the optimisation field have been made (de Castro & Von Zuben, 2002; Kelsey & Timmis, 2003). However, as mentioned by Emma Hart and Jonathan Timmis (2005), maybe partly for historical reasons, many of the AIS practitioners may be interested in the optimisation field by way of working in other biologically inspired fields such as EC, and thus in terms of optimisation the distinctive line between EC and AIS is vague. In other words, there is not a formal distinctive framework for AIS applied to optimisation. The situation is even worse when dealing with the MOP case since it is hard to find a way of defining Antigen and the *affinity* due to the implicit Antigen population to be recognised (Chen & Mahfouf, 2006).

The biological foundations of AIS are based on various immunological models which coexist to explain the functions of immune systems, each of them being only from one particular point of view and sometimes being even contradictory with each other. Hence, there is not ‘a’ formal accepted immunological model that is well recognised in the immunology community. However, from the computational perspective, each of these models is a valid model provided an efficient algorithm can be extracted from it. Sometimes, several immunological models may even be synergized in order to produce a single algorithm such that the problem under investigation can be solved. Obviously, pragmatism is a widely adopted methodology in the development of AIS at the current stage. There are two models and two phenomena which are found to be very useful especially from the computational viewpoint: (1) Model1: The Clonal Selection Principle; (2) Model2: Immune Network Theory; (3) Phenomenon1: Vaccination and Secondary Response; and (4) Phenomenon2: adaptive antibody’s

[2.1] In this thesis, Abs and B-cells are not distinguished (refer to Section 2.2.1).

concentration. These immunological models are introduced in the following Sections. Detailed reviews of the existing immune algorithms are referred to Section 3.1.1.

2.2.1 Model 1: Clonal Selection Priciple

The Clonal Selection Principle describes the basic features of an immune response to an antigenic stimulus, and establishes the idea that only those cells that recognize the antigen are selected to proliferate. Figure 2.1 visualises the steps involved in such a process.

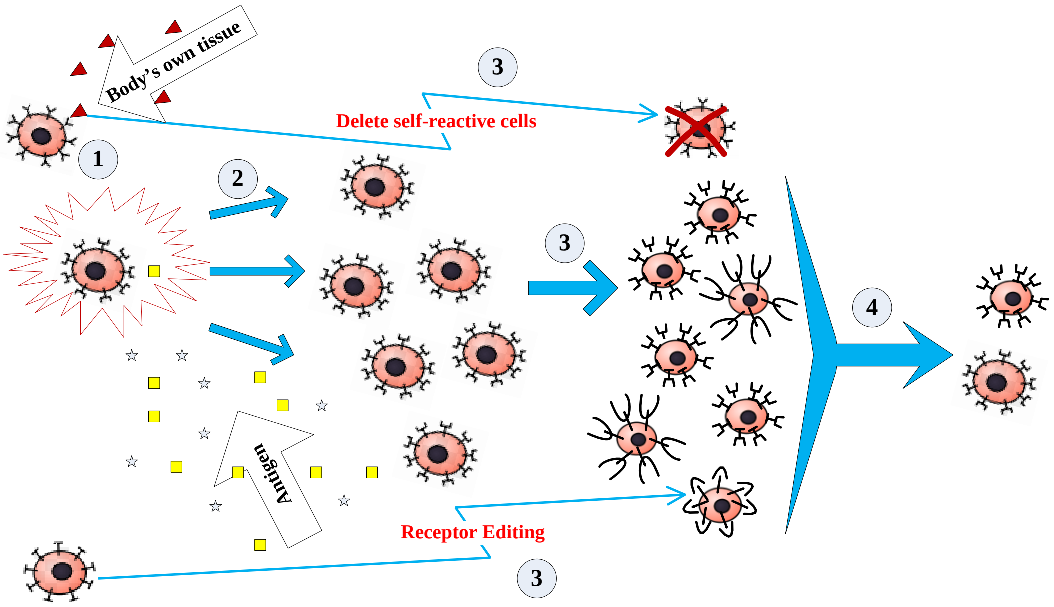


Figure 2.1 The Clonal Selection Principle: 1-Selection; 2-Proliferation; 3-Affinity maturation; 4-Reselection.

The key procedures of this principle are:

1) **Selection:** the B-cell with a higher affinity than a threshold is selected to clone itself;
2) **Proliferation:** the selected B-cells produce many offspring with the same structure as themselves; the clone size is proportional to the individual's affinity;
3) **Affinity Maturation:** this procedure consists of *Hypermutation* and *Receptor Editing*; in the former case, clones are subjected to a high-rate mutation in order to distinguish them from their parents; the higher the affinity, the lower the mutation rate; in the latter case, the cells with a low affinity, or self-reactive cells, can delete their self-reactive receptors or develop entirely new receptors;

4) **Reselection:** after affinity maturation, the mutated clones and edited cells are reselected to ensure that only those cells with a higher affinity than a certain threshold survive.

The whole process is performed iteratively until a certain stable state (i.e. the concentration of B-cells with higher affinities is not changed) is achieved.

2.2.2 Model 2: Immune Network Theory

Immune Network Theory states that ‘Abs’ not only include paratopes but also epitopes. This results in the fact that ‘Abs’ can be stimulated by recognizing other ‘Abs’, and for the same reason can be suppressed by being recognised. Consequently, the immunological memory can be acquired by this self-regulation and mutual reinforcement learning of B-cells. The suppression function is a mechanism that allows to regulate the over-stimulated B-cells to maintain a stable memory and thus serves as the inspiration to control the over-crowded population during the optimisation process.

2.2.3 Phenomenon 1: Adaptive Antibody’s Concentration

The way that the immune system controls its Antibody’s concentration represents an interesting phenomenon from the perspective of the optimisation practitioners. Initially, only a small number of B-cells cruise in the body. If they encounter foreign ‘Ags’, some of them are activated and then they proliferate. This process is adaptive, i.e. the number of clones that are proliferated during the activation process and how many of them are maintained at each iteration step and at the end in order to neutralize ‘Ags’ is adaptive. This is somewhat predictable because if a large number of initial B-cells is available then undoubtedly it can kill any ‘Ags’ at the cost of spending more energy to activate B-cells and secrete ‘Abs’. However, only an optimal number of B-cells during each step is necessary (a less number means more time is needed to reach the required concentration; more means redundant B-cells are introduced).

2.2.4 Phenomenon 2: Vaccination and Secondary Immune Response

It is well known that if the vaccine (which is very similar to the real antigens in terms of their structures) is available and first applied to the immune systems, the immune systems can remember it and can respond quickly in the successive encounter of similar antigens. Such a response is called the secondary response in the immune community. As shown in Figure 2.2, the response lag of the secondary response is much smaller than that of the primary response, which means that, given a known vaccine, the immune response can be induced quickly and strongly in the second run.

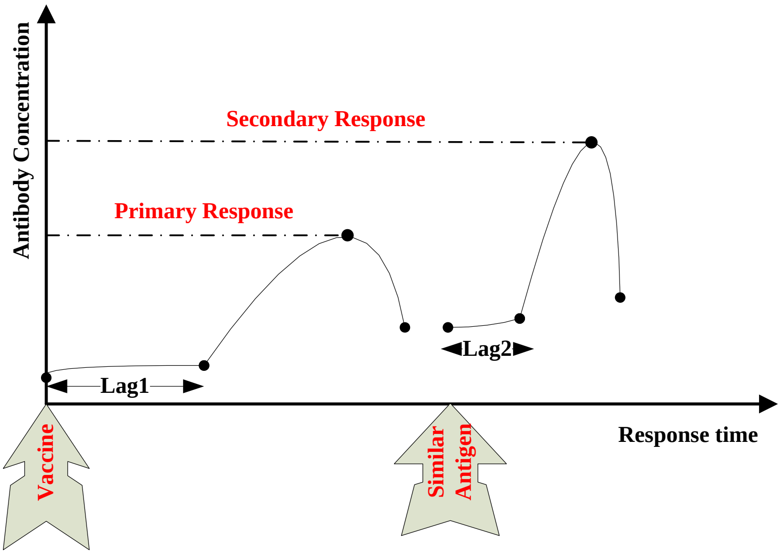


Figure 2.2 Vaccination and immune response.

2.3 Particle Swarm Optimisation (PSO)

Particle swarm optimisation (PSO) is a population based stochastic optimisation technique first described by Russell C. Eberhart and James Kennedy in 1995 (Eberhart and Kennedy, 1995), inspired by social behaviour of bird flocking or fish schooling. PSO shares many similarities with EC techniques such as GA in that the system is also initialised with a pool of random solutions and searches for optima by iteratively updating these candidate solutions. However, since PSO was initially devised to solve real-valued optimisation problems the potential solution is encoded as a real-valued string. Furthermore, because of the real-valued coding scheme PSO does not have a crossover operator which in contrast represents one of the main evolution operators in GA.

In PSO, the potential solutions, called particles, fly through the problem space by following the two current optimum particles: (1) each particle keeps track of its coordinates in the problem space which are associated with the best solution (fitness) it has achieved so far; (2) another "best" value that is tracked by the particle swarm optimiser is the best value, obtained so far by any particle in the neighbours of the particle; when a particle takes all the population as its topological neighbours, the best value is a global best. At each time step, PSO changes the velocity of each particle toward its two current optimum particles according to its distances to the two optimum particles. This separates random numbers being generated for acceleration toward the optimum locations.

2.4 Bio-Inspired Optimisation

Bio-inspired optimisation concerns mainly the way in which to extract useful metaphors from biology to provide sound structures for solving engineering problems. In the following subsections, two themes are explored, namely single-objective optimisation and multi-objective optimisation. The discussions are not restricted to a particular computing paradigm, such as the ones described in the previous sections. Rather, the intention is to expose the common features embedded in the modern heuristic based optimisation algorithms.

2.4.1 Single-Objective Optimisation (SOP)

Despite the apparent focus on MOP in this project SOP is the basis of all types of the optimisation. Hence, in this subSection, four important issues are addressed, which relate to the local search, the global search, the uni-modal optimisation and the multi-modal optimisation. Due to such different emphases, a special attention should be given to each individual, which brings various challenges to the design stage of a specific algorithm for any one of the aforementioned research directions. In the following discussion, only minimisation is considered without any loss of generality.

The most famous local search algorithms fall into the category of gradient-based optimisation. In this case, the search is directed by the derivative of the objective function. Since the direction always leads the candidate solution to the place which results in a smaller objective value than the previous position does, this type of optimisation represents obviously a local search mechanism. The disadvantage of such a mechanism is obvious: once the solution is trapped at a local minimum there is no way to come out of it. Another concomitant disadvantage lies in the fact that the objective function should be differentiable, although the gradient-based search algorithms can be fast and accurate. The Nelder-Mead Simplex search (Nelder & Mead, 1965) represents another type of the local search mechanism. It is derivative-free and can be categorised as the simplest version of the heuristic search method.

Due to its heuristic nature, any new move of the 'vertices' may not always minimise the objective function. Through the consecutive employments of *reflection*, *expansion*, *contraction* and *shrink*, all vertices of the simplex gradually converge to a locally optimal solution. The main difference between the Simplex and other stochastic search methods is that there is no mechanism in the design of the Simplex to ensure that the vertices escape from the local optimum.

In most situations, the search space contains several minima. Thus, a good balance between exploitation and exploration in the search space is the only insurance to locate the global area. For example, a legitimate step to extend Simplex to the global version is to restart the algorithm with a new simplex starting at the current best value. Restarting with a new simplex can be viewed as exploration, and preserving the current best value is a type of exploitation and elitism. In this sense, the above approach can be seen as the rudiment of a population-based GA, despite the fact that in the latter case a pool of individuals parallel searching for the optimum replaces the sequential restarting of the algorithm. By using a pool of individuals, a GA parallel explores more of the search space and thus increases the chance of finding the global optimum. Another mechanism to enhance the global search capability of GA is to utilise a mutation operator. Despite the great success in the early stage of the development of GA, single-point and two-point crossover operators are only feasible in the binary case. The stagnation of finding a feasible crossover operator in the real-valued situation is the main reason for the limitations of GA, which include premature convergence, slow convergence and low accuracy especially, when it is applied to continuous optimisation problems. Many research endeavors which specifically targeted at solving continuous optimisation problems have been proliferating. Most of these share some common features if one looks into the meaning behind their variation operators. More details can be found in Section 3.3.3.

In the presence of the coexistence of many local optima, designers are often more interested in obtaining as many local optima as possible rather than the global one, and in doing so they increase their choices for decision making. This is where the multi-modal optimisation can play an important role. Under this scenario, keeping the diversity of the candidate solutions plays a key role in preserving a set of solutions. In GA, this has been achieved by introducing the sharing method (Goldberg, 1989). In this way, different species can format and co-exist in the final population. However, two associated problems with the sharing method and GA are:

1) It is sensitive to the setting of the sharing parameters;
2) It depends highly on the population size when preserving the diversity of the population.

2.4.2 Multi-Objective Optimisation (MOP)

Many real-world problems are inherently of a multi-objective nature with often conflicting goals. Generally, MOP consists of minimizing/maximizing the following vector function:

$$f(x) = [f_1(x), f_2(x), \dots, f_m(x)]^T \tag{2.1}$$

subject to *J* inequality and *K* equality constraints as follows:

$$\begin{aligned} g_j(x) &\geq 0 \quad j = 1, \dots, J; \\ h_k(x) &= 0 \quad k = 1, \dots, K; \end{aligned} \tag{2.2}$$

where $x = [x_1, x_2, \dots, x_n]^T \in \Omega$ is the vector of decision variables and Ω is the feasible region. Classical methods that deal with MOP often use a higher-level of information about the problem to be optimised to choose a preference vector so that multiple objectives can be aggregated into a single objective. In doing so, MOP is actually transformed into a SOP. However, because of its high dependence on the preference information this approach is sometimes subjective and impractical. Facing the possibility of lacking the problem information, the idea of simultaneously finding a set of uniform-distributed optimal solutions through a single run, rather than several runs receives more and more attention. Bio-inspired optimisation algorithms are very ideal for the implementation of this idea due to the following reasons: first, they are population based search methods; second, they are derivative-free search methods; third, they effectively use previous knowledge.

Hitherto, many well-known implementations of this concept were proposed (Deb, 2001; Zitzler & Laumanns, 2001; Knowles & Corn, 2000; Jin, Olhofer & Sendhoff, 2001), and two text books (Deb, 2001; Coello Coello *et al.*, 2007) are available. One web repository http://www.lania.mx/~ccoello/EMOO/ is maintained by Dr. Carlos A. Coello Coello.

By carefully studying the differences of the existing MOP algorithms, it is possible to group them into three categories, viz. the Weighted-aggregation-based method, the Pareto-based method and the Archive-based method.

I. The idea of the *Weighted-aggregation-based method* is to randomly change weight combinations of the objectives during each generation so that the population can approach the different locations of the Pareto front. Schaffer's work-VEGA (Schaffer & Grefenstette, 1985), which is normally regarded as the first implementation of GA applied to MOP, falls into this family by implicitly performing a linear combination of the objectives where the weights depend on the distribution of the population at each generation. The problem associated with this method is that it suffers from the curse of 'concave Pareto front'. In such a situation, solutions tend to converge to some portion of the front rather than residing on the whole front.

II. The *Pareto-based method* relates mainly to how to assign fitness values to individuals in the population according to the concept of Pareto dominance. The first proposal was made by Goldberg (1989) as a means of assigning an equal probability of reproduction to all non-dominated individuals in the population. The method consisted of assigning ranks to the non-dominated individuals and removing them from contention, then repeating the same operations until the remaining population is empty. Following this idea, Deb *et al.* (2001) proposed the Non-dominated Sorting Algorithm II (NSGAII). In such an implementation, solutions were classified into different ranks according to the aforementioned procedures. Some newly developed features were included in NSGA-II, such as elitism. The algorithm employs a crowded tournament selection operator to keep diversity so that it does not need to specify any niching parameters. The main problem associated with NSGAII is that individuals with the same rank have the same fitness; in the later runs, all individuals will be classified in 'rank 1' and thus have the same fitness; in such a situation the selection pressure will diminish. SPEA2 was later proposed as an improved version of SPEA (Zitzler *et al.*, 2001). It distinguishes itself from other algorithms by using a different fitness assignment procedure, which for each individual takes into account how many individuals that it dominates and it is dominated by in the union of internal and external population; density estimation is also incorporated into the algorithm to calculate the fitness of each individual. In doing so, SPEA2 successfully resolved the problem associated with NSGAII as mentioned above. An enhanced archive truncation method was developed to guarantee the preservation of the boundary solutions and the diversity of the population. The main problem with SPEA2 is its high computational cost in the fitness assignment procedure and the archive truncation process. Both NSGAII and SPEA2 highly depend on their initial population size.

III. Knowles and Corne (2000) proposed a new baseline for approximating the Pareto front which can be viewed as the gestation of the *Archive-based method*. The Pareto Archived Evolutionary Strategy (PAES) was developed by introducing the concept of archiving. PAES does not use the dominance concept to carry out the fitness assignment. The Pareto concept is only used to compare the mutated individuals against the existing archive consisting of non-dominated solutions previously found. An adaptive grid partition method is applied to the archive to preserve the diversity of the solutions so far found. Traditionally, the weighted aggregation method could not solve the concave problem. However, using the concept of archiving to record any non-dominated solutions so far found, it is possible to find solutions on the concave front. Jin *et al.* (2001) discussed this issue and demonstrated how a single objective optimisation method combined with a dynamic change of a weighed aggregation plus an archive can deal with both convex and concave problems. The advantage of the archive-based method lies in the simplification of the fitness assignment procedure.

2.5 Artificial Neural Networks (ANNs)

ANNs are the simulation of the human brain and are constructed by connecting a set of artificial neurons in the form of a network. Hence, there are two important things in an ANN's configuration, i.e. the configuration of a single artificial neuron and the configuration of a network. A single artificial neuron is shown in Figure 2.3.

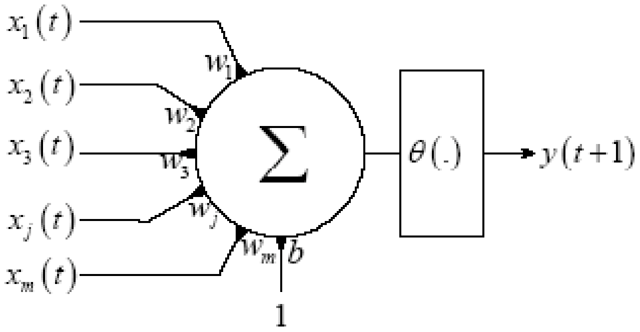


Figure 2.3 A simple artificial neuron.

The quantity, b, represents the threshold value and is represented as a constant signal of unity applied to an auxiliary input line with weight value b. The *activation function*, θ, is a design choice dependent on the task one wishes to solve. If the inputs are binary and the activation function is the unit step function it is called McCulloch-Pitts (MCP) unit. McCulloch and

Pitts showed (1943) showed that a network of MCP units can represent any logical function. If the activation function, θ, is simply the unit gain one call it the linear unit in this case. When one views it as a linear unit, then, the neuron's weights can be identified directly from the input-output data set using Least Squares algorithms (as long as the outputs are linear in the weights). If one has a non-linear relationship between input and output, the only task one needs to do before training an ANN is to pre-process the inputs so that they can be applied to the input lines of the neuron. This type of process represents in fact polynomial expansion. By taking a high enough degree of polynomial one can approximate any function that is smooth enough as accurately as one wishes (the Stone-Weierstrass Theorem (Stone, 1948)). Hence, using a single linear unit we can approximate a wide range of functions.

The problem with the linear unit is that if one wants more accuracy of an approximation one should take a relatively high degree of the polynomial expansion and thus more weights to be identified. To circumvent this problem the non-linear unit was proposed to replace the activation function to a non-linear function. By doing so, the outputs is non-linearly related both to the inputs and, more importantly, to the weights and Least Squares algorithms no longer apply to this case. The gradient descent method can resolve this problem based on the fact that one can use Mean Squared Error (MSE) as the cost function. The gradient descent method works as follows: change the weight values ($\vec{w}$) at each iteration step so that the weight vector always leads the cost to the extreme. The mathematical formation is as follows (refer also to Section 5.4):

$$\frac{\partial \vec{w}(t)}{\partial t} \propto \frac{\partial \mathrm{J}(\vec{\mathrm{w}})}{\partial \vec{w}(t)}; \qquad \text{where } \vec{\mathrm{w}}(0) = \vec{w}^{0} \tag{2.3}$$

where $\vec{w}^{0}$ determines the starting point on the convex error (hyper-) surface and t represents the current iteration.

However, a single adaptive non-linear unit can only represent sigmoidal functions of its input. The real power comes just as for MCP unit, when a network of logistic units is connected together with each other and this gives rise to the Multi-layer Perceptron (MLP) that is used in this project. The MLP is a layered network of units that learn to solve a wide variety of tasks that can be described as “finding mappings from *n*-dimensional input-space into *q*-dimensional output-space’. It consists of one input layer, several hidden layers and one output layer. MLP is strictly feed-forward. To update the weights in each layer the Back-Error-Propagation (BEP) algorithm (refer to Section 5.4), is adopted. It is nothing more than a gradient descent method in spite of the fact that now one may be faced with two fundamentally different situations: (1) Computation of the gradient with respect to weights in the output layer; (2) Computation of the gradient with respect to weights in any of the hidden layers.

In 1990, Cybenko (1989) proved that an MLP with only one sigmoidal hidden-layer and a linear output layer is a universal approximator in that it can approximate any well-behaved function to arbitrary accuracy provided there are enough hidden units. Comparing with two-layer network with the polynomial expansion approach one will find the hidden-layer is a sort of pre-processor but here the pre-processing function is found by the learning algorithm rather than being fixed *a priori*. By doing so, one can get a far more compact model.

2.6 Fuzzy Logical Theory and Fuzzy Rule Based Systems (FRBS)

In the real world, there are numerous systems which contain extremely non-linear, time-varying and uncertain behaviour. These make the development of computerised systems for them not a straightforward algorithmic solution because of the inherent uncertainty which arises as a natural occurrence in these types of applications. The human operator can often be an adequate controller by being able to construct acceptable models of processes in his/her mind. Such models which do not include any mathematical equations are therefore easier to handle. In other words, the human operator has the ability to interpret linguistic statements about the process and to think in a qualitative fashion rather than in a quantitative one. Fuzzy logic theory is indeed inspired from these observations and first introduced by Zadeh (1965). The main advantage of fuzzy systems is that they can combine human expertise together with sensory measurements and mathematical models.

2.6.1 Fuzzy Logic Theory

Fuzzy logic can easily be introduced via the concepts of a fuzzy set. A fuzzy set is a set without a crisp, clearly defined boundary. It can contain elements with only a partial degree of membership. In other words, a fuzzy set is a class of objects with a continuum of grads of membership (Zadeh, 1965). A formal definition of fuzzy set is given by Zadeh (1965) as follows:

> *"Let X be a space of points (objects), with a generic element of X denoted by x. Thus, $X = \{x\}$. Then, a fuzzy set (class)A in X is characterised by a membership (characteristic) function $f_A(x)$ which associates with each point in X a real number in the interval $[0, 1]$, with the value of $f_A(x)$ at x representing the 'grade of membership' of x in A."*

Where the *membership function* is a curve that defines how each point in the input space is mapped to a membership value (or degree of membership) between 0 and 1. The value of the membership function determines whether the element belongs to the fuzzy set A and if so, to what degree. It determines a degree of certainty i.e. degree of truth.

The main difference between classical (crisp) sets and fuzzy sets is the way their membership functions take values. The value of the membership function of a classical set can only take two values, either 0 or 1. So, it is discrete and can only represent black or white. These two kinds of sets are illustrated in Figure 2.4, as well as one commonly used membership function - the Gaussian membership function in Figure 2.5, which is a smooth function and can introduce extra smoothness and is used in this project.

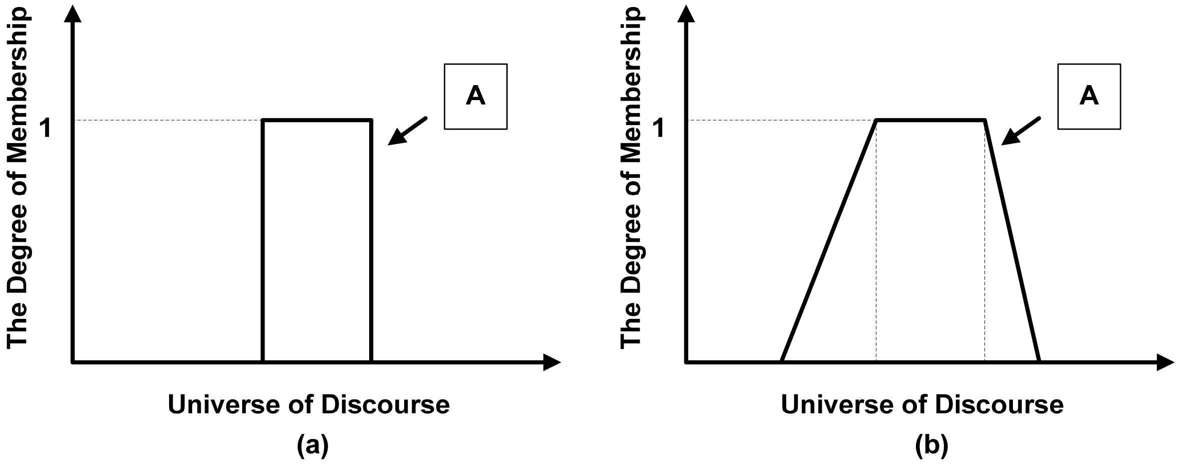


Figure 2.4 (a) representation of a classical set; (b) representation of a fuzzy set.

The Gaussian membership function that was used in this project has the formula of follows:

$$\mu_A(x) = \exp\left(-\frac{1}{2} \cdot \frac{(x - center)^2}{\sigma^2}\right) \tag{2.4}$$

Where 'centre' denotes the centre of the bell-shape curve and σ denotes the standard deviation.

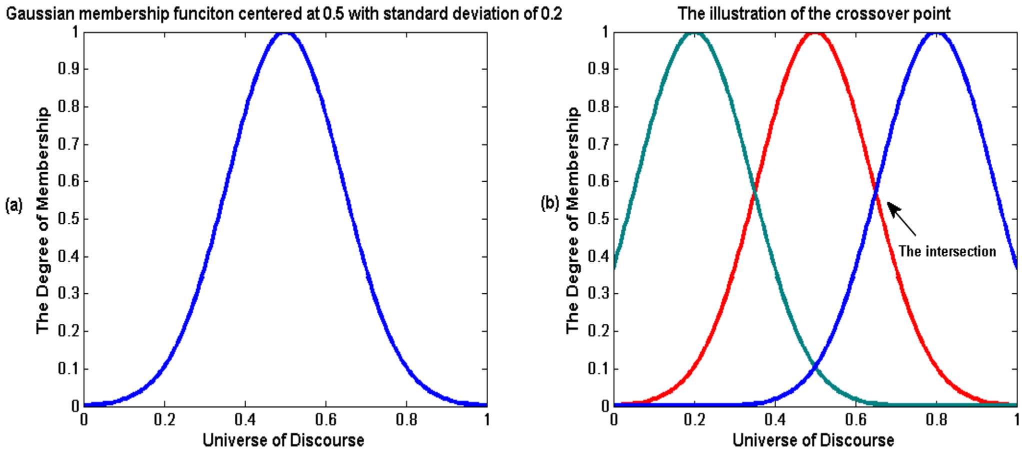


Figure 2.5 (a) the shape of a Gaussian membership function; (b) the illustration of the crossover points.

The importance of the intersection, as shown in Figure 2.5 (b), is that it marks that point at which, for a particular membership function, the certainty of belonging changes. For membership degrees higher than the crossover point the 'certainty' of belonging to a particular membership function is higher than the certainty of not belonging. For membership degrees lower than the crossover point the opposite happens for the certainty of belonging. For different applications, the crossover point can vary between 0.2 and 1.

To complete the fuzzy logic theory, Zadeh also introduced a set of basic operators that can be viewed as extensions of the corresponding definitions for ordinary sets. These operators consist of *Containment*, *Union, Intersection, Complement* and *Cartesian product.* The following part mainly addresses the *Cartesian product* since it represents a relationship between fuzzy variables and paves the basis for fuzzy inference.

In fact, many application problem descriptions include fuzzy relations. A fuzzy system is usually represented by statements or rules of the following form:

$$If\ A\ then\ B\ or\ A \rightarrow B$$

A fuzzy relation of the form $A \rightarrow B$ is denoted R and is defined as a relationship of two fuzzy sets $A\epsilon U, B\epsilon V$ and it is a subset on the *Cartesian product*. U and V can be the same or different universes of discourse. R will be characterised by the membership function $\mu_R(u,v), where\ u\epsilon U\ and\ v\epsilon V$ such that:

$$R = A * B = \sum \mu_R(u,v)/(u,v) = \begin{cases} \sum \min\,(\mu_A(u), \mu_B(v)) \\ \quad \sum \mu_A(u) \cdot \mu_B(v) \end{cases} \tag{2.5}$$

Where R is also called the relational matrix and the 'sum' does not represent a mathematical operation but shows rather all possible combinations of all elements of both universes of discourse. The *Cartesian product* can be extended to a product of more than two sets, e.g. if C is also a fuzzy set of the universe of discourse w, then:

$$R = A * B * C = \sum \mu_R(u,v,w)/(u,v,w) = \begin{cases} \sum \min\left(\mu_A(u), \mu_B(v), \mu_C(w)\right) \\ \sum \mu_A(u) \cdot \mu_B(v) \cdot \mu_C(w) \end{cases} \tag{2.6}$$

Having defined the fuzzy relations and *Cartesian product* one can use it to interpret fuzzy conditional statement such as:

$$\begin{aligned} & If\ A1\ then\ \left(if\ B1\ then\ \left(if\ C1\ then\ (if\ \dots)\right)\right) \\ Else\ & If\ A2\ then\ \left(if\ B2\ then\ \left(if\ C2\ then\ (if\ \dots)\right)\right) \\ & \dots \qquad\qquad\qquad\qquad \dots \\ \equiv\ & (A1 * B1 * C1 \dots) \oplus (A2 * B2 * C2) \oplus \dots \end{aligned} \tag{2.7}$$

Where, 'Else' is equivalent to the connective OR. Once the conditional statements have been made, one can infer relevant information from these statements as follows:

Let R be a fuzzy relation in $U * V$ and A is a fuzzy set in U then the fuzzy set B in V is given by:

$$B = A \circ R \tag{2.8}$$

B is inferred from A using the relational matrix R which defines the mapping between U and V, and the operation $\circ$ is defined as the "max-min" operation. The above process is called as *Compositional rule of inference* and forms the basis of one type of fuzzy inference system which is discussed in the following section.

2.6.2 Fuzzy Inference Systems and FRBS

Fuzzy inference is the process of formulating the mapping from a given input to an output using fuzzy logic. The mapping then provides a basis from which decisions can be made, or patterns discerned. There are many types of fuzzy inference systems (FRBSs). The two most popular types of fuzzy inference systems are the Mamdani-type (Mamdani, 1974) and Sugeno-type (Takagi *et al.*, 1985). These two types of inference systems vary somewhat in the way outputs are determined. The consequence part of the Mamdani-type is a fuzzy set while the consequence part of the Sugeno-type is a set of functions with the arguments that are the linguistic variables of the antecedent part.

Mamdani's method was based on Zadeh's fuzzy algorithms for complex systems (Zadeh, 1973). It first calculates the relational matrix for each rule, and then calculates the overall relational matrix (also called the overall implied fuzzy set; refer to Section 5.3.1 for more details). Finally, by using the *Composition rule of inference* the output fuzzy set can be found. This type of inference is easily understandable by human experts and the rules are easier to formulated and maintained. However, the inference based on the overall implied fuzzy set has an inherent drawback which is computationally expensive and more importantly is not differentiable with respect to membership function parameters. As will be discussed in Section 5.3.1, this leads to the problem of not being able to use the BEP algorithm to refine fuzzy models. Hence, a modified Mamdani inference system is proposed in Section 5.4.2, which can get around the aforementioned problems.

Another form of fuzzy inference, proposed by Takagi and Sugeno (1985), has fuzzy sets involved only in the premise part. By using Takagi and Sugeno's fuzzy inference scheme (TSK), one can describe the fuzzy if-then rule as follows:

$$R_i\!: If\ x_1\ is\ A_i^1\ and\ x_2\ is\ A_i^2, \dots\ , and\ x_j\ is\ A_i^j\ Then\ y_i = g_i(x_1, x_2, \dots, x_j) \tag{2.9}$$

R_i denotes the ith rule to be concerned. 'g_i' is a function which can be the linear combination or quadratic. Normally, using linear combination for 'g_i' is enough since the fuzzy if-then rule has already embedded non-linear inherently. If a linear model structure is assumed then a rule base with k rules takes the following format:

$$\begin{array}{c} R_1\!: If\ x_1\ is\ A_1^1\ and\ x_2\ is\ A_1^2, \dots\ , and\ x_j\ is\ A_1^j\ Then\ y_1 = b_1^0 + b_1^1 \cdot x_1 + \cdots + b_1^j \cdot x_j \\ \dots \\ R_k\!: If\ x_1\ is\ A_k^1\ and\ x_2\ is\ A_k^2, \dots\ , and\ x_j\ is\ A_k^j\ Then\ y_k = b_k^0 + b_k^1 \cdot x_1 + \cdots + b_k^j \cdot x_j \end{array} \tag{2.10}$$

$$Let\ \beta_i = \frac{\mu_{A_i^1}(x_1) * \mu_{A_i^2}(x_2) * \dots * \mu_{A_i^j}(x_j)}{\sum_{i=1}^{k}\left(\mu_{A_i^1}(x_1) * \mu_{A_i^2}(x_2) * \dots * \mu_{A_i^j}(x_j)\right)} \tag{2.11}$$

Where, β_i actually represents the certainty of each rule contributed by the premise of corresponding rule. The output from the input $(x_1, x_2, .., x_j)$ is obtained as follows:

$$y = \sum_{i=1}^{k}\left(b_i^0 \cdot \beta_i + b_i^1 \cdot x_1 \cdot \beta_i + \cdots + b_i^j \cdot x_j \cdot \beta_i\right) \tag{2.12}$$

When a set of input-output data is given, one can obtain the consequent parameters $(b_i^0, b_i^1, .., b_i^j), (i = 1, \ldots, k; j = 1, \ldots, n)$ via some learning algorithms, such as those introduced in Sections 2.1~2.5. Hence, the method as proposed by Takagi and Kang involves an iterative search to determine: (1) the best model structure; and (2) the best membership function parameters. In Sections 4.4.2 and 5.4.1, a special case of the TSK FRBS, namely Singleton FRBS, is employed due to its simple and interpretable structure.

The general process of the fuzzy inference and its schematic diagram is shown in Figure 2.6. The 'rule base' contains a number of fuzzy if-then rules and the 'database' defines the membership functions of the fuzzy sets used in the fuzzy rules. Usually, the rule base and the database are jointly referred to as the 'Knowledge base'. The 'decision-making unit' performs the inference operations on the rules and two interfaces perform fuzzification and defuzzification respectively.

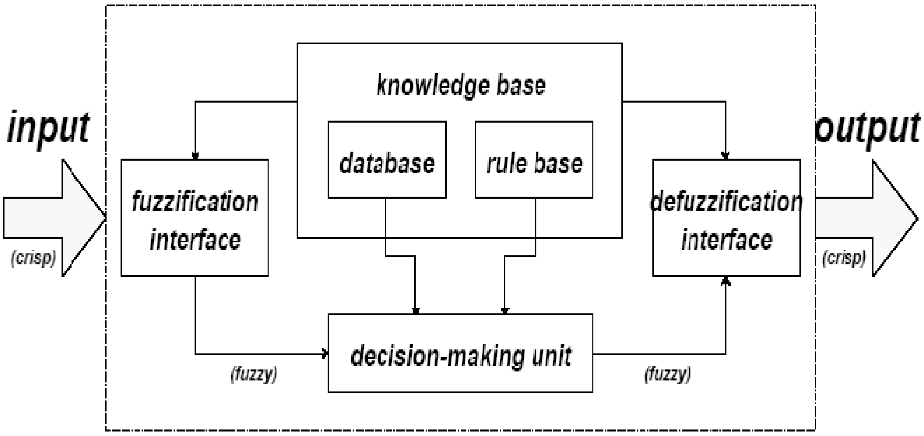


Figure 2.6 Fuzzy Inference Systems (FRBS) (Jang, 1993).

2.6.3 Neuro-Fuzzy Systems (NFS)

NFS refers to combinations of ANNs and FRBSs, which results in a hybrid intelligent system that synergizes these two techniques by combining the human-like reasoning style of fuzzy systems with the learning and connectionist structure of neural networks. NFS incorporates the human-like reasoning style of fuzzy systems through the use of fuzzy sets and a linguistic model consisting of a set of IF-THEN fuzzy rules. The main strength of NFS is that they are universal approximators (Kosko, 1994), and at the same time they still include the ability to explicitly express the embedded knowledge. Jang and Sun (1993) established the functional equivalence of a standard Gaussian radial basis function (RBF's) networks and a restricted form of TSK fuzzy inference model. The most visible work belongs to ANFIS proposed by Jang (1993). In Chapter 5, some of these techniques are employed to find an accurate enough fuzzy model in the first place.

2.7 Summary

In this chapter, some bio-inspired techniques, such as EC, AIS, PSO, ANNs and FRBS, were introduced. The techniques described in this chapter are by no means exhaustive. Other bio-inspired mechanisms still exist, such as ant colony optimisation and membrane computing. The selected methods are the ones which most relate to the project. Apart from the description of each technique, some common features carried nowadays by most modern heuristic based optimisation algorithms are also explored. In the following chapters, the implementations of the mentioned techniques are presented. In particular, Chapter 3 describes a population adaptive based immune algorithm for solving MOP.

Chapter 3

A Population Adaptive Based Immune Algorithm

“How can computers be programmed so that problem-solving capabilities are built up by specifying ‘what is to be done' rather than ‘how to do it'?"

John H. Holland, Adaptation in Natural and Artificial System, 1975

The primary objective of this chapter is to introduce Artificial Immune Systems (AIS) as a relatively new bio-inspired optimisation technique and to show its appeal to engineering applications. To this aim, a novel Population Adaptive Based Immune Algorithm (PAIA)[3.1] inspired by four immunological models for solving multi-objective optimisation problems (MOP) is proposed.

The algorithm is shown to be insensitive to the initial population size; the population and clone size are adaptive with respect to the search process and the problem at hand. It is argued that the algorithm can largely reduce the number of evaluation times and is more consistent with the vertebrate immune system than the previously proposed algorithms. Results suggest that the algorithm is a valuable alternative to already established evolutionary based optimisation algorithms, such as NSGAII (Deb, 2001), SPEA2 (Zitzler, Laumanns, Thiele, 2001) and VIS (Freschi and Repetto 2005).

Such promising results further formed the basis for the extraction of a general framework from the PAIA as the guide to design immune algorithms, under which clear definitions of immune operators and their roles are provided.

[3.1] Different versions of PAIA are not discriminated at this stage. However, it will be more specific and clear when one approaches the corresponding context.

3.1 AIS Based Optimisation

As mentioned in Section 2.2, there are three main application domains which most AIS research efforts have hitherto focused on, viz. fault diagnosis, computer security, and data analysis, although the main application areas in this thesis are optimisation and modelling. The reason behind this is that it is relatively easy to create a direct link between the real immune system and the aforementioned three application areas. However, as already stated in Chapter 2, such links are vague in the field of optimisation. Furthermore, as pointed by Emma Hart and Jonathan Timmis (2005), the distinction line between Evolutionary Computing and AIS is also fuzzy. Hence, in the following two sections, the current state of the AIS-based optimisation is first reviewed, which leads to the desiderate questions to be solved.

3.1.1 Current State

It is worth noting that most AIS-based research which relates to SOP identified the diversity of the population as the main advantage of AIS over conventional evolution algorithms and the slow convergence as its drawback. In the early days, AIS was mainly integrated into other evolutionary algorithms to overcome the well known problem of premature convergence in searching for the global optimum. In these developments, GA was combined with AIS to model the somatic mutation and gene recombination by means of two GA operators, viz. crossover and mutation in order to maintain the diversity of the population (Wang, Gao & Ovaska, 2004). More recently, some newly developed optimisation algorithms, which are solely based on the immune mechanisms, were proposed. Most of these algorithms establish the single objective multi-modal optimisation problem as a target. It is the diversity that gestates the motivation of using the immune based algorithms for solving multi-modal problems. Fukuda *et al.* (1998) proposed an Immune Algorithm (IA) which is based on the somatic theory and network hypotheses. The somatic theory contributes to increasing the diversity of antibodies and as a result to increasing the possibility of finding a global solution as well as local optimal solutions. The network hypotheses contributes to the control of the proliferation of clones. An Optimisation version of artificial immune network model (Opt-aiNet) is an augmented version of the Clonnal selection Algorithm (ClONAG) (de Castro & Von Zuben, 2002; de Castro & Timmis, 2002) by combining Network hypothesis with the Clonal Selection Principle. The main features of Opt-aiNet include dynamically adjustable population size, balance between exploitation and exploration of the search space and the ability of locating the multiple optima.

Freschi and Repetto (2005) argued that AIS has, in its elementary structure, the main features required to solve MOP. Their vector artificial immune system (VIS) is mainly based on immune network theory. Unlike other immune algorithms, the clonal selection of the fittest antibodies is not based on the calculation of affinity, instead, it is based on a ranking scheme which is a modified version of the scheme adopted by SPEA2. The diversity of the antibody's population is maintained via the network suppression and the newcomers inserted in the outer-loop. Since in the clonal selection step, the best mutated clone for each cell replaces the original parent rather than selecting the best mutants from the union of all parents and clones, the speed of convergence of this algorithm may be slower than that of the one adopting the latter selection scheme. Coello Coello *et al.*'s Multi-objective Immune System Algorithm (MISA) (2005) mainly takes ideas from the Clonal Selection Principle. Antibodies in their algorithms are encoded into binary strings. The algorithm sacrifices some biological metaphors in exchange for a better performance. There is no explicit affinity calculation within the algorithm, and thus both the selection and clone processes cannot be based on it. Apart from this, due to the binary encoding scheme, both the convergence and accuracy are deteriorated when the same algorithm is used to deal with continuous optimisation problems. Both the aforementioned algorithms fix the number of clones that each parent can proliferate. Pareto-optimal Neighbour Immune Algorithm (PNIA) (Gong *et al.*, 2006) adopts a new way in defining affinity. The fitness (affinity) is calculated according to the crowding-distance proposed in NSGAII and is only assigned to the dominant individuals. Clone, recombination and hypermutation are only applied to the dominant individuals. Non-dominated selection is performed on the union of all kinds of the population. The clone size in this algorithm is adaptively determined by the corresponding affinity. Due to the selection scheme adopted in PNIA, both the convergence speed and the accuracy are improved. It seems that in PNIA there is no explicit diversity mechanism except that the over-crowed antibodies are removed from the population. It is worth noting that the population size in all these three algorithms is fixed.

3.1.2 Call for Solutions

From the discussions of the last section, the following problems associated with the already developed AIS-based MOP algorithms are exposed.

- ✧ No formal systematic framework (each algorithm has its own structure and is very much different from the others).
- ✧ Not consistent with the immune mechanisms, e.g. 1) the clone size in most algorithms is fixed; however, in real immune systems, the clone size is proportionate to the corresponding affinity; 2) the population size is fixed; however, in real immune systems, the 'Ab''s concentration is adaptively changing.
- ✧ Not effectively using the information from the decision variable space. In most cases affinity is only related to the dominance of each solution in the objective space.
- ✧ The already developed AIS-based MOP algorithms are coupled with other evolutionary mechanisms.

Apart from the above problems and based on the description in Section 2.1 and 2.4, generally speaking, one can also easily identify the following disadvantages related to the existing evolutionary algorithms with the GA as their representative.

- ✧ Premature convergence and low accuracy.
- ✧ The population size is problem-contingent and crucial for the search capability.
- ✧ Slow convergence.
- ✧ The sharing parameters are problem-dependent (not generic enough).

All the previous considerations justify the 'rationale' behind the PAIA (Chen and Mahfouf, 2006; 2008a). PAIA is the synthesis of the four immune metaphors, where the Clonal Selection Principle is used to provide a selection pressure to effectively drive the population towards the Pareto front over many iteration steps; the Network Theory is used to regulate the dynamics of the population; the adaptive antibody's concentration is the main inspiration for the design of the PAIA's structure so that the population is adaptive at each iteration step; and the vaccination and the secondary response is used to develop the so-called 'multi-stage' optimisation. The aims of PAIA are:

1) providing a generic AIS framework for MOP solving;
2) making the population size adaptive to the problem;

3) reducing the number of evaluation times so that only the necessary evaluations are carried-out (speed up the convergence).

The detailed steps of the PAIA are described in the next section.

3.2 A Population Adaptive Based Immune Algorithm (PAIA)

3.2.1 Description of the PAIA

The terms and definitions used in this subsection can be found in Section 2.2. The PAIA can be described via the following steps:

1. **Initialisation:** a random 'Ab' population is first created.

2. **Identify_Ab:** one random 'Ab' $(x_{identified})$ in the first non-dominated front is identified.

3. **Activation:** the identified 'Ab' is used to activate the remaining dominated 'Abs' (x_d). The dominated 'Abs' affinity value (NB: affinity is the inverse of affinity value) is calculated according to Eq. 3.1, where *n* is the dimension of the decision variables.

$$aff\_val_d = \frac{\sum_{i=1}^{n}\left(x_{identified}(i)-x_d(i)\right)}{n} \tag{3.1}$$

The non-dominated 'Abs' affinity value is calculated as follows: **I.** if the size of dominated 'Abs' is not zero, the affinity value equals the minimum affinity value of the dominated 'Ab' divided by two; **II.** otherwise, the affinity value is calculated according to Eq. 3.2, where *N* is the size of non-dominated 'Abs'.

$$aff\_val_{nd} = \sum_{j=1}^{N} \frac{\left(\sum_{i=1}^{n} \frac{\left(x_{identified}(i)-x_j(i)\right)}{n}\right)}{N} \tag{3.2}$$

In this way, the 'Ag-Ab' affinity is indirectly embedded in 'Abs' affinity since the non-dominated 'Abs' always have the smallest affinity value (the highest affinity).

4. **Clonal Selection:** Clonal selection consists of three steps: **I.** 'Abs' with the smallest affinity value are selected, i.e. the non-dominated Abs are always selected; **II.** The

'Abs' in the remaining population with an affinity value smaller than a threshold (δ) are selected; **III.** the unselected 'Abs' are kept in a different set.

5. **Clone: I.** for the selected 'Abs', a maximum clone size (N_{cmax}) is pre-defined; then a fraction of N_{cmax} is allocated to each selected 'Ab' according to its affinity percentage, i.e. the higher the percentage the larger the fraction is assigned; **II.** Unselected 'Abs' are cloned once regardless of their affinity.

6. **Affinity Maturation: I.** the selected 'Abs' are subjected to *hypermutation*, i.e. one dimension of the 'Ab' is randomly chosen to mutate; the mutation rate is proportional to the affinity value (inversely proportional to affinity); the whole process is calculated using Eq. 3.3. **II.** the unselected 'Abs' are submitted to *receptor editing* which means more than one dimensions (two, in PAIA) are randomly chosen to mutate; the mutation rate is calculated using Eq. 3.3.

$$x_{new}(i) = x_{old}(i) + \alpha \cdot N(0,1) \; i = 1, \dots, n; \; \alpha = \frac{exp(aff\_val)}{exp(1)} \tag{3.3}$$

where *N(0, 1)* is a Gaussian random variable with zero mean and standard deviation 1. *i* represents the dimension that has been chosen to mutate.

7. **Reselection:** the mutated/edited and their corresponding parents are mixed together and reselected: **I.** all non-dominated 'Abs' are selected; **II.** if the number of current non-dominated 'Abs' (NCR) is less than the initial population size (IN), the 'Abs' from the next non-dominated front are selected according to their recalculated 'Abs' affinity value (the ones with smaller affinity values are favoured) to fill the difference between these two; this process continues until the difference is filled; **III.** only when NCR is greater than IN and the number of non-dominated 'Abs' in the last iteration (NPR) can *Network Suppression* be invoked to suppress too-close 'Abs'.

8. **Network Suppression:** the *Euclidian* distance in objective space between any two 'Abs' is calculated; if it is less than a predefined network threshold (σ) the one with larger affinity value is suppressed and deleted; this operator is invoked in step 7 when certain conditions are satisfied.

9. **Iteration:** the process is repeated from step 2 until certain conditions are met.

In the following sections, the performance metrics and the ZDT test suites are first introduced so that adequate comparisons between the proposed algorithm and other well-known algorithms are suitably carried out.

3.2.2 Performance Metrics

Two performance metrics, namely the Generational Distance (GD) and the Spread Δ are employed to evaluate the convergence and distribution of the final solutions, which are defined as follows.

- **Generational Distance:** GD measures the closeness of the obtained Pareto solution set Q from a known set of Pareto-optimal set P^*.

$$GD = \frac{\left(\sum_{i=1}^{|Q|} d_i^m\right)^{1/m}}{|Q|} \qquad (3.4)$$

  For a two-objective problem (m=2), d_i is the *Euclidean* distance between the solution $i \in Q$ and the nearest member of P^*. A set of $|P^*|$=500 uniformly distributed Pareto-optimal solutions is used to calculate GD.

- **Spread:** Δ measures the diversity of the solutions along the Pareto front in the final population. where d_i is the distance between the neighbouring solutions in the Pareto solution set Q. $\bar{d}$ is the mean value of all d_i. d_m^e is the distance between the extreme solutions of P^* and Q along the *m*th objective. It is worth noting that for discontinued problems, such as ZDT3 described in Section 3.2.3, Δ is calculated in each continuous region and averaged as follows.

$$\Delta = \frac{\sum_{m=1}^{M} d_m^e + \sum_{i=1}^{|Q|} |d_i - \bar{d}|}{\sum_{m=1}^{M} d_m^e + |Q| \cdot \bar{d}} \qquad (3.5)$$

3.2.3 Preliminary Results on ZDT Series Problems

In this section, the PAIA is compared with two well-known algorithms-NSGAII and SPEA2, and one immune-based algorithm-Vector Immune Algorithm (VIS). By comparing with NSGAII and SPEA2, it is shown that the PAIA is a valuable alternative to standard algorithms; by comparing PAIA with VIS, the difference between these two immune algorithms is identified; Table 3.1 defines the ZDT series problems (Deb, 2001). The ZDT series problems have two objectives and represent the same type of problems with a large decision variable space, a concave and discrete Pareto front, many local optima and variable density of the decision variable space and the objective space. At this stage, only the test functions ZDT1~ZDT4 are considered.

For such comparisons to be fair, the experimental configuration refers to the experiments in Deb's book (2001). The maximum function evaluation for NSGAII and SPEA2 is set to 25000. For PAIA, although the population is adaptive the final population can be controlled by σ. Hence, one can set an adequate value for σ so that the final population size and evaluation times are around 100 and 25000 respectively. To make the comparison fair, VIS is also run using the same setting for PAIA. NSGA II failed to converge for ZDT4 even with a larger number of evaluation times, while on the other hand, although PAIA and VIS may not fully converge within 25000 evaluations they had no difficulty to converge using larger evaluations. For this reason, one can also compare PAIA and VIS when both have fully converged. Hence, in the following space of this section, two experiments are conducted, with the first one (Experiment 1) concentrating on the comparison of the results obtained using 25000 evaluation times and the second one (Experiment 2) focusing on the comparison of the results obtained when PAIA and VIS have fully converged.

TABLE 3.1
ZDT SERIES PROBLEMS (DEB, 2001)

Problems	Definition
ZDT1	30-variable problem with a convex Pareto front. $f_1 = x_1, g = 1 + \frac{9}{n-1} \cdot \sum_{i=2}^{n} x_i, f_2 = g \cdot \left(1 - \sqrt{f_1/g}\right), 0 \le x_i \le 1, n = 30$
ZDT2	30-variable problem with a concave Pareto front. $f_1 = x_1, g = 1 + \frac{9}{n-1} \cdot \sum_{i=2}^{n} x_i, f_2 = g \cdot (1 - (f_1/g)^2), 0 \le x_i \le 1, n = 30$
ZDT3	30-variable problem with disconnected Pareto fronts. $f_1 = x_1, g = 1 + \frac{9}{n-1} \cdot \sum_{i=2}^{n} x_i, f_2 = g \cdot \left(1 - \sqrt{f_1/g} - (f_1/g) \cdot \sin(10 \cdot \pi \cdot f_1)\right),$ $0 \le x_i \le 1, n = 30$
ZDT4	10-variable problem with 100 local Pareto fronts. $f_1 = x_1, g = 1 + 10 \cdot (n-1) + \sum_{i=2}^{n} (x_i^2 - 10 \cdot \cos(4 \cdot \pi \cdot x_i), f_2 = g \cdot \left(1 - \sqrt{f_1/g}\right),$ $n = 10, 0 \le x_i \le 1, -5 \le x_i \le 5, i = 2,3, \dots, n$
ZDT6	10-variable problem with a concave and non-uniform distributed Pareto front. $f_1 = 1 - \exp(-4 \cdot x_1) \cdot sin^6(6 \cdot \pi \cdot x_1), g = 1 + 9\left(\frac{(\sum_{i=2}^{n} x_i)}{9}\right)^{0.25},$ $f_2 = g \cdot (1 - (f_1/g)^2), 0 \le x_i \le 1, n = 10$

❖ Experiment 1 (25000 Evaluations)

This experiment is designed to compare the performance of different algorithms using 25000 evaluation times. The parameter settings for different algorithms are as follows:

- **SPEA2:** Population size 100, archive size 100, mating pool size 100, the distribution index for crossover 20, the distribution index for mutation: 20, maximum generation 250; crossover probability 1 and mutation probability 1/ *(the dimension of the decision variable).*

- **NSGA II:** Population size 100, maximum generation 250, crossover probability 0.9 and mutation probability 1/(string-length). 30 bits are used to code variable.

- **PAIA:** IN=7, δ=0.4, N_{cmax}=95, σ=0.005 for ZDT1~ZDT4.

Figure 3.1 shows the results found by PAIA. The results of SPEA2 and PAIA shown in Tables 3.2 and 3.3 are the average values over 10 independent runs. The results of NSGAII are from Deb (2001) and the results of VIS are from Chen and Mahfouf *et al.* (2006).

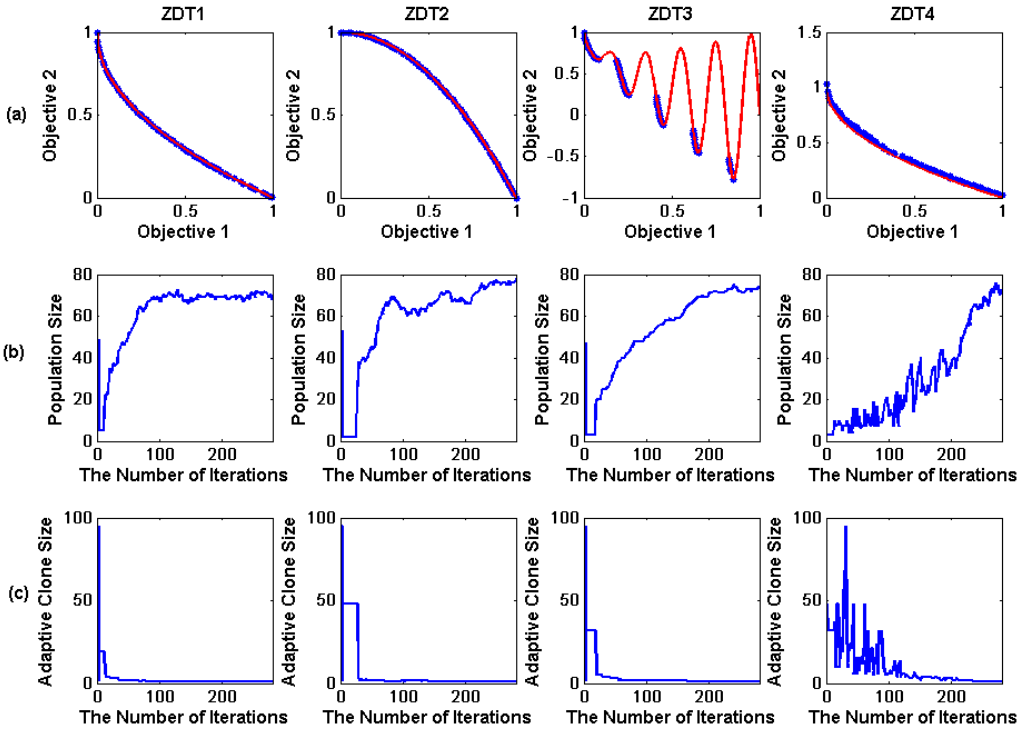


Figure 3.1 (a) Pareto solutions obtained by PAIA on ZDT1~ZDT4; (b) Adaptive population size vs. iteration; (c) Adaptive clone size (the assigned maximum clone size among all Abs) vs. iteration.

TABLE 3.2

MEAN AND VARIANCE VALUES OF THE CONVERGENCE MEASURE GD FOR ZDT SERIES PROBLEMS

Algorithm	ZDT1		ZDT2		ZDT3		ZDT4	
	GD	σ^2	**GD**	σ^2	**GD**	σ^2	**GD**	σ^2
NSGAII	8.94e-4	0	8.24e-4	0	4.34e-2	4.20e-5	3.228	7.3076
SPEA2	2.64e-4	4.68e-10	1.05e-4	1.21e-11	1.69e-4	1.74e-10	**4.68e-4**	1.36e-8
VIS	1.81e-3	1.97e-7	1.21e-3	1.04e-6	1.58e-3	2.26e-7	0.1323	4.20e-2
PAIA	**1.43e-4**	1.56e-9	**1.04e-4**	2.2e-11	**1.58e-4**	4.6e-10	1.20e-3	1.88e-7

TABLE 3.3

MEAN AND VARIANCE VALUES OF THE DIVERSITY MEASURE Δ FOR ZDT SERIES PROBLEMS

Algorithm	ZDT1		ZDT2		ZDT3		ZDT4	
	Δ	σ^2	Δ	σ^2	Δ	σ^2	Δ	σ^2
NSGAII	0.4633	4.16e-2	0.4351	2.46e-2	0.5756	5.08e-3	0.4795	9.84e-3
SPEA2	**0.1575**	1.44e-4	**0.1523**	1.31e-4	**0.1638**	2.90e-2	**0.1555**	4.34e-4
VIS	0.5420	8.25e-3	0.6625	2.58e-2	0.6274	1.60e-2	0.1011	1.37e-3
PAIA	0.3368	1.10e-3	0.3023	7.07e-4	0.4381	1.50e-3	0.3316	1.20e-3

TABLE 3.4

FINAL POPULATION SIZE AND EVALUATION TIMES OF PAIA

Test suite	**Final Population**		**Evaluation Times**	
	Mean	**Max/min**	**Mean**	**Max/min**
ZDT1	96	101/87	25372	26467/24494
ZDT2	101	106/96	25950	26649/25371
ZDT3	94	102/89	25365	26155/24587
ZDT4	96	103/85	25910	26654/25203

Results shown in Tables 3.2~3.4 indicate that PAIA reached a better performance in terms of GD for ZDT1~ZDT3 problems than any of other three algorithms using a similar number of evaluation times. From Figure 3.1 (b), it can be seen that the population adaptively increases/decreases during each iteration step and can be finally controlled by σ, which means that only necessary 'Abs' are maintained during and at the end of the search. From Figure 3.1 (c), it can also be seen that the clone size is adaptively decided by the number of selected 'Abs' and their corresponding affinity. If the number of selected 'Abs' is small, each selected 'Ab' can be assigned a large clone size so that the population is large enough to explore the objective space. Although the results of PAIA for ZDT4 are much better than for NSGAII and VIS, it has not fully converged to the 'true' Pareto front. It is worth noting that this result can be further improved by using more iteration steps and such results will be described by Experiment 2 next.

❖ **Experiment 2 (full convergence)**

In this experiment, the number of iterations was 180 for ZDT1 and ZDT2, 280 for ZDT3 and 500 for ZDT4. Other parameters are similar to those of Experiment 1. Figure 3.2 shows the Pareto front obtained by PAIA, and Tables 3.5 and 3.6 summarise the results over the ZDT test problems.

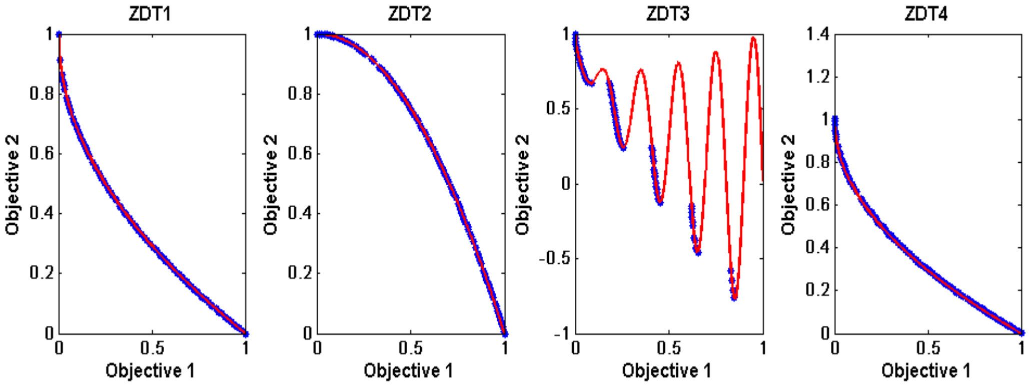


Figure 3.2 Pareto solutions of PAIA on ZDT1~ZDT4.

Via this particular experiment, it was found that PAIA possesses very fast convergence; for ZDT1 and ZDT2, 180 iterations were enough for its convergence, and for ZDT4 500 iterations were enough. Beyond these points, results could not be notably improved. For all the four test problems, both PAIA and VIS obtained good performances (except ZDT4 in VIS) in terms of metrics (3.4) and (3.5).

From Table 3.6, one can see that PAIA generally uses fewer evaluations to lead to good results. Although the algorithm used 46899 evaluations to fully converge, it only used 25910 (see Table 3.4) evaluations to obtain similar results as those produced by VIS (see Table 3.5). This is due to two reasons: 1) PAIA only preserves necessary ‘Abs’ during each iteration step so that only the necessary evaluations are carried out; 2) PAIA uses an ‘adaptive clone size’ so that only necessary clone size is assigned to each selected ‘Ab’. It can be seen from Figure 3.1(c) that in most cases the clone size is 1, although VIS uses a fixed clone size of usually 4. This can lead to two main problems: 1) in the early stages of the optimisaiton rum, a fixed clone size may not be large enough to speed up the convergence; 2) in the later stages of the run, a fixed clone size may be too large so that at each iteration step many unnecessary clones are produced.

TABLE 3.5
MEAN AND VARIANCE VALUES OF GD AND Δ FOR PAIA AND VIS ON ZDT SERIES PROBLEMS

Algorithm	ZDT1		ZDT2		ZDT3		ZDT4	
	GD	σ^2	**GD**	σ^2	**GD**	σ^2	**GD**	σ^2
VIS	1.32e-4	1.12e-9	1.10e-4	2.2e-12	1.23e-4	1.9e-11	1.23e-3	1.12e-6
PAIA	1.58e-4	2.31e-9	1.06e-4	5.7e-11	1.58e-4	4.6e-10	4.96e-4	1.53e-8
Algorithm	Δ	σ^2	Δ	σ^2	Δ	σ^2	Δ	σ^2
VIS	0.3142	6.31e-4	0.2123	3.12e-3	0.3451	1.22e-3	0.0834	1.12e-4
PAIA	0.3522	1.10e-3	0.3443	1.50e-3	0.4381	1.50e-3	0.3058	1.00e-3

TABLE 3.6
FINAL POPULATION SIZE AND EVALUATION TIMES OF PAIA AND VIS ON ZDT TEST PROBLEMS

Test suite	Final Population Size		Evaluation Times	
	PAIA(mean)	**VIS**	**PAIA(mean)**	**VIS**
ZDT1	93	100	15844	28523
ZDT2	95	100	15856	29312
ZDT3	94	100	25365	32436
ZDT4	97	100	46899	38956

3.2.4 Drawbacks of PAIA

Despite the encouraging results achieved in ZDT1~ZDT3 problems, PAIA generally uses more evaluations compared to other GA based algorithms, e.g. SPEA2 (refer to Table 3.2), in order to converge to the 'optimal' solutions when applied to problems with many local optima such as those found in ZDT4. Furthermore, when PAIA is applied to the DTLZ test suite (Deb *et al.*, 2005) (refer to Section 3.3.2 for the descriptions of DTLZ test problems), it fails to converge fully to the Pareto fronts of, for instance, DTLZ1 and DTLZ3, both having many local optima. Generally speaking, PAIA has no problem in finding the global trade-offs, provided enough evaluations are carried-out. This is recognised as the problem associated with the mutation operator which is not adequately designed and as a result many evaluations may be wasted on evaluating the local optima. Furthermore, although not a drawback in itself, worthy of noticing is that the distribution of the final population provided by PAIA can be further improved as shown in Table 3.3, the difference in terms of the distribution of the final population between PAIA and SPEA2 is relatively large, especially for problems with discontinued Pareto fronts; hence the space for the improvements.

3.3 An Improved Version of PAIA (PAIA2)

3.3.1 Basic Ideas behind the Improvements

In light of the observations presented in Section 3.2.4, a modified PAIA with the Simulated Binary Crossover (SBX) (Deb and Agrawal, 1994), as the recombination operator and a modified mutation operator, is proposed (Chen and Mahfouf, 2008). Density information is also incorporated to calculate the affinity of each Antibody in order to allow a more uniform distribution of the final population. The basic idea of the modified mutation operator is to let the mutation rate of each antibody decrease when the optimisation process evolves so that a more focused search is introduced in the later iterations. This decreasing rate can be controlled through a predefined parameter. SBX is a real-code GA crossover and is similar to a mutation operator in the way that it allocates two 'children' alongside their 'parents' by a calculated distance (the only difference is that it uses two solutions to calculate the distance to mutate). The reason for choosing this operator is that in the later iterations solutions are normally close to each other in the decision variable space (especially, when the problem has many local optima), in this case, the modified mutation operator is not good enough to produce an adequate mutation rate (it is either too small or too large). Also, SBX uses two solutions to calculate the distance to mutate, in other words, it takes into account the crowding information in the decision variable space. The mutation operator is very good at finding directions and strengths to mutate in the early iterations and SBX is very good at fine-tuning the distance to mutate in the later iterations. By combining them both, one can reach a very fast convergence and a good accuracy. In the implementation of SPEA2, the author used an adaptation of the *k-th* nearest neighbor method to calculate the density at any point, which is an inverse of the distance to the *k-th* nearest neighbor. In PAIA2, the density estimation is also added to each 'Ab' so that the calculated affinity can reflect this kind of information as well.

Specifically, the following equations presented in Section 3.2.1 are, indeed, in need of some modifications:

(1) Eqs. 3.1 and 3.2 in step 3 are modified with added density information in a bid to obtain solutions with a more uniform distribution, i.e.

$$aff\_val_d = \frac{\sum_{i=1}^{n}\left(x_{identified}(i)-x_d(i)\right)}{n} + D(d) \tag{3.6}$$

$$aff\_val_{nd} = \sum_{j=1}^{N}\frac{\left(\sum_{i=1}^{n}\frac{\left(x_{identified}(i)-x_j(i)\right)}{n}\right)}{N} + D(j) \tag{3.7}$$

where, $D(j)$ (the same for $D(d)$) is the density of the jth antibody and can be calculated by Eq. 3.8. σ_j^k is the distance between point j and the kth nearest point of j. k is set to 2.

$$D(j) = \frac{1}{\sigma_j^k+2} \tag{3.8}$$

(2) The mutation rate α calculated by Eq. 3.3 in step 6 is modified into Eq. 3.9 to incorporate the conception of ‘gradual decrease’, where, r is a decreasing rate and is calculated according to Eq. 3.10.

$$\alpha = r \times \frac{\exp(aff\_val)}{\exp(1)} \tag{3.9}$$

$$r = 1 - rand^{\left(\left(1-\frac{G}{Gen}\right)^{b}\right)} \tag{3.10}$$

Where G is the current iteration and Gen is the predefined total number of iterations. b is a control parameter and equals to 1 in this project. It is worth noting that the selected ‘Abs’ are also submitted to *recombination* which is implemented using SBX with the distribution index (refer to Section 3.2.3, Experiment 1) being 20 in this project.

(3) In the *Reselection* step (step 7), not only the mutated, edited and their corresponding parents but also the recombined clones are mixed together and reselected.

3.3.2 Simulation Studies Using ZDT and DTLZ Series Problems

The benchmark functions used in this Section are ZDT1~ZDT4, ZDT6 and DTLZ1~DTLZ7 (Deb *et al.*, 2005) The DTLZ problems are scalable test problems with three or more objectives and are characterised by a concave (DTLZ2~DTLZ4) and a discrete (DTLZ7) Pareto front, a variable density of the decision variable space and the objective space (DTLZ4, DTLZ6) and many local optima (DTLZ1, DTLZ3). The definitions about DTLZ test suites are defined in Table 3.7.

TABLE 3.7
DTLZ TEST PROBLEMS (DEB *ET AL.*, 2005)

DTLZ1: this test problem is a *M*-objective problem with a linear Pareto-optimal front. Minimize $f_1(X)=\frac{1}{2}x_1x_2\cdots x_{M-1}(1+g(X_M))$, Minimize $f_2(X)=\frac{1}{2}x_1x_2\cdots(1-x_{M-1})(1+g(X_M))$, $\vdots$ Minimize $f_{M-1}(X)=\frac{1}{2}x_1(1-x_2)(1+g(X_M))$, Minimize $f_M(X)=\frac{1}{2}(1-x_1)(1+g(X_M))$, subject to $0\le x_i\le 1$, for $i=1,2,\dots,n$, where $g(X_M)=100\left[\lvert X_M\rvert+\sum_{x_i\in X_M}(x_i-0.5)^2-\cos(20\pi(x_i-0.5))\right]$.	**DTLZ2**: this problem is its concave Pareto-optimal area. Minimize $f_1(X)=(1+g(X_M))\cos(x_1\frac{\pi}{2})\cdots\cos(x_{M-2}\frac{\pi}{2})\cos(x_{M-1}\frac{\pi}{2})$, Minimize $f_2(X)=(1+g(X_M))\cos(x_1\frac{\pi}{2})\cdots\cos(x_{M-2}\frac{\pi}{2})\sin(x_{M-1}\frac{\pi}{2})$, Minimize $f_3(X)=(1+g(X_M))\cos(x_1\frac{\pi}{2})\cdots\sin(x_{M-2}\frac{\pi}{2})$, $\vdots$ Minimize $f_{M-1}(X)=(1+g(X_M))\cos(x_1\frac{\pi}{2})\cdots\sin(x_2\frac{\pi}{2})$, Minimize $f_M(X)=(1+g(X_M))\sin(x_1\frac{\pi}{2})$, subject to $0\le x_i\le 1$, for $i=1,2,\dots,n$, where $g(X_M)=\sum_{x_i\in X_M}(x_i-0.5)^2$.
DTLZ3: this problem has 3^k-1 local Pareto-optimal fronts and one global Pareto-optimal front. *k* is the decision variable's dimension. Minimize $f_1(X)=(1+g(X_M))\cos(x_1\frac{\pi}{2})\cdots\cos(x_{M-2}\frac{\pi}{2})\cos(x_{M-1}\frac{\pi}{2})$, Minimize $f_2(X)=(1+g(X_M))\cos(x_1\frac{\pi}{2})\cdots\cos(x_{M-2}\frac{\pi}{2})\sin(x_{M-1}\frac{\pi}{2})$, Minimize $f_3(X)=(1+g(X_M))\cos(x_1\frac{\pi}{2})\cdots\sin(x_{M-2}\frac{\pi}{2})$, $\vdots$ Minimize $f_{M-1}(X)=(1+g(X_M))\cos(x_1\frac{\pi}{2})\cdots\sin(x_2\frac{\pi}{2})$, Minimize $f_M(X)=(1+g(X_M))\sin(x_1\frac{\pi}{2})$, subject to $0\le x_i\le 1$, for $i=1,2,\dots,n$, where $g(X_M)=100\left[\lvert X_M\rvert+\sum_{x_i\in X_M}(x_i-0.5)^2-\cos(20\pi(x_i-0.5))\right]$.	**DTLZ4**: this problem has more dense solutions near f_3-f_1 and f_1-f_2 planes. Minimize $f_1(X)=(1+g(X_M))\cos(x_1^\alpha\frac{\pi}{2})\cdots\cos(x_{M-2}^\alpha\frac{\pi}{2})\cos(x_{M-1}^\alpha\frac{\pi}{2})$, Minimize $f_2(X)=(1+g(X_M))\cos(x_1^\alpha\frac{\pi}{2})\cdots\cos(x_{M-2}^\alpha\frac{\pi}{2})\sin(x_{M-1}^\alpha\frac{\pi}{2})$, Minimize $f_3(X)=(1+g(X_M))\cos(x_1^\alpha\frac{\pi}{2})\cdots\sin(x_{M-2}^\alpha\frac{\pi}{2})$, $\vdots$ Minimize $f_{M-1}(X)=(1+g(X_M))\cos(x_1^\alpha\frac{\pi}{2})\cdots\sin(x_2^\alpha\frac{\pi}{2})$, Minimize $f_M(X)=(1+g(X_M))\sin(x_1^\alpha\frac{\pi}{2})$, subject to $0\le x_i\le 1$, for $i=1,2,\dots,n$, where $g(X_M)=\sum_{x_i\in X_M}(x_i-0.5)^2$.
DTLZ5: this problem has a Pareto front which is a curve. Minimize $f_1(X)=(1+g(X_M))\cos(\theta_1\frac{\pi}{2})\cdots\cos(\theta_{M-2}\frac{\pi}{2})\cos(\theta_{M-1}\frac{\pi}{2})$, Minimize $f_2(X)=(1+g(X_M))\cos(\theta_1\frac{\pi}{2})\cdots\cos(\theta_{M-2}\frac{\pi}{2})\sin(\theta_{M-1}\frac{\pi}{2})$, Minimize $f_3(X)=(1+g(X_M))\cos(\theta_1\frac{\pi}{2})\cdots\sin(\theta_{M-2}\frac{\pi}{2})$, $\vdots$ Minimize $f_{M-1}(X)=(1+g(X_M))\cos(\theta_1\frac{\pi}{2})\cdots\sin(\theta_2\frac{\pi}{2})$, Minimize $f_M(X)=(1+g(X_M))\sin(\theta_1\frac{\pi}{2})$, subject to $0\le x_i\le 1$, for $i=1,2,\dots,n$, where $g(X_M)=\sum_{x_i\in X_M}(x_i-0.5)^2$, *and* $\theta_i={}^{\pi}\!/_{4(1+g(r))}(1+2g(r)x_i)$, *for* $i=2,\dots,(M-1)$.	***DTLZ6**: this problem has variable density in decision variable space and objective space.* Minimize $f_1(X)=(1+g(X_M))\cos(\theta_1\frac{\pi}{2})\cdots\cos(\theta_{M-2}\frac{\pi}{2})\cos(\theta_{M-1}\frac{\pi}{2})$, Minimize $f_2(X)=(1+g(X_M))\cos(\theta_1\frac{\pi}{2})\cdots\cos(\theta_{M-2}\frac{\pi}{2})\sin(\theta_{M-1}\frac{\pi}{2})$, Minimize $f_3(X)=(1+g(X_M))\cos(\theta_1\frac{\pi}{2})\cdots\sin(\theta_{M-2}\frac{\pi}{2})$, $\vdots$ Minimize $f_{M-1}(X)=(1+g(X_M))\cos(\theta_1\frac{\pi}{2})\cdots\sin(\theta_2\frac{\pi}{2})$, Minimize $f_M(X)=(1+g(X_M))\sin(\theta_1\frac{\pi}{2})$, subject to $0\le x_i\le 1$, for $i=1,2,\dots,n$, where $g(X_M)=\sum_{x_i\in X_M}x_i^{0.1}$, *and* $\theta_i={}^{\pi}\!/_{4(1+g(r))}(1+2g(r)x_i)$, *for* $i=2,\dots,(M-1)$.
***DTLZ7**: this problem has 2^{M-1} disconnected Pareto-optimal regions in the search space.* Minimize $f_1(X_1)=x_1$, *Minimize* $f_2(X_2)=x_2,\dots,$ Minimize $f_{M-1}(X_{M-1})=x_{M-1}$, *Minimize* $f_M(X)=(1+g(X_M))h(f_1,f_2,\dots,f_{M-1},g),$, subject to $0\le x_i\le 1$, for $i=1,2,\dots,n$, where $g(X_M)=1+\frac{9}{\lvert X_M\rvert}\sum_{x_i\in X_M}x_i$, $h(f_1,f_2,\dots,f_{M-1},g)=M-\sum\left[\frac{f_i}{1+g}(1+\sin(3\pi f_i))\right]$.	

❖ Experiment 1 (ZDT1~ZDT6)

The parameter settings for different algorithms are kept the same as those which were used in Section 3.2.3. 25000 evaluations are used for each test problem. Figure 3.3 shows the Pareto fronts obtained by PAIA2, where the continuous lines represent the true Pareto fronts and the dots represent the obtained fronts using PAIA2.

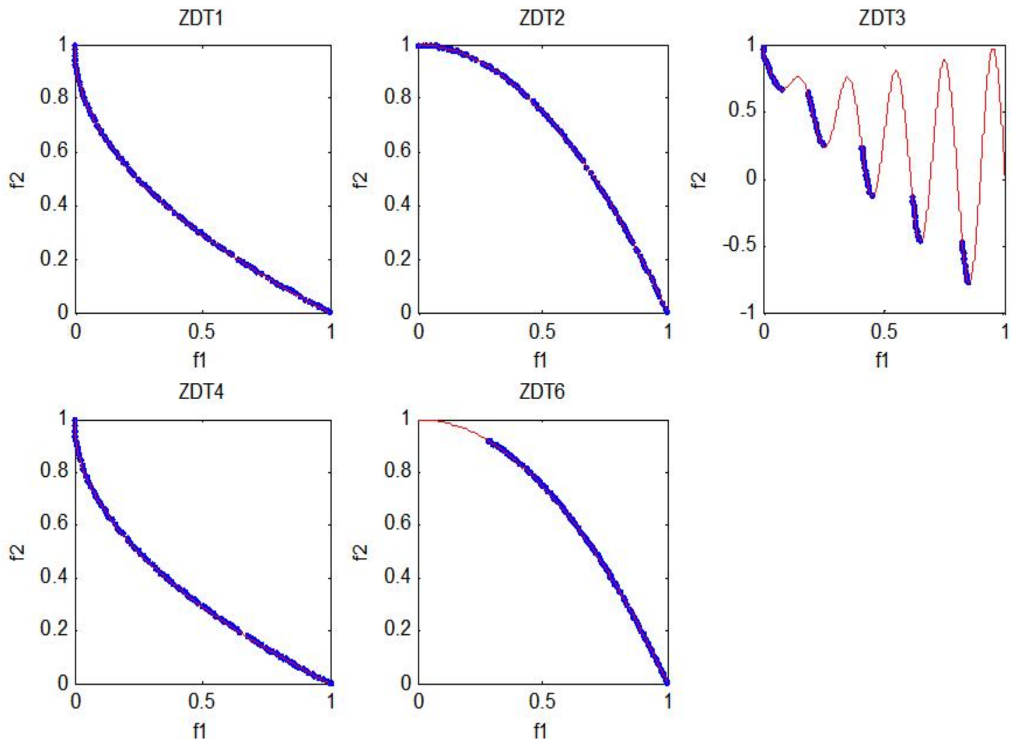


Figure 3.3 Pareto solutions of the modified PAIA2 on ZDT1~ZDT4 and ZDT6.

From the above figure, one can see that the PAIA2 approaches the true Pareto fronts with very good diversity and accuracy. The results shown in Table 3.8 also indicate that PAIA2 reached a better performance than any of other four algorithms in terms of convergence. Results of SPEA2 are comparatively as good as those results obtained using PAIA2. Generally, PAIA2 produces slightly better convergence properties, while SPEA2 produces a slightly better distribution as seen from Tables 3.8~3.9.

As already stated in the first part of this section, the original PAIA needs more evaluations to finally converge for problems consisting of many local minima. This is confirmed by the experiment and can also be seen in Table 3.8. For ZDT4, while PAIA did not quite converge to the true Pareto front using 25000 evaluations, the enhanced PAIA2 had no problem in finding the true Pareto front within 25000 evaluations with the aid of the new mutation operator and SBX. Due to the inclusion of the density information in the calculation of the affinity, PAIA2 slightly improved the performance as far as diversity is concerned.

TABLE 3.8

MEAN AND VARIANCE VALUES OF THE CONVERGENCE MEASURE GD FOR ZDT SERIES PROBLEMS

Test problems/Algorithms		**NSGAII**	**SPEA2**	**VIS**	**PAIA**	**PAIA2**
ZDT1	GD	8.94e-4	2.64e-004	1.81e-3	**1.43e-4**	2.45e-4
	σ^2	0	4.68e-010	1.97e-7	1.56e-9	4.44e-10
ZDT2	GD	8.24e-4	1.05e-004	1.21e-3	1.04e-4	**9.34e-005**
	σ^2	0	1.21e-011	1.04e-6	2.2e-11	3.69e-011
ZDT3	GD	4.34e-2	1.69e-004	1.58e-3	1.58e-4	**1.55e-004**
	σ^2	4.20e-5	1.74e-010	2.26e-7	4.60e-10	1.89e-010
ZDT4	GD	3.228	4.68e-004	0.1323	1.20e-3	**2.43e-004**
	σ^2	7.3076	1.36e-008	4.20e-2	1.88e-7	7.86e-010
ZDT6	GD	7.8067	1.81e-004	-	1.02e-4	**9.41e-005**
	σ^2	1.67e-3	6.65e-011	-	6.04e-12	1.87e-011

TABLE 3.9

MEAN AND VARIANCE VALUES OF THE DIVERSITY MEASURE Δ FOR ZDT SERIES PROBLEMS

Test problems/Algorithms		**NSGAII**	**SPEA2**	**VIS**	**PAIA**	**PAIA2**
ZDT1	Δ	0.4633	**0.1575**	0.5420	0.3368	0.3289
	σ^2	4.16e-2	1.44e-004	8.25e-3	1.10e-3	7.05e-004
ZDT2	Δ	0.4351	**0.1523**	0.6625	0.3023	0.3345
	σ^2	2.46e-2	1.31e-004	2.58e-2	7.07e-4	3.55e-004
ZDT3	Δ	0.5756	**0.1638**	0.6274	0.4381	0.3292
	σ^2	5.08e-3	2.90e-2	1.60e-2	1.50e-3	2.61e-004
ZDT4	Δ	0.4795	**0.1555**	0.1011	0.3316	0.3310
	σ^2	9.84e-3	4.34e-004	1.37e-3	1.20e-3	4.18e-004
ZDT6	Δ	0.6444	**0.3248**	-	0.4932	0.3210
	σ^2	3.50e-2	1.29e-004	-	3.56e-4	2.58e-004

Figure 3.4 shows the results from the original PAIA and PAIA2 under the same number of evaluations when they were applied to ZDT4. The original PAIA failed to fully converge to the Pareto front in this case, which justifies the new proposed mutation operator and the incorporation of SBX.

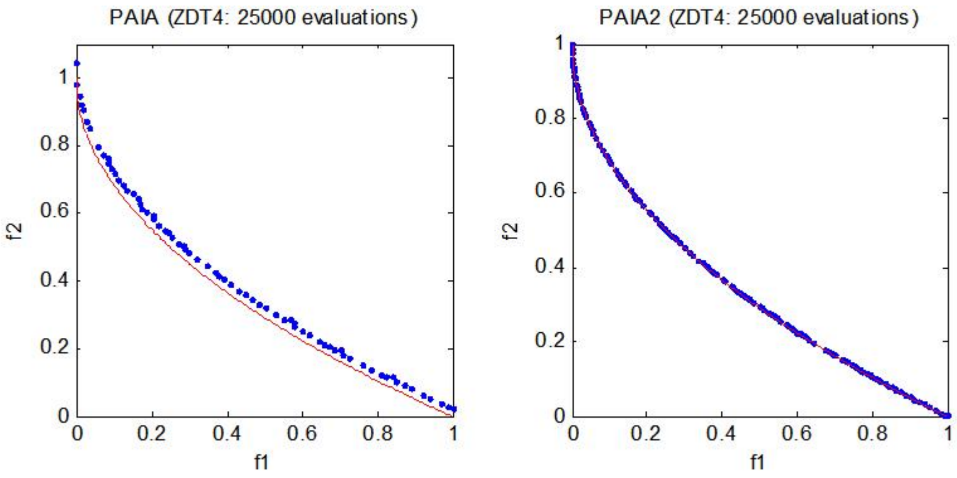


Figure 3.4 Pareto solutions of the original PAIA (left) and PAIA2 (right) on ZDT4.

To examine how efficient PAIA2 is compared to PAIA, VIS and SPEA2, all four algorithms are run as many evaluations as necessary until adequate convergence and diversity (this means that both metrics cannot be significantly improved by only increasing the number of evaluations) are obtained. Table 3.10 summarises the results after 10 independent runs.

TABLE 3.10
EVALUATION TIMES OF PAIA2, PAIA, VIS AND SPEA2 WHEN THEY ARE FULLY CONVERGED

Test suite	**Evaluation Times**			
	PAIA2 (GD/Δ)	**PAIA(GD/Δ)**	**VIS (GD/Δ)**	**SPEA2(GD/Δ)**
ZDT1	**7500** (**2.48e-4**/0.2990)	15844 (1.58e-4/0.3522)	28523 (1.32e-4/0.3142)	8000 (2.66e-4/0.1814)
ZDT2	**7000** (**9.12e-5**/0.3567)	15856 (1.06e-4/0.3443)	29312 (1.10e-4/0.2123)	11000 (9.50e-5/0.1589)
ZDT3	**7500** (1.81e-4/0.4201)	25365 (1.58e-4/0.4381)	32436 (1.23e-4/0.3451)	9000 (**1.69e-4**/0.1489)
ZDT4	**20000** (**2.90e-4**/0.3140)	46899 (4.96e-4/0.3058)	46899(1.23e-3/0.0834)	**20000** (5.56e-4/ 0.1879)
ZDT6	**3900** (**1.40e-4**/0.4569)	8766 (1.48e-4/0.5929)	-	18000 (2.65e-4/0.3172)

It can be seen that for all the five test problems, PAIA2 generally uses a fewer evaluations than VIS does. Furthermore, results of PAIA2 are comparatively as good as those obtained by SPEA2. It is worth noting that PAIA2 only used 3900 evaluations for ZDT6 compared to 18000 evaluations used by SPEA2. The justification behind such a big difference lies in the fact that PAIA2 uses information from both the objective and the decision variable spaces to calculate the affinity. For the problem having a variable density in both the decision variable space and the objective space, the aforementioned scheme seems very effective. A similar observation is encountered in Experiment 2 for DTLZ6. The big difference in the number of evaluations needed by PAIA2 and VIS indicates that there must be some fundamental differences in the design of the algorithms, which can mainly be attributed to the adaptive clone and population size adopted by PAIA2.

❖ Experiment 2 (DTLZ1~DTLZ7)

In this experiment, PAIA2 is compared to SPEA2. For the sake of fairness in comparisons, the experiment configuration refers to those in Deb *et al.* (2001). The maximum function evaluations and the number of decision variables are shown in Table 3.11. All the parameter settings for SPEA2 are kept unchanged except for the maximum generations. For PAIA2, all the parameters are kept the same as the last experiment expect for the network suppression threshold which will be shown along with the corresponding plots. Figures 3.5~3.11 show the results of PAIA2 and SPEA2 from various angles of view.

TABLE 3.11
THE MAXIMUM FUNCTION EVALUATIONS AND THE NUMBER OF DECISION VARIABLES

Test suite	**The maximum function evaluations**	**The number of decision vairables**	**The number of objectives**
DTLZ1	30000	7	3
DTLZ2	30000	12	3
DTLZ3	50000	12	3
DTLZ4	20000	12	3
DTLZ5	20000	12	3
DTLZ6	50000	12	3
DTLZ7	20000	22	3

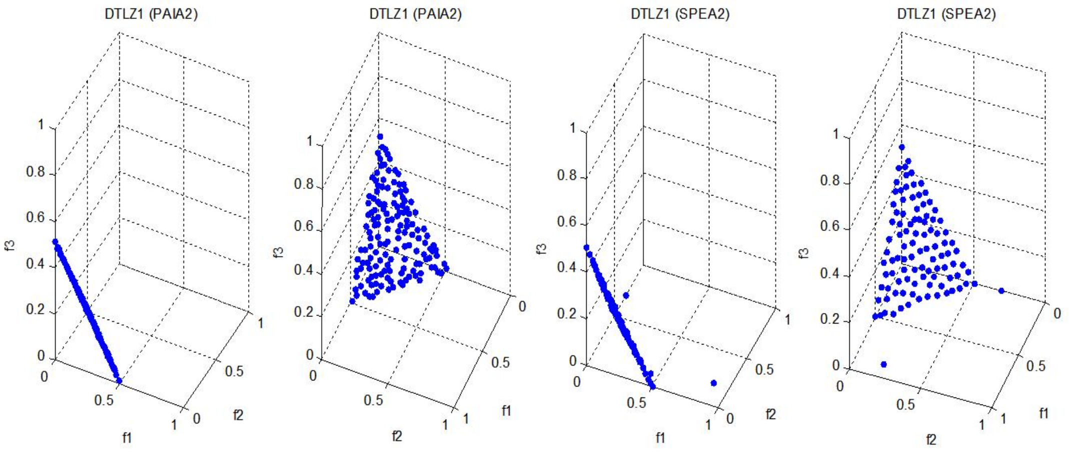


Figure 3.5 The results of PAIA2 ($\sigma = 0.03$) and SPEA2 on DTLZ1.

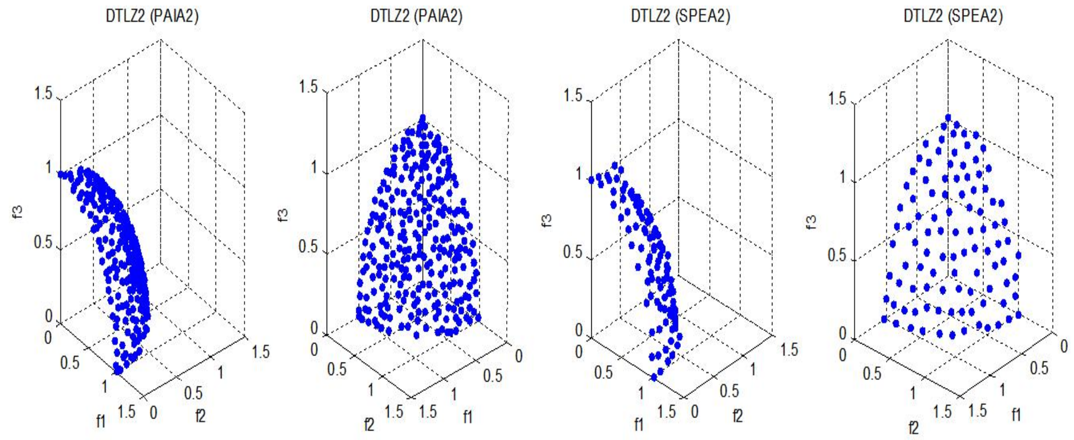


Figure 3.6 The results of PAIA2 ($\sigma = 0.03$) and SPEA2 on DTLZ2.

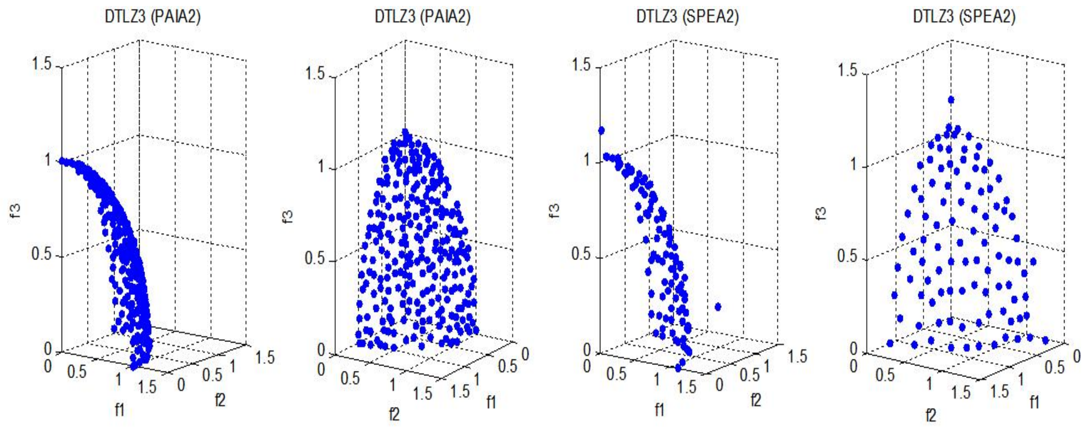


Figure 3.7 The results of PAIA2 ($\sigma = 0.03$) and SPEA2 on DTLZ3.

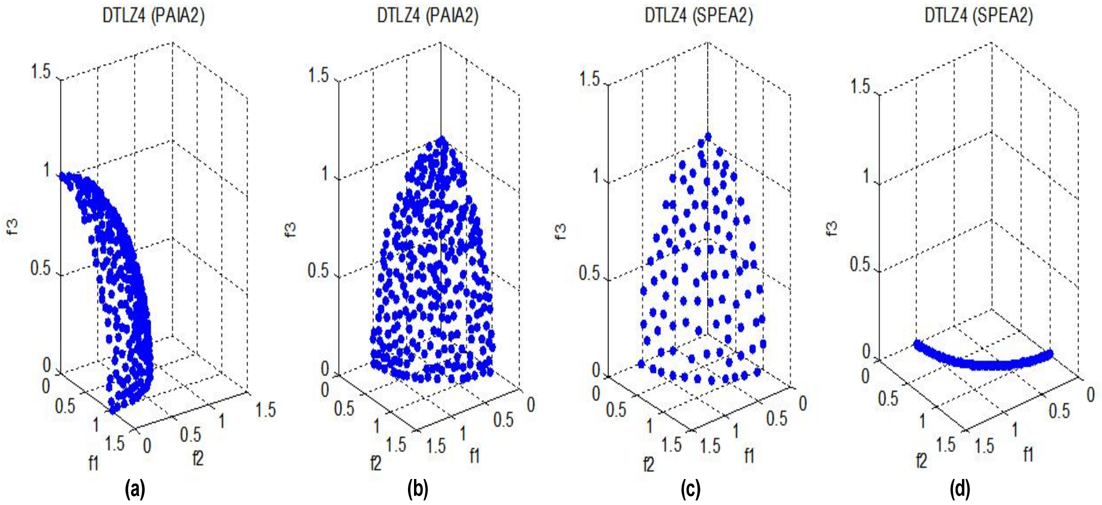


Figure 3.8 The results of PAIA2 ($\sigma = 0.03$) and SPEA2 on DTLZ4.

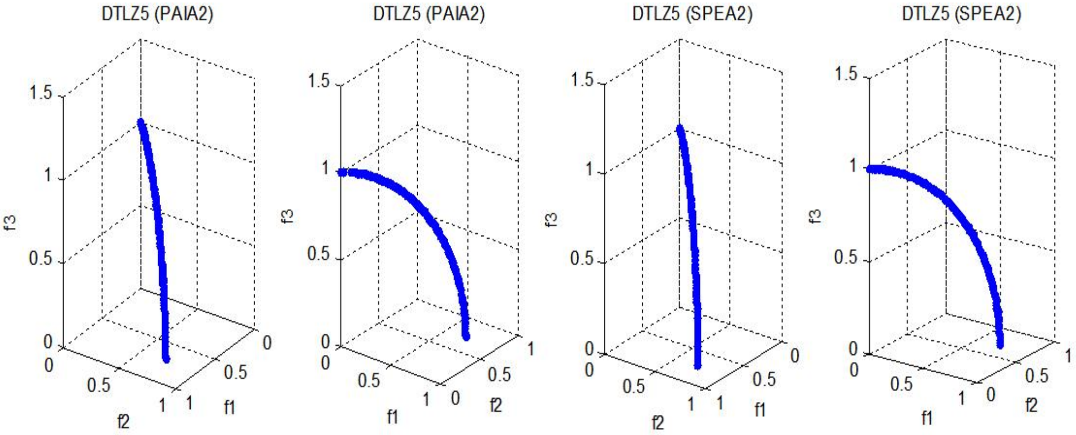


Figure 3.9 The results of PAIA2 ($\sigma = 0.004$) and SPEA2 on DTLZ5.

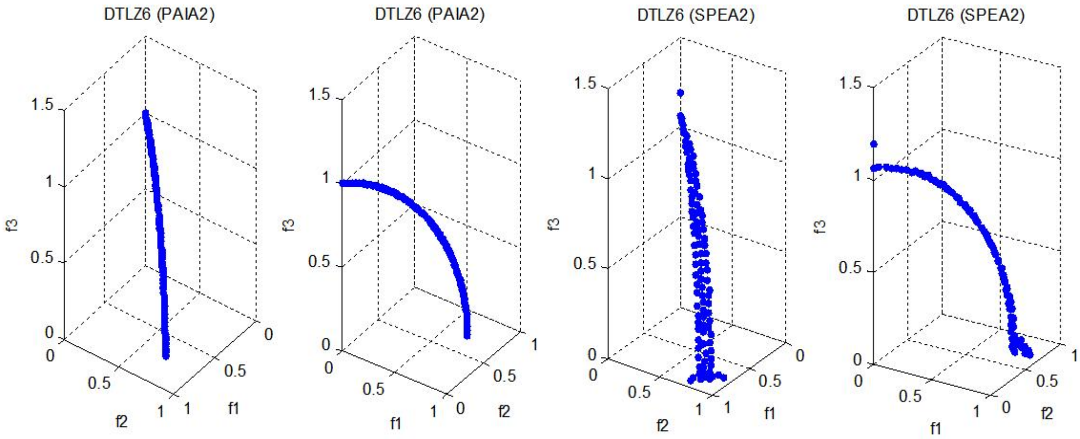


Figure 3.10 The results of PAIA2 ($\sigma = 0.004$) and SPEA2 on DTLZ6.

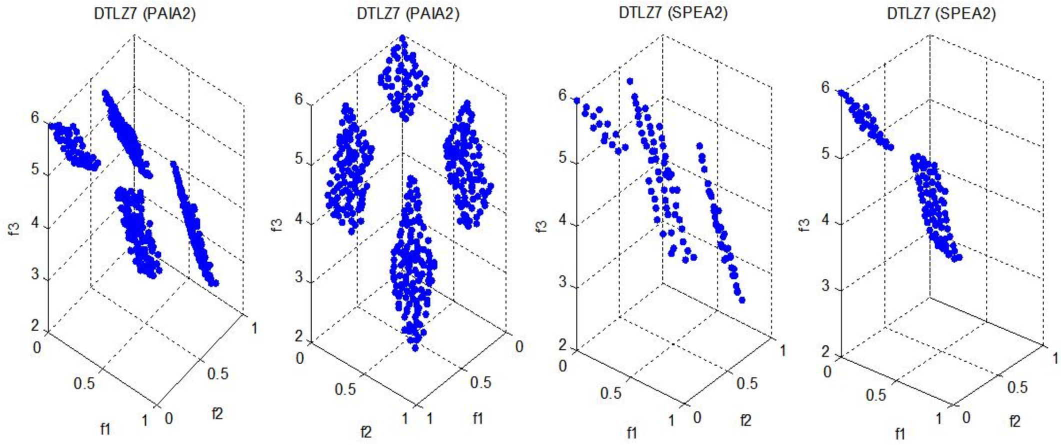


Figure 3.11 The results of PAIA2 ($\sigma = 0.01$) and SPEA2 on DTLZ7.

For all seven problems, PAIA2 consistently produced better results than SPEA2. For DTLZ1 and DTLZ3, SPEA2 was not able to converge to the true Pareto front although the overall results were very close to the optimal solutions. For DTLZ4, since the density of the decision variable is different, SPEA2 has a tendency to converge to the verge of the whole front. As one can see from Figure 3.8, SPEA2 led to two outcomes:

1) Converged to any one verge out of three verges (Figure 3.8 (d));
2) Converged to the whole Pareto front (Figure 3.8 (c)).

Which outcome it finally reaches highly depended on the initial population. Also, PAIA2 had no problem in finding the spread solutions along the whole Pareto front. For DTLZ6, SPEA2 encountered two problems:

1) It cannot converge to the Pareto front;
2) The Pareto front is not truly a curve due to the variable density of the solutions in the objective space.

For the same problem, PAIA2 produced a very good approximation of the Pareto front. It is worth recalling that, for DTLZ6, PAIA2 can use much less evaluations than SPEA2 does (less than 5000 evaluations compared to 50000 evaluations in SPEA2). Figure 3.12 shows the results of PAIA2 using 5000 evaluations. As already mentioned in Experiment 1, and in contrast to SPEA2, PAIA2 utilises information from both the decision variable space and the objective space. Hence, it is very good at dealing with problems having variable densities both in the objective and decision variable spaces (e.g. DTLZ4 and DTLZ6)

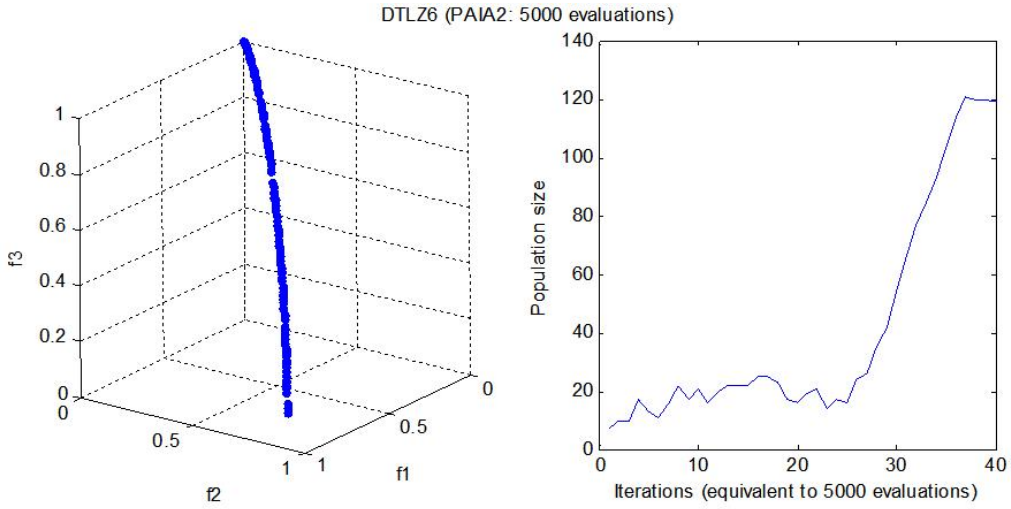


Figure 3.12 The results of PAIA2 on DTLZ6 using 5000 evaluations.

3.3.3 Discussions

From the description and the experiments of PAIA2, one can conclude the following two points as the most important parts when implementing an optimisation algorithm for any type of the optimisation problems:

(1) A good balance between exploitation and exploration of the search space.
(2) The diversity of the population.

However, special attention should be given when one tries to embody these two points for different types of optimisation problems. In the field of real-valued optimisation, the first point is normally implemented by generating the offspring around their parents. The number of parents could be one, two, or any number depending on the specific application. The distance of the new solutions to their corresponding parents depends on how good their parents are and the stage of the search process. Normally, if their parents are very good in terms of their fitness or in the late stage of the search the distance is a small value so that a focused search can be carried out, and vice versa. In this way, a good balance between exploitation and exploration is achieved. In this sense, as far as real-valued optimisation is concerned, the distinction line between mutation and crossover diminishes since they all tend to allocate their children to the places according to the calculated distances; the naive crossover operator (Goldberg, 1989) is not applicable in this case anymore. There are many ways of maintaining the diversity of the population, e.g. random mutation, large population size, or the insertion of new random individuals. PAIA2 is the embodiment of the above features via the following implementations:

1) Hypermutation maintains a good balance between exploitation and exploration of the search space by providing a small mutation rate to the good Abs and vice versa.

2) The decreasing rate *r* (see Eq. 3.10) allows a finer search to be carried-out in the late stage of the optimisation process. Figure 3.13 depicts the change of *r* against the iteration step when 200 iterations are used.

3) The recombination operator-SBX allows a more focused search in the late stage of the optimisation. In this case solutions are normally close to each other, and thus the calculated distance is small.

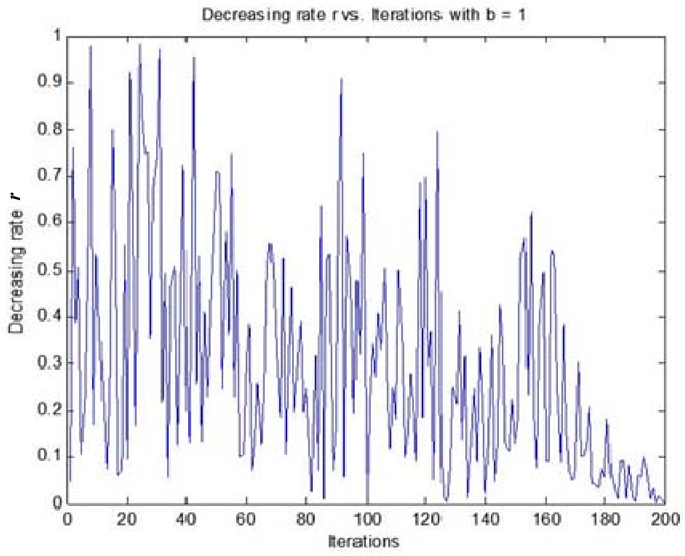


Figure 3.13 Decreasing rate r vs. iterations.

4) Receptor editing explores more search space by employing mutations in more positions with large mutation rates.

5) Adaptive clone size ensures the diversity of the 'Abs' population.

3.4 Effects of the User Specified Parameters

3.4.1 Effect of the Initial Population Size

In PAIA2, the population size is not fixed. It is regulated by the network suppression threshold σ so that any too-close 'Abs' are suppressed. It will be finally stabilized, which is a sign for the convergence of the algorithm. Due to the nature of the adaptive population as one can see from Figure 3.1 in Section 3.2.3, irrespective of the initial size used the population can be adaptively adjusted to a reasonable size according to the need of the problem. Figure 3.14 and Table 3.12 take ZDT2 and ZDT3 as examples, without any loss of generality, to show that even with 1 as the initial size the algorithm can still find the Pareto front.

Although the initial size is not crucial to the success of PAIA2, Table 3.12 clearly indicates that in the case of 1 as the initial size more evaluations are needed compared to the one with 7 as its initial size. Hence, a carefully chosen initial size (7 in this case) does reduce the computation load of PAIA.

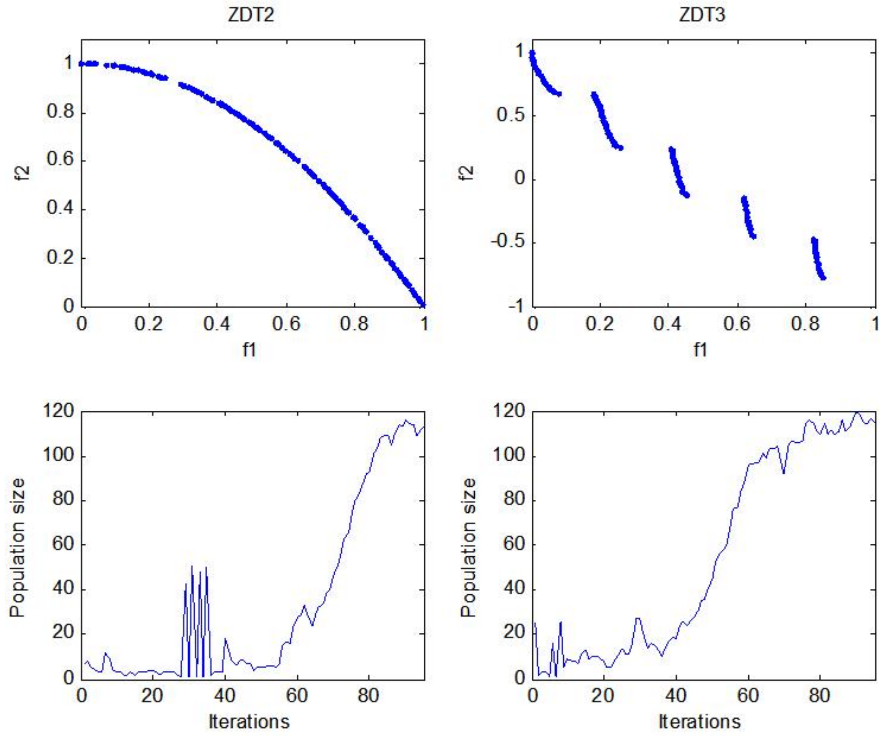


Figure 3.14 PAIA2 with 1 as the initial population size.

TABLE 3.12

THE MAXIMUM FUNCTION EVALUATIONS AND THE NUMBER OF DECISION VARIABLES

Test suite	**Evaluation Times**	
	PAIA2 (GD/Δ) with 7 as the initial size	**PAIA2 (GD/Δ)** with 1 as the initial size
ZDT2	7000 (9.12e-5/0.3567)	12000 (9.28e-5/0.3340)
ZDT3	7500 (1.81e-4/0.4201)	14000 (1.47e-4/0.2860)

3.4.2 Effect of the Clonal Selection Threshold

Theoretically, the Clonal Selection Threshold δ can vary between 0 and 1. Figure 3.15 uses ZDT4 to show how δ affects the convergence performance of PAIA2. The results reported here are the average values of 10 independent runs with a varying δ at 0, 0.1,…,1 (marked with squares in Figure 3.15).

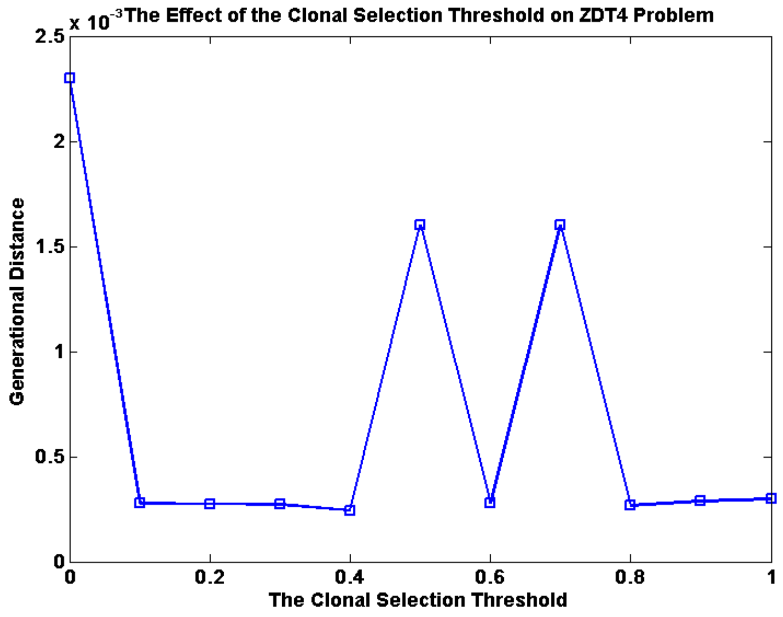


Figure 3.15 The effect of the Clonal Selection Threshold δ on ZDT4 problem.

As one can see from Figure 3.15, δ has only a small impact on the algorithm's convergence performance. A big value of δ means that more Abs can be selected to undergo the clone process and the affinity maturation process. While, a small value of δ imposes more selection pressure upon the population. Hence, when δ approaches 0 only the best solutions among the population have the chance to be selected, which suppresses the diversity of the population and leads to a relatively inferior performance. When δ is greater than 0.4 the convergence index of PAIA2 is oscillating, which reflects the fact that even worse solutions have the chance to be selected. Hence, values between 0.1 and 0.4 for δ represent all good choices, which on the one hand will lead to an appropriate diversity and on the other hand will discourage bad solutions being included in the next iteration.

3.4.3 Effect of the Network Suppression Threshold

One promising property of PAIA2 is that it generally finds more solutions (which can be tuned with the Network threshold σ) with similar or less evaluations than those required by other evolutionary algorithms, such as SPEA2. By tuning the Network threshold, one can obtain more options in a single run without increasing the number of evaluations dramatically. Figure 3.16 shows that even without increasing the number of evaluations, the obtained non-dominated solutions increased from 109 to 357!

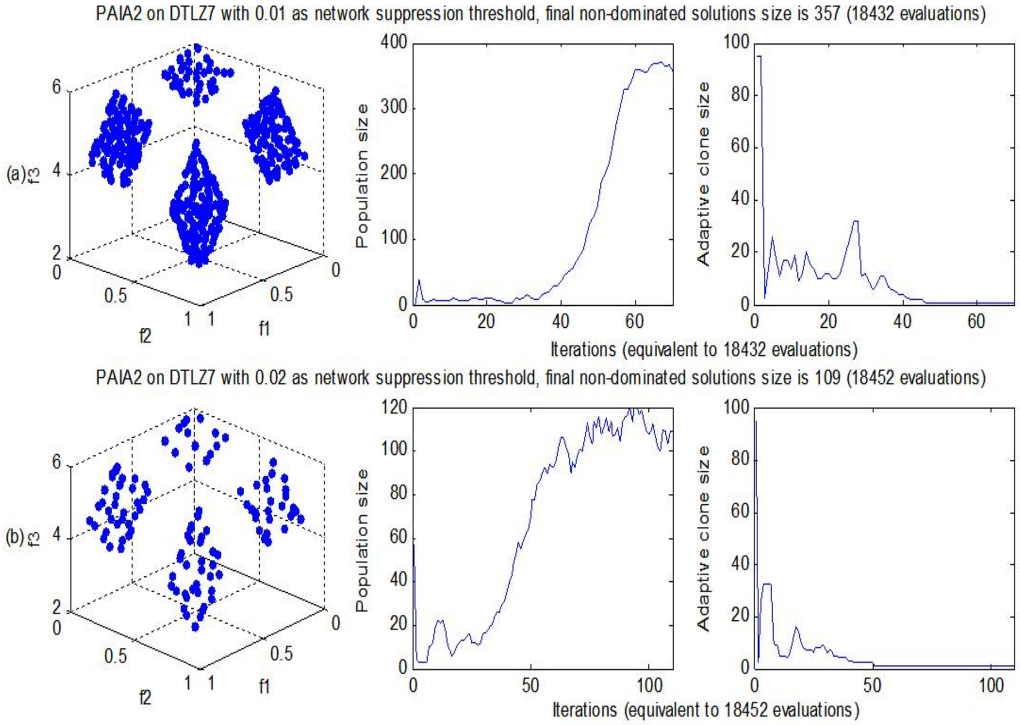


Figure 3.16 (a) $\sigma = 0.01$, 18432 evaluations; (b) $\sigma = 0.02$, 18452 evaluations.

Although this parameter is generally not an important factor which decides on the convergence and the distribution indices of the PAIA2, it has nevertheless to be set to an appropriate value. An insufficient design of σ will lead to a population which is either sparse or overcrowding.

3.4.4 Effect of the Maximum Clone Size

In order to study the effect of the maximum clone size N_{cmax} on the performance of PAIA2, ZDT4 is taken as an example without any loss of generality. All other parameters are kept the same as those used in Section 3.2.3, except N_{cmax} which is varied here at the values of 1, 10, 20,...,100, 200,...,500 (marked as squares in Figure 3.17). Figure 3.17 (a) illustrates how the convergence index GD is affected by the varying N_{cmax}. Figure 3.17 (b) demonstrates the least number of required evaluation times for PAIA2 with varying N_{cmax} to fully converge.

Although the convergence index GD is not greatly affected by N_{cmax}, the least number of required evaluation times for PAIA2 to fully converge increases a lot when N_{cmax} equals 0 or exceeds 200. Hence, a value between 10 and 100 represents a good choice for N_{cmax}.

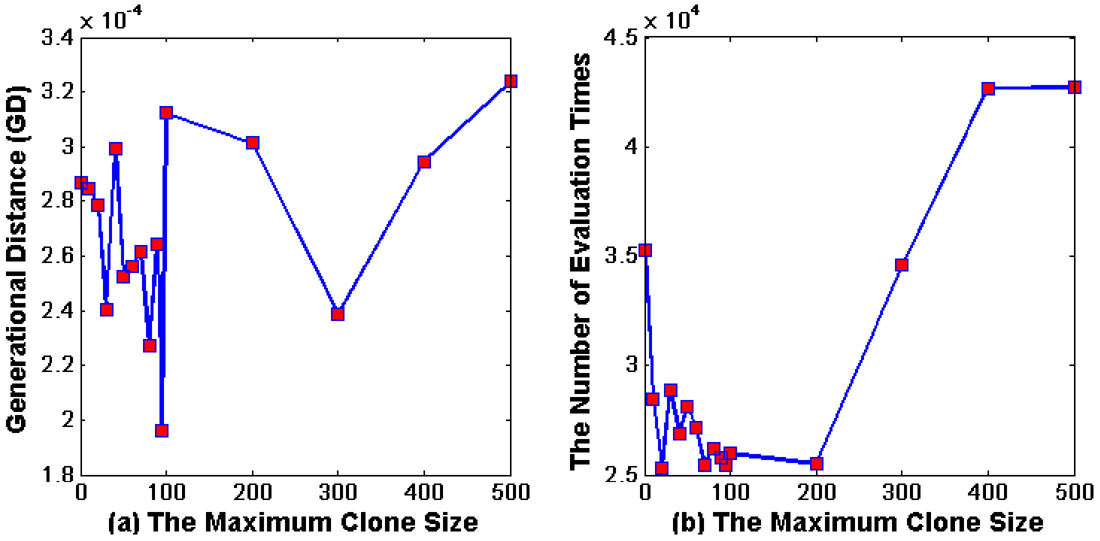


Figure 3.17 The effect of the maximum clone size N_{cmax} on ZDT4 problem: (a) how the convergence index GD is affected by the varying N_{cmax}; (b) the least number of required evaluation times for PAIA2 with varying N_{cmax} to fully converge.

3.4.5 Adaptive Clone Size and Adaptive Population Size

As mentioned in Experiment 2 in Section 3.2.3, the big difference in the needed number of evaluations of PAIA2 and other immune based algorithms, such as VIS, indicates that there must be some fundamental differences in the design of the algorithms. In Section 3.2.3, such differences are summarised as the adaptive clone size and adaptive population size.

Figure 3.18 only takes results from ZDT4 as an example to provide a graphical explanation about the aforementioned differences. From Figure 3.18, it can be seen that the clone size is dictated by the population size and their corresponding affinities. If the population size is small, each selected 'Ab' can be assigned a large clone size so that the size of the activated clones is large enough to explore the objective space.

Most previous research studies, such as those associated with VIS, fixed the clone size (4 in VIS), which can generally lead to two main problems:

1) In the early stage, a fixed clone size may not be large enough to speed up the convergence;
2) In the later stage, a fixed clone size may be too large so that at each iteration step too many unnecessary clones are produced.

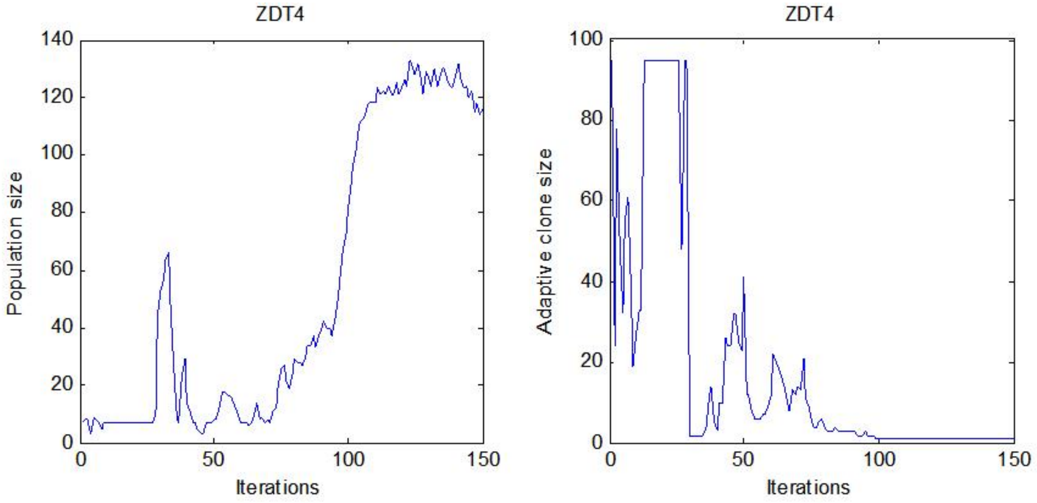


Figure 3.18 Adaptive population size and adaptive clone size (the assigned maximum clone size among all Abs) vs. iteration.

3.5 A Multi-Stage Optimisation Procedure (M-PAIA2)

In this section, a new algorithm for optimisation procedure has been proposed, which aims at reducing the computational cost when dealing with problems having many local optima. The associated procedure is inspired by the vaccination process and the secondary response of the immune systems (refer to Section 2.2.4).

3.5.1 Connection between 'Multi-Stage' and Immunology

It is well known that if the vaccine (which is very similar to real antigens in terms of their structures) is available and first applied to the immune systems, the immune systems can remember it and can respond quickly in the successive encounter with similar antigens. Such a response is called the 'secondary response' in the immune community.

The same mechanism can be emulated in the MOP algorithm by first obtaining a solution which can be viewed as the vaccine, and then invoking the immune algorithm with the vaccine as one of the initial population to find the remaining solutions. Here, the problem relates to how to acquire the vaccine in the first place without any knowledge about the problem to be solved. The process of obtaining the vaccine should be computationally inexpensive otherwise the exercise concerned with adding this additional mechanism would be fruitless.

Figure 3.1 (b) is a reflection of the immune response from 'Ab' when stimulated by 'Ag'. It can be seen that the population ('Ab' concentration) keeps increasing with the presence of antigenic stimuli until a stable concentration level is achieved. If without local optima, a problem (i.e. ZDT1~ZDT3) can be regarded as an unvaccinated immune system whose 'Ab' concentration bears characteristics illustrated in the first three graphs in Figure 3.1 (b), then such characteristic is seen as primary immune response. Also, when a problem has many local optima and these optima share some common features (ZDT4), it corresponds to an immune system with continuous vaccinations. As in the last graph of Figure 3.1 (b), the 'Ab' concentration initially reacts as a primary response; however, in the following vaccinations its peak values match each optima and this is recognised here as a 'secondary response'. The process shown in Figure 3.1 (b) also indicates that the most efficient way of dealing with MOP is to quickly obtain any of the solutions residing on the Pareto front. Based on this solution, the extension to the other parts of the Pareto front can be easily obtained.

In the light of the above discussions, a multi-stage optimisation procedure is proposed, which divides the whole search procedure into two separate stages with the first being the vaccination process. In the first stage, a single objective optimisation method is used to quickly find a solution on the Pareto front, and in the second stage, PAIA2 is used as a post-processing algorithm to approximate the rest trade-offs along the Pareto front. In the following space, the reason as to why the first stage can reduce the computational cost of the whole optimisation process is explained next.

3.5.2 Why 'Multi-Stage'?

In the MOP context, the direction information is not fully used. As far as the dominance-based method (refer to Section 2.4.2) is concerned, the partial order of the candidates according to their dominance is given; in such a case candidates can only progress in a general direction (in the sense of dominance concept rather than a fixed direction leading to a more optimised front). In the weighted aggregation method (refer to Section 2.4.2), the weight combination is adaptively changed, hence, candidates always change their directions to minimise the current weight combination. Both the aforementioned cases result in an ineffective search due to many searches being wasted to find more dominant solutions in the current non-dominated front rather than progress to a more optimised front. Also, most single objective optimisation algorithms use directional information in a more ordered fashion since there is only one objective to deal with. The effect of this is that single objective optimisation algorithms are more efficient in terms of approaching the solution on the true Pareto front. Figure 3.19 is a graphical explanation of the above arguments via a 2-dimensional problem with one or two objectives plotted in the decision variable spaces without any loss of generality.

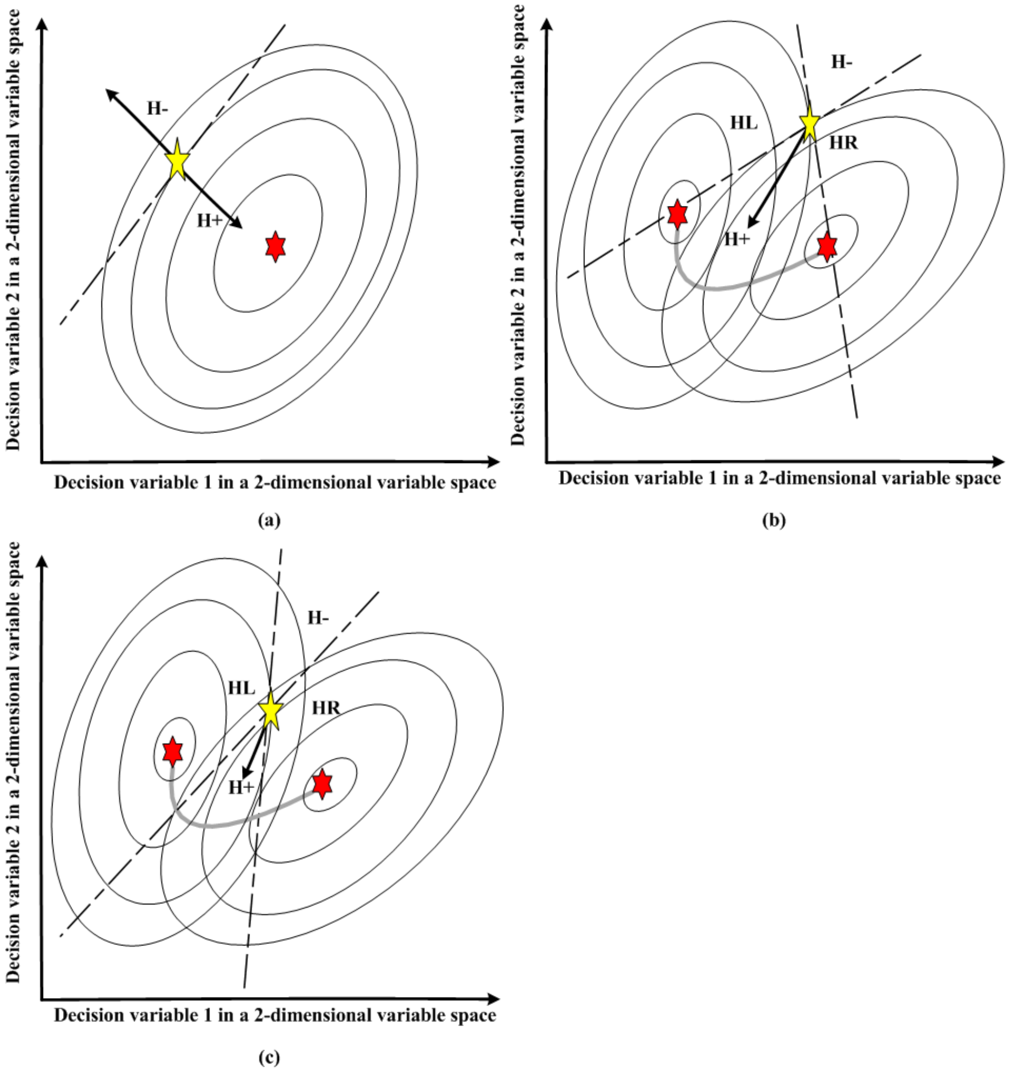

Decision variable 2 in a 2-dimensional variable space
H-
H+
Decision variable 1 in a 2-dimensional variable space
(a)
Decision variable 2 in a 2-dimensional variable space
H-
HL
HR
H+
Decision variable 1 in a 2-dimensional variable space
(b)
Decision variable 2 in a 2-dimensional variable space
H-
HL
HR
H+
Decision variable 1 in a 2-dimensional variable space
(c)

Figure 3.19 (a) Single objective optimisation; (b) two-objective optimisation with candidate solutions far from the true Pareto front; (c) two-objective optimisation with solutions close to the Pareto front.

In Figure 3.19, pentacles (yellow ones) are the starting points from which new solutions will be produced. The 'stars' (red ones) in the middle of the ellipses are the global optimum of each objective. In stochastic search methods, pentacles can move in any direction with equal opportunity. In a single objective optimisation case, the pentacle has 50% chance to choose the right direction, represented by H+ half plane in Figure 3.19 (a), to move. In the two-objective case, this is more involved as Figure 3.19 (b) and (c) indicate. The lines connecting two stars are the Pareto solutions that one wishes to approach. Again, H+ plane represents the right direction to go since if the newly generated solution falls into this area it will simultaneously optimise two objectives. However, as one can see from the figure, the probability of choosing the right direction becomes smaller in this case as compared to the single-objective case. More often than not, there is a greater probability of choosing a direction which falls into HL and HR planes so that only trade-offs can be found rather than better solutions. The situation becomes more severe when the candidates are close to the Pareto solutions. In such a case many searches are wasted by moving from one place to another place on the same trade-off front. This is the reason why for most evolutionary algorithms it becomes inefficient to progress any further to the true Pareto front in the later iterations.

From the above discussions, it is argued that the most efficient way to deal with MOP problem is to divide the search process into two separate stages. In the first stage, a single objective optimisation algorithm is used to find any solution on the Pareto front. The solution found in the first stage serves as the vaccine in the second stage to quickly find the rest solutions on the Pareto front. In doing so, one maximizes the possibility of choosing the appropriate direction in both stages.

3.5.3 Comparisons between PAIA2 and M-PAIA2

In this section, DTLZ1 is taken as an example to show the efficiency of M-PAIA2 as compared to PAIA2. Figure 3.20 shows the graphical results of DTLZ1. In the first stage, PAIA2 is used as a single objective optimiser to find an optimum corresponding to a fixed weight combination of the objectives (Figure 3.20 (a)). The solution found in the first step is then fed into PAIA2 as the initial population to find the rest solutions. From Figure 3.20, it can be seen that in the first stage 6948 evaluations are executed, and in the second stage 9719 evaluations are needed for the rest solutions, which leads to 16667 evaluations in total to cover the whole Pareto front compared to 30000 evaluations in Experiment 2 of Section 3.3.2. With the solution found in the first stage, PAIA2 is able to quickly find the remaining solutions on the Pareto front as shown in Figure 3.20 (d), which corresponds to the secondary response in the immune systems. Figure 3.20 (b) shows the variation curve of the population size by only using the PAIA2. The curve fluctuates with each peak corresponding to a local Pareto front. More evaluations are needed to finally stabilise the population size in this case.

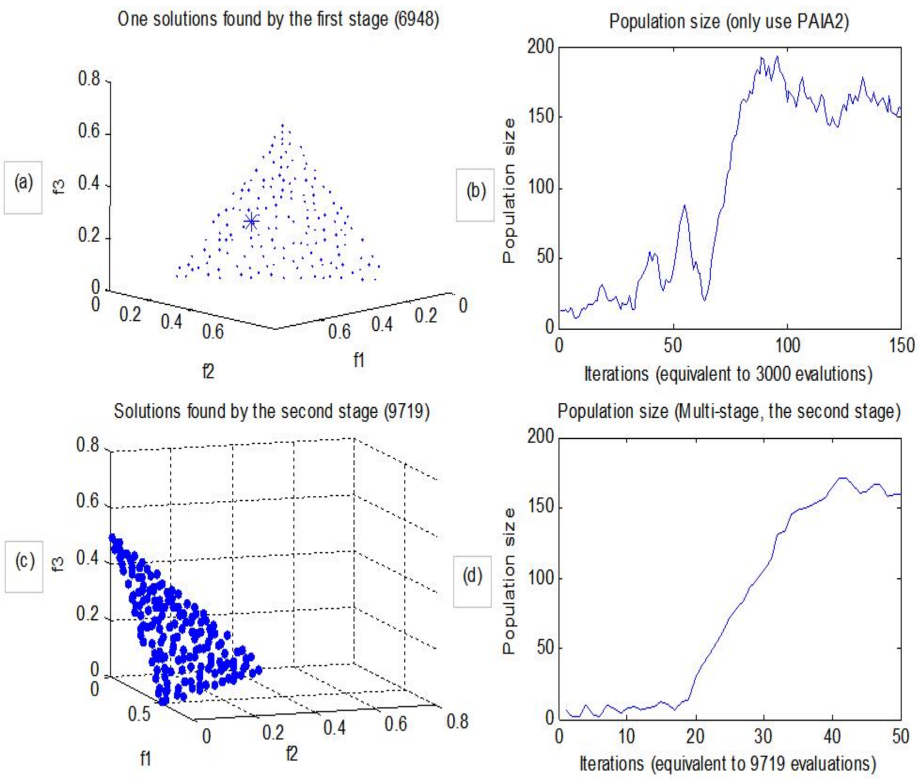


Figure 3.20 Results from multi-stage optimisation procedure on DTLZ1: (a) solution found by the first stage; (b) adaptive population size vs. iterations using PAIA2; (c) non-dominated solutions found by the second stage; (d) adaptive population size vs. iterations using multi-stage optimisation procedure (the second stage).

3.6 General Framework of AIS-based Multi-Objective Optimisation (MO) Algorithms

3.6.1 The Framework

Although PAIA2 is a specific MO algorithm, the main structure of the algorithm can be extracted as a ‘generic’ AIS framework for MOP solving, as shown in Figure 3.21.

Two types of activation are emulated, namely ‘Ag-Ab’ activation and ‘Ab-Ab’ activation, so that one obtains information from both the objective space (‘Ag-Ab’ affinity) and the decision variables space (‘Abs’ affinity) to select ‘Abs’. The Clonal Selection and Clone prefer good ‘Abs’ by providing them with more chances to be cloned so that they always dominate the whole population. Furthermore, the Clone itself contributes singnificantly to the diversity of the population. Affinity Maturation includes hypermutation, receptor editing and recombination, the former two of which increase the diversity of the population so that more objective landscape can be explored, and the last one of which efficiently uses the information contained in the solutions so that fine search can be executed in the late stage of the optimisation. Reselection ensures that good mutants are inserted into the memory set and bad 'Abs' *apoptosis*. Network Suppression regulates the population so that it is adaptive to the search process. Newcomers are used to further increase the diversity of 'Abs'. It is argued here that each part of the framework can be implemented by various means; while the basic structure remains unchanged. The framework is more consistent with the previously discussed immune mechanisms, and thus it can serve as a guide to design AIS-based optimisation algorithms.

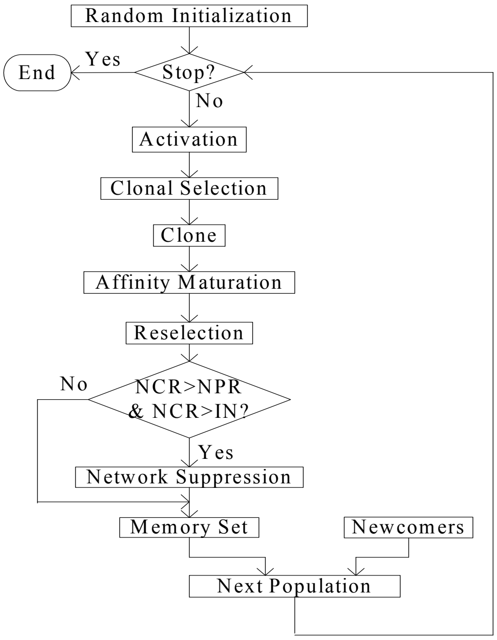


Figure 3.21 Generic AIS framework for MOP solving (NCR: the number of current non-dominated Abs; NPR: the number of non-dominated Abs in the last iteration; IN: the initial Abs size; Stop: at least one iteration step is executed).

3.6.2 Comparisons with other Bio-Inspired Methods

The superiority of the proposed immune based algorithm is based on the fact that AIS is inspired by a different regime of natural mechanisms. Thus, it is important to clarify the differences between AIS and other evolutionary-based algorithms (with GA as their representative) to finally highlight the extra advantages that AIS can deliver upon. The fundamental differences can be summarised as follows:

(1) **Reproduction mechanism:** AIS represents a type of asexual reproduction; while a population-based GA represents the counterpart. With the latter, the offspring is produced by crossing the chromosomes of both parents. Via the former, each 'Ab' copies itself to produce many clones.

(2) **Selection scheme:** for a population-based GA, good solutions are included in the mating pool with a high probability. For AIS, good solutions are always selected.

(3) **Evolution strategy:** for a population-based GA, the whole population evolves by using 'crossover'. The hypothesis is as follows: if both parents are the good ones their crossed offspring would have a high probability of becoming even better solutions; mutation is only used to jump out of the local optima hence the diversity is very important, otherwise, GA is likely to reach premature convergence; for AIS, since clones are duplicates of their predecessor the evolution of the population depends mainly on the mutation of the clones. Recombination is also applied to the clones. Only in the later stages of the search can this operator take effect.

(4) **Elitism:** for a population-based GA, during each generation, the whole population is replaced with the offspring after mating; hence 'elitism' has to be introduced to preserve good solutions found hitherto, otherwise they would be lost during successive generations; for AIS, the mutated clones and their predecessors are mixed together to compete for survival, hence 'elitism' is inherently embedded in AIS.

(5) **Population control:** for a population-based GA, since one has to specify the size of the mating pool in the first place the population size is thus fixed during each generation; if one only selects good solutions into the mating pool and makes the pool size flexible to the number of selected solutions GA could reach premature convergence due to its evolutionary strategy; a reasonable pool size is necessary so that in the early stage sub-optimal solutions can also get into the pool to increase population diversity; for AIS, a mating pool does not exist, hence the population should be flexible and finally controlled by the mutual influences of 'Abs'.

(6) **Diversity preservation:** for a population-based GA, diverity is maintianed through the mutation and an adequate population size; obviously, the population size is problem-contingent; for AIS, diversity is maintained through a prolific mechanism, affinitiy maturation, network suppression and the insertion of the newcomers.

(7) **Fitness (affinity) assignment.** For conventional evolutionary algorithms, ranking and fitness assignment are only based on the information from the objective space; AIS, however, combines both to calculate affinity. Hence, it effectively uses the information from both the objective and decision variable space.

Based on such differences, it can be concluded that AIS, and specifically PAIA2, possesses further strengths which cannot be found in the conventional evolutionary algorithms.

(1) **Adaptive population.** Due to point 5 mentioned above, PAIA2 possesses an adaptive population size which can adjust to an adequate size according to need of the problem under investigation. This adaptive rather than a fixed population leads to the following three advantages: 1) Initial population size is not problem-dependent; 2) more solutions can be obtained without significantly increasing the number of evaluations by tuning the network suppression threshold; 3) only necessary evaluations are exercised because only necessary population and clones are maintained and produced in each iteration step.

(2) **Good starting point.** By using the multi-stage optimisation procedure, an optimised solution can be included in the initial population to bias the search process, which reduces the computational load of the whole optimisation process.

(3) **Good convergence.** Due to points 3 and 7, a good balance between exploitation and exploration of the search space is achieved, especially when the problem has variable density in the objective and decision variable spaces.

(4) **Fast convergence.** The slow convergence observed in the previous works is eliminated, and fast convergence is claimed as one advantage of the proposed algorithm instead. In PAIA2, even a small initial size (e.g. 7) can lead to a very fast convergence because one is supposed to only select good ‘Abs’ and let them reproduce with an adaptive clone size. In the early iteration this cannot only provide sufficient ‘Abs’ to support the search but also accelerate the convergence speed.

3.7 Summary

In this chapter, an enhanced version of PAIA and a multi-stage optimisation procedure are proposed. For all experiments, significant results, either improving the convergence or reducing the computational cost are observed. In the next chapter, an evolutionary based clustering algorithm is described, which can bridge the gap between the *unsupervised learning* and *supervised learning*. An example of using the proposed clustering algorithm to extract initial fuzzy rule based systems from data is also introduced.

Chapter 4

An Evolutionary Based Clustering Algorithm

"An intelligent being cannot treat every object it sees as a unique entity unlike anything else in the universe. It has to put objects in categories so that it may apply its hard-won knowledge about similar objects encountered in the past, to the object at hand."

Steven Pinker, How the Mind Works, 2002

In this chapter, a brief introduction to data clustering is given, which is followed by the discussion of an evolutionary based clustering algorithm. Experimental studies of the proposed clustering algorithm were carried out in order to justify such hybridisation. Then, the relationship between *unsupervised learning* and *supervised learning* is expounded so that one can easily generalise it to the relationship between data clustering and the elicitation of the initial FRBS.

4.1 Introduction to Data Clustering

Knowledge in some sense can be defined as the ability that distinguishes the 'similar' from the 'dissimilar'. However, given the amount of information (in other words, data) encountered in the real life, such ability is sometimes limited, which in turn gives rise to limited knowledge. Without further abstraction, more information may simply lead to more difficulties for human beings to uncover the underlying structure. The key to the success of handling vast amount of data lies in the capability of retrieving representatives or prototypes from anfractuous information via a certain level of abstraction. Representing the data by fewer prototypes necessarily loses certain fine details, but achieves simplification and interpretability. One of the vital means which implements the abstraction of this type is data

clustering, which is also termed *unsupervised learning* in the machine learning community. A loose definition of clustering can be defined in terms of internal cohesion-*homogeneity* and external isolation-*separation* (Everitt *et al.*, 2001, p6) such that data points within the same cluster share the most common features, while data points of different clusters share the most dissimilar ones. A much more formal definition is given below, wherein the employment of the notations is not only for this chapter but also for the rest of the thesis:

> Given a set of data points $X_m = (x_{m1}, \dots, x_{md}, \dots, x_{mn}) \in \Psi$, where, $m = 1, \dots, N$ is the number of the data samples in the given data set; $d = 1, \dots, n$ is the dimensions of the feature variables; and Ψ is the feasible feature space, clustering is to group the given data points according to some similarity or dissimilarity measure ϖ. The result from clustering is a set of clusters C_i (or centres), wherein $i = 1, \dots, k$, such that each data point X_m is either assigned to one of the clusters, or has the membership of each cluster.

From the above definition, a number of crucial issues associated with data clustering arose, and some of them are listed below:

- What is the type of feature variables?
- Which feature variable is important and should be included in the clustering?
- What is the similarity or dissimilarity measure?
- How the similarity or dissimilarity measure is used?
- How many clusters is appropriate?
- Can data points belong to different clusters?

The first issue deals with what type of feature variables that a clustering algorithm can handle. Possible feature types include categorical values, continuous values, or a combination of both. The second issue relates to the feature selection or input selection so that irrelevant features are discriminated and excluded accordingly. The third issue is important since different similarity or dissimilarity measures will definitely affect the shape and the number of the obtained clusters to a great extent. Regarding the fourth and sixth issues, they are important since they provide criterions to categorise different clustering algorithms, although from different perspectives different categories may be obtained. The fifth issue relates to the cluster validation which deals with the problem of monotonic decrease associated with most similarity and dissimilarity measures in a bid to automatically make a decision on the optimal number of clusters. Summing up the above discussions gives the general steps normally involved in a data clustering task. Figure 4.1illustrates these steps.

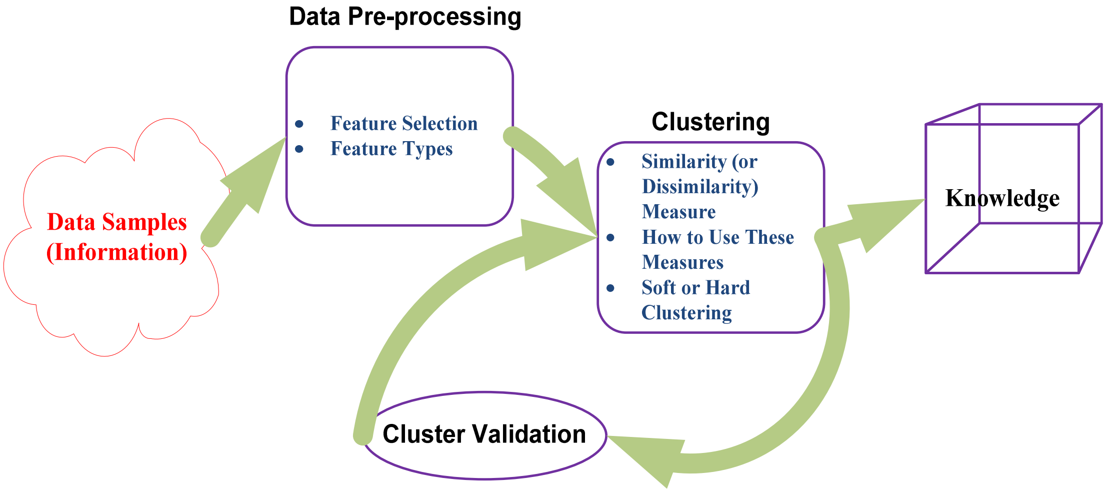


Figure 4.1 General steps involved in data clustering.

Despite the equal importance of each individual step shown in Figure 4.1, particular attention has been given in this chapter to two of them, namely *Clustering* and its *Validation,* due to the fact that an assumption of the pre-processed data has been made in this project. Readers are also referred to three comprehensive surveys (Jain *et al.*, 1999; Berkhin, 2002; Xu *et al.*, 2005) and a textbook (Everitt *et al.*, 2001) for other steps involved in data clustering.

4.1.1 Similarity and Dissimilarity Measures

One of essential issues in data clustering is to make a decision on how ‘close’ individuals are to each other, or how far apart they are. Such a decision is based on the measure of similarity or dissimilarity (*proximity* is another general term). Typically, a distance is the measure of dissimilarity and is normally used for continuous features, while a similarity measure is more important for categorical ones (Xu *et al.*, 2005).

In fact, such a measure itself has twofold meanings: the first implies the relationship between individuals, and the other one is referred to the relationship between clusters. The former serves as the constitutive factors of the latter, and the latter is the core for most clustering algorithms. Due to the apparent focus of continuous features in this project, only dissimilarity measure is discussed. Among many dissimilarity measures, Minkowski metric is the most popular one which works on the level of individuals:

$$\varpi(X_i, X_j) = \left(\sum_{d=1}^{n} |x_{id} - x_{jd}|^r\right)^{1/r} \quad r \geq 1 \tag{4.1}$$

where, d is the dimension of the ith and jth data points (X_i and X_j). Minkowski metric varies when r takes different values, and the *Euclidean* distance is a special case when r equals to 2.

Having the measure which can distinguish individuals does not necessarily mean that one can discriminate different groups. Nonetheless, a dissimilarity measure on the basis of individuals does build up a ground for the assessment of group proximity. The key is to represent clusters with their corresponding prototypes so that dissimilarity measures, such as Eq. 4.1, are still effective. Two very different techniques to account for group proximity exist, namely 'inter-cluster distance' and 'within-cluster variance', and various approaches have been proposed depending on the ways of extracting the prototypes. The approaches under the stream of 'inter-cluster distance' include the nearest-neighbour technique, the farthest-neighbour technique, the median technique. Figure 4.2 shows the difference between these techniques. There are techniques, such as *average linkage* (see Section 4.1.2.1), which do not require prototypes explicitly. However, every single individual can be *de facto* regarded as a prototype in such a case.

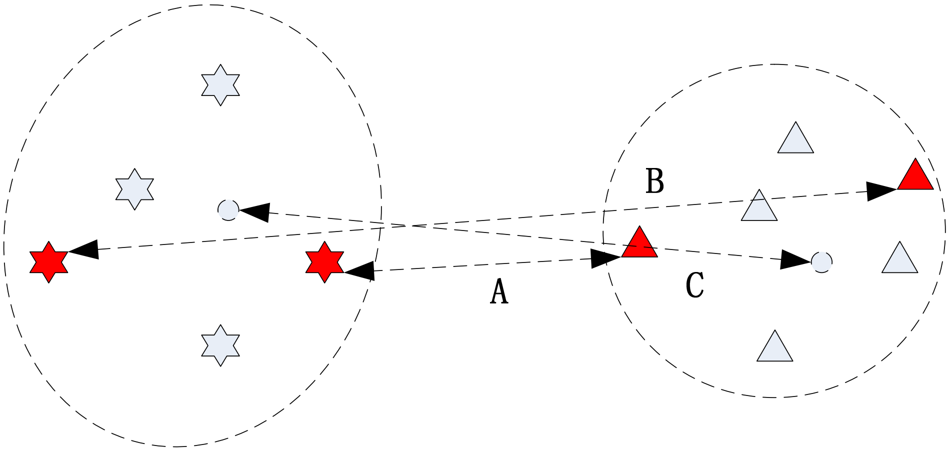


Figure 4.2 Illustration of different prototypes in 'inter-cluster distance': A. *nearest-neighbour technique*; B. *farthest-neighbour technique*; C. *median technique*.

As far as 'within-cluster variance' is concerned, a centroid is normally used to represent the core of the corresponding cluster. Hence, the variance of the individuals within a cluster can be measured as the sum of the squared error between the individuals (X) and the core (C) as shown in Eq. 4.2, where k is the number of clusters and N is the number of the individuals.

$$\varpi(C_l) = \sum_{X_m \in C_l} \|X_m - C_l\|^2 \quad l = 1, \dots, k; m = 1, \dots, N \tag{4.2}$$

In the broad sense, clustering is an optimisation process in that it tries to find a set of optimal clusters such that an ‘inter-cluster distance’ is maximized and a ‘within-cluster variance’ is minimised (see Section 4.1.2.2). There are techniques that do not explicitly use proximity metric, such as *mixture model* (see Section 4.1.2.2) and density-based clustering (see Section 4.1.2.3). However, the implicit use of such metric can still be found in those methods.

4.1.2 Classification of Clustering Algorithms

There are different ways to categorise clustering algorithms. They all hold similar objectives, but out of different considerations. One possible classification consists of hierarchical clustering, partition-based clustering, density-based clustering, and search-based clustering.

4.1.2.1 Hierarchical Clustering

Hierarchical clustering is so far the most popular clustering scheme, which builds a hierarchy of clusters by successive *agglomerative* or *divisive* operations. In the *agglomerative* case, clustering starts from the individual data point by considering it as a cluster. The calculation of the linkage (distance) between different clusters is then carried out, which leads to the fusion of two closest clusters into a bigger one. The whole process is iterative until a certain level of granulation is achieved. For the *divisive* version, it is the other way around. The whole data set is treated as a single cluster at the start. The similarities of each individual to the other individuals in the same group are then calculated so that a ‘splinter cluster’ can be formed which only contains the most dissimilar one from the main cluster. Some individuals in the main cluster will finally join the ‘splinter cluster’ since they are closer to the ‘splinter cluster’ than to the main cluster, which completes the divisive operation once. Such division repeats until each data point is classified as a singleton cluster. Due to the high demand of computation (Xu *et al.*, 2005), *divisive* method is not widely used. However, the observations for *agglomerative* clustering are still hold for *divisive* clustering in most cases.

Linkage plays an important role in the clustering process, irrespective of whether it is *agglomerative* or *divisive*. For *Agglomerative* clustering, the representatives are as follows:

single linkage (Sneath, 1957), *complete linkage* (Sorensen, 1948), *average linkage* (Sokal *et al.*, 1958), *centroid linkage* (Sokal *et al.*, 1958), *median linkage* (Gower, 1969) and *Ward's method* (Ward, 1963). More complicated variants, such as CURE (Guha *et al.*, 1998), are adapted from the above basic linkages. Different linkages normally lead to different clustering results, e.g. *Single linkage* tends to find irregular (chaining) clusters since it calculates the distance between two clusters via the *nearest-neighbour* technique, while *complete linkage*, due to the use of the *farthest-neighbour* technique, tends to find compact clusters with equal diameters (Everitt *et al.*, 2001).

Figure 4.3 shows such discrepancy when different linkages are applied to the same data set. Without prior knowledge of data and the objective of clustering, it is hard to estimate from Figure 4.3 which represents a better choice. The assessment of the clustering result is beyond the scope of this thesis. However, in Section 4.4, it is pointed out that if the objective of clustering is to elicit an initial data-driven FRBS, then methods which can produce compact clusters are superior. And for the same reason, if the objective of clustering is to discriminate objects with irregular shapes, then methods which can identify chain-like clusters have priority.

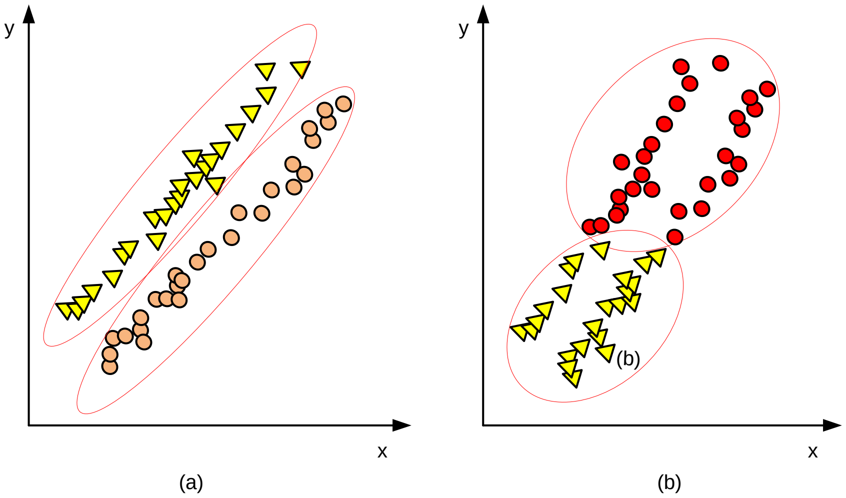


Figure 4.3 An illustrative example of a two-dimension data set: (a) single linkage tends to find chain-like clusters; (b) complete linkage tends to find compact clusters.

Hierarchical clustering is a one-pass method. Once a fusion or division has been made, individuals cannot change their identities within the hierarchy. Hence, if the first move is based on the wrong suggestion, either merging or separation, there is no chance to rectify such defect afterwards. Another drawback is the high computational demand, especially for the high dimensional data. The most distinctive feature of hierarchical clustering lies in its capability of conveying a data structure-map so that one can decide on which abstraction level the clusters are retrieved.

4.1.2.2 Partition-Based Clustering

Partition-based clustering contains a rich class of developments. Despite such diversity, they all divide data into a pre-defined number of partitions (clusters) and gradually improve the quality of such partition by reassigning individuals among clusters. Unlike hierarchical clustering, partition-based clustering only obtains a single partition of data instead of a clustering structure (Jain *et al*., 1999). *Squared error clustering* is the representative of this class.

Squared error clustering is an instance of the utilisation of the 'within-cluster variance'. The general form of the squared error is described as follows:

$$\varpi(C_1, C_2, \dots, C_k) = \sum_{l=1}^{k} \sum_{m=1}^{N} (\mu_{ml})^{\lambda} \|X_m - C_l\|^2 \quad l = 1, \dots, k \tag{4.3}$$

$$C_l = \left(\sum_{m=1}^{N} (\mu_{ml})^{\lambda} \cdot X_m\right) / \left(\sum_{m=1}^{N} (\mu_{ml})^{\lambda}\right) \tag{4.4}$$

where;

μ_{ml}: the membership of *m*th individual to the *l*th cluster;

λ: the fuzzification parameter.

During the course of the clustering, centroids are updated using Eq. 4.4. Depending on the manners the membership μ is defined, squared error clustering can be further divided into two categories, namely hard clustering and soft (fuzzy) clustering. When μ takes continuous values between (0, 1) and is updated during every iterative step, it results in fuzzy clustering. In such a case, individuals no longer belong to a unique cluster. Instead, they pertain to every cluster with a certain degree of membership. Fuzzy *C*-Means (FCM) devised by Bezdek (1981) is the most well-known one of this type, which has an intuitive connection with FRBS (see Section 4.4). When μ takes binary values, i.e. 0 or 1, it leads to hard clustering. *K*-Means algorithm (Hartigan, 1975, 1979) falls into this category. Due to the use of the 'within-cluster variance', squared error clustering tends to find compact and hyper-spherical clusters. The computational cost of squared error clustering is normally less than that of *hierarchical clustering* so that it has the potential to work with large data sets. However, a major drawback of this type lies in its sensitivity to the initial partitions, which not only result in a convergence to the local optima, but also produce different clusters given different initial settings. However, given the simplicity of squared error clustering, it is still being widely used and many variants (hybridisations) have been proposed to offset the mentioned problems (see Section 4.1.2.4 and 4.2.3).

4.1.2.3 Density-Based Clustering

Instead of using distance-based similarity measures, density-based clustering utilises the local density of points to group similar data. The motivation behind such an idea stems from the intention of grouping non-convex (or chain-like) clusters which generally represents a great challenge for a distance-based clustering approach. *Density-based clustering* normally involves two steps:

(1) The first step estimates the density function associated with each data point so that a so-called *density attractor* of the defined density function can be found via optimisation techniques;

(2) The second step consists of investigating the densities of the *density attractors* and each data point attached to these *density attractors*; if both densities are greater than some threshold ξ then a density-based cluster is formed by connecting the corresponding density attractors and including the attracted data points.

In order to estimate the density function kernel density estimation and *k*-nearest neighbour approach are commonly adopted. Although, density-based clustering can successfully classify chain-like data points, it suffers from the problem when clusters have different densities.

4.1.2.4 Search-Based Clustering

Search-based clustering solves a clustering problem by viewing it as an optimisation problem. By iteratively searching for the optimum of the objective (cost) function, a set of optimal clusters will emerge. Search-based clustering covers miscellaneous implementations which may be overlapped with other clustering categories, e.g. partition-based clustering and density-based clustering. In terms of the optimisation techniques employed in this type of the clustering, it ranges from gradient based optimisation to ANNs and to EAs. The superiority of using EA-based clustering over other optimisation techniques lies in its global search capability so that it is less sensitive to the initial settings (Bezdek *et al.*, 1994; Maulik *et al.*, 2000).

4.1.2.5 The Relationship between Different Clustering Categories

In some sense, classification of clustering algorithms is by itself a clustering problem. Hence, different similarity measures (in this case, it is the criterion that groups similar clustering algorithms) are bound to result in different classifications, which also means that no matter how hard one tries to separate the resulting categories a certain level of association can still be found.

The intersection between hierarchical clustering and density-based clustering lies in the fact that a single linkage used in the former resembles the idea of the latter in that it views clusters as a connected dense component which can grow in any direction that density leads (Berkhin, 2002). If a complete linkage is applied, the performance of hierarchical clustering is more like that of partition-based clustering which normally leads to the ellipsoid-shape clusters. Both partition-based clustering and density-based clustering can be viewed as the special cases of search-based clustering in that the former is trying to relocate centres so that the objective function is optimised and the latter is trying to find the maximum peaks of the density functions. There is not a ubiquitous clustering algorithm that can be applied to every application. The choice of the type of the clustering algorithms depends heavily on the nature of the problem.

4.1.3 Cluster Validation

Cluster validation relates to the question of how many clusters are more adequate. In most applications, such as partition-based clustering and search-based clustering, the user has to estimate the number of clusters and fix this number during the search process. For hierarchical clustering, even if a complete hierarchy has been obtained one still has to decide the abstraction level so that a partition to the users' interests can be retrieved. An informal way of deciding this number in hierarchical clustering involves the observation of large changes in the fusion level so that a so-called *best cut* can be found to cut the dendrogram. A similar philosophy has been applied to partition-based clustering and search-based clustering by plotting the values of the clustering criterion against the number of groups. Large changes of levels in the plot are normally the indication of an 'optimal' partition (Everitt *et al.*, 2001).

However, such informal approaches are very subjective due to the fuzzy definition of 'large'. In order to overcome such subjectivity, a number of cluster validity indices (Bezdek, 1974; Fukuyama *et al.*, 1989; Xie *et al.*, 1991; Chen *et al.*, 2004) have been introduced in order to detect the 'right' number of partitions. The idea is to transform the clustering criteria, which is initially monotonic decrease with the increased number of partitions, into cluster validity index such that one can associate a minimal turning point of the index with the 'right' number of groups. Although some methods, such as Subtractive Clustering (Chiu, 1994) and density-based clustering do not require *a priori* the number of clusters, the user still has to decide on the radius for the former and thresholds for both cases, which are subjective and have a significant impact on the number of the clusters. This issue will be discussed again at the end of the chapter and in Chapter 5 since it is closely related to the number of rules in a data-driven FRBS.

4.1.4 Type of Clustering Used in This Project

As mentioned in Section 4.1.2.5, choosing the appropriate type of the clustering method is more of art than science. It depends heavily on the nature of the problem and on the user's intention. Since the main aim of this thesis is to extract transparent fuzzy predictive models, the task of clustering is reduced to the elicitation of an initial data-driven FRBS. Such a modelling task normally favours clustering techniques which can produce compact clusters. Hence, hierarchical clustering with complete linkage and partition-based clustering are the ideal candidates. However, in view of the computational cost, partition-based clustering, in particular *K*-means clustering seems more appropriate in the case of this present research due to its simplicity in implementation and its low computational demand, especially in the presence of a large data set. In order to address the well-known 'sensitivity' problems associated with *K*-means clustering, a real-coded GA (Deb *et al.*, 2002) is incorporated into *K*-means clustering. The effect of such hybridisation is an enhanced search by incorporating the local search capability rendered by the hill-climbing optimisation with the global search ability provided by the GAs. Section 4.2 details such a choice.

4.2 Hybridisation of G3PCX and K-means (G3Kmeans)

4.2.1 Introduction to G3PCX

G3PCX (Deb *et al.*, 2002) is the abbreviation of the Generalised Generation Gap (G3) model and the Parent-Centric Recombination (PCX). It is a computationally efficient genetic algorithm, specially designed for the real parameter optimisation. The design of G3 model emanates from the realisation that a population alteration model also plays a vital role in a real-valued optimisation process, and it should be different from a standard binary based genetic algorithm. The G3 model includes the following four steps:

Step 1: From the population P, select the best parent and '$\mu - 1$' other parents randomly, where μ is set to 3 by Deb (Deb *et al.*, 2002).

Step 2: Generate λ offspring from the chosen μ parents using the PCX, where λ is set to 2 according to Deb (Deb *et al.*, 2002).

Step 3: Choose two parents at random from the population P.

Step 4: From a combined subpopulation of two chosen parents and λ created offspring, choose the best two solutions and replace the chosen the chosen two parents (in step 3) with these solutions.

In terms of the selection scheme of the above G3 model, it is rather similar to the one described in Section 3.2.1 where elitism is also adopted implicitly by selecting the best solutions from the combined population of parents and their progeny. As far as the variation operators are concerned, Deb (2001, p. 110-112) pointed-out that a binary coded GA or a real-valued GA with simple naive crossover is no longer sufficient for the real-parameter optimisation. Hence, a new crossover (recombination) based on the parent-centric principle was proposed, which in many ways resembles the affinity maturation operator used in PAIA2 (refer to Section 3.3.1) in that they are both based on the assumption that potential good solutions are most likely to appear in the region close to their parents which have qualified the 'fitness test' in the selection operator. The PCX-based operator first calculates the mean vector $\vec{g}$ of the chosen μ parents so that for each offspring, one parent $X^{(P)}$ is chosen with equal probability. The direction vector $\overrightarrow{d^{(P)}} = X^{(P)} - \vec{g}$ is then calculated. Afterwards, from each of the other $(\mu - 1)$ parents, perpendicular distances D_i to the vector $\overrightarrow{d^{(P)}}$ are computed and their average $\overline{D}$ is found. The offspring is thus created as follows:

$$X_{offspring} = X^{(P)} + \omega_\zeta \cdot \overrightarrow{d^{(P)}} + \sum_{i=1, i\neq p}^{\mu} \omega_\eta \cdot \overline{D} \cdot \vec{e}^{(i)} \tag{4.5}$$

where $\vec{e}^{(i)}$ are the $(\mu - 1)$ orthonormal bases that span the subspace perpendicular to the vector $\overrightarrow{d^{(P)}}$. The parameters ω_ζ and ω_η are zero-mean normally distributed variables with variance σ_ζ^2 and σ_η^2 respectively. Deb *et al.* (2002) further compared PCX with various Mean-Centric Recombination (MCR) operators and concluded that the use of PCX is computationally more efficient than MCR operators, especially in the early iterations in which the centroid of the chosen parents may have a large distance from each parent. Hence, creating potential good solutions around such centroid in the early iterations may not be a clever choice, which more often than not requires a large number of iterations or a large population size to eventually converge. Figure 4.4 visualises the philosophy behind PCX and shows the density of the potential solutions produced by three parents in a 2-dimensional decision variable space (represented as the red circles at [1.2 1.1], [1.2 1.25], [1.1, 1.1]) using the PCX recombination.

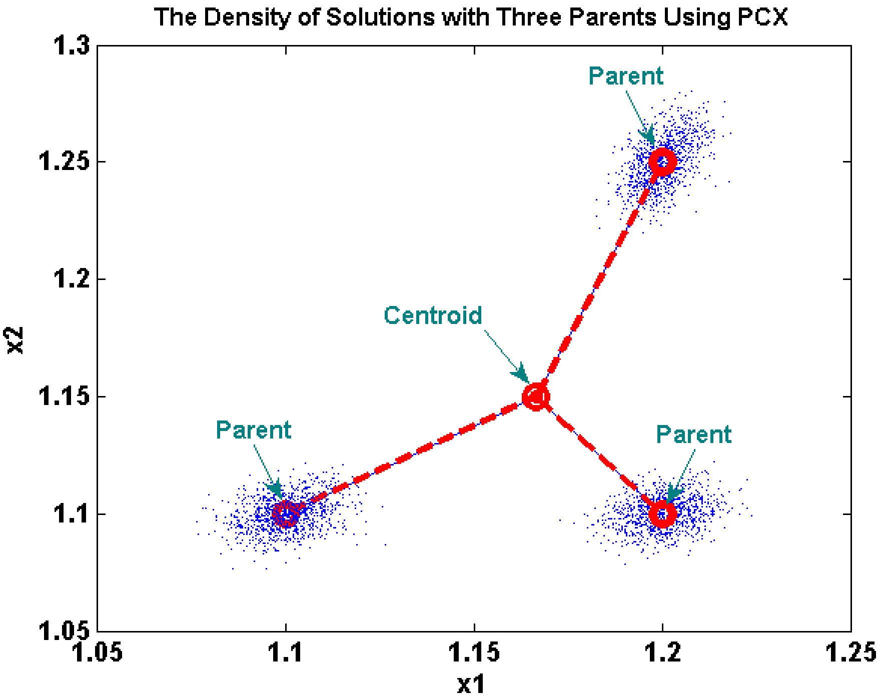


Figure 4.4 The density of solutions with three parents using PCX.

4.2.2 Description of G3Kmeans

The G3PCX algorithm introduced in Section 4.2.1 is hybridised with *K*-means clustering algorithm with the aim of overcoming the well-known problems associated with *K*-means algorithms, viz. its sensitivity to the initialisation and its convergence to the local optima. Such a hybridised clustering algorithm is termed G3Kmeans and the detailed steps are described as follows:

Step 1: **Initialisation:** The randomly generated 'k' cluster centres are encoded in each chromosome in a concatenated form. 'P' chromosomes are generated in the initial population.

Step 2: **Assigning data points:** Each data point is assigned to one cluster with the centre of C_i using Eq. 4.6:

$$X_m \in C_i : if \begin{cases} \|X_m - C_i\| < \|X_m - C_l\| \\ m = 1,2,\dots,N; i,l = 1,2,\dots,k; l \neq i \end{cases} \tag{4.6}$$

where, $\| \ \|$ is the Euclidean norm and N is the number of data samples. After the assignment, cluster centres encoded in the chromosome are updated by calculating the mean value of each cluster using Eq. 4.4.

Step 3: **Fitness computation:** the fitness value of each individual is calculated using Eq. 4.3, and for clarity it is rewritten here by replacing the membership μ_{ml} with a binary value of 0 or 1:

$$\varpi(C_1, C_2, \dots, C_k) = \sum_{l=1}^{k} \sum_{X_m \in C_l} \|X_m - C_l\|^2 \quad l = 1,\dots,k \tag{4.7}$$

where, ϖ is a within-cluster-distance metric to be optimised (minimised), and $C_1, C_2, \dots C_k$ are k cluster centres.

Step 4: **Parent-Centric Crossover (PCX):** Generate λ offspring from the μ parents using the PCX recombination mentioned in Eq. 4.5.

Step 5: **Fitness computation:** the cluster centres and fitness values of the offspring are updated and calculated again as what have been done in the step 2 and 3 accordingly.

Step 6: **Parents to be replaced:** choose two parents at random from the population P.

Step 7: **Replacement:** From the combined subpopulation of two chosen parents and λ created offspring, choose the best two solutions and replace the chosen two parents (in step 6) with these solutions.

Step 8: **Iteration:** the aforementioned steps from step 2 are repeated for a specified generations or until the standard deviation of the fitness values of the last five iterations becomes less than a threshold *stable*, and the final solution is the one with the smallest fitness value at the end of the execution.

It is worth mentioning that in the following experiments within this chapter and the experiments afterwards all the user-specified parameters are set as those suggested by Deb *et al.* (2002) unless otherwise stated. Hence, $\sigma_{\zeta}^2 = \sigma_{\eta}^2 = 0.1, P = 100,\ \lambda = 2, \mu = 3, stable = 0.001$.

4.2.3 Rationale of the Hybridisation

In the last decades, we have seen many efforts in hybridising GAs with the conventional partition-based clustering algorithms (Bezdek *et al.*, 1994; Hall *et al.*, 1999; Krishna *et al.*, 1999; Bandyopadhyay *et al.*, 2002; Sheng *et al.*, 2006). As briefly mentioned in Sections 4.1.2.4 and 4.1.4, the ‘rationale’ behind such a hybridisation idea lies in the fact that most optimisation techniques used in partition-based clustering are inherently hill-climbing techniques which are very sensitive to the initial settings and may lead to convergence to local optima. One of the earliest GA-based clustering implementations was carried-out by Bezdek *et al.* (1994) by hybridising a GA with Fuzzy *C*-Means (FCM), in which a binary coded chromosome was adopted to represent cluster centres so that a GA can be used to iteratively search for the optimal fuzzy partitions. Hall *et al.* (1999) improved the efficiency of such a GA-based fuzzy clustering algorithm by coding the centres with a binary Gray code representation in which any two adjacent numbers are one bit different. The authors of both research contributions argued that coding centres in the chromosome is more efficient than coding the membership matrix (μ_{ml}, refer to Eq. 4.4 in Section 4.1.2.2). Similar efforts have been made to hybridise GAs with *K*-means clustering. Instead of coding centres in the chromosome, Murthy *et al.* (1996) proposed to code the cluster identity number which is assigned to each data point. Hence, the length of the chromosome is the same as the number of data points, which makes the algorithm vulnerable to the large data set. Krishna *et al.* (1999) proposed to code the hard membership matrix (μ_{ml}) in a binary form so that a GA framework can be applied to find the optimal hard membership matrix which minimises the 'within-cluster variance'. Bandyopahyay *et al.* (2002) acknowledged the comments made earlier by Bezdek (1994) and noticed that the clustering problem under a GA framework is actually a real-valued optimisation problem. Hence, a real-valued GA was adopted in their work. Cluster centres are encoded in the chromosome with floating-point values. Recently, Sheng *et al.* (2006) incorporated GAs into *K*-medoids clustering using an integer encoding scheme.

Despite the great achievements reported in the aforementioned research contributions, several related problems still deserve special attention:

(1) In the early implementations, binary-coded GAs were widely adopted. However, GA-based clustering can actually be viewed as a real-valued optimisation problem. Applying binary-coded GAs to such a continuous search space will result in a so-called 'Hamming cliffs' difficulty associated with certain strings (such as 0111 and 1000). In such a scenario, a transition to a neighbouring solution (in real space) requires the alteration of many bits. Hamming cliffs may cause difficulties in a gradual search, especially in a continuous search space (Deb, 2001, p. 110).

(2) Binary-coded GAs suffer from the problem of imprecision, especially when they are used for a real-valued optimisation problem. More precision simply means longer strings which will in turn increase the search space. It also means a large population size in order to have an effective search.

(3) Both Murthy *et al.*'s work (1996) and Krishna *et al.*'s work (1999) cannot deal with a large data set since the length of the chromosome increases as the number of the data points increases.

(4) Although Bandyopahyay *et al.* (2002) adopted a real-valued GA, a so-called Naive crossover (single-point crossover) was used in their work, which is similar to the crossover operators used in binary-coded GAs. However, as mentioned by Deb (2001, p.112), this crossover operator does not have an adequate search power. Hence, it surrenders its search responsibility to the mutation operators, which may not be effective as well.

In the light of the above problems, the proposed G3Kmeans algorithm chooses a real-valued GA as its optimisation method which only encodes cluster centres in the chromosome. Hence, the length of the chromosome rests only with the number of the cluster centres. The search power of G3Kmeans is significantly enhanced by combining PCX with the hill-climbing operator. Such a combination takes full advantage of global search capability mainly attributed to the PCX recombination and local search ability rendered by the hill-climbing operator. As a result, G3Kmeans is more robust to the initialisation as opposed to other conventional partition-based clustering and is more efficient than the mentioned algorithms of the same kind. In the next Section, two synthetic data sets, iris data set and a real data set from the steel industry are utilised to validate the proposed G3Kmeans. The results are then compared to those of FCM, *K*-means, Subtractive clustering, GA-clustering (Murthy *et al.*, 1996) and KGA (Bandyopadhyay *et al.*, 2002). Comparisons on the use of different clustering algorithms to elicit an initial data-driven FRBS are also provided in Section 4.4.3 via a benchmark example.

4.3 Experimental Studies

4.3.1 Artificial Data Sets

4.3.1.1 Test Problem 1

The first test problem consists of 4 randomly generated Gaussian clusters around the nominal cluster centres[4.1]. Each cluster contains 100 data points as shown in Figure 4.5. Figure 4.6 shows the evolution curve of G3Kmeans[4.2]. Although the evolution takes 11 generations to finish it only takes 5 generations to reach the minimum objective value.

In order to obtain a quantitative comparison with different clustering algorithms, objective values are calculated using Eq. 4.7 and are used as the measure for the algorithms' efficacy. The objective value of the original clusters is also computed using the nominal centres as the baseline to see if a specific clustering algorithm can approach to the nominal centres as close as possible. Table 4.1 summarises the results of FCM, Subtractive clustering, *K*-means and G3Kmeans algorithms respectively. The results are the average values of 20 independent

[4.1] They are called nominal centres here since the objective value of the clusters generated around these centres may not represent the minimum objective value as opposed to those of the identified clusters (see Table 4.1 for more details).

[4.2] The objective value is calculated using the normalised data and this is held for the following problems unless otherwise stated.

runs. The standard deviations of the results are also calculated to show if the algorithm is robust to different initialisations and runs.

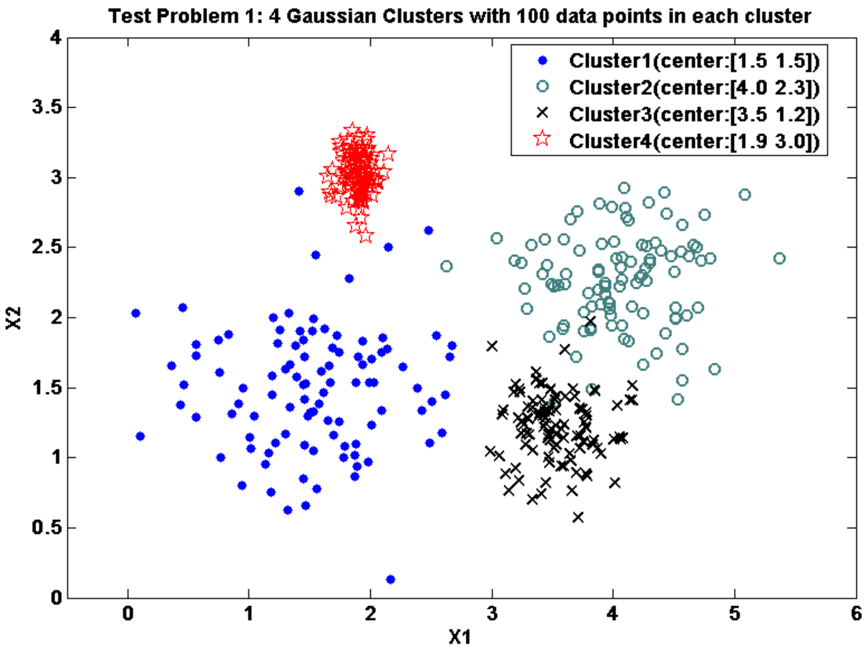


Figure 4.5 4 Gaussian clusters with 100 data points per cluster.

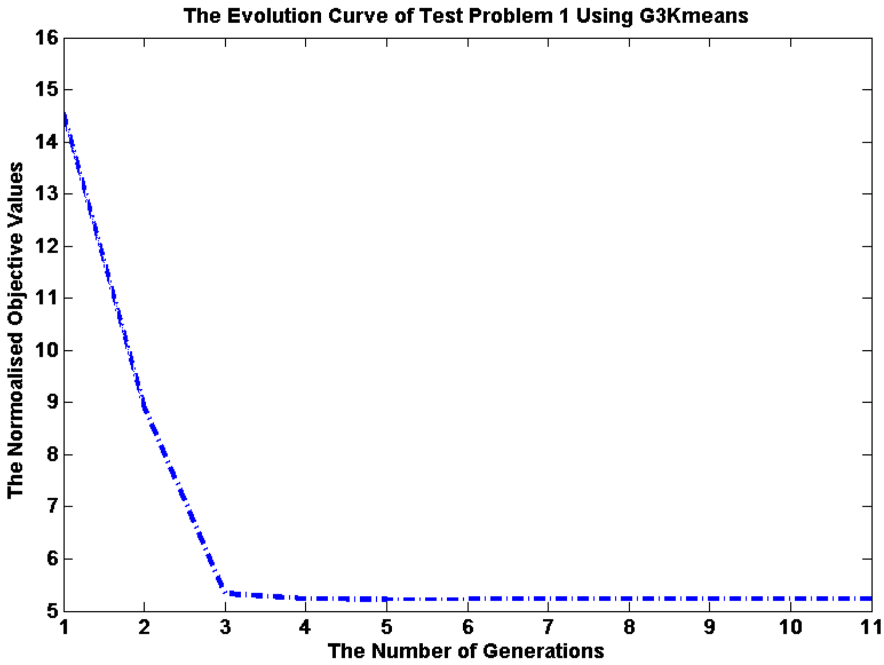


Figure 4.6 The evolution curve of the first test problem using G3Kmeans.

TABLE 4.1

COMPARISONS OF THE OBJECTIVE VALUES BETWEEN DIFFERENT CLUSTERING ALGORITHMS ON TEST PROBLEM 1

Methods	Maximum	Minimum	Mean	Standard Deviation	Time (second)
Original Clusters*	5.2880	5.2880	5.2880	0	-
FCM	5.2533	5.2522	5.2522	5.1640e-005	0.0271
K-means	5.2284	5.2284	5.2284	0	**0.0090**
Subtractive Clustering	7.6686	7.6686	7.6686	0	0.0706
G3Kmeans	**5.2284**	**5.2284**	**5.2284**	**0**	0.7283

*The objective value of the original clusters is obtained using nominal *centres*.

Figure 4.7 displays the identified cluster centres obtained by G3Kmeans.

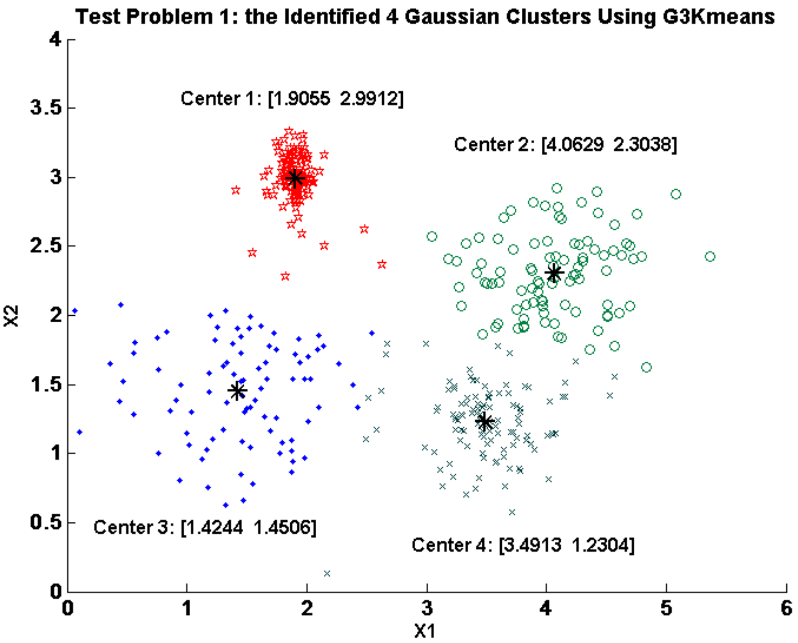


Figure 4.7 The identified 4 clusters of the first test problem using G3Kmeans.

For this simple problem, *K*-means and G3Kmeans algorithms can both approach to the global optimal partition (nominal centres) as their objective values represent the minimum ones among all the candidates. The standard deviations of K-means algorithm on test problem 1 are zero, which means for this problem there are no local optima. The FCM based algorithm constantly led to near optimal solutions which are very close to the nominal centres. The small variations of the objective values associated with FCM indicate that even for this simple problem different initialisations will inevitably lead to slightly different results. The radius of Subtractive Clustering is set to its default value, i.e. 0.5. The results produced via Subtractive Clustering consistently show its lack of accuracy, not only in this problem but also in the subsequent test problems. Hence, Subtractive Clustering is normally used *a priori* as a method to estimate the number of clusters for other clustering algorithms. Figure 4.8 shows the distribution of the nominal centres and the identified cluster centres via different algorithms.

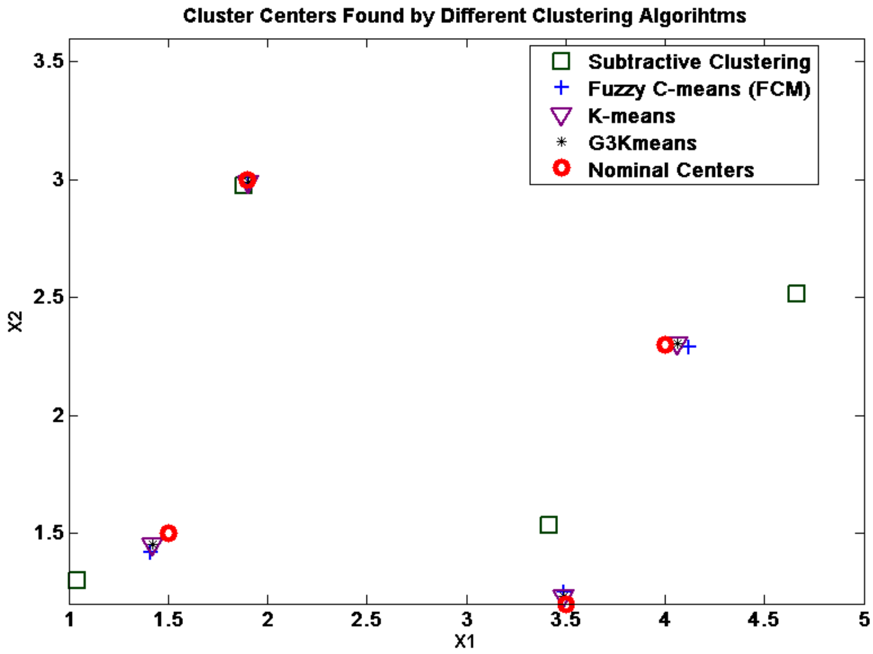


Figure 4.8 The distribution of identified cluster centres for test problem 1.

4.3.1.2 Test Problem 2

In order to test the robustness of the proposed algorithm to the data contaminated by random noise, the same Gaussian clusters as those used in test problem 1 are generated, which are then combined with 200 randomly distributed noisy data. Figure 4.9 shows the data distribution associated with this set.

The results are also the average values of 20 independent runs. The difficulties of this test problem lie in the facts that the classes are not well separated and the search space presents many local optima due to the presence of noise. Figures 4.10 and 4.11 show the evolution curve of the G3Kmeans[4.3] and the identified clusters using G3Kmeans clustering. G3Kmeans takes 11 generations to finish. However, the algorithm has already converged to the minimum objective value within 6 generations.

[4.3] The objective values are calculated using the whole data set, i.e. including noisy points.

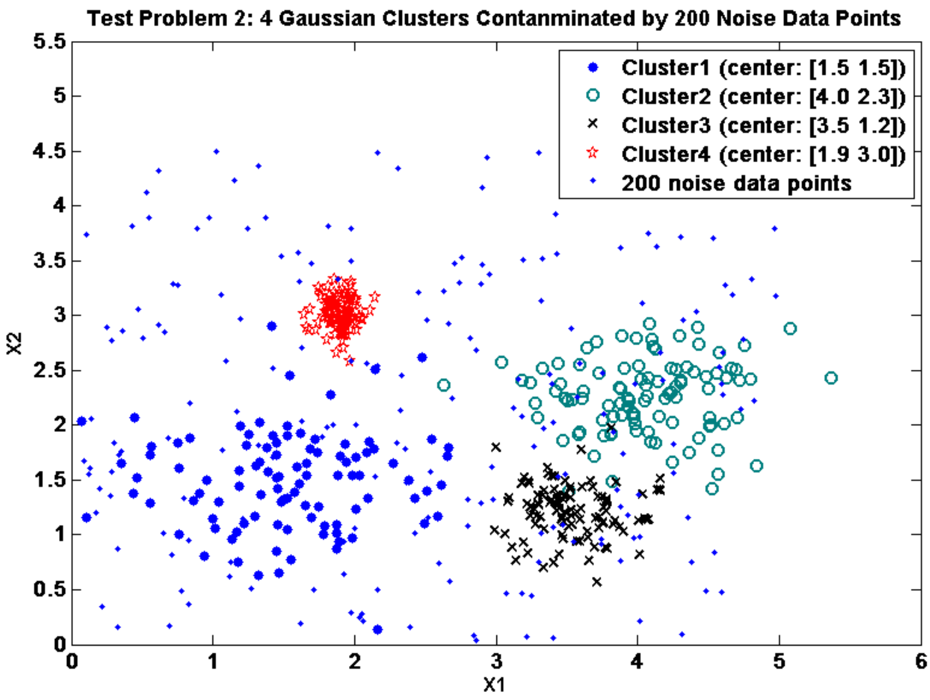


Figure 4.9 4 Gaussian clusters contaminated by 200 noise data points.

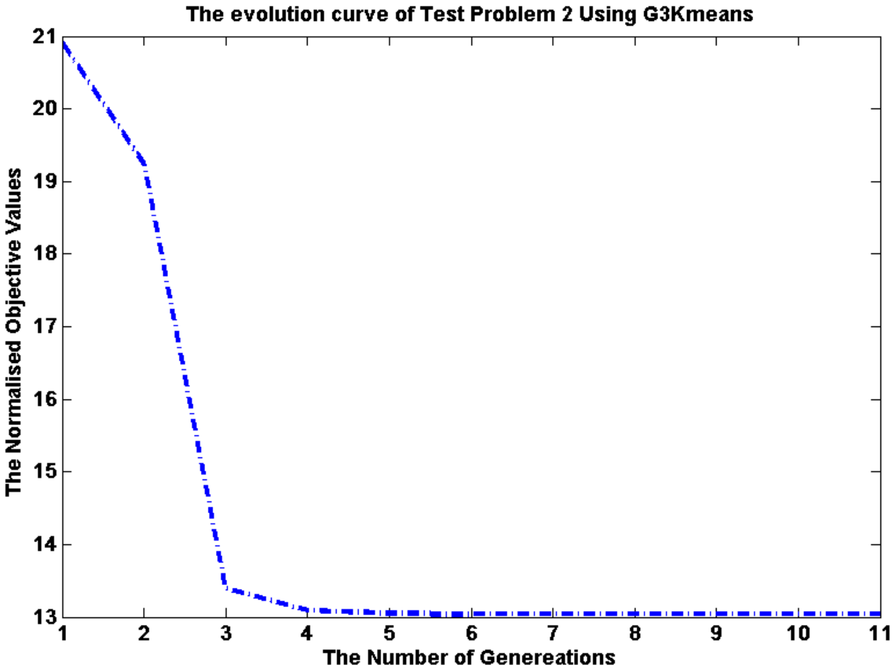


Figure 4.10 The evolution curve of the second test problem using G3Kmeans.

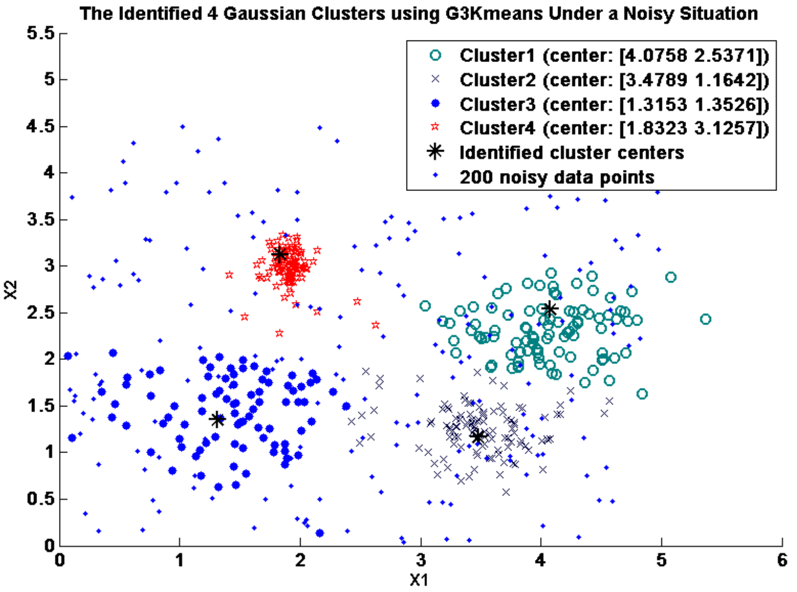


Figure 4.11 The identified 4 clusters of the second test problem using G3Kmeans.

Due to the presence of noise, the identified cluster centres are slightly different from those shown in Figure 4.7. However, in terms of classifying the data points into the right categories, the results from the first and the second test problems are very similar, which means G3Kmeans is robust even in the presence of noise. Table 4.2 summarises the results of FCM, Subtractive clustering, *K*-means and G3Kmeans.

TABLE 4.2

COMPARISONS OF THE OBJECTIVE VALUES BETWEEN DIFFERENT CLUSTERING ALGORITHMS ON TEST PROBLEM 2

Methods	Maximum	Minimum	Mean	Standard Deviation	Time (second)
Original Clusters	13.8697	13.8697	13.8697	0	-
FCM	13.1898	13.1895	13.1897	1.0138e-004	0.0269
K-means	18.3056	13.0382	13.5173	1.5881	**0.0130**
Subtractive Clustering	16.2954	16.2954	16.2954	0	0.2000
G3Kmeans	**13.0382**	**13.0382**	**13.0382**	**0**	0.8008

For this problem, both FCM and *K*-means are sensitive to the initialisations. Due to the presence of noise, test problem 2 includes several local optima, which correspond to the non-zero standard deviations produced by these two algorithms. In fact, *K*-means algorithm misclassifies the clusters twice in 20 runs. The larger the standard deviation, the more likely an algorithm depends on the initial condition. Figure 4.12 shows the distribution of the identified cluster centres via different clustering algorithms and the nominal centres. It can be seen from Table 4.2 and Figure 4.12 that the *K*-means algorithm not only depends on its initial condition but also leads to higher objective values which implies that the centres found by *K*-means may be remote from the nominal ones.

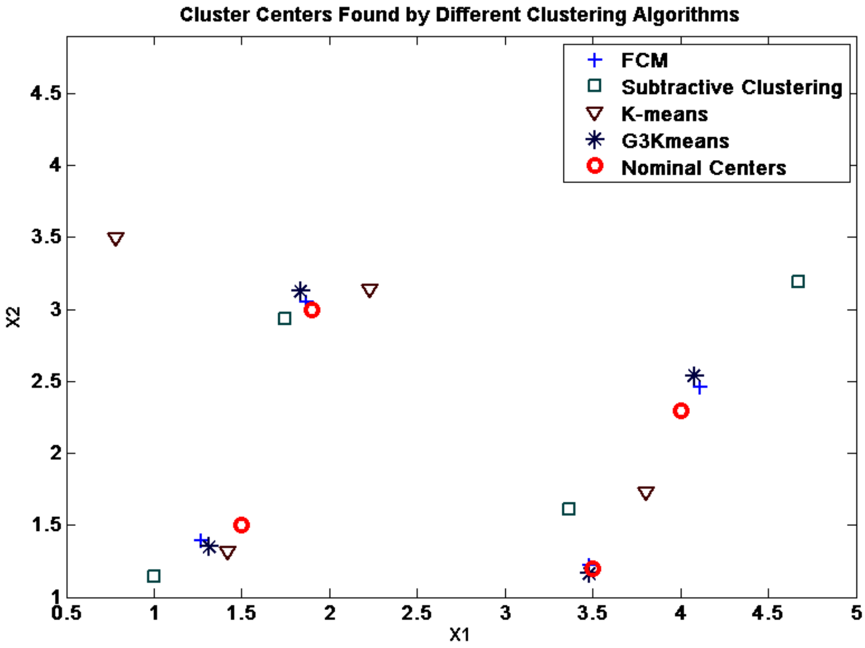


Figure 4.12 The distribution of the identified cluster centres for test problem 2.

4.3.2 Real World Problems

4.3.2.1 Iris data

In this section, G3Kmeans is compared with other GA-based *K*-means algorithms, e.g. GA-clustering (Murthy *et al.*, 1996) and KGA (Bandyopadhyay *et al.*, 2002), to justify the discussed ‘rationale’ of the proposed hybridisation (refer to Section 4.2.3) using the Iris data set (Fisher, 1936). The Iris data set consists of 150 patterns belonging to three categories of Iris. Each of the patterns is described by four real-valued features in ‘centimetres’, which are the sepal length, sepal width, petal length and petal width. Each of the categories consists of 50 patterns. The difficulties of the Iris data lie in the facts that the problem possesses two overlapped classes and presents many local optima. Figure 4.13 shows the evolution curve of G3Kmeans. It can be seen from this graph that G3Kmeans takes 11 generations to finish. However, the algorithm converged to the minimum objective value (6.9981) within 5 generations. Table 4.3 summarises the results[4.4] over 20 independent runs. The results of GA-clustering and KGA are extracted from Bandyopadhyay *et al.* (2002).

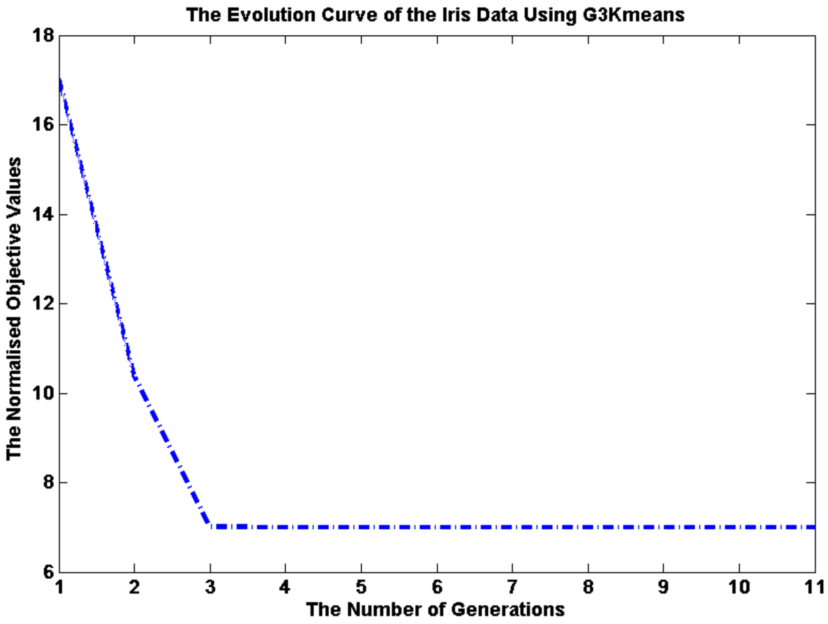


Figure 4.13 The evolution curve of the Iris data set using G3Kmeans.

TABLE 4.3

COMPARISONS OF THE OBJECTIVE VALUES BETWEEN DIFFERENT CLUSTERING ALGORITHMS ON THE IRIS DATA

Methods	Maximum	Minimum	Mean	Standard Deviation	Time (second)
FCM	79.4566	79.4516	79.4557	1.6000e-3	0.0252
K-means	142.9149	**79.0031**	95.0244	27.4598	**0.0052**
Subtractive Clustering	84.6800	84.6800	84.6800	0	0.0114
GA-clustering	139.7782	124.1274	135.4048	-	-
KGA	97.1008	97.1008	97.1008	0	-
G3Kmeans	**79.0031**	**79.0031**	**79.0031**	**0**	0.4623

For this problem, the *K*-means algorithm misclassified the clusters 4 times in 20 runs, which correspond to its maximum objective value shown in Table 4.3. A large standard deviation associated with *K*-means algorithm indicates that the Iris data set consists of many local optima and confirms that the *K*-means algorithm is vulnerable to such a scenario. The results of G3Kmeans are far superior to those of GA-clustering and KGA for the reasons discussed in Section 4.2.3. In fact, GA-clustering is unable to provide meaningful clusters within 1000

[4.4] Unlike the results presented in Table 4.1 and 4.2, the objective values shown in Table 4.3 are calculated using the original data so that the results produced by G3Kmeans can be compared with GA-clustering and KGA.

iterations (Bandyopadhyay *et al.*, 2002). This is mainly due to the coding scheme adopted by GA-clustering, which needlessly increases the search space and thus requires more computational efforts to converge. It is worth mentioning that Subtractive clustering cannot offer the correct number of clusters with its default radius. Hence, the radius is set to 0.6 for this problem in order to obtain the same number of clusters as produced by other clustering algorithms. Figure 4.14 shows the identified Iris classes and their centres via every two features.

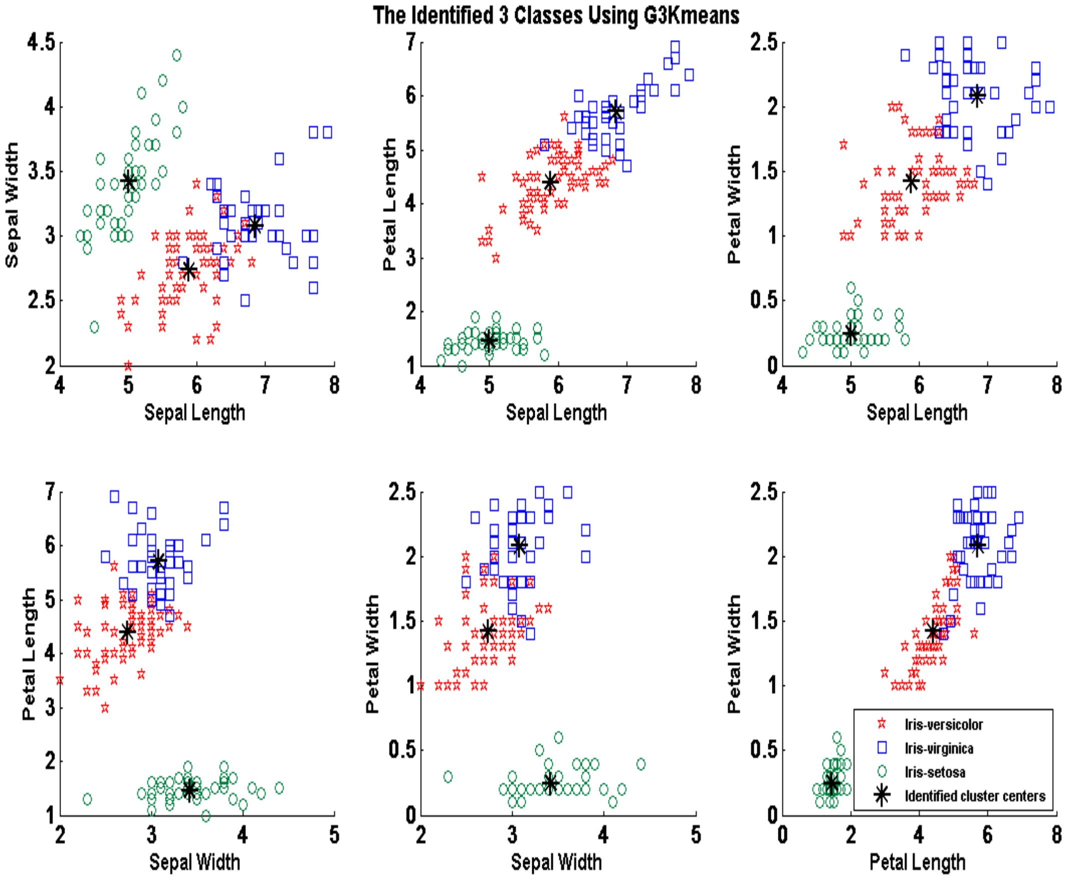


Figure 4.14 The identified 3 Iris classes using G3Kmeans.

4.3.2.2 Real Data from the Steel Industry

In order to test the scalability of the proposed G3Kmeans algorithm to high dimensional problems, a real data set from the steel industry, viz. Ultimate Tensile Strength (UTS), is used as the test problem. The UTS data set consists of 3760 data samples each of which has 16 dimensions. The detailed description of the UTS data set can be found in Section 6.3.4.

For a real world problem, one normally does not know how many clusters are inherent in the data. Conventionally, the best one can do is to use trial-and-error or the cluster validity mentioned in Section 4.1.3 to select the right number of clusters. As one will see in Sections 4.4.2 and 5.2, the problem of choosing the adequate number of clusters is somehow alleviated in our work by utilising a multi-objective optimisation framework. In such a case, one is allowed to overestimate the number of clusters in the first place so that the optimisation algorithm can find out the most appropriate number of clusters afterwards. Hence, the number of clusters for this problem is set to 12 as an overestimated number without any loss of generality. Figure 4.15 shows the evolution curve of G3Kmeans. As one can see from Figure 4.15, G3Kmeans took 36 generations to terminate. However, it actually converged to the minimum objective value (530.3131 in this particular run) within 30 generations.

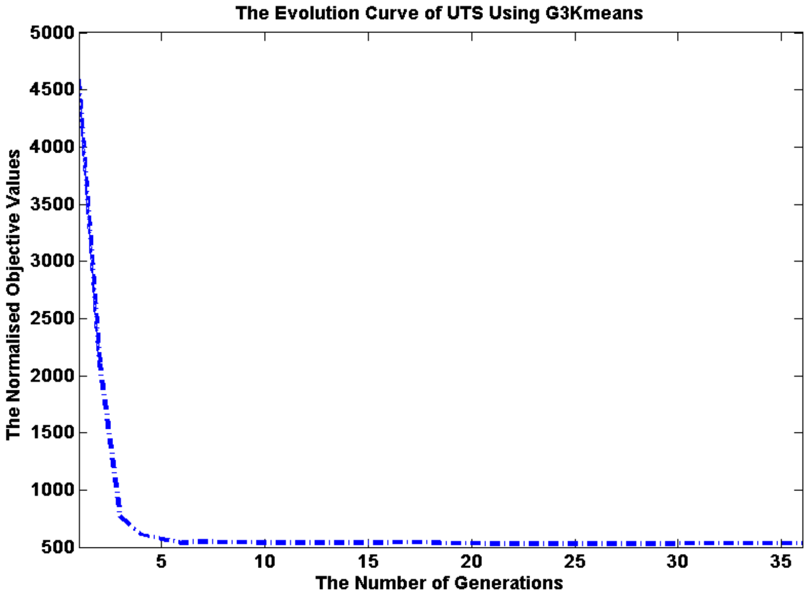


Figure 4.15 The evolution curve of the UTS data using G3Kmenas.

Table 4.4 summarised the results of G3Kmeans over 20 independent runs and compared them with the results produced by FCM, *K*-means and Subtractive Clustering. As can be seen from Table 4.4, G3Kmeans outperformed the other three clustering algorithms in terms of the objective values, and consistently finding near optimal clusters which are believed to be very close to the global optimal clusters. The *K*-means algorithm is very sensitive to the initial settings since the results produced by *K*-means represent the highest standard deviation among all the clustering methods. Although the results of FCM present a smaller deviation compared to those of G3Kmeans, FCM failed to approach the global optimal partition since its objective values are far higher than the smallest objective value found by G3Kmeans.

TABLE 4.4
COMPARISONS OF THE OBJECTIVE VALUES BETWEEN DIFFERENT CLUSTERING ALGORITHMS ON THE UTS DATA

Methods	Maximum	Minimum	Mean	Standard Deviation	Time (second)
FCM	1040.500	1034.100	1038.600	**1.4292**	1.4950
K-means	662.8515	547.2090	591.6639	30.5610	**0.3280**
Subtractive Clustering	828.2510	828.2510	828.2510	0	4.4580
G3Kmeans	**537.4079**	**524.7958**	**528.7017**	2.9452	48.3259

4.3.3 Discussions

As one can conclude from the above experiments, the performances of FCM, *K*-means and the proposed G3Kmeans are similar for simple clustering problems featuring low dimensionality and without local optima, e.g. test problem 1. The real power of G3Kemans lies in its capability of handling high dimensional and non-linear problems, which normally present many local optima.

Due to the parallel search of multiple search spaces and the possession of a population pool, it is not surprising that G3Kmeans generally takes more time to provide a final solution than FCM, *K*-means and Subtractive Clustering do. However, if one compares G3Kmeans with other algorithms of the same type, such as GA-clustering and KGA, one will conclude that G3Kmeans is more efficient than other GA-based clustering methods. For the Iris data set, both GA-clustering and KGA need 1000 iterations to provide the final solution which are equivalent to 50000 evaluation times. In fact, GA-clustering cannot even converge within 50000 evaluation times. For the same problem, G3Kmeans only takes 11 generations which equal to 1020 evaluation times to converge.

Such superiority is mainly attributed to the combined local and global search operators adopted in G3Kmeans, which are specially designed for the real-valued optimisation. The encoding scheme of the proposed method also ensures a reasonable search space as opposed to GA-clustering algorithm.

4.4 The Relationship between Clustering and FRBS

4.4.1 Identification of the Relationship

Generally speaking, clustering algorithms are unsupervised learning schemes. The main characteristic of unsupervised learning is to automatically ‘mine’ the relationship embedded within a group of unlabelled data without any structural assumptions about them. On the contrary, supervised learning schemes normally assume a known causal structure of the data, which means the inputs (i.e. feature variables) and the outputs (i.e. categories in a discrete space and real values in a continuous case) have been discriminated from the outset. Instances of supervised learning schemes include all types of learning classifiers and the predictive modelling methods based on the techniques such as ANN, Neuro-Fuzzy Systems (NFS) and evolutionary fuzzy systems.

In practice, unsupervised learning is usually exploited as the first learning step to induce knowledge especially when the ‘curse of dimensionality’ becomes a serious issue. One of the earliest such endeavours in the field of fuzzy modelling has been made by Yoshinari *et al.* (1993). In their work, structure-free fuzzy models are created based on a generalised fuzzy clustering approach. Since there is not any assumption about the data structure, the fuzzy model can be used in any direction. As a result, any variable can be estimated with the rest ones as the inputs. Such initial knowledge (fuzzy models or relations) can then be refined in the manner of supervised learning, which leads to a combined unsupervised and supervised learning scheme.

In the last two decades, such a combined learning approach has been successfully applied to the elicitation of FRBS (Chiu, 1994; Genther, 1994; Delgado *et al.*, 1996; Chiu, 1997; Delgado *et al.*, 1997; Stenes, 2000). Among many of such implementations, Chiu (1994) proposed a Subtractive Clustering algorithm which is specifically designed for the fuzzy rule-base modelling and can be viewed as an extended version of the mountain clustering algorithm (Yager *et al.*, 1994). Subtractive Clustering is operated on the product space of inputs and outputs and can automatically estimate the number of clusters (hence, the number of rules in FRBSs). Each cluster centre is in essence a prototypical data point that exemplifies a characteristic of the system. The identified cluster centres and radiuses correspond to the centroids and the spreads of the exponential membership functions that are used for the premise part of FRBSs. The parameters of the linear consequents are computed via a 'recursive least-squares' method. Chiu (1997) further extended the above fuzzy modelling methodology to a fuzzy classification scenario. A gradient decent algorithm was developed to tune the parameters pertaining to the membership functions in a bid to improve the classification accuracy. Genther *et al.* (1994) argued that fuzzy (soft) clustering is more suitable for the elicitation of FRBSs than hard clustering methods (e.g. Subtractive Clustering). Their argument is based on the fact that nature objects tend to belong with certain degrees of membership to all classes, which seems to have more intuitive connection with the concept behind the fuzzy sets. In order to associate the information provided by FCM with fuzzy membership functions, the authors approximated the projections of the cluster on each dimension via triangular membership functions. Such a 'projection' idea has also been adopted by Delgado *et al.* (1996), in which a set of fuzzy measures work in conjunction with a hierarchical clustering algorithm to automatically detect the suitable number of clusters. Such pre-processed clustering are then used to initialise the algorithms of the FCM type. Delgado *et al.* (1997) further proposed and compared various clustering based fuzzy modelling implementations, ranging from the direct use of the clusters' membership function (refer to Eq. 4.8) to the projections of the clusters, and from clustering on the product space of inputs and outputs to clustering on separate spaces. The conclusions drawn from their work are that the direct use of the clusters' membership function normally leads to an accurate initial fuzzy model; while on the other hand, projection based method tend to produce descriptive FRBSs at the cost of their accuracy.

Summing up the above discussions leads to the conclusion that clustering is incorporated into fuzzy modelling especially when the numerical data reflects a high dimensionality mapping between input and output spaces. The purpose of clustering is to extract the relationship between independent variables so that the initial fuzzy structure with only a conservative number of rules can be obtained. One may have the following options when one attempts to use the clustering based fuzzy modelling approach:

[1] Fuzzy (soft) clustering is operated on the product space of inputs and outputs. The resulted clusters are directly used to build the fuzzy model. Figure 4.16 (upper FRBS) illustrates this choice.

[2] Clustering (soft or hard clustering) is operated on the product space of inputs and outputs. The projections of clusters on each 'universe of discourse' form the fuzzy sets for each rule in FRBS. Figure 4.16 (lower FRBS) demonstrates such an idea.

[3] Clustering is operated on the separate input and output spaces. One of the same steps as those described in [1] and [2] is then used to build FRBSs.

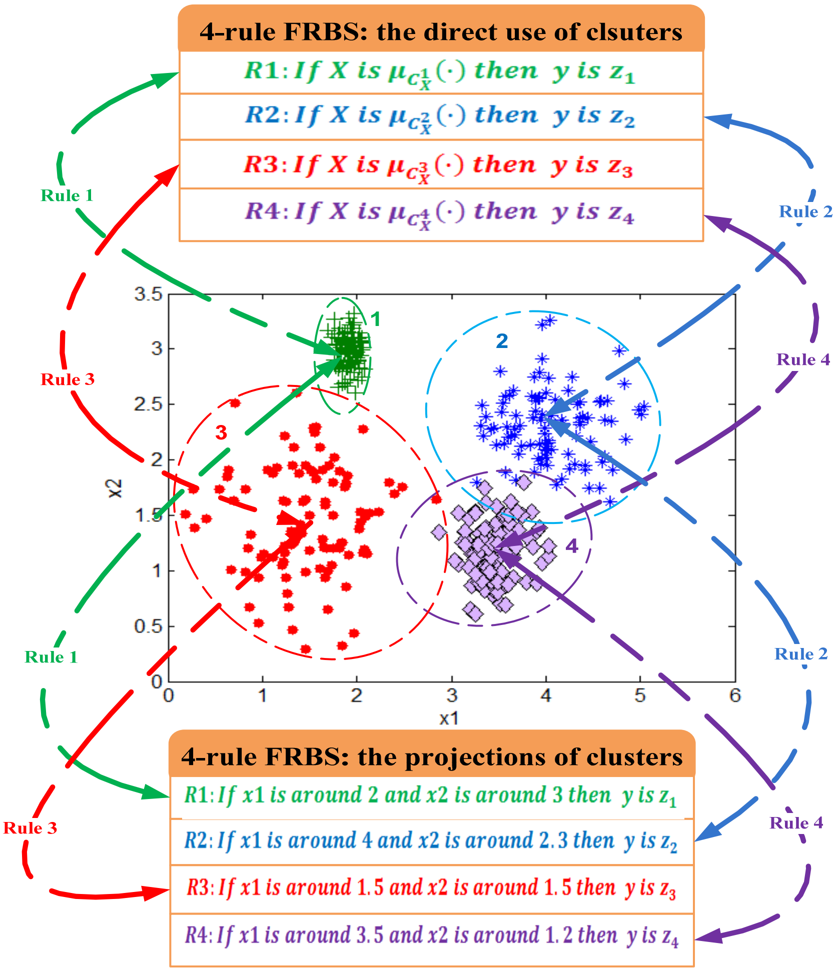

4-rule FRBS: the direct use of clsuters
$R1: If\ X\ is\ \mu_{C_X^1}(\cdot)\ then\ \ y\ is\ z_1$
$R2: If\ X\ is\ \mu_{C_X^2}(\cdot)\ then\ \ y\ is\ z_2$
$R3: If\ X\ is\ \mu_{C_X^3}(\cdot)\ then\ \ y\ is\ z_3$
$R4: If\ X\ is\ \mu_{C_X^4}(\cdot)\ then\ \ y\ is\ z_4$
Rule 1
Rule 2
Rule 3
Rule 4
Rule 1
Rule 2
Rule 3
Rule 4
1
2
3
4
x2
3.5
3
2.5
2
1.5
1
0.5
0
0
1
2
3
4
5
6
x1
4-rule FRBS: the projections of clusters
$R1: If\ x1\ is\ around\ 2\ and\ x2\ is\ around\ 3\ then\ \ y\ is\ z_1$
$R2: If\ x1\ is\ around\ 4\ and\ x2\ is\ around\ 2.3\ then\ \ y\ is\ z_2$
$R3: If\ x1\ is\ around\ 1.5\ and\ x2\ is\ around\ 1.5\ then\ \ y\ is\ z_3$
$R4: If\ x1\ is\ around\ 3.5\ and\ x2\ is\ around\ 1.2\ then\ \ y\ is\ z_4$

Figure 4.16 Creating FRBS through clustering: (1) upper FRBS is built by the direct use of clusters; (2) lower FRBS is built by the projections of clusters.

Figure 4.16 visualises the process of extracting a FRBS based on clustering methods for a two-input problem. For illustration purpose, clustering results are shown only in the input space. In practice, the results may be obtained from the product space of inputs and outputs, or from separate input and output spaces. In the following discussions, no difference has been made for the space from which clustering results are obtained. Rather, the emphasis is placed on how these clustering results are utilised for fuzzy modelling.

The upper FRBS shown in Figure 4.16 represents a rapid prototyping method (Delgado *et al.*, 1997) of emanating fuzzy models, where $\mu_{C_X^l}(\cdot)$ calculates the degree of membership to which a feature sample belongs to the lth cluster. Depending on different implementations, z_l

can be the linear function of the inputs, a singleton or the membership function $\mu_{C_Y^l}(\cdot)$. If FCM is employed for clustering, $\mu_{C_X^l}(\cdot)$ and $\mu_{C_Y^l}(\cdot)$ can be calculated as follows, where λ is the fuzzification parameter:

$$\begin{aligned} \mu_{C_X^l}(X_m) &= \left\{\sum_{i=1}^{k} \frac{\left\|X_m - C_X^l\right\|^2}{\left\|X_m - C_X^i\right\|^2}\right\}^{-1/(\lambda-1)} \\ \mu_{C_Y^l}(Y_m) &= \left\{\sum_{i=1}^{k} \frac{\left\|Y_m - C_Y^l\right\|^2}{\left\|Y_m - C_Y^i\right\|^2}\right\}^{-1/(\lambda-1)} \end{aligned} \qquad m = 1, \ldots, N \tag{4.8}$$

Rapid prototyping method is characterized by its easy implementation and yet accurate predictions. However, the apparent drawback associated with this method lies in the fact that the above advantages are obtained at the sacrifice of the model's transparency. It is worth mentioning that the above membership grades are different from μ_{ml} mentioned in Section 4.1.2.2, where μ_{ml} is calculated using Eq. 4.8 on the whole data space which does not distinguish inputs and outputs.

The lower FRBS shown in Figure 4.16 represents the 'projection' based fuzzy modelling approach. The terms, such as 'around 2' (which is in essence a fuzzy set centred on 2), are found through the projections of the clusters onto each dimension. Since fuzzy sets on each dimension are available, the projection based method conveys more semantic meanings than the rapid prototyping method. However, due to the loss of information during the projection process, the projection based fuzzy modelling approach normally results in a less accurate initial FRBS.

Since the aim in this research work is to elicit a transparent knowledge base without too much compromise on the model's accuracy, the projection based fuzzy modelling approach seems more suitable. The problem of having a less accurate initial fuzzy model can somehow be compensated via a subsequent fine-tuning procedure. Soft clustering seems indispensable only when one decides to use the rapid prototyping method. In such a case, clusters' membership functions (Eq. 4.8) play a vital role in forming the FRBS. If the projection based fuzzy modelling approach is selected, the need of the information provided by fuzzy clustering is relaxed since those fine details attached to clusters' membership functions will inevitably be lost during the projection procedure. Hence, G3Kmeans is a suitable clustering algorithm for the modelling purpose of this research. The only issue which remains to be solved is to find a way so that the radiuses (spreads) of the identified clusters can be estimated via the already known cluster centres.

4.4.2 Elicitation of Initial Singleton FRBSs Using G3Kmeans

In Section 2.6.2, two different types of FRBS, namely TSK and Mamdani FRBS, were introduced. Here, the general form of FRBS is revisited for ease of understanding:

$$R_i\!: If\ x_1\ is\ A_i^1\ and\ x_2\ is\ A_i^2, \dots\ , and\ x_j\ is\ A_i^n\ Then\ y_i = Z_i$$

where, A_i^j is the ith linguistic value (fuzzy set) for the jth linguistic variable x_j defined over the universe of discourse $\mho_j$; the function $\mu_{A_i^j}(x_j)$ associated with A_i^j that maps $\mho_j$ to [0, 1] is the corresponding membership function; R_i represents the ith rule in the rule base, and y_i is the output of the ith rule. Typically, Z_i can be the function of the inputs or the linguistic value of the output, which differentiate FRBS into TSK (the former) and Mamdani (the latter) FRBS. In order to build such a rule-base via the proposed G3Kmeans algorithm, one has to establish a certain mechanism so that $\mu_{A_i^j}(x_j)$ and the corresponding output Z_i can be linked with the extracted clusters. In the following, such a mechanism is explained using a Singleton FRBS as an example. For Mamdani FRBS, the process is almost the same except some minor modifications in the output and the inference method, which will be discussed in detail in Sections 5.3.1 and 5.4.2.

First, it is assumed that the Gaussian membership function is used for the inputs of FRBS. In such a case, the ith identified cluster centre C_X^i in the input space corresponds directly to the centriods of the Gaussian membership functions responsible for the ith rule. The spreads of the corresponding Gaussian membership functions are obtained by first calculating the U matrix as follows:

$$U(i,m) = \left(\sum_{l=1}^{k} \frac{\|X_m - C_i\|}{\|X_m - C_l\|}\right)^{-1} \tag{4.9}$$

where, $C_1, C_2, \dots, C_k$ are k cluster centres, and $U(i,m)$ specifies the degree of data point m belonging to the ith cluster. Spread σ_i^j is thus deduced as follows:

$$
\begin{aligned}
&If: Guassian\ membership\ function\ is\ used\\
&then:\ exp\left(-\frac{1}{2}\cdot\left(\frac{x_m^j-c_i^j}{\sigma_{im}^j}\right)^2\right)=U(i,m)\\
&\Rightarrow \sigma_{im}^j=\sqrt{\frac{-(x_m^j-c_i^j)^2}{2\cdot\log\left(U(i,m)\right)}} \qquad m=1,\dots,N \qquad (4.10)\\
&\Rightarrow \sigma_i^j=\rho\cdot\max_{\mathrm{m}\in[1,\mathrm{N}]}(\sigma_{im}^j)
\end{aligned}
$$

where, j indicates the dimension of the spread in the input space for the *ith* cluster, N is the total number of data points. The maximum value of σ_{im}^j is picked to ensure a certain degree of overlap between different clusters. This also ensures a smooth transition of the predictions over different regions. ρ is used to adjust the degree of overlap. In practice, values between 0.85 and 1 are good choices. In the following experiments, ρ is set to 0.95 without any loss of generality.

Hence, the Gaussian membership function on each dimension can be specified using Eq. 4.11. It is worth mentioning that the Gaussian membership function defined in Eq. 4.11 is the projection of the overall Gaussian cluster on the jth dimension as follows:

$$\mu_{A_i^j}(x_m^j)=\exp\left(-\frac{1}{2}\cdot\left(\frac{x_m^j-c_i^j}{\sigma_i^j}\right)^2\right) \qquad (4.11)$$

Hence, the overall Gaussian cluster can be defined as the product of the membership functions on each dimension as shown in Eq. 4.12.

$$\mu_i(X_m)=\mu_{A_i^1}(x_m^1)\cdot\mu_{A_i^2}(x_m^2)\cdot\dots\cdot\mu_{A_i^n}(x_m^n)=\prod_{j=1}^{n}\exp\left(-\frac{1}{2}\cdot\left(\frac{x_m^j-c_i^j}{\sigma_i^j}\right)^2\right) \qquad (4.12)$$

Figure 4.17 illustrates a 2-dimensional overall Gaussian cluster and its corresponding projections on each dimension.

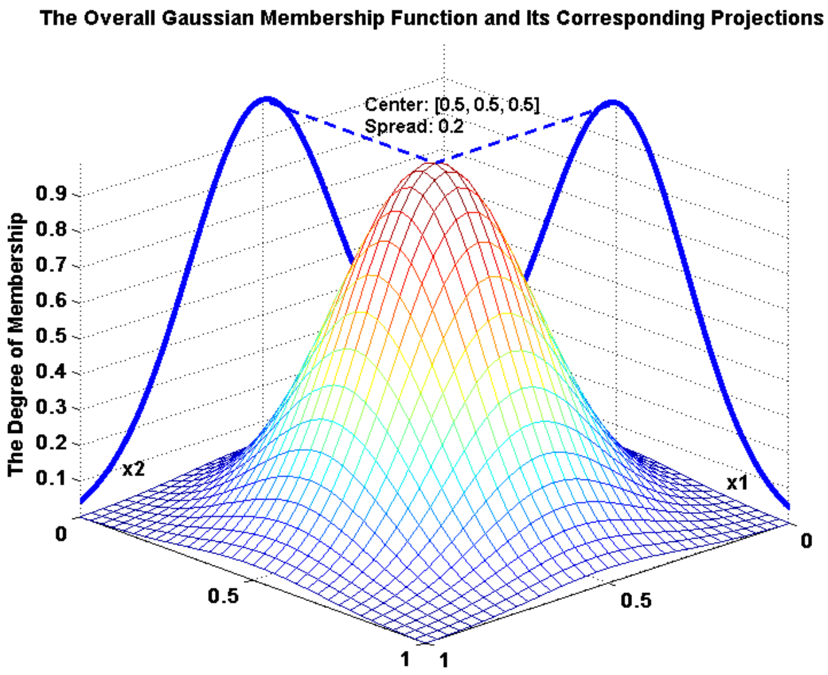


Figure 4.17 A 2-dimensional overall Gaussian membership function and its projected membership functions on x1 and x2 dimensions.

The output of each rule Z_i is equal to c_i^y. Substituting Eq. 4.11 and c_i^y into the general form of FRBS mentioned earlier leads to an initial Singleton FRBS. Next, if Centriod of Area (COA) defuzzification method is employed, the crisp output of the initial FRBS can be computed as follows:

$$y^{crisp} = \frac{\sum_{i=1}^{k} Z_i \cdot \mu_i(X)}{\sum_{i=1}^{k} \mu_i(X)} \overset{\text{def}}{=} y^{crisp}(X|\theta) \qquad (4.13)$$

$\theta = \left(c_i^y, c_i^j, \sigma_i^j | i = 1, .., k;, j = 1, .., n\right)$ is the parameter vector in which each individual parameter is linked directly to the cluster centres and spreads. This vector is subject to further tuning in Chapter 5 so that the predictive performance of the initial fuzzy model can be improved.

It is worth mentioning that the number k, i.e. the number of clusters, is directly related to the number of fuzzy rules. However, in this project, no explicit approach has been devised to detect this number during the clustering process. Instead, an overestimated number of clusters are initially assumed. As pointed by Setnes (2000), an overestimated number of clusters may increase the possibility that all important regions in the data are covered, and the result becomes less dependent on the initialisation. Such an idea of using an overestimated number of clusters has been extended to that of fuzzy rules in this project. Hence, a FRBS with the overestimated number of rules is obtained after clustering.

Unlike Setnes who utilised the orthogonal least squares (OLS) method (Wang *et al.*, 1992; Chen *et al.*, 1991) to remove less important clusters during the clustering process, these overestimated number of clusters were not directly dealt with in any part of the clustering procedure. Instead, a rule base with an overestimated number of rules is believed to be able to cover every vital aspect in the search space just as what has been assumed for the overestimated number of clusters. Such a rule-base may over-fit the training data due to the unnecessary complex structure. However, just as what has been done to remove the redundant clusters a more compact FRBS with a good generalisation ability can be obtained via pruning and merging operations. Such operations are discussed in detail in Section 5.5.4. In the next Section, a benchmark example is used to illustrate the fuzzy rule base extraction process. Modelling results based G3Kmeans are compared with those based on FCM, Subtractive Clustering and *K*-means algorithms.

4.4.3 An Example of Application

The benchmark example used in this Section is a nonlinear static system with two inputs and one output, which has been studied by Sugeno *et al.* (1993). The system is defined as follows:

$$y = (1 + x_1^{-2} + x_2^{-1.5})^2, \qquad 1 \leq x_1, x_2 \leq 5 \tag{4.14}$$

In order to make a fair quantitative comparison with the results reported in Delgado *et al.* (1997), the same 50 input-output data pairs are used. The maximum allowable number of clusters is set to 5 for G3Kmeans, FCM, Subtractive Clustering and *K*-means, which is the same number as what has been set in Delgado's work.

Hence, after clustering, a set of 5-rule FRBSs are elicited via different clustering algorithms. For this problem, G3Kmeans takes 10 iterations to terminate. However, as one can see from Figure 4.18, after 4 iterations, the algorithm had already converged!

Figure 4.19 shows the three-dimensional I/O graph of the nonlinear system along with the data points that have been classified into five clusters using G3Kmeans. As shown in Figure 4.19, G3Kmeans can automatically locate the regions which, after identifying the cluster centres, capture the main features of data.

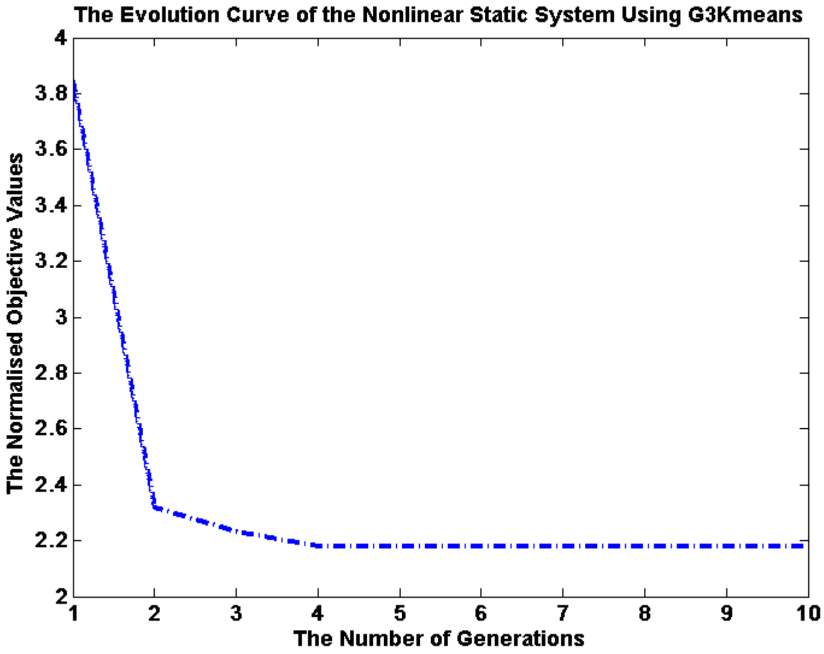


Figure 4.18 The evolution curve of the nonlinear static system using G3Kmeans.

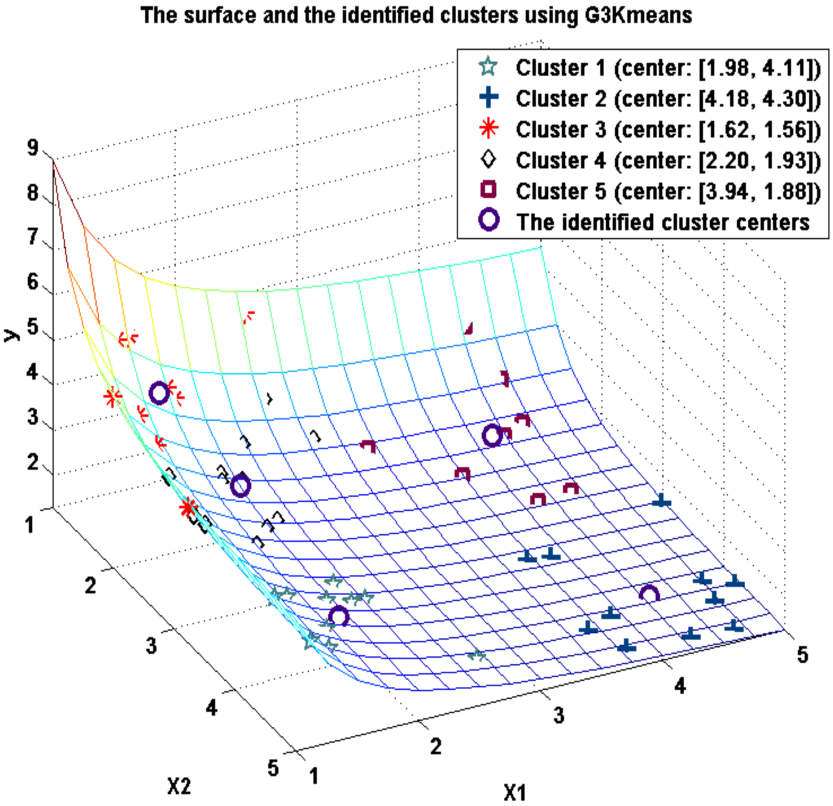


Figure 4.19 The surface of the nonlinear static systems and the identified clusters.

A close inspection of the identified clusters shown in Figure 4.19 reveals that 5 clusters may represent an overestimated number as one may possibly bring this number down to 4 clusters by merging cluster 3 and 4 without too much damage to the model's accuracy; such results will be shown in Section 5.5.6, where similar rules (clusters) are fused. Converting these identified clusters into fuzzy rules is straightforward via Eqs. 4.11~4.13. Figure 4.20 illustrates the overall Gaussian membership functions whose projections on each input dimension form the rules shown in Figure 4.21.

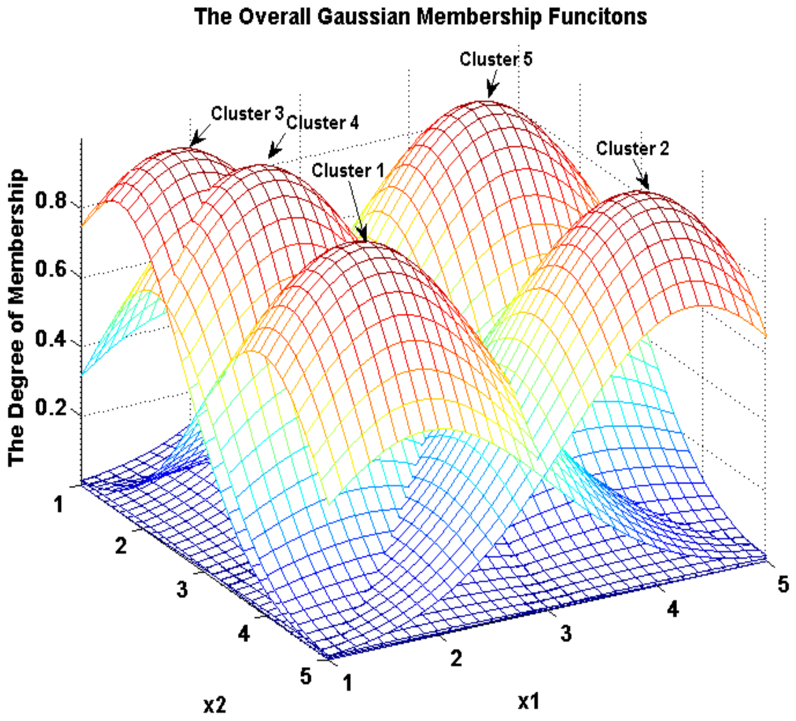


Figure 4.20 The overall Gaussian membership functions and their corresponding clusters.

Figure 4.21 shows individual rules of the converted FRBS and the projected membership functions on each input dimension. The resulting fuzzy rule-base is interpretable to human experts since each fuzzy set can be related with a linguistic value. As a matter of fact, each rule shown in Figure 4.21 corresponds to a cluster and an overall Gaussian membership function shown in Figures 4.19 and 4.20. For example, rule 1 corresponds to cluster 5. As one can see from Figure 4.21 (b), some fuzzy sets are heavily overlapped, which leads to the difficulty in a semantic interpretation. This issue is further discussed in Chapter 5.

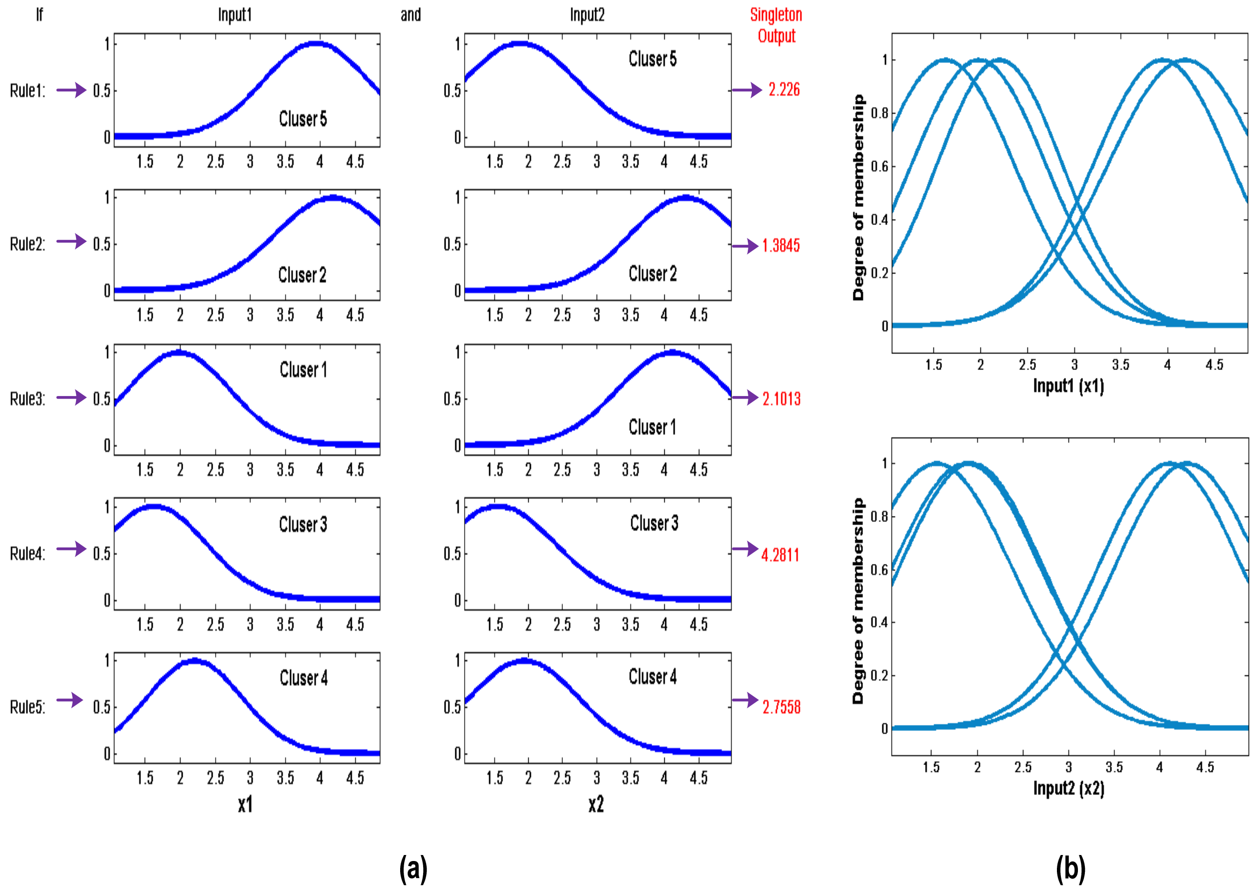


Figure 4.21 (a) individual rules in the fuzzy rule base; (b) projected membership functions on each input dimension.

In order to conduct a quantitative comparison on the clustering performance, Table 4.5 summarises the objective values using different clustering algorithms. To compare the performances of the obtained initial FRBSs using different configurations, the root mean square error (RMSE) is utilised to measure the degree of the discrepancy between the actual outputs and the predicted outputs, which is defined in Eq. 4.15. All results shown in Table 4.5 are the average values over 20 independent runs.

$$RMSE = \sqrt{\frac{\sum_{m=1}^{N}(y^{crisp}(X_m|\theta) - y_m)^2}{N}} \tag{4.15}$$

TABLE 4.5
COMPARISONS OF THE OBJECTIVE VALUES AND THE PERFORMANCES BETWEEN DIFFERENT FUZZY MODELING METHODS BASED ON DIFFERENT CLUSTERING ALGORITHMS ON A NONLINEAR STATIC SYSTEM WITH FIVE RULES

Modeling Methods[1]	**Objective Values**				**Initial FRBS predictive Performance**		**Time (second)**
	Min.	Max.	Mean	Std.	RMSE	Std.	
FCM	2.2659	2.2667	2.2663	3.15e-04	0.6290	0.0049	0.0080
K-means	2.1795	2.9406	2.4734	0.3009	0.6236	0.0070	0.0066
Subtractive Clustering	2.4413	2.4413	2.4413	**0**	0.6204	**0**	0.0160
G3Kmeans	**2.1795**	**2.1795**	**2.1795**	**0**	**0.5954**	**0**	0.5460
EST5[2]	-	-	-	-	0.672	-	-

[1] Fuzzy modeling methods based different clustering algorithms.
[2] Fuzzy modeling methods proposed by Delgado *et al.* (1997).

The same conclusions in terms of the clustering performance, as those in Section 4.3.3, can be drawn from the above results. G3Kmeans has proved to be robust and not sensitive to the initial settings. It was successful in finding the global optima in the sense that a within-cluster-distance metric ϖ (refer to Eq. 4.7) is globally minimised. The hence elicited initial FRBS based on these compact clusters leads to the best predictive performance when compared to the previously mentioned methods. *K*-means and FCM are both sensitive to the initialisations, which may partially be responsible for the less accurate elicited FRBSs. Subtractive Clustering is robust subject to different initialisations. However, the clustering results produced by Subtractive Clustering are only sub-optimal. This is confirmed by its less accurate initially elicited FRBS. EST5 represents the most inaccurate implementation. EST5 uses the approximations of the extensional hulls of the clusters to form membership functions. Such approximation and projection processes greatly affect the performance of the elicited FRBS.

4.5 Summary

In this chapter, an evolutionary based clustering algorithm, namely G3Kmeans, is introduced. The proposed algorithm is tested extensively through the artificial and real data sets. The results show that the proposed algorithm is superior to other more traditional clustering algorithms in that:

1) It is robust to different initial settings;

2) It can approach to the global optimal partitions very closely, especially for high-dimensional problems;
3) It is computationally more efficient compared to other evolutionary based clustering algorithms.

G3Kmeans is also suitable for eliciting FRBS without any prior assumption about the underlying data structure. The performance of the elicited FRBS using G3Kmeans is superior to the performances of those elicited via *K*-means, FCM, Subtractive Clustering and EST5. In the next chapter, an extension of the G3Kmeans algorithm by combining it with the proposed PAIA algorithm described in Chapter 3 for multi-objective fuzzy modelling will be introduced.

Chapter 5

An Immune Inspired Multi-Objective Fuzzy Modelling (IMOFM)

"From computing with numbers to computing with words - from manipulation of measurements to manipulation of perceptions."

Lotfi A. Zadeh, Int. J. Appl. Math. Comput. Sci., 2002

In this chapter, an immune inspired multi-objective fuzzy modeling (IMOFM) mechanism is proposed. IMOFM adopts a multi-stage modeling procedure and a variable length coding scheme to account for the enlarged search space due to the simultaneous optimisation of the rule-base structure and its associated parameters. IMOFM tries to challenge Zadeh's Principle of Incompatibility, which may facilitate the ultimate goal of 'computing with words'. In this chapter, it is shown how to elicit an accurate and yet transparent FRBS from quantitative data.

5.1 Introduction

5.1.1 Data-Driven Modelling (DDM)

Traditionally, modelling tasks involve the building of mathematical equations which can best describe the underlying process. Such a modelling practice normally requires a deep understanding of the systems under investigation, hence the reason why it is often referred to as *knowledge-driven modelling*. On the contrary, Data-Driven modelling (DDM), inspired principally from artificial intelligence techniques, is based on limited knowledge of the modelling process and relies on the data describing the input and output mapping. DDM is able to make abstractions and generalisations of the process and plays often a complementary role to knowledge-based models. A simple example of DDM is the linear regression in which the coefficients of the regression equation are 'trained' through the available data. Figure 5.1 illustrates a one-input-and-one-output system which can be approximated via a 'straight line'. The coefficients of the line equation are trained so that the line is best fitted into the data points in the sense of the least-squares error or other forms of error measures.

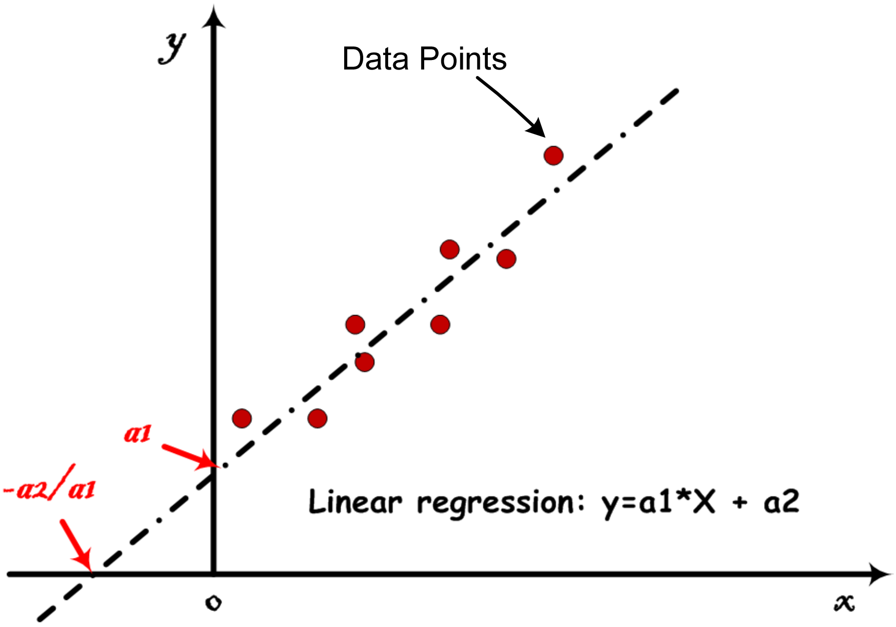


Figure 5.1 The linear regression.

For complex systems, the linear regression may not be sufficient, which leads to the need for the non-linear regression techniques. Among many of these techniques, ANN, fuzzy rule-based systems and Neural-Fuzzy Systems (NFS) have been receiving more attention during the last two decades due to the facts of not only being able to approximate practically any given function to an arbitrary accuracy (Kosko, 1994; Wang *et al.*, 1992), but also being able to generalise reasonably well to any previously 'unseen' situations. The prevalence of these nonlinear regression techniques is largely attributed to the breakthrough in the nonlinear optimisation techniques, such as the BEP and the EC. In the following space, all these issues are covered since they all make their appearances in the development of the proposed modelling framework.

5.1.2 Relationship between FRBS and DDM

Since the first introduction of 'fuzzy logic', FRBS have been widely used in control engineering (Passino *et al.*, 1998). However, the predominant approach in the traditional design of fuzzy rule-based systems highly relies on human experts, which makes the fuzzy modelling process similar to the design of expert systems except traditional expert systems were based on the classical Boolean logic and thus were not well suited to managing the progressiveness in the underlying process phenomena (Guillaume, 2001). Both FRBS and expert systems share some common features:

- They all include the so-called 'knowledge base' which uses some knowledge representation formalism to capture the domain expert knowledge;
- They all acquire a process of inducing knowledge from the expert or other resources and codifying such knowledge according to the formalism.
- They may or may not have learning components. Once the model is developed it will replace human experts in the same real world problem solving situation so that it can aid human workers to make decisions or control.

If knowledge is induced from resources such as data rather than from experts it is in essence a data-driven methodology. Chapter 4 described one such method in which clustering is used to automatically induce hidden (implicit) knowledge from data. If learning components are further incorporated into the procedure of coarse knowledge inducement, the accuracy of the raw knowledge base can be improved to a certain degree depending on the quality of the historic data and the power of the learning mechanism.

Although learning components can improve the quality of the model it may suffer from two serious problems, e.g. the deterioration of the model's interpretability and the over-fitting to the training patterns. These problems are formally discussed in Section 5.1.4 following the brief introduction of the concept concerning model accuracy and interpretability. Among many existing solutions to overcome the aforementioned problems, evolutionary based approaches are reviewed in Section 5.1.5. From Section 5.2 onwards, IMOFM is introduced, which represents an alternative tactic to solve the above problems.

5.1.3 Accuracy vs. Interpretability

As Casillas *et al.* (2001) pointed out, modelling is the task that simplifies a real system or complex reality with the aim of easing its understanding. Hence, the development of *reliable* and *comprehensible* models must be the main theme of any modelling tasks. By 'reliable' it is meant the model's capability of faithfully representing the real system, in other words 'the model accuracy'. By 'comprehensible' it is meant the model's capability of expressing the behaviour of the real systems in a comprehensible way, in other words 'the model interpretability'. However, as Zadeh conjectured in his Principle of Incompatibility (Zadeh, 1973) cited as below,

> *"As the complexity of a system increases, our ability to make precise and yet significant statements about its behaviour diminishes until a threshold is reached beyond which precision and significance (or relevance) become almost mutually exclusive characteristics."*

it is very likely that accuracy and interpretability may well be exclusive requirements in a modelling process. Since both requirements are vital and cannot always be possessed at the same time, a good balance between them is the best outcome that one can achieve. The reflection of these in a fuzzy modelling scenario represents a dilemma of designing FRBS. This issue is further discussed in Section 5.1.4. As far as interpretability is concerned, it is mainly a subjective property and normally refers to at least one or all of the following aspects in a fuzzy modelling scenario:

- The distribution of the fuzzy sets on each dimension should be well separated so that meaningful (distinguishable) linguistic terms can be associated with them.
- The number of fuzzy sets for each dimension and the number of rules should not be excessive. This is closely related to the cognitive studies, one of which reported by Miller (1956), which shows that the optimal number of chunks of information simultaneously held in human short-term memory should be seven, plus or minus two. This implies that redundant rules and fuzzy sets should be merged or deleted.
- The number of input variables involved in each rule should be optimal, which means input variables are subject to either a global selection, in which case none of the rules in the rule base can use the deleted input variables, or a local selection, in which case the selection is done at the individual rule level (Guillaume, 2001).
- The rule base should be complete and consistent (Guillaume, 2001; Jin *et al.*, 1999).

Otherwise, the knowledge represented by the rule base is incomplete, and different conclusions given similar premises would certainly confuse its users.

The 'accuracy vs. interpretability' issue can also be formulated as a multi-objective optimisation problem. Figure 5.2 shows the Pareto front in a bi-objective fuzzy modelling scenario where two competing objectives, viz. the predictive error (accuracy) and the rule-base complexity (interpretability), are minimized simultaneously. The aim is to find a set of 'approximate Pareto FRBSs' as close to the true Pareto front as possible.

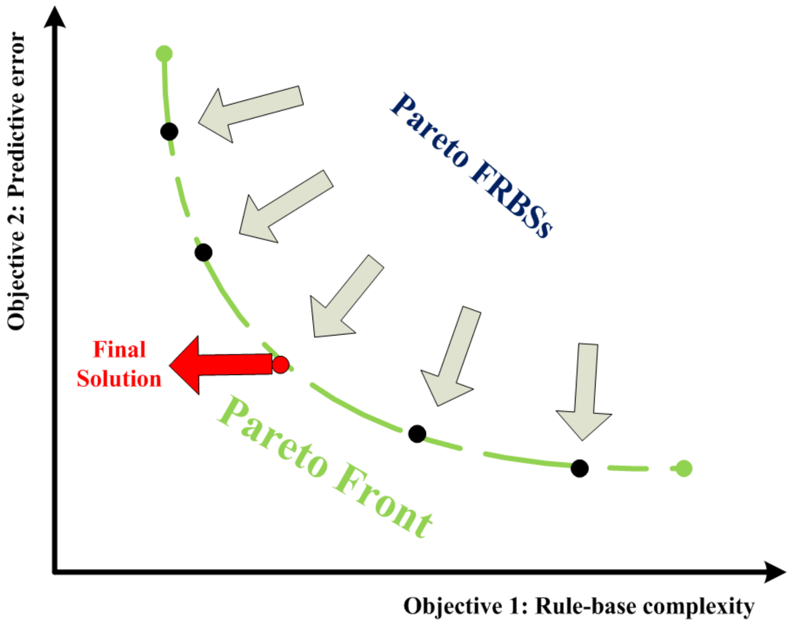

Objective 2: Predictive error
Pareto FRBSs
Final Solution
Pareto Front
Objective 1: Rule-base complexity

Figure 5.2 Pareto front in a bi-objective fuzzy modelling case.

By finding a set of solutions, human can understand the underlying problem in a much greater depth, and finally a single optimal solution to a specific scenario is finally selected and applied. In the above case, if one requires certain interpretability (transparency) of the FRBS along with its good predictive accuracy the middle circle could be the one that fulfils the user's need. As already stated by Jiménez *et al.* (2001), this should result in a 'minimal' human intervention during the modelling process.

5.1.4 The Dilemma of Building FRBS

The main advantage of using FRBS as a modelling tool over other modelling methods lies in its additional ability of integrating human expertise in the form of vague or imprecise statements rather than crisp mathematics, for many real-world systems' knowledge can only be described by experts with nature language. Depending on what degree to which such expertise is involved, fuzzy modelling may pertain to 'white' box, 'black' box or 'grey' box modelling. Previous research on fuzzy modelling was mainly concerned with the way to synthesis a rule-base with domain-dependent knowledge from human experts, such as operators, and hence render the task of tuning the parameters associated with the antecedent and consequent parts as an optimisation problem, e.g. recursive least-squares or gradient-based methods. Without the tuning process following the synthesis step, the above approach is indeed equivalent to 'white'-box modelling and the elicited model can be regarded as descriptive (linguistic) FRBS (Cordon *et al.*, 2001), which may give rise to the following four limitations:

- ✧ Often, expert knowledge is not available or is limited;
- ✧ It is very hard to handle problems with a significant amount of data to be processed and analysed;
- ✧ The synthesis approach suffers from the 'curse' of dimensionality;
- ✧ The way to design such a fuzzy system is not domain-independent and thus no systematic (or unique) design procedure can be followed.

In all these cases, a knowledge extraction emanating purely from experts fails to provide a satisfactory solution. However, discovering knowledge from data can help in overcoming the aforementioned limitations by augmenting FRBS with an additional learning layer.

In the past two decades, many successes in the hybridisation of FRBS and learning methods have been registered. The most representative of these must be the so-called neuro-fuzzy system, which incorporates learning methods normally used in neural networks for FRBS (Jang, 1993). Almost at the same time, attempts of hybridising clustering methods with fuzzy systems were carried out and led to very promising results (refer to Section 4.4.1). The aim of these types of hybridisation techniques is to automatically elicit rules from large collections of learning data. Despite the great success using the aforementioned paradigms, the following challenges have also been identified:

1. The designer still needs to set the abstraction level or the number of clusters;
2. The need to set the starting points for clustering and neural networks;
3. Most importantly, the elicited FRBS can only be described as approximate FRBS (Cordon *et al.*, 2001) rather than being labelled as the descriptive one.

The main drawback of approximate FRBS compared to the descriptive one is its degradation in terms of interpretability of the rule-base due to the automatic learning process, which yields overlapped fuzzy sets. Although such approximate FRBS retains some basic level of interpretability, it may become more 'black'-box oriented although often its performance is much improved compared to the descriptive FRBS.

The shift between the descriptive and approximate FRBS represents a dilemma for designing of fuzzy models. The last two decades have witnessed the popularity of the latter by compromising 'interpretability' (significance) with 'accuracy' (precision), which deviates from the original intention of FRBS which must always try to challenge Zadeh's Principle of Incompatibility. Taking this into account, one can find that EAs, in particular GAs, have a long history of being incorporated into fuzzy logic and demonstrate a possible route to the remedy for the dilemma. This may ultimately facilitate the achievement of 'grey'-box modelling. The next Section reviews the existing EAs-based approaches for tackling the above mentioned dilemma.

5.1.5 Literature Review of Previous Works

Originated from Karr's work (Karr, 1991), the GA approach in fuzzy systems was initially utilised to adjust the parameters of membership functions, which leads to no significant difference when compared to other learning paradigms. The real significance of employing EAs for optimising FRBSs comes from EAs' flexibility in terms of being able to encode and evolve almost every component of the FRBS (Herrera, 2008). Such a flexibility offers a solution so that one can take into account the interpretability (structure) and the performance of the FRBS in a more coherent way. Broadly speaking, there currently exist two different EA-based streams to tackle the interpretability issues: the first stream is mainly concerned with the linguistic modelling, in which a set of pre-specified fuzzy partitions are given *a priori* by experts or users (grid partition); the task is then to find an optimal FRBS in terms of its compactness and performance (Ishibuchi *et al.*, 1995; Ishibuchi *et al.*, 1997; Ishibuchi *et al.*, 2001; Ishibuchi *et al.*, 2004; Alcalá *et al.*, 2007; Cococcioni *et al.*, 2007); the second stream generally uses the approximate fuzzy model as the starting point; hence, the task is to improve the model's explanatory ability, which may have been lost during the automatic learning process, through a set of similarity-driven simplification and parameter adjusting operations (Setnes *et al.*, 1998; Setnes *et al.*, 2000; Roubos *et al.*, 2001; Jiménez *et al.*, 2001;

Jiménez *et al.*, 2002; Jin *et al.*, 1999; Jin *et al.*, 2000; Wang *et al.*, 2005; Gonzάlez *et al.*, 2007, Chen *et al.*, 2004).

In the first stream, the earliest noticeable attempt was made by Ishibuchi *et al.* (1995), in which a fuzzy classifier is built using the pre-specified linguistic terms (fuzzy sets). These linguistic terms are fixed during the course of the evolution so that their physical meanings are retained. Only the fuzzy rules are subject to the selection via GA so that a compact rule-base can be evolved from a large number of candidate rules, which should lead to a more interpretable FRBS. Since the selection process removes irrelevant and inconsistent rules, the accuracy is also improved. In the works of Ishibushi *et al.* (1997), Ishibushi *et al.* (2001) and Ishibushi *et al.* (2004), extensions to the above 'rule selection' idea were made in both single objective and multi-objective configurations. It is worth mentioning that, in Ishibushi *et al.*'s work (2004), the GA is not only used to select the optimal combination of rules but also to learn the granularity of different fuzzy partitions for each input, which leads to a more accurate fuzzy model while the linguistic feature is not compromised. Further relevant researches include those which were proposed by Alcalά *et al.* (2007) and Cococcioni *et al.* (2007). In Alcalά *et al.*'s work (2007), apart from the rule selection, the authors also tuned the linguistic terms by a modified GA. However, such tuning is only operated in a local sense in order to maintain their original semantics. One interesting paper in the second stream is attributed to Setnes *et al.* (1998), in which the TSK model is elicited via a fuzzy clustering algorithm for its premises and a parameter estimation method for its consequents. A similarity measure is taken so that similar fuzzy sets can be merged. Consequently, similar rules are merged as well. Hence, the distinguishability of membership functions and the compactness of the rule-base are improved. Although this rule-base simplification method does not relate to the EA directly, it has since inspired many EA-based fuzzy modeling algorithms within this trend (Setnes *et al.*, 2000; Roubos *et al.*, 2001; Jiménez *et al.*, 2001; Jiménez *et al.*, 2002; Jin *et al.*, 1999; Jin *et al.*, 2000; Wang *et al.*, 2005). In Gonzάlez *et al.*'s work (2007), the idea of rule pruning is used to delete less relevant rules within a multi-objective optimisation framework. The similarity measure is not explicitly used in this work.

Comparing the two streams leads to the following: in the linguistic modelling stream, the target problems are normally associated with classifications and low-dimensional function approximations; hence, the effect of the 'curse of dimensionality' due to the grid partition and the need for the parameter tuning due to the performance requirement are not serious issues. In the latter case, high-dimensional approximations are often the case; as a result, an approximate FRBS is a better choice to start with due to the accuracy and compactness requirements. Within the second stream, EA-based multi-objective fuzzy modelling has become a recent hotspot for function approximations due to its ability of producing a set of compromised FRBSs (Jiménez *et al.*, 2001; Jiménez *et al.*, 2002) and (Wang *et al.*, 2005; González *et al.*, 2007). However, this is a rather new developing area with several other issues to be addressed. Among which, it is believed that the following considerations are the most important:

- ✧ most well-known multi-objective optimisation algorithms used in fuzzy modeling, e.g. NSGA II (Deb, 2001), are originally designed to solve real-valued problems; in order to use such type of algorithms to simultaneously optimise the rule-base structure and the membership function parameters, similarity-driven simplifications are normally selected as the mutation operators for the former (Jiménez *et al.*, 2001; Jiménez *et al.*, 2002; Wang *et al.*, 2005), and the heuristic variations (crossover) are proposed for the latter (Jiménez *et al.*, 2001; Jiménez *et al.*, 2002; Wang *et al.*, 2005; González *et al.*, 2007); however, the search power of these optimisation algorithms relies heavily on their original variation (search) operators; other components of the algorithms are mainly used to advocate diversity and elitism; without using the original variation operators, even if the general framework is kept fixed it is likely that the search capability, in terms of the real-valued optimisation part, may be compromised, and this is the partial reason to explain the necessity to include a gradient-based optimisation for the enhancement of the parameter optimisation in González *et al.*'s work (2007);
- ✧ The reason behind the use of the heuristic variation operators for the parameter optimisation is that the structure optimisation leads to individuals with different sizes, e.g. rule base length, which makes the conventional variation operators invalid. Hence, new techniques that can cope with the variable length coding and can facilitate the use of the original variation operators are needed.

With the aim of solving high-dimensional approximation problems, the proposed modelling framework-IMOFM falls into the second stream. To address the above two issues, the research work in Chen & Mahfouf's works (2006, 2008a) (refer to Chapter 3) is extended, which has been shown to be effective for real-valued multi-objective optimisation, to a fuzzy modeling scenario. A new distance index (Chen & Mahfouf, 2008b; 2009) that is able to cope with the variable-length individuals and unconstraint optimisation is also proposed. The main focus points are two types of FRBS, viz. Singleton FRBS and Mamdani FRBS, due to their simplicity and their ability to express semantics in both premises and consequents. In the next section, IMOFM is introduced, which is in essence a three-stage modelling procedure which mimics the proposed multi-stage immune optimisation procedure already discussed in Section 3.5.

5.2 The Framework of the Proposed Modelling Method

IMOFM is a systematic multi-objective fuzzy modelling framework, which can be regarded as a three-stage modelling procedure. The first two stages are equivalent to the vaccination process in the first stage of the immune optimisation procedure (see Section 3.5). By doing so, an initial 'vaccine model' (prior knowledge, in some sense) can efficiently be elicited. Another reason of including the first two modelling stages, especially the second one, is that by doing so the most complex-rule base can survive under the pressure of 'Pareto' selection. Without including the refining step (the second stage), the rule-base with a complex structure may be regarded inferior to the less complex-rule base in a 'Pareto' sense. Even if both the most complex and less complex rule-bases are inaccurate in the early evolutionary stages, the 'Pareto' selection favours the one with a simpler structure. Hence, one may lose the chance of evolving the most accurate FRBS, which normally comes with a complex structure (refer to Section 5.6.1). The 'vaccine model' is then used in the third stage to seed the initial population of PAIA2 in order to obtain a set of Pareto fuzzy models with improved interpretability.

To tackle the problem of simultaneously optimising the rule-base structure and parameters, a variable length coding scheme is adopted, and a new distance index is proposed to cope with the variable-length individuals, which should improve the efficiency of the search (see Section 5.5.3 for more details).

Figure 5.3 represents a schematic diagram of such a framework and each stage depicted in this figure is expanded in depth in the following sections.

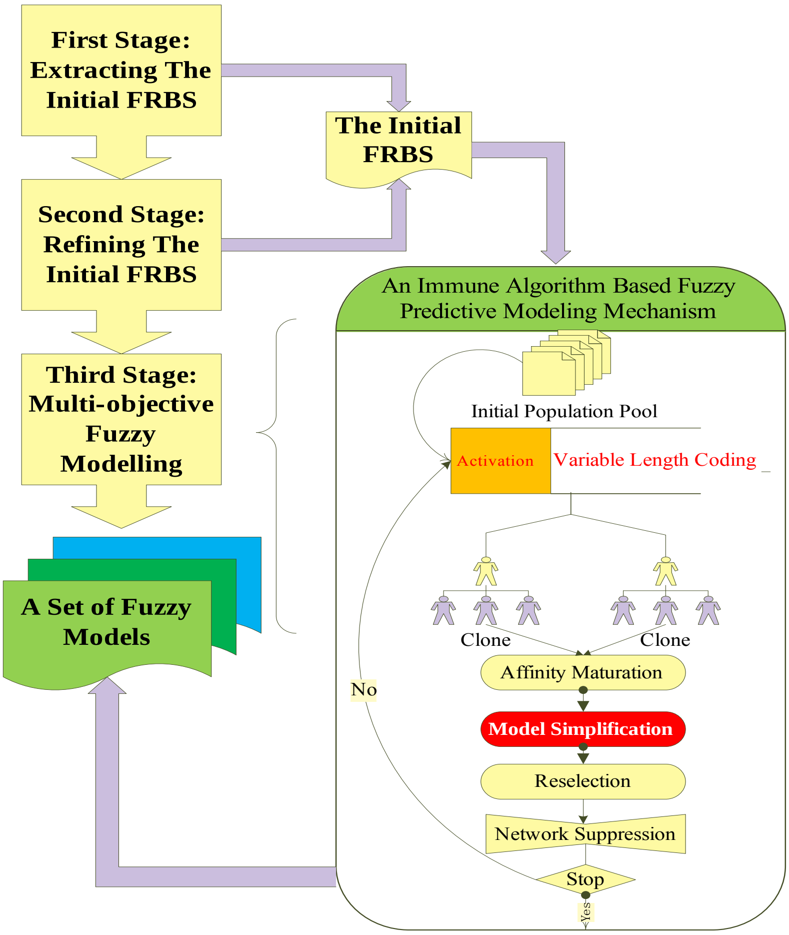

First Stage: Extracting The Initial FRBS
The Initial FRBS
Second Stage: Refining The Initial FRBS
Third Stage: Multi-objective Fuzzy Modelling
A Set of Fuzzy Models
An Immune Algorithm Based Fuzzy Predictive Modeling Mechanism
Initial Population Pool
Activation
Variable Length Coding
Clone
Clone
Affinity Maturation
Model Simplification
Reselection
Network Suppression
No
Stop
Yes

Figure 5.3 The proposed IMOFM framework.

5.3 First Stage: Elicitation of Initial FRBSs

Section 4.4.2 gives detailed steps on how to elicit an initial Singleton FRBS from data using the G3Kmeans algorithm, which serves as the first modelling stage in IMOFM_S (IMOFM_S stands for the Singleton version of IMOFM). Hence, in the following space, special attentions have been given to the Mamdani version of IMOFM, viz. IMOFM_M. IMOFM_M differs from the original Mamdani FRBS (Mamdani, 1974) in that IMOFM_M adopts a different T-norm, S-norm and defuzzification mechanism.

5.3.1 Elicitation of the Initial Mamdani FRBS

The original Mamdani FRBS is based on the so-called ‘sup-star compositional rule of inference’ (see Section 2.6.2 and Eqs. 5.1~5.3) and the overall implied fuzzy set (see Section 2.6.2 and Eq. 5.3) (Passino *et al.*, 1998, p. 63), which are defined as follows:

$$\mu_{\hat{B}_i}(y_m) = \mu_i(X_m) * \mu_{B_i}(y_m) \tag{5.1}$$

$$\mu_i(X_m) = \mu_{A_i^1}(x_m^1) \cdot \mu_{A_i^2}(x_m^2) \cdot \dots \cdot \mu_{A_i^n}(x_m^n) \tag{5.2}$$

$$\mu_{\hat{B}}(y_m) = \mu_{\hat{B}_1}(y_m) \oplus \mu_{\hat{B}_2}(y_m) \oplus \cdots \oplus \mu_{\hat{B}_i}(y_m), \ \ i = 1 \dots k \tag{5.3}$$

where, X_m and y_m are the inputs and output of the mth data point; x_m^n indicates the nth input of the mth data point; and k is the number of fuzzy rules in the rule-base. The 'sup' corresponds to the ⊕ operation, and the 'star' corresponds to *. A special instance of the 'sup-star', which uses maximum for ⊕ and minimum for *, was adopted in the original Mamdani implementation, and the centre of average defuzzification was applied on the overall implied fuzzy set in order to derive a crisp output, which leads to two problems as mentioned by Passino (1998, p. 64):

(1) The overall implied fuzzy set $\hat{B}$ is itself difficult to compute;

(2) The defuzzification techniques based on the overall implied fuzzy set are also difficult to compute.

More importantly, if an analytical solution cannot be deducted from the defuzzification step the gradient based optimisation method, such as the BEP technique, cannot be utilised. Hence, in this work, the centre of gravity defuzzfication is applied on the implied fuzzy set (Eq. 5.1). Instead of using minimum and maximum, 'product' is used for * and 'plus' is used for ⊕. Unlike traditional Mamdani FRBS which may use the same type of membership functions for premises and consequents, IMOFM_M uses Gaussian membership functions for the premises (refer to Section 4.4.2) and the bell-shape membership functions for the consequents (Eq. 5.4).

$$\mu_{B_i}(y_m) = \frac{1}{1+\left(\frac{y-c_i^y}{\sigma_i^y}\right)^2} \tag{5.4}$$

Where, c_i^y and σ_i^y are the centre and the spread of the ith membership function of the output. Hence, a Mamdani FRBS can be formulated as follows:

$$y^{crisp} = \frac{\sum_{i=1}^{k} b_i \cdot \int_y \ \mu_{\hat{B}_i}(y)\, dy}{\sum_{i=1}^{k} \int_y \ \mu_{\hat{B}_i}(y)\, dy} = \frac{\sum_{i=1}^{k} b_i \cdot \mu_i(X) \cdot \int_y \ \mu_{B_i}(y)\, dy}{\sum_{i=1}^{k} \mu_i(X) \cdot \int_y \ \mu_{B_i}(y)\, dy} \overset{\text{def}}{=} y^{crisp}(X|\theta) \tag{5.5}$$

where, b_i is the centre of area of the membership function $\mu_{B_i}(y)$ and is the peak (c_i^y) if $\mu_{B_i}(y)$ is symmetric; y^{crisp} is the final defuzzified output of the FRBS. $\theta = \left(b_i, \sigma_i^y, c_i^j, \sigma_i^j\right)$ is the parameter vector in which each individual parameter is linked directly to the identified cluster centres and spreads. This vector is subject to further fine-tuning in a bid to improve the model's predictive performance. $\int_y \mu_{\hat{B}_i}(y)\,dy$ denotes the area under $\mu_{\hat{B}_i}(y)$ over the output interval y: $[y_L, y_U]$ and $\int_y \mu_{B_i}(y)\,dy$ is calculated using Eq. 5.6.

$$\int_y \mu_{B_i}(y)\,dy = \sigma_i^y \left[arctan\left(\frac{y_U - b_i}{\sigma_i^y}\right) - arctan\left(\frac{y_L - b_i}{\sigma_i^y}\right)\right] \overset{\text{def}}{=} g\left(b_i, \sigma_i^y\right) \qquad (5.6)$$

Hence, after the first stage, a Singleton/Mamdani FRBS with the pre-specified number of rules is extracted from the numerical data, which is analytical and can be refined further using gradient based techniques, as will be introduced in Section 5.4.

5.3.2 An Example of Application

The benchmark example tested in Section 4.4.3 is employed again to demonstrate the results of the first modelling stage using IMOFM_M. The number of rules is again set to 5. Figure 5.4 shows individual rules of the initial FRBS and the membership functions on each dimension (including the output dimension).

Comparing Figure 5.4 with Figure 4.19, one can find that the premises of Mamdani FRBS and Singleton FRBS for this particular problem are the same since they are all extracted by G3Kmeans. The only difference lies in their consequents. Instead of singleton values, Mamdani FRBS uses fuzzy sets for its consequents as well, which makes Mamdani FRBS more interpretable when compared to the Singleton one. Fuzzy outputs convey vagueness information that is inherent in the model's knowledge-base and may be well designated by linguistic terms (Mencar *et al.*, 2005).

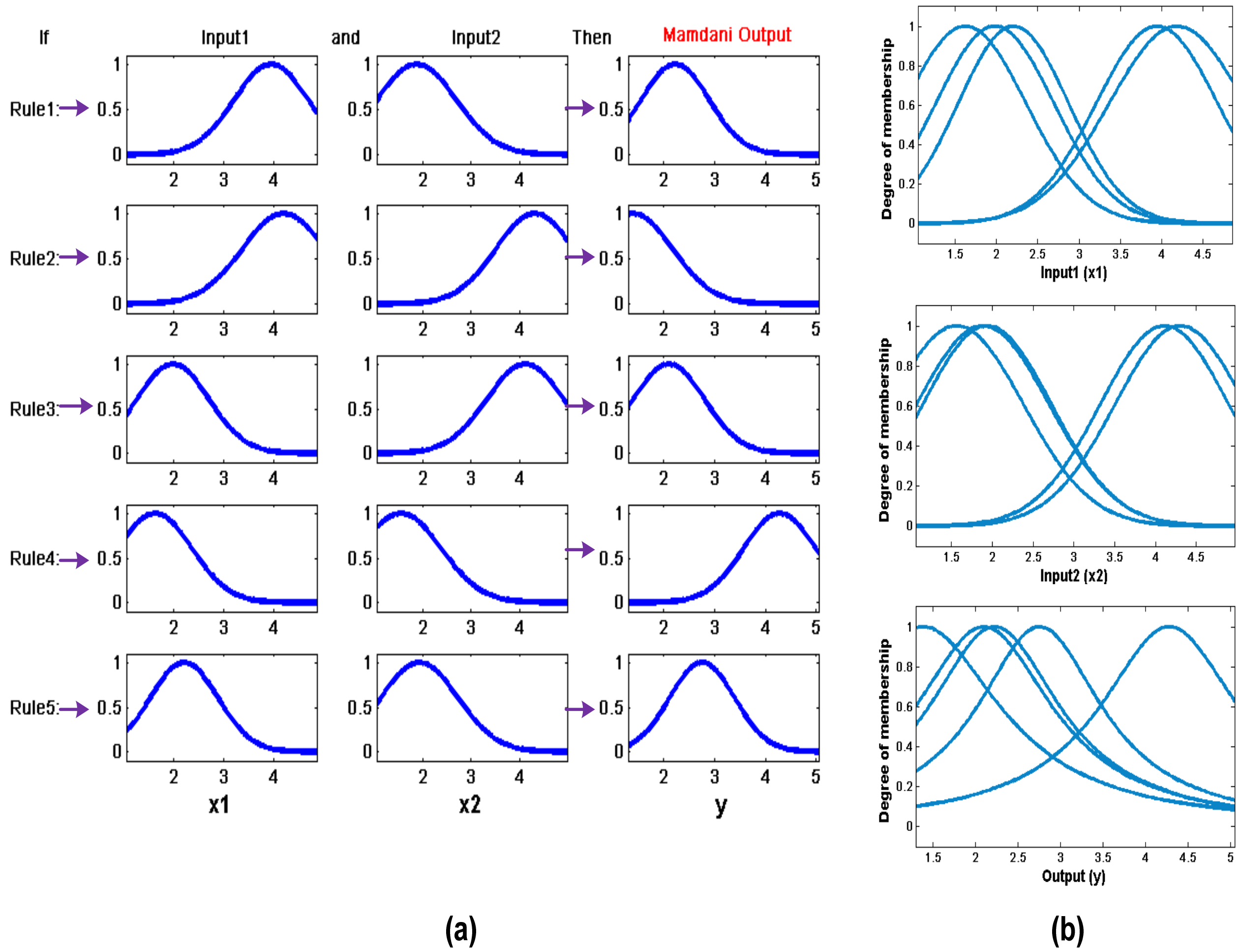


Figure 5.4 (a) individual rules in a Mamdani FRBS; (b) membership functions of each dimension.

Table 5.1 summarised the predictive performance of IMOFM_M and IMOFM_S, which are the average values of 20 independent runs. The results of IMOFM_S are adapted from Table 4.5. The detailed comparison of IMOFM_S and IMOFM_M can be found in Section 6.4.

TABLE 5.1
THE PREDICTIVE PERFORMANCES OF THE FIRST MODELING STAGE OF IMOFM_S AND IMOFM_M ON A NONLINEAR STATIC SYSTEM WITH FIVE RULES

Modeling Methods	The Predictive Performance of Initial FRBS	
	RMSE (average)	Std.
IMOFM_S	0.5954	0
IMOFM_M	0.6078	0

5.4 Second Stage: Refinement of Initial FRBSs

The initial fuzzy model extracted from the first modeling stage is not optimal from two perspectives:

(1) The structure of FRBS is not optimal as far as the interpretability is concerned. As one can see from Figures 4.21 and 5.4, the FRBS elicited from the first modeling stage contains redundant fuzzy sets and rules.

(2) The membership function parameters need to be tuned further as far as the accuracy is concerned.

A constrained BEP algorithm is thus utilised to first improve the accuracy of the initial FRBS so that a ‘vaccine model’ can be obtained for the next operation in the multi-objective optimisation stage. As mentioned by Gonz$\acute{a}$lez *et al*. (2007), if the initial population can be constructed using some heuristics, e.g. an optimised FRBS in terms of its predictive performance, then many generations of evolutionary search can be saved. The ‘vaccine model’ constructed by the first two stages acts similarly to these heuristics. In the subsequent Sections, the BEP updating formulas for IMOFM_S and IMOFM_M are given. Interested readers are referred to Passino’s book (Passino, 1998, p. 246-252) for the detailed BEP deduction for Singleton FRBS, and to Appendix A for the Mamdani FRBS.

5.4.1 Back-Error-Propagation Algorithm for Singleton FRBS

Recall Eq. 4.13 discussed in Section 4.4.2, where a Singleton FRBS is deffuzified with respect to a parameter vector $\theta = \left(b_i, c_i^j, \sigma_i^j | i = 1,..,k;, j = 1,..,n\right)$. Here, $b_i$is the output of the ith rule and equals to c_i^y in this work; c_i^j and σ_i^j are the centre and the spread of the ith membership function for the jth input. The BEP algorithm is developed such that the predictive performance of a Singleton FRBS can be improved subject to adjusting the parameters in θ. By taking the partial derivatives of Eq. 4.13 with respect to each parameter included in θ, one can obtain a set of parameter updating laws as follows, where, $\lambda_1 \sim \lambda_3$ and $\beta_1 \sim \beta_3$ are user-specific parameters and are the step seizes and the gains of momentum terms respectively (refer to Section 4.4.2 for the definitions of other parameters).

$$Singleton\ Consequent\ Updating\ Law: \quad b_i(t+1) = b_i(t) - \lambda_1 \cdot \varepsilon_m(t) \cdot \frac{\mu_{i(t)}(X_m)}{\sum_{i=1}^{k} \mu_{i(t)}(X_m)} + \beta_1 \cdot \Delta b_i(t-1) \tag{5.7}$$

$$Centre\ of\ the\ Premise\ Updating\ Law: c_i^j(t+1) = c_i^j(t) - \lambda_2 \cdot \varepsilon_m(t) \cdot \frac{\mu_{i(t)}(X_m) \cdot q(t)}{\sum_{i=1}^{k} \mu_{i(t)}(X_m)} + \beta_2 \cdot \Delta c_i^j(t-1) \tag{5.8}$$

$$Spread\ of\ the\ Premise\ Updaing\ Law: \sigma_i^j(t+1) = \sigma_i^j(t) - \lambda_3 \cdot \varepsilon_m(t) \cdot \frac{\mu_{i(t)}(X_m) \cdot r(t)}{\sum_{i=1}^{k} \mu_{i(t)}(X_m)} + \beta_3 \cdot \Delta \sigma_i^j(t-1) \tag{5.9}$$

$$where: \quad \begin{aligned} \varepsilon_m(t) &= y^{crisp}(X_m|\theta(t)) - y_m \\ q(t) &= \left(b_i(t) - y^{crisp}(X_m|\theta(t))\right) \cdot \left(\frac{x_m^j - c_i^j(t)}{(\sigma_i^j(t))^2}\right) \\ r(t) &= \left(b_i(t) - y^{crisp}(X_m|\theta(t))\right) \cdot \frac{\left(x_m^j - c_i^j(t)\right)^2}{\left(\sigma_i^j\right)^3} \end{aligned} \tag{5.10}$$

$$and \quad \begin{aligned} \Delta b_i(t-1) &= b_i(t) - b_i(t-1) \\ \Delta c_i^j(t-1) &= c_i^j(t) - c_i^j(t-1) \\ \Delta \sigma_i^j(t-1) &= \sigma_i^j(t) - \sigma_i^j(t-1) \end{aligned} \tag{5.11}$$

5.4.2 Back-Error-Propagation Algorithm for Mamdani FRBS

By using Eqs. 5.5 and 5.6 already developed in Section 5.3.1 and taking the partial derivatives of Eq. 5.5 with respect to each parameter in $\theta = \left(b_i, \sigma_i^y, c_i^j, \sigma_i^j | i = 1, .., k; , j = 1, .., n\right)$, one can end up with the following parameter updating formulas. Here, $b_i$is the output of the ith rule and equals to c_i^y (the centre of the ith output membership function) in this work; σ_i^y is the spread of the ith output membership function; c_i^j and σ_i^j are the centre and the spread of the ith membership function for the jth input; $\lambda_1 \sim \lambda_4$ and $\beta_1 \sim \beta_4$ are user-specific parameters and are the step seizes and the gains of momentum terms respectively (refer to Section 5.3.1 for the definitions of other parameters).The detailed deduction steps can be found in Appendix A.

$$\textit{Centre of the Consequents Updating Law}: \quad b_i(t+1) = b_i(t) - \lambda_1 \cdot \varepsilon_m(t) \cdot \frac{\mu_{i(t)}(X_m) \cdot \left[g\left(b_{i(t)}, \sigma^y_{i(t)}\right) + b_{i(t)} \cdot g'\left(b_{i(t)}\right) - g'\left(b_{i(t)}\right) \cdot y^{crisp}\left(X_m | \theta(t)\right) \right]}{\sum_{i=1}^{k} \mu_{i(t)}(X_m) \cdot g(b_{i(t)}, \sigma^y_{i(t)})} + \beta_1 \cdot \Delta b_i(t-1) \tag{5.12}$$

$$\textit{Spread of the Consequents Updating Law}: \quad \sigma^y_i(t+1) = \sigma^y_i(t) - \lambda_2 \cdot \varepsilon_m(t) \cdot \frac{\mu_{i(t)}(X_m) \cdot g'\left(\sigma^y_{i(t)}\right) \cdot \left[b_{i(t)} - y^{crisp}\left(X_m | \theta(t)\right) \right]}{\sum_{i=1}^{k} \mu_{i(t)}(X_m) \cdot g(b_{i(t)}, \sigma^y_{i(t)})} + \beta_2 \cdot \Delta \sigma^y_i(t-1) \tag{5.13}$$

$$\textit{Centre of the Premise Updating Law}: \quad c^j_i(t+1) = c^j_i(t) - \lambda_3 \cdot \varepsilon_m(t) \cdot \frac{g\left(b_{i(t)}, \sigma^y_{i(t)}\right) \cdot \left[b_{i(t)} - y^{crisp}\left(X_m | \theta(t)\right) \right]}{\sum_{i=1}^{k} \mu_{i(t)}(X_m) \cdot g(b_{i(t)}, \sigma^y_{i(t)})} \cdot \mu_{i(t)}(X_m) \cdot \left[\frac{x^j_m - c^j_{i(t)}}{(\sigma^j_{i(t)})^2} \right] + \beta_3 \cdot \Delta c^j_i(t-1) \tag{5.14}$$

$$\textit{Spread of the Premise Updating Law}: \quad \sigma^j_i(t+1) = \sigma^j_i(t) - \lambda_4 \cdot \varepsilon_m(t) \cdot \frac{g\left(b_{i(t)}, \sigma^y_{i(t)}\right) \cdot \left[b_{i(t)} - y^{crisp}\left(X_m | \theta(t)\right) \right]}{\sum_{i=1}^{k} \mu_{i(t)}(X_m) \cdot g(b_{i(t)}, \sigma^y_{i(t)})} \cdot \mu_{i(t)}(X_m) \cdot \left[\frac{(x^j_m - c^j_{i(t)})^2}{(\sigma^j_{i(t)})^3} \right] + \beta_4 \cdot \Delta \sigma^j_i(t-1) \tag{5.15}$$

$$\begin{aligned} & \varepsilon_m \triangleq y^{crisp}(X_m | \theta) - y_m \\ \textit{where};\quad & g'(b_i) \triangleq g^{,}\left(b_i, \sigma^y_i\right)|_{b_i} = \frac{1}{1 + \left(\frac{y_L - b_i}{\sigma^y_i}\right)^2} - \frac{1}{1 + \left(\frac{y_U - b_i}{\sigma^y_i}\right)^2} \\ & g'\left(\sigma^y_i\right) \triangleq g^{,}\left(b_i, \sigma^y_i\right)|_{\sigma^y_i} = \frac{1}{{}_{,}\sigma^y_i} \cdot \left[g\left(b_i, \sigma^y_i\right) + \frac{y_L - b_i}{1 + \left(\frac{y_L - b_i}{\sigma^y_i}\right)^2} - \frac{y_U - b_i}{1 + \left(\frac{y_U - b_i}{\sigma^y_i}\right)^2} \right] \end{aligned} \tag{5.16}$$

Comparing Eqs. 5.7~5.11 with Eqs. 5.12~5.16 leads to the conclusion that the two sets of parameter updating formulas (one for IMOFM_S and the other one for IMOFM_M) are very similar to one another. The only difference lies in the fact that the latter (Eqs. 5.12~5.16) include extra items, such as $g\left(b_i, \sigma^y_i\right)$ (refer to Eq. 5.6) and its partial derivatives with respect to b_i and σ^y_i, which allows the updating formulas to adjust the spreads of the output membership functions as well.

5.4.3 Constraint Back-Error-Propagation Algorithm

One problem associated with the above BEP updating formulas is that they include no constraints with respect to the update mechanism of these parameters. Hence, during the course of the optimisation, the centres are likely to be placed outside the boundaries. Although this does not affect the ultimate accuracy of FRBS, it may cause confusion for the users when assigning linguistic labels, and more importantly it may violate the search space which will be defined in the next modelling stage. Hence, in this work, a constraint handling scheme is added, which checks the boundary violation for centres during each iteration step and drives any violated centres back to the boundaries. The process is illustrated in Figure 5.5.

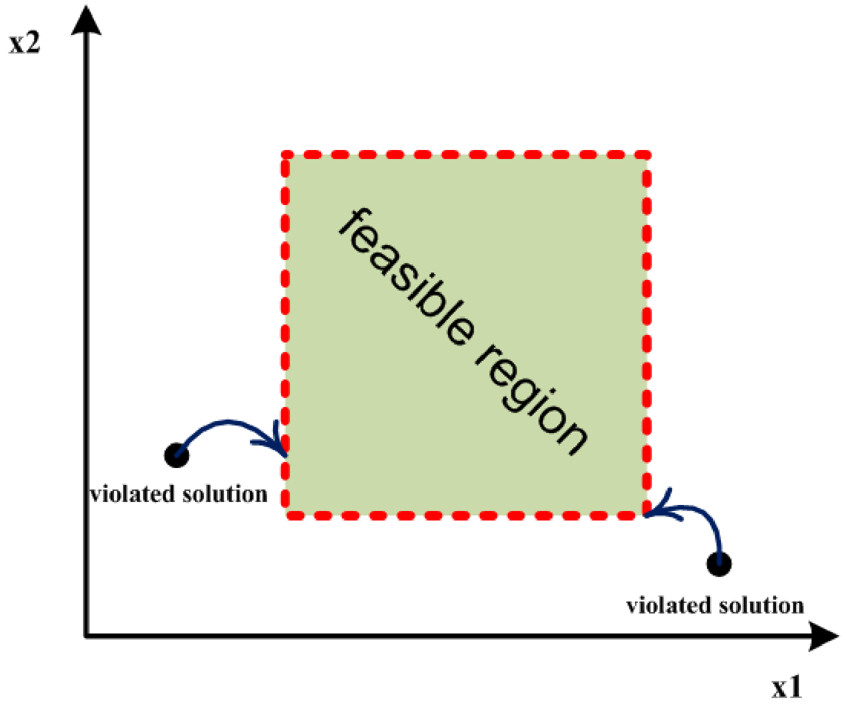


Figure 5.5 Violated solutions are dragged back to the boundaries.

5.4.4 An Example of Application

As the continuation of the example shown in Sections 4.4.3 and 5.3.2, the elicited FRBSs in those sections are further optimised (viz. parameter optimisation) using the developed BEP updating formulas. It is worth mentioning that the step sizes $\lambda_1 \sim \lambda_4$ and the gains of momentum terms $\beta_1 \sim \beta_4$ are all set to 0.03 in this work without any loss of generality. The number of iterations is set to 1500 for Singleton FRBS and 600 for Mamdani FRBS, which are the empirical numbers that ensure the convergence of the BEP algorithm. Since this example is only exploited for illustration purposes and the data itself is very limited, the whole data set is used for training. Hence, the over-training problem is not the particular concern in this section. Such problem will be formally dealt with in Chapter 6 by dividing the data set into training and testing sets for all applications. For some applications, such as Ultimate Tensile Strength, a small extra data set is also available, which serves as the validation set in our work. Figures 5.6 and 5.7 show the refined Singleton and Mamdani FRBSs along with their membership functions.

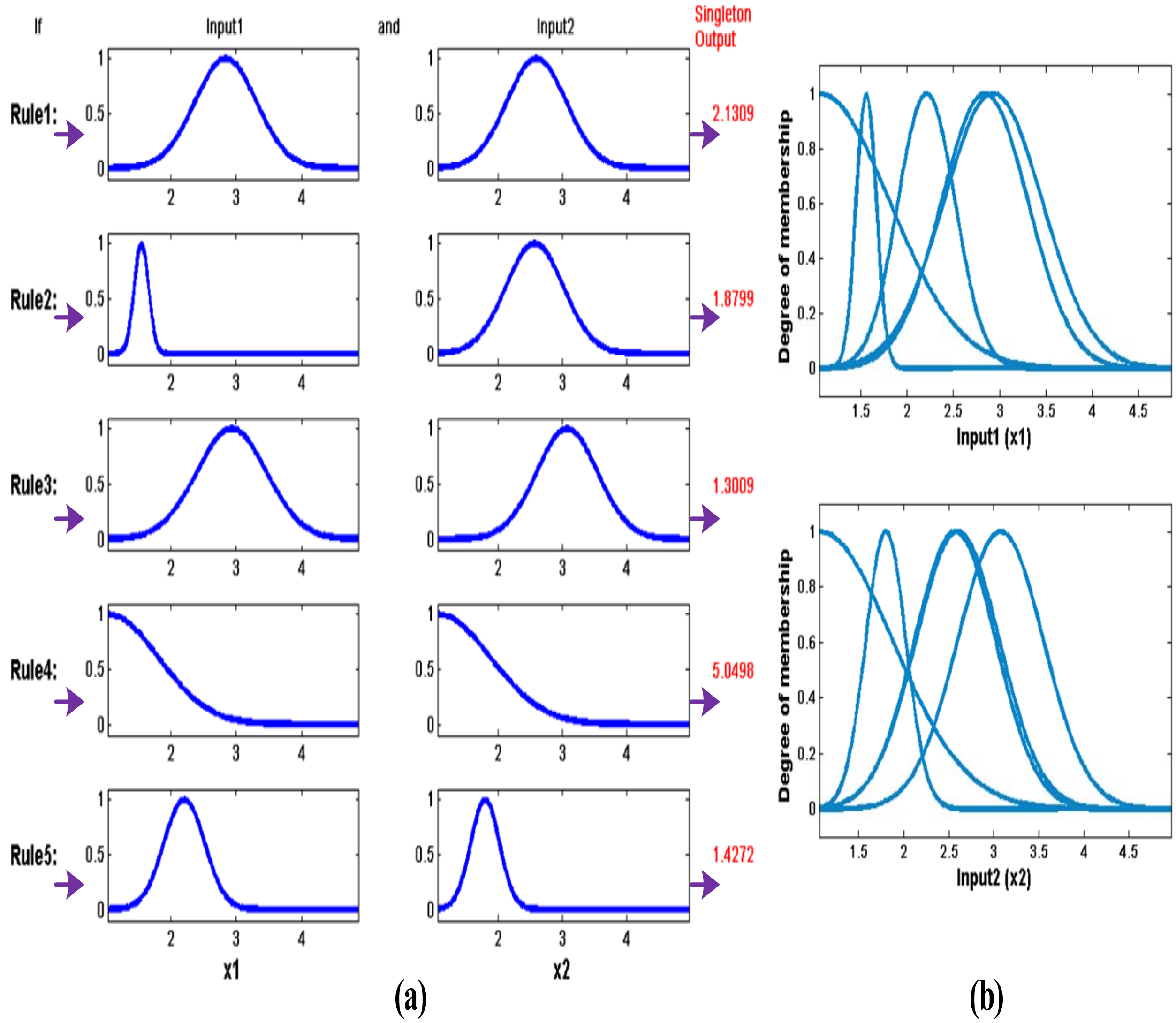


Figure 5.6 (a) the refined Singleton FRBS; (b) its associated membership functions.

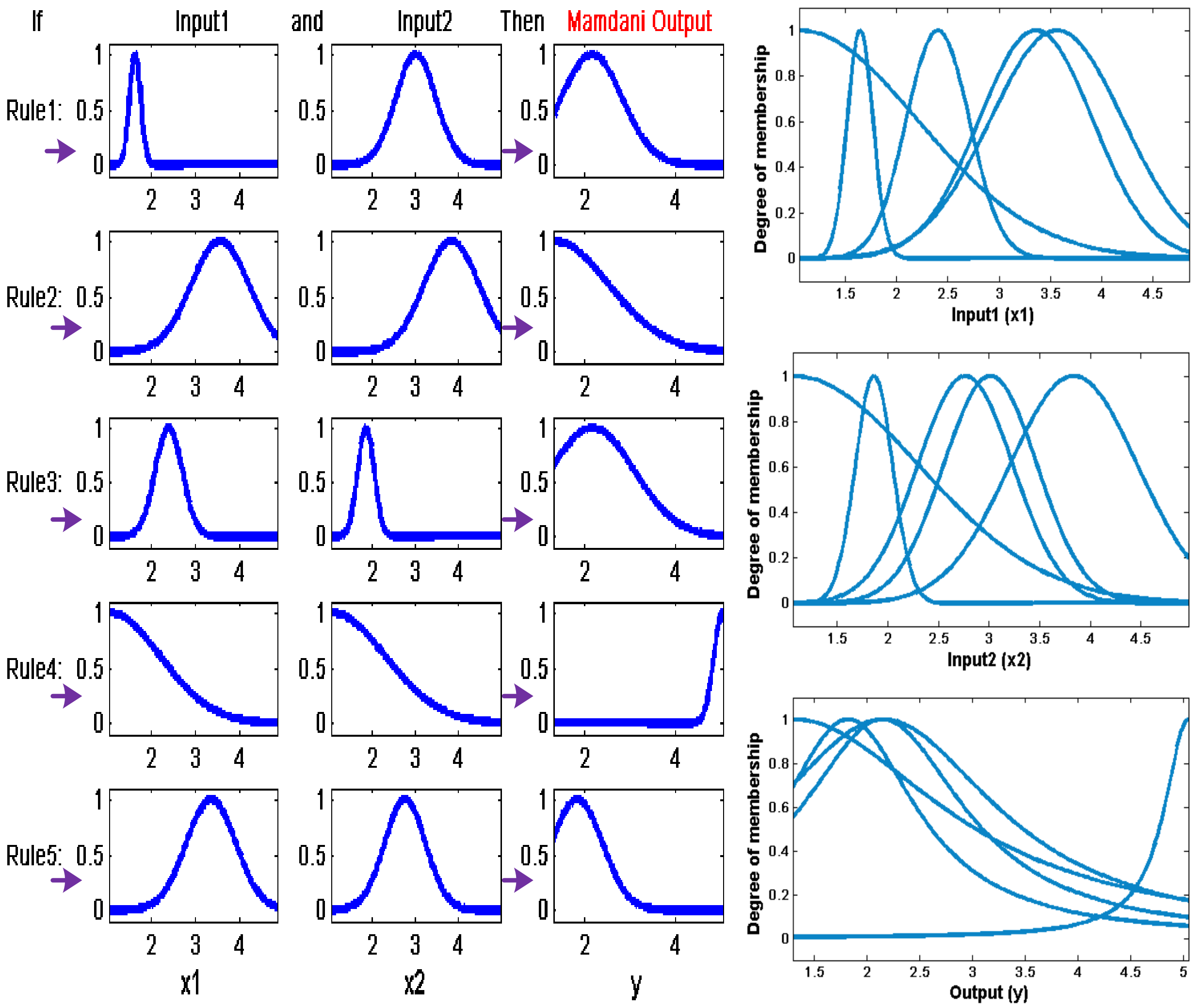


Figure 5.7 (a) the refined Mamdani FRBS; (b) its associated membership functions.

As one can see from Figures 5.6 and 5.7, the knowledge discovered by Singleton FRBS and Mamdani FRBS is consistent in terms of the distributions and the combinations of the membership functions (linguistic terms). However, the Mamdani FRBS has the advantage of being able to express clear semantic meanings in its consequents due to the inclusion of the width. As mentioned in Sections 4.4.3 and 5.1.4, the automatic rule induction process and unconstrained optimisation often lead to a deteriorated interpretability, and this is firmly supported by Figures 4.21, 5.6 and 5.7. It is because of this reason that the third modelling stage is a necessity and is normally included to improve model transparency. Figure 5.8 shows the predictive performances of the refined Singleton and Mamdani FRBSs by plotting their predicted outputs against the real outputs.

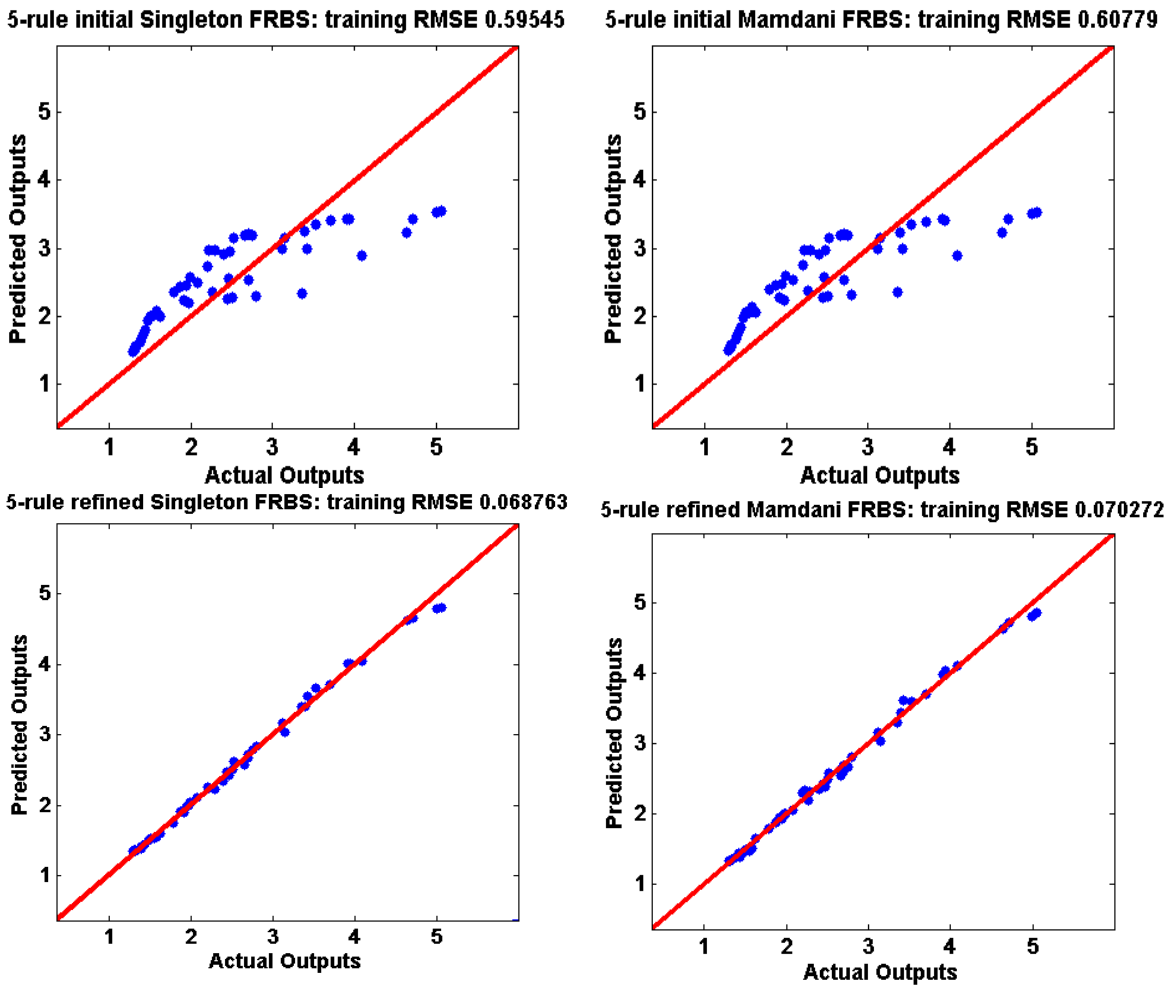


Figure 5.8 (a) the predictive performances of the initial and the refined Singleton FRBS; (b) the predictive performances of the initial and the refined Mamdani FRBS.

Table 5.2 summarises the predictive performances of the second modelling stage when using Singleton FRBS and Mamdani FRBS. The results are the average values of 20 independent runs. It can be seen from this table that, after the BEP refinement, both FRBSs' predictive performances are singificantly improved.

TABLE 5.2
THE PREDICTIVE PERFORMANCES OF THE SECOND MODELING STAGE OF IMOFM_S AND IMOFM_M ON A NONLINEAR STATIC SYSTEM WITH FIVE RULES

Modeling Methods	**The Predictive Performance of FRBSs from the 2nd Stage**		
	RMSE (average)	Std.	Time (sec.)
IMOFM_S	0.0688	0	120
IMOFM_M	0.0702	0	37

5.5 Third Stage: Immune Algorithms-based Multi-Objective Fuzzy Modelling

An optimal FRBS can be obtained by optimising the rule-base structure and membership function parameters either simultaneously or separately. The previous two modeling stages can be viewed as the instances of a separate structure and parameter learning. The drawbacks of the separate learning option are as follows:

- Only a 'sub-optimal' result may be obtained since both the structure and the parameters of the rule-base need to cooperate to provide a satisfactory FRBS.
- The separate learning structure relies too strongly on subjective judgment. Hence, only *challenge 2*, namely the need to set the start points, as mentioned in Section 5.1.4 would have been solved by the first two stages, which should mainly be attributed to the global search capacity of the G3Kmeans algorithm. As far as the other two limitations are concerned, one still has to set the initial abstraction level and only an approximate FRBS with obscure semantics can be elicited as a result.

To improve the interpretability of such an approximate FRBS, the authors in (Setnes *et al.*, 1998; Setnes *et al.*, 2000; Roubos *et al.*, 2001; Chen *et al.*, 2001) performed model simplifications and fine-tunings. The learning procedure described in these research investigations can still be labeled as being a separate learning process so that model simplifications rely heavily on the pre-specified thresholds according to the designer's choice. Wang *et al.* (2005) proposed a hierarchical scheme to evolve both parts. However, a rule matrix was required, which rendered the scheme vulnerable to high dimensional problems due to the exponential increase in the matrix dimension. Research work reported in (Jiménez *et al.*, 2001; Jiménez *et al.*, 2002; González *et al.*, 2007) adopted a variable length coding strategy in order to cope with high dimensional problems. However, as mentioned in Section 5.1.5, only heuristic variation operators are used in these works, which did not do justice to the idea of using variable length coding. In fact, it may somehow impede the search power of EAs as far as the real-valued optimisation part is concerned. Apart from these problems, research investigations in (Setnes *et al.*, 1998; Setnes *et al.*, 2000; Roubos *et al.*, 2001; Jiménez *et al.*, 2001; Jiménez *et al.*, 2002; Wang *et al.*, 2005; González *et al.*, 2007) dealt with TSK FRBS with linear functions as their consequents, which detracts from the linguistic attempts of the authors' proposed methods.

The proposed approach in this current research work utilises a multi-objective optimisation framework and a variable length coding scheme, which does not suffer from 'the curse of dimensionality'. A set of FRBSs representing the trade-offs between interpretability and accuracy are obtained through a single run, and only the maximum allowable number of rules is required *a priori*, which reduces any user intervention during the whole design process to a minimum level. As can be seen from Figure 5.3, a 'variable length coding scheme' and a 'model simplification' are integrated into the original PAIA2 to account for parameter and structure optimisation. A new distance index is proposed to facilitate the use of the original variation operator in PAIA2. Details of these operators and the way of formulating objective functions and the initial population pool are explained next.

5.5.1 Formulation of the Objective Functions

Ishibuchi (2004) formed three objective functions with the first one being concerned with the classification accuracy and the rest two focusing on the structure optimisation. In the work presented by Jimeńez (2001), similar objectives were formed with the first one relating to the predictive accuracy and the rest two being concerned with transparency and compactness measures.

However, not all the above objectives represent conflicting objectives which may lead to the difficulty in achieving a good distribution over the entire Pareto front. Furthermore, as pointed out by Ishibuchi (2008), the search capability of evolutionary multi-objective optimisation algorithms is severely deteriorated by the increase in the number of objectives. Hence, only two conflicting objective functions are formulated with the first focusing on the prediction accuracy and the second on the structure simplification as described in Eq. 5.17, where, *Nrule* is the number of fuzzy rules in FRBS; *Nset* is the total number of fuzzy sets; RL is the summation of the rule length of each rule.

$$\begin{array}{ll} Objective\ 1: & RMSE = \sqrt{\frac{\sum_{m=1}^{N}(y^{crisp}(X_m|\theta) - y_m)^2}{N}} \\ Objective\ 2: & Nrule + Nset + RL \end{array} \tag{5.17}$$

5.5.2 Formation of Initial Population Pool

The vaccine model elicited from the first two stages is used to seed the initial population pool so that a set of initial FRBSs will be randomly generated around the original vaccine model using the following equations:

$$\begin{gathered} C_{initial}{}_i^j = \alpha \cdot range^j \cdot randn + C_{vaccine}{}_i^j \\ \sigma_{initial}{}_i^j = \beta \cdot randn + \sigma_{vaccine}{}_i^j \\ C_{initial}{}_i^y = \alpha \cdot range^y \cdot randn + C_{vaccine}{}_i^y \\ \sigma_{initial}{}_i^y = \beta \cdot randn + \sigma_{vaccine}{}_i^y \end{gathered} \tag{5.18}$$

$$range = \min(|C_{vaccine} - U_{limit}|, |C_{vaccine} - L_{limit}|) \tag{5.19}$$

where, $C_{vaccine}{}_i^j$ and $\sigma_{vaccine}{}_i^j$ are the centre and spread of the ith rule and the jth input membership function in the original vaccine FRBS extracted from the first two modelling stages. $C_{vaccine}{}_i^y$ and $\sigma_{vaccine}{}_i^y$ are the centre and the spread of the ith rule's consequent. When IMOFM is used for evolving the Singleton FRBS, $\sigma_{vaccine}{}_i^y$ is not included. $randn$ is a random number within [0, 1]. '$range$' defines the minimum interval between the centre and its corresponding upper U_{limit} and lower L_{limit} limits of the input (or the output) variable, whichever is smaller. The inclusion of '$range$' is to ensure that the newly generated centres are most likely within the inputs' (or the output's) domains. Any violation of the domains will be corrected by dragging those centres (or consequents) back to the upper or lower limits, whichever is closest. α and β are the user specified parameters which define how much different the newly generated FRBSs are from the original vaccine one in order to maintain a certain diversity in the initial population.

Finally, the newly generated FRBSs and the original vaccine model will all be included in the initial population pool. Such a 'forming' approach only acquires the knowledge about the maximum allowable number of rules and the data so that emphasis of the third modelling stage is placed on the automatic elicitation of a set of FRBSs in the 'Pareto' sense. Alternatively, if more information about the system is available the initial population pool can be formed using expert knowledge, or some heuristics, which means that the first two modelling stages are not necessarily needed. The aim of the third modelling stage is then to locate more solutions between these already known models, i.e. filling up the gaps in the limited prior knowledge. Figure 5.9 visualises the aforementioned two 'forming' options. As already shown in Section 3.4.1, PAIA2 is not sensitive to the size of initial population, which ensures the feasibility of the proposed forming method as one can form as many initial populations as necessary.

In the following experiments, the initial population pool is formed using Eqs. 5.18 and 5.19 since no prior knowledge about the problems is assumed. α and β are set to 0.2 and 0.1 respectively in the following experiments without any loss of generality.

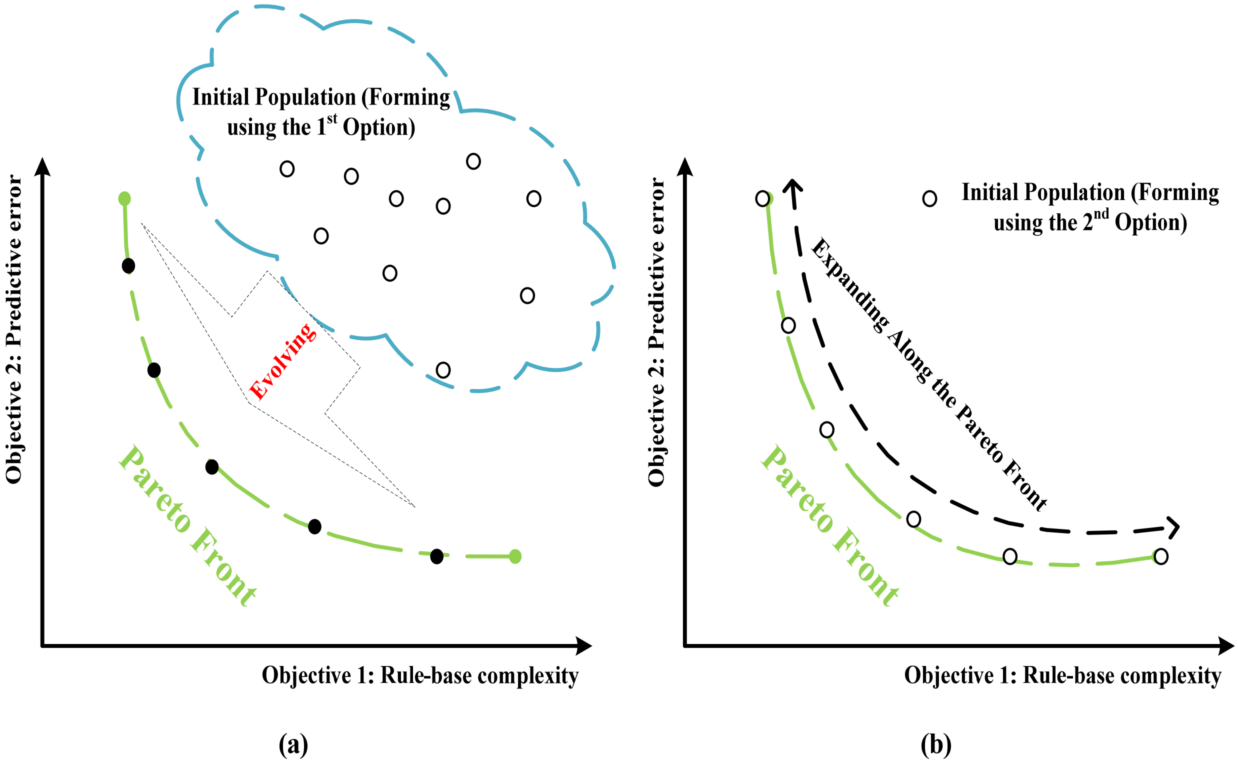

Initial Population (Forming using the 1st Option)
Objective 2: Predictive error
Evolving
Pareto Front
Objective 1: Rule-base complexity
(a)
Objective 2: Predictive error
Initial Population (Forming using the 2nd Option)
Expanding Along the Pareto Front
Pareto Front
Objective 1: Rule-base complexity
(b)

Figure 5.9 (a) If the initial population is formed using the 1st option, IMOFM is responsible for evolving the population towards the Pareto front; (b) if the initial population is formed using the 2nd option, IMOFM is responsible for filling the gaps between the limited knowledge.

5.5.3 A Variable Length Coding Scheme

The encoding scheme plays a vital role in all types of EA-based optimisation. As far as multi-objective fuzzy modelling is concerned, different encoding schemes have been proposed and can be broadly divided into two categories:

1. Encoding based on the global data base (linguistic term set);
2. Encoding based on the effective rule parameters.

The former is mainly found in the linguistic modeling stream (Ishibuchi *et al.*, 2004; cococcioni *et al.*, 2007; refer to Section 5.1.5), in which a global linguistic term set (data-base) is given *a prior* so that a string or a rule matrix can be formed as the chromosome in order to select the effective rules and linguistic terms from the candidate set; key to this type of encoding is that the global data-base is kept unchanged. The latter is mainly found in the approximate modelling stream (see Section 5.1.5) due to the lack of global data-base (Jim*é*nez *et al.*, 2001; Jim*é*nez *et al.*, 2002; Wang *et al.*, 2005). In the research investigation carried-out by Alcal*á* *et al.* (2007) and Gonz*á*lez *et al.*(2007), variants of the first encoding scheme were described, in which the encoding comprised the structure coding and the parameter (data-base) coding. The structure coding controls the 'on-and-off' of the genes in the parameter coding. The drawback of using the first encoding scheme and its variants is that it suffers from 'the curse of dimensionality'. In such a case, the length of the chromosome grows exponentially with the increased dimensions. A typical problem associated with these variants is illustrated in Figure 5.10 (a). Since most heuristic search methods rely on the interaction between individuals in the phenotypic space, which is the major thrust directing the search mechanism, an ineffective real-valued optimisation may be induced because some active parameter genes (grey ones) may interact with the inactive ones (blank ones). Such a problem can also be found in the work of Zhang *et al.* (2007) (see Figures 5.10 (b) and (c)), where a fixed length coding according to the maximum allowable number of rules is adopted, an ineffective optimisation may be induced because some parts of the long FRBS may interact with the 'inactive' part of the one with fewer rules. Conversely, if only the effective rule parameters are included in the coding, a variable length coding scheme is inevitable. One of the first attempts of this type for designing fuzzy controllers has been proposed by Cooper *et al.* (1994). Similar coding schemes can be found in (Jim*é*nez *et al.*, 2001; Jim*é*nez *et al.*, 2002; Wang *et al.*, 2005). Such a variable length coding scheme, which only encodes effective rules, is also employed in this work to account for the efficiency of the search and the curse of dimensionality. Since only the parameters of effective rules are encoded, the increase of the code length is only linear to the variable's dimension. Figures 5.10 (b) and (c) give examples of how to encode Singleton and Mamdani FRBSs with the different number of rules.

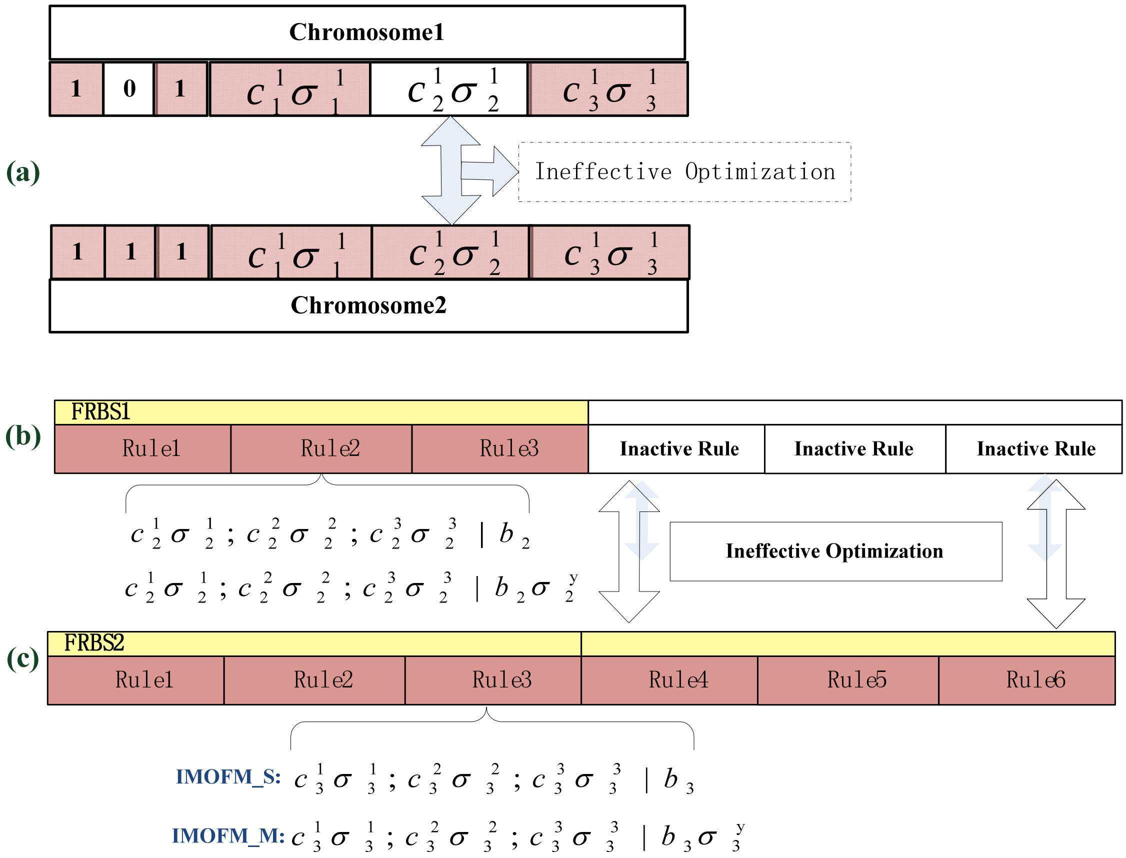


Figure 5.10 (a) Ineffective optimisation caused by the interaction of inactive gene (grey ones) and active gene (blank ones); (b) and (c) variable length coding scheme for a three-rule Singleton/Mamdani FRBS and a six-rule Singleton/Mamdani FRBS.

Given the variable length coding scheme and the unconstrained optimisation, a concomitant effect of the so-called 'unordered sets of rules' (Magdalena, 1998) may occur as shown in Figure 5.11, where FRBS1 and FRBS2 are exactly the same. However, because of the blind search mechanism, values encoded in 'Rule1' and 'Rule7' became different within the two FRBSs. Alternatively, rules may be deleted, e.g. Rule7 in FRBS2. Hence, a special procedure is required to align the closest rules from different FRBSs in order to have a meaningful crossover on the 'unordered sets of rules' (Cooper *et al.*, 1994; Magdalena, 1998). Although this problem has been realised and solved early-on during the development of the binary GA-based fuzzy controller, it was somehow overlooked later in the development of real-valued GA-based fuzzy models. In the research work proposed by Jiménez *et al.* (2001, 2002), arithmetic crossovers based on the random rules are employed both on the rule level and parameter level to account for the parameter tuning. As pointed out by Cooper *et al.* (1994), such crossovers are equivalent to combining the mother's gene for good vision and father's gene for curly hair, which does not make much sense. In Gonzảlez *et al.*' proposal (2007), although the alignment procedure is used, the so-called 'naive real-valued crossover' (Deb, 2001, p. 112) is included for the real parameter tuning, which impedes the search as far as the real-valued optimisation is concerned.

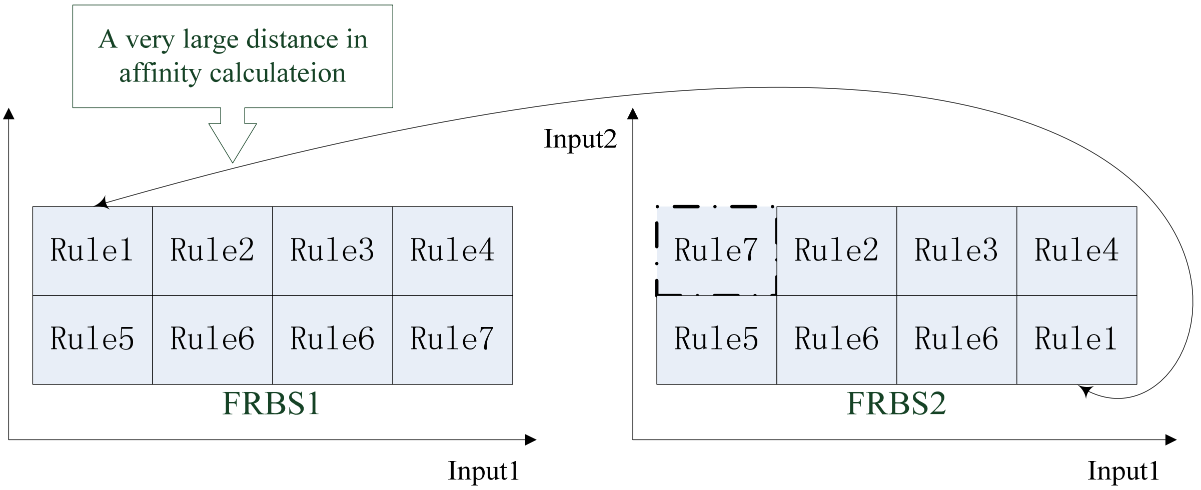


Figure 5.11 The problems associated with the FRBS having different rule lengths and unconstrained optimisation.

A similar problem is encountered if one wishes to use PAIA2 in the fuzzy modeling scenario. In the case shown in Figure 5.11, a very large distance is produced as the affinity value if the conventional distance measure, e.g. Euclidean distance, is directly used. In PAIA2 this would lead to a very large mutation, however, only a small or a non-jump is needed if the two interacted FRBSs are similar or exactly the same. To tackle the aforementioned problems, a new distance index is proposed to calculate the affinity for PAIA2 in the activation step. This will facilitate the use of the original effective search operator, viz. affinity maturation. The basic idea is to find the distance of the closest rules in different FRBSs rather than the distance of the corresponding rules. Hence, 'Rule1' in FRBS1 will be paired with 'Rule7' in FRBS2. The mathematical description of the idea is as follows:

$$dist\left(R_j, R_k\right) = \frac{\sum_{i1=1}^{k1}\sum_{l=1}^{rl}\left(R_j^{i1}(l)-R_k^{C_{i1}}(l)\right)+\sum_{i2=1}^{k2}\sum_{l=1}^{rl}\left(R_k^{i2}(l)-R_j^{C_{i2}}(l)\right)}{rl\cdot(k1+k2)} \tag{5.20}$$

where, R_j and R_k are two FRBSs with $k1$ and $k2$ rules; rl is the length of the rule; $R_k^{C_{i1}}$ ($R_j^{C_{i2}}$) represents the closest rule in R_k (R_j) with respect to the $i1th$ ($i2th$) rule in R_j (R_k). The above distance index is used to replace the one in PAIA2 for calculating the affinity (see Section 3.2.1 and Eqs. 3.1~3.2).

5.5.4 Improvement of Interpretability

As one can see from Figure 5.3, a model simplification step is added to PAIA2. The aim is to remove the redundancy both in the rules and in the fuzzy sets so that one can achieve the FRBS structure optimisation along with the accuracy at the same time. There are five steps involved in the model simplification module, which are discussed in the following Sections. The effects of the thresholds introduced in Sections 5.5.4.1~5.5.4.5 will also be analysed in Section 5.6.2.

5.5.4.1 Removing Unimportant Rules

Inspired by the idea behind neural network pruning, the unimportant rules are those rules that contribute the least to any prediction error increase when not including this rule, as described by Eq. 5.21. This occurs because other rules may already have covered the input region under these rules.

$$Insignificant_{rule} = \min_i |RMSE_{AR} - RMSE_{\tilde{i}}| \quad i = 1, \dots, k \tag{5.21}$$

where, $RMSE_{AR}$ is the root mean square error when all the rules in the rule base are used for predicting; $RMSE_{\tilde{i}}$ is the predictive error associated with the rule base when the ith rule is temporarily excluded. Insignificant rules are deleted when the following condition is met:

$$\left(\frac{cr}{maxr}\right) \cdot rnd > p_m \tag{5.22}$$

where, cr is the number of rules in the current FRBS; $maxr$ is the maximum allowable number of rules, which equals the number of clusters used in the first modelling stage; *rnd* is a random number between [0, 1]. p_m is a design parameter which limits the fewest rules in FRBS (in other words, the maximum rules that can be regarded as the insignificant rules) and has been set to 0.5 in this work without any loss of generality. At each iteration step, each cloned individual has one insignificant rule removed unless the rule base reaches the fewest rules designated by Eq. 5.22.

5.5.4.2 Removing Singleton Rules

Singleton rules are those rules which include fuzzy sets that are similar to the singleton set. Such rules should be removed because they may not be fired in most cases and may not be desirable for the generation of an interpretable rule-base (Wang *et al.*, 2005). These may be deleted subject to the following condition:

$$\sum_{j=1}^{n} \frac{\sigma_i^j}{n} < th\_rdr \triangleq rnd \cdot rdr \qquad (5.23)$$

where, n is the input dimension; th_rdr is a design parameter which randomly changes between $[0, rdr]$ every t iterations and rdr is 0.01 in the following experiments without any loss of generality. At each iteration step, one singleton rule is removed for each cloned individual given condition 5.23 is met.

5.5.4.3 Merging Similar Rules

During the simplification and the optimisation operations, rules may have similar fuzzy sets in the antecedent part. These rules should be merged together by taking the mean values of those fuzzy sets to keep the FRBS consistent and parsimonious. To measure the similarity of rules, the so-called similarity of rule premise (SRP) (Jin *et al.*, 1999) is used in this thesis. The following condition should be met for merging a pair of similar rules of each cloned individual at each iteration step:

$$SRP(i,l) \triangleq min\left\{S\left(A_i^j, A_l^j\right), \begin{matrix} j = 1, \dots, n \\ i, l = 1, \dots, k; i \neq l \end{matrix}\right\} > th\_mr \triangleq rnd \cdot (1 - mr) + mr \qquad (5.24)$$

where, $S\left(A_i^j, A_l^j\right)$ are the similarity between two fuzzy sets and will be explained in Section 5.5.4.5; th_mr is the threshold which randomly changes between [mr, 1] every t (specified by the user) iterations and mr is 0.95 in this work without any loss of generality.

The above three operations (see Sections 5.5.4.1~5.5.4.3) are applied to the rule level as visualised by Figure 5.12.

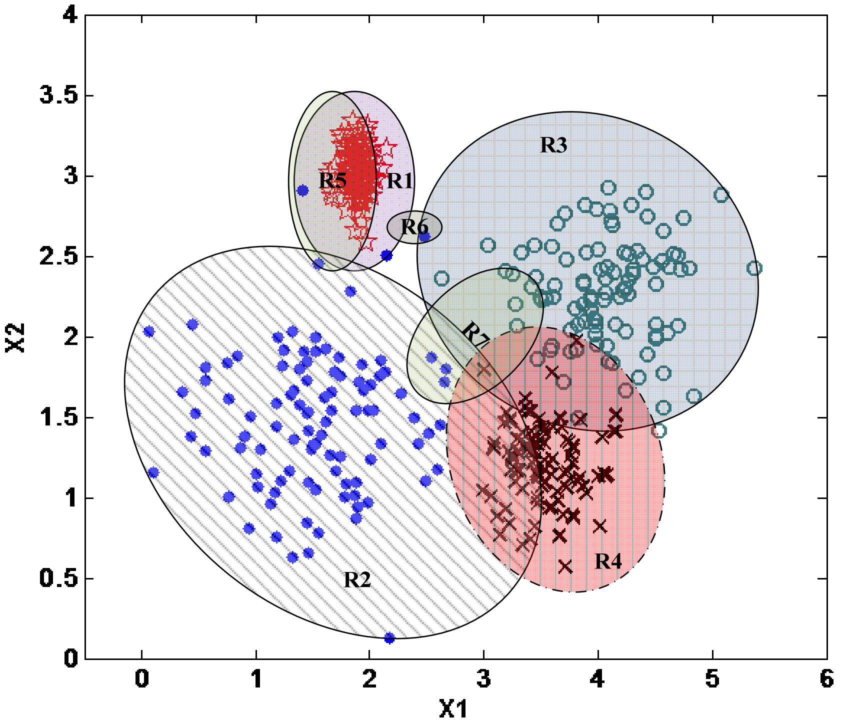


Figure 5.12 The example used in Section 4.3.1.1 with two inputs: (1) R1 and R5 are similar rules; (2) R6 is the singleton rule; (3) R7 is the insignificant rule.

5.5.4.4 Removing Universal Fuzzy Sets

Fuzzy sets which meet the following condition are regarded as universal fuzzy sets and are therefore deleted:

$$S\left(A_i^j, U\right) > th\_ufs \triangleq rnd \cdot (1 - ufs) + ufs \tag{5.25}$$

where, U is the universal fuzzy set; th_ufs is the threshold which randomly changes between $[ufs,\ 1]$ every t generations and ufs is 0.85 in this work. For computation purpose, if the width of a fuzzy set is more than two times wider than the universe of discourse of the corresponding dimension, it is regarded as the universal fuzzy set. Figure 5.13 illustrates such a case, where the *centre* of the fuzzy set is 0.5 and the spread is 2 on the universe of discourse: [0, 1].

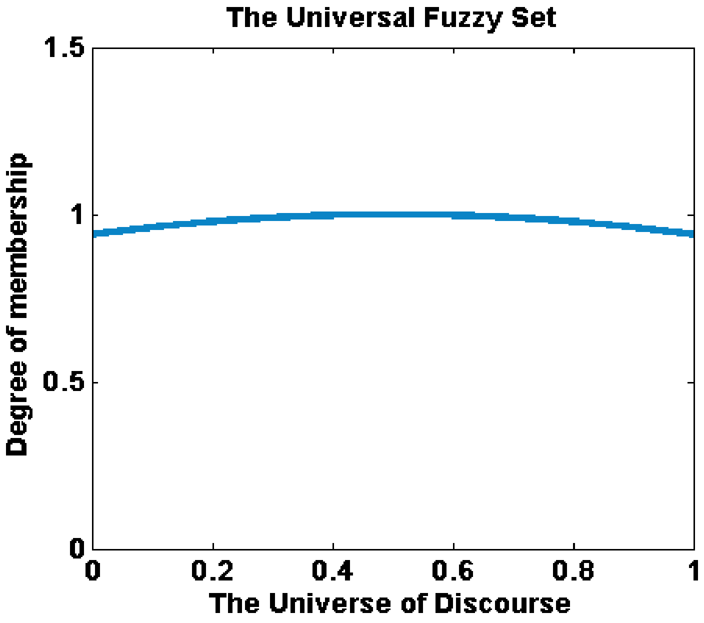


Figure 5.13 A fuzzy set with its spread more than two times wider than the universe of discourse is regarded as the universal fuzzy set.

5.5.4.5 Merging Similar Fuzzy Sets

Jin (2000) proposed a simplified similarity measure based on the distance measure if Gaussian membership functions are involved. Although this measure does not satisfy all the conditions mentioned by Setnes *et al.* (1998), it works well when it tries to locate similar fuzzy sets in our case. Two fuzzy sets are considered to be similar if the following condition is met:

$$\begin{gathered} S\left(A_i^j, A_l^j\right) \; or \; S\left(A_i^y, A_l^y\right) > th\_sfs \triangleq rnd \cdot (1 - sfs) + sfs \\ S\left(A_i^j, A_l^j\right) = \frac{1}{1+\sqrt{(c_i^j - c_l^j)^2 + (\sigma_i^j - \sigma_l^j)^2}} \\ S\left(A_i^y, A_l^y\right) = \frac{1}{1+\sqrt{(c_i^y - c_l^y)^2 + (\sigma_i^y - \sigma_l^y)^2}} \end{gathered} \tag{5.26}$$

where, th_sfs is the threshold which randomly changes between [sfs, 1] every t generations and sfs is set to 0.95 in this work. The mean values of two similar fuzzy sets are calculated in order to substitute the original two fuzzy sets. It is worth mentioning that $S\left(A_i^y, A_l^y\right)$ is also checked if IMOFM_M is used. Figure 5.14 shows an example of merging two fuzzy sets with the same width and different centres at 0.45 and 0.5 respectively.

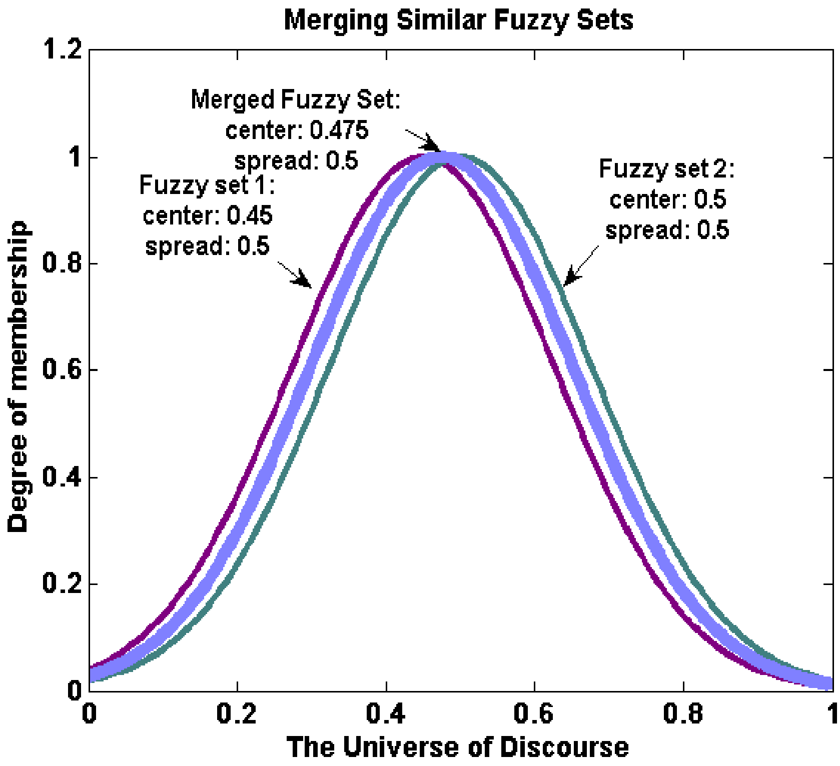


Figure 5.14 An example of merging similar fuzzy sets.

It is worth mentioning that all the simplification processes, except for the ‘insignificant rules’, have only α chance to be evoked at each iteration, where α is taken to be 20% in this work without any loss of generality. The similarity measures mentioned in Sections 5.5.4.4 and 5.5.4.5 will be checked for each fuzzy set. Only the ones with the maximum similarity values will be deleted or merged during each iteration step provided the conditions mentioned in Eqs. 5.25 and 5.26 are also met. For this reason and because of the elitism which records any non-dominated solution found at each iteration step during the experiments, it was found that the aforementioned thresholds are not critical parameters. Section 5.6.2 expands on such observation.

5.5.5 Algorithm Implementation Issues

Due to the simultaneous optimisation of the rule base structure and its parameters in the third modelling stage, some issues regarding the practical implementation of the algorithm should be treated with a special caution and deserves more exploration in this Section. In the following space, three issues are discussed, which are all vital to the proposed mechanisms for improving model’s interpretability in the third modelling stage.

The first issue is that the rule-base should be normalised before the third modelling stage is use so that all the *centre*s lie within the interval [0, 1]. This is to ensure that the affinity maturation operator described in Section 3.2.1 is still an effective search operator even if the scales of different dimensions are quite different. Eq. 5.27 provides the formulas on how to normalise a rule base. Where, U^j and L^j are the upper and lower limits on the jth input dimension; U^y and L^y are the upper and lower limits on the output dimension. It is worth mentioning that $\sigma_i^y\_norm$ is only calculated when IMOFM_M is adopted.

$$\begin{gathered} c_i^j\_norm = (c_i^j - L^j)/(U^j - L^j) \\ \sigma_i^j\_norm = \sigma_i^j/(U^j - L^j) \\ c_i^y\_norm = (c_i^y - L^y)/(U^y - L^y) \\ \sigma_i^y\_norm = \sigma_i^y/(U^y - L^y) \end{gathered} \tag{5.27}$$

In real applications, the differences between different dimensions are frequently encountered and are usually up to many orders of magnitudes. Hence, normalisation is a very important step to ensure a good optimisation result.

The second issue is raised because of the actual use of the rule base coding (see Figure 5.15) for the parameter optimisation and the rule-base itself for the structure optimisation. Such a scheme of using different representations of the same model for different optimisation purposes calls for a link to bridge the gap between the two representations. The link is particularly important when the structure optimisation is performed along with the parameter optimisation. Without the link, the parameter optimisation operated over the coding representation may lose vital structural information which is constantly modified during the structure simplifications (optimisation). Figure 5.15 shows one such scenario which may cause the mentioned problem if such a link is missing (the example is demonstrated via a Mamdani FRBS, however, the observation is applicable to Singleton FRBS as well).

The upper part of Figure 5.15 is a 3-rule Mamdani FRBS with two inputs and one output. Suppose A_1^1 and A_3^1 are very similar such that the conditions defined in Eq. 5.26 are all met, these two membership functions will then be combined into a single one ($\breve{A}_1^1$) after the step of 'merging similar membership functions' (refer to Section 5.5.4.5). However, when converting this simplified rule-base into its coding representation, the code itself will not know that c_1^1 and c_3^1 are indeed from the same membership function $\breve{A}_1^1$. If it happens that PAIA2 chooses c_1^1 and σ_1^y as its mutation points, without the link between the modified rule base and its coding representations, PAIA2 will not apply the same optimisation that has been applied to c_1^1 to c_3^1. Hence, after the parameter optimisation, the converted rule-base from this coding representation will have two distinctive membership functions for the first input in Rules 1 and 3, i.e. $\hat{A}_1^1$ and $\breve{A}_1^1$. In other words, the parameter optimisation may not take into account the structure optimisation, which also means that the parameter optimisation and the structure optimisation cannot work concurrently unless the link between them is already set up.

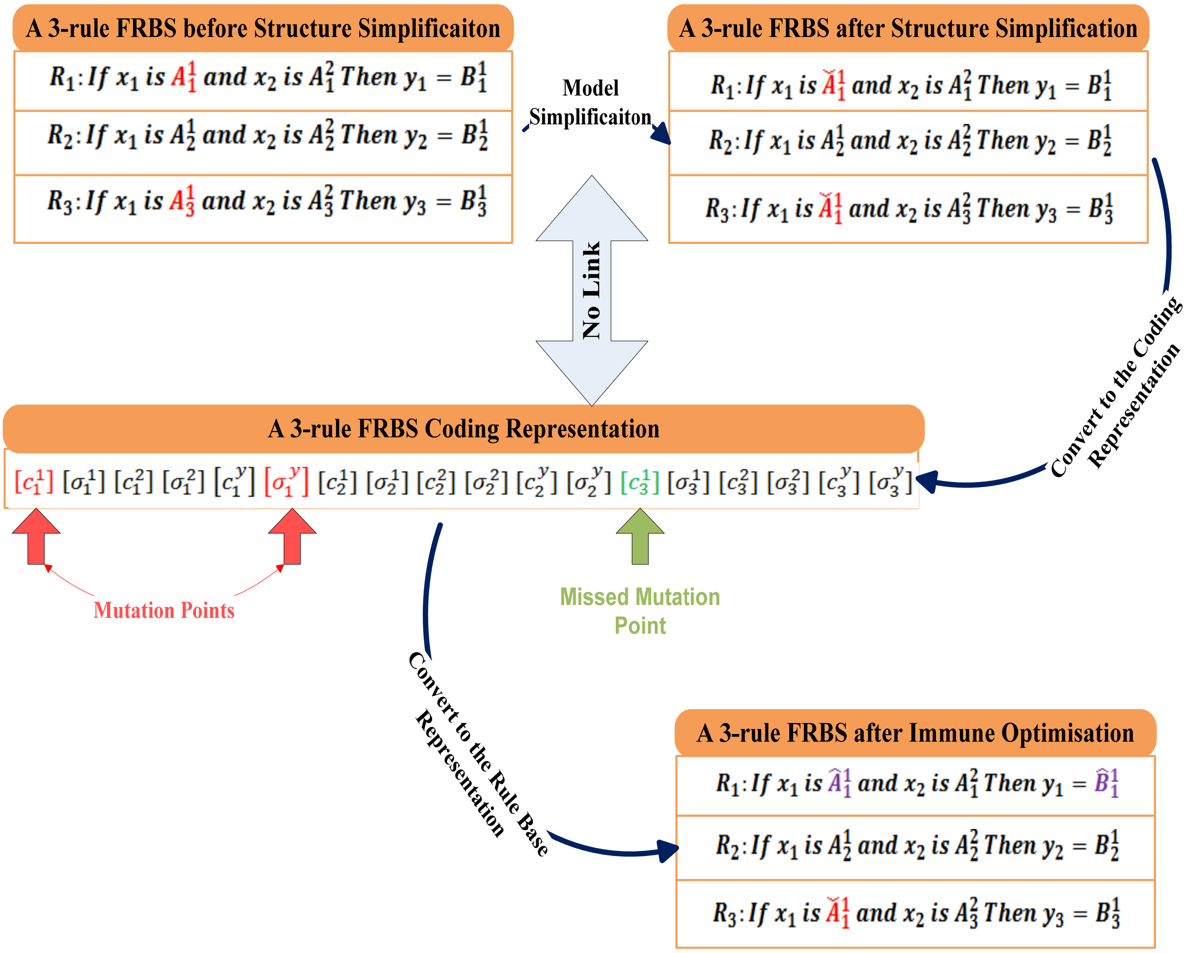

A 3-rule FRBS before Structure Simplificaiton
R_1: If x_1 is A_1^1 and x_2 is A_1^2 Then $y_1 = B_1^1$
R_2: If x_1 is A_2^1 and x_2 is A_2^2 Then $y_2 = B_2^1$
R_3: If x_1 is A_3^1 and x_2 is A_3^2 Then $y_3 = B_3^1$
Model Simplificaiton
A 3-rule FRBS after Structure Simplification
R_1: If x_1 is $\breve{A}_1^1$ and x_2 is A_1^2 Then $y_1 = B_1^1$
R_2: If x_1 is A_2^1 and x_2 is A_2^2 Then $y_2 = B_2^1$
R_3: If x_1 is $\breve{A}_1^1$ and x_2 is A_3^2 Then $y_3 = B_3^1$
No Link
Convert to the Coding Representation
A 3-rule FRBS Coding Representation
$[c_1^1]$ $[\sigma_1^1]$ $[c_1^2]$ $[\sigma_1^2]$ $[c_1^y]$ $[\sigma_1^y]$ $[c_2^1]$ $[\sigma_2^1]$ $[c_2^2]$ $[\sigma_2^2]$ $[c_2^y]$ $[\sigma_2^y]$ $[c_3^1]$ $[\sigma_3^1]$ $[c_3^2]$ $[\sigma_3^2]$ $[c_3^y]$ $[\sigma_3^y]$
Mutation Points
Missed Mutation Point
Convert to the Rule Base Representation
A 3-rule FRBS after Immune Optimisation
R_1: If x_1 is $\hat{A}_1^1$ and x_2 is A_1^2 Then $y_1 = \hat{B}_1^1$
R_2: If x_1 is A_2^1 and x_2 is A_2^2 Then $y_2 = B_2^1$
R_3: If x_1 is $\breve{A}_1^1$ and x_2 is A_3^2 Then $y_3 = B_3^1$

Figure 5.15 If no links are set up for the rule base and its coding representation, a missed mutation point may be induced.

In order to build up such a link, two concepts, namely ‘FISmap’ and ‘RULE’, are introduced into IMOFM. The example shown in Figure 5.15 is reinterpreted in Figure 5.16 with the aid of ‘FISmap’ and ‘RULE’. The only difference is that ‘Rule 2’ is now subject to deletion.

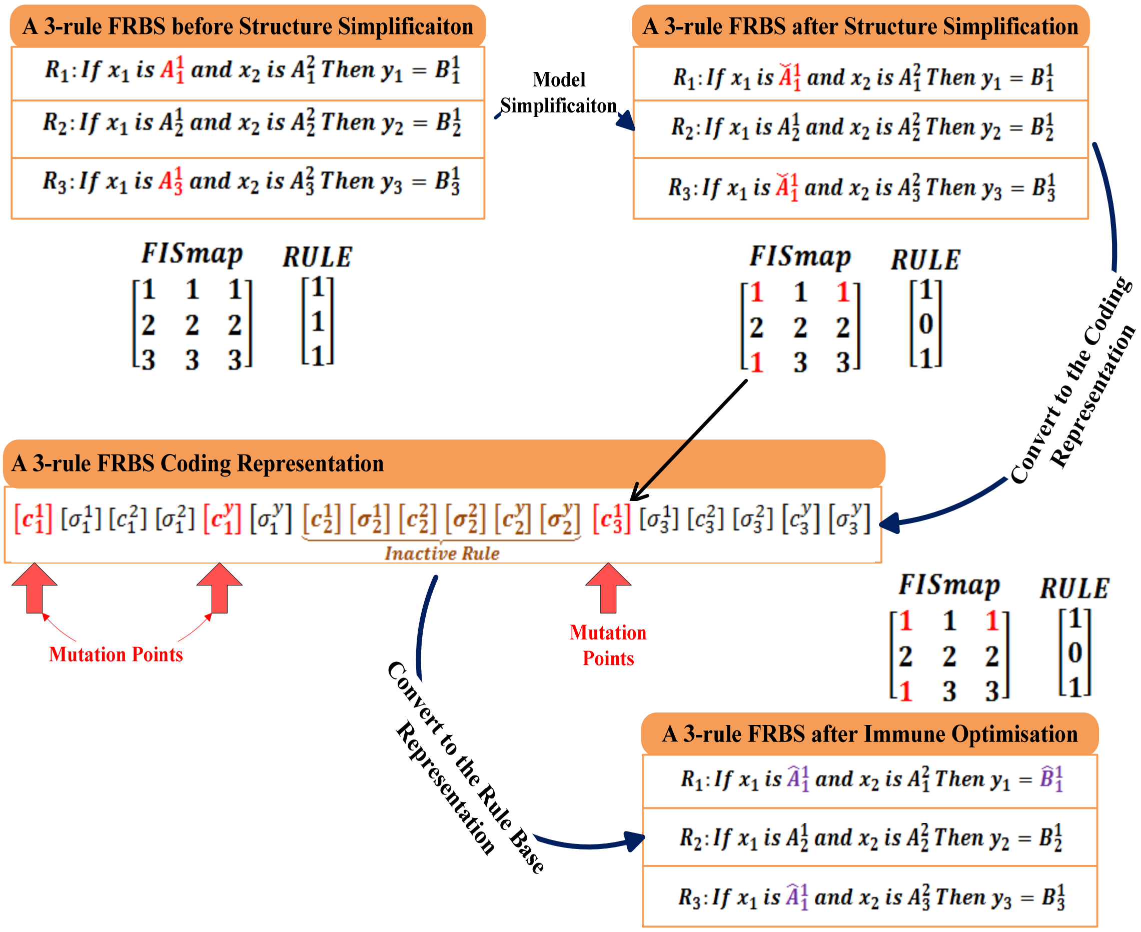

A 3-rule FRBS before Structure Simplificaiton
R_1: If x_1 is A_1^1 and x_2 is A_1^2 Then $y_1 = B_1^1$
R_2: If x_1 is A_2^1 and x_2 is A_2^2 Then $y_2 = B_2^1$
R_3: If x_1 is A_3^1 and x_2 is A_3^2 Then $y_3 = B_3^1$
Model Simplificaiton
A 3-rule FRBS after Structure Simplification
R_1: If x_1 is $\breve{A}_1^1$ and x_2 is A_1^2 Then $y_1 = B_1^1$
R_2: If x_1 is A_2^1 and x_2 is A_2^2 Then $y_2 = B_2^1$
R_3: If x_1 is $\breve{A}_1^1$ and x_2 is A_3^2 Then $y_3 = B_3^1$
FISmap
RULE
$\begin{bmatrix}1&1&1\\2&2&2\\3&3&3\end{bmatrix}$ $\begin{bmatrix}1\\1\\1\end{bmatrix}$
$\begin{bmatrix}1&1&1\\2&2&2\\1&3&3\end{bmatrix}$ $\begin{bmatrix}1\\0\\1\end{bmatrix}$
Convert to the Coding Representation
A 3-rule FRBS Coding Representation
$[c_1^1]\,[\sigma_1^1]\,[c_1^2]\,[\sigma_1^2]\,[c_1^y]\,[\sigma_1^y]\,\underbrace{[c_2^1]\,[\sigma_2^1]\,[c_2^2]\,[\sigma_2^2]\,[c_2^y]\,[\sigma_2^y]}_{\text{Inactive Rule}}\,[c_3^1]\,[\sigma_3^1]\,[c_3^2]\,[\sigma_3^2]\,[c_3^y]\,[\sigma_3^y]$
Mutation Points
Mutation Points
Convert to the Rule Base Representation
FISmap
RULE
$\begin{bmatrix}1&1&1\\2&2&2\\1&3&3\end{bmatrix}$ $\begin{bmatrix}1\\0\\1\end{bmatrix}$
A 3-rule FRBS after Immune Optimisation
R_1: If x_1 is $\hat{A}_1^1$ and x_2 is A_1^2 Then $y_1 = \hat{B}_1^1$
R_2: If x_1 is A_2^1 and x_2 is A_2^2 Then $y_2 = B_2^1$
R_3: If x_1 is $\hat{A}_1^1$ and x_2 is A_3^2 Then $y_3 = B_3^1$

Figure 5.16 If a link is set up, no missed mutation points are induced; inactive rules are not actually deleted but marked so that it will not participate in any computations afterwards.

The first concept is the so-called 'FISmap' matrix, which is a $k \times n$ matrix for IMOFM_S and $k \times (n+1)$ for IMOFM_M. The elements of the ith row are all initialised to their row number and will be constantly updated so that it can reflect the current status of the rule base. The number stored in each element serves as the identification number of each membership function. For example, during the interpretability improvement operation at each iteration step, if the membership function of the first input in the ith rule is very similar to the membership function of the same input in the jth rule, two membership functions in the rule-base representation will be merged into a single one. In order to reflect such changes in the rule base structure, FISmap is updated, and if $i < j$, FISmap(i, 1) will remain to its initialised number 'i' and FIS(j, 1) will be updated using the smaller number 'i'. By doing so, two similar membership functions would have been combined into a single one and their corresponding identification numbers would also have been updated using the smaller value. If some membership functions' spreads are wider enough (refer to Section 5.5.4.4 for the definition of 'wide') to be considered as the universal fuzzy set, then the corresponding element in FISmap is updated using 'inf' ('inf' represents infinity in Matlab®) to reflect this fact. However, the real spreads in the rule base representation are set to 5 (recall the universe of discourse is normalised within the interval [0, 1]) rather than 'inf' for the computational purpose. The second concept relates to a so-called vector 'RULE', which is a $k \times 1$ vector initialised with 1. This vector serves as the flag to indicate which rule in the rule-base is active and which rule is inactive. Rules satisfy the conditions defined in Eqs. 5.22~5.24 are deleted or merged, which lead to the corresponding elements in 'RULE' flipping from 1 to 0 (hence, 'Rule 2' is an inactive rule as shown in Figure 5.16).

As one can see from Figure 5.16, if PAIA2 chooses c_1^1 and σ_1^y as the mutation points, the next step is to check FISmap to see if there are any elements in the same columns of FISmap whose identification numbers are the same as FISmap(1, 1) and FISmap(1, 3). If there are such elements, such as FISmap(3, 1) in this case, a calculation is carried-out so that FISmap(3, 1) is mapped into the index of c_3^1 in the coding representation. It is worth mentioning that due to the use of 'RULE', the variable length coding scheme is realised without the need of deleting the inactive rules. 'RULE' is consulted before the new distance index (Eq. 5.20) and all the structure simplifications mentioned in Section 5.5.4 can actually be performed. Hence, only active rules are involved in those computing.

The third issue only relates to IMOFM_M. In a Mamdani FRBS, the spreads of the output membership functions are also subject to unconstrained optimisation. Hence, it is very likely that some spreads become wide enough to be considered as the universal fuzzy set. However, it is every hard to associate any meaningful linguistic terms with the universal fuzzy set for the consequents. The solution to this problem is to impose a constraint on the spreads of the output membership functions so that they will not exceed 1 in a normalised universe of discourse.

5.5.6 An Example of Application

As the continuation of the example shown in Sections 4.4.3, 5.3.2 and 5.4.4, the elicited FRBSs in these Sections are further optimised (viz. simultaneous optimisation of the parameters and the rule-base structure) in this section using the developed IMOFM_S and

IMOFM_M. The refined Singleton and Mamdani FRBSs (see Section 5.4.3) are used as the 'vaccine FRBSs' to generate a set of seven initial individuals using Eqs. 5.18 and 5.19, each including five rules. The number of iterations is set to 1200 for both IMOFM_S and IMOFM_M. The network suppression threshold of PAIA2 is set to 0.0008 for this example to manage the population within the solution pool. The effect of this parameter on the final solutions is analysed in Section 5.6.2. Other parameters of PAIA2 are kept the same as those introduced in Section 3.4. In order to obtain a quantitative comparison of the proposed method with other well-known fuzzy modelling paradigms, IMOFM is compared with the methods proposed by Wang *et al.*, (2005), Lin *et al.*, (1997), Sugeno *et al.*, (1993), Delgado *et al.*, (1997) and Chen *et al.*, (2004). Table 5.3 summarises such comparative results focusing on their predictive performances (RMSE). The results in Table 5.3 include the average values of 30 runs.

TABLE 5.3

COMPARISONS OF THE PREDICTIVE PERFORMANCE OF THE DIFFERENT MODELING METHODS FOR THE EXAMPLE

Modeling Methods (Ref.)	No. of rules	No. of fuzzy sets$^{\&}$	No. of Parameters	Consequents	Performance (RMSE training)	
Y. H. Lin *et al.*, (1997)	6	12 trapzoidal*/Gaussian$^{@}$	30*/42$^{@}$	Singleton	0.5925*	0.0707$^{@}$
M. Sugeno *et al.*, (1993)	6	12 trapzoidal	72	Fuzzy sets	0.5639*	0.2811$^{@}$
M. Delgado *et al.*, (1997)	5	10	25	Singleton	0.5604*	0.3391$^{@}$
H. L. Wang *et al.*, (2005)						
Initial	6	12 Gauss2mf.	66	Linear	-	0.1755$^{@}$
Pareto FRBS1	7	6 Gauss2mf.	45	Linear		0.0298$^{\#}$
Pareto FRBS2	4	3 Gauss2mf.	24	Linear		0.0520$^{\#}$
Pareto FRBS3	3	2 Gauss2mf.	17	Linear		0.0719$^{\#}$
M. Y. Chen *et al.*, (2004)						
Pareto FRBS1	4	8 Gaussian	28	Linear		0.0656$^{\#}$
Pareto FRBS2	4	5 Gaussian	22	Linear		0.0883$^{\#}$
Pareto FRBS3	3	5 Gaussian	19	Linear		0.1382$^{\#}$
Pareto FRBS4	2	4 Gaussian	14	Linear		0.2750$^{\#}$
IMOFM_S (NB: Average results over 30 runs are presented here)						
Average execution time (Intel(R) Core(TM)2 Duo CPU, 2.27 GHz): 3rd stage: 213sec						
Initial FRBS	5	10 Gaussian	25	Singleton	0.5954*	0.0688$^{@}$
Pareto FRBS1(30 times)	5	10 Gaussian	25	Singleton	0.0688$^{\#}$	σ^2: 0
Pareto FRBS2(30 times)	5	9 Gaussian	23	Singleton	0.0696$^{\#}$	σ^2: 0
Pareto FRBS3(12 times)	5	8 Gaussian	21	Singleton	0.0875$^{\#}$	σ^2: 0.0044
Pareto FRBS4(29 times)	4	8 Gaussian	20	Singleton	0.0930$^{\#}$	σ^2: 0.0105
Pareto FRBS5(30 times)	4	7 Gaussian	18	Singleton	0.1152$^{\#}$	σ^2: 0.0101
Pareto FRBS6(29 times)	3	6 Gaussian	15	Singleton	0.1417$^{\#}$	σ^2: 0.0045
Pareto FRBS7(21 times)	3	5 Gaussian	13	Singleton	0.1884$^{\#}$	σ^2: 0.0042
Pareto FRBS8(30 times)	2	4 Guassian	10	Singleton	0.2484$^{\#}$	σ^2: 0.0015
Pareto FRBS9(2 times)	2(6^{T})	3 Gaussian	8	Singleton	0.7087$^{\#}$	σ^2: 0.0022
Pareto FRBS10(25 times)	2(5^{T})	3 Gaussian	6	Singleton	0.4769$^{\#}$	σ^2: 0.0719
Pareto FRBS11(1 time)	2(5^{T})	2 Gaussian	6	Singleton	0.7392$^{\#}$	σ^2: 0
Pareto FRBS12(21 times)	2(4^{T})	2 Gaussian	6	Singleton	0.7070$^{\#}$	σ^2: 0.0259
Pareto FRBS13(1 times)	1	2 Gaussian	5	Singleton	1.0326$^{\#}$	σ^2: 0
Pareto FRBS14(22 times)	1(2^{T})	1 Gaussian	3	Singleton	1.0326$^{\#}$	σ^2: 0

Table 5.3 to be continued...

Table 5.3 continued...

IMOFM_M (NB: Average results over 30 runs are presented here)						
Average execution time (Intel(R) Core(TM)2 Duo CPU, 2.27 GHz): 3rd stage: 229sec						
Initial FRBS	5	15 Gaussian	30	Mamdani	0.6078^*	$0.0702^@$
Pareto FRBS1(14 times)	5	15 Gaussian	30	Mamdani	$0.0633^{\#}$	σ^2: 0.0005
Pareto FRBS2(25 times)	5	14 Gaussian	28	Mamdani	$0.0651^{\#}$	σ^2: 0.0023
Pareto FRBS3(22 times)	5	13 Gaussian	26	Mamdani	$0.0691^{\#}$	σ^2: 0.0017
Pareto FRBS4(7 times)	5	12 Gaussian	24	Mamdani	$0.0711^{\#}$	σ^2: 0.0033
Pareto FRBS5(1 time)	5	11 Gaussian	22	Mamdani	$0.0756^{\#}$	σ^2: 0
Pareto FRBS6(10 times)	4	12 Gaussian	24	Mamdani	$0.0743^{\#}$	σ^2: 0.0013
Pareto FRBS7(26 times)	4	11 Gaussian	22	Mamdani	$0.0781^{\#}$	σ^2: 0.0034
Pareto FRBS8(25 times)	4	10 Gaussian	20	Mamdani	$0.0961^{\#}$	σ^2: 0.0032
Pareto FRBS9(3 times)	4	9 Gaussian	18	Mamdani	$0.1212^{\#}$	σ^2: 0.0042
Pareto FRBS10(28 times)	3	9 Gaussian	18	Mamdani	$0.1311^{\#}$	σ^2: 0.0152
Pareto FRBS11(28 times)	3	8 Gaussian	16	Mamdani	$0.1846^{\#}$	σ^2: 0.0193
Pareto FRBS12(12 times)	3	7 Gaussian	14	Mamdani	$0.2257^{\#}$	σ^2: 0.0014
Pareto FRBS13(25 times)	2	6 Gaussian	12	Mamdani	$0.2482^{\#}$	σ^2: 0.0019
Pareto FRBS14(28 times)	2(5^T)	5 Gaussian	10	Mamdani	$0.2718^{\#}$	σ^2: 0.0617
Pareto FRBS15(8 times)	2(4^T)	4 Gaussian	8	Mamdani	$0.4712^{\#}$	σ^2: 0.0154
Pareto FRBS16(17 times)	2(4^T)	4 Gaussian	8	Mamdani	$0.7040^{\#}$	σ^2: 0.0110
Pareto FRBS17(27 times)	1(2^T)	2 Gaussian	4	Mamdani	$1.0326^{\#}$	σ^2: 0

[&] For IMOFM_S, it is the number of fuzzy sets in its inputs; for IMOFM_M, it is the number of fuzzy sets in its inputs and output.

[*] Initial model extracted directly from data using clustering algorithms or grid partition methods.

[@] Refined model or the consequents are computed through the estimation methods.

[#] Simplified model after model simplification and parameter fine tuning.

[T] Total number of rule length.

σ^2 Stardard deviation of the results obtained from 30 runs.

One challenge associated with EAs-based multi-objective fuzzy modelling algorithms is how to include the results from different runs. This is because the algorithms of this type are stochastic in their nature. Different runs will lead to slightly different FRBS configurations. Hence, Table 5.3 also records the number of each FRBS' configuration found within the 30 runs using the whole three-stage modelling procedure. Most configurations are found more than 20 times within 30 runs, which suggests that the proposed modelling method is robust and consistent. It is worth mentioning at this stage that the FRBS with a short rule length was identified. This is mainly attributed to the merging of some fuzzy sets with the universal fuzzy set. The proposed method is also compared to other modelling approaches with singleton as their consequents, and it was found to represent the most accurate results with simpler rule-base structures. Although, Lin *et al.* (1997) used six rules and led to comparably good results, trapezoidal membership functions were used, which included more parameters to be tuned compared to the Gaussian membership functions used in the proposed work. In contrast, Wang *et al.* (2005) and Chen *et al.* (2004) adopted linear TSK structure. The reason for including these TSK modelling methods is that they are the representatives in terms of eliciting transparent FRBSs, and there are almost no similar efforts which have been made using Singleton or Mamdani modelling approaches. As can be seen from this table, these two methods produced slightly better predictions using fewer rules, e.g. four rules, compared to five rules in the proposed work. However, due to the linear combinations in the consequents of TSK models, the number of parameters involved in these two works and the proposed work is more or less the same. Apart from this, although linear TSK models generally use fewer rules and still provide better predictions, the linear combinations in the consequents are very hard to interpret in terms of linguistic terms, which more or less deviates from their original intentions of using MOP algorithm or model simplifications to elicit transparent FRBSs. However, the proposed method (both IMOFM_S and IMOFM_M) can provide interpretable rule-bases with comparable good predictions. No conclusion will be drawn regarding the comparisons of IMOFM_S and IMOFM_M at this point. Such comparisons are available in Section 6.5 after further results on different test problems are investigated.

Figures 5.17 and 5.18 show the Pareto fronts of the example using IMOFM_S and IMOFM_M from one of the 30 runs.

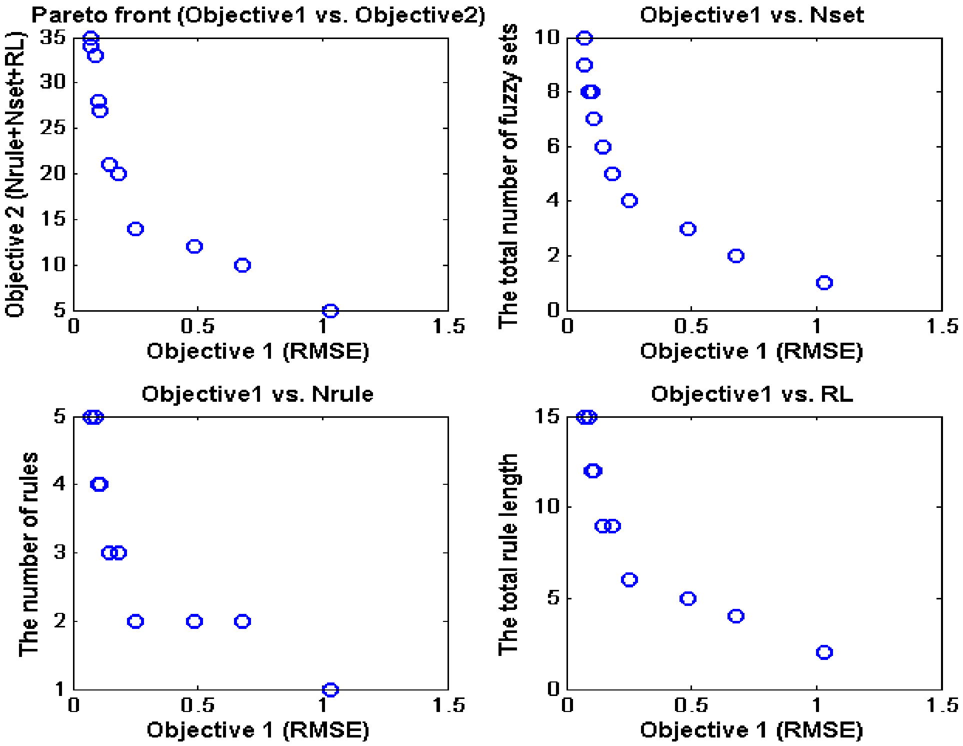


Figure 5.17 The Pareto fronts obtained using IMOFM_S from the third modelling procedure for the example: (a) Objective1 vs. Objective2; (b) Objective1 vs. Nset; (c) Objective1 vs. Nrule; (d) Objective 1 vs. RL.

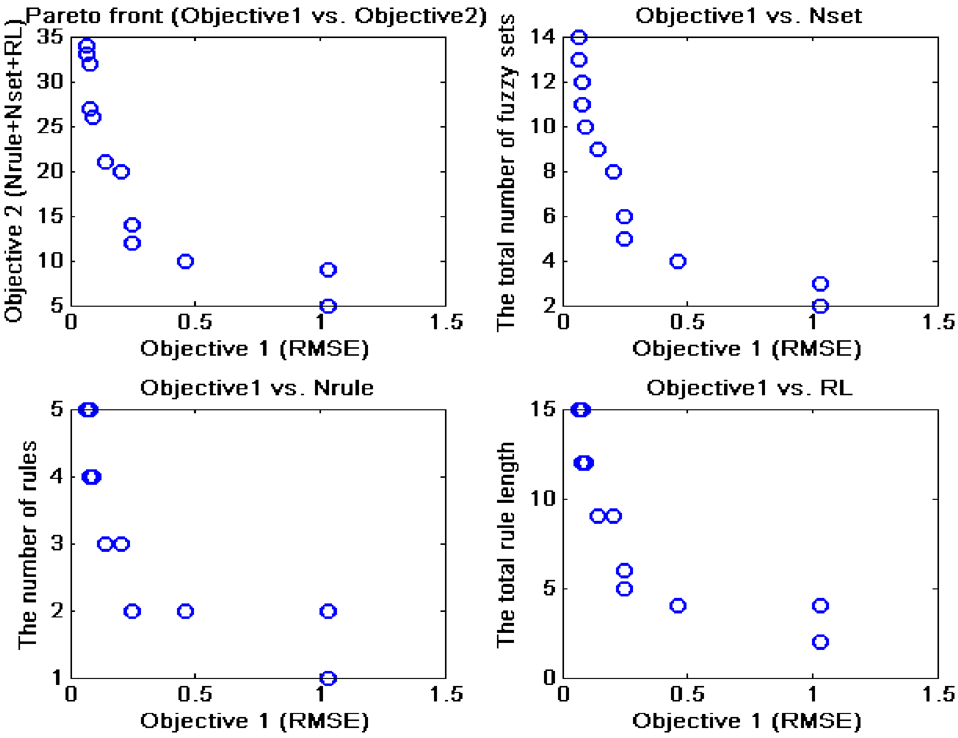


Figure 5.18 The Pareto fronts obtained using IMOFM_M from the third modelling procedure for the example: (a) Objective1 vs. Objective2; (b) Objective1 vs. Nset; (c) Objective1 vs. Nrule; (d) Objective 1 vs. RL.

In this project, the decision-making procedure was not explicitly investigated. A rather intuitive approach has been carried-out to inspect the Pareto fronts and to focus on each individual FRBS. In doing so, a 4-rule simplified FRBS with 7 fuzzy sets is chosen as a possible solution for IMOFM_S and a 4-rule simplified FRBS with 7 fuzzy sets in its inputs and 3 fuzzy sets in its consequents is chosen as a possible solution for IMOFM_M because of their acceptable predictive performances and their improved transparency. Given the limited and sparse data in this example, the results proved that the proposed modelling method has a good learning capability.

Figure 5.19 illustrates how the previously elicited two five-rule ‘vaccine FRBSs’ with highly overlapped membership functions (refer to Figures 5.6 and 5.7) are simplified to two 4-rule FRBSs with fewer interpretable fuzzy sets using IMOFM_S and IMOFM_M.

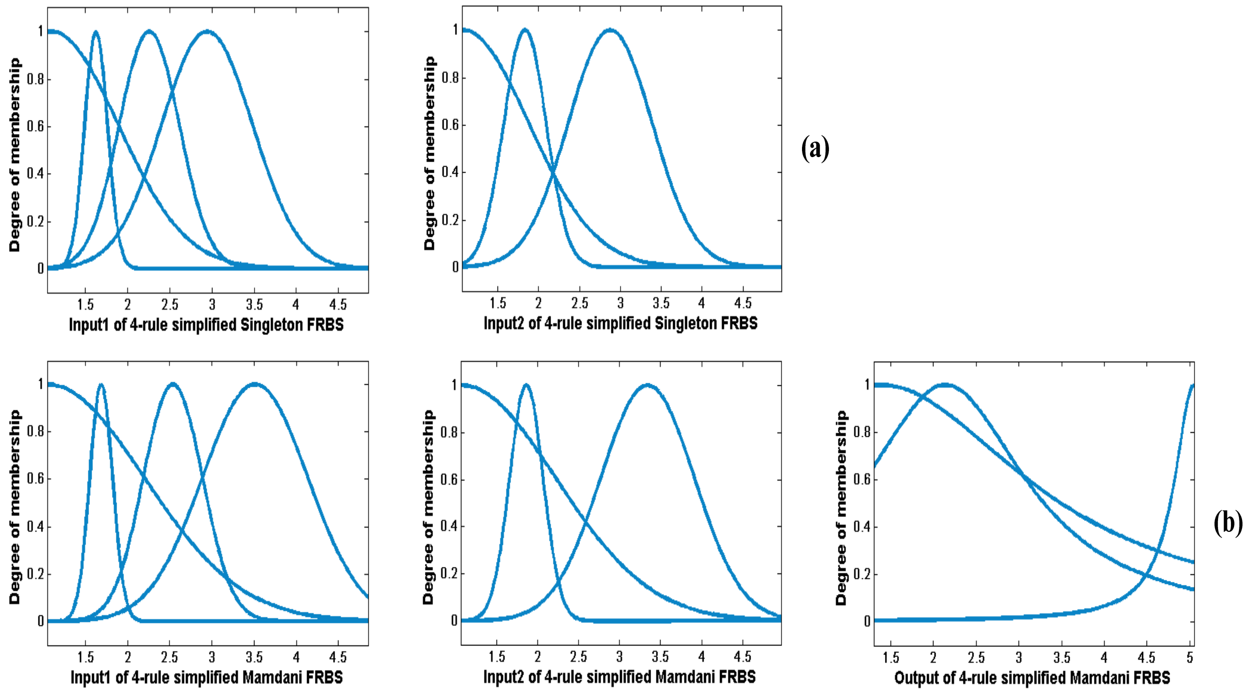


Figure 5.19 (a) membership function distribution of the 4-rule Simplified Singleton FRBS; (b) membership function distribution of the 4-rule simplified Mamdani FRBS.

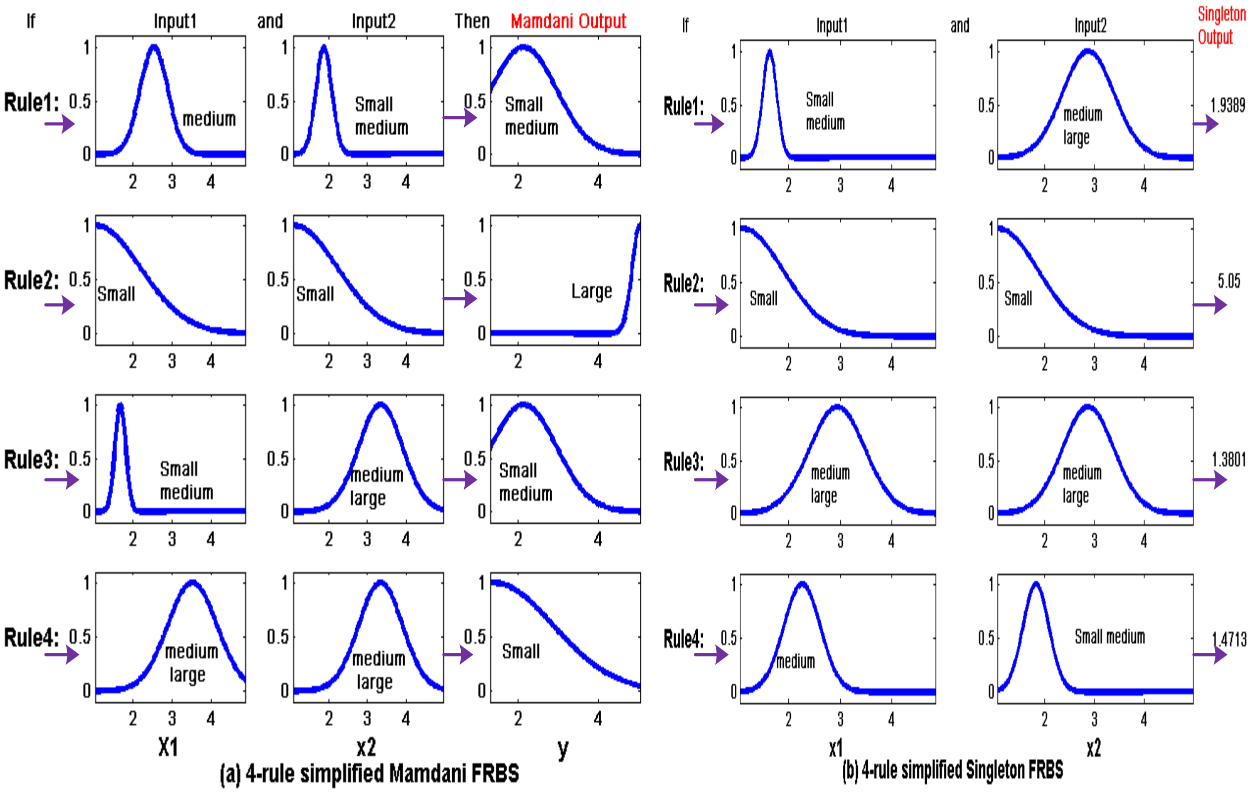


Figure 5.20 (a) 4-rule simplified Mamdani FRBS; (b) 4-rule simplified Singleton FRBS.

Figure 5.20 compares individual rules of the simplified 4-rule Singleton and Mamdani FRBS. As can be seen from Figure 5.20, although the rule-bases are extracted via different canonical forms, the knowledge expressed by such rule-bases is rather consistent. A closer investigation of consequents of the two simplified FRBSs reveals that, for IMOFM_M, due to the inclusion of fuzzy sets and the merging operations in its consequents, the simplified FRBS is more transparent than that elicited via IMOFM_S. Figure 5.21 shows the predictive performances of the simplified Singleton and Mamdani FRBSs by plotting their predicted outputs against the real outputs. Both FRBSs led to good predictions.

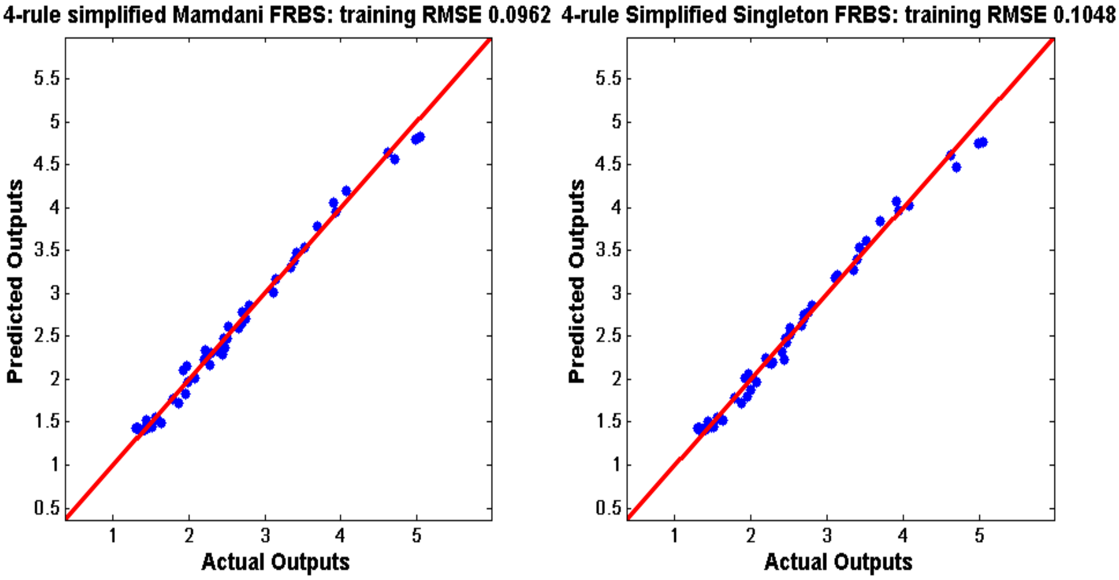


Figure 5.21 The predictive performances of the simplified Mamdani and Singleton FRBSs.

Figure 5.22 shows the 3-D surfaces of the simplified FRBSs using their inputs and outputs. As can be seen from this figure, the surfaces of both FRBSs are smooth over the definition ranges, which indicate that both IMOFM_S and IMOFM_M are good at interpolation. Comparing Figure 5.22 with Figure 4.19, it can be found that the predicted surfaces do not reproduce the original one perfectly even though the predicted outputs are very close to the real ones after the second and the third modelling stages. The reason behind this is that the data samples used in this experiment are not uniformly distributed so that not all the regions are reflected in the collected data. Data-driven modelling cannot address such extrapolation problem unless there are data represented specifically in such regions. However, with the limited and sparse data given by this example, this result does show that the proposed modelling method has a good learning capability using the given data.

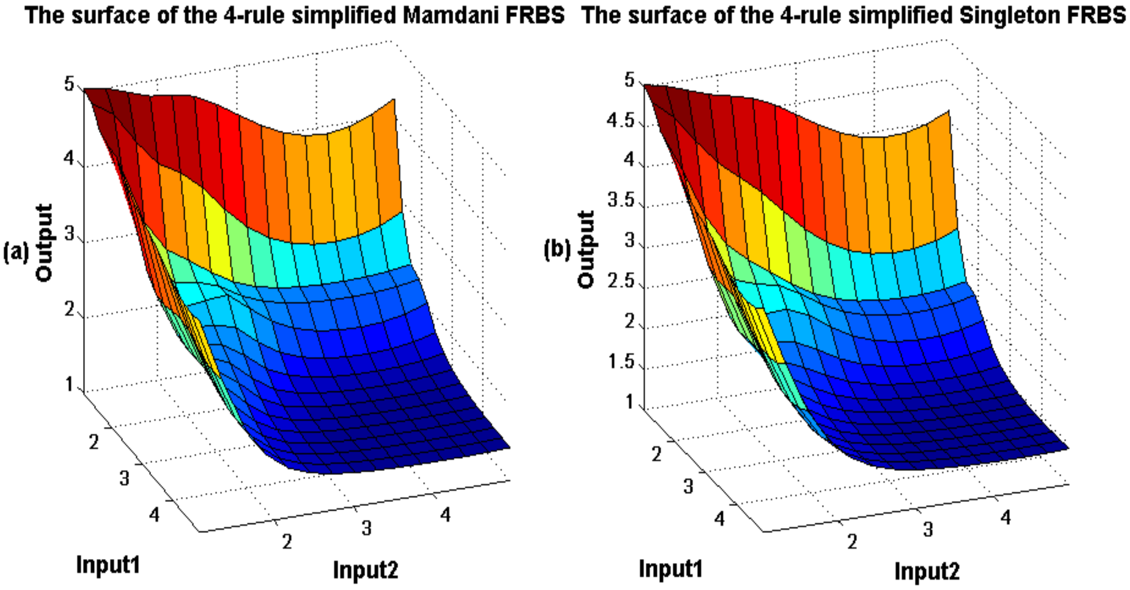


Figure 5.22 3-D surfaces of the simplified Mamdani and Singleton FRBSs.

Finally, the Pareto fronts obtained using IMOFM_S and IMOFM_M over 30 runs are given in Figures 5.23 and 5.24 as the complement to Table 5.3. Figures 5.23 and 5.24 reinforce the observation made earlier: although IMOFM is a stochastic algorithm, it is robust since most solutions found during different runs are very similar.

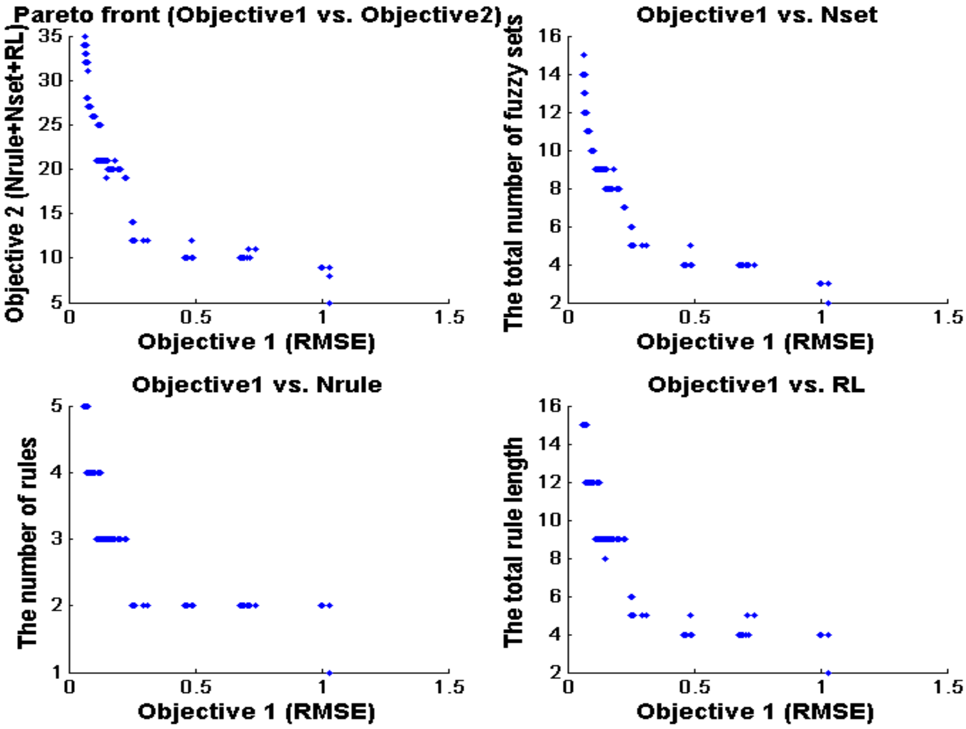


Figure 5.23 The Pareto fronts obtained using IMOFM_M over 30 runs.

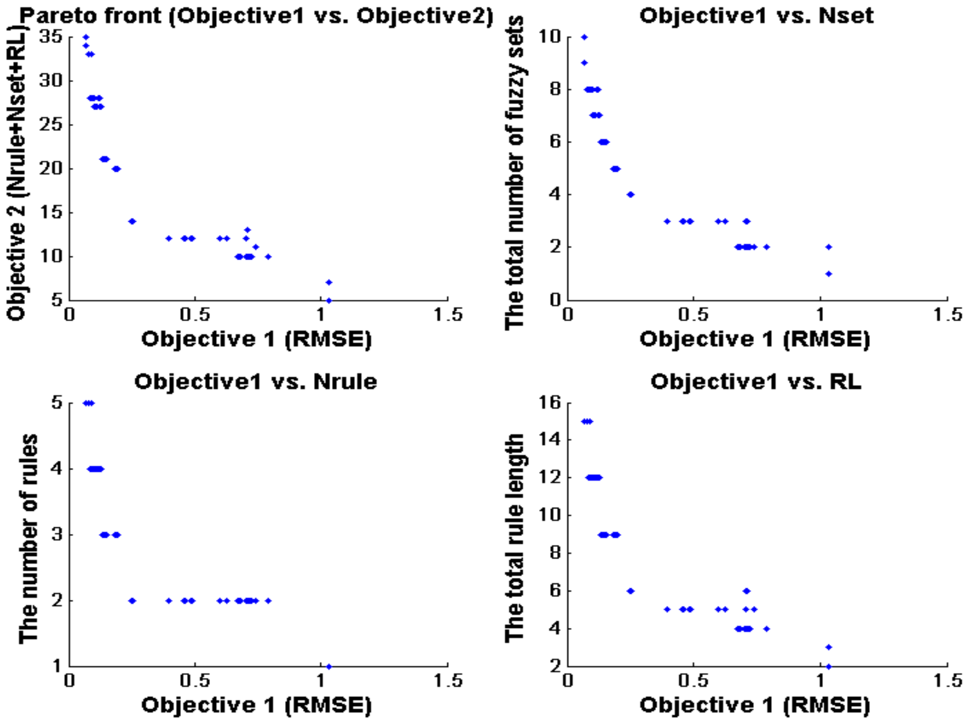


Figure 5.24 The Pareto fronts obtained using IMOFM_S over 30 runs.

5.6 Analysis of the Proposed Modelling Method

5.6.1 Influence of the Modelling Stages on Performance

In order to test the influences of the first two stages, two variants of the proposed modelling scheme are investigated:

1) The combination of the first stage and the third stage;
2) Only the third stage.

In the first case, an initial 5-rule FRBS is generated using G3Kmeans, which is then fed to the third stage without any refinement. While in the latter case, the initial 5-rule FRBS is randomly generated within the variable domains. Table 5.4 summarises the results of the two variants. Only the results obtained by IMOFM_S are presented in this Section. However, the observations made in this section are always held for IMOFM_M, unless otherwise stated.

TABLE 5.4

COMPARISONS OF THE PREDICTIVE PERFORMANCE OF THE DIFFERENT MODELING STAGES FOR THE EXAMPLE

Modeling Methods (Ref.)	No. of rules	No. of fuzzy sets	No. of Parameters	Consequents	Performance (RMSE training)
IMOFM_S (the first stage and the third stage), NB: the results of one random run are presented here					
Execution time (Intel(R) Core(TM)2 Duo CPU, 2.27 GHz): 1st stage : 0.41sec; 3rd stage: 415sec; numbeer of iterations: 3000					
Initial FRBS	5	10 Gaussian	25	Singleton	0.6069[*]
Pareto FRBS1	5	6 Gaussian	17	Singleton	0.1183[#]
Pareto FRBS2	4	6 Gaussian	16	Singleton	0.1268[#]
Pareto FRBS3	3	5 Gaussian	13	Singleton	0.1724[#]
Pareto FRBS4	2	4 Gaussian	10	Singleton	0.2475[#]
Pareto FRBS5	2(4[T])	2 Gaussian	6	Singleton	0.7235[#]
Pareto FRBS6	1	1 Gaussian	3	Singleton	1.0326[#]
IMOFM_S (only the third stage), NB: the results of one random run are presented here					
Execution time (Intel(R) Core(TM)2 Duo CPU, 2.27 GHz): 3rd stage: 423sec; number of iterations: 4000					
Initial FRBS	5	10 Gaussian	25	Singleton	1.0363[*]
Pareto FRBS1	4	7 Gaussian	18	Singleton	0.1116[#]
Pareto FRBS2	4	6 Gaussian	16	Singleton	0.1223[#]
Pareto FRBS3	3	5 Gaussian	13	Singleton	0.1502[#]
Pareto FRBS4	3(8[T])	4 Gaussian	11	Singleton	0.1753[#]
Pareto FRBS6	3(7[T])	4 Gaussian	11	Singleton	0.3211[#]
Pareto FRBS5	3(6[T])	4 Gaussian	11	Singleton	0.3252[#]

[*] Initial model extracted directly from data using clustering algorithms or grid partition methods.

[#] Simplified model after model simplification and parameter fine tuning;

[T] Total number of rule length.

It is worth mentioning that for the two variants, the thresholds for the model simplification are set at higher values so that the FRBSs with more rules are given a better chance of surviving in the early stages of the evolution, otherwise, FRBSs with more rules may be replaced by FRBSs with fewer rules since both of them are inaccurate in the early iterations. In such a case, the ‘Pareto’ selection process favours the one with fewer rules. This problem has already been solved by Jiménez *et al.* (2001) using a ‘niche’ concept, where each niche maintains a set of FRBSs with the same number of rules. A substitution only happens within each niche so that one can evolve a set of FRBSs with the different number of rules without the worry of losing individuals with more rules. However, the thresholds are not important parameters if one chooses to use the whole three-stage modelling procedure. The proposed three-stage procedure does not need the aforementioned ‘niche’ concept if all the stages work as a unified procedure. In such a case, the most accurate FRBS is always the one with the number of rules close to the maximum value. More importantly, this accurate FRBS will direct the search from the most complex structure (the more accurate one) to the simplest ones (the less accurate ones). This ensures the coexistence of FRBSs with various complexities during the ‘Pareto’ selection.

As can be seen from Table 5.4, more iterations and more time are needed for the two variants to achieve a similar predictive performance as that obtained using the three-stage modelling procedure. This is because that the two variants have to evolve from a totally random stage (inaccurate individuals). Only a few Pareto FRBSs are obtained compared to the ones elicited via the three-stage procedure. The most complex structure which is supposed to evolve to the most accurate FRBS is discarded during the optimisation for the reasons described above. In terms of the predictive performance of the evolved 'Pareto' FRBSs, one can always find the counterparts within the proposed modelling schemes and its variants. However, the variants may lose the chance of evolving into the most accurate (the most complex) FRBS. All these justified the inclusion of the first two stages.

5.6.2 Influence of the User-Specified Parameters on Performance

It is a common phenomenon that an EA-based algorithm includes a number of user-specified parameters and IMOFM is not an exception. It includes a set of user-defined parameters, among which some are inherited from PAIA2 (referred to Section 3.4) and others are mainly associated with the third modelling stage. Hence, in this Section, the investigations on how these parameters affect the performance of IMOFM are carried out. The emphasis has been given to two PAIA2 affiliated parameters, namely the initial population size and the network suppression threshold, and five model-simplification parameters, namely p_m, rdr, mr, ufs and sfs.

In Section 3.4.1, it was concluded that the initial population size is not a critical parameter as far as PAIA2 is concerned. This parameter has indeed an impact on the speed of the algorithm's convergence. However, given enough evaluation times it has been shown that the initial population size and the accuracy of PAIA2 have no causal relationship. In order to confirm that such a fact is still applicable in the case of IMOFM, a series of experiments are conducted with the initial population size varied from 1 to 10. Other parameters are kept the same as those used in previous sections.

The results shown in Figure 5.25 are the average values of 10 independent runs, each of which executes 1000 iterations (which are considered as enough evaluation times) using IMOFM_S.

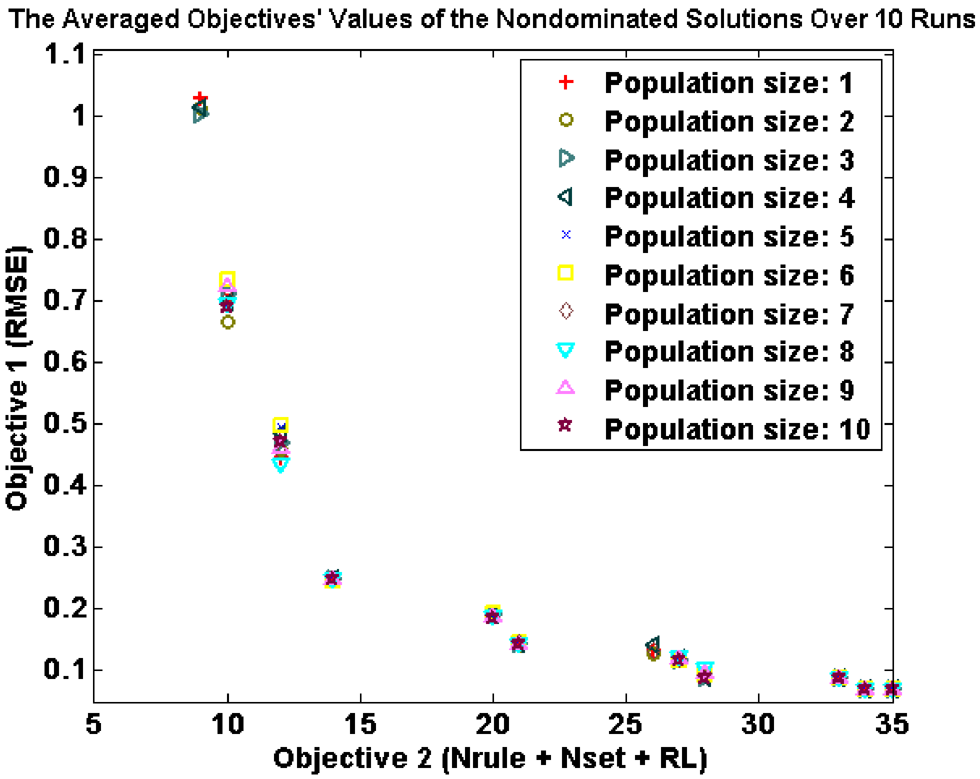


Figure 5.25 The averaged objectives' values of the non-dominated solutions found in 10 independent runs with their initial population sizes varied from 1 to 10.

As one can see from Figure 5.25, the non-dominated FRBSs with different initial population sizes produced very close Pareto fronts, which means that the initial population size is not a critical parameter for IMOFM just as the conclusion made in Section 3.4.1 for PAIA2.

Figure 5.26 shows the averaged predictive performances of different Pareto FRBSs found in the 10 runs with their initial population sizes varied from 1 to 10 (marked as the red squares). As can be seen from Figure 5.26, the Pareto FRBSs with different initial population size have only a small variance in terms of their predictive performances. It is because of this property that the 'initial population pool forming method' proposed in Section 5.5.2 gains its legitimacy. Hence, the conclusions made in the previous sections and to be made in the subsequent sections using 7 as the initial population size are sufficient without any loss of generality.

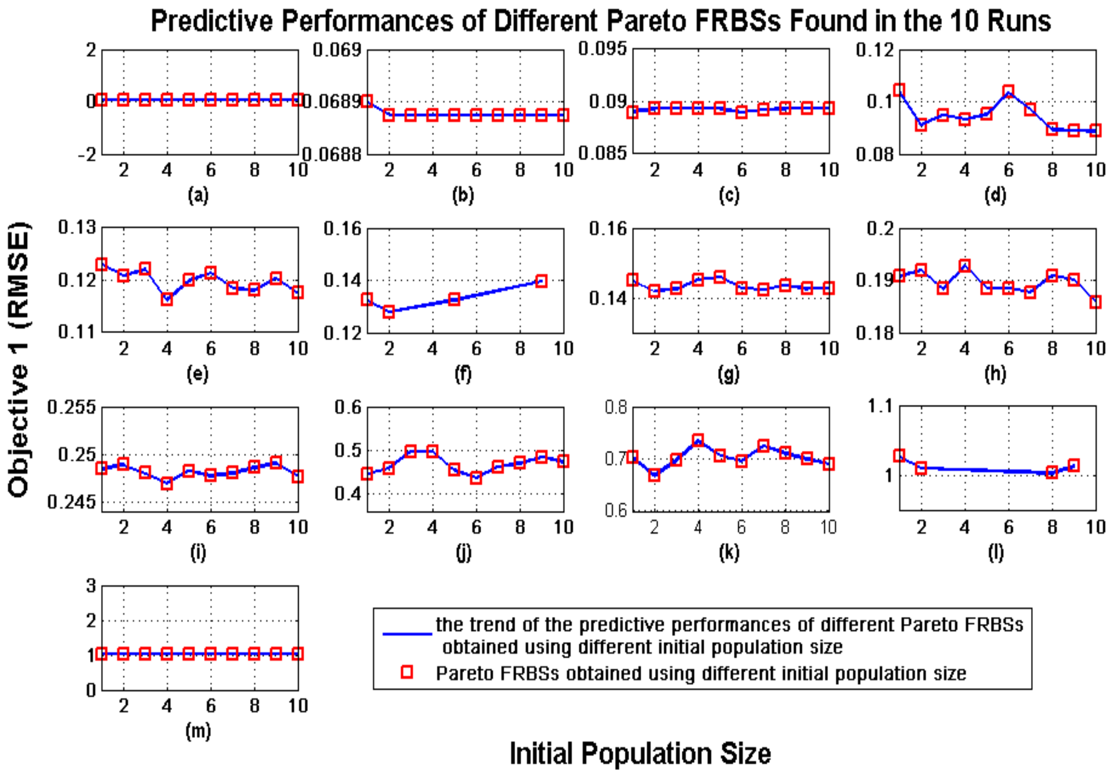


Figure 5.26 The averaged predictive performances of different Pareto FRBSs found in the 10 runs: (a) 5-rule FRBS with 10 fuzzy sets; (b) 5-rule FRBS with 9 fuzzy sets; (c) 5-rule FRBS with 8 fuzzy sets; (d) 4-rule FRBS with 8 fuzzy sets; (e) 4-rule FRBS with 7 fuzzy sets; (f) 4-rule FRBS with 6 fuzzy sets; (g) 3-rule FRBS with 6 fuzzy sets; (h) 3-rule FRBS with 5 fuzzy sets; (i) 2-rule FRBS with 4 fuzzy sets; (j) 2-rule FRBS with 3 fuzzy sets; (k) 2-rule FRBS with 2 fuzzy sets; (l) 2-rule FRBS with 1 fuzzy sets; (m) 1-rule FRBs with 1 fuzzy sets.

In Section 3.4.3, it was concluded that the network suppression threshold is not a critical parameter as far as the convergence accuracy is concerned. However, it is an important factor as far as the number of the obtained Pareto solutions is concerned. Without the need for increasing the evaluation times greatly, more Pareto solutions may be obtained by simply adjusting the network suppression threshold. This property is regarded as one of the advantages of PAIA2 comparing to other MOEAs in Section 3.6.2. Whether this property is still held for IMOFM will be investigated next. Figure 5.27 shows the number of Pareto FRBSs obtained using 0.1 and 0.0008 as their network suppression thresholds. The results are obtained over 10 independent runs (marked as the red squares in Figure 5.27), each of which executes 1000 iterations.

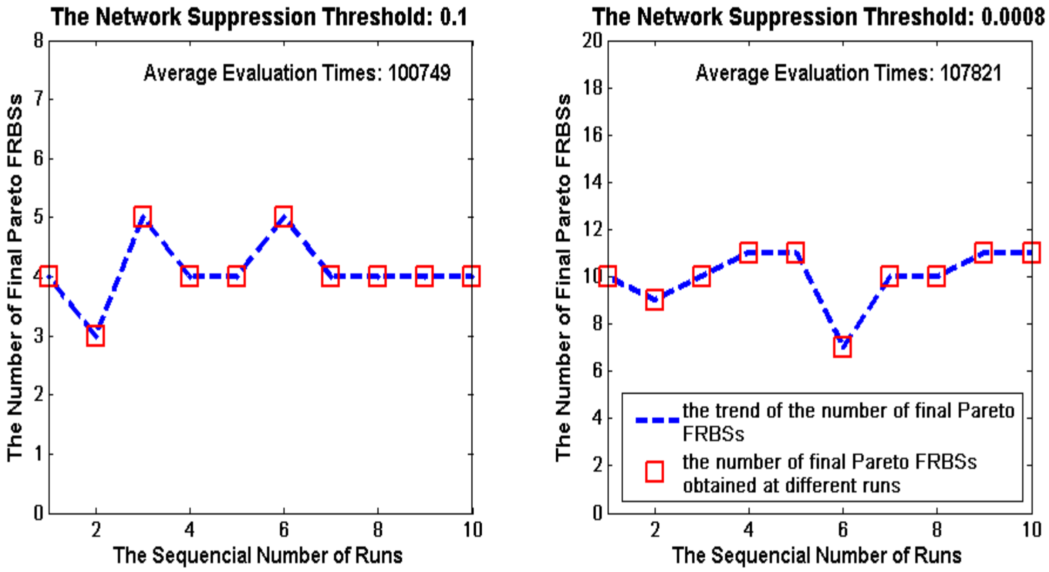


Figure 5.27 The number of Pareto FRBSs obtained using 0.1 (right) and 0.0008 (left) as the network suppression threshold.

As can be seen from Figure 5.27, different network suppression thresholds did affect the final number of Pareto solutions. Without greatly increasing the evaluation times, the number of Pareto FRBSs has increased from around 4 (in which case, 0.1 is the network suppression threshold) to 10 (in which case, 0.0008 is the network suppression threshold).

Figure 5.28 shows the Pareto fronts obtained using two different network suppression thresholds from 1 of 10 runs. A bigger value of the threshold means more Pareto solutions will be suppressed. A Smaller value of the threshold means more Pareto solutions will be allowed to enter into the memory set during each iteration step; hence, more Pareto FRBSs are obtained in the final population set. Using a smaller threshold reduces slightly the evaluation times; however, using fewer evaluation times will not compromise the accuracy of the algorithm. Hence, the choice of the network suppression threshold is problem-dependent and is subject to specific requirements. Since this parameter does not affect the predictive performance of the elicited model, the conclusions made in the previous sections and to be made in the subsequent sections with 0.0008 are sufficient.

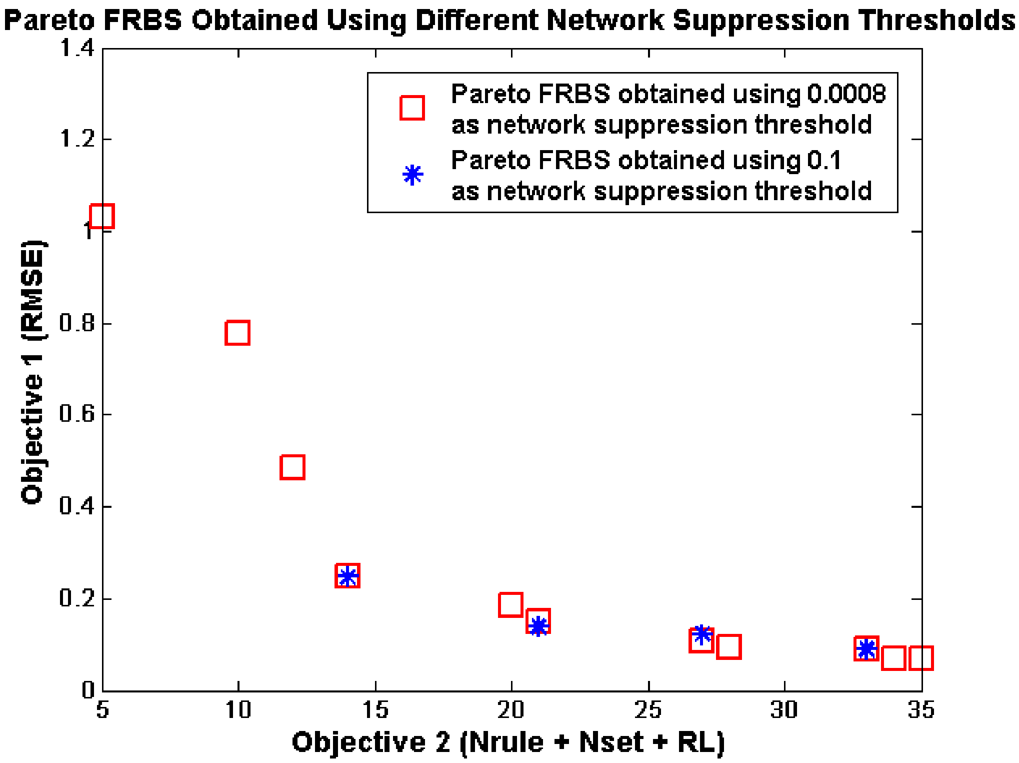


Figure 5.28 Pareto FRBS obtained using different network suppression thresholds from one of 10 runs.

In order to investigate the effects of the model-simplification parameters, only one parameter is selected each time with its value varied from 0 to 1. Other parameters are kept as constant so that one can concentrate on analysing the effect of the selected parameter.

The first parameter to be investigated is p_m which is responsible for removing insignificant rules. p_m can vary from 0 to 1, in which 1 means no rules can be regarded as insignificant rules and 0 means the opposite. Any value between 0 and 1 means given enough evaluation times a proportion of rules will be regarded as insignificant rules. In order to bypass the effects of other model simplification parameters, rdr, mr, ufs and sfs are set to 0, 1, 1 and 1 respectively.

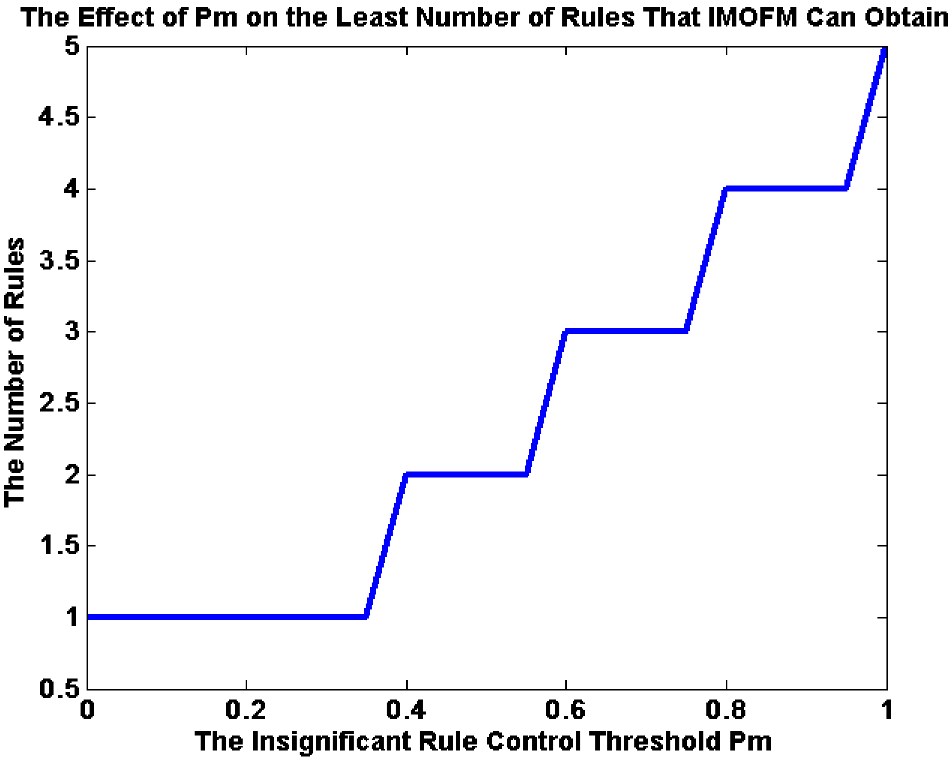


Figure 5.29 The effect of p_m on the least number of rules that IMOFM can obtain.

Figure 5.29 shows how p_m affects the least number of rules that IMOFM can obtain. According to Eq. 5.22, if p_m equals to 1, all rules will be kept in the rule base; while, if p_m equals to 0, IMOFM will have the chance to find the simplest rule base with the number of rules being 1. If p_m takes values in between, IMOFM will find the rule bases with their number of rules varied from 1 to 5.

Figure 5.30 shows the Pareto front obtained using $p_m = 0.5$. Table 5.5 summarises the predictive performances of Pareto FRBS with $p_m = 0.5$. As one can see from this graph and Table 5.5, predictive accuracy-wise, they are very similar as those shown in Figure 5.17 and Table 5.3. However, without the involvement of other model simplification parameters, only a smaller number of Pareto FRBS can be obtained. This is because no singleton rules can be deleted, and no similar rules or fuzzy sets can be merged unless they are exactly the same.

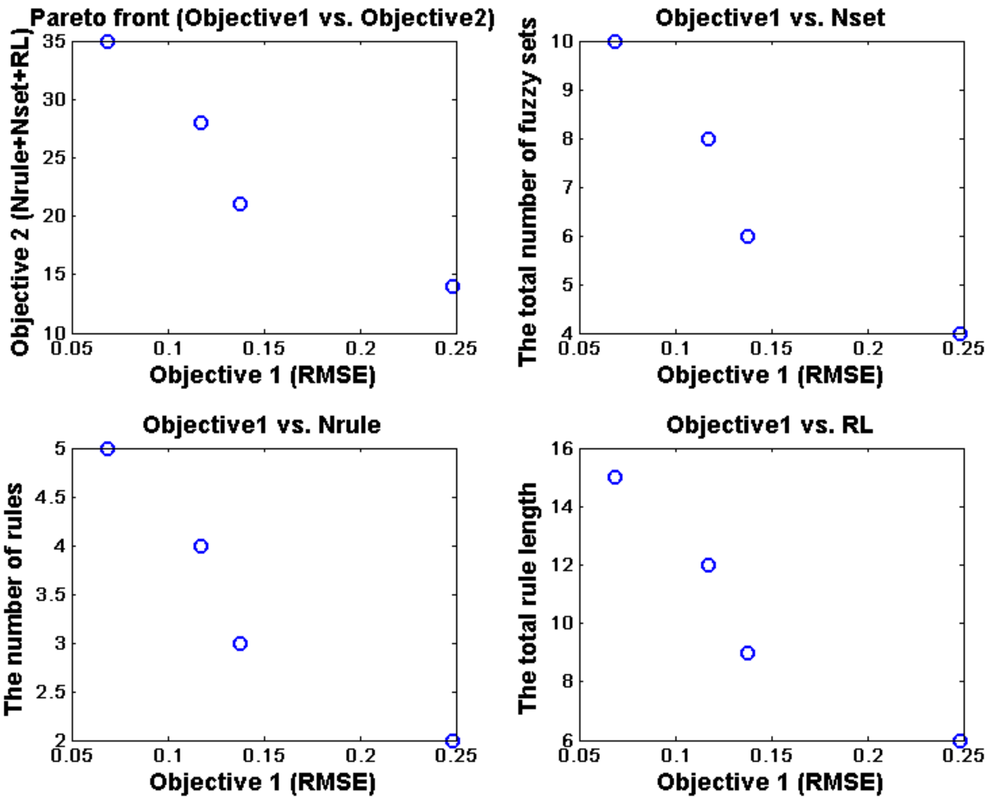


Figure 5.30 The Pareto front obtained using IMOFM_S with $p_m = 0.5$, $rdr = 0$, $mr = 1$, $ufs = 1$ and $sfs = 1$.

TABLE 5.5

THE PREDICTIVE PERFORMANCES OF IMOFM_S WITH $p_m = 0.5$,$rdr = 0$, $mr = 1$, $ufs = 1$ AND $sfs = 1$

Pareto FRBS	No. of rules	No. of fuzzy sets in inputs	No. of Parameters	Consequents	Performance (RMSE training)
Pareto FRBS 1	5	10	25	Singleton	0.06873
Pareto FRBS 2	4	8	20	Singleton	0.11697
Pareto FRBS 3	3	6	15	Singleton	0.13735
Pareto FRBS 4	2	4	10	Singleton	0.24818

Hence, p_m controls the least number of rules. It is worth mentioning that such a number is also affected by other model simplification parameters. Hence, the least number of rules may not be strictly specified by p_m. p_m is set to 0.5 so that half of the initial rules can be regarded as insignificant rules given enough evaluation times in this work without any loss of generality.

The next model-simplification parameter under investigation is the threshold rdr which controls the deletion of singleton rules. rdr is varied from 0 to 1 with $p_m = 0.5$, $mr = 1$,

$ufs = 1$, $sfs = 1$, and all other parameters are kept the same as those used in the previous sections.

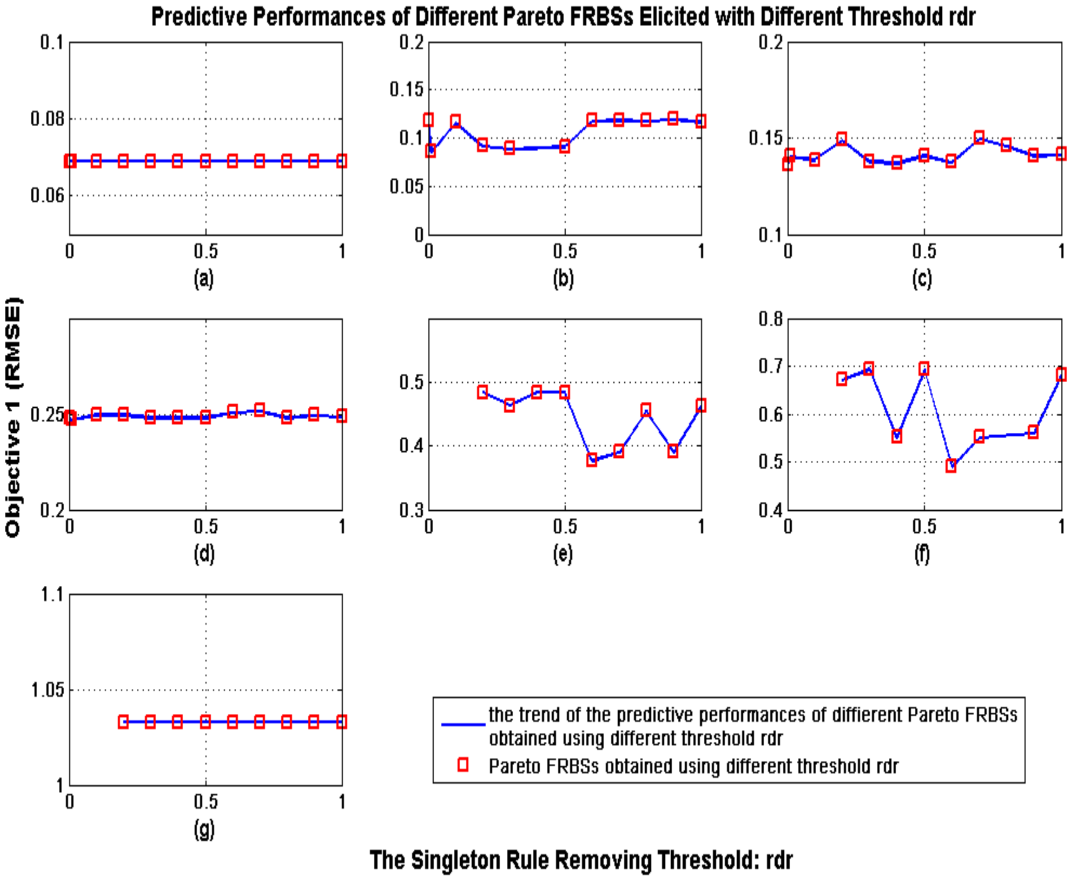


Figure 5.31 The predictive performances of different Pareto FRBSs elicited with different rdr values: (a) 5-rule FRBS with 10 fuzzy sets; (b) 4-rule FRBS with 8 fuzzy sets; (c) 3-rule FRBS with 6 fuzzy sets; (d) 2-rule FRBS with 4 fuzzy sets; (e) 2-rule FRBS with 3 fuzzy sets; (f) 2-rule FRBS with 2 fuzzy sets; (g) 1-rule FRBS with 1 fuzzy sets.

Figure 5.31 indicates that rdr is not a critical parameter as far as the predictive performance of the elicited FRBS is concerned. However, a small value of rdr reduces the chances of a fuzzy rule being considered as a singleton rule (as shown in Figure 5.31, IMOFM with rdr smaller than 0.2 cannot find (e), (f) and (g)). As mentioned in Section 5.5.4.2, a singleton fuzzy rule may represent exceptions. Hence, one should keep this threshold as a small value in case that useful fuzzy rules are deleted. As a general guideline, any values between 0 and 0.2 may represent an adequate choice for rdr. Hence, in this research work, rdr is set to 0.01 without any loss of generality.

In order to investigate the effect of the threshold mr which is responsible for merging similar rules, mr is varied from 0 to 1 with $p_m = 0.5$, $rdr = 0.01$, $ufs = 1$, $sfs = 1$, and all other parameters are kept the same as those used in the previous sections. Figure 5.32 clearly demonstrates that a small value of mr encourages more rules to be considered as similar rules and thus to be merged. Hence, with a smaller value of mr, more Pareto FRBSs and FRBS with fewer rules are expected in the final solution set.

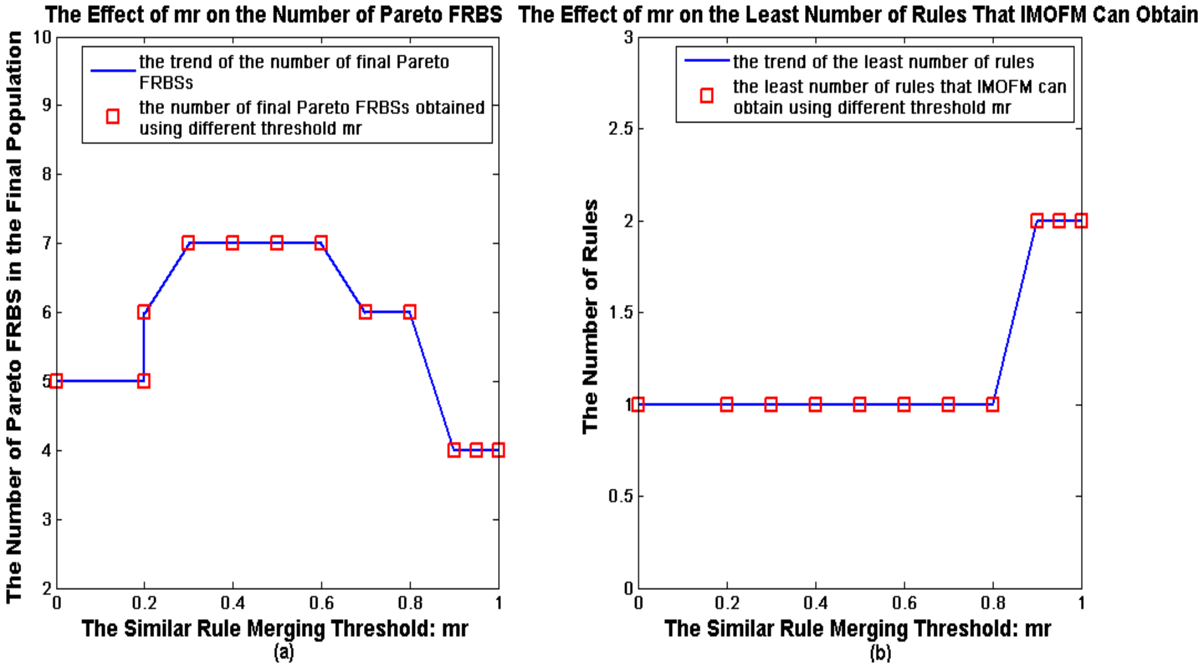


Figure 5.32 The effects of mr on: (a) the number of Pareto FRBS in the final population; (b) the least number of rules that IMOFM can obtain.

However, one cannot conclude just from Figure 5.32 that a smaller value of mr is more preferable. With $ufs = 1$ and $sfs = 1$, the merging of similar fuzzy sets and the deletion of fuzzy sets similar to the universal fuzzy set are disabled. In such a situation, a small value of mr will result in rules being merged even if they are quite different, and this may ultimately affect the predictive performances of the elicited FRBS.

Figure 5.33 shows the predictive performances of four Pareto FRBSs against different values of mr. As can be seen from Figure 5.33, IMOFM with mr varied between 0.9 and 1 generally produces more accurate predictions. More importantly, if the operations of merging similar fuzzy sets and deleting universal fuzzy sets are added in, even with a large value of mr, IMOFM is able to find Pareto FRBS with fewer rules (which means more Pareto FRBS can be found). Hence, mr with a large value, e.g. between 0.9 and 1, is a preferable choice as far as the predictive accuracy is concerned. This parameter is set to 0.95 in the previous and subsequent sections without any loss of generality.

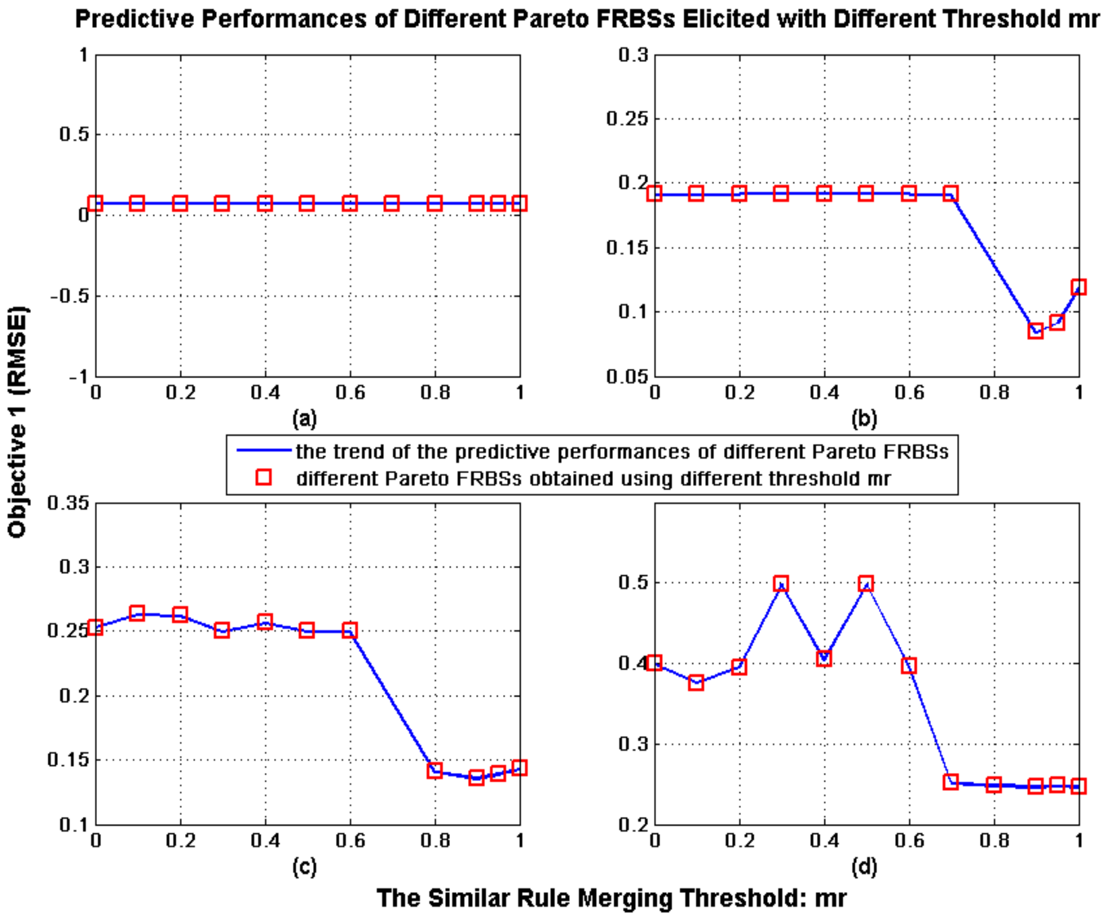


Figure 5.33 The predictive performances of different Pareto FRBSs elicited with different mr values: (a) 5-rule FRBS with 10 fuzzy sets; (b) 4-rule FRBS with 8 fuzzy sets; (c) 3-rule FRBS with 6 fuzzy sets; (d) 2-rule FRBS with 4 fuzzy sets.

ufs is the threshold which decides when a fuzzy set can be regarded as the universal fuzzy set and is thus deleted. A small value of this threshold means more fuzzy sets can be considered as the universal fuzzy set, and vice versa. If $p_m = 0.5$, $rdr = 0.01$, $mr = 0.95$, $sfs = 1$, and all other parameters are kept the same as those used in the previous sections, a small value of ufs will result in more Pareto FRBS to be found since in such a case the probability of having similar rules are increased due to the deletion of some fuzzy sets. Hence, IMOFM has more probability of finding FRBS with fewer rules. Figure 5.34 demonstrates such an effect when ufs varied from 0 to 1.

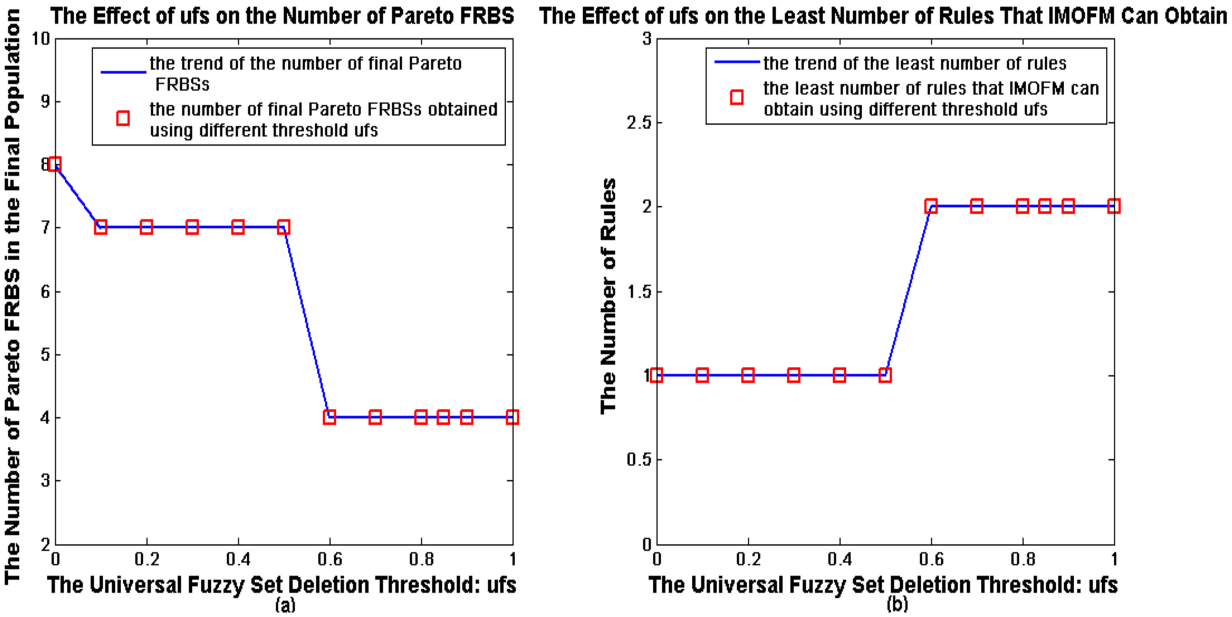


Figure 5.34 The effects of ufs: (a) on the number of Pareto FRBS in the final population; (b) on the least number of rules that IMOFM can obtain.

However, similarly to the conclusion made about mr, a small value of ufs does not necessarily mean that it represents a good choice. Useful fuzzy sets may be deleted just because they are ‘wide enough’ to be considered as the universal fuzzy set in the case of using a small value of ufs. In such a case, the predictive performances of the elicited FRBS may be affected.

Figure 5.35 demonstrates how different ufs affects the predictive performances. Generally speaking, values between 0.6 and 1 are good values. As can be seen in the next paragraph, if sfs is set to values smaller than 1, then the values of ufs will not make any difference in terms of the number of Pareto FRBS that IMOFM can find. Hence, ufs is set 0.85 in this work without any loss of generality.

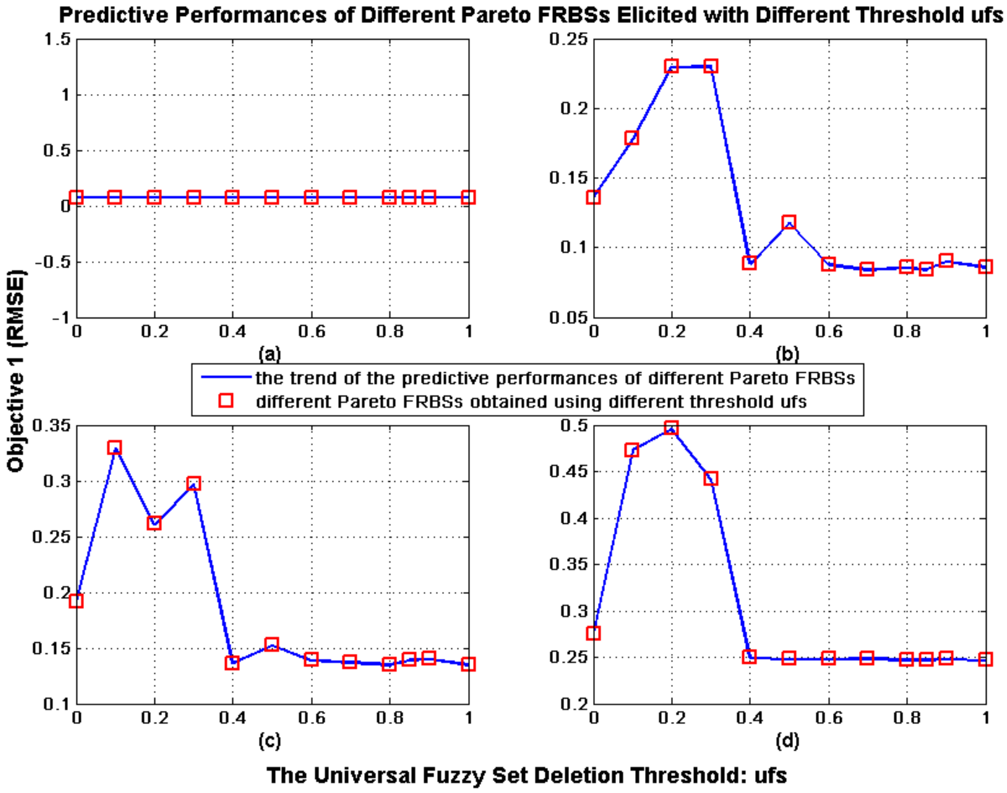


Figure 5.35 The predictive performances of different Pareto FRBSs elicited with different ufs values: (a) 5-rule FRBS with 10 fuzzy sets; (b) 4-rule FRBS with 8 fuzzy sets; (c) 3-rule FRBS with 6 fuzzy sets; (d) 2-rule FRBS with 4 fuzzy sets.

The last model-simplification parameter to be investigated is sfs. Intuitively, this parameter should be set to values smaller than 1 so that not only the same fuzzy sets but also similar ones can be merged. When similar fuzzy sets are merged, rules may become similar so that they will be merged consequently, a consideration which makes the rule-base more compact even with other parameters set to the aforementioned values. Hence, in the part, sfs is varied from 0 to 1, with $p_m = 0.5$, $rdr = 0.01$, $mr = 0.95$, $ufs = 0.85$, and all other parameters are kept the same as those used in the previous sections.

Figure 5.36 confirms that when sfs takes values smaller than 1 IMOFM can produce more Pareto FRBSs and the FRBS with a more compact structure.

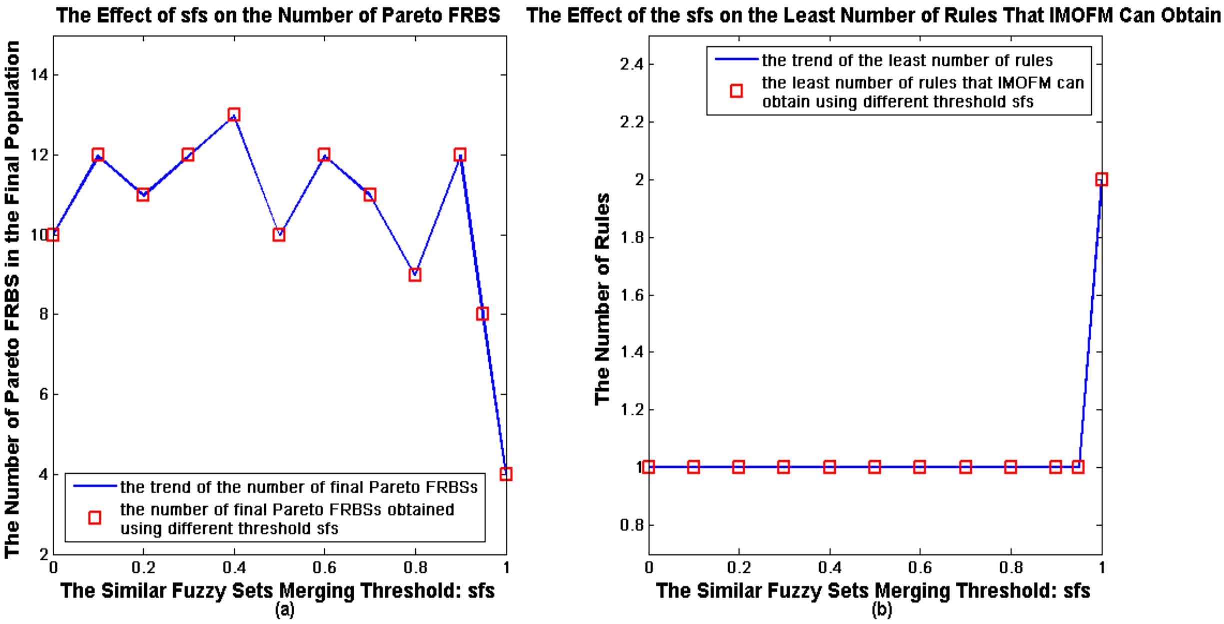


Figure 5.36 The effects of sfs: (a) on the number of Pareto FRBS in the final population; (b) on the least number of rules that IMOFM can obtain.

Figure 5.37 shows the predictive performances of different Pareto FRBSs elicited with different sfs. As can be seen from this figure, not all Pareto FRBS configurations can be found by different values of sfs, e.g. results of Figure 5.37 (d) cannot be found using sfs with values smaller than 0.9. However, as long as sfs does not equal to 1, most FRBS configurations can be found via various values of sfs. Hence, the emphasis here is rather placed on evaluating the predictive accuracy of the elicited FRBSs so that a generally guideline of this parameter can be derived. As can be seen from Figures 5.37 (e), (f), (h), (i), (n) and (p), values between 0.9 and 1 normally give more accurate FRBS than using a smaller value. Hence, sfs is set to 0.95 in this work without any loss of generality.

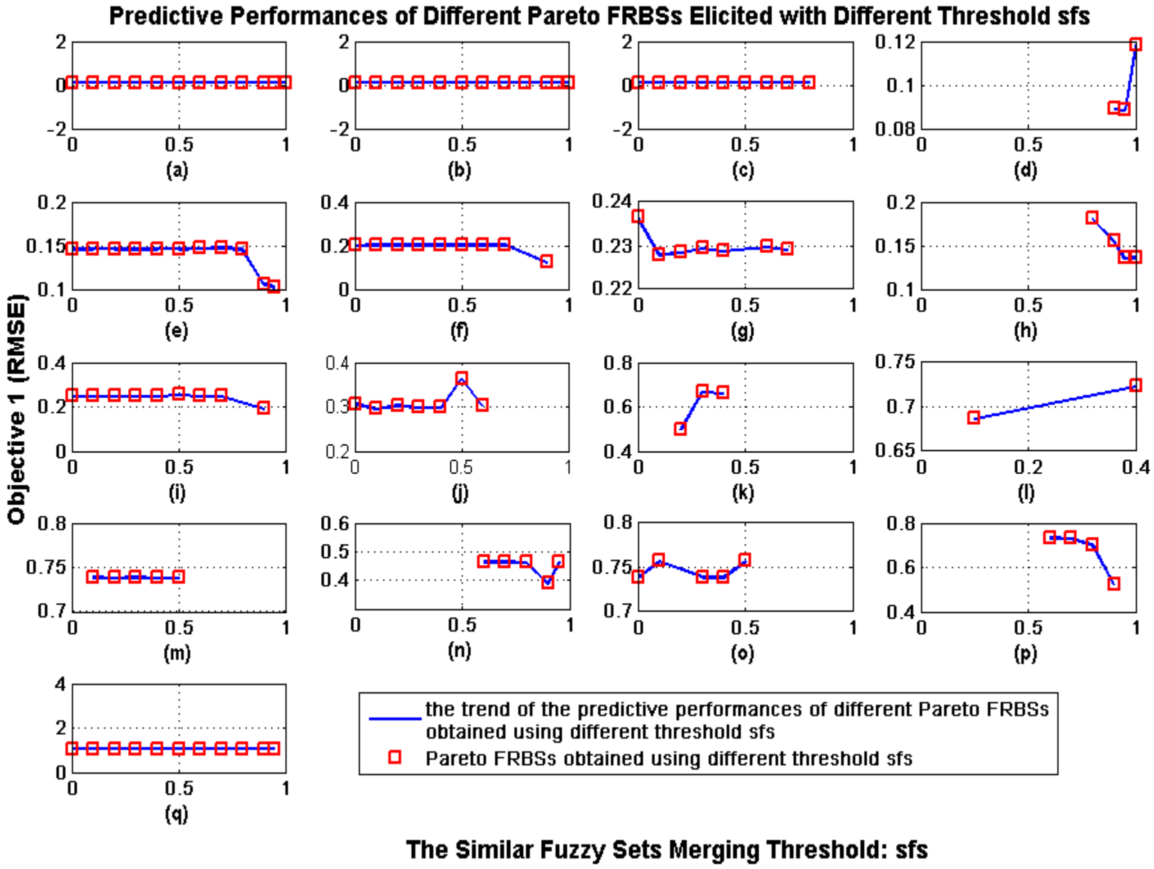


Figure 5.37 The predictive performances of different Pareto FRBSs found in the 10 runs with different sfs values: (a) 5-rule FRBS with 10 fuzzy sets; (b) 5-rule FRBS with 9 fuzzy sets; (c) 5-rule FRBS with 8 fuzzy sets; (d) 4-rule FRBS with 8 fuzzy sets; (e) 4-rule FRBS with 7 fuzzy sets; (f) 4-rule FRBS with 6 fuzzy sets; (g) 4-rule FRBS with 5 fuzzy sets; (h) 3-rule FRBS with 6 fuzzy sets; (i) 3-rule FRBS with 5 fuzzy sets; (j) 3-rule FRBS with 4 fuzzy sets ; (k) 3-rule FRBS with 5 fuzzy sets (objective 2: 18); (l) 3-rule FRBS with 3 fuzzy sets; (m) 2-rule FRBS with 4 fuzzy sets; (n) 2-rule FRBS with 3 fuzzy sets; (o) 2-rule FRBs with 2 fuzzy sets; (p) 2-rule FRBS with 2 fuzzy sets (objective 2 : 10); (q) 1-rule FRBS with1 fuzzy set.

5.6.3 Influence of the Variable Length Coding Scheme on Performance

In order to investigate the influence of the variable length coding scheme on model predictive performance, two experiments are conducted. The first experiment consists of utilising the proposed three-stage modelling procedure with the new distance index proposed in Section

5.5.3 being replaced with the original distance measure proposed in Section 3.2.1 so that one can obtain a direct comparison of the modelling approaches with and without the variable length coding scheme. The second experiment consists of recording the approximate Pareto fronts found during the search process. By plotting such progress one can also explore how the proposed modelling method with the variable length coding scheme can cope with the simultaneous optimisation of the rule-base structure and its parameters.

Table 5.6 summarises the results of the first experiment. All the user-defined parameters of both modelling methods, viz. with and without the variable length coding scheme, are kept as those mentioned in the last Section. The results are the average values of 15 independent runs.

TABLE 5.6
THE COMPARISON OF THE MODELING APPROACHES WITH AND WITHOUT VARIABLE LENGTH CODING SCHEME

Modeling Approach	**Fuzzy Models**	**The Number of Fuzzy Sets in Each Inputs**	**Performance (RMSE training)**	**Improvement (%)**
IMOFM_S **(without the variable Length coding scheme)**	Pareto FRBS 1	4 rules: [4 3]	0.1261	-
IMOFM_S **(with the variable length coding scheme**	Pareto FRBS 1	4 rules: [4 3]	0.1198	5%

As can be seen from the above table, with the variable length coding scheme, the average improvement is 5%. Such an improvement can be made even more significant when more rules and higher dimensional problems are involved. This will be confirmed in Section 6.3.4 (refer to Table 6.5) where the modelling mechanical properties of heat treated alloy steel is considered.

Figure 5.38 shows a snapshot of the approximate Pareto fronts at 10, 100, 500, 800, 1000 and 1200 iterations respectively. As one can see from this figure, the evolution starts from the most accurate FRBS and expands the Pareto front during the course of the optimisation. The variable length coding and the new distance index play an important role in expanding the rest part of Pareto front and in fine-tuning of the parameters of the evolved simpler FRBSs. The MO search process is efficient since after 800 iterations it has already approached very closely to the approximate Pareto front.

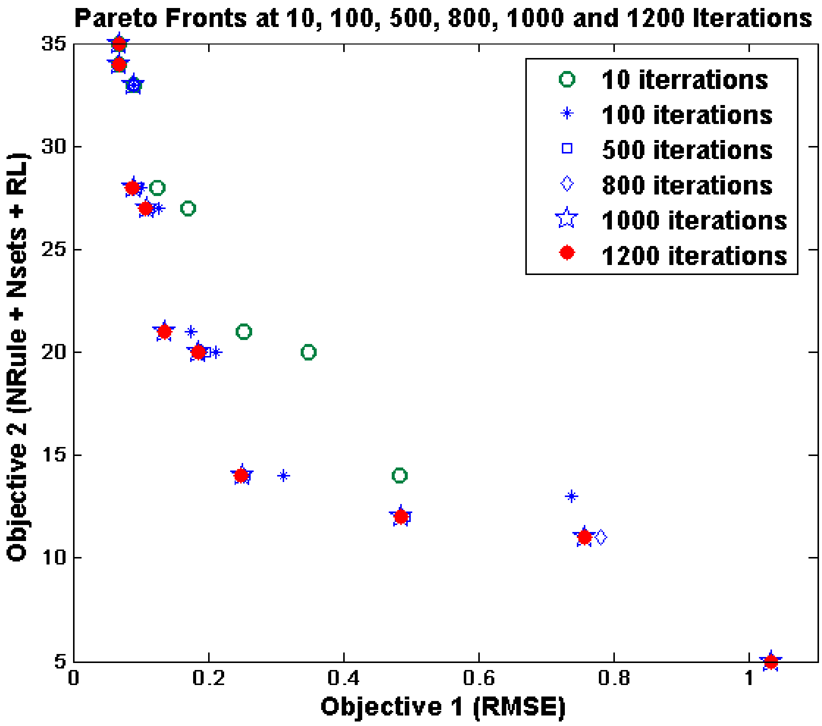


Figure 5.38 The snapshot of the Pareto FRBS at 10, 100, 500, 800, 1000 and 1200 iterations.

5.7 Summary

In the chapter, a systematic immune inspired multi-objective fuzzy modelling framework, namely IMOFM, is introduced. The main novelty of the proposed modelling framework are considered as follows:

- ✧ The proposed modelling approach is not sensitive to the initial settings due to the evolutionary based clustering algorithm used in the first stage.
- ✧ The initial abstraction level (the initial number of rules in the rule-base) is not an important factor anymore since in the third stage a set of Pareto FRBS with different structure are elicited. Only the maximum allowable number of rules is required *a priori*.
- ✧ Due to the vaccination process used in the three-stage modelling procedure, the efficiency and predictive accuracy of the modelling are improved.
- ✧ By using the variable length coding scheme and a new distance index, the problem of the so-called ‘unordered set of rules’ is resolved, which leads to a more efficient optimisation of the parameters. The effect of the variable length coding scheme and the new distance index used in this paper is equivalent to the synapsing variable-length crossover (SVLC) mentioned by Hutt & Warwick (2007) in that common parental

sequences are automatically preserved in the offspring with only the genetic differences being exchanged.

This simple modelling framework provides the user with more options on a set of elicited optimal models and leaves the designer's intervention to a minimum level. The framework is currently implemented via two types of fuzzy rule-base, viz. Singleton FRBS and Mamdani FRBS. The results on a benchmark example suggest that IMOFM is capable of simultaneously optimising both the rule base structure and its parameters. Because of the added model simplification module, several user-specified parameters are introduced. The similarity measures described in Eqs. 5.24~5.26 will be checked for each fuzzy set, and only the ones with the maximum similarity values will be deleted or merged during each iteration step. For this reason and because of the elitism which records any non-dominated solution found at each iteration step during the experiments these parameters were found not to be critical as long as they are kept within the recommended ranges. In the next chapter, IMOFM will be further tested with two benchmark functions and will then be applied to the modelling mechanical properties of heat treated steels.

Chapter 6

Transparent Knowledge Extractions Using an Immune Inspired Multi-Objective Fuzzy Modelling (IMOFM)

"Inferring models from observations and studying their properties is really what science is about. The models ("hypotheses," "laws of nature," "paradigms," etc.) may be of more or less formal character, but they have the basic feature that they attempt to link observations together into some pattern."

Lennart Ljung, System Identification-Theory for the User, 1999

In this chapter, different performance indices are introduced to quantify the performance of an Immune inspired Multi-Objective Fuzzy Modelling (IMOFM) scheme. IMOFM is further tested with two benchmark functions and is then applied to the modeling of mechanical properties of heat treated steels. In all cases, IMOFM is able to produce not only accurate but also transparent fuzzy models. More importantly, the knowledge expressed by these transparent models is consistent with the domain knowledge. Finally, the comparison of IMOFM_S and IMOFM_M is given at the end of the chapter.

6.1 Performance Indices

In this section, different indices are introduced due to the fact that every single one of them just reflects a fraction of the model's performance. By inspecting all these indices, more objective evaluations are expected.

6.1.1 Root Mean Square Error

In Section 4.4.3, RMSE has been briefly introduced. For the convenience of the description, this index is elaborated again in this section. RMSE is a quadratic scoring rule which measures the average magnitude of the error. The equation for RMSE is given in Section 4.4.3 and is rewritten in Eq. 6.1, where $y_m^{predict}$ and y_m are the predicted and the actual outputs of the mth data sample; N is the total number of the data samples. Expressing Eq. 6.1 in words, the difference between the predicted outputs and corresponding observed values are each squared and then averaged over the samples. Finally, the square root of the average is taken. Since the errors are squared before they are averaged, the RMSE gives a relatively high weight to large errors. This means that the RMSE is most useful when large errors are particularly undesirable. Since square root is taken after the averaged squared value is obtained, the RMSE has the same unit as the data samples. The small value of RMSE is expected for a good fit of the model.

$$RMSE = \sqrt{\frac{\sum_{m=1}^{N}(y_m^{predict} - y_m)^2}{N}} \tag{6.1}$$

6.1.2 R-Square

This statistical measure is included to evaluate more objectively the fit of the model, which is the square of the Pearson product moment correlation coefficient calculated from the measured and predicted values (Tenner, 1991). This index can be interpreted as the proportion of the variance in y_m attributed to the variance in $y_m^{predict}$ and is given by Eq. 6.2:

$$R^2 = \frac{N\cdot\left(\sum_{m=1}^{N} y_m^{predict}\cdot y_m\right) - \left(\sum_{m=1}^{N} y_m^{predict}\right)\cdot\left(\sum_{m=1}^{N} y_m\right)}{\sqrt{\left[N\cdot\sum_{m=1}^{N}\left(y_m^{predict}\right)^2 - \left(\sum_{m=1}^{N} y_m^{predict}\right)^2\right]\cdot\left[N\cdot\sum_{m=1}^{N}(y_m)^2 - \left(\sum_{m=1}^{N} y_m\right)^2\right]}} \tag{6.2}$$

The R-square value varies between 1 for a perfect fit to 0 for no fit. Hence, a big value of R-square is expected for a model with a good predictive performance.

6.1.3 Confidence Band

This index was proposed by Zhang (2008) to measure how confident can one be in each small scope of the predictions. A so-called α%-range confidence value is designed as follows:

$$CB_{S_r} = \sqrt{\frac{\sum_{y_m^{predict}, y_m \in S_r} \left(\varepsilon_m - \overline{\varepsilon_{S_r}}\right)^2}{N_{S_r}}} \quad , \quad 1 < r < N \tag{6.3}$$

Where, S_r is a prediction scope defined by the rth prediction value $y_r^{predict}$ with the lower and upper bounds being $y_r^{predict} \mp 0.005 \cdot \alpha \cdot L^{y^{predit}}$; $L^{y^{predict}}$ is the total range of the prediction values and it equals to the maximal prediction value minus the minimal prediction value; ε_m is defined as $\varepsilon_m = y_m^{predict} - y_m$ and $\overline{\varepsilon_{S_r}}$ is the average value of ε_m over the scope S_r; N_{S_r} is the number of data samples within the scope S_r. As mentioned by Zhang (2008), since it is not realistic to calculate the $\alpha\%$-range confidence band for every possible prediction, some averagely distributed prediction values are selected to provide the confidence values which will be viewed as the representatives of all possible predictions within the same scope. Hence, r is normally less than the total number of data samples and is problem-dependent, which basically defines the granularity of the index over the output range. A lower value of this index indicates a higher confidence for the corresponding predictions.

6.1.4 Error Band

Instead of the above statistical measures, the $\pm 10\%$ error bands provide an additional visual aid to judge how well does the model fit the data. Predictions outside the $\pm 10\%$ error bands indicate a bad fitting in such regions. Figure 6.1 shows an example with the perfect fitting line over the interval [1, 10] and its $\pm 10\%$ error bands.

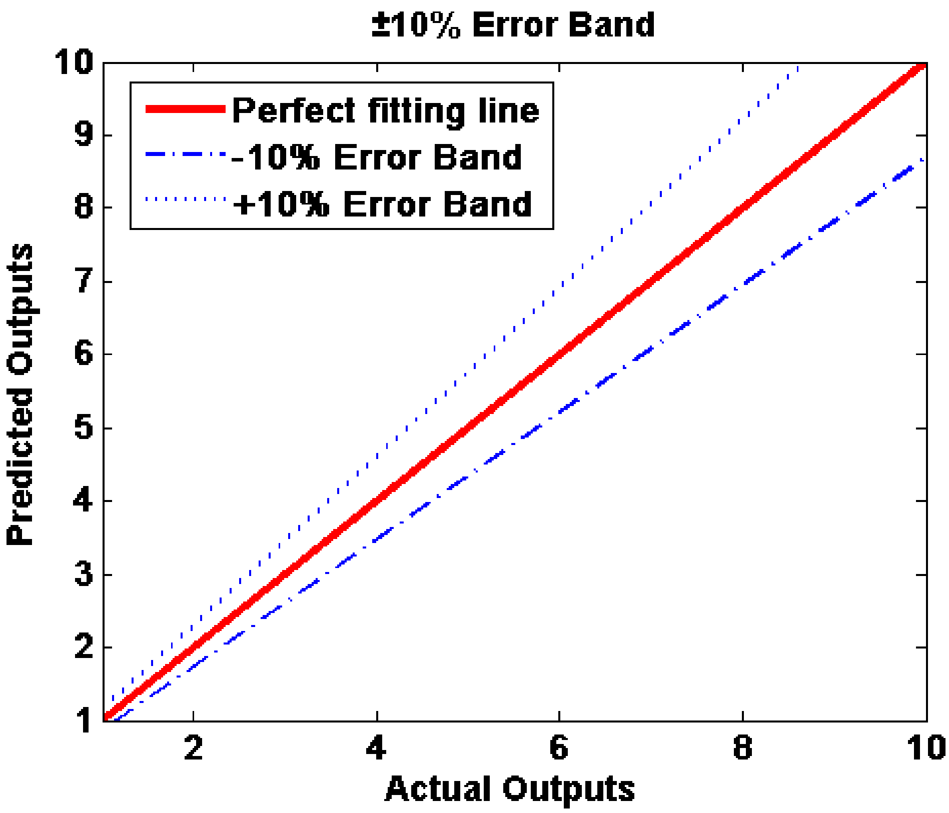


Figure 6.1 ±10% error bands over the interval [1, 10].

6.2 Benchmark Problems

In the previous chapters, a simple nonlinear function was employed as the example to show how each modelling stage works and how the user-specified parameters affect the modelling performance. However, only the leaning ability of IMOFM was investigated. In this Section, such studies are extended to the investigation of the generalisation ability of IMOFM using another nonlinear function approximation problem and a dynamic system identification problem.

6.2.1 Nonlinear Function Approximation

The benchmark problem studied in this Section is a nonlinear static system with two inputs and a single output, taken from Lin *et al.* (1997), which can be described as follows:

$$y = x_2 \sin(x_1) + x_1 \cos(x_2) \quad 0 \leq x_1, x_2 \leq \pi \tag{6.4}$$

To compare with the results in Lin *et al.* (1997), Huang *et al.* (2002), Wong *et al.* (1999), 441 evenly distributed data points from $\{x_1, x_2\} = \{0, \frac{\pi}{20}, \frac{2\pi}{20}, \dots, \frac{19\pi}{20}, \pi\}$ were generated so that 441 input-output data pairs were obtained using Eq. 6.4. To investigate the generalisation capability of IMOFM, Another 100 randomly generated data samples are used as the testing data set. Due to the potential overtraining possibility, the testing data set also serves as the watchdog so that in the second modelling stage the training will be halted if Eq. 6.5 is met, where, $RCK(iteration)$ is the testing RMSE at the current iteration and E_h is the tolerance of the testing RMSE increase which will be explained later:

$$RCK(iteration) - \min\big(RCK(1), \dots, RCK(iteration - 1)\big) > E_h \qquad (6.5)$$

Theoretically, the BEP algorithm used in the second modelling stage is a type of gradient descent optimisation method. Hence, it is supposed to produce more optimal solutions at each iteration step. However, due to the difficulty in setting an optimal step size, such as λ and β in Sections 5.4.1 and 5.4.2, the training and testing RMSE may increase for several iterations and then may decrease during the optimisation process, which would mean that the BEP algorithm also has a limited ability to escape some local optima. Hence, if one allows the BEP algorithm to stop right after the testing RMSE starts to increase, one may lose the chance of obtaining a more accurate model. However, one cannot tolerate too much increase of the testing RMSE since it may indicate the start of the over-fitting phase. For this reason, E_h is introduced in this work, which specifies the tolerance of the testing RMSE increase during the training procedure and is a problem-dependent parameter; E_h is set to 0.05 for this problem.

The maximum number of iterations is set to 1500 so that the second modelling stage will stop training at 1500 iterations even if Eq. 6.5 is not met. The maximum allowable number of rules in the initial FRBS is set to 9. Hence, ‘9’ clusters were obtained in the first modelling stage using G3Kmens, which are then transformed into a 9-rule Singleton/Mamdani fuzzy model. In order to obtain a statistical report, the IMOFM is allowed to run 10 times independently. Figures 6.2 and 6.3 show the training and testing processes of the second modelling stage, which are extracted from one of 10 runs.

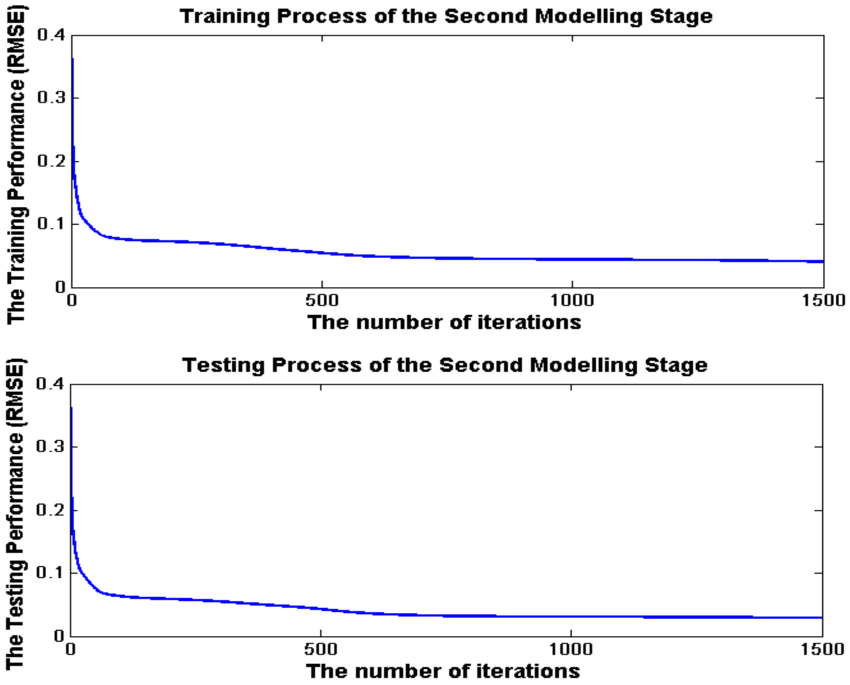


Figure 6.2 The training and testing process of the IMOFM_S on the nonlinear function approximation problem.

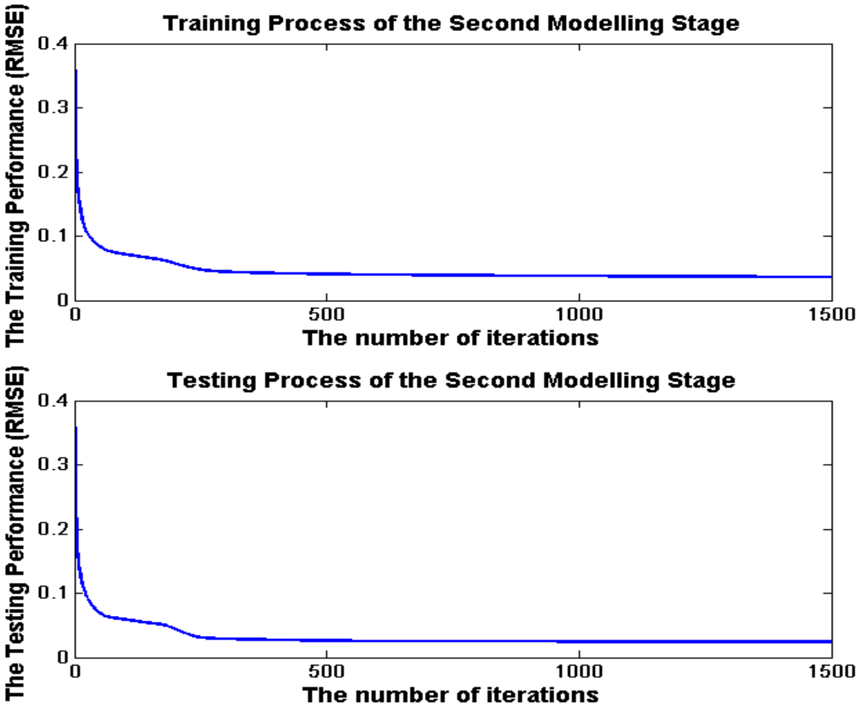


Figure 6.3 The training and testing process of the IMOFM_M on the nonlinear function approximation problem.

Figures 6.4~6.7 shows the predictive performances of the first and the second modelling stages of IMOFM_S and IMOFM_M.

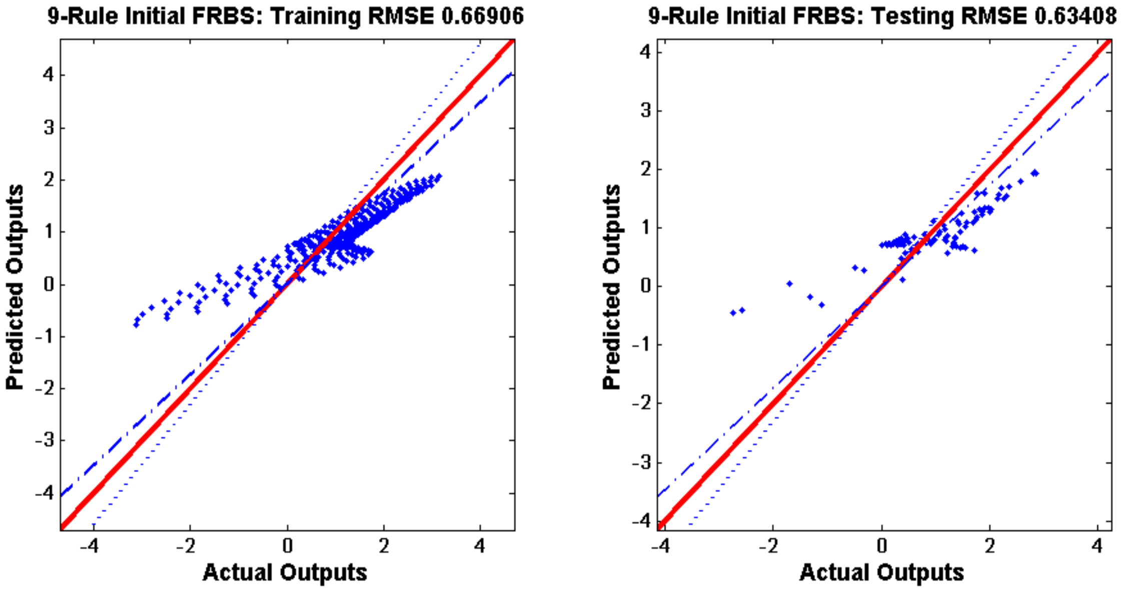


Figure 6.4 The predictive performances of the first modelling stage of IMOFM_S on the nonlinear function approximation problem.

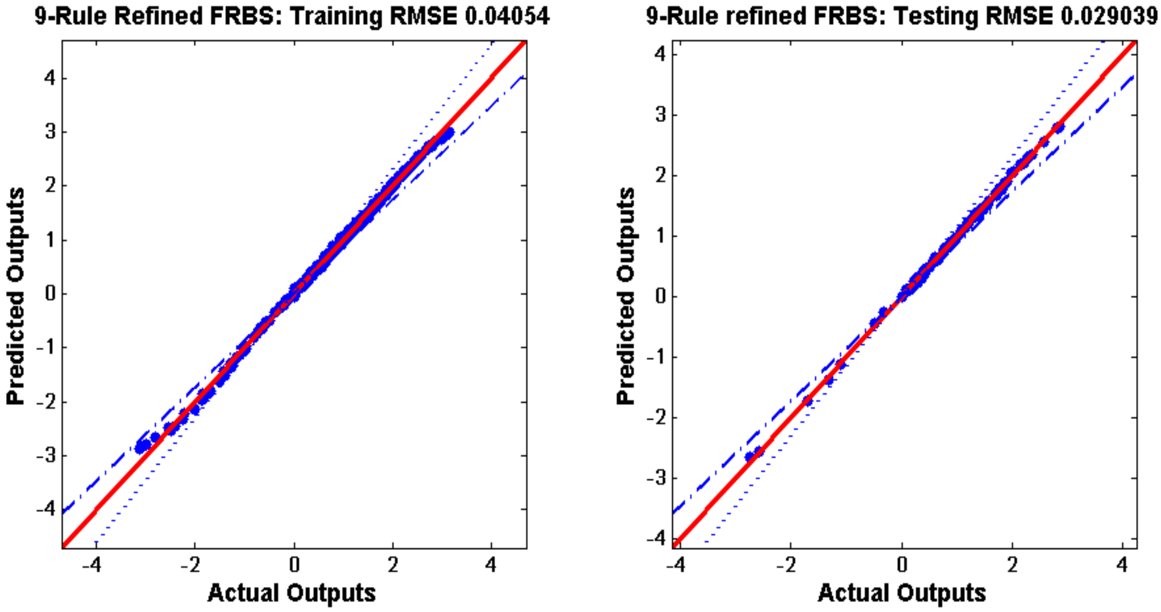


Figure 6.5 The predictive performances of the second modelling stage of IMOFM_S on the nonlinear function approximation problem.

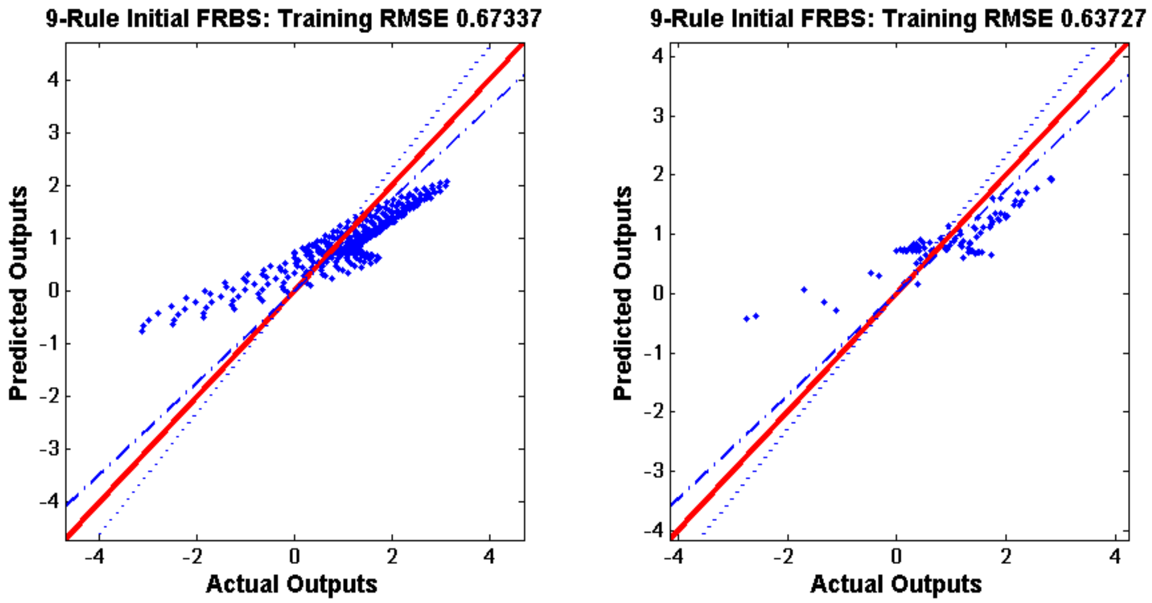


Figure 6.6 The predictive performances of the first modelling stage of IMOFM_M on the nonlinear function approximation problem.

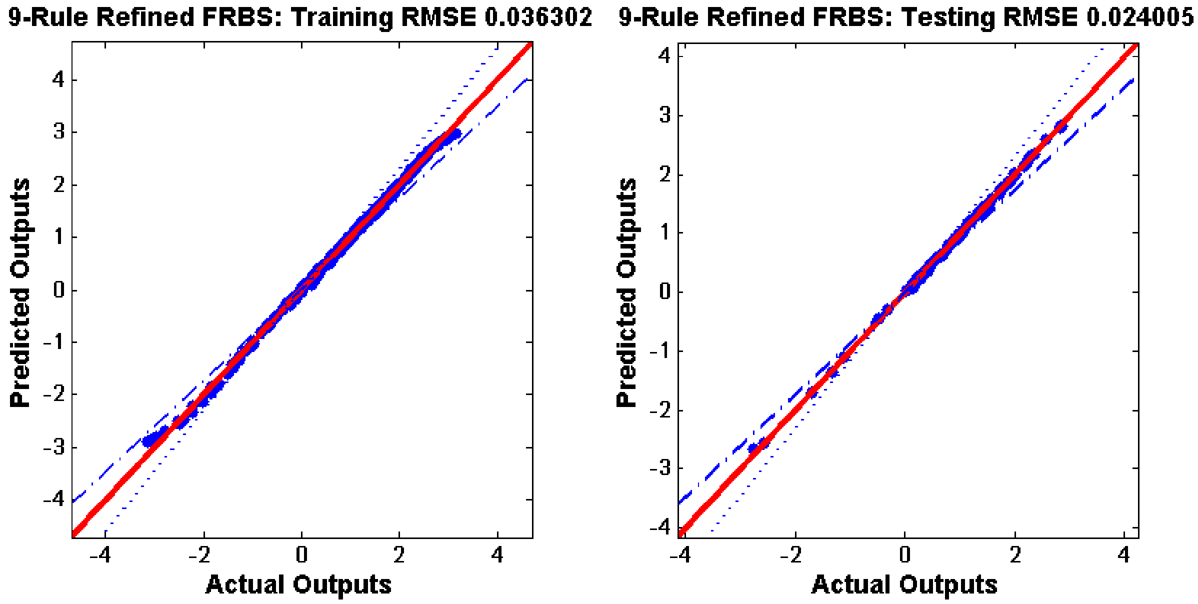


Figure 6.7 The predictive performances of the second modelling stage of IMOFM_M on the nonlinear function approximation problem.

It can be seen from Figures 6.4~6.7 that the predictive performances of the first modelling stage are not accurate enough since for most of the data samples the predictions are outside the ±10% error bands. However, after the second modelling stage, the predictive performances are improved significantly so that most of the predictions are within the ±10% error bands. More importantly, the first two stages of IMOFM not only led to a very good learning but also to excellent generalisation properties. In order to further investigate the performances of each modelling stage, the 3-Dimension surface of Eq. 6.4 and its approximations using the elicited Singleton and Mamdani fuzzy models from each modelling stage are plotted in Figure 6.8.

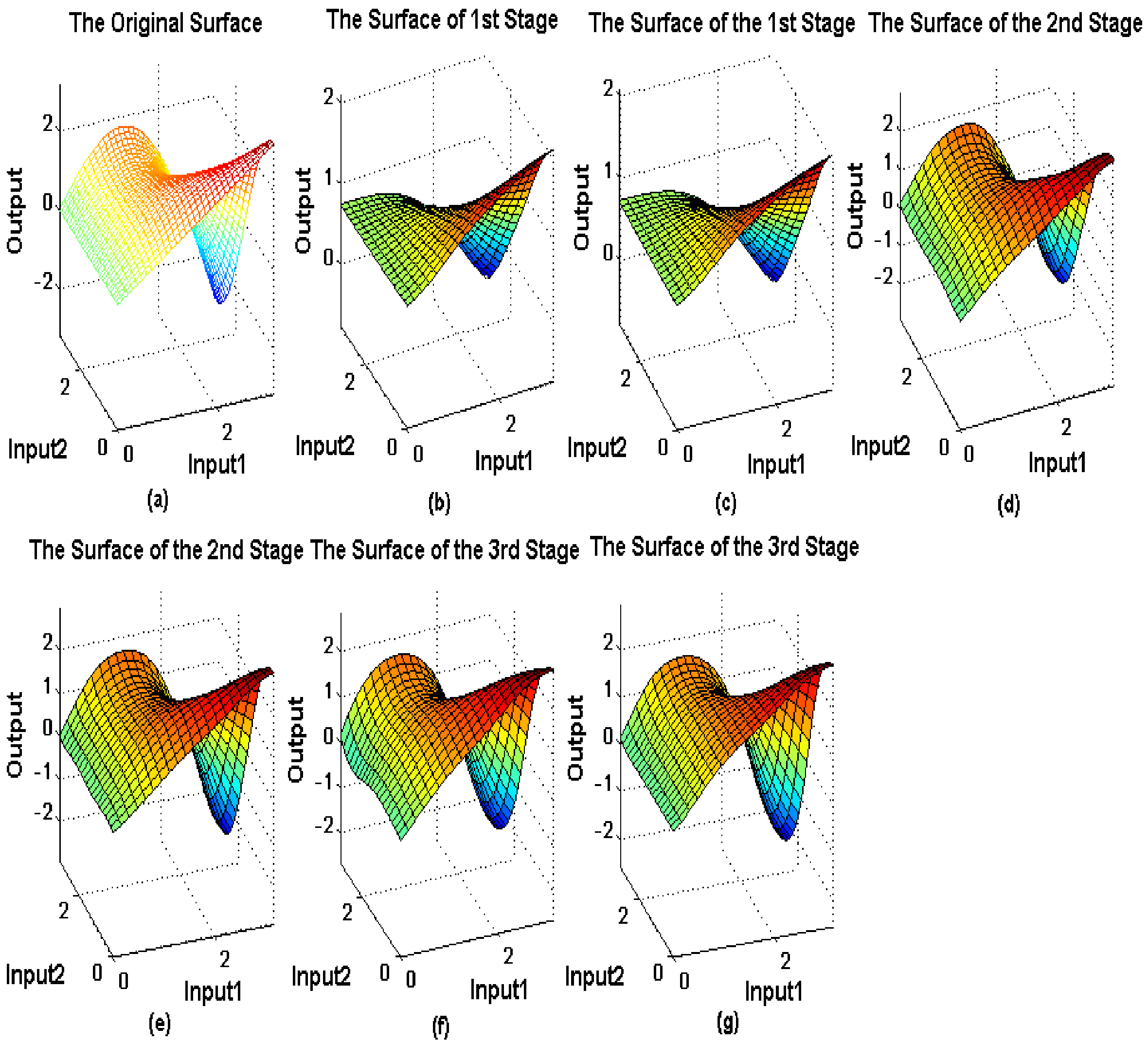


Figure 6.8 Nonlinear system approximation using IMOFM: (a) the actual nonlinear system surface; (b) and (c) the approximated surface obtained from the first modelling stage using IMOFM_S and IMOFM_M; (d) and (e) the approximated surface obtained from the second modelling stage using IMOFM_S and IMOFM_M; (f) and (g) the approximated surface obtained from the third modelling stage using IMOFM_S and IMOFM_M.

As can be seen from Figures 6.8 (b) and (c), the predicted outputs using the initial Singleton and Mamdani fuzzy models are not accurate enough. However, the 3-D surfaces of these initial fuzzy models resemble the surface of the one defined by Eq. 6.4, which suggests it as a good start point for the following refining and multi-objective optimisation stages. After the second stage using a constrained BEP algorithm, both models' predictive accuracies have improved significantly, as shown in Figures 6.8 (d) and (e). Figures 6.8 (f) and (g) show the approximate surfaces of a 5-rule simplified Singleton fuzzy model (with 10 fuzzy sets in its inputs) and a 4-rule Mamdani fuzzy model (with 7 fuzzy sets in its inputs and 4 fuzzy sets in its output). These simplified fuzzy models are one of the many Pareto solutions found by the third modelling stage.

Table 6.1 records the Pareto FRBS which appears more than 5 times over 10 runs. The predictive performances of the initial FRBSs extracted from data using various clustering algorithms are more or less the same, in which Huang's method represents the best result. However, the aim of the clustering operation is not to find out the most accurate model in the first place, but rather it is to discover the right structure behind data so that the refined model based on this structure can be as accurate as possible. It is in this respect that one can regard the proposed G3Kmeans as a better fuzzy partitioning method over the other three methods since it provides the most accurate result after the refinement. Based on the 9-rule refined FRBSs, after the third modelling stage, a 5-rule and a 4-rule simplified FRBS is evolved and selected as the final fuzzy models for use.

As can be seen from Table 6.1, although the rule-base complexity has been greatly simplified, its predictive performance is not sacrificed too much. More importantly, IMOFM displays also a very good generalisation capability for the testing data set. The big values of R-square for most of Pareto FRBSs shown in Table 6.1 reinforce further the good learning and generalisation capabilities of IMOFM.

TABLE 6.1

COMPARISONS OF THE PREDICTIVE PERFORMANCES FOR THE DIFFERENT MODELING METHODS USING THE NONLINEAR FUNCTION APPROXIMATION PROBLEM

Modeling Methods (Ref.)	No. of rules	No. of fuzzy sets$^{\&}$	No. of Parameter s	Consequents	Performance			
					Training (RMSE)	Testing (RMSE)	Training (R-Square)	Testing (R-Square)
Lin's method (1997)	9	18 trapzoidal	72	Singleton	0.4000*/0.1265$^{@}$	-	-	-
Hang's method (2002)	9	18 Gaussian	63	Linear	0.3805*/0.0680$^{@}$	-	-	-
Wong's method (1999)	9	18 Gaussian	45	Singleton	0.4047*/0.0889$^{@}$	-	-	-
IMOFM_S (NB: Average results over 10 runs are presented here)								
Average execution time (Intel(R) Core(TM)2 Duo CPU, 2.27 GHz): 3rd stage: 213sec								
Initial FRBS	9	18 Gaussian: [9 9]	45	Singleton	0.6691*/0.0405$^{@}$	0.6341*/0.0290$^{@}$	0.9011*/0.9993$^{@}$	0.8851*/0.9996$^{@}$
Pareto FRBS1	9	18 Gaussian: [9 9]	45	Singleton	0.0395$^{\#}$	0.0286$^{\#}$	0.9993$^{\#}$	0.9996$^{\#}$
Pareto FRBS2	9	17 Gaussian: [9 8]	43	Singleton	0.0395$^{\#}$	0.0280$^{\#}$	0.9993$^{\#}$	0.0996$^{\#}$
Pareto FRBS3	9	16 Gaussian: [8 8]	41	Singleton	0.0416$^{\#}$	0.0304$^{\#}$	0.9993$^{\#}$	0.9995$^{\#}$
Pareto FRBS4	8	16 Gaussian: [8 8]	40	Singleton	0.0468$^{\#}$	0.0362$^{\#}$	0.9991$^{\#}$	0.9994$^{\#}$
Pareto FRBS5	8	15 Gaussian: [7 8]	38	Sinlgeton	0.0471$^{\#}$	0.0357$^{\#}$	0.9991$^{\#}$	0.9994$^{\#}$
Pareto FRBS6	8	14 Gaussian: [7 7]	36	Singleton	0.0484$^{\#}$	0.0388$^{\#}$	0.9990$^{\#}$	0.9994$^{\#}$
Pareto FRBS7	7	14 Gaussian: [7 7]	35	Singleton	0.0594$^{\#}$	0.0465$^{\#}$	0.9985$^{\#}$	0.9990$^{\#}$
Pareto FRBS8	7	13 Gaussian: [7 6]	33	Singleton	0.0664$^{\#}$	0.0539$^{\#}$	0.9982$^{\#}$	0.9987$^{\#}$
Pareto FRBS9	6	12 Gaussian: [6 6]	30	Singleton	0.0796$^{\#}$	0.0656$^{\#}$	0.9973$^{\#}$	0.9979$^{\#}$
Pareto FRBS10	6	11 Gaussian: [6 5]	28	Singleton	0.0847$^{\#}$	0.0700$^{\#}$	0.9970$^{\#}$	0.9976$^{\#}$
Pareto FRBS11	5	10 Gaussian: [5 5]	25	Singleton	0.1037$^{\#}$	0.0889$^{\#}$	0.9955$^{\#}$	0.9962$^{\#}$
Pareto FRBS12	4	8 Gaussian: [4 4]	20	Singleton	0.1211$^{\#}$	0.1035$^{\#}$	0.9939$^{\#}$	0.9948$^{\#}$
Pareto FRBS 13	4	7 Gaussian: [4 3]	18	Singleton	0.1235$^{\#}$	0.1047$^{\#}$	0.9936$^{\#}$	0.9947$^{\#}$
Pareto FRBS 14	4	6 Gaussian: [4 2]	16	Singleton	0.1309$^{\#}$	0.1144$^{\#}$	0.9928$^{\#}$	0.9936$^{\#}$
IMOFM_M (NB: Average results over 10 runs are presented here)								
Average execution time (Intel(R) Core(TM)2 Duo CPU, 2.27 GHz): 3rd stage: 229sec								
Initial FRBS	9	27 Gaussian: [9 9 9]	54	Mamdani	0.6734*/0.0363$^{@}$	0.6373*/0.0240$^{@}$	0.9018*/0.9994$^{@}$	0.8859*/0.9997$^{@}$
Pareto FRBS1	8	22 Gaussian: [8 7 7]	44	Mamdani	0.0352$^{\#}$	0.0240$^{\#}$	0.9995$^{\#}$	0.9997$^{\#}$
Pareto FRBS2	8	21 Gaussian: [8 6 7]	42	Mamdani	0.0417$^{\#}$	0.0330$^{\#}$	0.9993$^{\#}$	0.9993$^{\#}$
Pareto FRBS3	7	19 Gaussian: [7 6 6]	38	Mamdani	0.0615$^{\#}$	0.0544$^{\#}$	0.9986$^{\#}$	0.9984$^{\#}$
Pareto FRBS4	7	18 Gaussian: [7 5 6]	36	Mamdani	0.0702$^{\#}$	0.0626$^{\#}$	0.9980$^{\#}$	0.9981$^{\#}$

Table 6.1 to be continued...

Table 6.1 continued...

Pareto FRBS5	6	16 Gaussian: [6 5 5]	32	Mamdani	0.0714[#]	0.0647[#]	0.9978[#]	0.9980[#]
Pareto FRBS6	5	12 Gaussian: [5 3 4]	24	Mamdani	0.1001[#]	0.0892[#]	0.9962[#]	0.9958[#]
Pareto FRBS7	4	11 Gaussian: [4 3 4]	22	Mamdani	0.1027[#]	0.0875[#]	0.9957[#]	0.9964[#]
Pareto FRBS8	4	9 Gaussian: [3 3 3]	18	Mamdani	0.1435[#]	0.1282[#]	0.9909[#]	0.9915[#]
Pareto FRBS9	4	8 Gaussian: [3 2 3]	16	Mamdani	0.1786[#]	0.1525[#]	0.9860[#]	0.9882[#]

[&] For IMOFM_S, it is the number of fuzzy sets in its inputs; for IMOFM_M, it is the number of fuzzy sets in its inputs and output.

[*] Initial model extracted directly from data using clustering algorithms or grid partition methods.

[@] Refined model or the consequents are computed through the estimation methods.

[#] Simplified model after model simplification and parameter fine tuning.

[T] Total number of rule length.

Figures 6.9 and 6.10 show the Pareto fronts from one of the 10 runs. As already stated in Section 5.5.6, the decision-making procedure was not explicitly investigated in this work. A rather intuitive approach has been carried out to inspect the Pareto fronts and each individual FRBS. In doing so, a 5-rule simplified FRBS with 10 fuzzy sets for its inputs is chosen as a possible solution for IMOFM_S and a 4-rule simplified FRBS with 7 fuzzy sets in its inputs and 4 fuzzy sets in its consequents is chosen as a possible solution for IMOFM_M because of their acceptable predictive performances and their improved transparency. Figures 6.11, 6.12 and 6.13 show how the initially elicited two 9-rule 'vaccine FRBSs' from the first two modelling stages with highly overlapped membership functions are simplified to a 5-rule and a 4-rule FRBSs with fewer interpretable fuzzy sets.

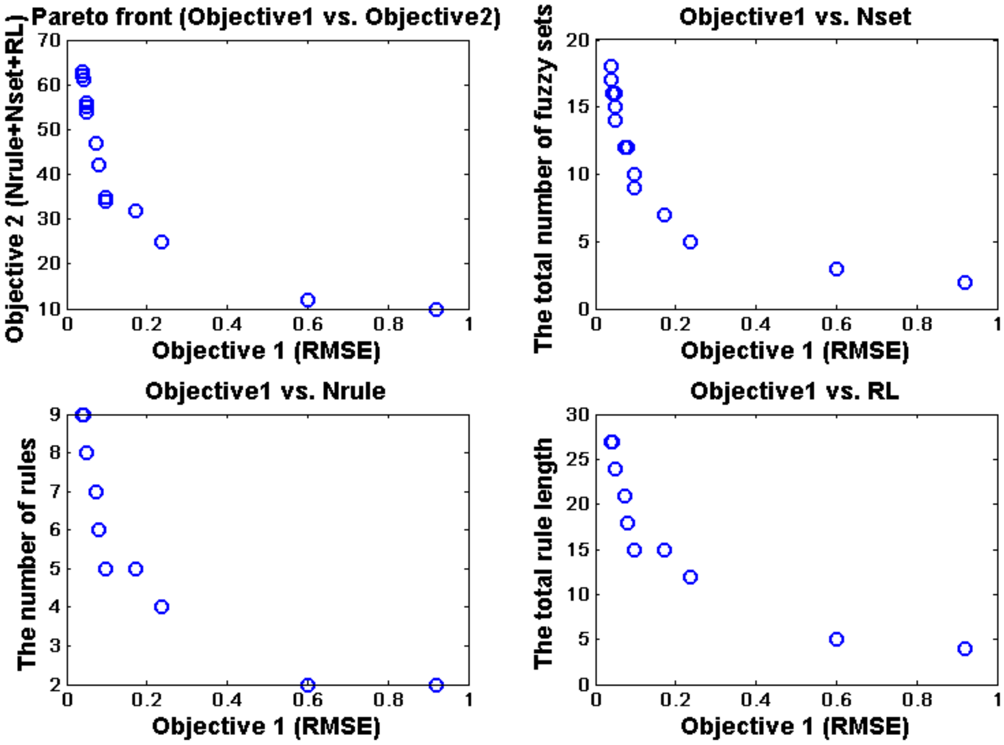


Figure 6.9 The Pareto fronts obtained using IMOFM_S from the third modelling procedure for the nonlinear function approximation problem: (a) Objective1 vs. Objective2; (b) Objective1 vs. Nset[6.1]; (c) Objective1 vs. Nrule[6.2]; (d) Objective1 vs. RL[6.3].

[6.1] Nset is the total number of fuzzy sets in the fuzzy rule-base;

[6.2] Nrule is the number of rules in the fuzzy rule-base;

[6.3] RL is the summation of the rule length of each rule;
(the above definitions of Nset, Nrule and RL are held throught this chapter, and more details on these definitions can be found in Section 5.5.1).

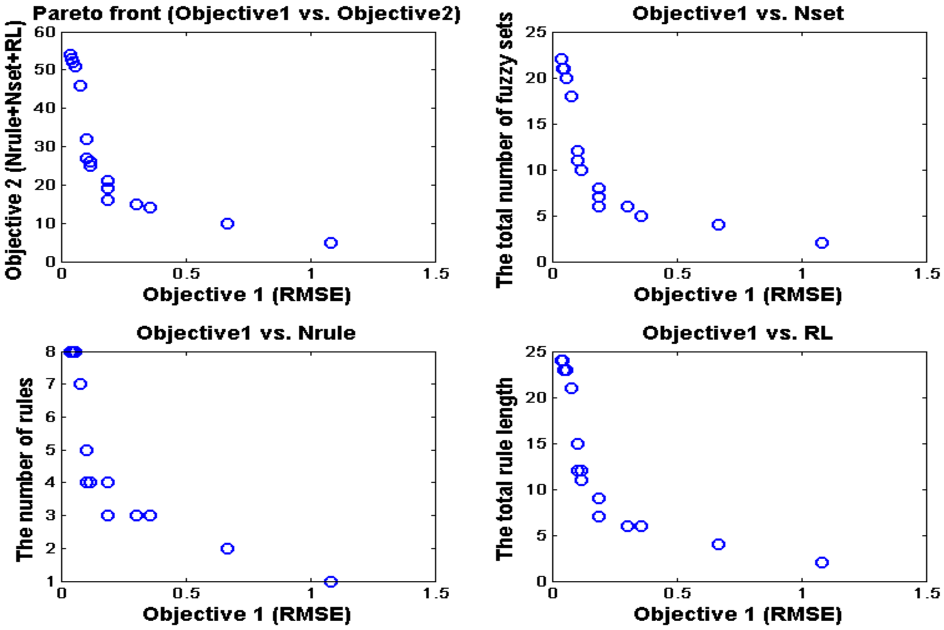


Figure 6.10 The Pareto fronts obtained using IMOFM_M from the third modelling procedure for the nonlinear function approximation problem: (a) Objective1 vs. Objective 2; (b) Objective1 vs. Nset; (c) Objective1 vs. Nrule; (d) Objective1 vs. RL.

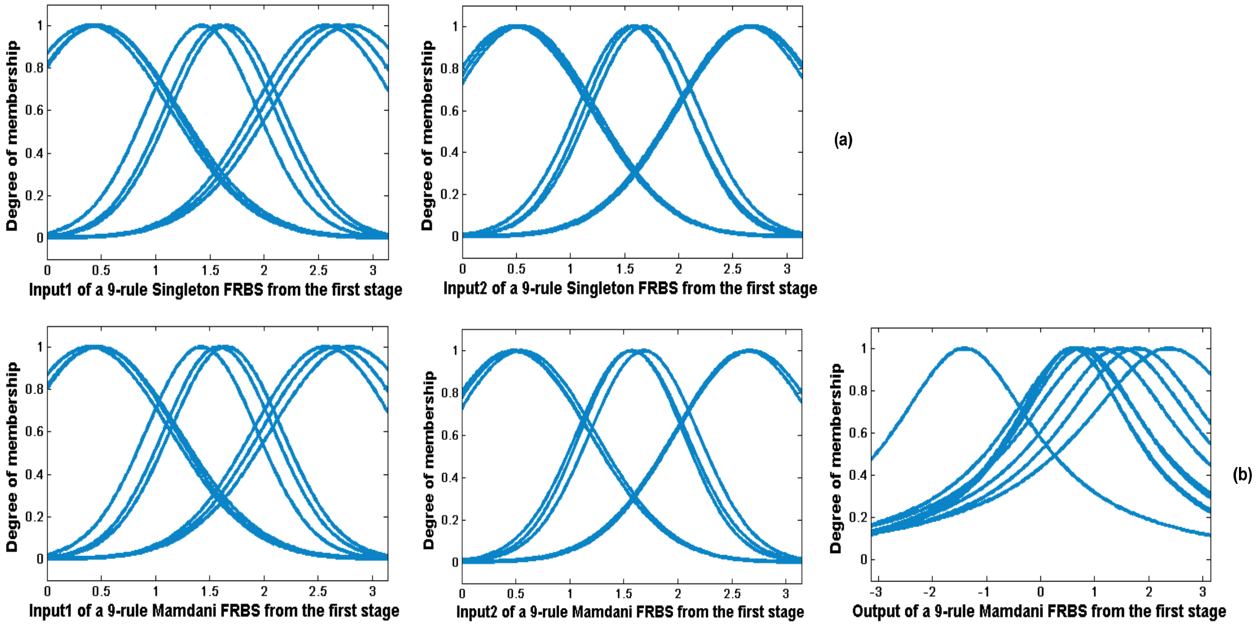


Figure 6.11 The nonlinear function approximation problem: (a) membership function distribution of the 9-rule Singleton FRBS from the first modelling stage; (b) membership function distribution of the 9-rule Mamdani FRBS from the first modelling stage.

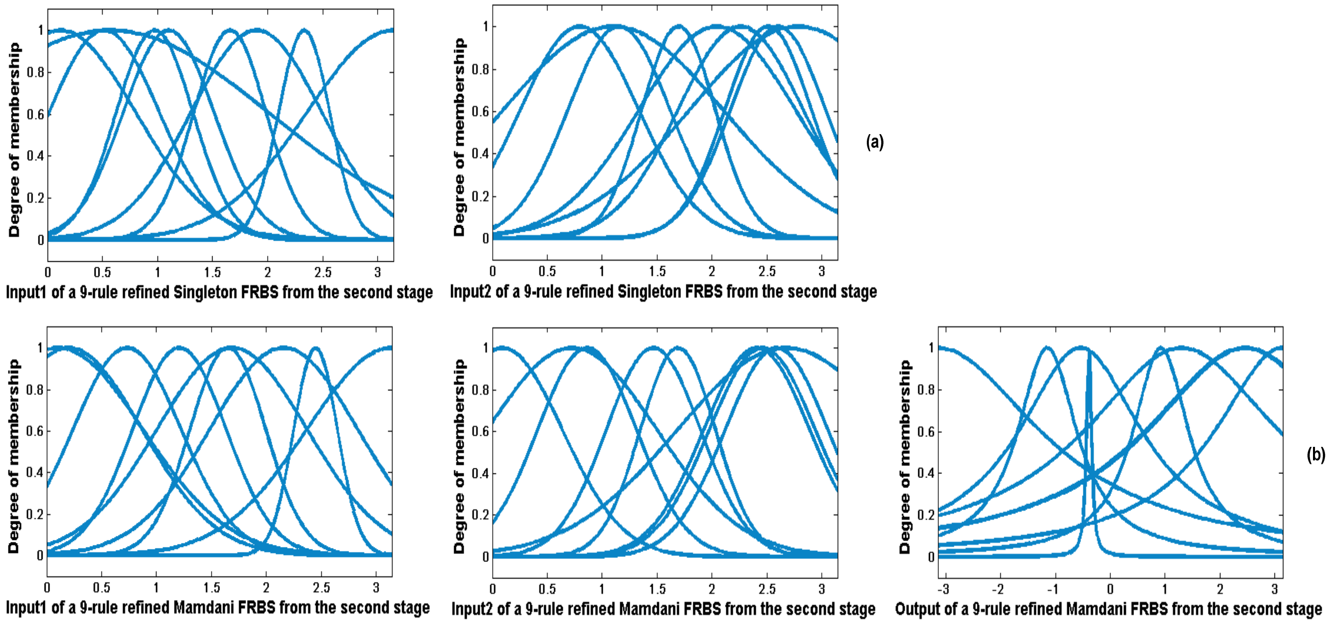


Figure 6.12 The nonlinear function approximation problem: (a) membership function distribution of the 9-rule Singleton FRBS from the second modelling stage; (b) membership function distribution of the 9-rule Mamdani FRBS from the second modelling stage.

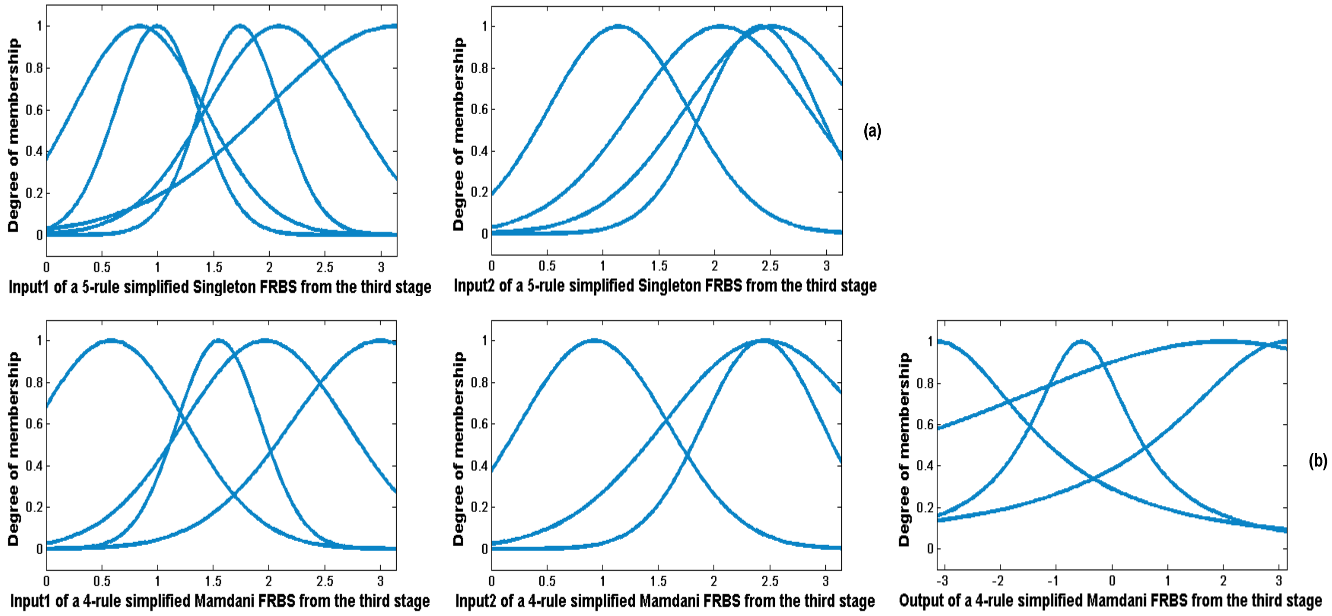


Figure 6.13 The nonlinear function approximation problem: (a) membership function distribution of the 5-rule simplified Singleton FRBS from the third modelling stage; (b) membership function distribution of the 4-rule simplified Mamdani FRBS from the third modelling stage.

Figure 6.14 compares the individual rules in the simplified 5-rule Singleton and 4-rule Mamdani FRBSs.

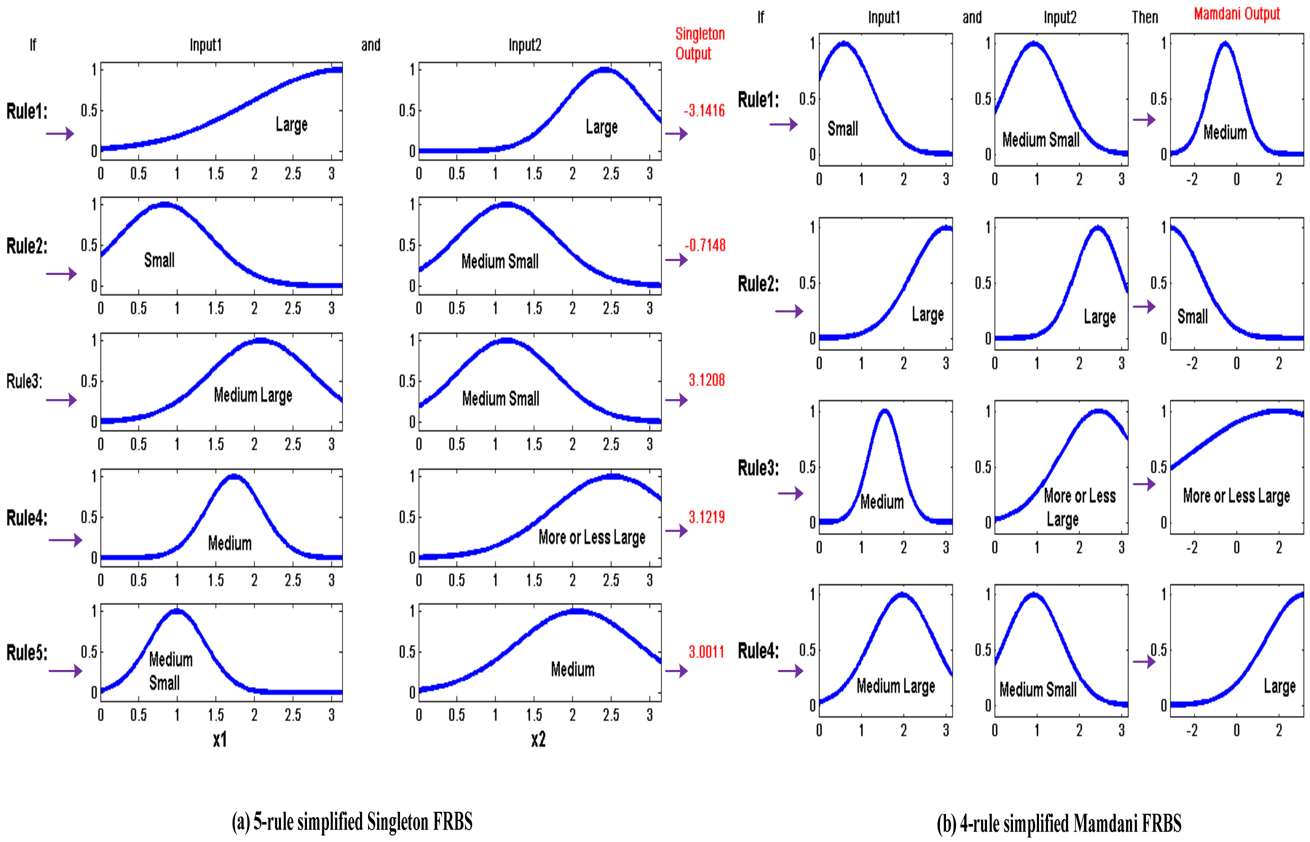


Figure 6.14 (a) 5-rule simplified Singleton FRBS; (b) 4-rule simplified Mamdani FRBS.

As can be seen from Figure 6.14, although the rule bases are extracted via different canonical forms, the knowledge hence expressed is consistent. For IMOFM_M, due to the inclusion of fuzzy sets and the merging operations in its consequents, the hence simplified FRBS is more transparent than the one elicited via IMOFM_S.

Figure 6.15 shows the predictive performances of the simplified Singleton and Mamdani FRBSs by plotting their predicted outputs against the real outputs. As can be seen from the same figure, most of the training and testing predictions are within the ±10% error bands, which indicates that IMOFM not only leads to a good predictive performances but also possesses a good generalisation ability.

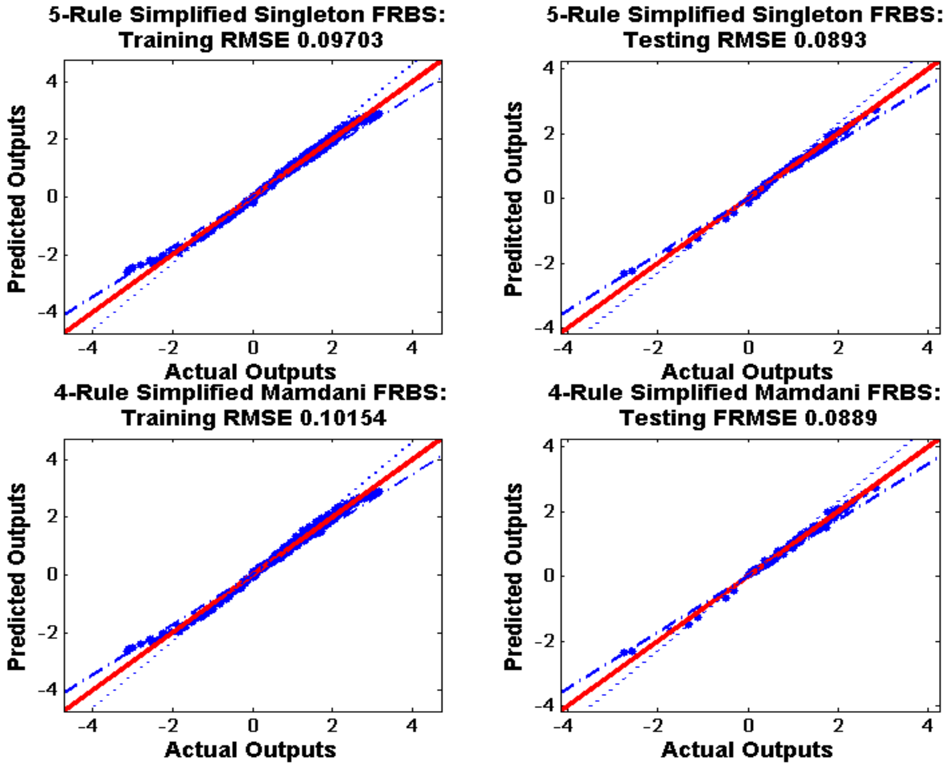


Figure 6.15 The predictive performances of the simplified Singleton (upper part) and Mamdani (lower part) FRBSs obtained from the third modelling stage.

Figure 6.16 shows the 5%-range confidence band of the simplified FRBSs on the training data set.

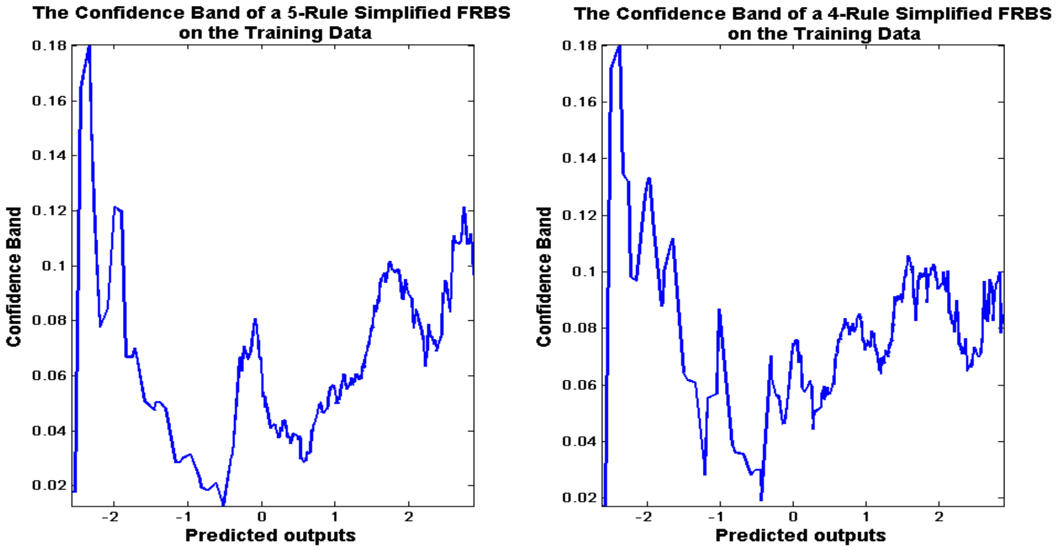


Figure 6.16 5%-range confidence band of a 5-rule simplified Singleton FRBS (left) and a 4-rule simplified Mamdani FRBS (right) on the training data set.

As can be seen from Figure 6.16, the middle parts of the predictions over the range between -2 and $+2$ represent the most reliable parts of the predictions. The predictions fall into the smaller output values convey less confidence since they are rather scattered, or in other words, less consistent.

6.2.2 Dynamic System Identification

The benchmark example studied in this section is a second-order nonlinear plant also used by Setnes *et al.* (2000), Jiménez *et al.* (2001), Wang *et al.* (2005) and Chen *et al.* (2004) in their research. This example is employed to show the learning and the generalisation ability of the proposed modelling scheme. The system is defined as follows:

$$y(k) = g\big(y(k-1), y(k-2)\big) + u(k) \tag{6.6}$$

$$\text{where,}\;\; g\big(y(k-1), y(k-2)\big) = \frac{y(k-1)y(k-2)(y(k-1)-0.5)}{1+y^2(k-1)y^2(k-2)} \tag{6.7}$$

$$K: sampling\ interval;\, y: output;\, u: input$$

The goal is to approximate the nonlinear component $g\big(y(k-1), y(k-2)\big)$ of the plant with a fuzzy model. As in Setnes *et al.*'s work (2000), 400 simulated data points were generated from the plant model using Eq. 6.6. Starting from the equilibrium state (0, 0), 200 samples of training data were obtained with a random input signal $u(k)$ uniformly distributed in [-1.5, 1.5], followed by 200 testing samples obtained using a sinusoid input signal $u(k) = \sin\left(\frac{2\pi k}{25}\right)$. The 400 simulated data samples are shown in Figure 6.17.

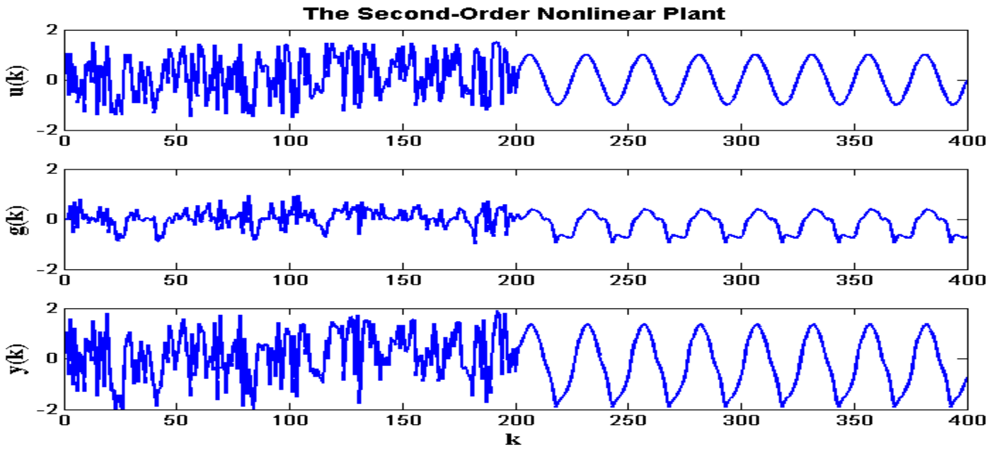


Figure 6.17 Input $u(k)$, unforced system $g(k)$, and output $y(k)$ of the plant in Eq. 6.6.

The maximum allowable number of rules is set to 5 for IMOFM_S and 8 for IMOFM_M. The number of iterations is set to 1200 for the third modelling stage. 30 independent runs were executed and the results shown in this section are the average values of the runs. Other parameter settings are kept the same as those in Section 5.6.2. Figures 6.18 and 6.19 show the evolutions of training and testing processes of the second modelling stage, each of which is extracted from one of 30 runs.

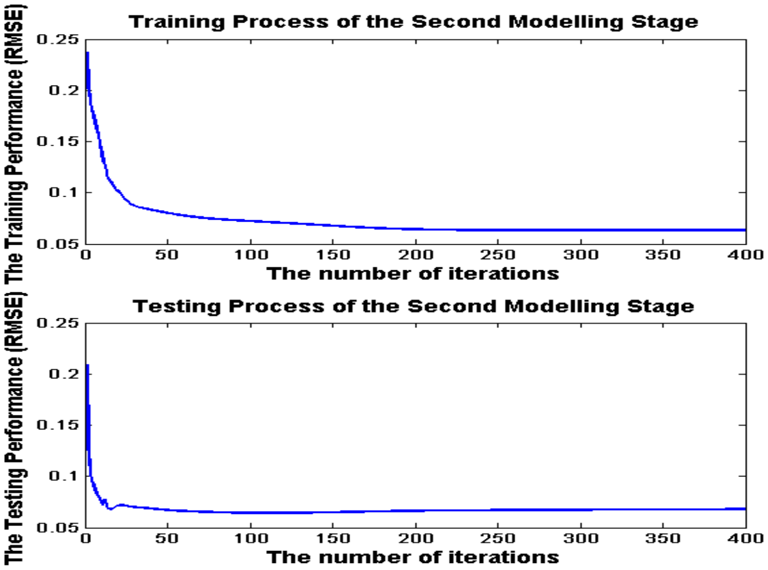


Figure 6.18 The training and testing process of the IMOFM_S on the second-order nonlinear plant.

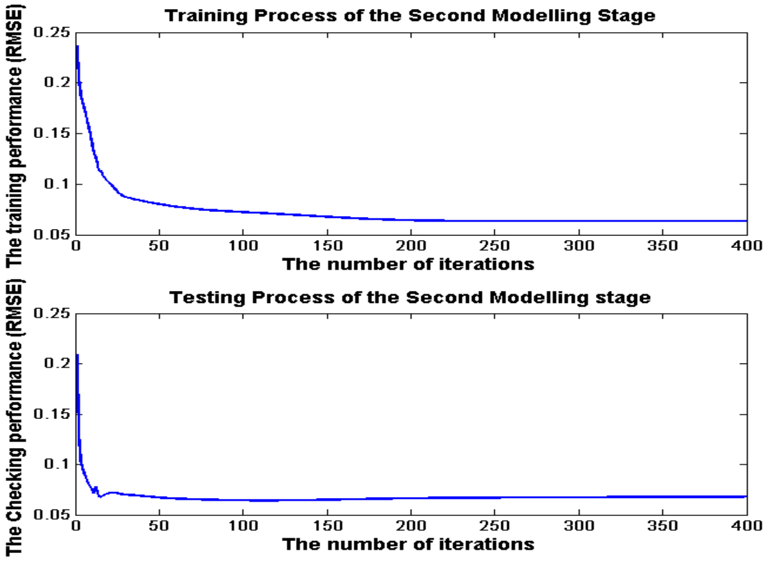


Figure 6.19 The training and testing process of the IMOFM_M on the second-order nonlinear plant.

Figures 6.20~6.23 show the predictive performances of the first and the second modelling stages of IMOFM_S and IMOFM_M.

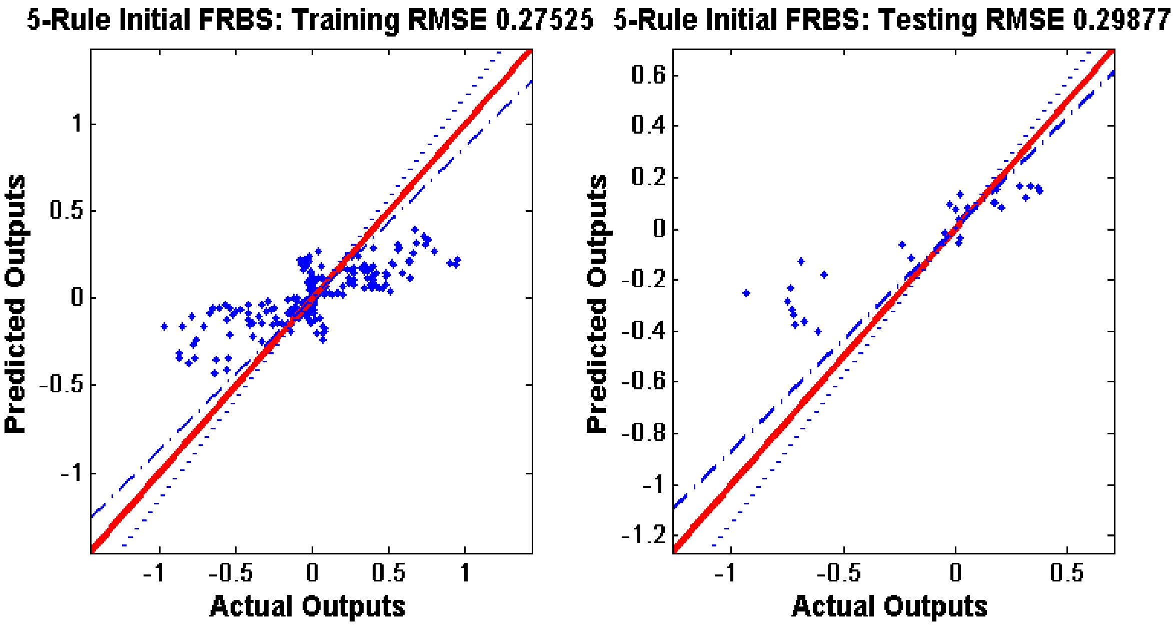


Figure 6.20 The predictive performances of the first modelling stage of IMOFM_S on the second-order nonlinear function approximation problem.

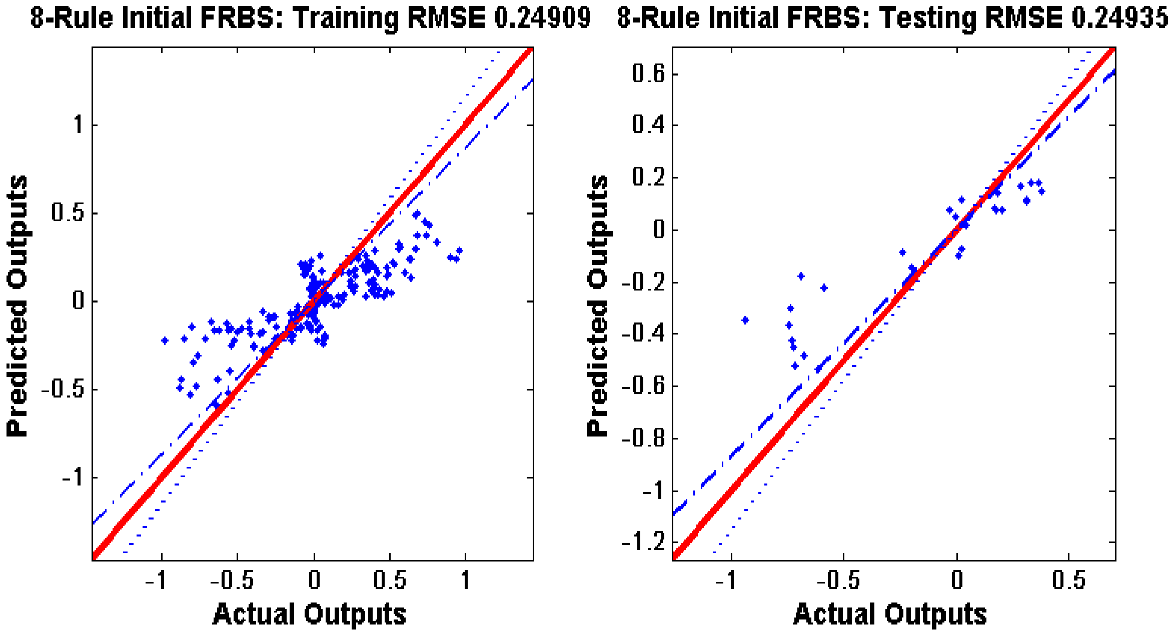


Figure 6.21 The predictive performances of the first modelling stage of IMOFM_M on the second-order nonlinear function approximation problem.

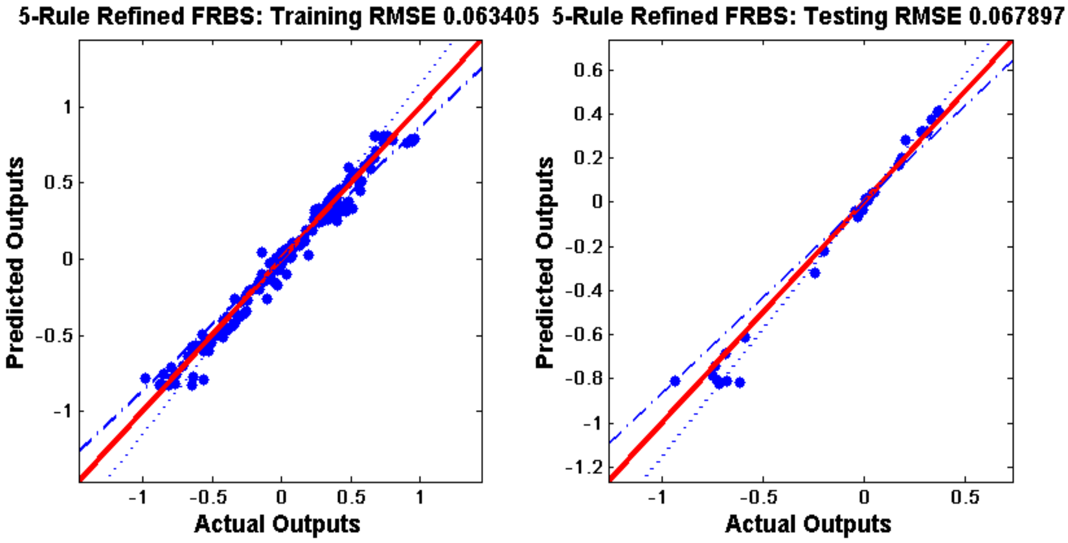


Figure 6.22 The predictive performances of the second modelling stage of IMOFM_S on the second-order nonlinear function approximation problem.

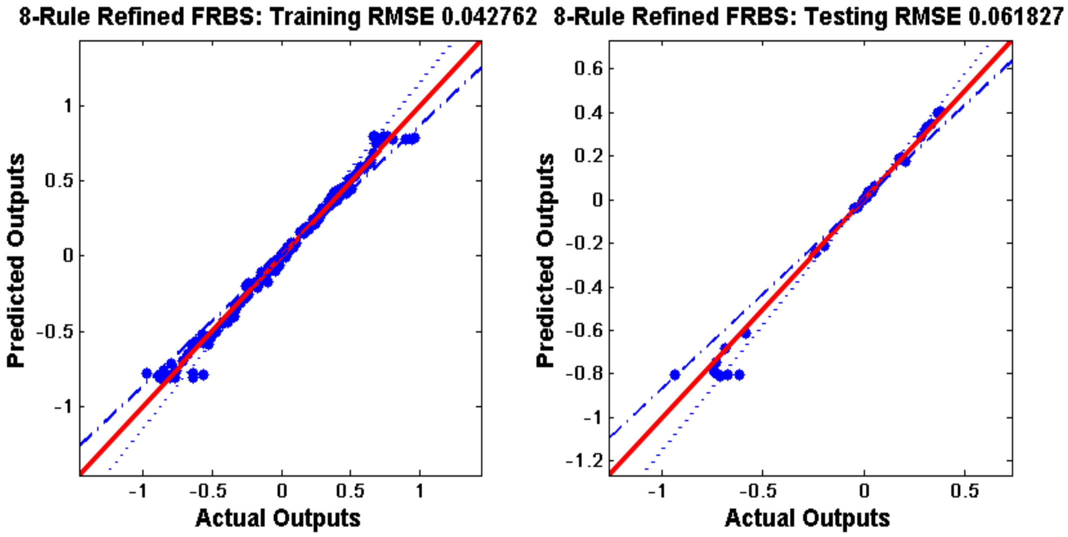


Figure 6.23 The predictive performances of the second modelling stage of IMOFM_M on the second-order nonlinear function approximation problem.

In order to investigate further the performances of each modelling stage, the 3-Dimension surface of Eq. 6.7 and its approximations using the elicited Singleton and Mamdani fuzzy models from each modelling stage are plotted in Figure 6.24, where (f) and (g) represent the approximate surfaces of a 3-rule simplified Singleton fuzzy model (with 4 fuzzy sets in inputs) and a 4-rule simplified Mamdani fuzzy model (with 4 fuzzy sets in inputs and 3 fuzzy sets in output) respectively.

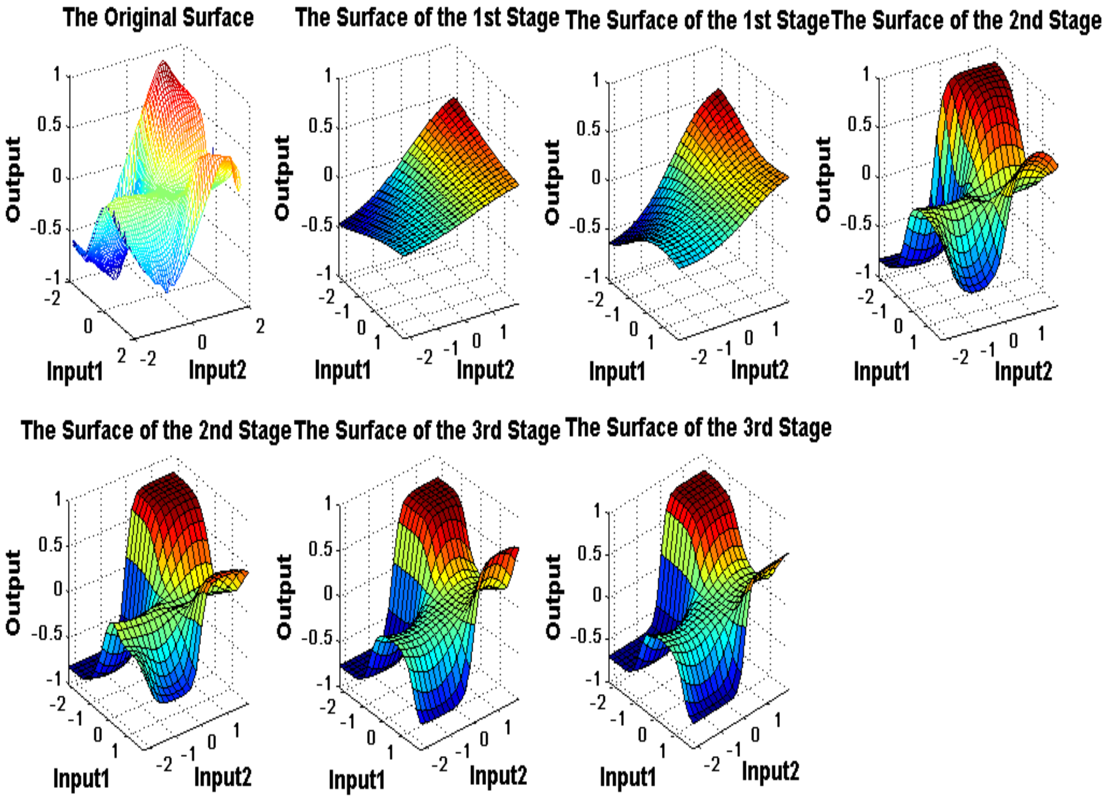


Figure 6.24 The second-order nonlinear system approximation using IMOFM: (a) the actual nonlinear system surface; (b) and (c) the approximated surface obtained from the first modelling stage using IMOFM_S and IMOFM_M; (d) and (e) the approximated surface obtained from the second modelling stage using IMOFM_S and IMOFM_M; (f) and (g) the approximated surface obtained from the third modelling stage using IMOFM_S and IMOFM_M.

Table 6.2 summarises the comparison of the proposed algorihtms' results (both IMOFM_S and IMOFM_M) with those presented by Stenes *et al.* (2000), Jiménez *et al.* (2001), Wang *et al.* (2005) and Chen *et al.* (2004). Each configuration of the Pareto FRBS presented in the above table appeared more than 10 times over 30 runs. When compared with the singleton FRBS presented by Stenes *et al.* (2000), the proposed modelling procedure shows the overall better predictive performance by using fewer rules. Again, linear TSK FRBSs led to a better predictive performance due to the use of linear combinations in their consequents which are hard to interpret. One interesting finding from the experiments of this example using the proposed modelling procedure is the one related to the relationship between the model's complexity and its generalisation ability.

TABLE 6.2

Comparisons Of The Predictive Performances for The Different Modeling Methods Using the Second-order Nonlinear Function Approximation Problem

Modeling Methods (Ref.)	No. of rules	No. of fuzzy sets$^{\&}$	No. of Parameters	Consequents	Performance			
					Training (RMSE)	Testing (RMSE)	Training (R-Square)	Testing (R-Square)
M. Stenes's method (2000)								
Configration1	7	14 triangular	49	Singleton	0.1265*/0.0548$^@$	0.0346*/0.0221$^@$	-	-
Configration2	5	10 triangular	45	Linear	0.0762*	0.05*	-	-
Simplified FRBS1	5	8 triangular	39	Linear	0.0274$^{\#}$	0.0187$^{\#}$	-	-
Simplified FRBS2	4	4 triangular	24	Linear	0.0346$^{\#}$	0.0217$^{\#}$	-	-
F. Jiménez's method (2001)								
Pareto FRBS1	5	5 trapezoidal	30	Linear	0.0447$^{\#}$	0.0361$^{\#}$	-	-
Pareto FRBS2	5	6 trapezoidal	33	Linear	0.0243$^{\#}$	0.0297$^{\#}$	-	-
H. L. Wang‘s method (2005)								
Initial	5	10 Gauss2mf	55	Linear	0.0374$^@$	0.0513$^@$	-	-
Pareto FRBS2	5	3 Gauss2mf	27	Linear	0.0154$^{\#}$	0.0173$^{\#}$	-	-
Pareto FRBS1	4	3 Gauss2mf	24	Linear	0.0234$^{\#}$	0.0233$^{\#}$	-	-
Pareto FRBS2	4	3 Gauss2mf	24	Linear	0.0237$^{\#}$	0.0158$^{\#}$	-	-
M. Y. Chen's method (2004)								
Initial	5	10 Gaussian	55	Linear	0.0138$^@$	0.0195$^@$	-	-
Pareto FRBS2	5	6 Gaussian	27	Linear	0.01$^{\#}$	0.0179$^{\#}$	-	-
Pareto FRBS1	4	5 Gaussian	22	Linear	0.0332$^{\#}$	0.0192$^{\#}$	-	-
IMOFM_S (NB: Average results over 30 runs are presented here)								
Average execution time (Intel(R) Core(TM)2 Duo CPU, 2.27 GHz): 3rd stage: 283 sec.								
Initial FRBS	5	10 Gaussian: [5 5]	25	Singleton	0.2753*/0.0654$^@$	0.2988*/0.0673$^@$	0.7809*/0.9859$^{\#}$	0.9351*/0.9922$^{\#}$
Pareto FRBS1	5	10 Gaussian: [5 5]	25	Singleton	0.0572$^{\#}$	0.0630$^{\#}$	0.9888$^{\#}$	0.9920$^{\#}$
Pareto FRBS2	5	9 Gaussian: [5 4]	23	Singleton	0.0584$^{\#}$	0.0651$^{\#}$	0.9881$^{\#}$	0.9916$^{\#}$
Pareto FRBS3	5	8 Gaussian: [5 3]	21	Singleton	0.0645$^{\#}$	0.0667$^{\#}$	0.9830$^{\#}$	0.9886$^{\#}$
Pareto FRBS4	5	7 Gaussian: [4 3]	19	Singleton	0.0693$^{\#}$	0.0676$^{\#}$	0.9823$^{\#}$	0.9895$^{\#}$
Pareto FRBS5	5	6 Gaussian: [4 2]	17	Sinlgeton	0.0703$^{\#}$	0.0702$^{\#}$	0.9800$^{\#}$	0.9882$^{\#}$
Pareto FRBS6	5(14^T)	6 Gaussian: [4 2]	17	Singleton	0.0730$^{\#}$	0.0677$^{\#}$	0.9814$^{\#}$	0.9894$^{\#}$
Pareto FRBS7	4	6 Gaussian: [3 3]	16	Singleton	0.0808$^{\#}$	0..0730$^{\#}$	0.9785$^{\#}$	0.9867$^{\#}$
Pareto FRBS8	4	5 Gaussian: [2 3]	14	Singleton	0.0803$^{\#}$	0.0732$^{\#}$	0.9761$^{\#}$	0.9867$^{\#}$
Pareto FRBS9	4	4 Gaussian: [2 2]	12	Singleton	0.0791$^{\#}$	0.0728$^{\#}$	0.9782$^{\#}$	0.9873$^{\#}$

Table 6.2 to be continued...

Table 6.2 continued...

Pareto FRBS10	4(11[T])	4 Gaussian: [2 2]	12	Singleton	0.0863[#]	0.0755[#]	0.9743[#]	0.9872[#]
Pareto FRBS11	3	5 Gaussian: [2 3]	13	Singleton	0.1113[#]	0.1082[#]	0.9726[#]	0.9899[#]
Pareto FRBS12	3(8[T])	4 Gaussian: [2 2]	11	Singleton	0.0889[#]	0.0639[#]	0.9724[#]	0.9899[#]
Pareto FRBS 13	2	4 Gaussian: [2 2]	10	Singleton	0.2193[#]	0.2211[#]	0.8152[#]	0.8215[#]
Pareto FRBS 14	2	3 Gaussian: [2 1]	8	Singleton	0.2218[#]	0.1701[#]	0.8153[#]	0.9600[#]
Pareto FRBS 15	2	2 Gaussian: [1 1]	6	Singleton	0.2495[#]	0.2024[#]	0.8125[#]	0.9500[#]

IMOFM_M (NB: Average results over 10 runs are presented here)

Average execution time (Intel(R) Core(TM)2 Duo CPU, 2.27 GHz): 3rd stage: 229sec

Initial FRBS	8	24 Gaussian: [8 8 8]	48	Mamdani	0.2488[*]/0.0437[@]	0.2485[*]/0.0627[@]	0.8233[*]/0.9936[@]	0.9296[*]/0.9928[@]
Pareto FRBS1	7	18 Gaussian: [4 7 7]	36	Mamdani	0.0424[#]	0.0598[#]	0.9995[#]	0.9997[#]
Pareto FRBS2	7	17 Gaussian: [4 6 7]	34	Mamdani	0.0438[#]	0.0603[#]	0.9929[#]	0.9923[#]
Pareto FRBS3	6	16 Gaussian: [4 6 6]	32	Mamdani	0.0482[#]	0.0595[#]	0.9914[#]	0.9917[#]
Pareto FRBS4	6	14 Gaussian: [4 5 6]	28	Mamdani	0.0491[#]	0.0604[#]	0.9917[#]	0.9925[#]
Pareto FRBS5	5	13 Gaussian: [3 5 5]	26	Mamdani	0.0540[#]	0.0590[#]	0.9898[#]	0.9923[#]
Pareto FRBS6	5	12 Gaussian: [3 4 5]	24	Mamdani	0.0568[#]	0.0603[#]	0.9897[#]	0.9925[#]
Pareto FRBS7	5	11 Gaussian: [3 3 5]	22	Mamdani	0.0578[#]	0.0619[#]	0.9876[#]	0.9922[#]
Pareto FRBS8	4	11 Gaussian: [3 4 4]	22	Mamdani	0.0811[#]	0.0602[#]	0.9807[#]	0.9919[#]
Pareto FRBS9	4	10 Gaussian: [3 3 4]	20	Mamdani	0.0861[#]	0.0611[#]	0.9797[#]	0.9921[#]
Pareto FRBS10	4	8 Gaussian: [2 2 4]	16	Mamdani	0.0906[#]	0.0749[#]	0.9662[#]	0.9901[#]
Pareto FRBS11	4	7 Gaussian: [2 2 3]	14	Mamdani	0.0929[#]	0.0836[#]	0.9600[#]	0.9889[#]

[&] For IMOFM_S, it is the number of fuzzy sets in its inputs; for IMOFM_M, it is the number of fuzzy sets in its inputs and output.

[*] Initial model extracted directly from data using clustering algorithms or grid partition methods.

[@] Refined model or the consequents are computed through the estimation methods.

[#] Simplified model after model simplification and parameter fine tuning.

[T] Total number of rule length.

As can be noticed from Table 6.2, when the model's complexity is reduced from a 4-rule Singleton FRBS to a 3-rule Singleton FRBS the generalisation capability is significantly improved. The complex structure may result in an over-fitting to the training data. By obtaining a set of FRBSs with various complexities in a single run, one can analyse the trade-off between the models' learning ability and their generalisation ability. This will provide further help in terms of choosing the 'right' model in the decision-making process. As already mentioned, this work does not propose a new decision-making process, however, for the sake of completeness, a Singleton FRBS with 3 rules and 4 fuzzy sets in its input and a Mamdani FRBS with 4 rules and 4 fuzzy sets in its input and 3 fuzzy sets in its output are chosen from the 'Pareto' FRBSs as the potential solutions for their good generalising properties and their good training performances. The 'Pareto' FRBS mentioned above are the one randomly chosen from 30 runs and their corresponding Pareto fronts are shown in Figures 6.25 and 6.26.

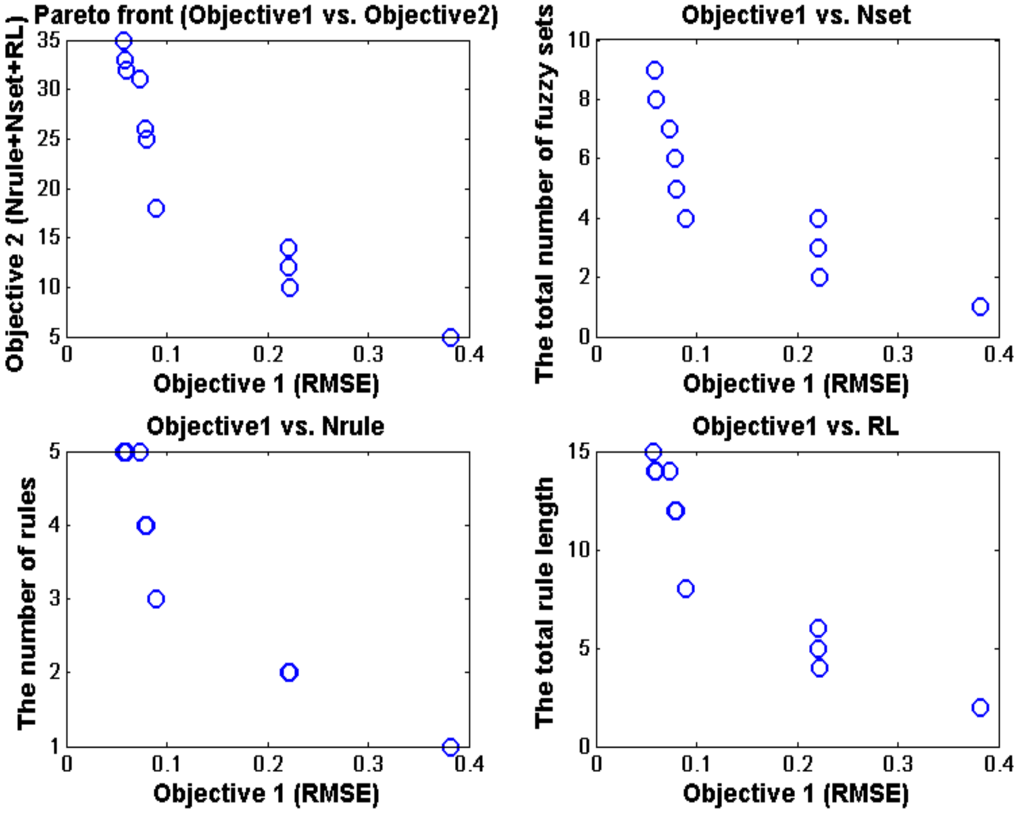


Figure 6.25 The Pareto fronts obtained using IMOFM_S from the third modelling procedure for the nonlinear function approximation problem: (a) Objective1 vs. Objective2; (b) Objective1 vs. Nset; (c) Objective1 vs. Nrule; (d) Objective1 vs. RL.

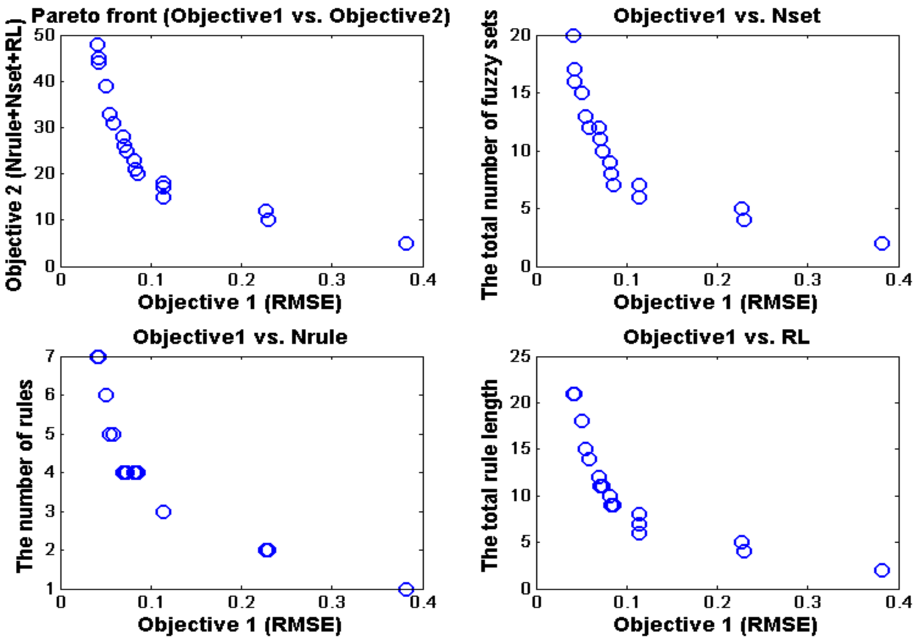


Figure 6.26 The Pareto fronts obtained using IMOFM_M from the third modelling procedure for the nonlinear function approximation problem: (a) Objective1 vs. Objective 2; (b) Objective1 vs. Nset; (c) Objective1 vs. Nrule; (d) Objective1 vs. RL.

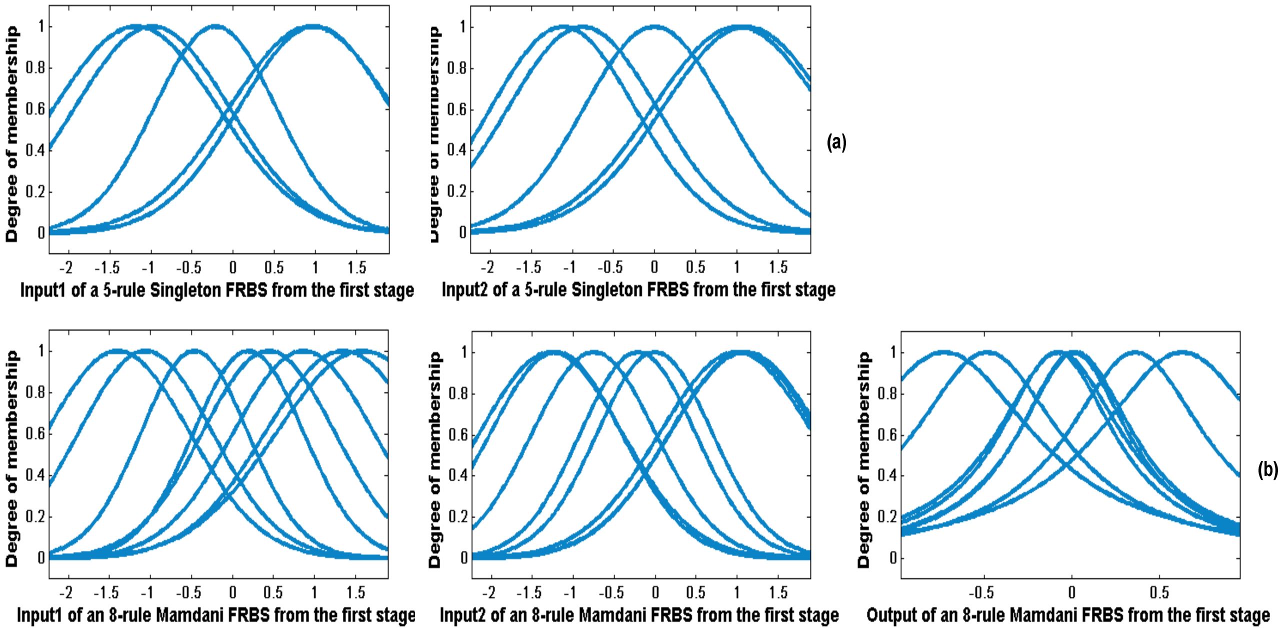


Figure 6.27 The dynamic system identification problem: (a) membership function distribution of the 5-rule Singleton FRBS from the first modelling stage; (b) membership function distribution of the 8-rule Mamdani FRBS from the first modelling stage.

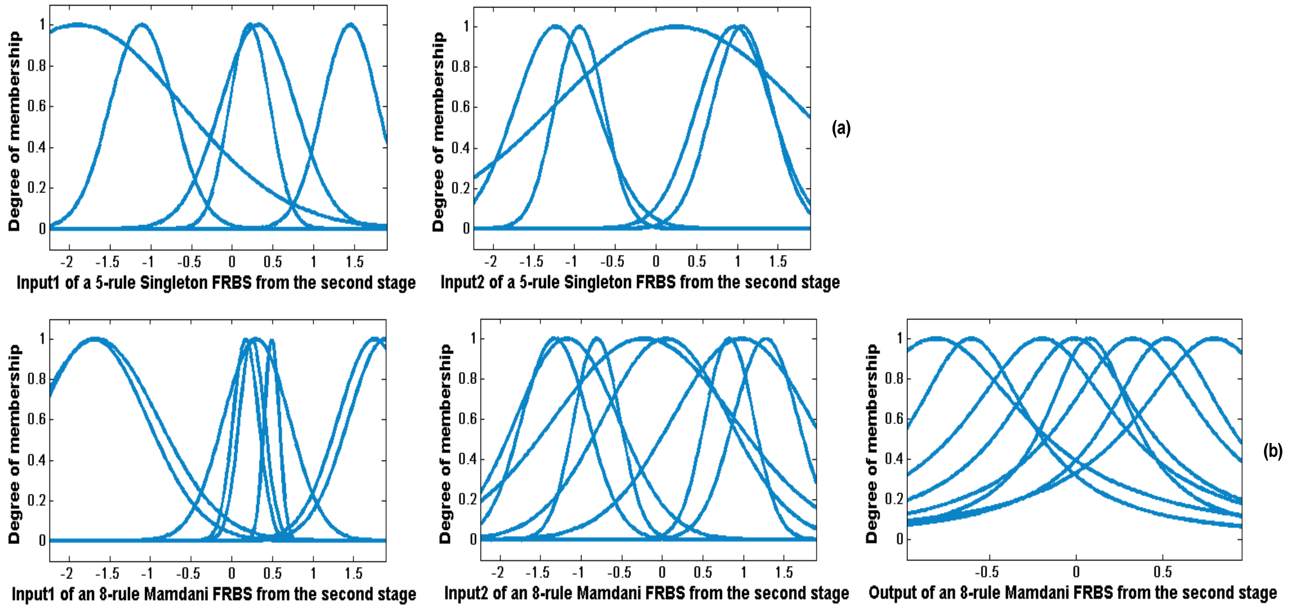


Figure 6.28 The dynamic system identification problem: (a) membership function distribution of the 5-rule Singleton FRBS from the second modelling stage; (b) membership function distribution of the 8-rule Mamdani FRBS from the second modelling stage.

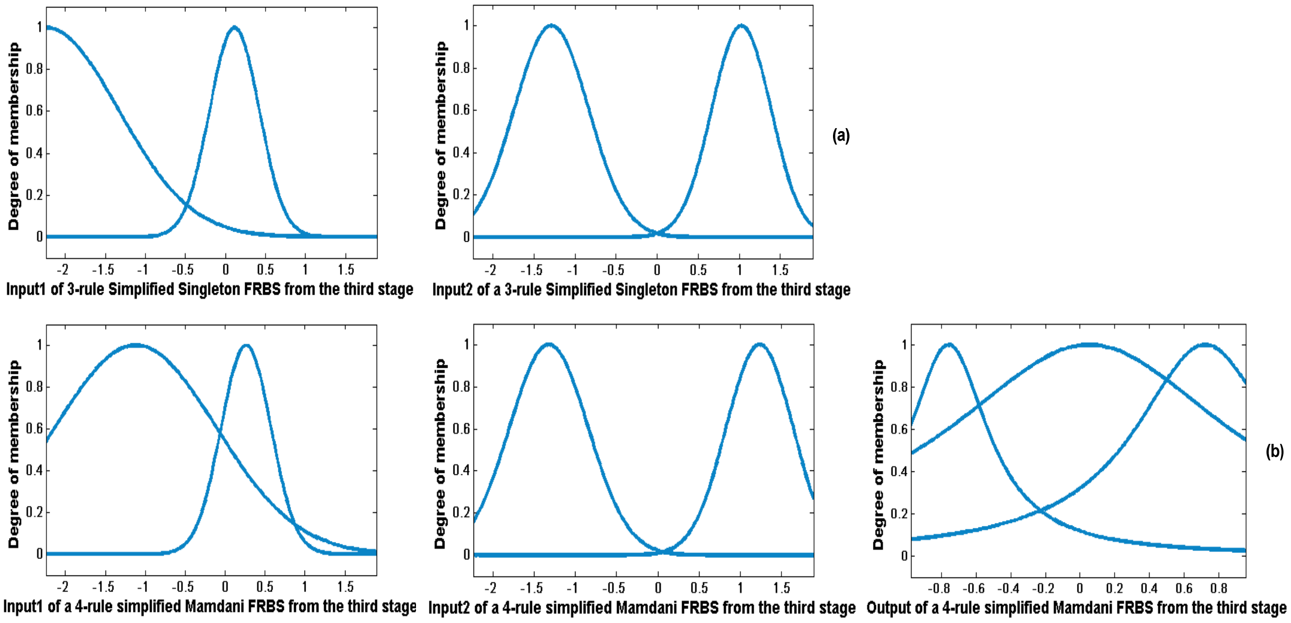


Figure 6.29 The dynamic system identification problem: (a) membership function distribution of the 3-rule simplified Singleton FRBS from the third modelling stage; (b) membership function distribution of the 4-rule simplified Mamdani FRBS from the third modelling stage.

Figures 6.27~6.29 illustrate how the initially elicited 5-rule Singleton and 8-rule Mamdani ‘vaccine FRBSs’ from the first two modelling stages with highly overlapped membership functions are simplified to a 3-rule Singleton and a 4-rule Mamdani FRBSs with fewer interpretable fuzzy sets using IMOFM_S and IMOFM_M. It is worth mentioning that in these simplified FRBS, a ‘don't care’ has been given to one of input *y(k-2)* (input2), as is shown in Figures 6.30 and 6.31.

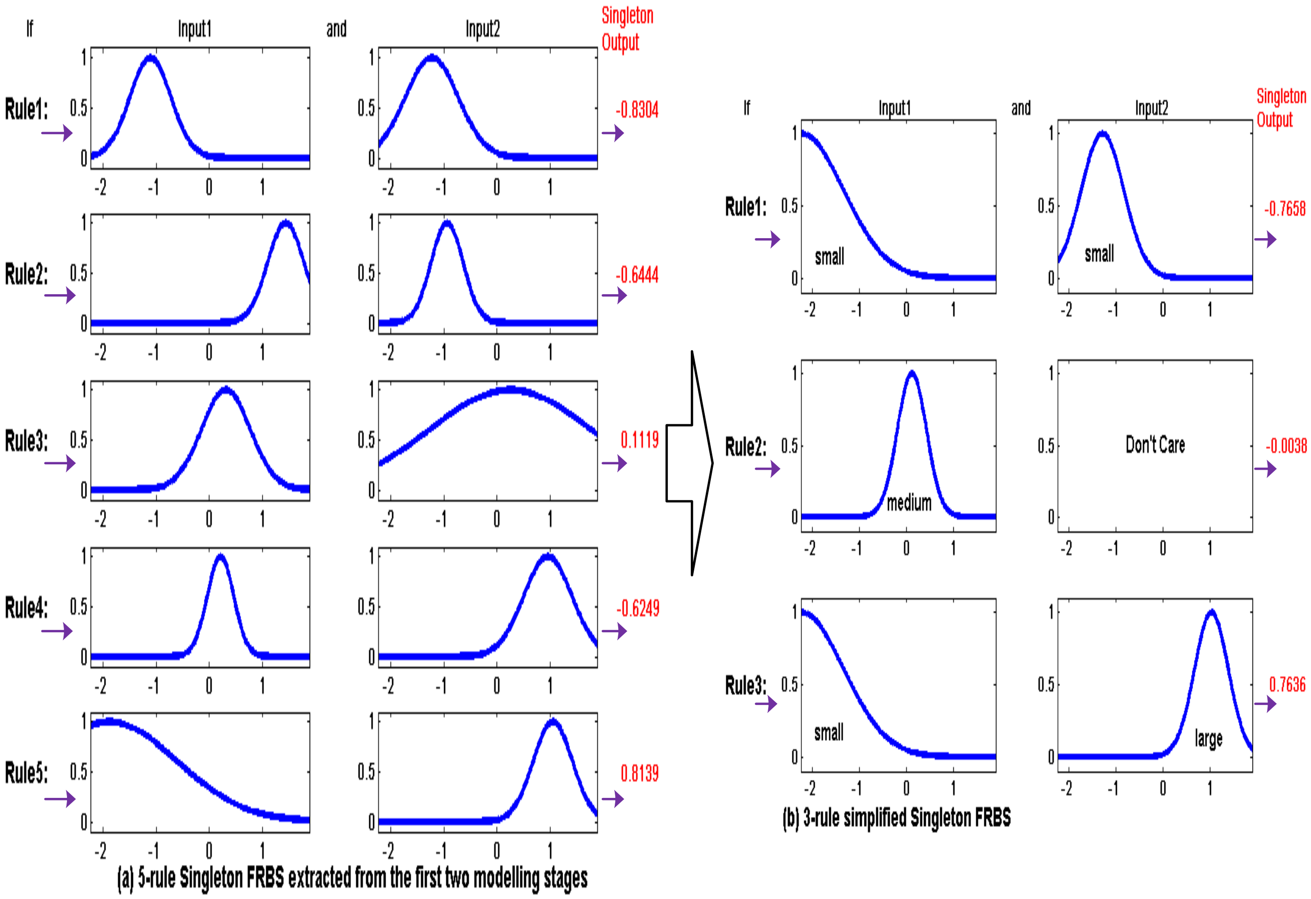
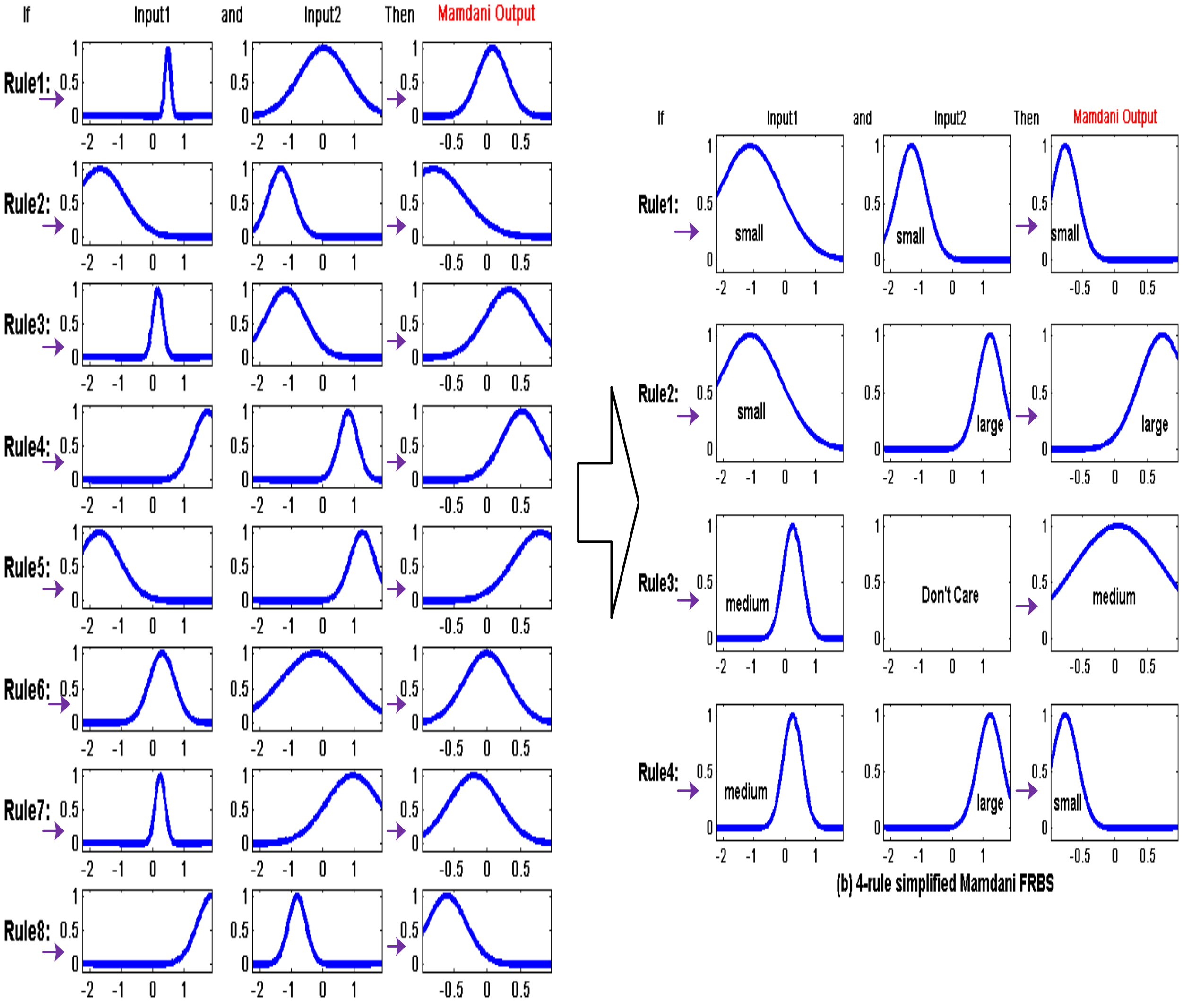


Figure 6.30 The dynamic system identification problem: (a) the 5-rule FRBS extracted from the first two modelling stages using IMOFM_S; (b) the 3-rule simplified Singleton FRBS after the third modelling stage using IMOFM_S.



(a) 8-rule Mamdani FRBS extracted from the first two modelling stages

(b) 4-rule simplified Mamdani FRBS

Figure 6.31 The dynamic system identification problem: (a) the 5-rule FRBS extracted from the first two modelling stages using IMOFM_M; (b) the 3-rule simplified Mamdani FRBS after the third modelling stage using IMOFM_M.

Comparing Figure 6.30 (b) and Figure 6.31 (b), one extra rule (Rule 4) appears in the 4-rule simplified Mamdani FRBS. Other rules from both FRBSs convey consistent knowledge about the underlying systems. Figure 6.32 shows the predictive performances of the simplified Singleton and Mamdani FRBSs by plotting their predicted outputs against the real outputs.

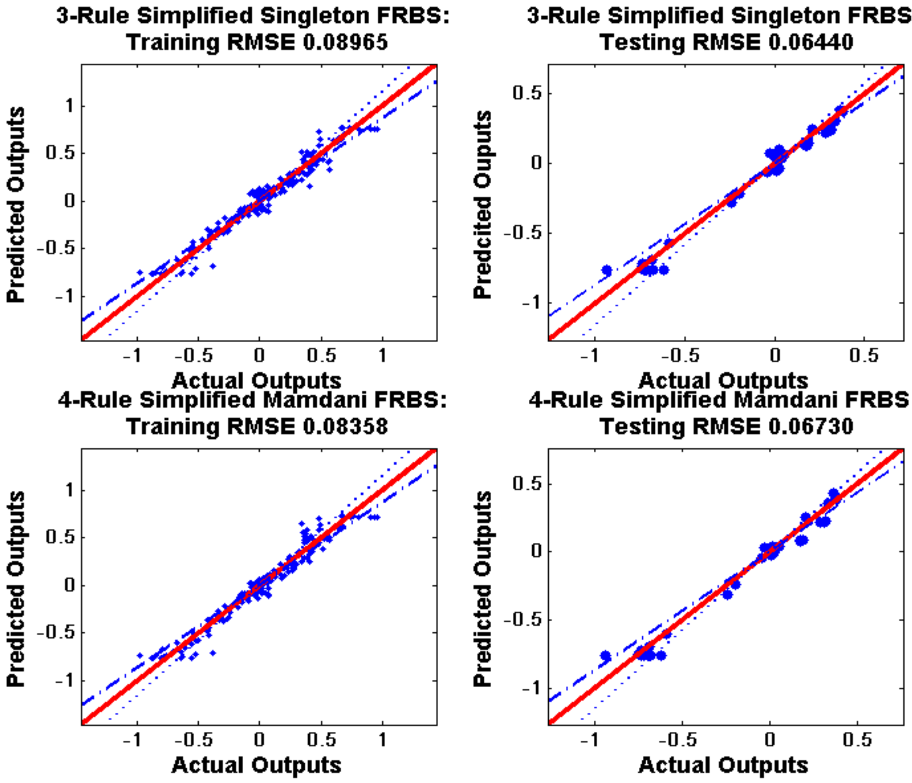


Figure 6.32 The predictive performances of the simplified Singleton (upper part) and Mamdani (lower part) FRBSs obtained from the third modelling stage.

Figure 6.33 shows the 5%-range confidence band of the 3-rule simplified Singleton and the 4-rule simplified Mamdani FRBS on the training data set.

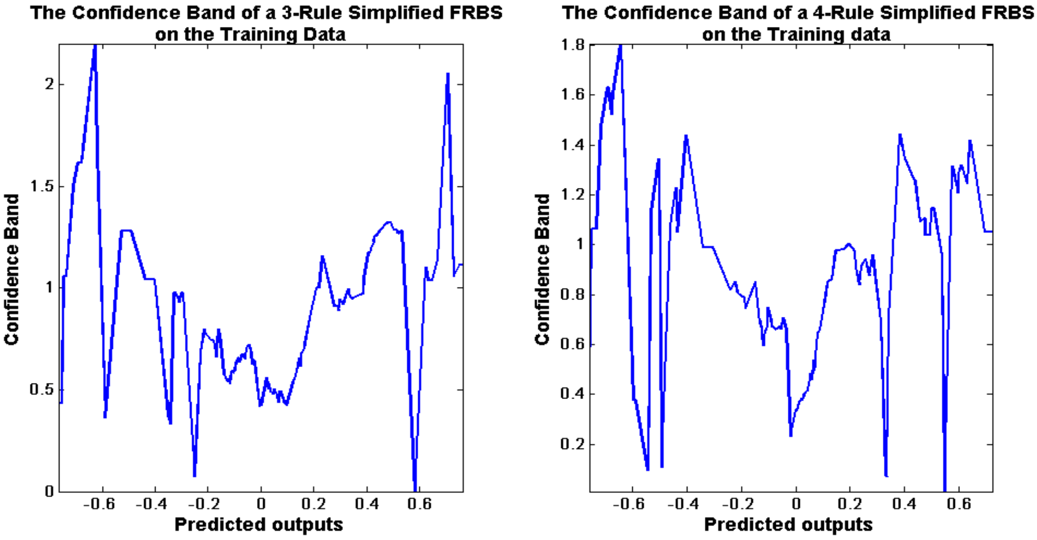


Figure 6.33 5%-range confidence bands of a 3-rule simplified Singleton FRBS (left) and a 4-rule simplified Mamdani FRBS on the training data set.

6.3 Predictions of Mechanical Properties of Heat-Treated Steel

The proposed modelling method is tested further with a real world engineering application associated with the mechanical property prediction of hot rolled steels. Specialist heat treatments are used to develop the required mechanical properties in a range of alloy steels. The mechanical properties of the alloy steels rest with many factors of which the followings are believed to be the major ones: tempering temperature, quench type, chemical compositions of the steel, geometry of the bar, test sample location on the bar, batch distribution in the furnace, measurement tolerances and variations in the process equipment and operators (Tenner, 1999). Traditionally, a heat treatment metallurgist would try to balance these factors using their metallurgical knowledge and experience in a bid to obtain the desired mechanical properties. However, due to the increasing complexity of the underlying system, it becomes more difficult even for the metallurgists to tune these parameters. Given the lack of the mathematical models which can account for these complex systems and a large amount of available industrial process data associated with the systems, data-driven modelling becomes more and more vital for assisting the metallurgist to predict the mechanical test results without actually doing it. Based on these models, further optimisations of the heat treatment process can also be developed, which is envisaged to be able to automate the steel design process and reduce the experimental costs.

In the past, several mechanical property models were developed which were mainly based on linear regression methods (Pickering 1978) or artificial neural networks (Tenner, 1999). The linear models are only designed for specific classes of steels and specific processing routes, and not sophisticated enough to account for more complex interactions, while neural networks are black-box modelling techniques and one cannot have a deep insight into the model (Zhang, 2008). Hence, transparent data-driven modelling framework for material property prediction is still needed.

In this section, the problem of predicting the mechanical properties of heat-treated steel is used as a case study, which involves knowledge acquisition from real industrial data. To this end, a brief overview of the steel-making process and the heat treatment process are first given, before the case studies of predicting Ultimate Tensile Strength (UTS), Reduction of Area (ROA) and Elongation are presented. It is worth noting that Impact Energy will be studied in Chapter 7.

6.3.1 An Overview of the Steel-Making Process

The basic steel making process consists of the following steps (Tenner, 1999):

- **Blast Furnace (BF) Process**: the iron ore is melted in the BF with coke, air and limestone as assistance to remove many embedded impurities.
- **Basic Oxygen Furnace (BOF) Process**: the molten iron from the blast furnace is transported to a BOF for a smelting process, using steel scrap, oxygen, and lime as assisting agents. The major element removed from the molten iron in this oxygen-based steel-making process is carbon, which is removed via oxidation to carbon monoxide (CO). Other impurities are also controlled in this stage depending on the targeted steel grade.
- **Electric Arc Furnace (EAF) Process**: alternatively, molten steel can be produced in an EAF; this procedure involves the melting of scrap charge by electric arcs. The main heat treatment process modelled in this project is fed by steel produced from an EAF. The reactions in the EAF are similar to those in the BOF (Tenner, 1999).
- **Ladle Metallurgy Process**: this is a new process which is increasingly employed by steel makers. It involves molten steel from the EAF or BOF being poured into a refining vessel, where the temperature and composition of the molten iron are closely controlled to produce various grades of steels as required by the customer (Tenner, 1999).
- **The Production of Ingots and Continuous Casting Process**: traditional steel-making involves the production of ingots. However, continuous casting is rapidly replacing the production of ingots. A casting machine is used to produce a continuous piece of solid steel, giving higher yield than ingot formation. In doing so, the intermediate step of rolling ingots into semi-finished sections is avoided.
- **Rolling, Forging and Heat Treatment Process**: this process is required to obtain the correct geometry and properties in the finished product.

6.3.2 Heat Treatment

Heat Treatment is the controlled heating and cooling of metals to alter their physical and mechanical properties without changing the product shape. Metallic materials normally consist of a microstructure of small crystals called ‘grains’. The nature of the grains, e.g.

grain size and orientation, is one of the most effective factors that can determine the overall mechanical behaviour of the metal and is closely related to the temperature at which the grain growth occurs. Hence, heat treatment provides an efficient way to manipulate the properties of the metal by controlling rate of diffusion and the rate of cooling within the microstructure. Steels are heat treated mainly for the following two reasons:

I. **Hardening**: to obtain sufficient hardness in steel it is common for industrial processes to aim for the martensitic microstructure as it can be later tempered to obtain the required mechanical properties (Tenner, 1999). A general used technique for hardening is quenching, which involves heating a metal into the austenitic crystal phase and then quickly cooling the heated metal into martensite structure (a hard brittle crystalline structure). However, on an industrial scale a rapid cooling may not be practical with large pieces of materials. Moreover, a rapid quench may result in cracking due to the thermal stresses. Hence, alloy additions are needed to improve the hardenability of steel, e.g. using chromium, molybdenum, manganese, nickel and occasionally vanadium as additions (Tenner, 1999). These elements act so that martensite can be formed at lower cooling rates. Cooling speeds are mainly determined by the quenching method (cooling mediums), which, from fastest to slowest, go from polymer, brine, fresh water, oil, and forced air. The cooling mediums used in this project are water, oil and air. The aforementioned process consists of the hardening stage of the heat treatment process. However, the martensite may result in brittle steel that would be impractical for most engineering applications. For this reason, softening stage is introduced to transform some of the martensite into a tougher structure.

II. **Softening**: softening is done to reduce strength or hardness, remove residual stress, improve toughness, restore ductility or refine grain size. Two widely used techniques are annealing and tempering. Annealing is a technique to recover cold work and relax stress within a metal. Annealing typically results in a soft, ductile metal. During annealing, small grains recrystallize to form large grains. The tempering process is very similar to the annealing process, except they may have different 'soaking' temperatures and may be cooled under different cooling rates. Untempered martensite, while very hard and strong, is too brittle to be useful. Hence, most applications may require that the quenched parts to be tempered at a lower tempering temperature (normally around 150°C) to impart some toughness, or to be temperd at a higher tempering temperatures (may be up to 700°C) to impart further ductility.

6.3.3 Introduction to Mechanical Properties and Their Testing

Strength, hardness, toughness, elasticity, plasticity, ductility, brittleness and malleability are mechanical properties used as measures of how metals behave under a certain load. These properties are described in terms of the types of force or stress that the metal must withstand and how these are resisted. As mentioned by Tenner (1999), mechanical properties can be broadly divided into static and dynamic properties. A static property is independent of the loading rate at which a force is applied to the test piece, while a dynamic property is dependent on this.

Typically, static properties include the followings:

- **Strength**: strength is the property that enables a metal to resist a force without deformation; three kinds of loading which may test a material's strength are tensile, compressive and shear.
- **Elasticity**: when a material has a load applied to it, the load causes the materials to deform; elasticity is the ability of a material to return to its original shape after the load is removed.
- **Plasticity**: this property is the opposite of strength; it refers to the readiness of a material that can be permanently deformed to a stretched state when a load is applied.
- **Ductility**: ductility allows a material to stretch, bend, or twist without cracking or breaking; this property makes it possible for a material to be drawn out longitudinally.
- **Malleability**: in comparison to ductility, malleability is the property that enables a material to deform by compressive forces without developing defects.
- **Hardness**: this is the property which measures a material's ability to resist permanent indentation.

Tensile testing is often employed to measure the above static properties, which results in the determination of a number of values, namely the UTS, the Proof Stress, the Yield Stress of the material, and the Elongation and ROA of the specimen. Among these values, in this project, particular interest has been given to the UTS, which is a measure of strength and represents a maximum strain that a material can withstand, the Elongation, which is measured as percentage changes in the gauge length, and the ROA, which is measured as diameter of the specimen after fracture. The Elongation and ROA provide a guide to the ductility of the steel.

Of the dynamic properties:

- ✧ **Toughness**: it is the ability of a material to withstand sudden loading and to be deformed without rupturing.
- ✧ **Brittleness**: it is the opposite of the plasticity and implies lack of ductility or toughness. A brittle metal is the one that breaks or shatters before it deforms.
- ✧ **Fatigue**: it is where a failure can result in a material when a load is applied repeatedly to that place.

Impact testing is a testing method that is used to quantify toughness of a metal. The principle of impact testing is to measure the energy necessary to fracture a standard notched bar specimen, by an impulse load imposed by a striker (Tenner, 1999). The energy absorbed by the specimen is measured by the angle of displacement of the pendulum after the fracture. The striker angle, shape, the depth of the test piece and rate of loading all affect the obtained results, therefore equipment and specimen size have to be standardized. This testing method is problematic, as will be discussed when we proceed to Section 7.4.

6.3.4 Predictions of Ultimate Tensile Strength (UTS)

UTS data set consists of 3760 data samples and includes 15 inputs and one output as shown in Table 6.3.

TABLE 6.3

THE INPUTS AND OUTPUT OF TENSILE STRENGTH DATA SET

Inputs	Test Depth	Size	Site	%C	%Si	%Mn	%S	%Cr
Max.	140	381	6	0.62	0.35	1.72	0.21	3.46
Min.	4	8	1	0.12	0.11	0.35	5e-4	0.05
Inputs	%Mo	%Ni	%Al	%V	Hardening Temperature	Cooling Medium	Tempering Temperature	/
Max.	1	4.16	1.08	0.27	980	3	730	/
Min.	0.01	0.02	5e-3	1e-3	820	1	170	/
Outputs	Tensile Strength (Max.: 1842; Min.: 516.2)							

Figure 6.34 shows the distribution of the data points on some dimensions. As can be seen from Figure 6.34, UTS data is very scattered at some regions, while at the other places it is very dense. Hence, this data set represents a great challenge for both clustering and modelling tasks.

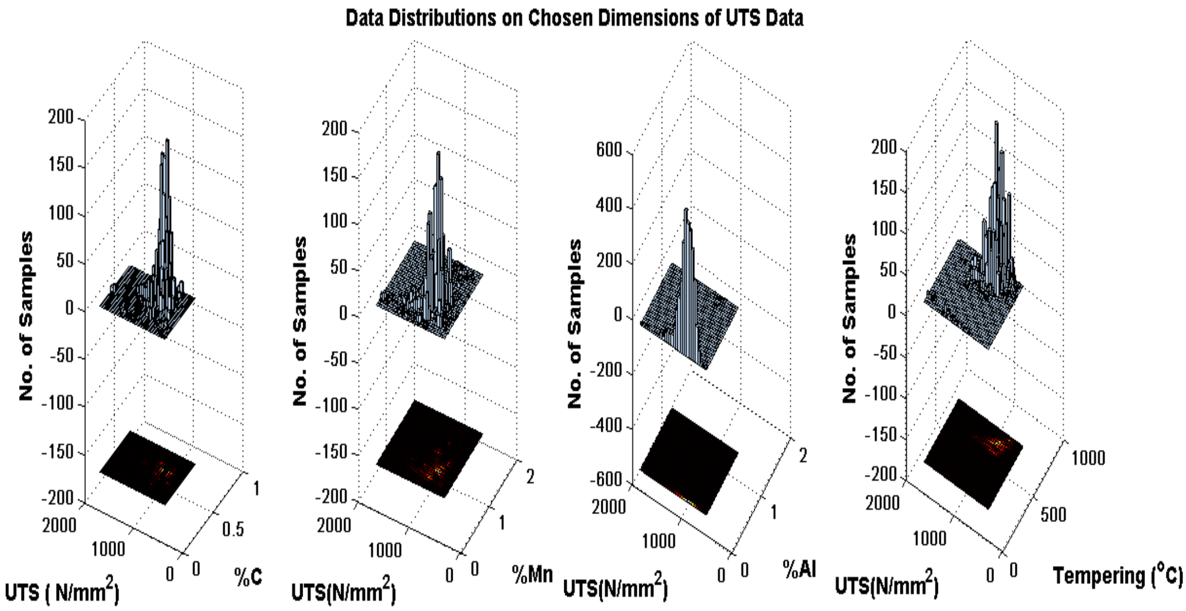


Figure 6.34 Data distributions on chosen dimensions of UTS data.

In order to compare with Zhang & Mahfouf's work (2007), the UTS data set is randomly divided into two parts: 75% of the data are used for training and the remaining data are used for testing. Another 12 more recent samples are used as the unseen data set to validate the generalisation properties of the model. The maximum number of rules is set to 12 for both IMOFM_S and IMOFM_M. The number of iterations for the second and third modelling stages are set to 500 and 1200 respectively. Other parameters are kept the same as those given in Section 5.6.2.

Figures 6.35 and 6.36 show the training and testing processes of the second modelling stage of IMOFM_S and IMOFM_M, each of which is extracted from 1 of 10 runs. As can be seen from these two figures, for the second modelling stage, IMOFM_S terminates at 350 iterations and IMOFM_M terminates at 452 iterations before the specified 500 iterations are reached.

The results presented in Table 6.4 include the average values of 10 independent runs. It is worth mentioning that the training and testing data sets are re-generated before each run with the same proportion indicated before, as will be done for the other mechanical properties. This is to ensure that the results presented here are not dependent on a particular data partition.

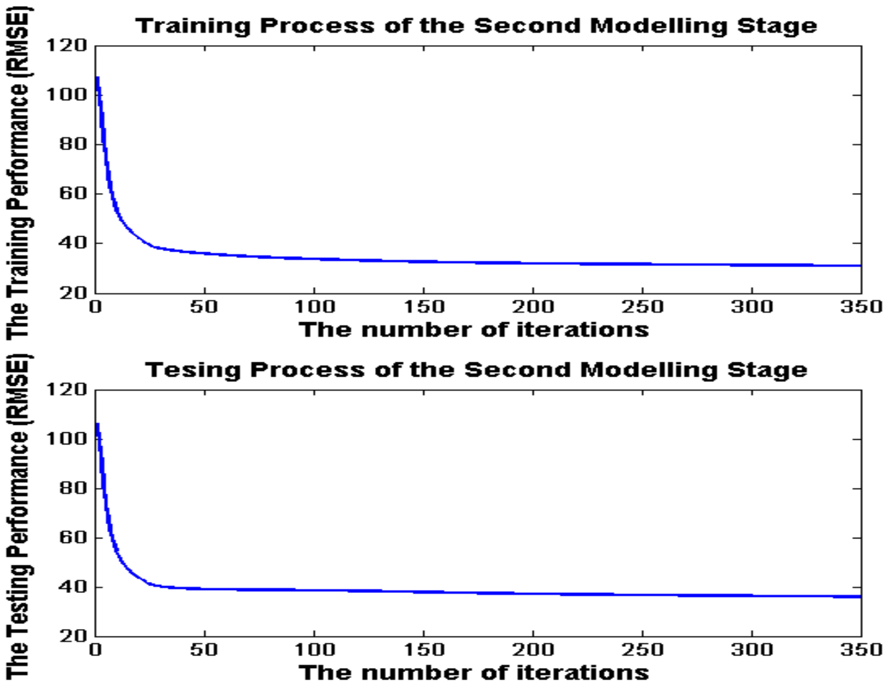


Figure 6.35 The training and testing process of the IMOFM_S on the UTS data.

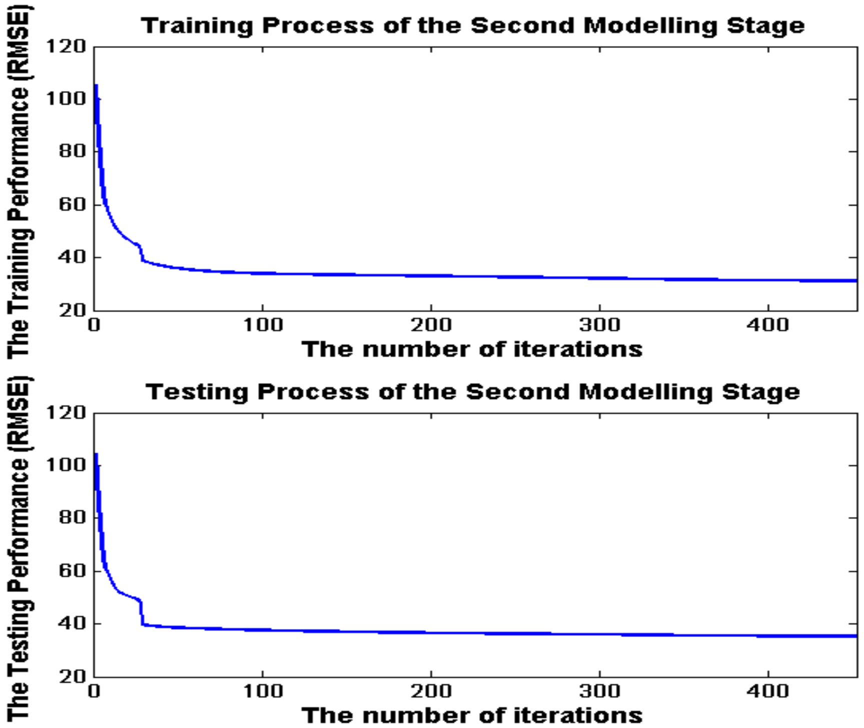


Figure 6.36 The training and testing process of the IMOFM_M on the UTS data.

TABLE 6.4

COMPARISONS OF THE PREDICTIVE PERFORMANCE FOR THE DIFFERENT MODELING METHODS USING THE UTS DATA

Modeling Methods (Ref.)	No. of rules	No. of fuzzy sets in inputs and output	Performance			
			Training (RMSE)	Testing (RMSE)	Training (R-Square)	Testing (R-Square)
Q. Zhang & M. Mahfouf (2007)						
Initial FRBS	12	Input:[12 12 12 12 12 12 12 12 12 12 12 12 12 12 12]; output: 12	100.54^{*}	108.26^{*}	-	-
Pareto FRBS1	12	Input: [9 11 10 12 8 10 8 9 10 10 6 11 10 10 10]; output: 10	$37.45^{\#}$	$43.07^{\#}$	-	-
Pareto FRBS2	9	Input: [7 8 7 8 5 6 4 6 8 8 2 6 7 8 7]; output: 9	$42.82^{\#}$	$43.90^{\#}$	-	-
IMOFM_S (NB: Average results over 10 runs are presented here)						
Average execution time (Intel(R) Core(TM)2 Duo CPU, 2.27 GHz): 1st stage: 1 min.; 2nd stage: 25 min.; 3rd stage: 3.7 hours						
Initial FRBS	12	Input: [12 12 12 12 12 12 12 12 12 12 12 12 12 12 12]; output: 12	$(113.54^{*}/ 30.93^{@})$	$(112.32^{*}/ 35.65^{@})$	$(0.6842^{*}/ 0.9826^{@})$	$(0.6181^{*}/ 0.9699^{@})$
Pareto FRBS1	11	Input: [8 11 10 11 11 11 8 11 10 11 6 10 10 11 11]; output: 11	$29.671^{\#}$	$34.42^{\#}$	$0.9830^{\#}$	$0.9709^{\#}$
Pareto FRBS2	10	Input: [4 7 8 8 4 7 3 8 7 7 3 4 4 7 7]; output: 10	$32.376^{\#}$	$34.82^{\#}$	$0.9798^{\#}$	$0.9702^{\#}$
Pareto FRBS3	10	Input: [4 7 8 8 4 6 3 8 7 7 3 4 4 7 7]; output: 10	$32.656^{\#}$	$35.19^{\#}$	$0.9794^{\#}$	$0.9696^{\#}$
Pareto FRBS4	10	Input: [4 7 8 8 4 7 3 7 6 6 2 3 3 7 7]; output: 10	$34.104^{\#}$	$35.80^{\#}$	$0.9775^{\#}$	$0.9686^{\#}$
Pareto FRBS5	9	Input: [4 5 6 6 2 4 3 6 5 6 2 2 3 7 6]; output: 9	$34.512^{\#}$	$35.96^{\#}$	$0.9770^{\#}$	$0.9683^{\#}$
Pareto FRBS6	8	Input: [2 4 4 7 3 4 3 5 4 5 2 2 3 6 6]; output: 8	$35.663^{\#}$	$37.62^{\#}$	$0.9754^{\#}$	$0.9675^{\#}$
Pareto FRBS7	8	Input: [2 4 4 7 3 3 3 5 4 5 2 2 3 6 6]; output: 8	$36.429^{\#}$	$37.62^{\#}$	$0.9743^{\#}$	$0.9654^{\#}$
Pareto FRBS8	8	Input: [3 5 4 4 1 4 2 6 4 6 2 2 3 4 4]; output: 8	$39.041^{\#}$	$40.12^{\#}$	$0.9705^{\#}$	$0.9603^{\#}$
Pareto FRBS9	7	Input: [3 4 4 4 1 3 3 4 3 4 1 1 2 6 5]; output: 7	$42.914^{\#}$	$43.87^{\#}$	$0.9642^{\#}$	$0.9523^{\#}$
Pareto FRBS10	6	Input: [3 2 2 4 1 3 2 4 3 3 1 1 1 3 5]; output: 6	$45.061^{\#}$	$43.48^{\#}$	$0.9605^{\#}$	$0.9531^{\#}$
IMOFM_M (NB: Average results over 10 runs are presented here)						
Average execution time (Intel(R) Core(TM)2 Duo CPU, 2.27 GHz): 1st stage: 1 min.; 2nd stage: 30 min.; 3rd stage: 4 hours						
Initial FRBS	12	Input: [12 12 12 12 12 12 12 12 12 12 12 12 12 12 12]; output: 12	$(120.43^{*}/ 31.21^{@})$	$(123.44^{*}/ 35.49^{@})$		
Pareto FRBS1	11	Input: [8 9 10 10 5 11 5 10 10 8 4 6 6 11 10 11]; output: 11	$30.914^{\#}$	$34.49^{\#}$	$0.9810^{\#}$	$0.9739^{\#}$
Pareto FRBS2	10	Input: [8 9 10 10 6 10 6 9 9 7 4 7 6 10 9]; output: 10	$31.210^{\#}$	$35.32^{\#}$	$0.9806^{\#}$	$0.9726^{\#}$
Pareto FRBS3	10	Input: [6 8 9 10 6 8 6 9 9 7 3 4 5 10 9]; output: 9	$31.231^{\#}$	$35.13^{\#}$	$0.9806^{\#}$	$0.9728^{\#}$
Pareto FRBS4	10	Input: [5 8 9 10 5 8 5 8 9 7 1 4 4 10 9]; output: 9	$31.421^{\#}$	$35.06^{\#}$	$0.9804^{\#}$	$0.9730^{\#}$
Pareto FRBS5	9	Input: [5 7 7 9 4 4 2 7 7 7 1 5 4 9 9]; output: 7	$33.503^{\#}$	$36.09^{\#}$	$0.9777^{\#}$	$0.9715^{\#}$
Pareto FRBS6	8	Input: [6 7 7 7 2 5 4 6 8 7 2 4 2 7 7]; output: 7	$33.683^{\#}$	$36.68^{\#}$	$0.9774^{\#}$	$0.9705^{\#}$

Table 6.4 to be continued...

Table 6.4 continued...

Pareto FRBS7	8	Input: [5 7 7 8 3 5 3 6 7 6 0 4 3 8 7]; output: 6	34.390[#]	37.42[#]	0.9764[#]	0.9692[#]
Pareto FRBS8	8	Input: [5 6 7 8 3 5 3 6 7 6 0 4 3 8 7]; output: 6	34.468[#]	37.44[#]	0.9763[#]	0.9692[#]
Pareto FRBS9	7	Input: [5 7 7 7 2 4 3 6 6 6 2 3 1 7 7]; output: 5	34.703[#]	36.44[#]	0.9760[#]	0.9708[#]
Pareto FRBS10	6	Input: [1 3 2 3 1 3 1 3 2 4 2 1 2 4 4]; output: 5	46.469[#]	45.12[#]	0.9567[#]	0.9548[#]

[*] Initial model extracted directly from data using clustering algorithms or grid partition methods.

[@] Refined model or the consequents are computed through the estimation methods.

[#] Simplified model after model simplification and parameter fine tuning.

As can be seen from Table 6.4, the proposed algorithm generally gives more accurate predictions with the same number of rules. More importantly, if one closely insprects the number of fuzzy sets involved in each input for the Zhang & Mahfouf's method and IMOFM, one could find that less fuzzy sets are needed for IMOFM, which leads to an even more simplified (transparent) structure. Due to constraints on space, only a few obtained 'Pareto' FRBSs from 10 runs are presented in Table 6.4 without any loss of generality.

The 'actual outputs vs. predicted outputs' plots from the three modelling stages are shown in Figures 6.37~6.38. The first two stages led to two 'vaccine FRBSs' both consisting of a maximum of 12 rules. The third stage produced a set of Pareto FRBSs and only an 8-rule simplified Singleton fuzzy model and a 7-rule simplified Mamdani Fuzzy model are chosen for the subsequent illustration purposes. As shown in Figures 6.37~6.38, although the initial FRBS extracted by G3Kmeans is not accurate, it does however capture the basic structure of the training data. After the second modelling stage, the model's predictive performance is improved so that the elicited fuzzy model is ready for use as far as its predictive accuracy is concerned. Furthermore, since 12 membership functions are involved in each input at the second modelling stage the rule-base's structure is very complex to interpret in terms of linguistic terms. A further simplification for the rule base can still be applied if one is trying to use this model to understand the underlaying behaviour of the system.

Without any prior knowledge as to how to simplify the model and to what degree, the proposed method provides a set of FRBSs after the third modelling stage, which represents various degrees of simplification. Among these options and after the inspection of the trade-off of these elicited FRBSs, the users can finally realise what degree of model simplification is the one that they really need. Figures 6.39~6.40 show the Pareto fronts of the UTS modelling problem using IMOFM_S (41 Pareto FRBSs) and IMOFM_M (47 Pareto FRBSs) from one of the 10 runs.

Figures 6.41~6.46 show how the initially elicited 12-rule 'vaccine FRBSs' from the first two modelling stages with highly overlapped membership functions are simplified to an 8-rule Singleton and a 7-rule Mamdani FRBSs with fewer interpretable fuzzy sets using IMOFM_S and IMOFM_M.

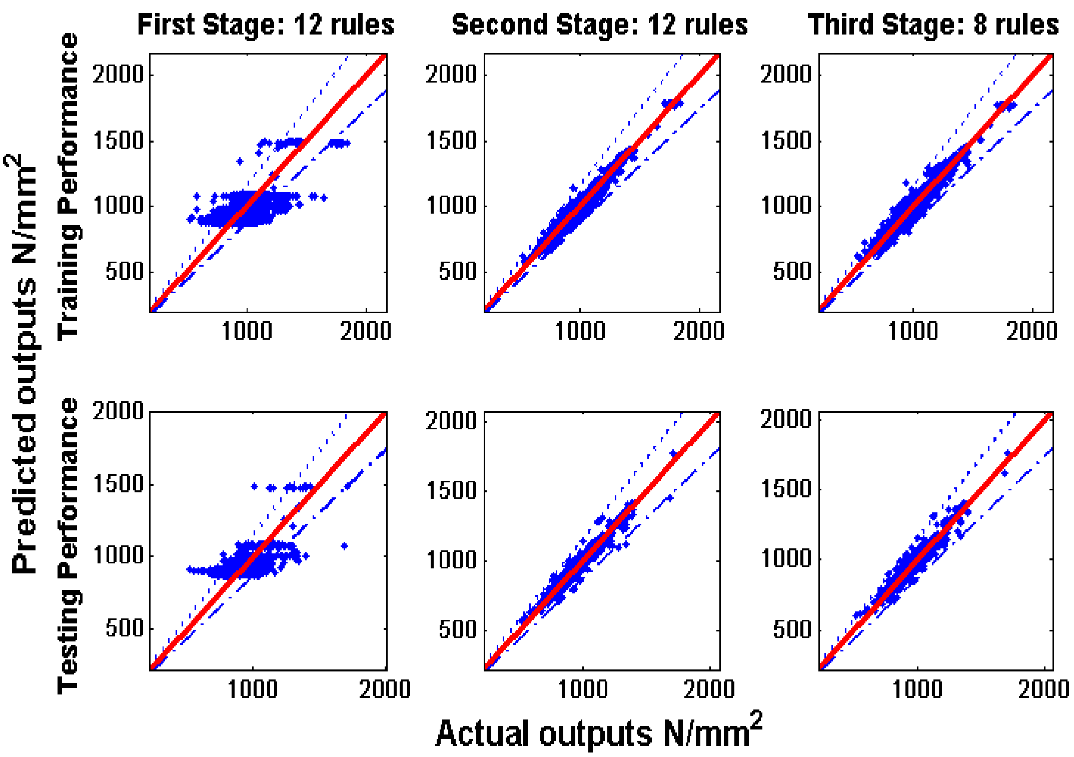


Figure 6.37 The prediction performances of the three stages for the training and testing data using IMOFM_S.

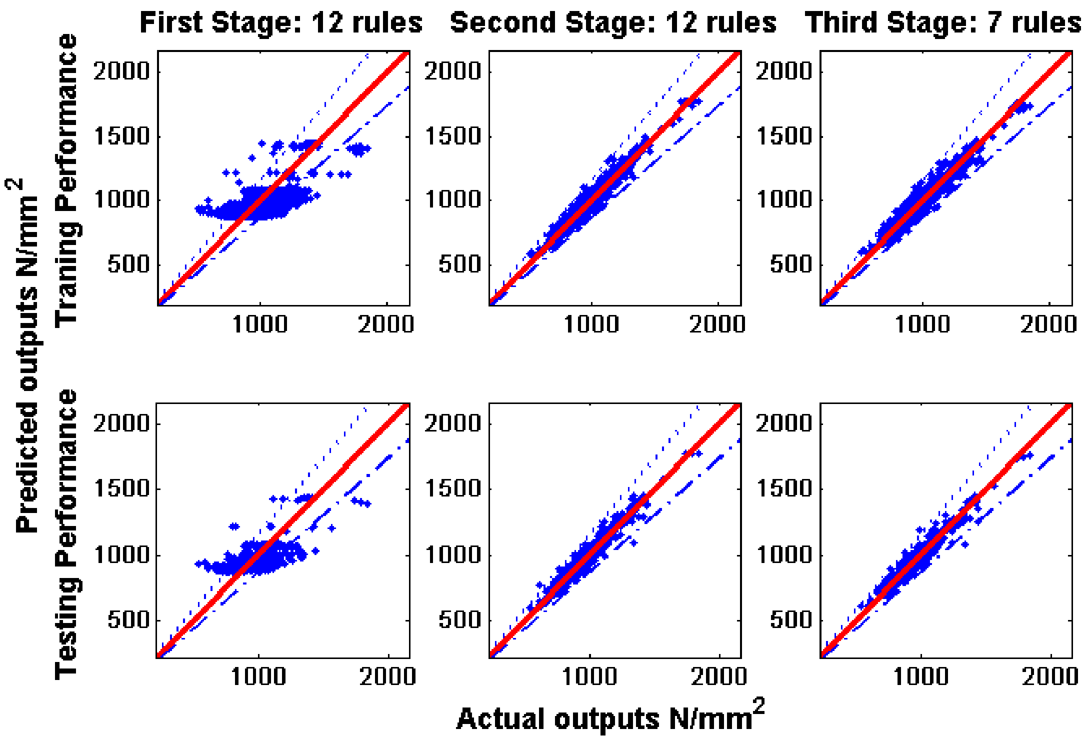


Figure 6.38 The prediction performances of the three stages for the training and testing data using IMOFM_M.

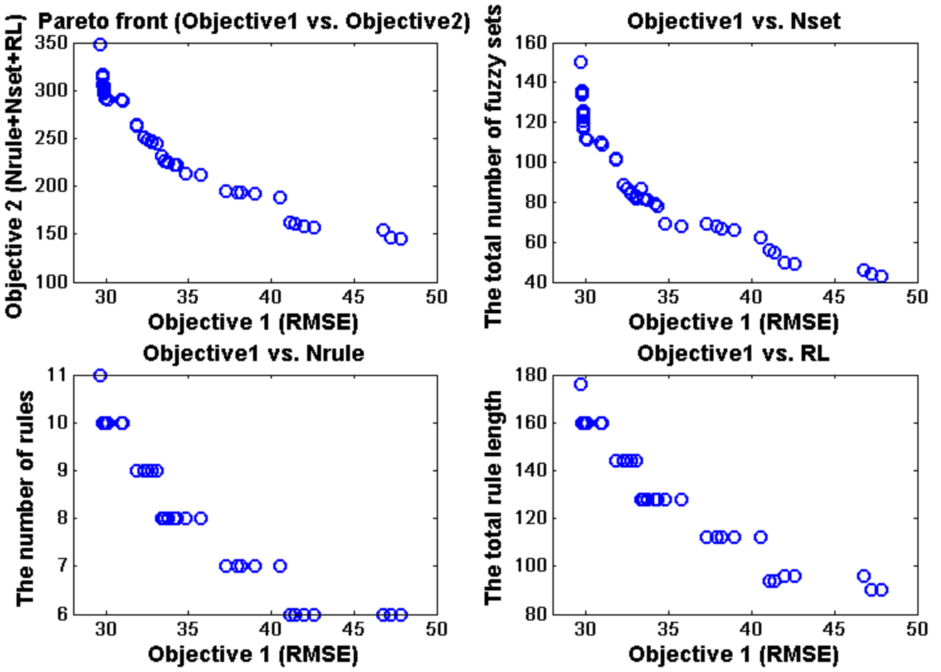


Figure 6.39 The Pareto fronts obtained using IMOFM_S from the third modelling procedure for the UTS modelling problem: (a) Objective1 vs. Objective2; (b) Objective1 vs. Nset; (c) Objective1 vs. Nrule; (d) Objective1 vs. RL.

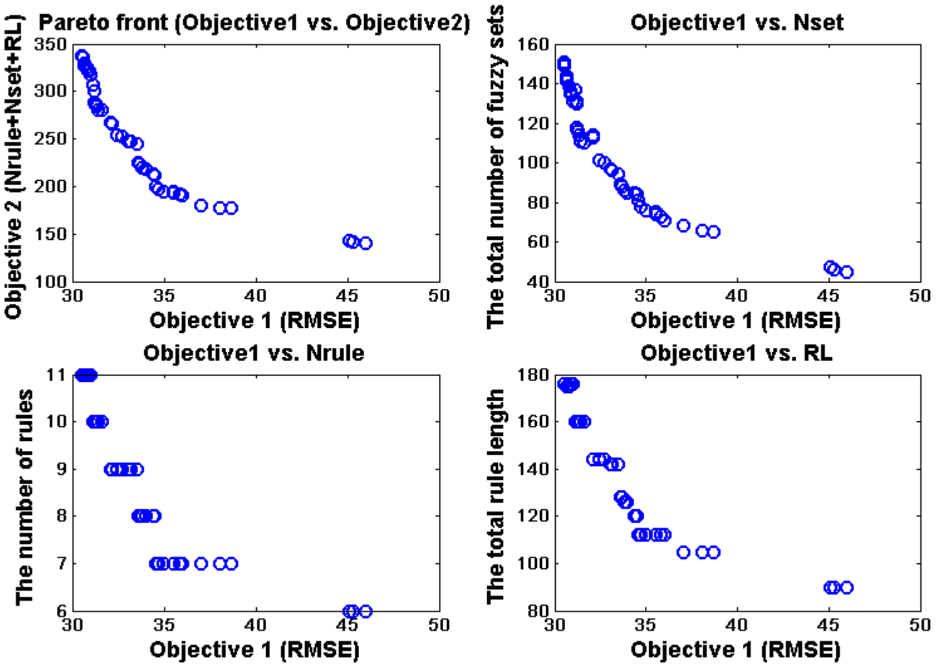


Figure 6.40 The Pareto fronts obtained using IMOFM_M from the third modelling procedure for the UTS modelling problem: (a) Objective1 vs. Objective2; (b) Objective1 vs. Nset; (c) Objective1 vs. Nrule; (d) Objective1 vs. RL.

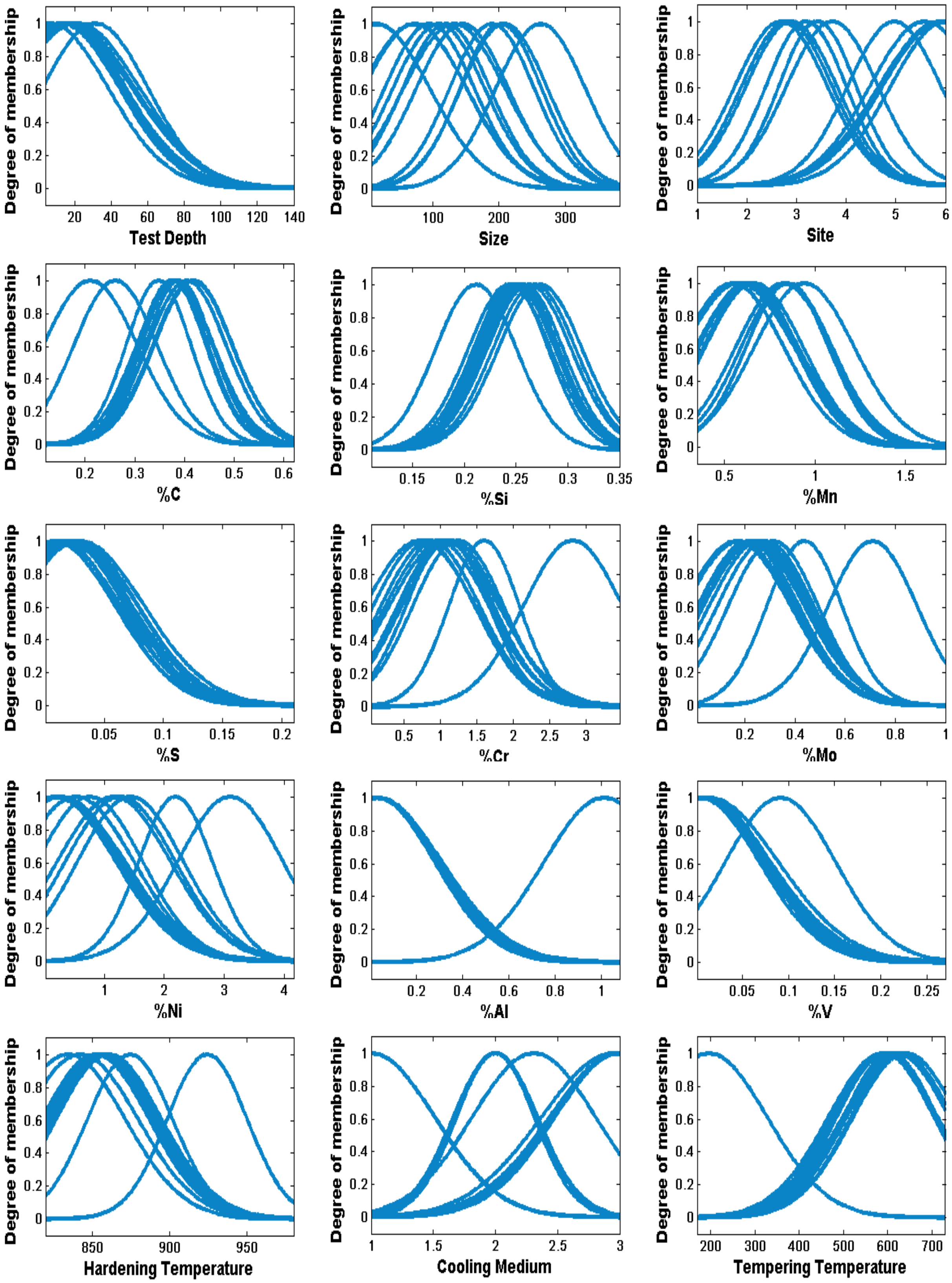


Figure 6.41 The distribution of membership functions of the 12-rule initial Singleton FRBS (from the first modelling stage) for UTS modelling.

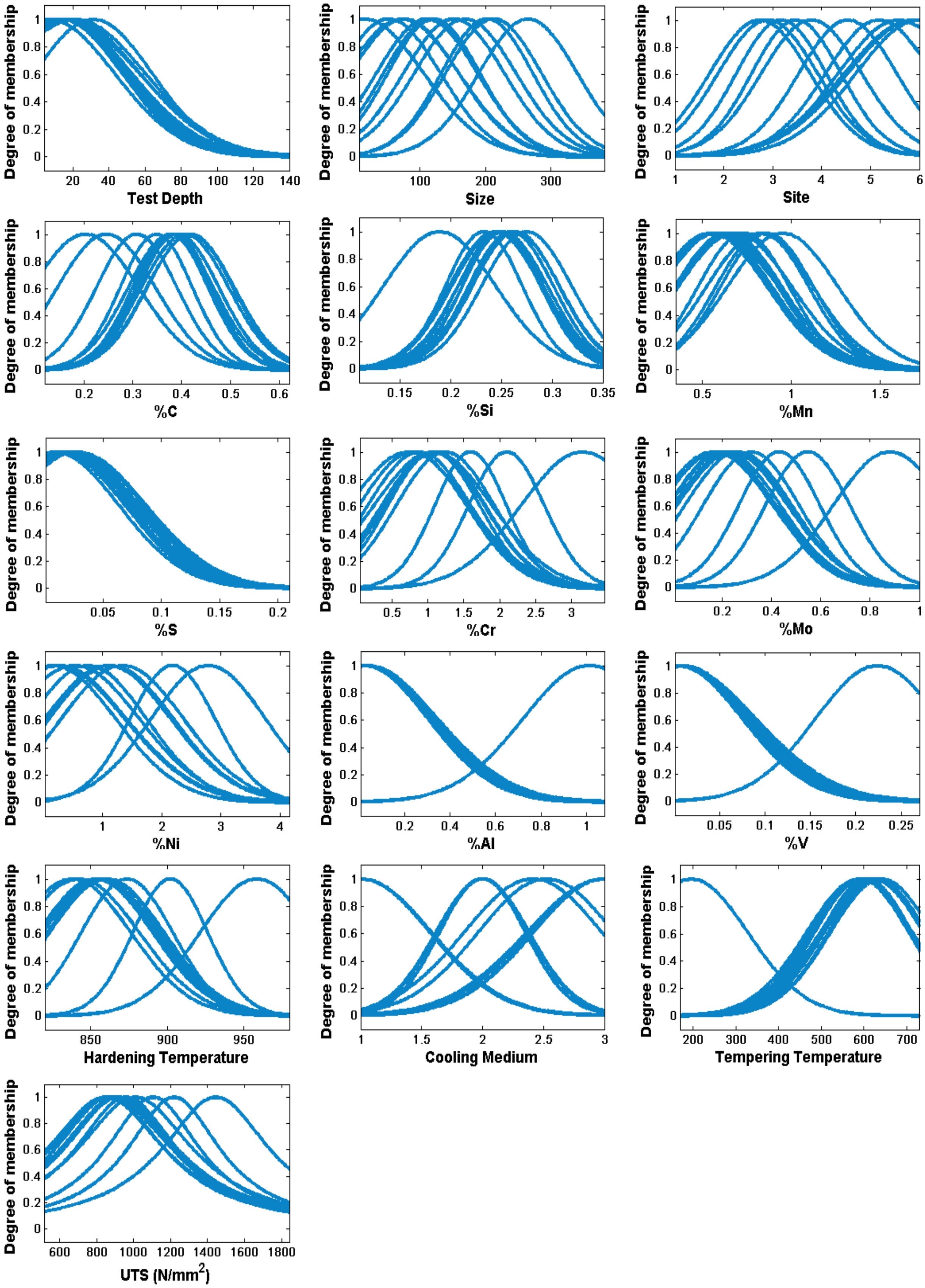


Figure 6.42 The distribution of membership functions of the 12-rule initial Mamdani FRBS (from the first modelling stage) for UTS modelling.

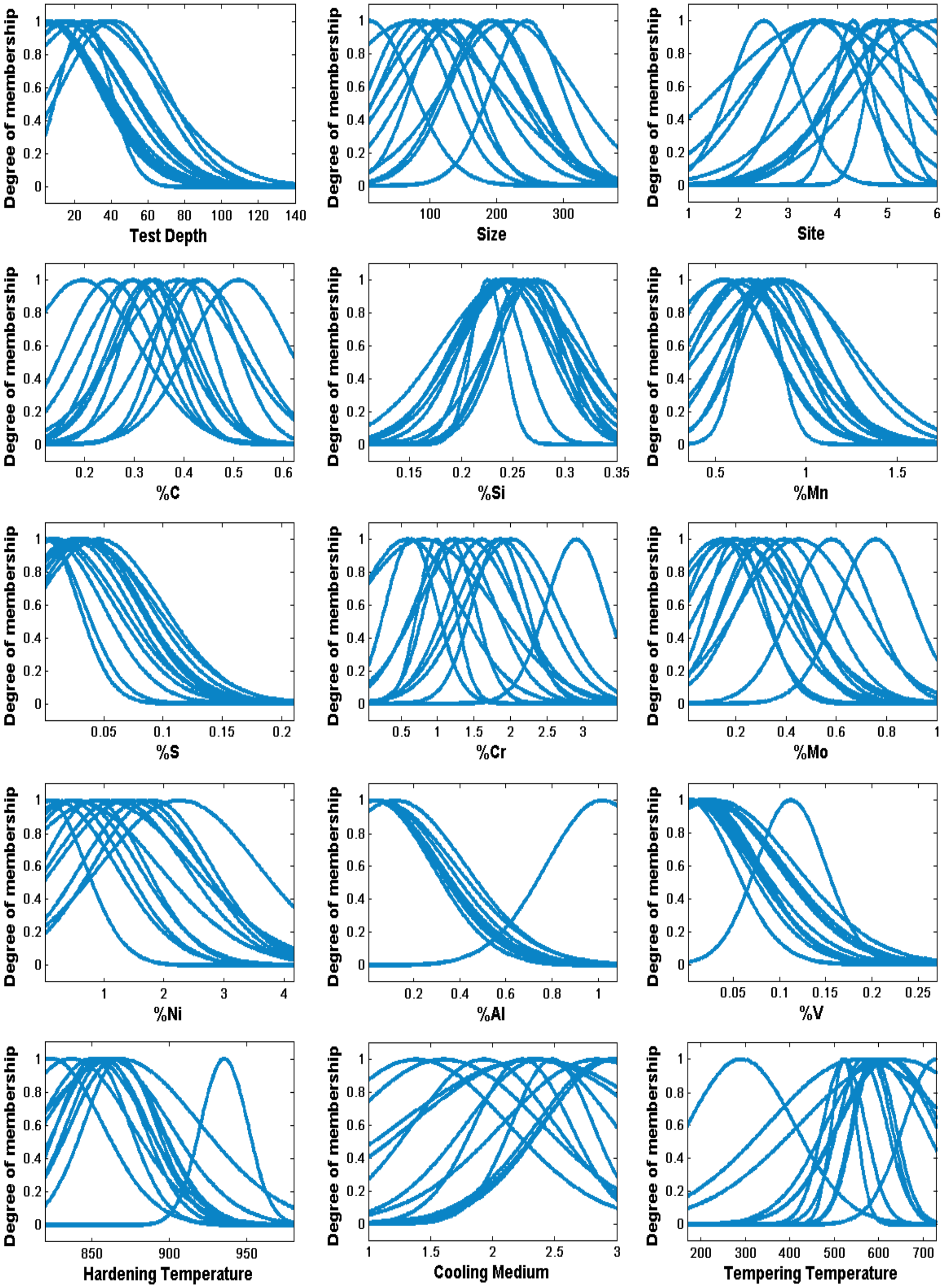


Figure 6.43 The distribution of membership functions of the 12-rule refined Singleton FRBS (from the second modelling stage) for UTS modelling.

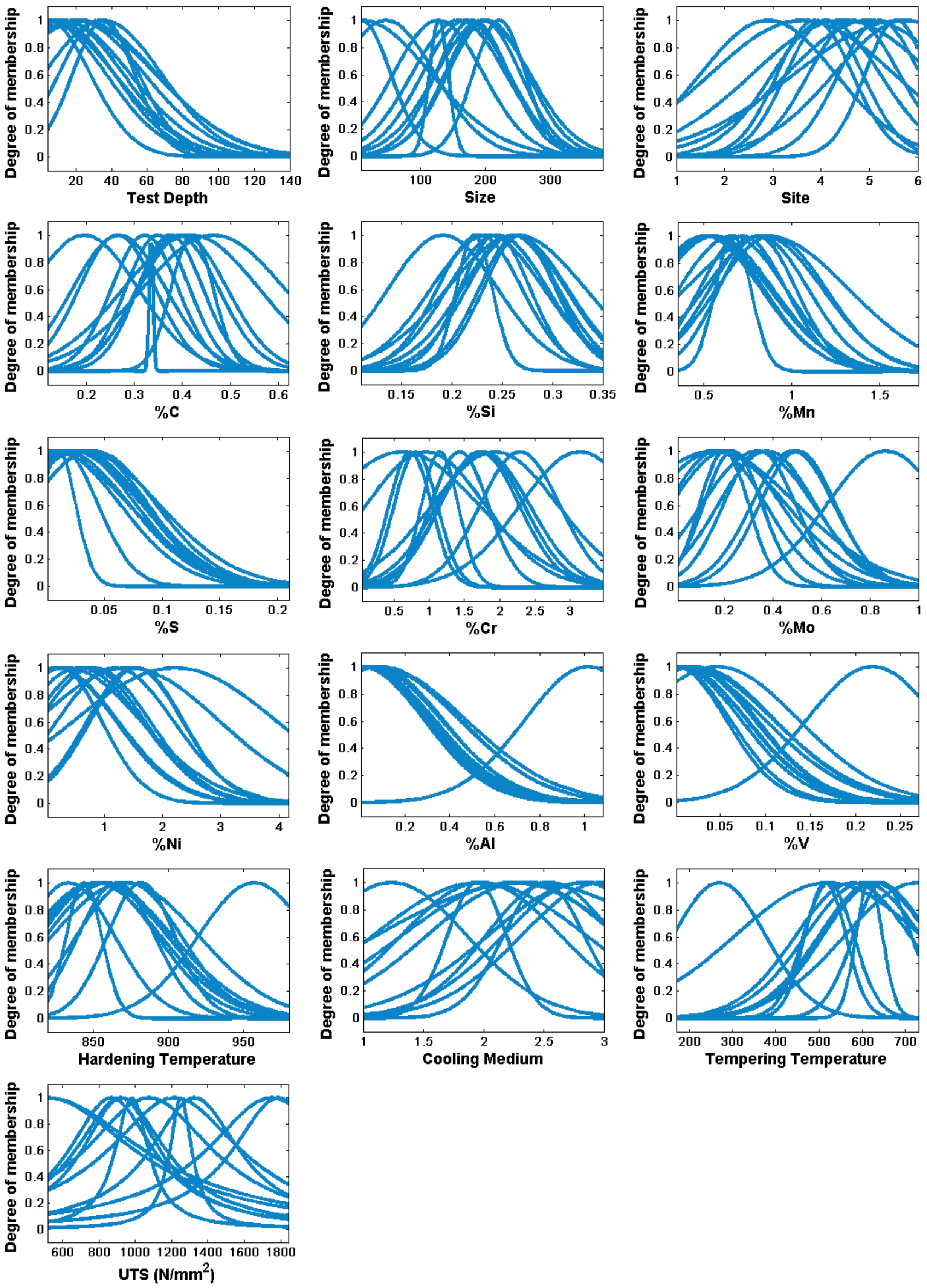


Figure 6.44 The distribution of membership functions of the 12-rule refined Mamdani FRBS (from the second modelling stage) for UTS modelling.

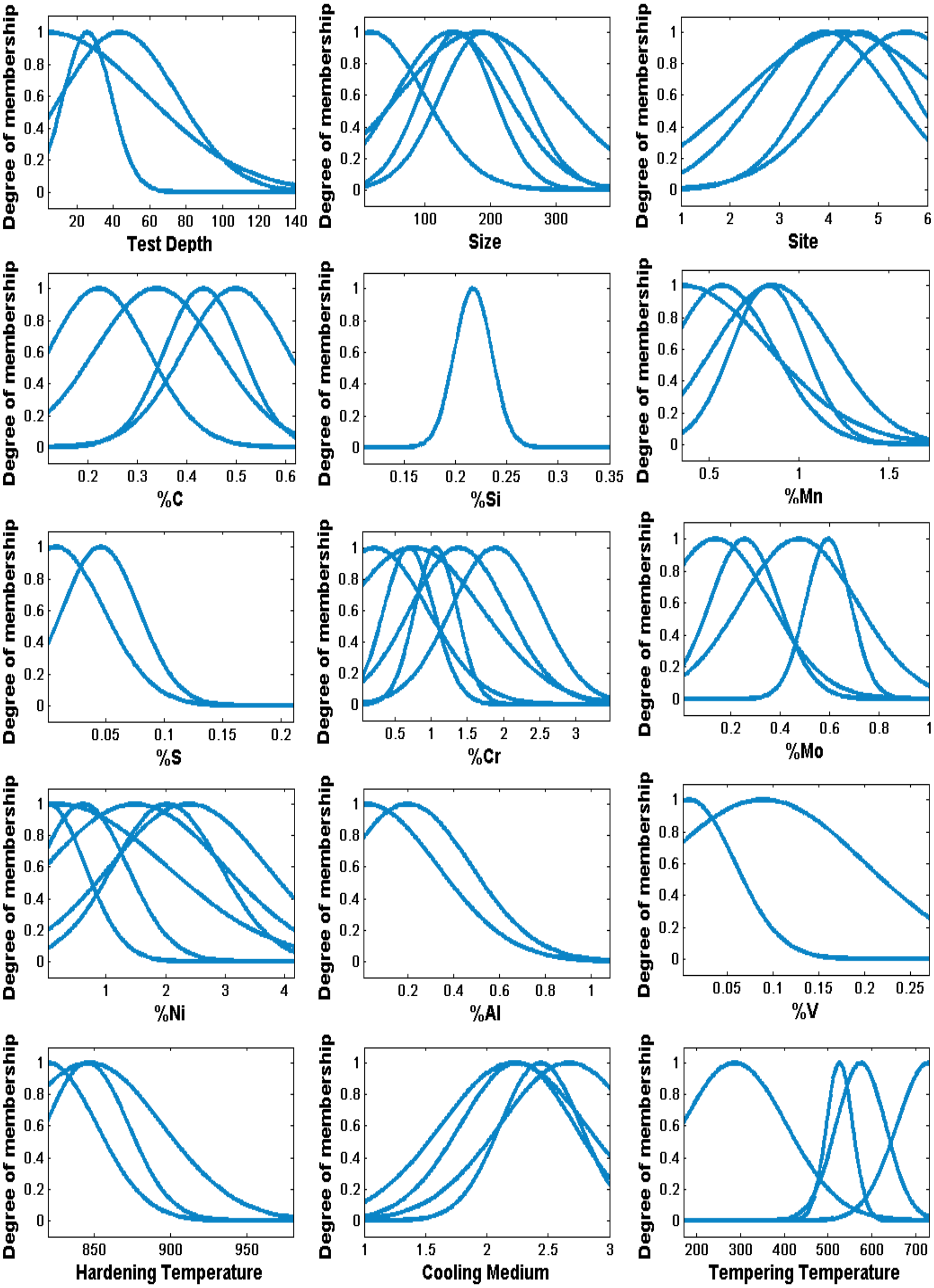


Figure 6.45 The distribution of membership functions of the 8-rule simplified Singleton FRBS (from the third modelling stage) for UTS modelling.

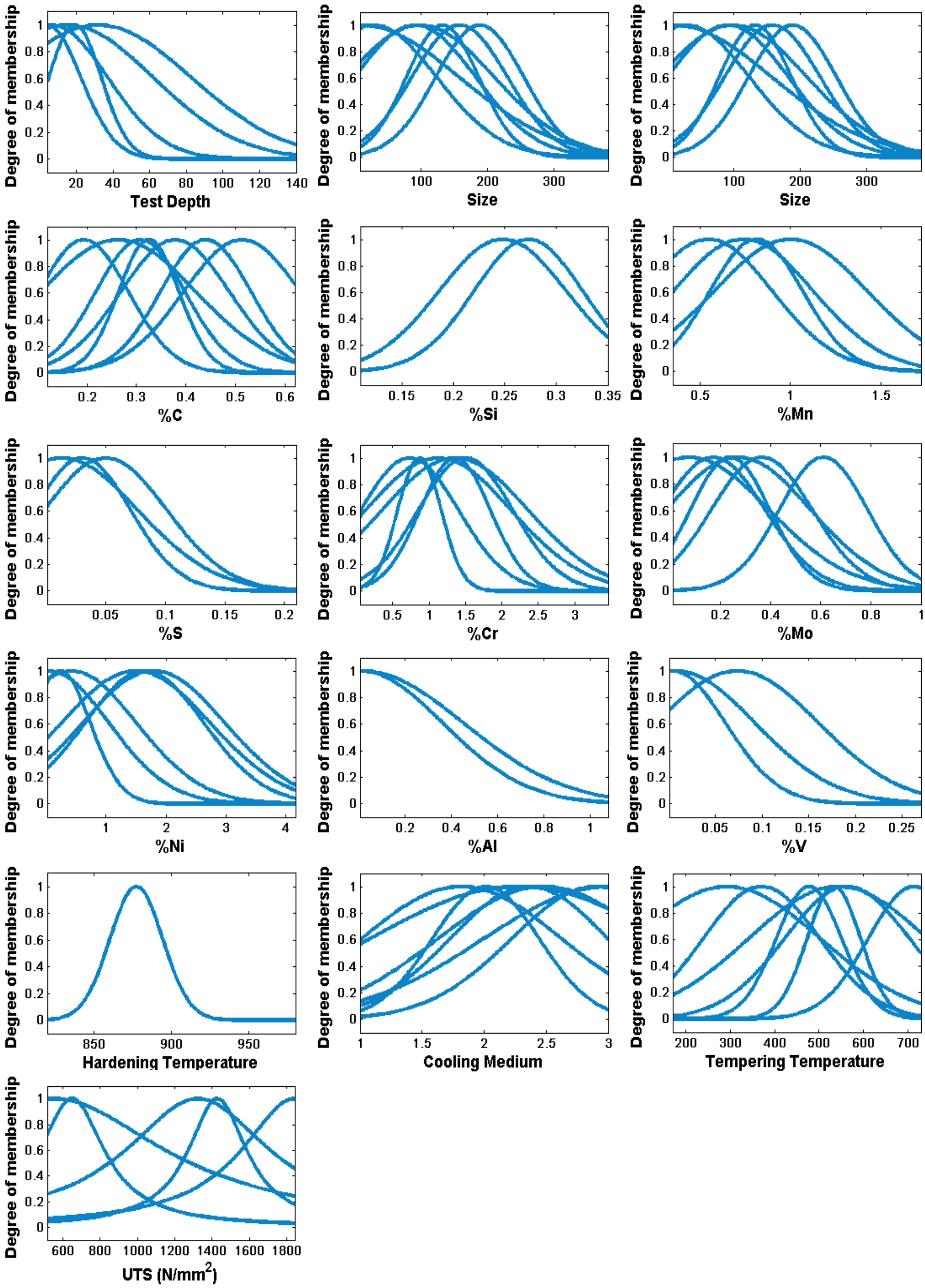


Figure 6.46 The distribution of membership functions of the 7-rule simplified Mamdani FRBS (from the third modelling stage) for UTS modelling.

Figure 6.47 shows 4 selected rules from the 8-rule simplified Singleton FRBS.

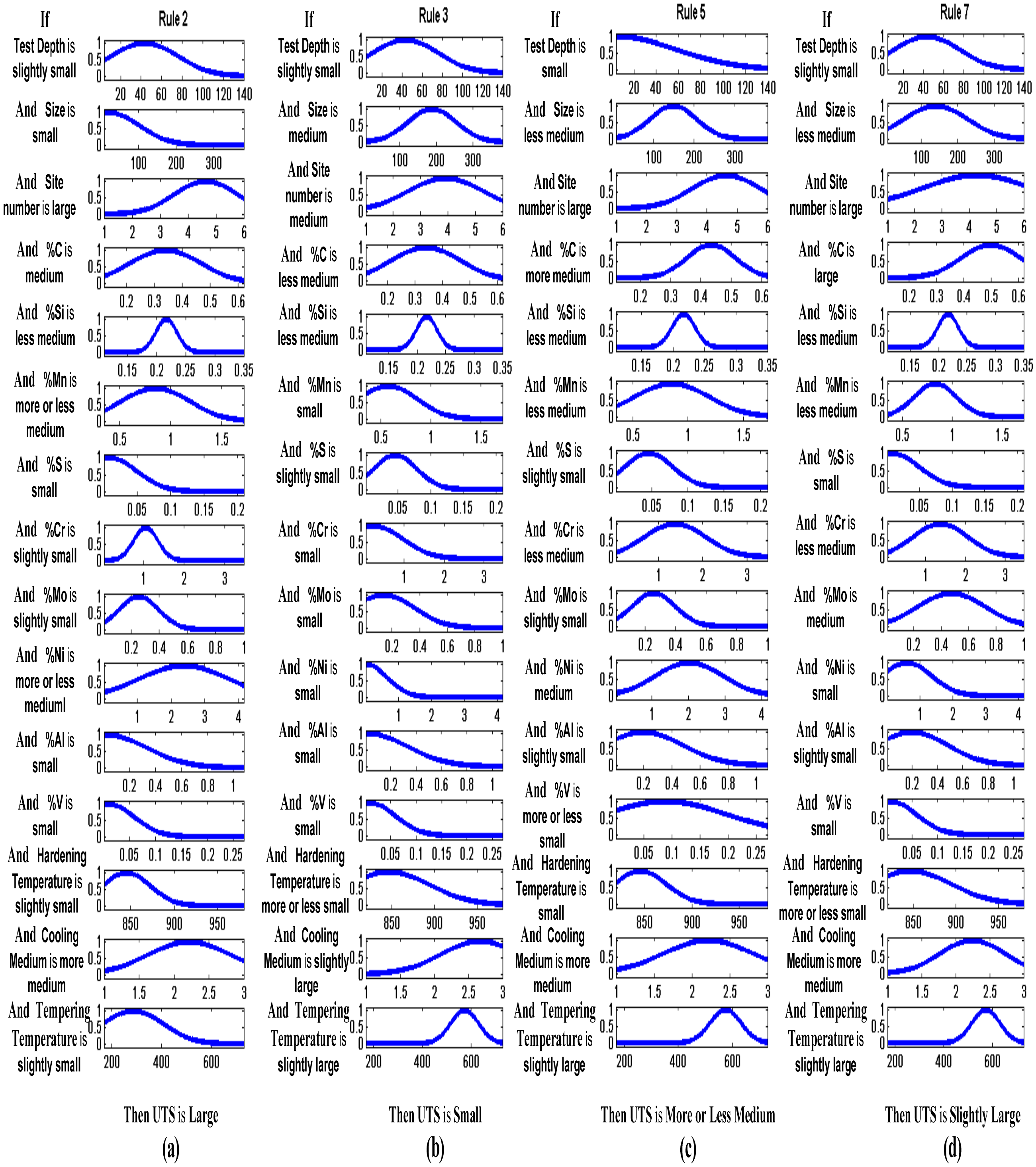


Figure 6.47 The selected rules from the 8-rule simplified Singleton FRBS: (a) rule 2; (b) rule 3; (c) rule 5; (d) rule 7.

Figure 6.48 shows 3 selected rules from the 7-rule simplified Mamdani FRBS.

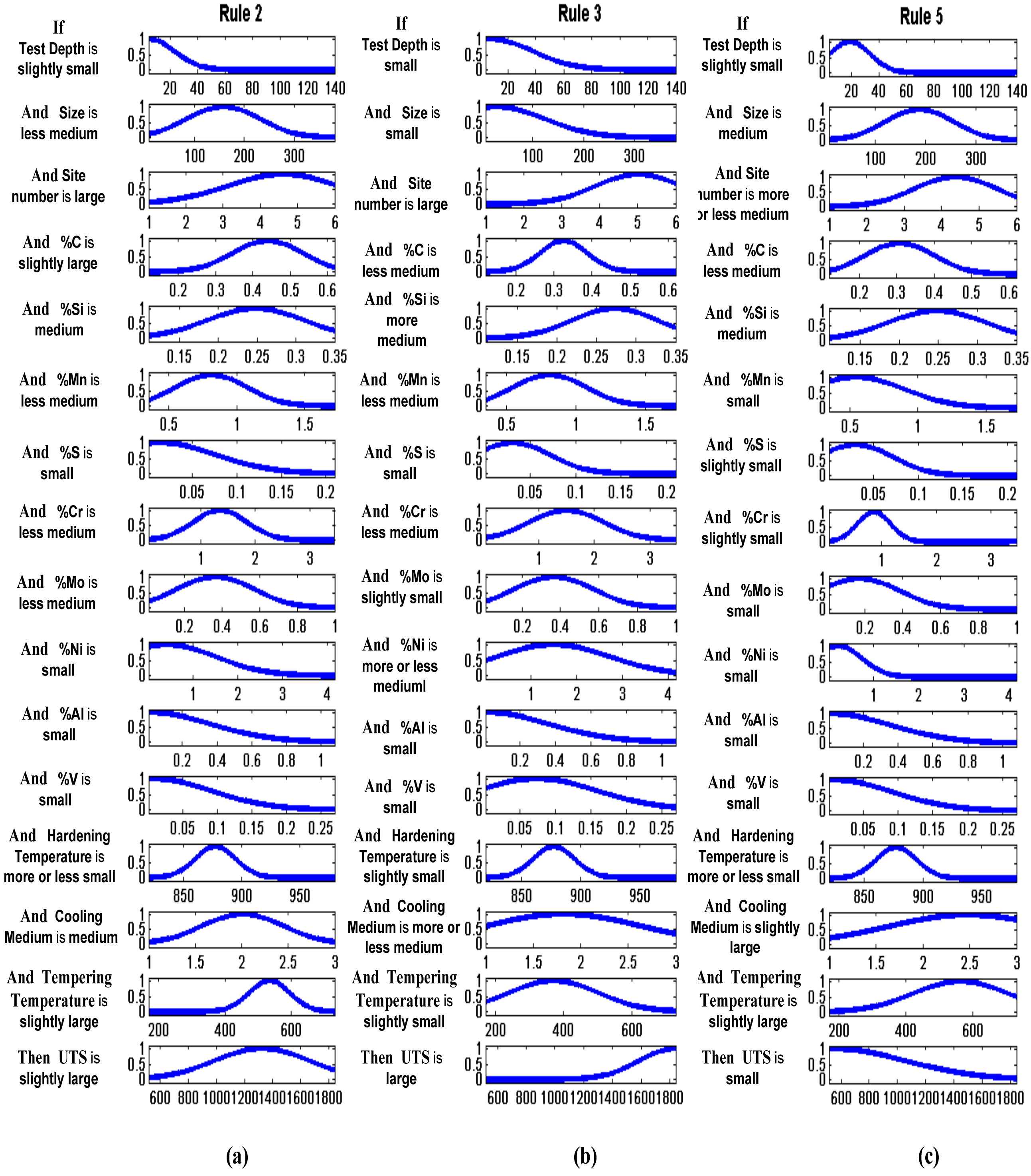


Figure 6.48 The selected rules from the 7-rule simplified Mamdani FRBS: (a) rule 2; (b) rule 3; (c) rule 5.

For such a high dimensional problem, verifying the physical interpretation of the obtained models is proved to be a difficult task. Hence, Figures 6.49~6.50 show the three-dimensional response surfaces of the UTS models by plotting two varying input variables against the output while keeping other input variables constant. The constant variables are set to the 'median' values of the dominant steel grade, as indicated in Table 6.5, which corresponds to 1%CrMo steel grade. It is worth mentioning that the knowledge conveyed by Figures 6.49~6.50 is rather consistent with the knowledge extracted by Tenner (1999) and Zhang & Mahfouf (2009), which has been verified to follow the expected behaviour as predicted by theory or by expert knowledge. As an example, one interesting finding that has been consistently mined via the composition based variable effect method (Tenner, 1999) and the response surface based method is the interaction effects of varying tempering temperature and carbon content. As shown in Figures 6.49~6.50, the strength of 1% CrMo steel is greatest at low tempering temperature and high carbon content, and is lowest at high tempering temperature and low carbon content. More importantly, with high carbon content, the effect of tempering temperature is much more non-linear than the one with low carbon content. Similar analyses can be conducted for variables to extract hidden knowledge.

TABLE 6.5
THE 'MEDIAN' MODEL INPUTS REPRESENTING THE 1%CRMO STEEL GRADE

The 'Median' Inputs of 1% CrMo Steel Grade	
Input Variables	**Values**
Test Depth (mm)	12.7
Size (mm)	180
Site (1-6)	3
C (%)	0.41
Si (%)	0.27
Mn (%)	0.78
S (%)	0.023
Cr (%)	1.08
Mo (%)	0.22
Ni (%)	0.19
Al (%)	0.027
V (%)	0.005
Hardening Temperature (^{0}C)	860
Cooling Medium (1-3)	3
Tempering Temperature (^{0}C)	630

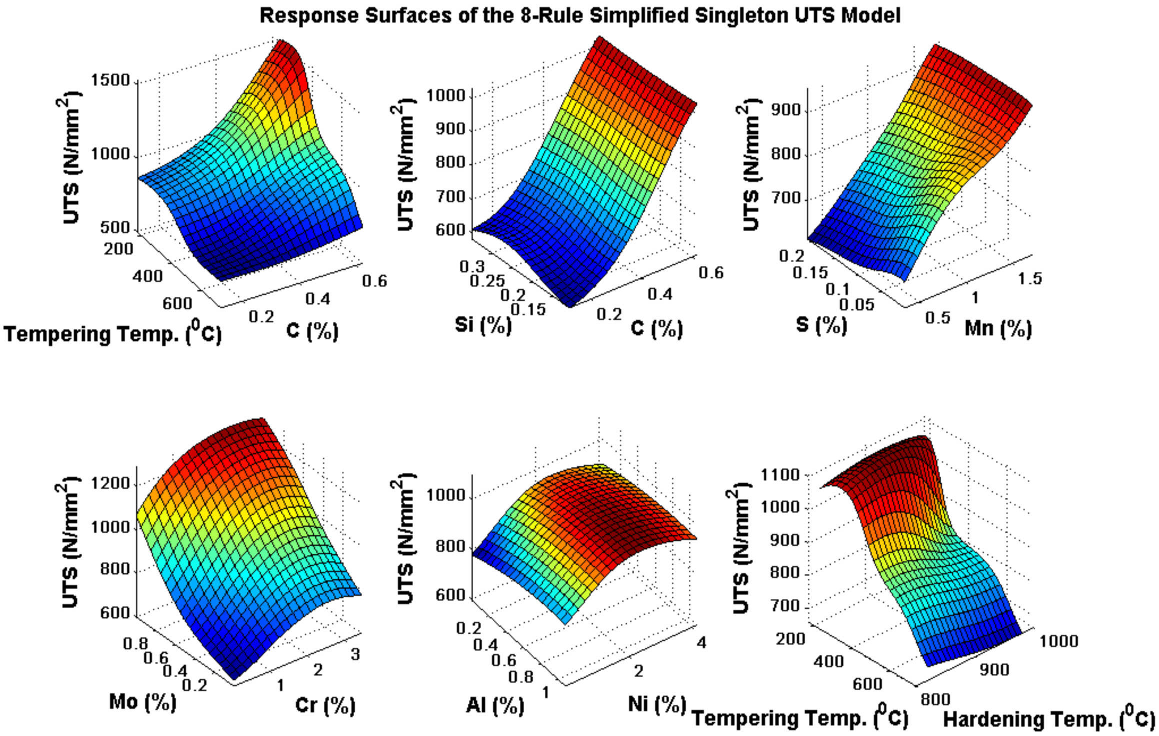


Figure 6.49 Response surfaces of the 8-rule simplified Singleton UTS model.

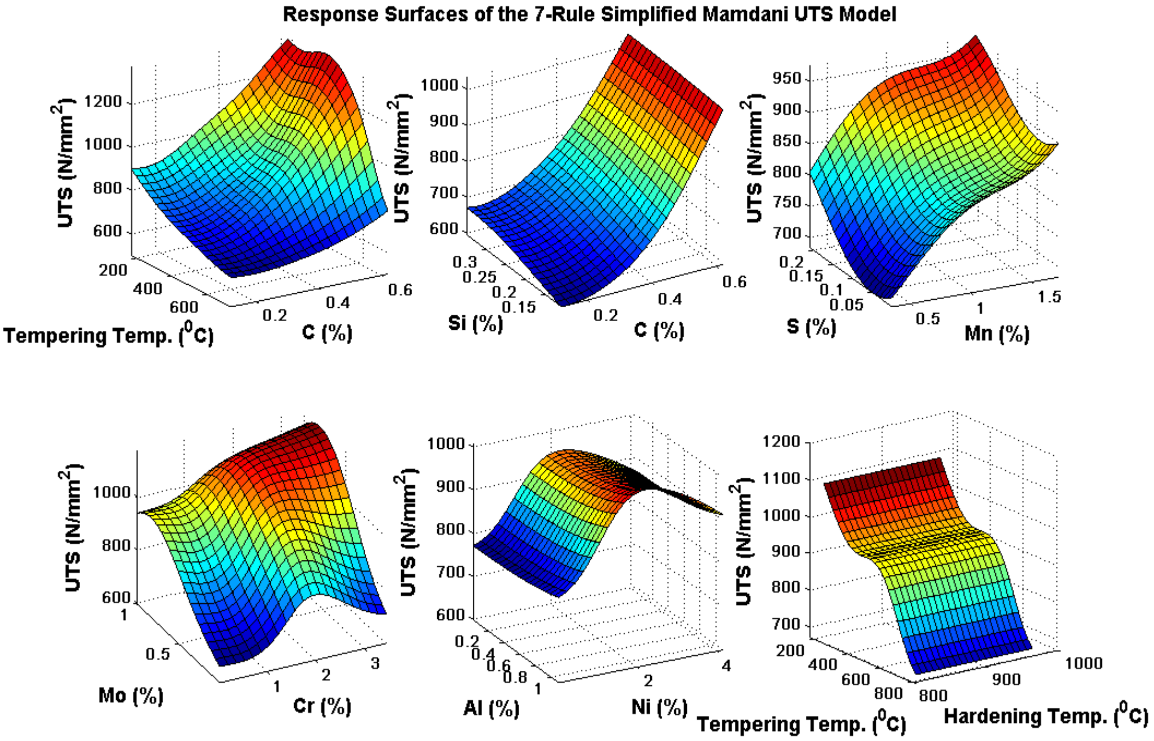


Figure 6.50 Response surfaces of the 7-rule simplified Mamdani UTS model.

As shown in Table 6.4, the 'vaccine FRBSs' produced from the second modelling stage are accurate as far as the training data is concerned and generalise well under similar situations represented by the testing data. However, such a good generalisation performance does not guarantee the same level of goodness for the 'vaccine models' under unseen situations. Hence, another 12 more recent samples were used as unseen examples for further investigating the generalisation ability of the elicited fuzzy models. By doing so, the problem of over-fitting specifically related to the second modelling stage (vaccine FRBS) is further revealed in Table 6.6. Such over-fitting is mainly attributed to the excessive number of rules and membership functions involved in the first two modelling stages. By employing the third modelling stage, these unnecessary parts are pruned so that the simplified fuzzy models can predict well even under unknown scenarios.

TABLE 6.6
THE VALIDATION PERFORMANCES OF IMOFM ON THE 12 UNSEEN DATA SAMPLES

Modeling Methods	Second Stage (single objective refining)		
	No. of rules	Validation (RMSE)	
IMOFM_S	12	53.62	
IMOFM_M	12	47.34	
Third Stage (multi-objective fuzzy modeling)			
Modeling Methods	**No. of rules**	**No. of Fuzzy sets in inputs**	Validation (RMSE)
IMOFM_S			
Pareto FRBS1	10	Inputs: [4 7 8 8 4 7 3 8 7 7 3 4 4 7 7], Outputs: 10	41.01
Pareto FRBS2	8	Inputs: [2 4 4 7 3 3 3 5 4 5 2 2 3 6 6], Output: 8	31.54
Pareto FRBS3	7	Inputs: [3 4 4 4 1 3 3 4 3 4 1 1 2 6 5], Output: 7	46.34
IMOFM_M			
Pareto FRBS1	10	Inputs: [8 9 10 10 6 10 6 9 9 7 4 7 6 10 9], Output: 10	35.65
Pareto FRBS2	7	Inputs: [5 7 7 7 2 4 3 6 6 6 2 3 1 7 7], Output: 5	37.80
Pareto FRBS3	6	Inputs: [2 2 2 5 2 2 1 4 3 3 0 2 1 2 4], Output: 5	49.87

Figure 6.51 shows the prediction performances of the 'vaccine FRBSs' and the simplified FRBSs on the validation data. As indicated by the graph, the refined FRBSs obtained from the second modelling stage cannot cope with those newly collected samples as some predictions are very close to or even outside the ±10% error bands. Conversely, the generalisation capability of the simplified FRBSs is very much improved due to the simplification of the rule-base structure. Figure 6.52 shows the confidence bands the UTS models on the training data.

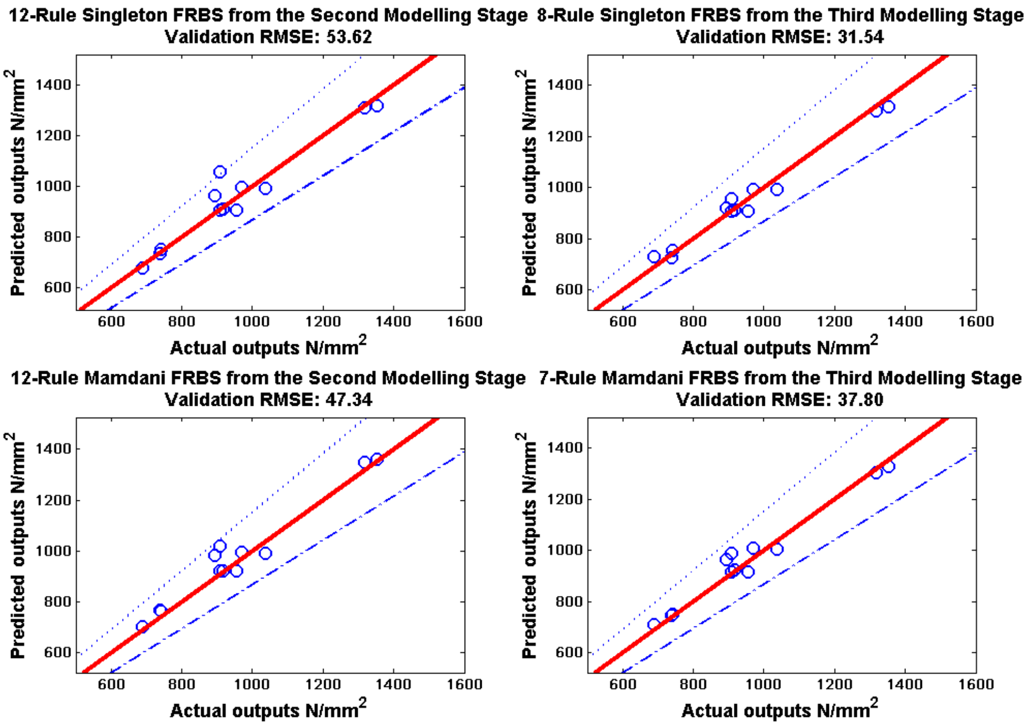


Figure 6.51 The prediction performances of the 12-rule refined FRBSs from the second modelling stage and the 8-rule and 7-rule simplified FRBSs from the third modelling stage.

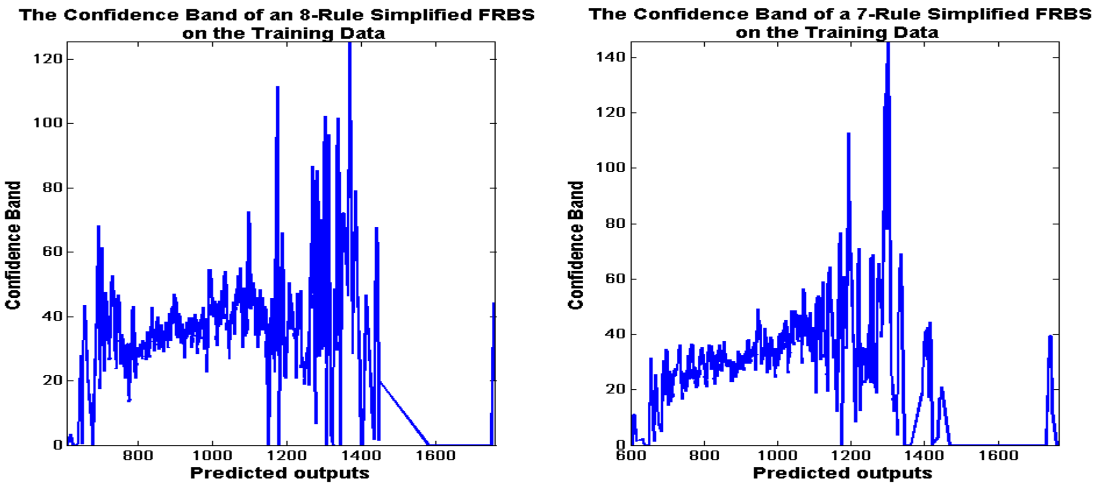


Figure 6.52 5%-range confidence bands of an 8-rule simplified Singleton FRBS (left) and a 7-rule simplified Mamdani FRBS (right) on the training data set.

As already discussed in Section 5.6.3, the variable length coding scheme and the new distance index can greatly improve the prediction performance of the simplified model and the process of optimisation. Such an improvement can be made even bigger when more rules and higher dimensional problems are involved. Figure 6.53 uses IMOFM_S as an example to demonstrate such an improvement and shows the snapshot of the approximate Pareto fronts at 10, 100, 500, 800, 1000 and 1200 iterations respectively. As can be seen from this figure, the evolution starts from the most accurate FRBS and expands the Pareto front during the course of the optimisation. The variable length coding and the new distance index play an important role in such a search process as the prediction accuracy of the simplified FRBSs at early iterations has been considerably improved.

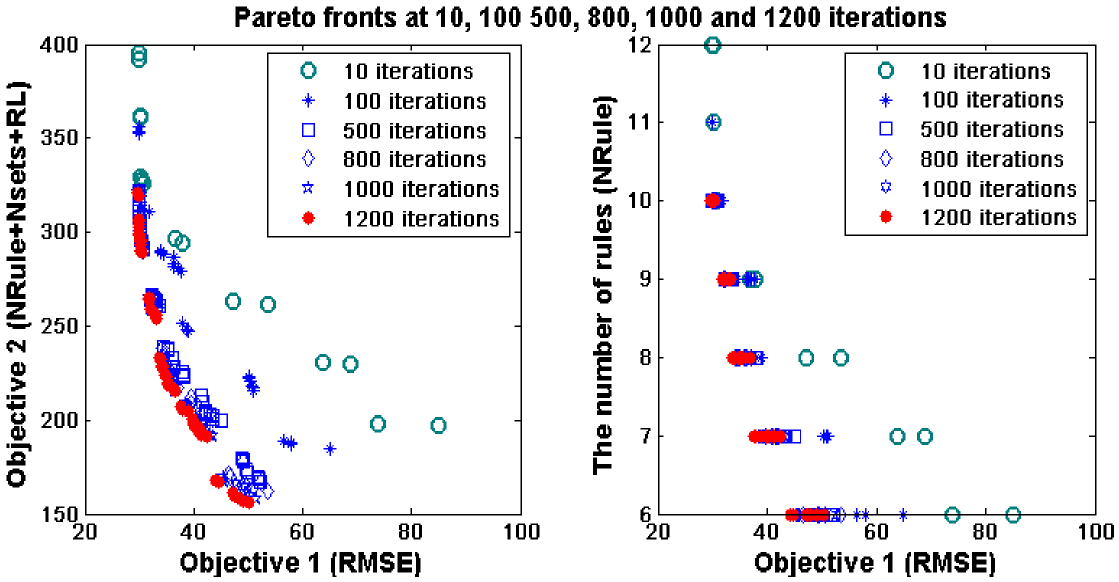


Figure 6.53 The snapshot of the Pareto FRBSs at 10, 100, 500, 800, 1000 and 1200 iterations.

Table 6.7 summarises the results of the UTS modelling problem using IMOFM_S with and without the variable length coding and the new distance index. The results were obtained from two random runs with each of them for one of IMOFM_S configurations. The improvements with the variable length coding and the new distance index are much bigger, especially for the FRBSs with fewer rules, comparing to that of the low dimensional problem presented in Section 5.6.3. Since the FRBS with fewer rules is more prone to suffering from the so-called ‘unordered set of rules’, IMOFM_S with the variable length coding and the new distance index is more effective in such a scenario as compared to the one without such a scheme. Figure 6.54 shows the Pareto fronts produced by two different IMOFM_S implementations, viz. with and without the variable length coding and the new distance index.

TABLE 6.7
THE COMPARISON OF THE MODELING APPROACHES WITH AND WITHOUT VARIABLE LENGTH CODING SCHEME

FRBS Configurations	No. of rules	Objective 1	IMOFM_S (without VLC) Training Performance (RMSE)	IMOFM_S (with VLC) Training Performance (RMSE)	Improvement (%)
Pareto FRBS1	11	348	29.782	29.671	0.3%
Pareto FRBS2	10	306	29.944	29.824	0.4%
Pareto FRBS3	10	304	29.952	29.839	0.4%
Pareto FRBS4	10	298	30.024	29.882	0.5%
Pareto FRBS6	9	263	31.972	31.865	0.3%
Pareto FRBS7	8	226	35.871	33.484	6.7%
Pareto FRBS8	8	225	36.273	33.733	7.0%
Pareto FRBS9	8	212	36.762	35.740	7.0%
Pareto FRBS10	7	193	40.854	38.194	6.5%
Pareto FRBS11	7	192	41.019	38.975	5.1%
Pareto FRBS12	7	188	42.333	40.536	4.2%
Pareto FRBS13	6	162	45.725	41.100	10.0%
Pareto FRBS14	6	161	45.869	41.400	9.8%
Pareto FRBS15	6	158	47.052	41.994	10.7%
Pareto FRBS16	6	157	47.780	42.581	10.9%

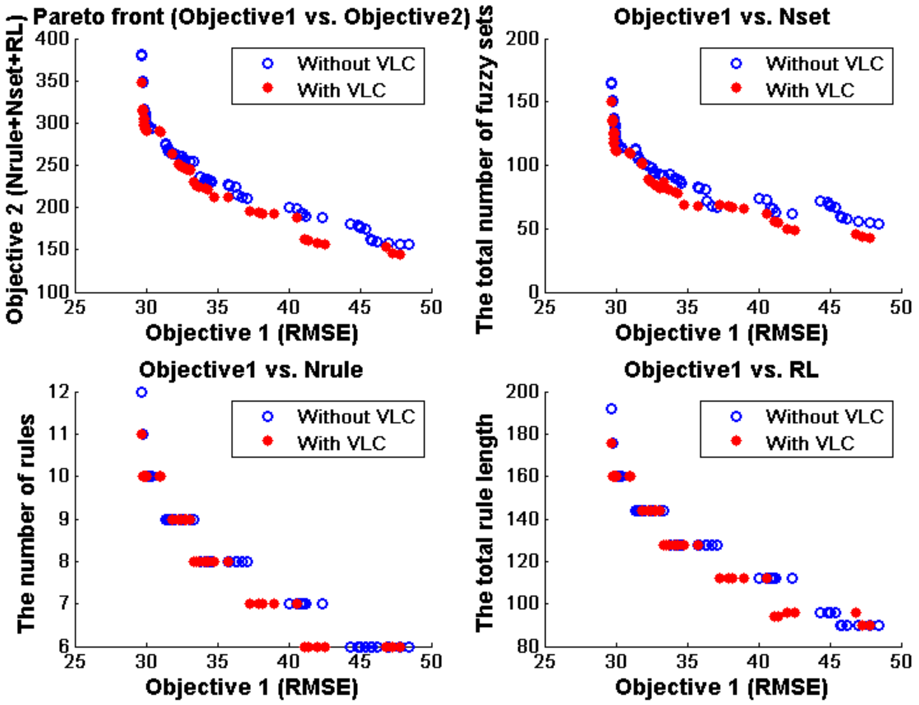


Figure 6.54 The Pareto fronts obtained using IMOFM_S with and without the variable length coding and the new distance index for the UTS problem.

6.3.5 Predictions of Reduction of Area (ROA)

Reduction of area (ROA) is part of the tensile testing procedure described in Section 6.3.3. The reduction of area is measured as the percentage change in the diameter of the specimen after fracture. This data set includes 3710 data samples. It has 15 inputs, which are the same as those shown in Table 6.3 with the same ranges, and 1 output, which is the ROA with the maximum value being 79.4% and the minimum value being 21.8%. In order to compare with Zhang & Mahfouf's work (2009), the ROA data set is randomly divided into two parts: 75% of the data are used for training and the remaining data are used for testing. The maximum number of rules is set to 12 for both IMOFM_S and IMOFM_M. The number of iterations for the second and the third modelling stages are set to 500 and 1200 respectively. Other parameters are kept the same as those given in Section 5.6.2.

Figures 6.55 and 6.56 show the training and testing processes of the second modelling stage of IMOFM_S and IMOFM_M, each of which is extracted from one of 10 runs.

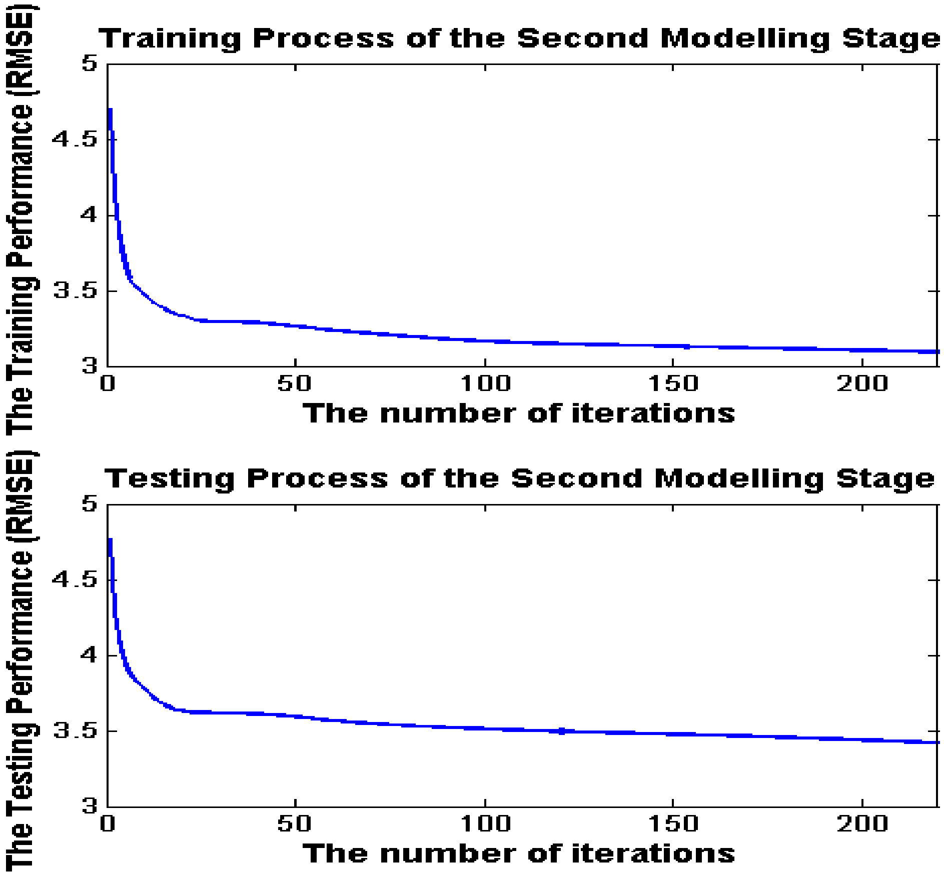


Figure 6.55 The training and testing process of the IMOFM_S on the ROA data.

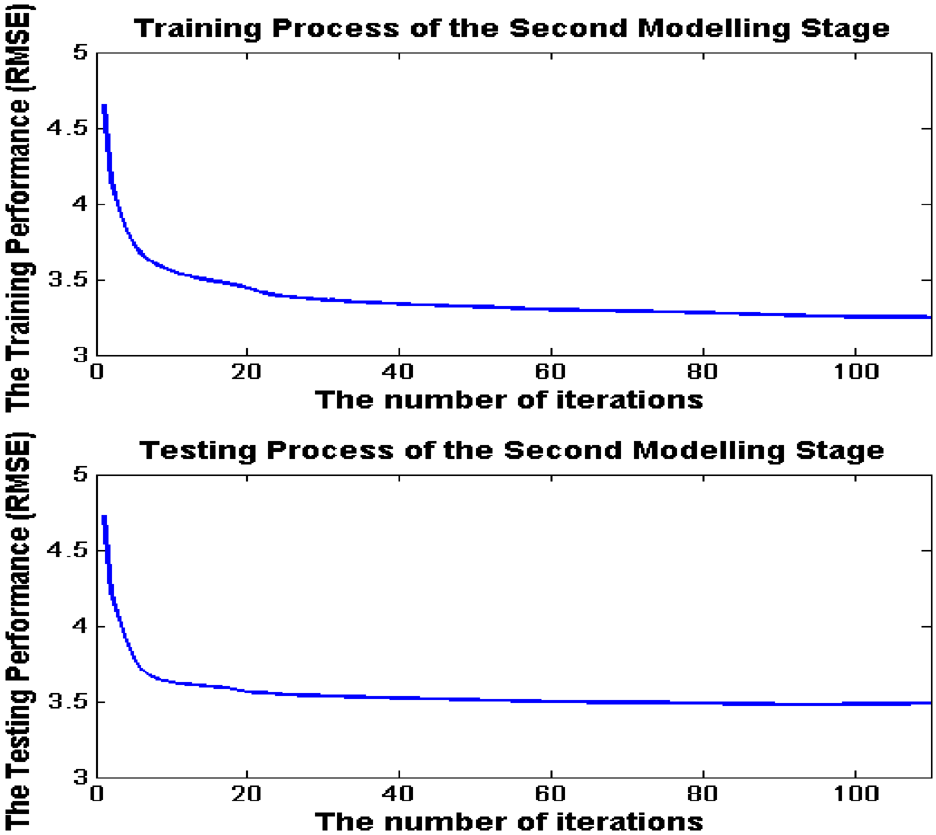


Figure 6.56 The training and testing process of the IMOFM_M on the ROA data.

The results presented in Table 6.8 include the average values of 10 independent runs.

TABLE 6.8

COMPARISONS OF THE PREDICTIVE PERFORMANCES FOR THE DIFFERENT MODELING METHODS USING THE ROA DATA

Modeling Methods (Ref.)	No. of rules	No. of fuzzy sets in inputs and output	Performance			
			Training (RMSE)	Testing (RMSE)	Training (R-Square)	Testing (R-Square)
Q. Zhang & M. Mahfouf (2007)						
Initial FRBS	20	Input:[20 20 20 20 20 20 20 20 20 20 20 20 20 20 20]; output: 20	5.92^{*}	5.44^{*}	-	-
Pareto FRBS1	15	Input: [14 12 13 14 13 14 11 14 13 15 6 10 13 12 13]; output: 13	$3.46^{\#}$	$3.75^{\#}$	-	-
Pareto FRBS2	7	Input: [5 4 3 4 5 4 4 5 6 4 4 5 5 3 4]; output: 6	$4.41^{\#}$	$4.40^{\#}$	-	-
IMOFM_S (NB: Average results over 10 runs are presented here)						
Initial FRBS	12	Input: [12 12 12 12 12 12 12 12 12 12 12 12 12 12 12]; output: 12	$(5.82^{*}/$ $3.08^{@})$	$(5.82^{*}/$ $3.43^{@})$	$(0.4766^{*}/$ $0.8859^{@})$	$(0.4447^{*}/$ $0.8506^{@})$
Pareto FRBS1	12	Input: [7 10 10 10 6 8 6 8 8 11 8 6 9 8 9]; output: 12	$2.98^{\#}$	$3.29^{\#}$	$0.8887^{\#}$	$0.8570^{\#}$
Pareto FRBS2	10	Input: [6 7 6 10 5 8 7 7 8 9 6 5 7 7 10]; output: 10	$3.00^{\#}$	$3.30^{\#}$	$0.8869^{\#}$	$0.8572^{\#}$
Pareto FRBS3	10	Input: [4 7 5 9 5 6 5 6 6 7 5 4 7 6 8]; output: 10	$3.06^{\#}$	$3.29^{\#}$	$0.8822^{\#}$	$0.8572^{\#}$
Pareto FRBS4	10	Input: [4 6 5 9 5 5 5 5 6 6 5 3 7 6 7]; output: 10	$3.09^{\#}$	$3.29^{\#}$	$0.8822^{\#}$	$0.8572^{\#}$
Pareto FRBS5	8	Input: [6 3 6 6 2 6 3 5 5 5 4 5 7 7 6]; output: 8	$3.16^{\#}$	$3.42^{\#}$	$0.8736^{\#}$	$0.8448^{\#}$
Pareto FRBS6	8	Input: [5 3 5 5 2 5 3 5 5 4 4 4 5 6 6]; output: 8	$3.18^{\#}$	$3.42^{\#}$	$0.8736^{\#}$	$0.8448^{\#}$
Pareto FRBS7	7	Input: [5 3 4 4 1 4 3 5 5 3 4 3 4 6 5]; output: 7	$3.28^{\#}$	$3.47^{\#}$	$0.8629^{\#}$	$0.8401^{\#}$
Pareto FRBS9	6	Input: [4 2 4 6 3 4 3 3 4 4 3 4 5 3 4]; output: 6	$3.38^{\#}$	$3.59^{\#}$	$0.8541^{\#}$	$0.8272^{\#}$
Pareto FRBS10	6	Input: [4 2 3 3 1 2 3 5 4 4 3 2 4 4 5]; output: 6	$3.40^{\#}$	$3.54^{\#}$	$0.8518^{\#}$	$0.8329^{\#}$
IMOFM_M (NB: Average results over 10 runs are presented here)						
Initial FRBS	12	Input: [12 12 12 12 12 12 12 12 12 12 12 12 12 12 12]; output: 12	$(5.72^{*}/$ $3.24^{@})$	$(5.97^{*}/$ $3.49^{@})$	$(0.4716^{*}/$ $0.8686^{@})$	$(0.5199^{*}/$ $0.8643^{@})$
Pareto FRBS1	9	Input: [6 7 7 7 3 6 5 4 5 6 3 3 5 6 7]; output: 7	$3.03^{\#}$	$3.34^{\#}$	$0.8799^{\#}$	$0.8697^{\#}$
Pareto FRBS2	8	Input: [5 7 7 5 3 5 3 7 4 7 3 1 5 5 7]; output: 8	$3.07^{\#}$	$3.32^{\#}$	$0.8766^{\#}$	$0.8707^{\#}$
Pareto FRBS3	8	Input: [5 7 7 5 2 5 3 7 4 7 3 1 4 5 7]; output: 7	$3.10^{\#}$	$3.32^{\#}$	$0.8746^{\#}$	$0.8703^{\#}$
Pareto FRBS4	7	Input: [4 6 5 5 2 4 5 4 4 7 3 1 4 5 7]; output: 5	$3.14^{\#}$	$3.30^{\#}$	$0.8708^{\#}$	$0.8716^{\#}$
Pareto FRBS5	7	Input: [2 5 5 5 2 4 5 6 3 6 3 1 2 5 7]; output: 5	$3.22^{\#}$	$3.38^{\#}$	$0.8636^{\#}$	$0.8652^{\#}$
Pareto FRBS6	6	Input: [4 5 5 4 1 4 4 3 2 4 3 1 2 5 6]; output: 5	$3.30^{\#}$	$3.40^{\#}$	$0.8561^{\#}$	$0.8632^{\#}$
Pareto FRBS7	6	Input: [2 3 5 4 1 3 4 4 3 5 3 0 2 5 6]; output: 5	$3.38^{\#}$	$3.48^{\#}$	$0.8483^{\#}$	$0.8553^{\#}$

[*] Initial model extracted directly from data using clustering algorithms or grid partition methods.

[@] Refined model or the consequents are computed through the estimation methods. [#] Simplified model after model simplification and parameter fine tuning.

Figures 6.57~6.58 show the ‘actual outputs vs. predicted outputs’ graphs from the three modelling stages.

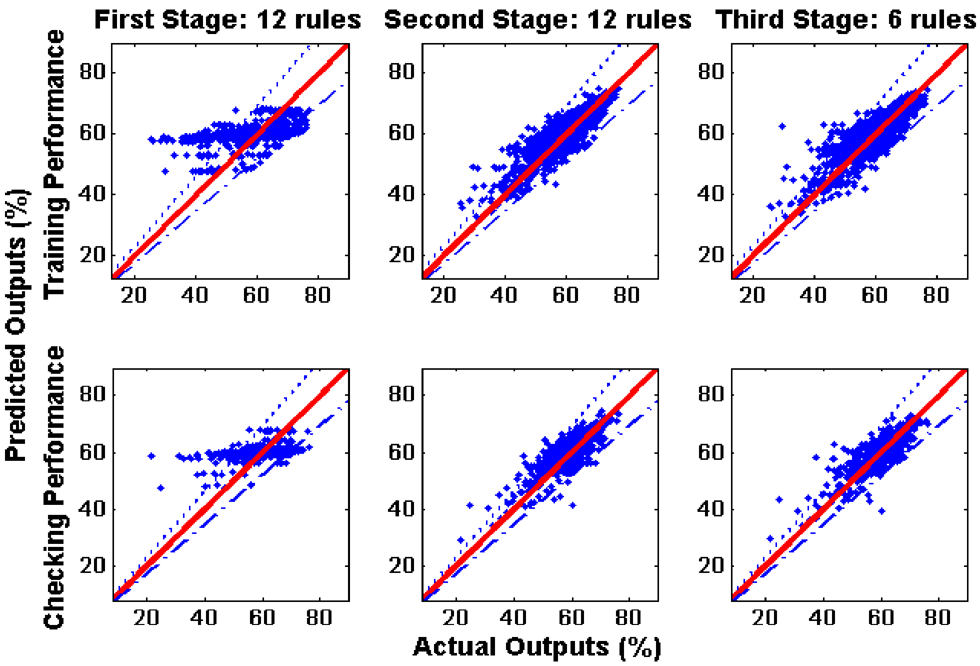


Figure 6.57 The prediction performances of the three stages for the ROA training and testing data using IMOFM_S.

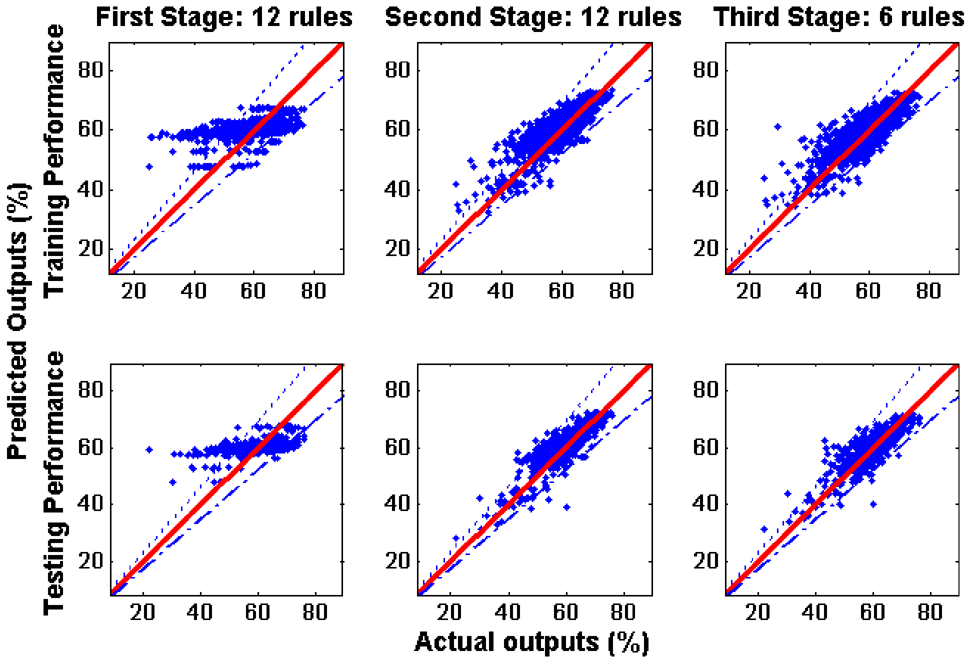


Figure 6.58 The prediction performances of the three stages for the ROA training and testing data using IMOFM_M.

Figures 6.59~6.60 show the Pareto fronts of the ROA modelling problem using IMOFM_S (28 Pareto FRBSs) and IMOFM_M (14 Pareto FRBSs) from one of 10 runs.

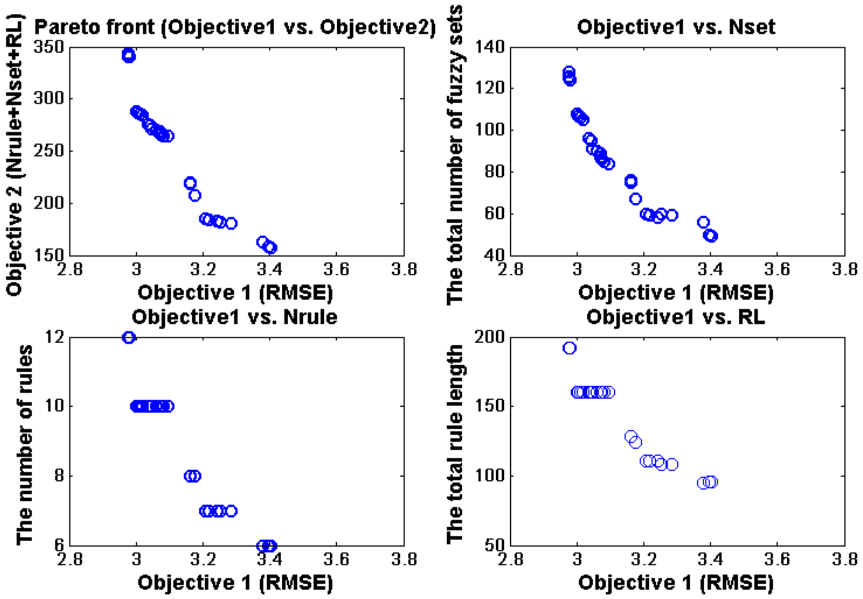


Figure 6.59 The Pareto fronts obtained using IMOFM_S from the third modelling procedure for the ROA modelling problem: (a) Objective1 vs. Objective2; (b) Objective1 vs. Nset; (c) Objective1 vs. Nrule; (d) Objective1 vs. RL.

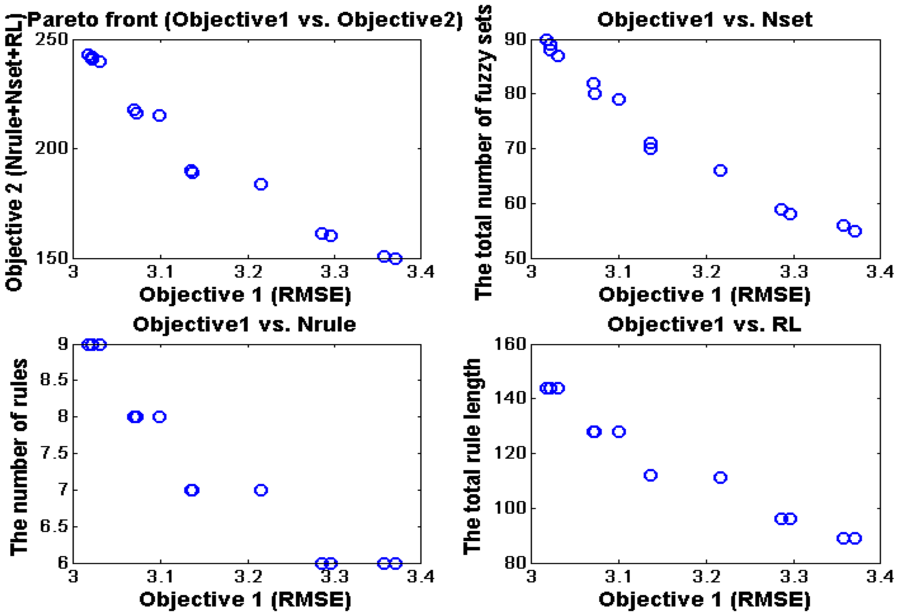


Figure 6.60 The Pareto fronts obtained using IMOFM_M from the third modelling procedure for the ROA modelling problem: (a) Objective1 vs. Objective2; (b) Objective1 vs. Nset; (c) Objective1 vs. Nrule; (d) Objective1 vs. RL.

Figures 6.61~6.66 show the distribution of membership functions of the fuzzy models found in different modelling stages.

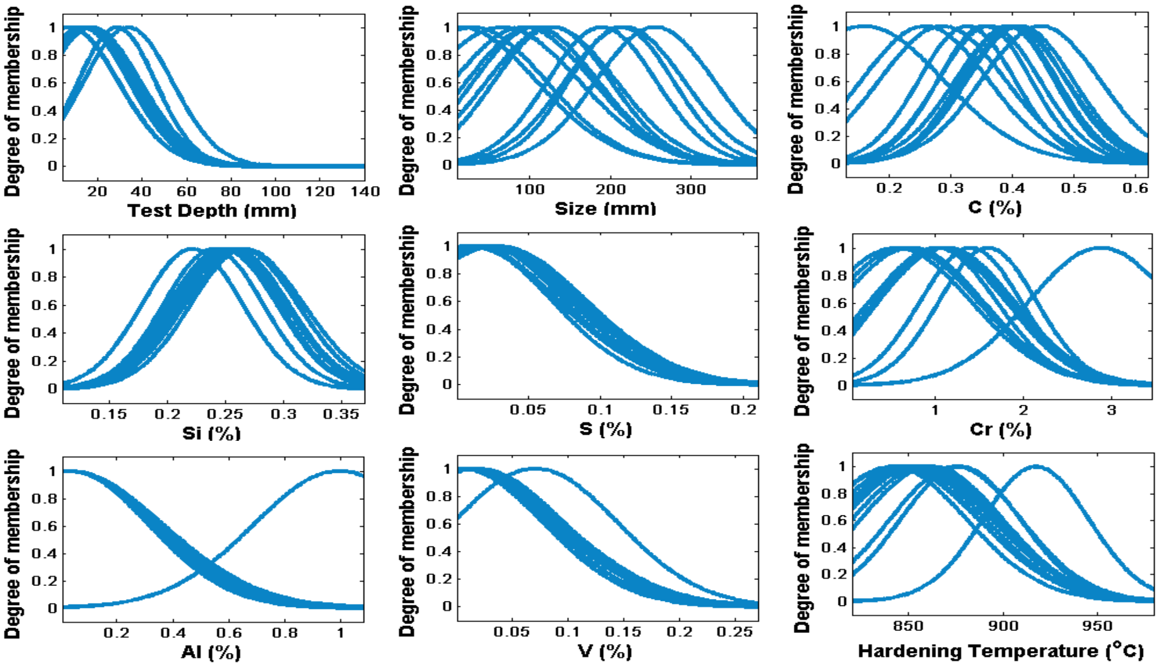


Figure 6.61 The distribution of membership functions of the 12-rule initial Singleton FRBS (from the first modelling stage) for ROA modelling.

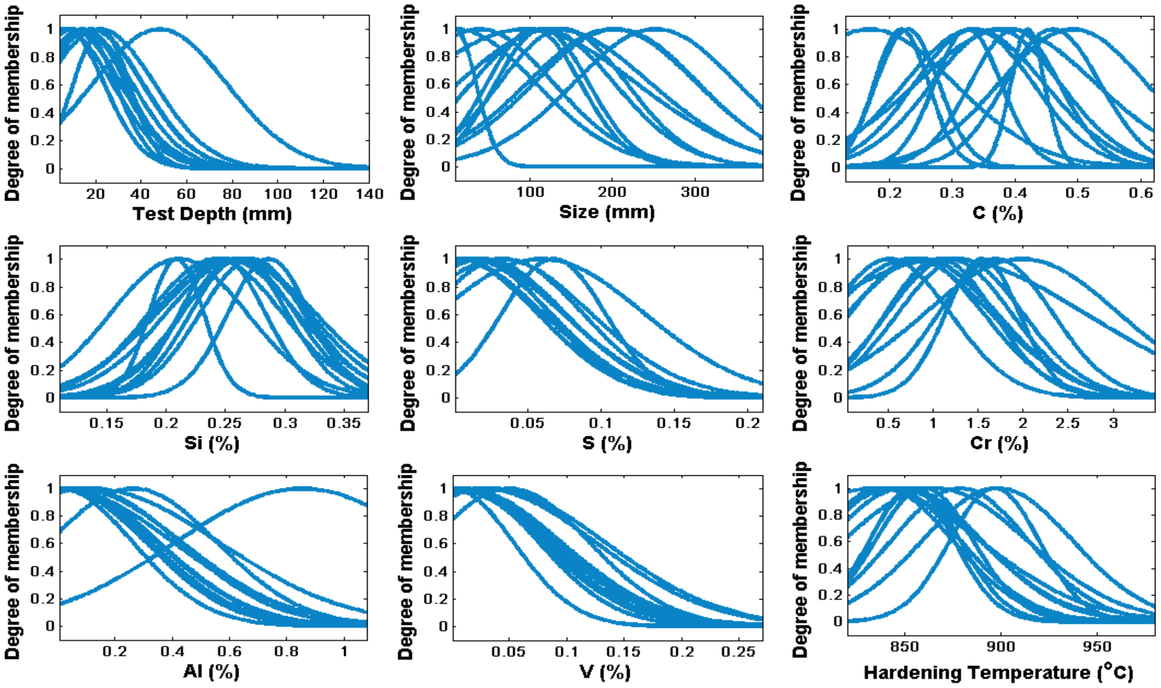


Figure 6.62 The distribution of membership functions of the 12-rule refined Singleton FRBS (from the second modelling stage) for ROA modelling.

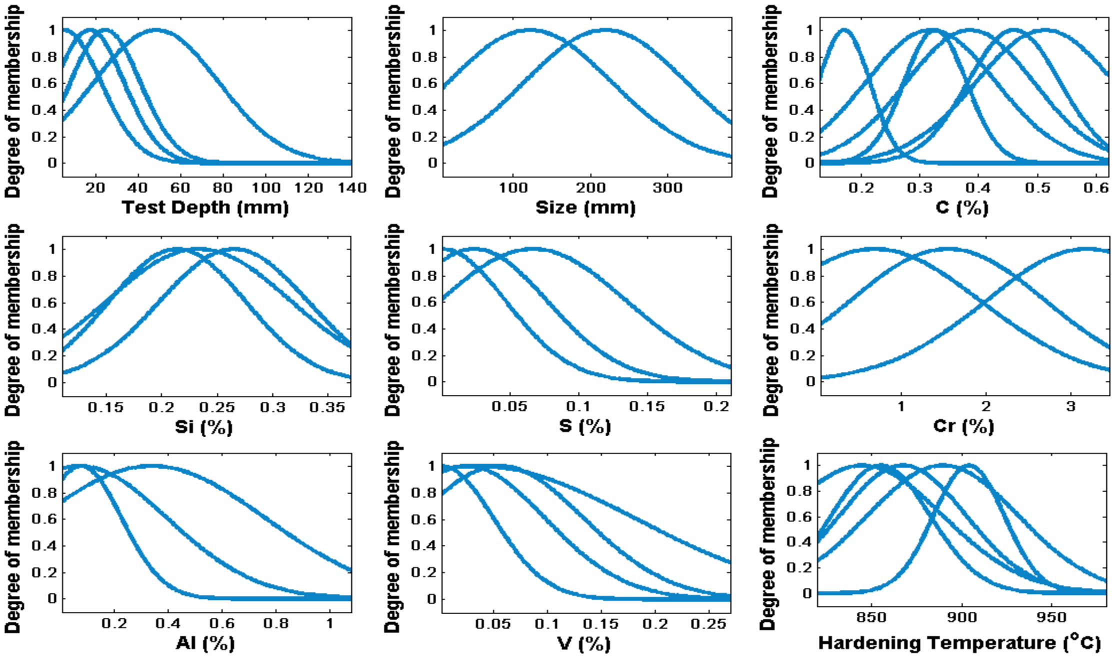


Figure 6.63 The distribution of membership functions of the 6-rule simplified Singleton FRBS (from the third modelling stage) for ROA modelling.

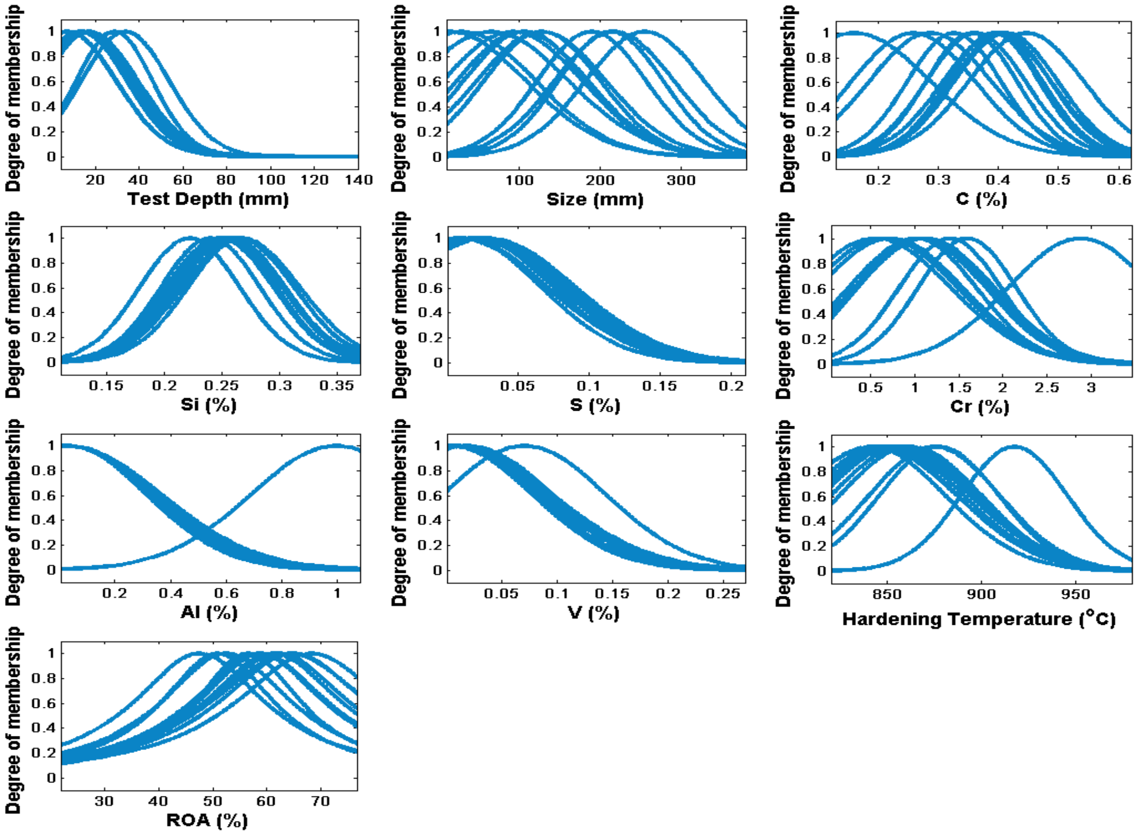


Figure 6.64 The distribution of membership functions of the 12-rule initial Mamdani FRBS (from the first modelling stage) for ROA modelling.

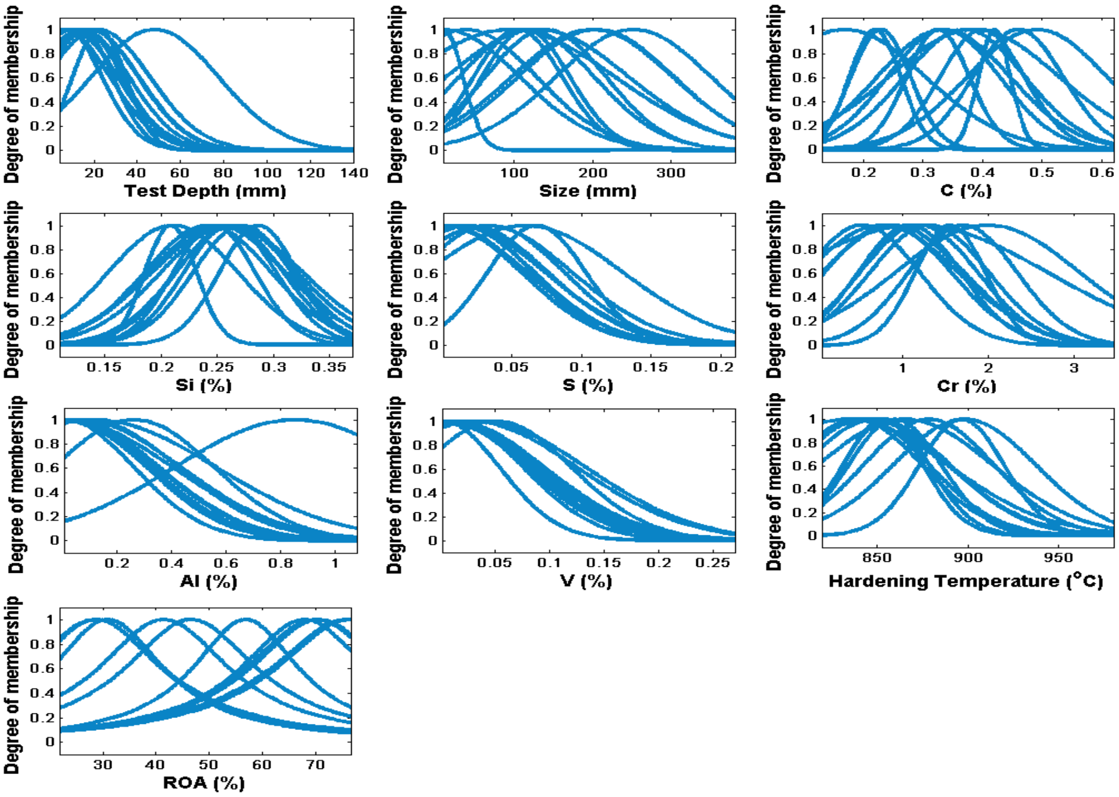


Figure 6.65 The distribution of membership functions of the 12-rule refined Mamdani FRBS (from the second modelling stage) for ROA modelling

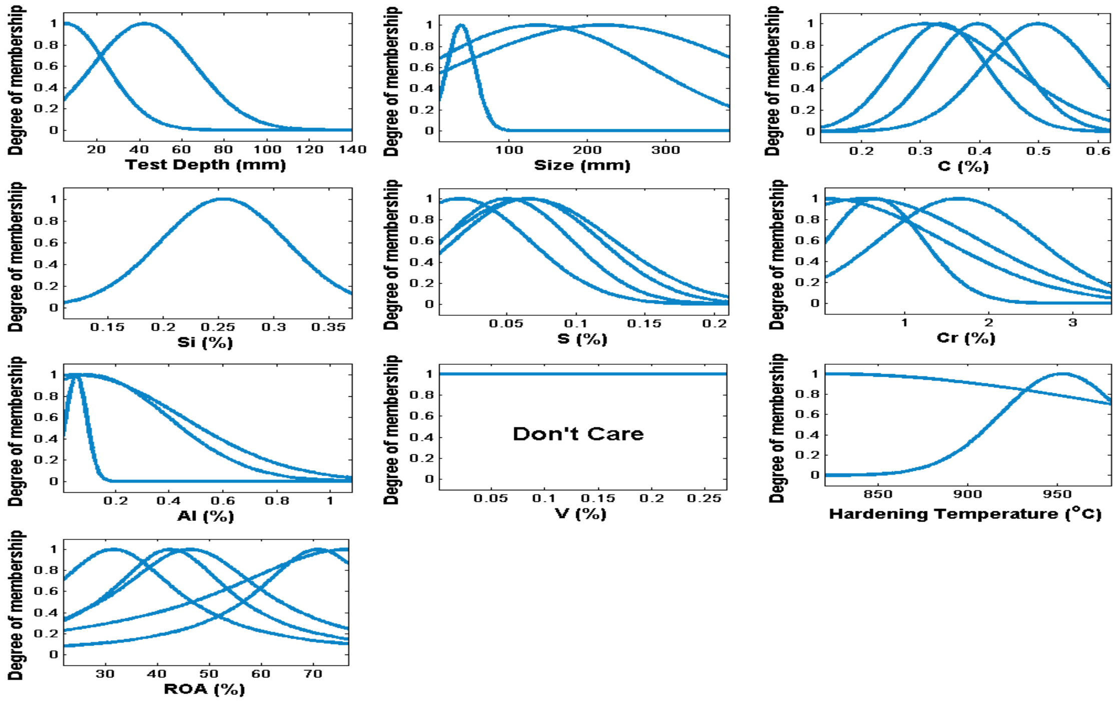


Figure 6.66 The distribution of membership functions of the 6-rule simplified Mamdani FRBS (from the third modelling stage) for ROA modelling.

As can be seen from Figure 6.66, IMOFM_M can not only be used to simplify the overlapped membership functions, but also can be used to select useful inputs. In the above example, V has been identified as a redundant factor among all chemical compositions for the decision of ROA.

Figures 6.67~6.68 show the three-dimensional response surfaces of the simplified Singleton and Mamdani ROA models. The 'median' values of the constant variables are referred to Table 6.5.

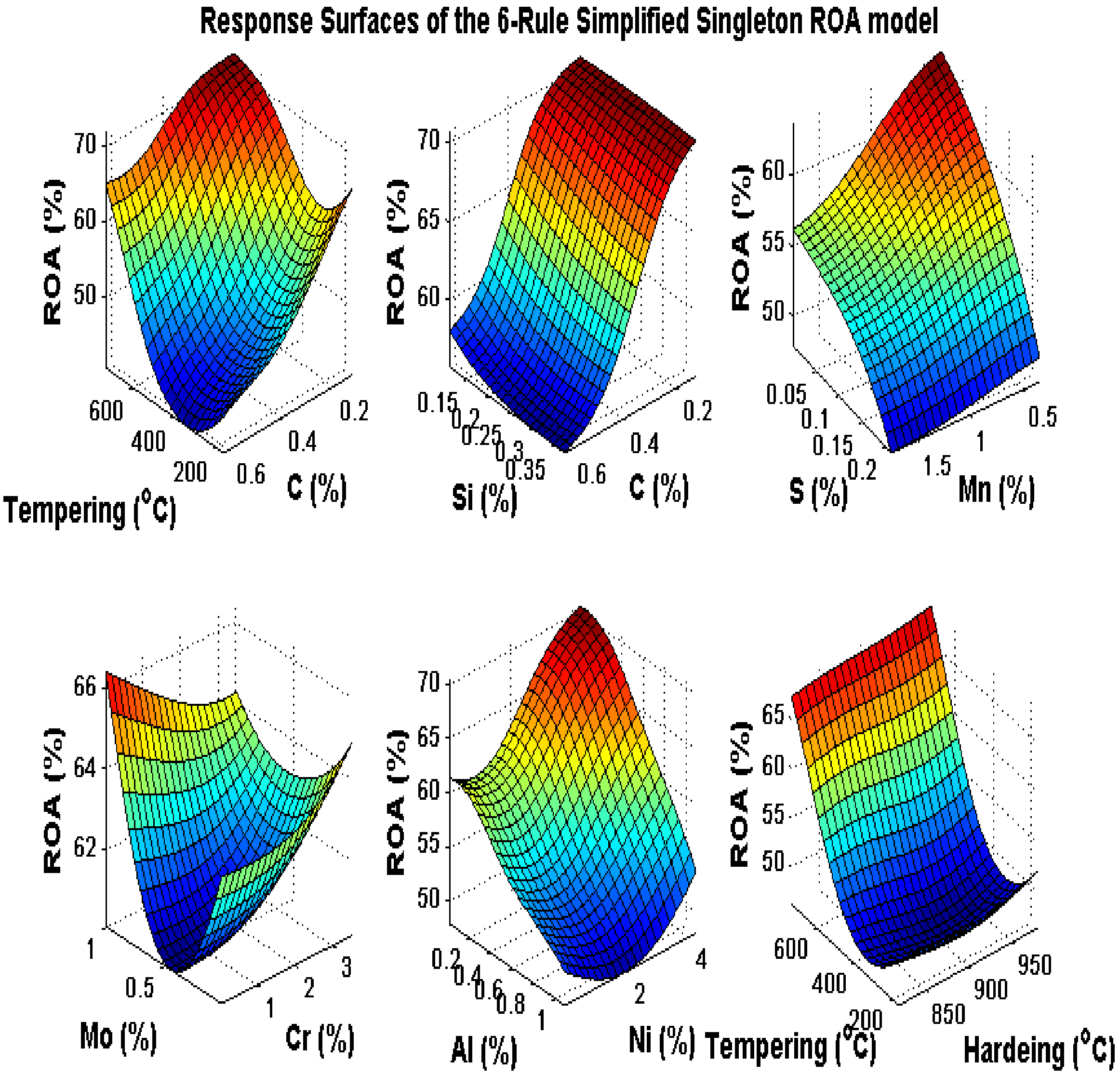


Figure 6.67 Response surfaces of the 6-rule simplified Singleton ROA model.

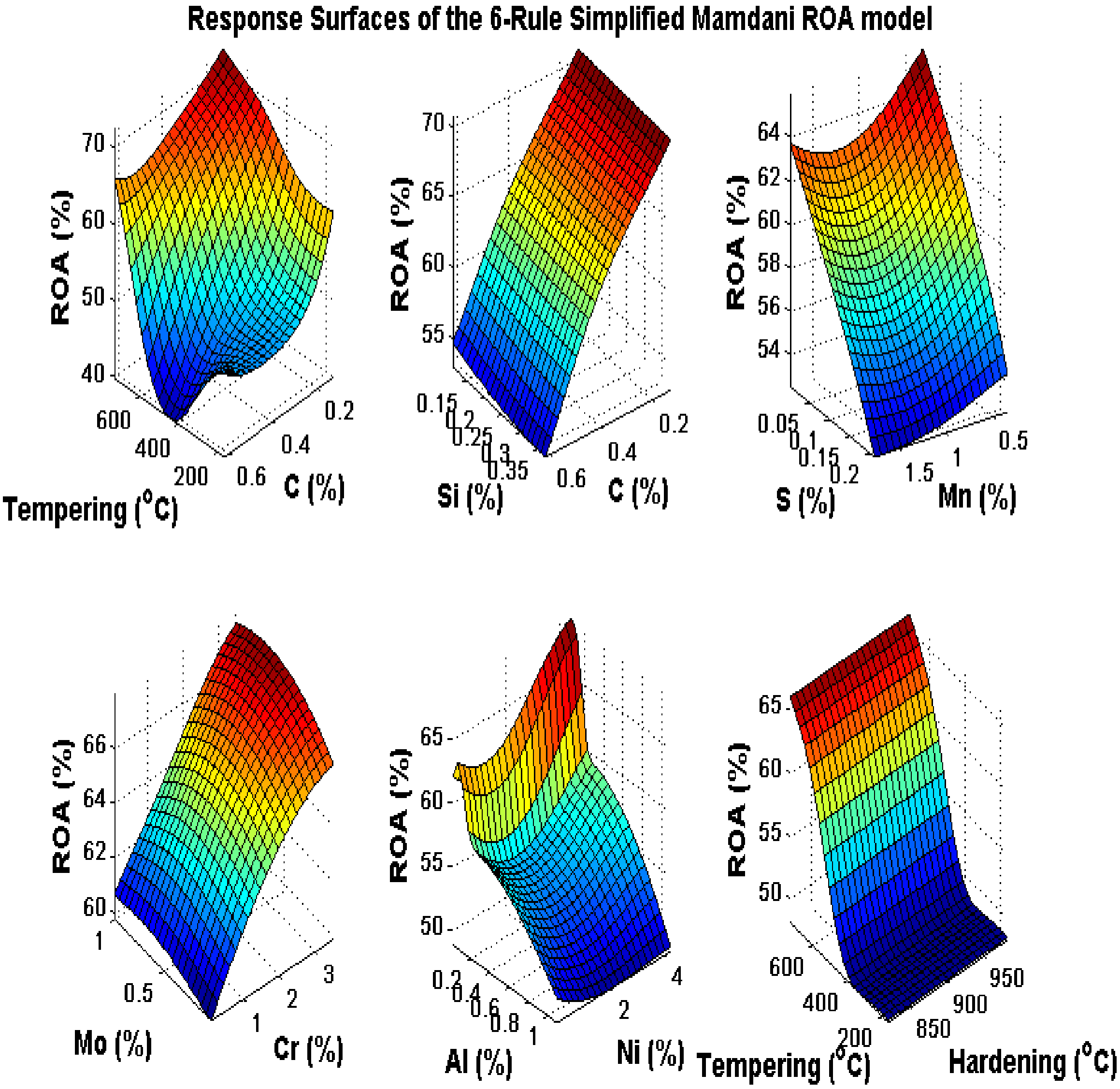


Figure 6.68 Response surfaces of the 6-rule simplified Mamdani ROA model.

The response surfaces of Mo-Cr of the Singleton and Mamdani ROA models are different in the regions where Mo is around 1 and Cr is around 0. This can be explained via Figure 6.69 (c). Indeed, the data samples are sparse in such a region, hence the result. Figures 6.69 (a) and (b) also explain the reason why the low value ROA points are modelled poorly, as one can see from Figures 6.57~6.58 and 6.70. The distribution of the ROA is obviously skewed towards to higher values, and this was thought to have affected the accuracy of the low-end values.

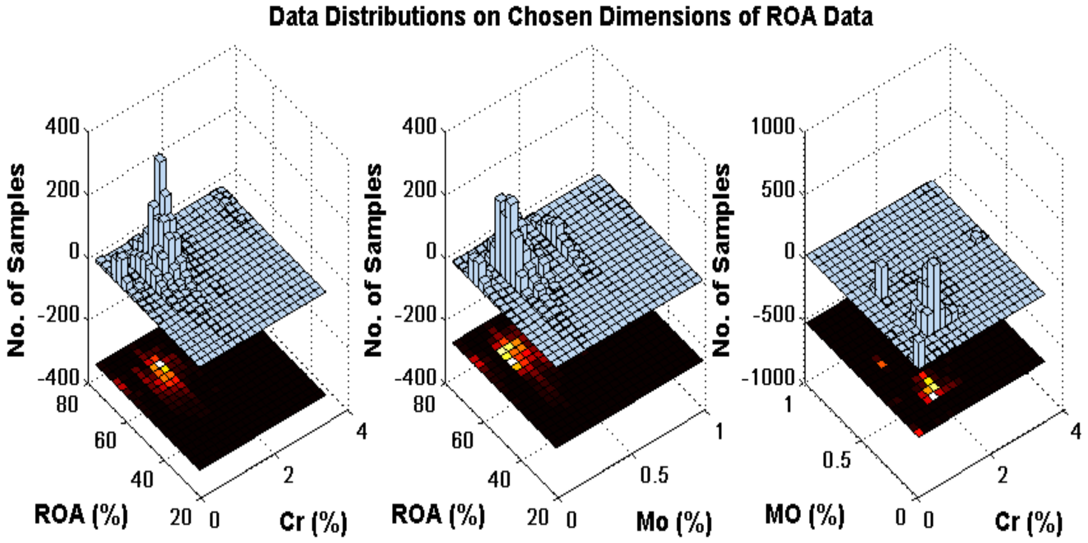


Figure 6.69 Data distributions on chosen dimensions of ROA data.

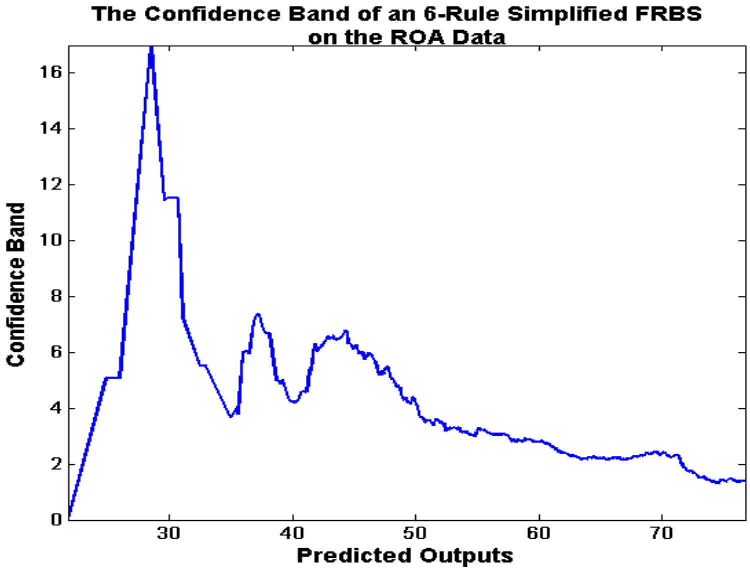


Figure 6.70 5%-range confidence band of the 6-rule simplified Singleton FRBS on the ROA data.

6.3.6 Predictions of Elongation

Elongation is the final property derived from the tensile strength test, as described in Section 6.3.3. Since elongation is measured as the percentage change in gauge length after fracture, this property is dependent on the gauge length used for the specimen, which may be defined as either 4 or 5 times the diameter of the specimen. Hence, apart from the inputs shown in

Table 6.3 for the UTS and ROA data, gauge length is also included, which is defined as 4 or 5 times the diameter of the specimen. The original data set consists of 3804 samples. However, as indicated by Figure 6.71, there is only one data sample whose output value is greater than 35. Hence, we removed this data points in the subsequent experiment. Table 6.9 describes the maximum and minimum values for each input and output. 75% of the data are used for training and the remaining data is used for testing.

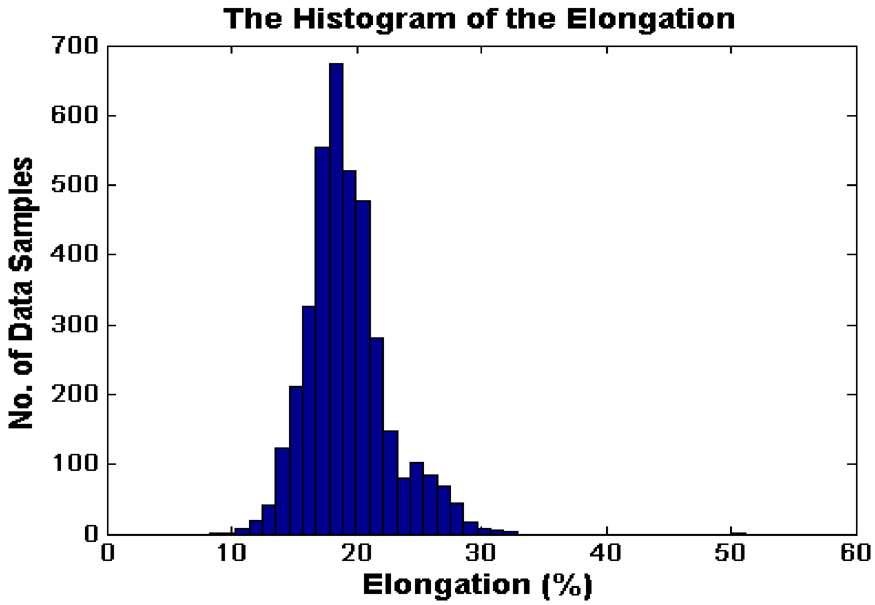


Figure 6.71 The histogram of the output of the elongation data.

TABLE 6.9

The Inputs And Output Of Elongation Data Set

Inputs	**Gauge Length**	**Test Depth**	**Size**	**Site**	**%C**	**%Si**	**%Mn**	**%S**
Max.	5	140	381	6	0.62	0.37	1.75	0.21
Min.	4	4	10	1	0.13	0.11	0.35	0.0005
Inputs	**%Cr**	**%Mo**	**%Ni**	**%Al**	**%V**	**Hardening Temperature**	**Cooling Medium**	**Tempering Temperature**
Max.	3.46	1	4.21	1.08	0.27	980	3	730
Min.	0.05	0.01	0.02	0.005	0.001	820	1	170
Output	**Elongation (%)** (Max.: 51.1; Min.: 8.2)							

The maximum number of rules is set to 12 for both IMOFM_S and IMOFM_M. The numbers of iterations for the second and third modelling stages are set to 150 and 1200 respectively. Other parameters are kept the same as those given in section 5.6.2. Figures 6.72 and 6.73

show the training testing processes of the second modelling stage. Table 6.10 summarises the predictive performances.

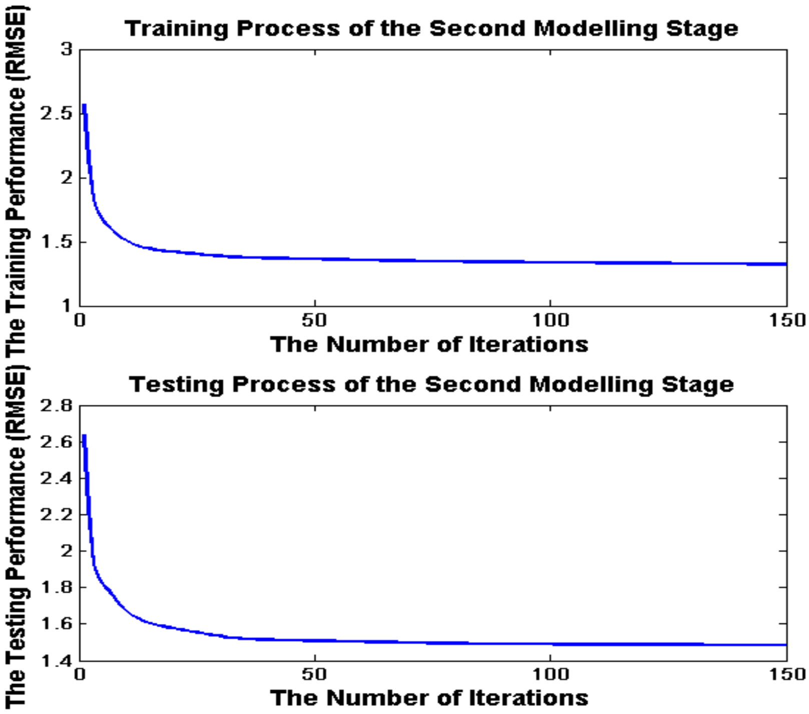


Figure 6.72 the training and testing process of the IMOFM_S on the elongation data.

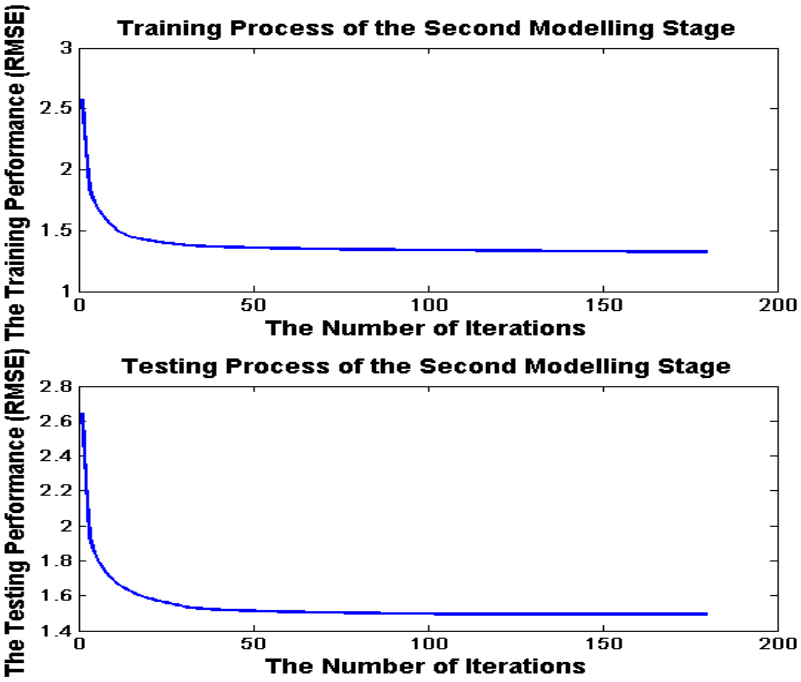


Figure 6.73 The training and testing process of the IMOFM_M on the elongation data.

TABLE 6.10

COMPARISONS OF THE PREDICTIVE PERFORMANCES FOR THE DIFFERENT MODELING METHODS USING THE ELONGATION DATA

Modeling Methods (Ref.)	No. of rules	No. of fuzzy sets in inputs and output	Performance			
			Training (RMSE)	Testing (RMSE)	Training (R-Square)	Testing (R-Square)
Q. Zhang & M. Mahfouf (2007)						
Initial FRBS	15	Input:[15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15]; output: 15	2.39*	2.23*	-	-
Pareto FRBS1	10	Input: [8 6 9 7 8 9 9 3 9 9 7 6 5 9 9 9]; output: 9	1.78$^{\#}$	1.76$^{\#}$	-	-
Pareto FRBS2	8	Input: [5 4 5 2 5 5 6 3 4 4 5 2 4 5 5 5]; output: 7	1.78$^{\#}$	1.65$^{\#}$	-	-
IMOFM_S (NB: Average results over 10 runs are presented here)						
Initial FRBS	12	Input: [12 12 12 12 12 12 12 12 12 12 12 12 12 12 12 12]; output: 12	(2.72*/ 1.32$^{@}$)	(2.83*/ 1.48$^{@}$)	(0.5461*/ 0.9117$^{@}$)	(0.5678*/ 0.8966$^{@}$)
Pareto FRBS1	12	Input: [10 8 12 11 10 10 11 6 11 12 11 6 9 11 11 12]; output: 12	1.31$^{\#}$	1.48$^{\#}$	0.9128$^{\#}$	0.8961$^{\#}$
Pareto FRBS2	11	Input: [7 6 9 9 9 7 8 4 10 10 10 6 7 10 10 11]; output: 11	1.31$^{\#}$	1.48$^{\#}$	0.9126$^{\#}$	0.8964$^{\#}$
Pareto FRBS3	11	Input: [8 7 8 9 9 5 8 3 9 8 9 5 2 9 9 11]; output: 11	1.31$^{\#}$	1.48$^{\#}$	0.9122$^{\#}$	0.8960$^{\#}$
Pareto FRBS4	9	Input: [7 6 8 8 7 3 7 3 7 9 8 5 5 8 8 9]; output: 9	1.32$^{\#}$	1.49$^{\#}$	0.9120$^{\#}$	0.8955$^{\#}$
Pareto FRBS5	9	Input: [5 5 6 8 7 2 6 3 6 8 8 5 3 6 8 8]; output: 9	1.34$^{\#}$	1.51$^{\#}$	0.9094$^{\#}$	0.8920$^{\#}$
Pareto FRBS6	8	Input: [4 5 6 8 6 2 6 3 7 7 8 5 3 6 7 8]; output: 8	1.34$^{\#}$	1.52$^{\#}$	0.9086$^{\#}$	0.8910$^{\#}$
Pareto FRBS7	8	Input: [4 5 6 8 5 1 5 2 6 7 7 5 3 7 7 7]; output: 8	1.35$^{\#}$	1.53$^{\#}$	0.9076$^{\#}$	0.8887$^{\#}$
Pareto FRBS9	6	Input: [4 3 6 6 4 1 5 2 6 5 5 3 2 4 5 6]; output: 6	1.38$^{\#}$	1.55$^{\#}$	0.9026$^{\#}$	0.8863$^{\#}$
Pareto FRBS10	6	Input: [4 3 6 6 4 1 4 2 5 5 4 3 2 3 5 6]; output: 6	1.39$^{\#}$	1.55$^{\#}$	0.9016$^{\#}$	0.8867$^{\#}$
IMOFM_M (NB: Average results over 10 runs are presented here)						
Initial FRBS	12	Input: [12 12 12 12 12 12 12 12 12 12 12 12 12 12 12 12]; output: 12	(2.73*/ 1.32$^{@}$)	(2.82*/ 1.49$^{@}$)	(0.5448*/ 0.9117$^{@}$)	(0.5684*/ 0.8955$^{@}$)
Pareto FRBS1	12	Input: [8 7 7 10 11 7 8 5 10 9 9 5 6 8 8 10]; output: 11	1.30$^{\#}$	1.49$^{\#}$	0.9138$^{\#}$	0.8954$^{\#}$
Pareto FRBS2	11	Input: [8 5 5 8 9 7 8 5 10 9 10 5 4 8 8 10]; output: 11	1.30$^{\#}$	1.47$^{\#}$	0.9137$^{\#}$	0.8982$^{\#}$
Pareto FRBS3	10	Input: [6 5 6 9 9 4 6 4 8 8 8 4 4 6 7 8]; output: 9	1.32$^{\#}$	1.48$^{\#}$	0.9115$^{\#}$	0.8968$^{\#}$
Pareto FRBS4	9	Input: [6 5 7 8 9 4 6 4 8 7 7 3 3 6 8 8]; output: 9	1.33$^{\#}$	1.51$^{\#}$	0.9096$^{\#}$	0.8929$^{\#}$
Pareto FRBS5	8	Input: [6 4 5 8 7 2 4 3 6 5 6 3 3 5 6 7]; output: 6	1.35$^{\#}$	1.52$^{\#}$	0.9071$^{\#}$	0.8912$^{\#}$
Pareto FRBS6	6	Input: [5 4 4 6 5 2 5 3 4 6 6 3 3 3 3 4]; output: 6	1.39$^{\#}$	1.54$^{\#}$	0.9019$^{\#}$	0.8869$^{\#}$
Pareto FRBS7	6	Input: [4 3 3 6 4 1 3 3 5 6 5 2 3 2 4 4]; output: 4	1.41$^{\#}$	1.55$^{\#}$	0.8992$^{\#}$	0.8869$^{\#}$

* Initial model extracted directly from data using clustering algorithms or grid partition methods.

$^{@}$ Refined model or the consequents are computed through the estimation methods. $^{\#}$ Simplified model after model simplification and parameter fine tuning.

Figures 6.74~6.75 show the 'actual outputs vs. predicted outputs' graphs from the three modelling stages.

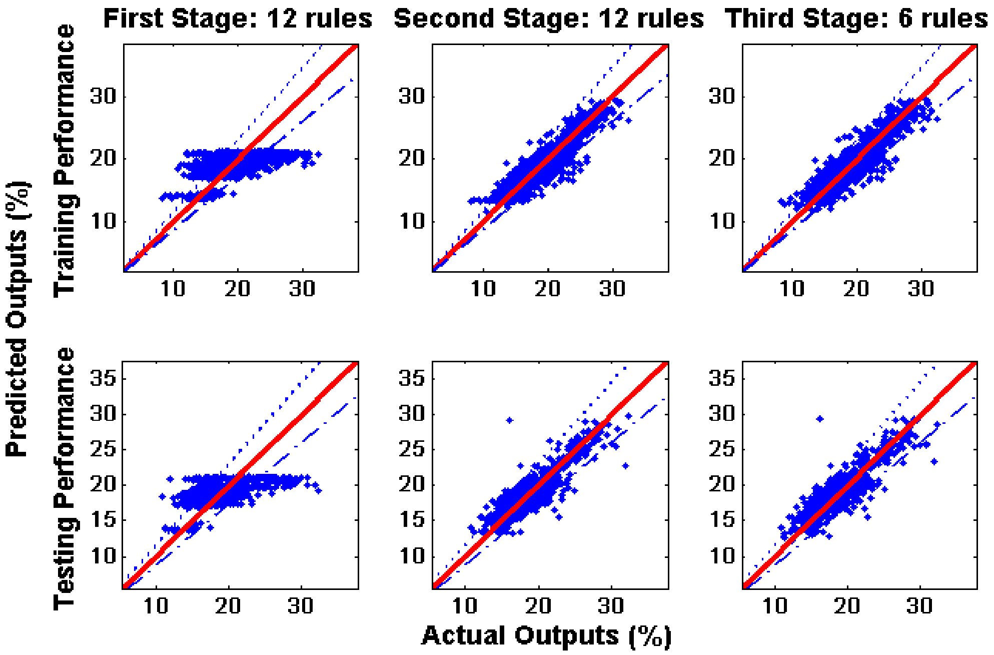


Figure 6.74 The prediction performances of the three stages for the elongation training and testing data using IMOFM_S.

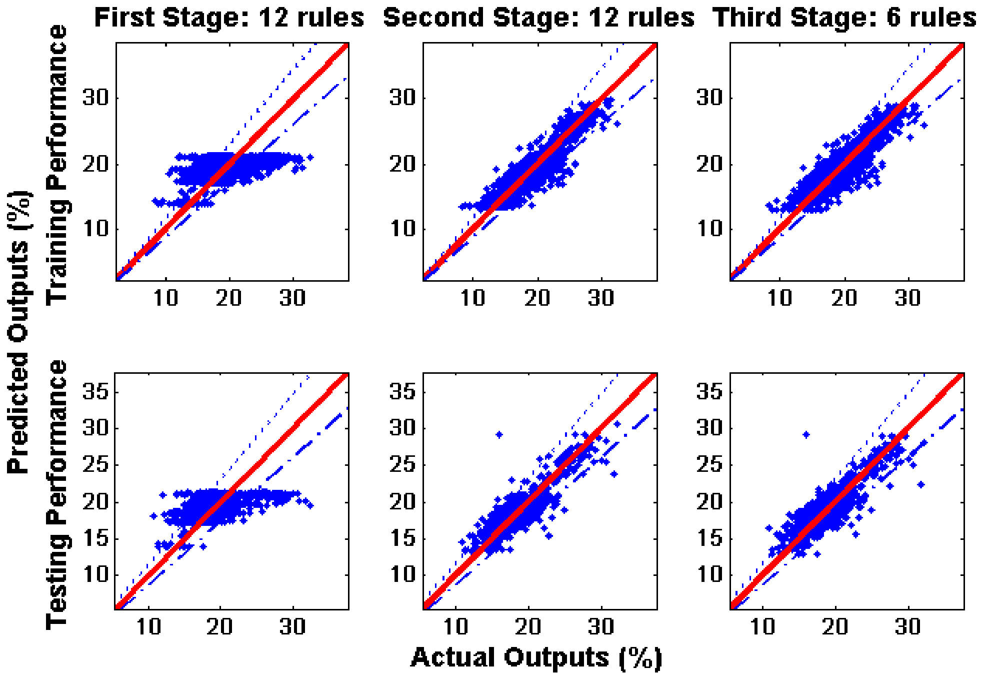


Figure 6.75 The prediction performances of the three stages for the elongation training and testing data using IMOFM_M.

Figures 6.76~6.77 show the Pareto fronts of the elongation modelling problem using IMOFM_S (28 Pareto FRBSs) and IMOFM_M (21 Pareto FRBSs) from one of 10 runs.

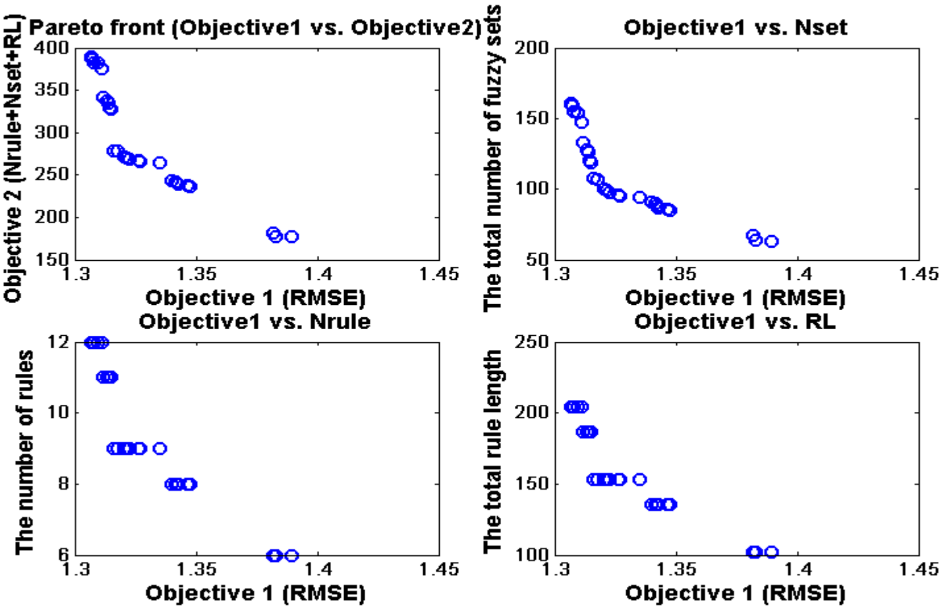


Figure 6.76 The Pareto fronts obtained using IMOFM_S from the third modelling procedure for the elongation modelling problem: (a) Objective1 vs. Objective2; (b) Objective1 vs. Nset; (c) Objective1 vs. Nrule; (d) Objective1 vs. RL.

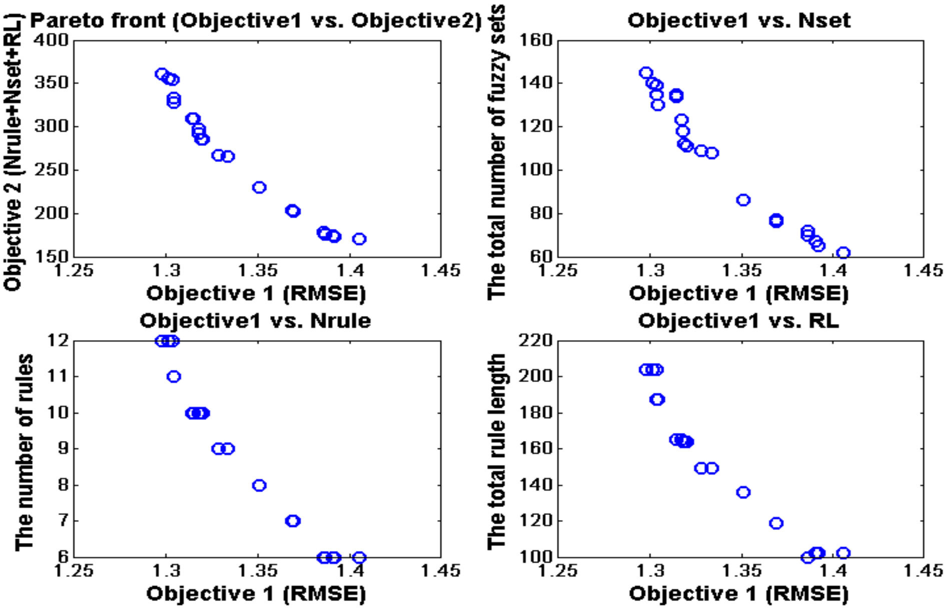


Figure 6.77 The Pareto fronts obtained using IMOFM_M from the third modelling procedure for the elongation modelling problem: (a) Objective1 vs. Objective2; (b) Objective1 vs. Nset; (c) Objective1 vs. Nrule; (d) Objective1 vs. RL.

Figures 6.78~6.83 show the distribution of some membership functions of the fuzzy models found in different modelling stages.

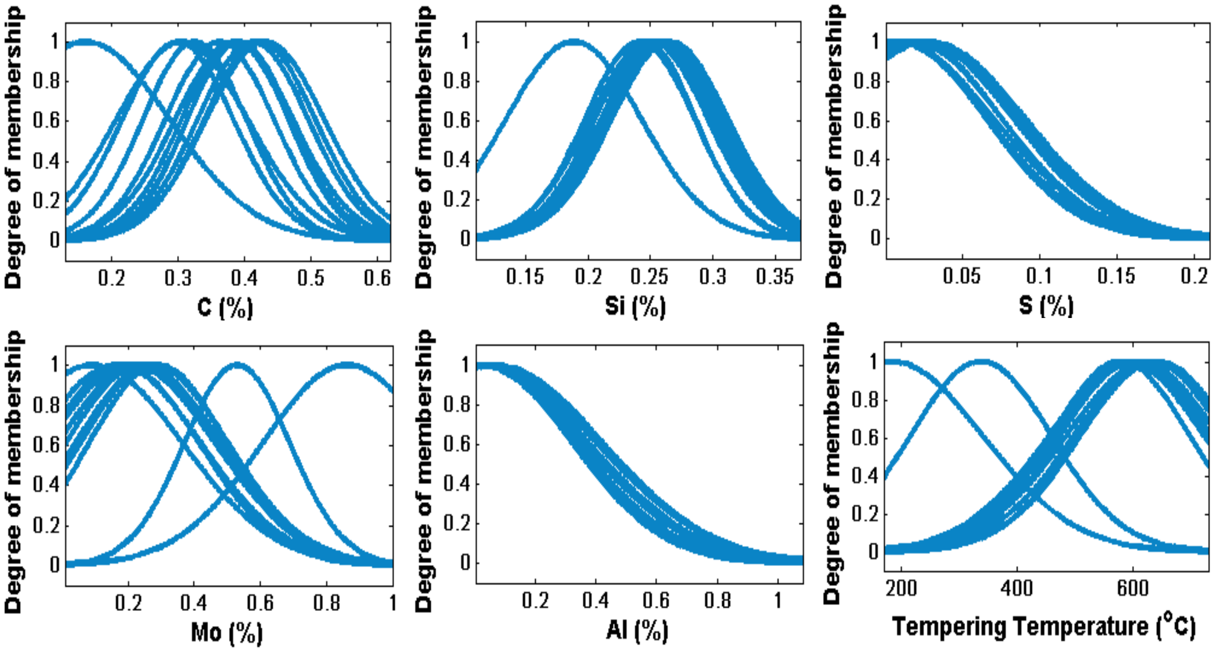


Figure 6.78 The distribution of some membership functions of the 12-rule initial Singleton FRBS (from the first modelling stage) for elongation modelling.

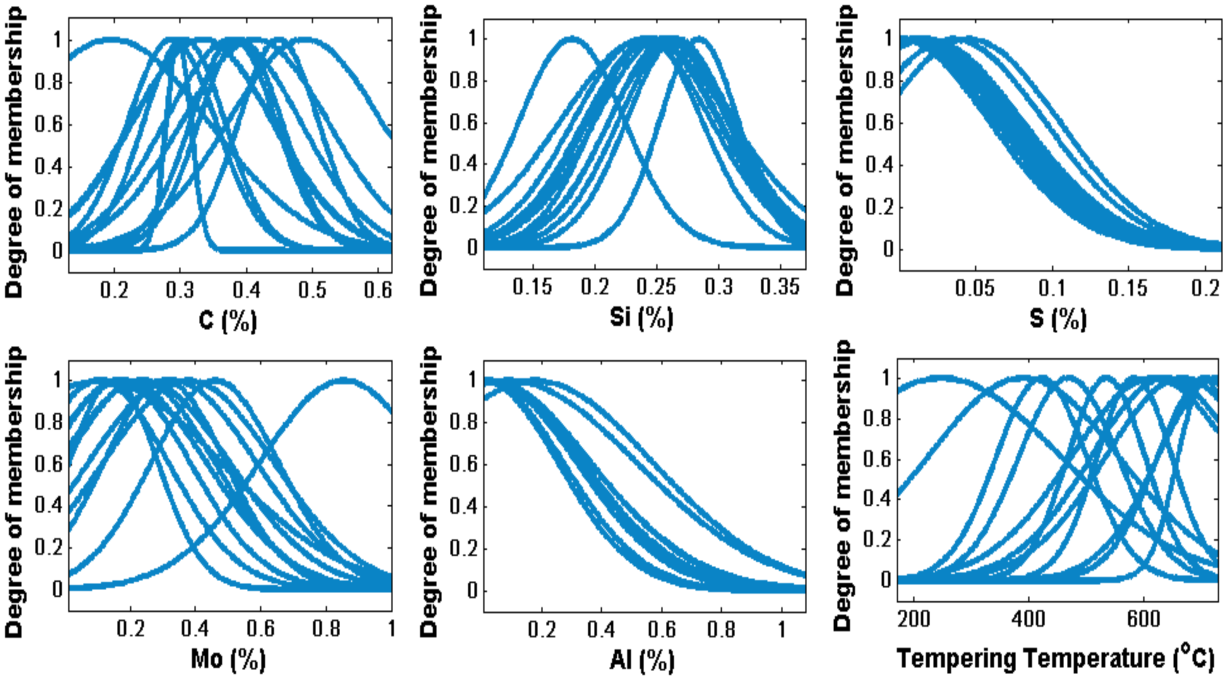


Figure 6.79 The distribution of some membership functions of the 12-rule refined Singleton FRBS (from the second modelling stage) for elongation modelling.

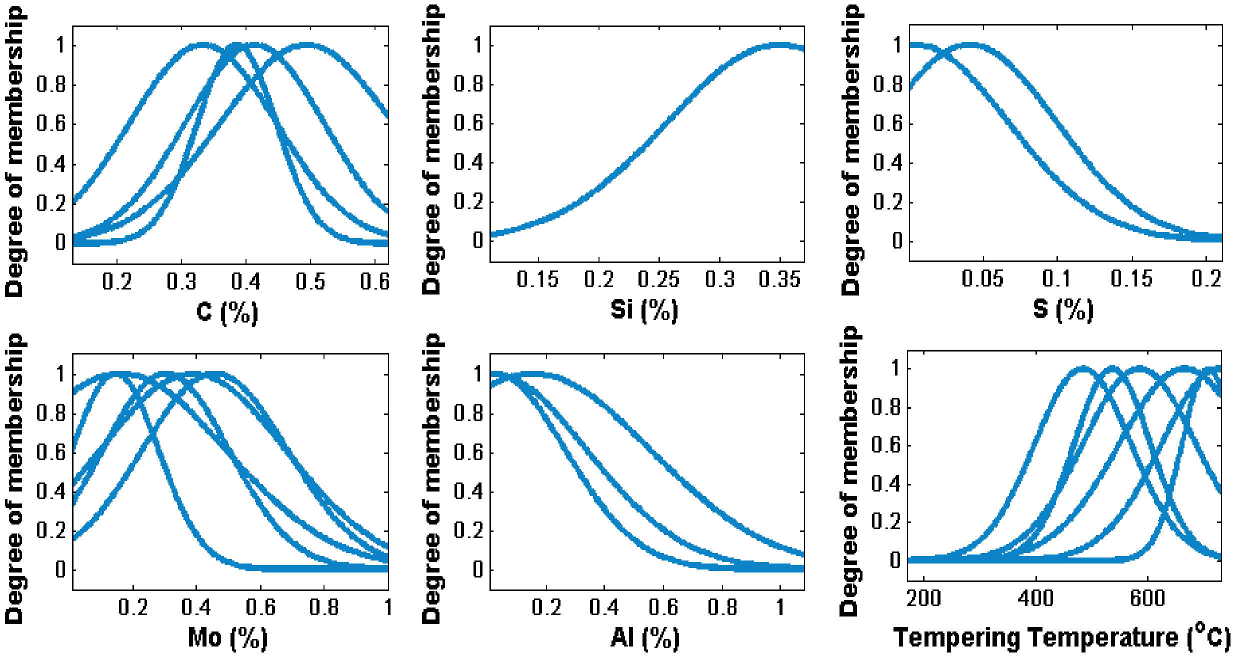


Figure 6.80 The distribution of some membership functions of the 6-rule simplified Singleton FRBS (from the third modelling stage) for elongation modelling.

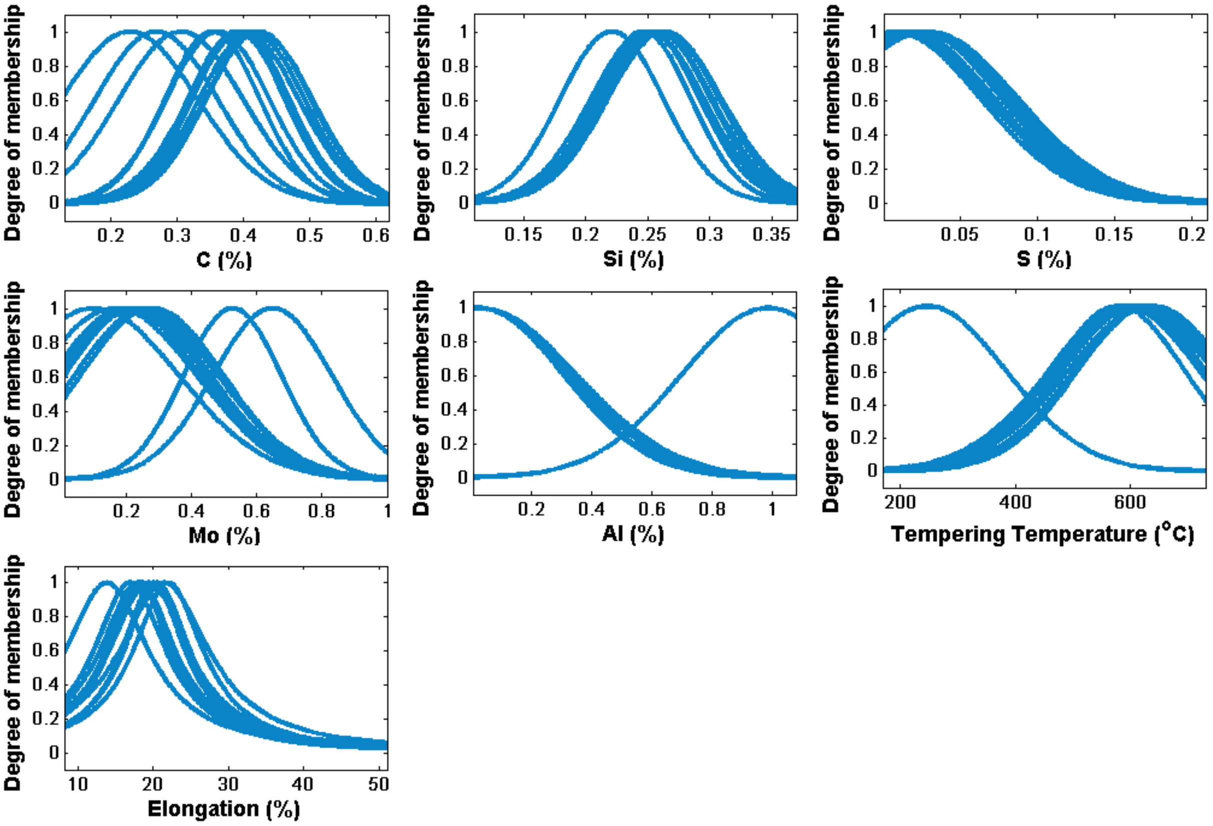


Figure 6.81 The distribution of some membership functions of the 12-rule initial Mamdani FRBS (from the first modelling stage) for elongation modelling.

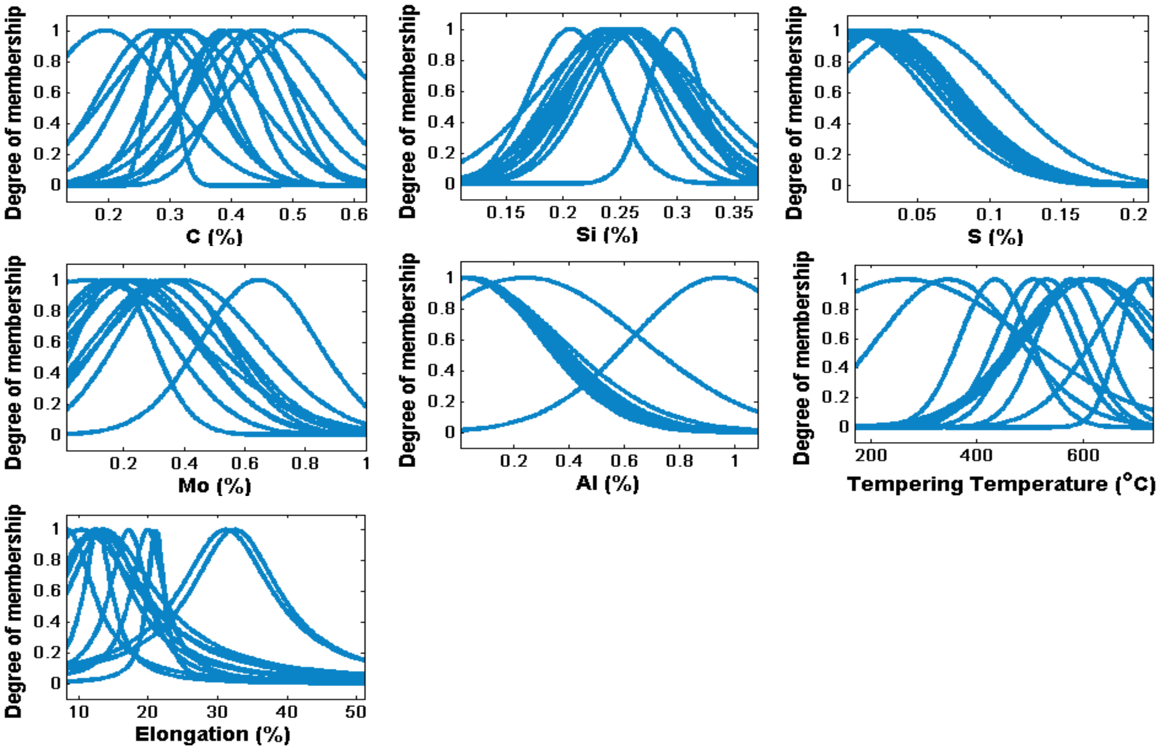


Figure 6.82 The distribution of some membership functions of the 12-rule refined Mamdani FRBS (from the second modelling stage) for elongation modelling.

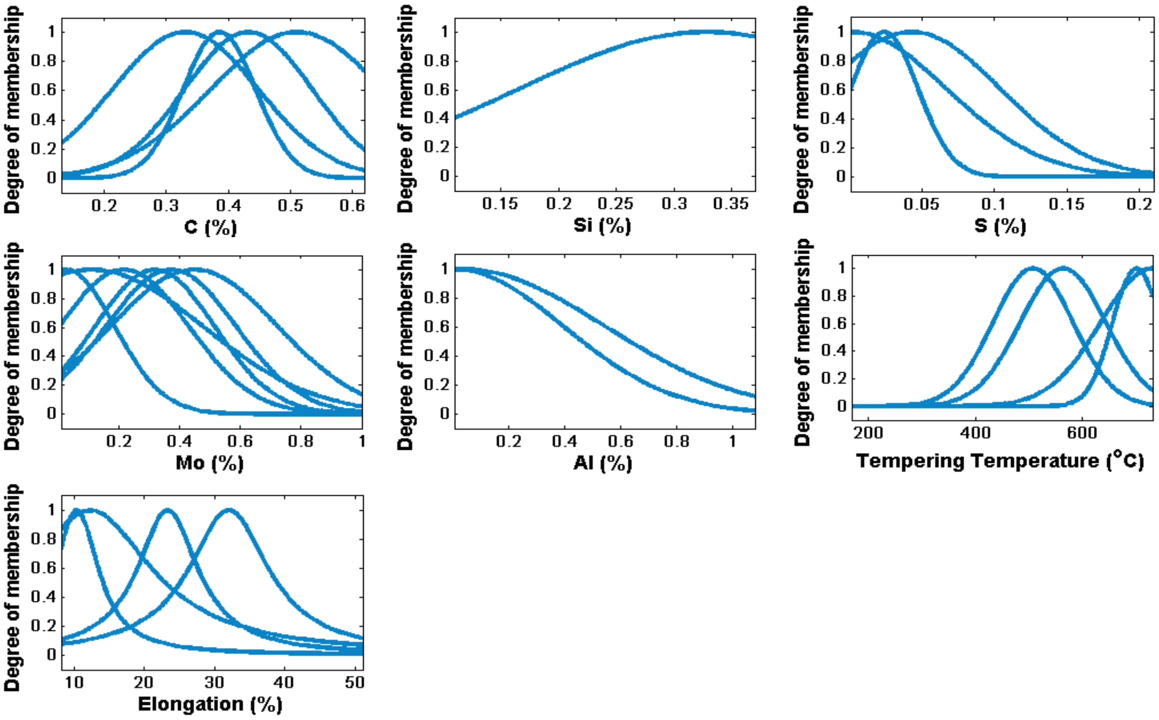


Figure 6.83 The distribution of some membership functions of the 6-rule simplified Mamdani FRBS (from the third modelling stage) for elongation modelling.

Figures 6.84~6.85 show the three-dimensional response surfaces of the simplified Singleton and Mamdani elongation models. The 'median' values of the constant variables are referred to Table 6.5. For gauge length, the 'median' value is set to 5.

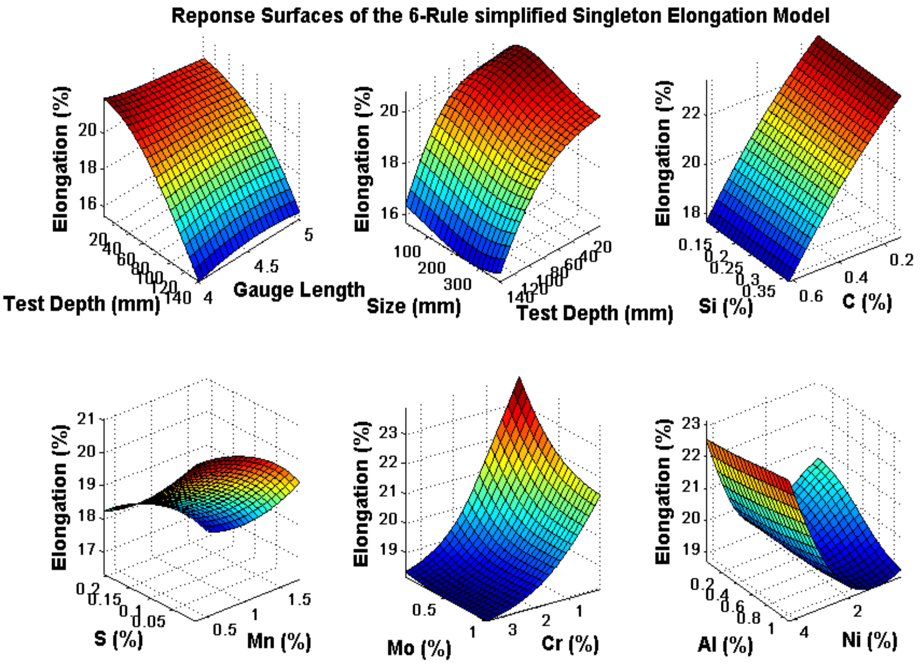


Figure 6.84 Response surfaces of the 6-rule simplified Singleton elongation model.

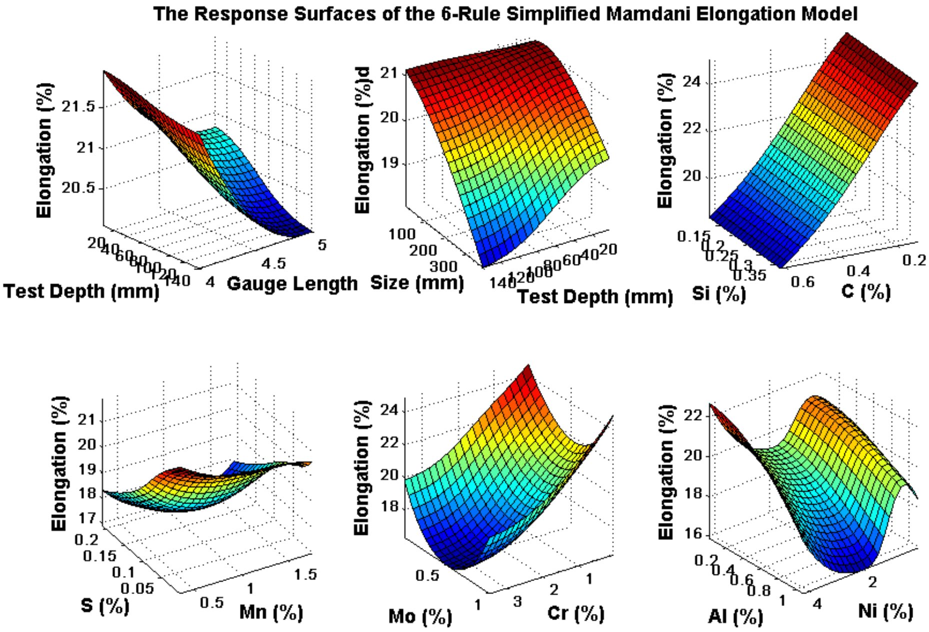


Figure 6.85 Response surfaces of the 6-rule simplified Mamdani elongation model.

The response surfaces of the Singleton and Mamdani ROA models are quite different for some inputs. This can be explained by the data which is very sparse in this case. Figure 6.86 shows the 5%-range confidence band of the 6-rule simplified Singleton FRBS on the elongation data.

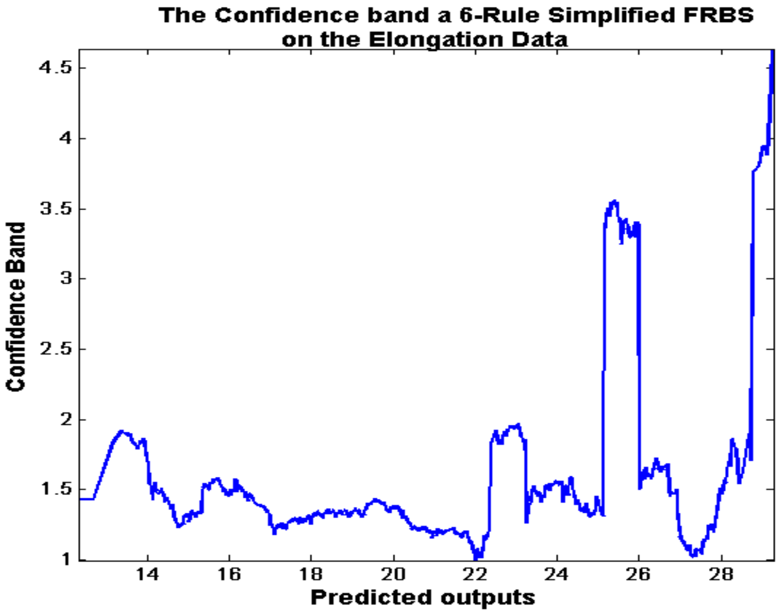


Figure 6.86 5%-range confidence band of the 6-rule simplified Singleton FRBS on the elongation data.

6.4 Comparison between Singleton FRBS and Mamdani FRBS

As shown in Tables 6.1, 6.2, 6.4, 6.8 and 6.10, generally, the predictive performance of the initial Mamdani FRBS in the first modelling stage is slightly worse than that of the singleton FRBS. However, using the proposed BEP updating formulas, the accuracy of such an inaccurate Mamdani FRBS has been improved greatly in the second modelling stage. More importantly, when unseen data are presented to the algorithm, a much better generalisation ability has been observed for the refined Mamdani FRBS comparing to the refined Singleton FRBS. This is confirmed if one looks into Table 6.6 where 12 unseen UTS data samples were used as the validation data. After the third modelling stage, generalisation ability of the Singleton FRBS is much improved due to the removal of the redundancies embedded within its structure. Those redundancies are responsible for the over-fitting of the training data, which may lead to a bad generalisation on unseen situations. Since the singleton FRBS is a special type of TSK model, good generalisation ability using fewer rules is observed for all the case studies. However, with a few more rules, Mamdani FRBS represents a competitive generaliser.

6.5 Summary

In this chapter, the proposed IMOFM is tested via two benchmark problems and is applied to predict the mechanical properties of alloy steels, such as UTS, ROA and elongation. The results show that the proposed IMOFM is a powerful modelling tool in that it can elicit not only accurate but also transparent models. Apart from that, IMOFM also shows its potential in selecting inputs and improving generalisation ability. The experiments have also shown that by using the variable length coding scheme and a new distance index, the problem of the so-called ‘unordered set of rules’ is resolved, which leads to a more efficient optimisation. In the next chapter, a special case of ‘stacked generalisation’ will be examined, which will be used, along with IMOFM, to model the impact energy.

Chapter 7

Improving the Prediction Accuracy of FRBSs

"Stacked generalisation works by deducing the biases of the generaliser(s) with respect to a provided learning set."

David H. Wolpert, Stacked Generalisation, 1992

In this chapter, the concept of 'Stacked generalisation' (Wolpert, 1992) is first introduced, which is a scheme for minimising the generalisation error rate of one or more generalisers (models). A special case of 'Stacked Generalisation' is also proposed in this chapter and applied to the modelling of Impact Energy data set. The results show that the proposed method can generally improve not only the training but also the generalisation performance. The theoretical justification of the proposed special case is also given in this chapter.

7.1 Introduction to Stacked Generalisation

Given the limited information and the presence of embedded systematic errors, it is often hard to learn the underlying process behaviour using the conventional feedback-based supervised learning (CFSL). In such a case, inaccuracies are inherent in the collected data in the form of the contaminated information and/or the shortage of some critical factors. A data-driven learning procedure is by no means able to uncover the underlying flaws-free system by learning directly from such flawed data. The learnt model may fit the learning examples perfectly, however, in terms of generalisation via unseen situations, the same model may perform badly. Figure 7.1 shows the conventional feedback-based supervised learning.

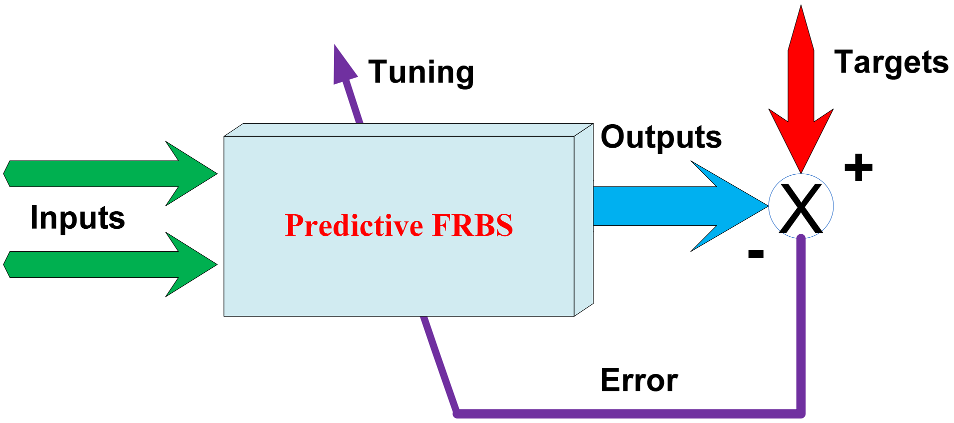


Figure 7.1 The conventional feedback-based supervised learning (CFSL).

‘Stacked Generalisation’ represents an ideal candidate for the aforementioned problem by concatenating an additional learning layer and the original one with the aim of improving the model’s generalisation property without the need for additional information (inputs). The basic idea of ‘Stacked Generalisation’ is to use a high-level model to combine lower-level model(s) to achieve a greater accuracy. ‘Stacked Generalisation’ normally consists of the following two steps (Ting & Witten, 1999):

1. The first step is to collect the output of each model into a new set of data. For each instance in the original training set, this data set represents every model’s predicted output of that instance, along with its observed output;
2. The new data set are treated as the data for another learning problem, and a learning algorithm is employed to solve this problem.

Wolpert (1992) called the original data and the model(s) constructed from them in the first step ‘level-0 data’ and ‘level-0 model(s)’, respectively, while the data and the learning algorithm in the second step are referred to as ‘level-1 data’ and ‘level-1 generaliser’. The process of ‘stacking’ can be iterated, resulting in stacked levels greater than 1. From now on, only 2-level ‘Stacked Generalisation’ is discussed.

There are many variations of ‘Stacked Generalisation’ as long as one follows the mentioned two steps to construct them. However, the primary implementation is as the technique for combining multiple generalisers. In such a case, ‘Stacked Generalisation’ can be viewed as a more sophisticated version of non-parametric statistics techniques like cross-validation. It provides a strategy by combining a set of generalisers rather than ‘winner-takes-all’. An instance of this type of 'Stacked Generalisation' in the field of ANNs is network ensembles (Krogh *et al.*, 1995). Theoretical proof that diversity of networks can lead to reduced generalisation error has been given by Sollich & Krogh (1996) by considering the task of approximating a target function f_0 from R^N to R. The target function is denoted $y(x)$ and only noisy samples of the target function can be obtained. The inputs x are taken to be drawn from a distribution $P(x)$. If an ensemble of K independent predictors $f_k(x)$ is available, a weighted ensemble average (the final output of the ensemble) is denoted as follows:

$$\overline{f}(x) = \sum_k w_k \cdot f_k(x) \qquad (7.1)$$

Where, w_k is a weight representing the strength of 'belief' in each predictor, which has a positive value and sums to one. For an input x, the error of the ensemble $\varepsilon(x)$, the error of the kth predictor $\varepsilon_k(x)$, and its 'ambiguity' $\alpha_k(x)$ are defined as follows:

$$\varepsilon(x) = \left(y(x) - \overline{f}(x)\right)^2 \qquad (7.2)$$

$$\varepsilon_k(x) = (y(x) - f_k(x))^2 \qquad (7.3)$$

$$\alpha_k(x) = \left(f_k(x) - \overline{f}(x)\right)^2 \qquad (7.4)$$

The error of ensemble can also be written as follows:

$$\varepsilon(x) = \overline{\varepsilon}(x) - \overline{\alpha}(x) \qquad (7.5)$$

Where, $\overline{\varepsilon}(x) = \sum_k w_k \cdot \varepsilon_k(x)$ and $\overline{\alpha}(x) = \sum_k w_k \cdot \alpha_k(x)$. When averaged over the input distribution $P(X)$, the following ensemble generalisation error is obtained as follows:

$$\varepsilon = \overline{\varepsilon} - \overline{\alpha} \qquad (7.6)$$

Sollich & Krogh pointed out that Eq. 7.6 is important in that it separates the generalisation error into a term that depends on the generalisation errors of the individual predictors and another term that contains all *correlation* between the predictors. Hence, the more the predictors differ, the lower the error will be, given ε_k remain constant.

Instead of viewing 'Stacked Generalisation' as an extension of concepts such as cross-validation, Wolpert (1992) argued that it can also be viewed as a means of collectively using all predictors to estimate their own generalising biases with respect to a particular training set, and then filter out those biases. This description leads to another primary implementation which only has a single level-0 generaliser. In such a case, 'Stacked Generalisation is a scheme for estimating the errors of a generaliser and then correcting those errors. In the next sections an Error Correction Scheme (ECS) will be introduced first, which falls into the second implementation discussed above. A mathematical proof regarding how much one can improve the predictive performance via the ECS will then be given. Impact Energy is employed as the case study to show the validity of the proposed ECS. Some possibilities to extend the current ECS are also discussed at the end of the chapter.

7.2 Basic Ideas for Prediction Improvements

7.2.1 Error Correction Scheme

Here, a special case of 'Stacked Generalisation' is presented, which relates to the case of when the first layer contains only one generaliser. In such a case, 'Stacked Generalisation' is reduced to a scheme for estimating the error of the model in the first layer. Figure 7.2 shows the special case of the 'Stacked Generalisation' based on the ECS.

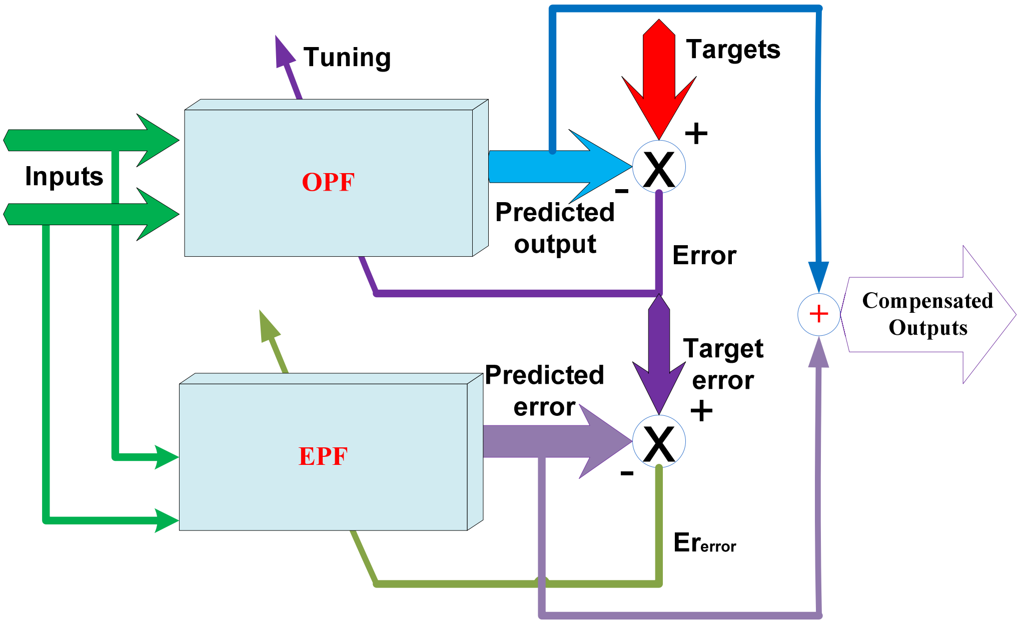


Figure 7.2 'Stacked Generalisation' based on the ECS.

The basic idea of ECS is to build an Error Predictive FRBS (EPF) apart from the Original Predictive FRBS (OPF) so that one can predict the errors associated with the OPF given the inputs of OPF. When a new scenario is encountered, the EPF will be able to predict the potential error and thus the predicted error can be used to compensate for the predicted output produced by the OPF. An improved predictive accuracy in terms of not only the learning but also the generalisation should be expected.

7.2.2 Theoretical Justification

A logical question relating to the proposed ECS may be as follows: how much exactly can one improve the predictive performance via the ECS? The following mathematical deduction will answer this question.

To measure the predictive performance of OPF, RMSE is used as follows:

$$RMSE_{OPF} = \sqrt{\frac{\sum_{m=1}^{N}\left(y_{prediction}(m)-y_{target}(m)\right)^2}{N}} \tag{7.7}$$

Where, N is the number of learning examples, $y_{target}, y_{prediction}$ are the targeted and predicted outputs. If the error produced by the OPF is defined using Eq. 7.8, the predictive performance of EPF is given by Eq. 7.9:

$$Error(m) = y_{target}(m) - y_{prediction}(m), \qquad m = 1, \ldots, N \tag{7.8}$$

$$RMSE_{EPF} = \sqrt{\frac{\sum_{m=1}^{N}\left(Error_{prediction}(m)-Error(m)\right)^2}{N}} \tag{7.9}$$

Where, $Error_{prediction}$ is the predicted error produced by EPF. Hence, the compensated outputs are calculated using Eq. 7.10:

$$y_{compensated}(m) = y_{prediction}(m) + Error_{prediction}(m), \qquad m = 1, \ldots, N \tag{7.10}$$

Hence, the predictive performance of the ECS can be calculated and rearranged by substituting Eq. 7.8 and 7.9 into 7.11 as follows:

$$RMSE_{ECS} = \sqrt{\frac{\sum_{m=1}^{N}\left(y_{compensated}(m) - y_{target}(m)\right)^2}{N}} \Longrightarrow$$

$$\Rightarrow \sqrt{\frac{\sum_{m=1}^{N}\left(y_{prediction}(m)+Error_{prediction}(m)-y_{target}(m)\right)^2}{N}} = \sqrt{\frac{\sum_{m=1}^{N}\left(Error_{prediction}(m)-Error(m)\right)^2}{N}} \triangleq RMSE_{EPF} \qquad (7.11)$$

Hence, with the ECS, one can improve the predictive performance of the OPF to the predictive performance of the EPF on the learning examples.

7.3 Experimental Studies on Impact Energy

The proposed ECS is used to model Charpy toughness (impact Energy) which is featured as the imprecise and scattered multidimensional data. However, the repeatability of the measurements of Charpy test is considerably poor due to the unknown internal fracture dynamics which propagate the energy during the fracture stage in an almost random manner. Hence, repeating the test a number of times, for the same input conditions, may result in measurements within a certain output space region but with some variability (Panoutsos and Mahfouf, 2008). Due to the constraints on the costs associated with such measurements, a very imprecise and sparse data set is obtained. Also, the industrial/customer demands a robust process in order to predict the toughness properties of steels. Hence, a modelling approach which can provide consistent predictions, even in the regions of low data density and high scatter, is required.

The variables used for the construction of the impact model, together with statistics of the data are shown in Table 7.1. Figure 7.3 shows the data distribution on some of the dimensions incolved.

TABLE 7.1
THE INPUTS AND OUTPUT OF IMPACT ENERGY DATA SET

Inputs	**Min.**	**Max.**	**Mean**	**SD.**
Test Depth (mm)	5.5	146.05	20.8	14.5032
Bar Size (mm)	11	381	172.488	80.839
Test Site (2-6)	2	6	3.7965	1.1219
C (%)	0.13	0.52	0.3942	0.0575
Si (%)	0.11	0.38	0.2548	0.0318
Mn (%)	0.41	1.75	0.8409	0.2172
S (%)	0.0008	0.052	0.0167	0.0089
Cr (%)	0.11	3.25	1.0752	0.2447
Mo (%)	0.02	0.98	0.2394	0.086
Ni (%)	0.03	4.21	0.3683	0.5192
Al (%)	0.003	0.047	0.027	0.0048
V (%)	0.001	0.26	0.0077	0.0223
Hardening Temperature (^{0}C)	810	980	864.0157	15.4689
Cooling Medium (1-3)	1	3	2.0855	0.415
Tempering Temperature (^{0}C)	190	730	647.1927	49.9249
Impact Temperature (^{0}C)	-59	23	-5.7869	26.4486
Output: Impact Energy (J)	3.4667	245.3333	89.6419	32.9701

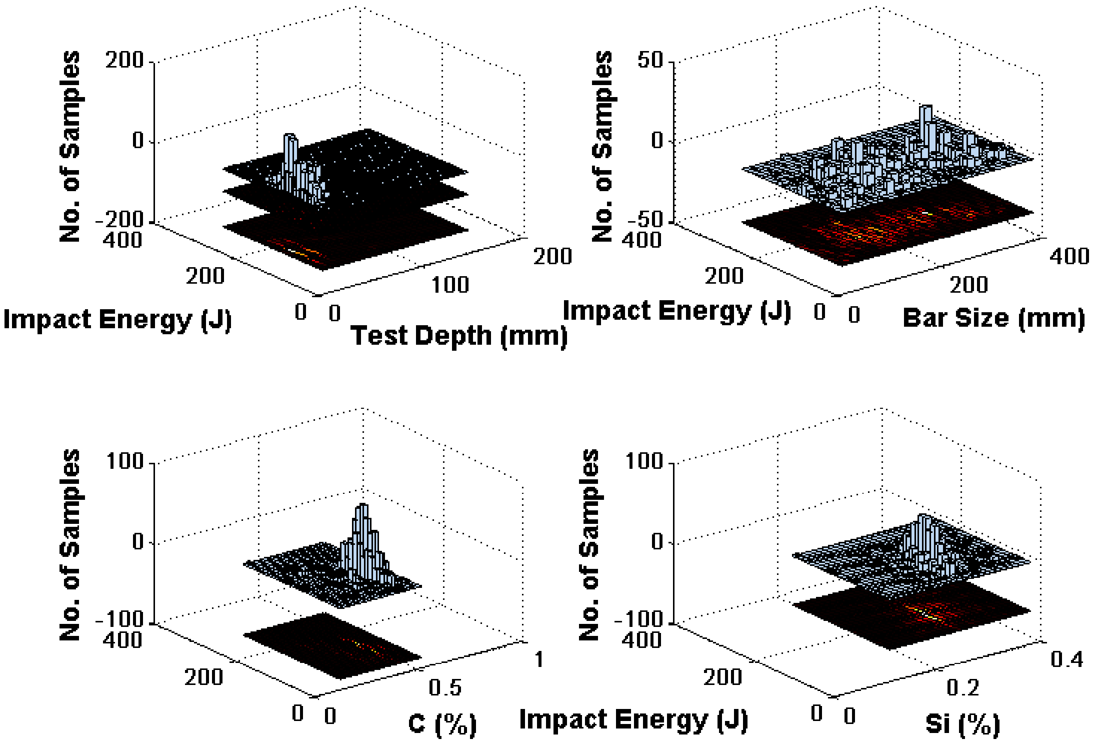


Figure 7.3 Sample of Impact Energy data space.

To evaluate the proposed ECS, it was decided to apply its associated algorithm to the modelling of impact energy of steel data which consists of 1661 data samples. 75% of samples are used as the training data and the rest for testing. An 11-rule FRBS is first generated using G3Kmeans clustering, which is then refined further via a back-propagation algorithm. The refined FRBS is used to seed the third modelling stage of IMOFM to generate a set of Pareto FRBSs. Hence, a set of OPFs is formed by utilising the obtained Pareto FRBSs. Then, a set of 15-rule FRBSs (for OPFs having more than 9 rules) and 11-rule FRBSs (for OPFs having less than 10 rules) is used to build the corresponding EPFs in the way described in Section 7.2.1. In doing so, one can investigate the improved predictive performance of the ECS not only for a particular OPF but also for a set of OPFs.

The number of iterations of the second modelling stage of IMOFM is set to 250, and the number of iterations of the third modelling stage is set to 1200. All other parameters are kept the same as those in chapter 6. For the ease of analysing, only Singleton FRBS is employed without any loss of generality. Table 7.2 summarises the predictive performances of the Pareto FRBSs (only a selection of them among 34 Pareto FRBSs is presented). The results of IMOFM modelling in terms of the Pareto fronts, the predictive performances from each modelling stage and the membership function distributions can be found in Appendix B, where a 6-rule simplified FRBS is used as an example. Table 7.3 summarises the predictive performances of EPF, OPF and ECS. As one can see from the table, by including the proposed ECS, the predictive performances of the Pareto FRBSs are improved.

TABLE 7.2

COMPARISONS OF THE PREDICTIVE PERFORMANCES FOR THE DIFFERENT MODELING METHODS USING THE IMPACT ENERGY DATA

Modeling Methods (Ref.)	No. of rules	No. of fuzzy sets in inputs and output	Performance			
			Training (RMSE)	Testing (RMSE)	Training (R-Square)	Testing (R-Square)
Q. Zhang & M. Mahfouf (2007)						
Initial FRBS	15	Input:[15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15]; output: 15	30.54[*]	31.44[*]	-	-
Pareto FRBS1	15	Input: [12 15 14 13 15 13 12 14 13 12 13 15 13 11 15]; output: 11	14.35[#]	17.10[#]	-	-
Pareto FRBS2	8	Input: [8 8 8 7 6 7 7 8 7 7 7 5 7 7 4 7]; output: 8	17.85[#]	19.03[#]	-	-
IMOFM_S (NB: Average results over 10 runs are presented here)						
Initial FRBS	11	Input: [11 11 11 1 11 11 11 11 11 11 11 11 11 11 11 11]; output: 11	(30.72[*]/ 15.47[@])	(30.13[*]/ 17.19[@])	(0.3777[*]/ 0.8858[@])	(0.3934[*]/ 0.8536[@])
Pareto FRBS1	10	Input: [7 8 8 9 6 8 9 6 7 5 6 3 8 9 8 9]; output: 10	14.96[#]	17.36[#]	0.8921[#]	0.8474[#]
Pareto FRBS2	10	Input: [7 8 9 8 5 8 9 7 5 6 6 2 6 8 8 8]; output: 10	15.28[#]	17.52[#]	0.8871[#]	0.8449[#]
Pareto FRBS3	9	Input: [6 7 7 7 5 7 8 7 6 4 5 3 5 8 8 7]; output: 9	15.73[#]	17.91[#]	0.8797[#]	0.8367[#]
Pareto FRBS4	9	Input: [6 7 7 7 4 7 8 7 6 4 5 3 5 8 8 7]; output: 9	15.83[#]	18.15[#]	0.8783[#]	0.8320[#]
Pareto FRBS5	7	Input: [4 7 6 6 3 6 5 6 3 2 6 2 4 5 5 6]; output: 7	16.41[#]	17.61[#]	0.8685[#]	0.8428[#]
Pareto FRBS6	7	Input: [4 6 5 6 3 5 5 4 4 1 4 1 2 4 5 6]; output: 7	16.89[#]	18.39[#]	0.8606[#]	0.8271[#]
Pareto FRBS7	6	Input: [4 6 5 4 2 3 5 6 4 3 3 2 4 5 6 6]; output: 6	17.68[#]	19.38[#]	0.8452[#]	0.8058[#]
Pareto FRBS8	6	Input: [3 6 5 4 2 3 5 5 3 3 3 2 4 4 6 6]; output: 6	17.80[#]	19.62[#]	0.8432[#]	0.8006[#]
Pareto FRBS9	5	Input: [4 4 5 5 2 3 3 5 3 2 3 2 3 4 4 4]; output: 5	17.82[#]	18.59[#]	0.8426[#]	0.8221[#]
Pareto FRBS10	5	Input: [2 5 3 3 1 1 2 4 1 1 2 1 3 2 3 4]; output: 5	19.20[#]	19.58[#]	0.8144[#]	0.8002[#]

[*] Initial model extracted directly from data using clustering algorithms or grid partition methods.

[@] Refined model or the consequents are computed through the estimation methods.

[#] Simplified model after model simplification and parameter fine tuning.

TABLE 7.3
TRAINING AND TESTING RESULTS FROM OPF, ECS AND EPF

OPF Configurations	OPF				ECS				EPF		Improvement	
	Training		Testing		Training		Testing		Training	Testing	Training (%)	Testing (%)
	RMSE	R^2	RMSE	R^2	RMSE	R^2	RMSE	R^2	RMSE	RMSE		
Initial model	15.47	0.88	17.19	0.85	15.09	0.89	16.74	0.86	15.09	16.74	2.5	2.6
Pareto FRBS1	14.96	0.89	17.36	0.84	14.62	0.90	17.01	0.85	14.62	17.01	2.2	2.0
Pareto FRBS2	15.28	0.89	17.52	0.84	14.50	0.90	17.18	0.85	14.50	17.18	5.1	1.9
Pareto FRBS3	15.73	0.88	17.91	0.84	15.55	0.88	17.78	0.84	15.55	17.78	1.1	0.7
Pareto FRBS4	15.83	0.88	18.15	0.83	15.59	0.88	17.93	0.84	15.59	17.93	1.5	1.2
Pareto FRBS5	16.41	0.87	17.61	0.84	15.77	0.88	17.12	0.85	15.77	17.12	3.9	2.8
Pareto FRBS6	16.89	0.86	18.39	0.83	15.76	0.88	17.14	0.85	15.76	17.14	6.7	2.8
Pareto FRBS7	17.68	0.84	19.38	0.80	16.21	0.87	18.58	0.82	16.21	18.58	8.3	4.1
Pareto FRBS8	**17.80**	**0.84**	**19.62**	**0.80**	**15.17**	**0.89**	**17.83**	**0.84**	**15.17**	**17.83**	**14.8**	**9.1**
Pareto FRBS9	17.82	0.84	18.59	0.82	17.01	0.86	18.41	0.83	17.01	18.41	4.5	0.9
Pareto FRBS10	19.20	0.81	19.58	0.80	17.60	0.85	19.03	0.81	17.60	19.03	8.3	2.8

In the following space, Pareto FRBS8 (a 6-rule simplified FRBS) is taken as an example to demonstrate the various aspects associated with the EPF, OPF and ECS.

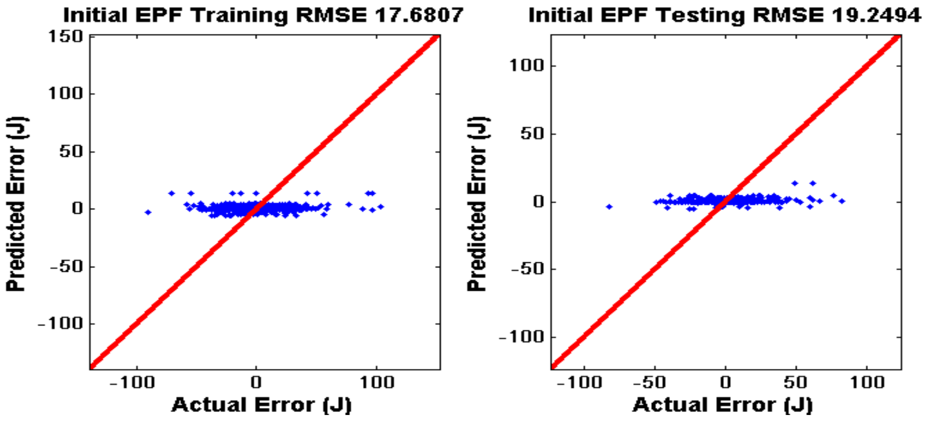


Figure 7.4 The predictive performance of the initial EPF.

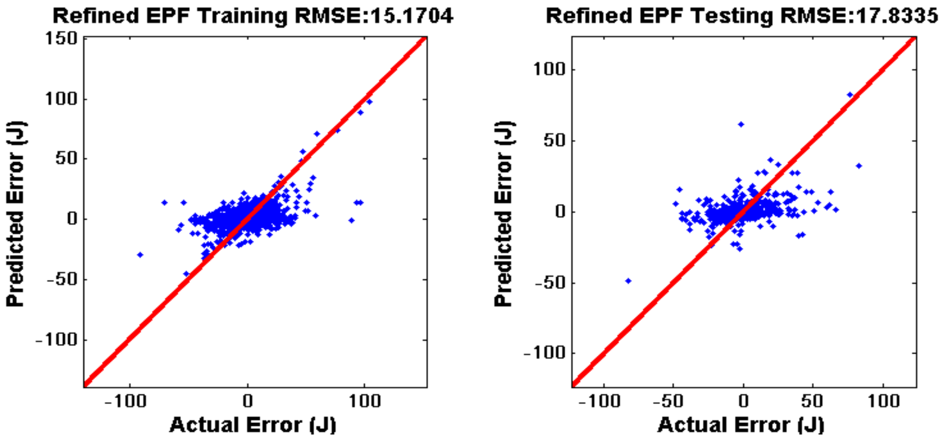


Figure 7.5 The predictive performance of the refined EPF.

Figures 7.4 and 7.5 show the predictive performances of the initial and the refined EPFs. It can be seen from the figure that the refined EPF correctly predicts some errors, which may be largely associated with the systematic error induced either by OPF or the data itself. For the errors which are close to the red line, it is most related to the noise when the data is collected. No model can predict the white noise. The improvement in the predictive performance of the ECS is mainly attributed to those embedded systematic errors which can be corrected after the compensation. Figure 7.6 shows the training process of the EPF.

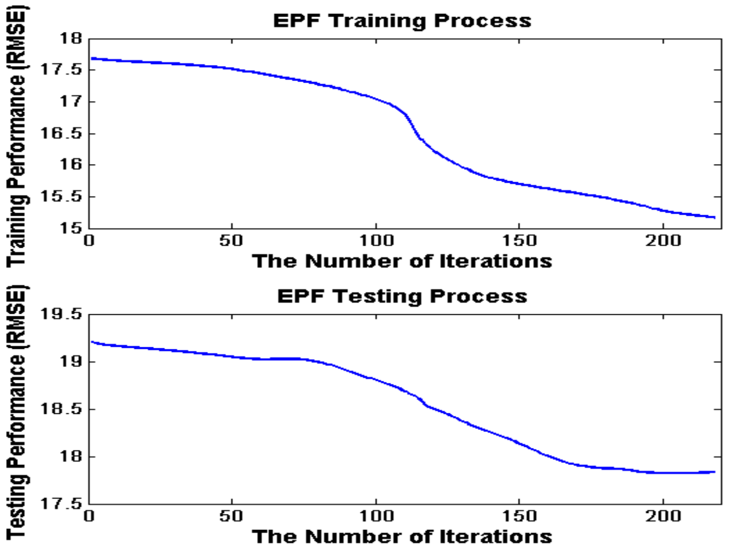


Figure 7.6 The training process of the EPF.

Figures 7.7 and 7.8 show the predictive performances of the OPF and the ECS.

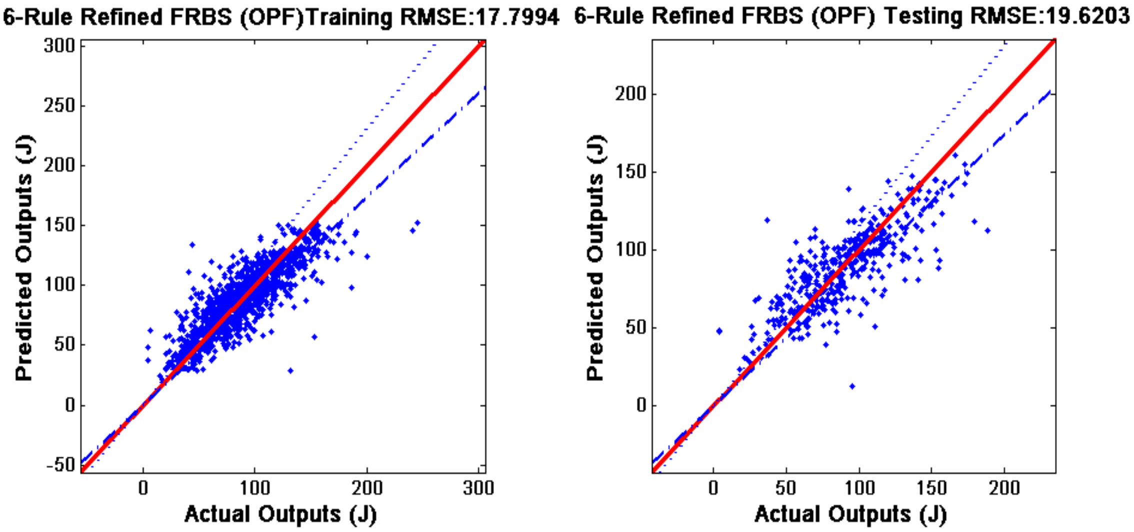


Figure 7.7 The predictive performance of the OPF.

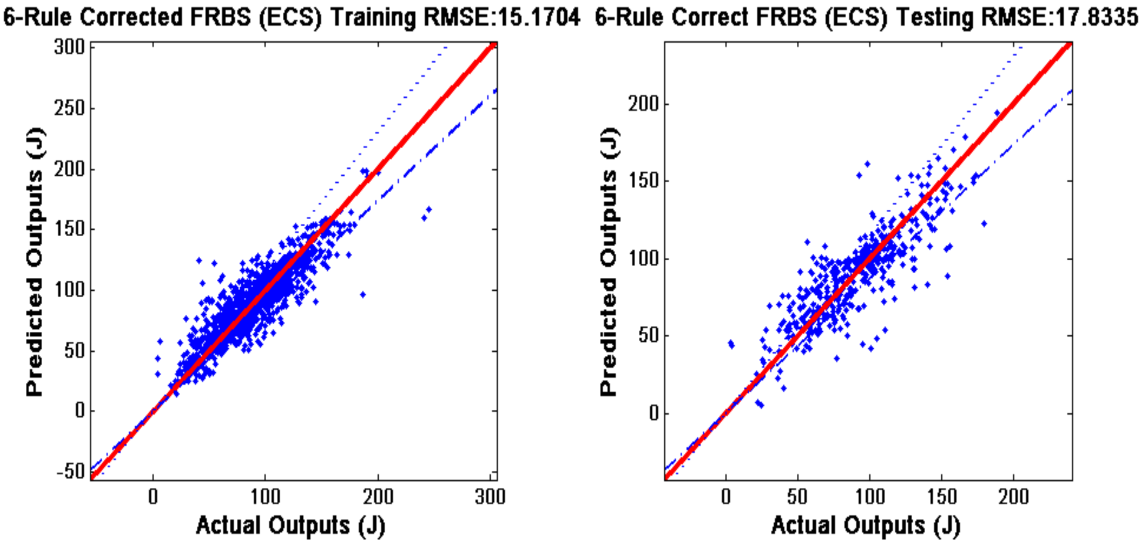


Figure 7.8 The predictive performance of the ECS.

As can be seen from Figures 7.7~7.8, the predictions given by the corrected model are more close to the red line.

7.4 Model Confidence Bands

The confidence bands introduce another type of measures which quantify how reliable the elicited model is in particular regions. Such information has been combined into the EPF as a part of inputs with the hope of obtaining a more accurate EPF, given that more relevant information is now available. However, as indicated by Table 7.4, after compensation, the improvements in the predictive performance is not as good as those presented in Table 7.3.

TABLE 7.4
TRAINING AND TESTING RESULTS FROM OPF, ECS AND EPF USING CONFIDENCE BAND AS THE EXTRA INPUT

OPF Configurations	OPF				ECS				EPF		Improvement	
	Training		Testing		Training		Testing		Training	Testing	Training (%)	Testing (%)
	RMSE	R^2	RMSE	R^2	RMSE	R^2	RMSE	R^2	RMSE	RMSE		
Pareto FRBS7	17.68	0.84	19.38	0.80	16.91	0.85	18.89	0.82	16.91	18.89	4.3	2.5
Pareto FRBS8	17.80	0.84	19.62	0.80	17.35	0.85	19.09	0.81	17.35	19.09	2.5	2.7

Possible explanations are as follows:

1. The calculation of the confidence values is still based on the model's predictions and its bias to the actual ones. $Error(m)$ (refer to Eq. 7.8) has already provided such

information by including $Error(m)$ as the output of the EPF's learning examples. Hence, including the confidence values does not necessarily lead to more information.

2. Indeed, including the confidence values as another input in the EPF's learning examples leads to a situation where the consequence is inferred by the consequence itself.
3. The confidence values are only a coarse measure which calculates the deviation within a scope S_r (refer to section 6.1.3). Hence, it is not specific enough to let the learning algorithm learn how to correct a particular error for a specific instance.

As Wolpert (1992) indicated, the type of generaliser that is suitable to derive the higher-level model and the type of attributes that should be used as its inputs remain as a 'black art' in the design of 'Stacked Generalisation'. The same problem was encountered in this research during the development of the ECS. For example, the number of rules which are appropriate for the EPF, and whether extra information, such as confidence bands, should be included in the inputs of the EPF remain open issues which deserve more attention in the future.

7.5 Summary

The development of a reliable empirical model is a key step towards realising model-based process control and monitoring. The proposed ECS is robust and particularly suitable for the case of imprecise and scattered data. Although, the work presented here relates to the use of FRBS as a modelling tool, it is not limited to such an implementation. In fact, ECS is a very general scheme which can be implemented via various modelling methods. In the next chapter, conclusions of this thesis and the future research directions will be discussed.

Chapter 8

Conclusions and Future Work

Every scientific endeavour tries to find the answers to the problems at hand and in doing so, raises several others. The work presented in this thesis is not an exception. Since it proposed to answer the following 3 questions:

1. How to use bio-inspired paradigms to account for the problems involving multiple conflicting goals?
2. How to automate the process of acquiring transparent knowledge from high dimensional data without too much damage to the predictive performance of the extracted knowledge base (e.g. FRBS)?
3. How to improve the predictive performance of the elicited model if it is driven by imprecise data?

This final chapter summarises what has been achieved in answering the above three questions and what are the open questions that deserve further research efforts.

8.1 Conclusions

To answer the first question, a novel Population Adaptive based Immune Algorithm (PAIA) and a multi-stage optimisation procedure for solving MOP were proposed. These algorithms are inspired by four immunological models, namely the Clonal Selection Principle, Immune Network Theory, Vaccination and Secondary Response and adaptive antibody's concentration. The algorithms have been tested with ZDT and DTLZ test suites, and in all cases have been shown to be insensitive to the initial population size. The population and clone size are adaptive with respect to the search process and the problem at hand. It is argued that the algorithm can largely reduce the number of evaluation times and is more consistent with the vertebrate immune system than the previously proposed algorithms. Results also suggest that the algorithms are valuable alternatives to already established evolutionary based optimisation algorithms, such as NSGAII (Deb, 2001), SPEA2 (Zitzler, Laumanns, and Thiele, 2001) and VIS (Freschi and Repetto 2005). A general framework is extracted from the PAIA as the guide to design immune algorithms, under which clear definitions of immune operators and their roles are provided.

Some common features included in by most modern heuristic search methods, especially within the field of real-valued optimisation, were discussed during the course of answering the first question, which are summarised as follows:

1. The offspring should be generated around the parents. The better the parents are, in terms of their fitness (or 'affinity' in AIS terminologies), the closer to the parents the offspring should be. In doing so, a widespread search (exploring) in the early stage of the optimisation is ensured and a more elaborate search (exploitation) in the late stage is emphasised.
2. As far as the real-valued optimisation is concerned, if a heuristic search method is implemented following the rule mentioned above, there will be no distinction between the commonly used terms such as 'crossover' and 'mutation'. Instead, 'recombination' and/or 'variation operator' are more precise terms to describe such proliferation behaviour inherent in most heuristic search methods.
3. The initial population size is no longer the only way to maintain the diversity of the population. One can always insert newcomers or intensify the 'variation' effort (by generating new solutions which are relatively far from the parents) in order to achieve such diversity (we will discuss this a bit more in the Section 8.2). Hence, the initial population size should not be an important factor any more.
4. As far as multi-objective optimisation problems are concerned, the multi-stage optimisation procedure may represent a more suitable solution, which allows a more focused and direct search in the first stage when solutions are still some way from the Pareto front. In the second stage, non-dominated concept can be included to extend the already found solutions into other areas of the Pareto front.

The fundamental differences between AIS and other evolutionary algorithms are also identified through their reproduction mechanism, selection scheme, evolution strategy, population control, diversity preservation and fitness (affinity) assignment (refer to Section 3.6.2).

In order to answer the second question, an evolutionary based clustering algorithm (G3Kmeans) and a multi-stage immune based multi-objective fuzzy modelling (IMOFM) method were proposed.

The proposed clustering algorithm is used to induce a coarse fuzzy rule-base from data. The method was tested extensively through the artificial and real data sets. The results show that the proposed algorithm is superior to other more traditional clustering algorithms in that:

1) It is robust to different initial settings;
2) It can approach very closely to the global optimal partitions, especially for high-dimensional problems;
3) It is computationally more efficient compared to other evolutionary based clustering algorithms.

The proposed IMOFM adopts a multi-stage modelling procedure and a variable length coding scheme to account for the enlarged search space due to the simultaneous optimisation of the rule-base structure and its associated parameters. The proposed modelling method applies to both Singleton FRBS and Mamdani FRBS.

The following points have been learnt during the development of IMOFM and are considered as important factors for any multi-objective fuzzy modelling algorithms:

1. There currently exist two different multi-objective based fuzzy modelling streams to tackle the interpretability issues: the first stream is mainly concerned with the linguistic modelling, in which a set of pre-specified fuzzy partitions are given *a priori* by experts or users (grid partition); the task is then to elicit an optimal FRBS in terms of its compactness and performance; the second stream generally takes the approximate fuzzy model as the start point; hence, the task is to improve the model's explanatory ability, which may have been lost due to the automatic learning process. Both streams end up with a 'semi-linguistic and semi-approximate' form after optimisation. However, as their names suggest, the first stream is more suitable for low dimensional problems with high requirement of interpretability, such as classification problems, and the second stream is more suitable to tackle problems with high dimensionality and high requirement of predictive performance, such as

approximation problems. Hence, anyone who wishes to enrich this exciting research field should consider first which stream is more appropriate to the problems at hand. In Section 8.2, the possibility of separating the knowledge base from the predictive model is discussed so that each of them can serve for a different purpose.

2. Rules should be realigned before any optimisation and simplification. Otherwise, it will result in the so-called 'unordered set of rules'. More importantly, if the optimisation is operated on the rule bases before realignment, it will break the rule of 'always proliferating around parents'. Such a rule is now widely accepted by the practitioners in the field of real-valued optimisation as key to success.

In order to answer the third question, a special case of 'Stacked Generalisation', viz. the ECS, was proposed. The basic idea of ECS is to build an Error Predictive FRBS (EPF) apart from the Original Predictive FRBS (OPF) so that one can predict the errors associated with the OPF given the inputs of OPF. When a new scenario is encountered, the EPF will be able to predict the potential error, and thus the predicted error can be used to compensate the predicted output produced by the OPF. An improved predictive accuracy in terms of not only the learning but also the generalisation was observed. The proposed scheme is particularly suitable to model imprecise data where systematic error is embedded.

8.2 Future Research Directions

As mentioned at the beginning of this chapter, when the immediate problems are solved, new problems will arise and they, too, should be investigated. Such problems remain as open questions and are discussed in the following part.

(1) **How to more efficiently generate newcomers?** In Chapter 3, it is believed that the diversity of the population can be further improved by inserting newcomers during the search process. Previous research normally inserts randomly generated individuals at each iteration step. In this way, meaningless individuals may be generated in 'not-so-good' regions which have already been searched (explored). Hence, investigations into ways that can form meaningful newcomers deserve more attention and the negative selection principle (Esponda *et al.*, 2004) may play a significant role in this process.

(2) **How to develop a unified MOP scheme?** A unified multi-objective optimisation scheme which in the early stage can focus on a particular pathway leading to the global optimum and in the later stage can extend such optimum into other parts of the Pareto front deserves more attention. The multi-stage optimisation procedure described in Chapter 3 is just an initial step towards such a unified MOP scheme.

(3) **Can one make unsupervised clustering more supervised? And how to automatically (systematically) define the number of clusters?** In Chapter 4, an evolutionary algorithm-based clustering algorithm was discussed. There are several new advancements in the clustering field that can easily make their ways into the current clustering based fuzzy modelling framework. One of such possibilities is to use supervised clustering (Setnes 2000; Gonzάlez *et al.*, 2002). The difference between supervised clustering and conventional clustering lies in that, as its name suggests, supervised clustering specifically makes use of the output information. Hence, the supervised clustering result is one more step close to the refined model, which makes supervised clustering scheme more suitable for function approximation problems. Clustering methods which can automatically decide the number of clusters deserve more attention. Sheng *et al.*, (2006) proposed that the number of clusters can be obtained automatically through minimizing cluster validity index, rather than a within-cluster-distance. Handl *et al.*, (2004) adopted a multi-objective optimisation framework to determine the number of clusters. All these methods can be used to enrich the current work.

(4) **Can generalising measures be devised to evaluate the modelling results from different multi-objective fuzzy modelling algorithms?** Through the discussions in Chapter 5, one may notice that it is not easy to categorically comment on the modelling results due to the stochastic nature of all EAs-based fuzzy modelling methods. The current solution is to run the IMOFM several times and average the results of each Pareto FRBS configuration. However, the Pareto FRBS configurations found by different fuzzy modelling approaches or even by different runs with the same algorithm may not be exactly the same, which causes the difficulty as far as the comparison is concerned. Hence, the performance metrics which can facilitate the comparison between different algorithms and runs, such as the generational distance and spread introduced in Section 3.2.2 for multi-objective optimisation algorithms, deserves more attentions.

(5) **Can the knowledge base be separated from the predictive model so that they can serve for different purpose?** As discussed in Chapter 5 and Section 8.1, a 'semi-linguistic and semi-approximate' fuzzy model is the best resort for the interpretability issues. However, too much obligations have been put on a single model. On the one hand, the model should predict well, while on the other hand, the model should be transparent enough. If a single model cannot offer both requirements even after some compromise, then the best way is to build separate models. Each model should fulfil a different requirement and the key to fulfilling such a requirement is to keep both of them consistent. The rapid prototyping method proposed by Delgado (1997) (refer to Section 4.4.1) represents a possible way and is based on the fuzzy clustering. Such a method is considered to be able to produce more accurate fuzzy models since the membership functions involved have more freedom rather than being restricted to a certain type, e.g. the Gaussian function. Since fuzzy models elicited via this method are very hard to interpret, it has not caught researchers' attentions in the field of multi-objective fuzzy modelling during the last decades. However, due to its easy implementation and relatively high accuracy, it deserves more exploration by incorporating an additional layer, e.g. knowledge translation layer, to make it more transparent.

(6) **Can multi-layered 'Stacked Generalisation' be used to further improve the model's predictive performance?** In Chapter 7, a single-layered 'Stacked Generalisation' was used to improve the generalisation ability of the elicited fuzzy rule-base, especially for the imprecise data. It is believed that, by using ensembles or cross-validation to create the error predictive models, one may further improve model's generalisation property. This would lead to a multi-layered (more than 2 layers) 'Stacked Generalisation'.

Bibliography

Alcalá R., Gacto M. J., Herrera F. (2007) "A Multi-Objective Genetic Algorithm for Tuning and Rule Selection to Obtain Accurate and Compact Linguistic Fuzzy Rule-Based Systems", *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, vol. 15 (5), pp. 539-557.

Back T. (1996) *Evolutionary Algorithms in Theory and Practice*, New York: Oxford University Press.

Bandyopadhyay S., Maulik U. (2002) "An Evolutionary Technique Based on K-Means Algorithm for Optimal Clustering in $\mathbb{R}^N$", *Information Sciences*, vol. 146, pp. 221-237.

Berkhin P. (2002) "Survey of Clustering Data Mining Techniques", *Technical Report*, Accrue Software, San Jose, CA.

Bezdek J. C. (1974) "Cluster Validity with Fuzzy Sets", *J. Cybernet.*, vol. 3(3), pp. 58-72.

Bezdek J. C. (1981) *Pattern Recognition with Fuzzy Objective Function Algorithms*, MA: Kluwer Academic Publishers Norwell.

Bezdek J. C., Hathaway R. J. (1994) "Optimisation of Fuzzy Clustering Criteria Using Genetic Algorithms", *The 1st IEEE Conference on Evolutionary Computation*, vol. 2(2), pp. 589-594.

Bodenhofer U., Herrera F. (1997) "The Lectures on Genetic Fuzzy Systems", *Technical Report SCCH-TR-0021*, Technical University, Bratislava, pp.1-69.

Burnet F. M. (1959) *The Clonal Selection Theory of Acquired Immunity*, UK: Cambridge at the University Press.

Casillas J., Cordon O., Del Jesus Mara J., Herrera F. (2001) "Genetic Tuning of Fuzzy Rule Deep Structures for Linguistic Modelling", *IEEE Transactions on Fuzzy Systems*, vol. 13, pp. 13-29.

Chen J., Mahfouf M. (2006) "A Population Adaptive Based Immune Algorithm for Solving Multi-objective Optimisation Problems", in *H. Bersini & J. Carneiro (Eds.): ICARIS 2006, LNCS 4163*, pp. 280-293.

Chen J., Mahfouf M. (2008a) "Artificial Immune Systems as a Bio-inspired Optimisation Technique and Its Engineering Applications", in *H. W. Mo (Eds.): Artificial Immune Systems and Natural Computing: Applying Complex Adaptive Technologies*, pp. 22-48.

Chen J., Mahfouf M. (2008b) "An Immune Algorithm Based Fuzzy Predictive Modeling Mechanism using Variable Length Coding and Multi-objective Optimisation Allied to Engineering Materials Processing", in *proceedings of the 2008 IEEE International Conference on Granule Computation (GrC2008),* pp.26-28.

Chen J., Mahfouf M. (2009) “An Artificial Immune Systems based Predictive Modelling Approach for the Multi-Objective Elicitation of Mamdani Fuzzy Rules”, in *Proc. Of the 2009 IEEE International Conference on Systems, Man, and Cybernetics (SMC2009).*

Chen M. Y., Linkens D. A. (2001) “A Systematic Neuro-Fuzzy Modeling Framework With Application to Material Property Prediction”, *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 31 (5), pp.781-790.

Chen M. Y., Linkens D. A. (2004) “Rule-base Self-generation and Simplification for Data-driven Fuzzy Models”, *Fuzzy Sets and Systems*, vol. 142, pp. 243-265.

Chen S., Cowan C. F. N., Grant P. M. (1991) “Orthogonal Least Squares Learning Algorithm for Radial Basis Function Networks”, *IEEE Transactions on Neural Networks*, vol. 2(2), pp. 302-309.

Chiu S. L. (1994) “Fuzzy Model Identification Based on Cluster Estimation”, *J. of Intelligent & Fuzzy Systems*, vol. 2 (3).

Chiu S. L. (1997) “An Efficient Method for Extracting Fuzzy Classification Rules from High Dimensional Data”, *Journal of Advanced Computational Intelligence*, vol. 1(1), pp. 31-36.

Cococcioni M., Ducange P., Lazzerini B., Marcelloni F. (2007) “A Pareto-based Multi-objective Evolutionary Approach to the Identification of Mamdani Fuzzy Systems”, *Soft Computing*, vol. 11, pp. 1013-1031.

Coello Coello C. A., Lamont Gary B., Van Veldhuizen David A. (2007) *Evolutionary Algorithms for Solving Multi-objective Problems*, London, New York: Kluwer Academic.

Coello Coello C. A., Cruz Cortes N. (2005) “Solving Multiobjective Optimisation Problems Using an Artificial Immune System”, *Genetic Programming and Evolvable Machines*, vol. 6(2), pp. 163-190.

Cordon O., Gomide F., Herrera F., Hoffmann F., Magdalena L. (2004) “Ten Years of Genetic Fuzzy Systems: Current Framework and New Trends”, *Fuzzy Sets and Systems*, vol. 141, pp. 5-31.

Cordon O., Herrera F., Hoffann F., Magdalena L. (2001) *Genetic Fuzzy Systems: Evolutionary Tuning and Learning of Fuzzy Knowledge Bases*, World Scientific, Singapore.

Cooper M. G., Vidal J. J. (1994) “Genetic Design of Fuzzy Controllers: The Cart and Jointed-Pole Problem”, *in Proceedings of the Third IEEE Conference on Fuzzy Systems*, vol. 2, pp. 1332-1337.

Cruz Cortes N., Coello Coello C. A. (2003) “Multi-objective Optimisation Using Ideas from the Clonal Selection Principle”, in *E. Cantu-Paz, et al. (Eds.)*: *Genetic and Evolutionary Computation (GECCO’2003), LNCS 2723*, pp. 158-170.

Cybenko G. (1989) “Approximations by Superpositions of A Sigmoidal Function”, *Mathematics of Signals and Systems*, vol. 2, pp. 303-314.

Deb K., Agrawal B. R. (1994) “Simulated Binary Crossover for Continuous Search Space”, *Technical Reports IITK/ME/SMD-94027*, Department of Mechanical Engineering, Indian Institute of Technology, Convenor.

Deb K. (2001) *Multi-Objective Optimisation using Evolutionary Algorithms*, Wiley, Chichester, U.K.

Deb K., Anand A., Joshi D. (2002) "A Computationally Efficient Evolutionary Algorithm for Real-Parameter Optimisation", *Evolutionary Computation*, vol. 10 (4), MIT Press, pp. 371-395.

Deb K., Thiele L., Laumanns M., Zitzler E. (2005) "Scalable Test Problems for Evolutionary Multiobjective Optimization", in *Abraham A. et al., (Eds.): Evolutionary Multiobjective Optimization Theoretical Advances and Applications*, pp. 105-145.

de Castro L. N., Von Zuben F. J. (2002) "Learning and Optimisation Using the Clonal Selection Principle", *IEEE Transactions on Evolutionary Computation*, vol. 6(3), pp. 239-251.

de Castro L. N., Timmis J. (2002) "An Artificial Immune Network for Multimodal Function Optimisation", *Proc. of the IEEE Congress on Evolutionary Computation (CEC' 2002)*, vol. 1, pp. 699-704.

Delgado M., Gómez-Skarmeta A. F., Vila A. (1996) "On the Use of Hierarchical Clustering in Fuzzy Modelling", *International Journal of Approximate Reasoning*, vol. 14(4), pp. 237-257.

Delgado M., Gómez-Skarmeta Antonio F., Martin F. (1997) "A Fuzzy Clustering-Based Rapid Prototyping for Fuzzy Rule-Based Modelling", *IEEE Transactions on Fuzzy Systems*, vol. 5(2), pp. 223-233.

Eberhart R. C., Kennedy J. (1995) "A New Optimiser Using Particle Swarm Theory", *The 6th International Symposium on Micro Machine and Human Science*, pp. 39-43.

Esponda F., Forrest S., Helman P. (2004) "A Formal Framework for Positive and Negative Detection Scheme", *IEEE Transaction on Systems, Man, and Cybernetics*, vol. 34(1), pp. 357-373.

Everitt B. S., Landau S., Leese M. (2001) *Cluster Analysis*, London: Arnold.

Farmer J. D., Packard N. H. (1986) "The Immune System, Adaptation, and Machine Learning", *Physica*, vol. 22D, pp. 187-204.

Fisher R. A. (1936) "The Use of Multiple Measurements in Taxonomic Problems", *Ann. Eugen.*, vol. 7(2), pp. 179-188.

Freschi F., Repetto M. (2005) "Multiobjective Optimisation by a Modified Artificial Immune System Algorithm", in *Christian Jacob et al. (Eds.): ICARIS 2004, LNCS 3627*, pp. 248-261.

Freschi F. (2006) *Multi-Objective Artificial Immune System for Optimisation in Electrical Engineering*, PhD Thesis, Politecnico di Torino, Department of Electrical Engineering, Torino, Italy.

Fukuda T., Mori K., Tsukiyama M. (1998) " Parallel Search for Multi-Modal Function Optimisation with Diversity and Learning of Immune Algorithm", *Artificial Immune Systems and Their Applications*, pp. 210-220.

Fukuyama Y., Sugeno M. (1989) "A New Method of Choosing the Number of Clusters for the Fuzzy c-mean Method", *5th Fuzzy Syst. Symp.*, pp. 247-250.

Genther H., Glesner M. (1994) "Automatic Generation of a Fuzzy Classification System Using Fuzzy Clustering Methods", in *Proc. ACM Symposium on Applied Computing (SAC'94)*, pp. 180-183.

Goldberg D. E. (1989) *Genetic Algorithms for Search, Optimisation, and Machine Learning*, MA: Addison-Wesley.

Gomez-Skarmeta A. F., Delgado M., Vila M. A. (1999) “About the Use of Fuzzy Clustering Techniques for Fuzzy Model”, *Fuzzy Sets and Systems*, vol. 106, pp. 179-188

Gong M. G., Jiao L. C., Du H. F., Bo L. F. (2006) “Multi-objective Immune Algorithm with Pareto-optimal Neighbor-based Selection”, *Technical Report IIIP-06-05*, Institute of Intelligent Information Processing, Xiandian University, China.

Gonzá́lez J., Rojas I., Pomares H., Ortega J., Prieto A. (2002) “A New Clustering Technique for Function Approximation”, *IEEE Transactions on Neural Networks*, vol. 13(1), pp. 132-142.

González J., Rojas I., Pomares H., Herrera L. J., Guillén A., Palomares J. M., Rojas F. (2007) “Improving the Accuracy While Preserving the Interpretability of Fuzzy Function Approximators by means of Multi-objective Evolutionary Algorithms”, *International Journal of Approximate Reasoning*, vol. 44(1), pp. 32-44.

Gower J. C., Ross G. J. S. (1969) “Minimum Spanning Rees and Single-linkage Cluster Analysis”, *Appl. Stat.*, vol. 18, pp. 54-64.

Guha S., Rastogi R., Shim K. (1998) “CURE: An Efficient Clustering Algorithm for Large Databases”, in *Proc. ACM SIGMOD Int. Conf. Management of Data*, pp. 73-84.

Guillaume S. (2001) “Designing Fuzzy Inference Systems from Data: An Interpretability-Oriented Review”, *IEEE Transactions on Fuzzy Systems*, vol. 9(3), pp. 426-443.

Hall L. O., Özyurt I. B., Bezdek J. C. (1999) “Clustering with a Genetically Optimised Approach”, *IEEE Transactions on Evolutionary Computation*, vol. 3(2), pp. 103-112.

Handl J., Knowles J. (2004) “Evolutionary Multi-objective Clustering”, *Parallel problem Solving from Nature-PPSN VIII, LNCS 3242*, pp. 1081-1091.

Hart E., Timmis J. (2005) “Application Areas of AIS: The Past, the Present and the Future”, in *C. Hacob et al. (Eds.): ICARIS 2005, LNCS 3527*, pp. 483-497.

Hartigan J. A. (1975) *Clustering Algorithms*, New York: Wiley.

Hartigan J. A., Wong M. A. (1979) “A k-means Clustering Algorithm”, *Journal of the Royal Statistical Society, Series C (Applied Statistics)*, vol. 28(1), pp. 100-108.

Hellendoorn H., Driankov D. (1997) *Fuzzy Model Identification, Selected Approaches*, Springer Berlin/Heidelberg.

Herrera F. (2008) “Genetic Fuzzy Systems: Taxonomy, Current Research Treads and Prospects”, *Evol. Intel.*, vol. 1 (1), pp. 27-46.

Holland J. H. (1975) *Adaptation in Natural and Artificial Systems*, MI: The University of Michigan Press.

Huang L. M., Ouyang C. S., Lee W. J., Lee S. J. (2002) “A Hybrid Clustering and SVD-Based Approach for Fuzzy-Neural System Modelling”, in *Proc. Of 7th Conference on Artificial Intelligence and Applications*, pp. 6-11.

Hutt B., Warwick K., (2007) “Synapsing Variable-Length Crossover: Meaningful Crossover for Variable-Length Genomes”, *IEEE Transactions on Evolutionary Computation*, vol. 11 (1), pp. 118-131.

Ishibuchi H., Nozaki K., Yamamoto N., Tanaka H. (1995) "Selecting Fuzzy If-Then Rules for Classification Problems Using Genetic Algorithms", *IEEE Transactions on Fuzzy Systems*, vol. 3 (3), pp.260-270.

Ishibuchi H., Murata T., Tüksen I. B. (1997) "Single-objective and Two-objective Genetic Algorithms for Selecting Linguistic rules for Pattern Classification Problems", *Fuzzy Sets and Systems*, vol. 89, pp. 135-150, 1997.

Ishibuchi H., Nakashima T., Murata T. (2001) "Three-objective Genetics-based Machine Learning for Linguistic Rule Extraction", *Information Sciences*, vol. 136, pp. 109-133.

Ishibuchi H., Yamamoto T. (2004) "Fuzzy Rule Selection by Multi-Objective Genetic Local Search Algorithms and Rule Evaluation Measures in Data Mining", *Fuzzy Sets and Systems*, vol. 141, pp. 59-88.

Ishibuchi H., Tsukamoto N., Nojima Y. (2008) "Evolutionary Many-Objective Optimisation: A Short Review", in *Proc. of 2008 IEEE Congress on Evolutionary Computation*, pp. 2424-2431.

Jain A. K., Murty M. N., Flynn P. J. (1999) "Data Clustering: A Review", ACM Computing Surveys, vol. 31(3), pp. 264-323.

Jang J.-S.R. (1993) "ANFIS: Adaptive-Network-Based Fuzzy Inference System", *IEEE Transactions on Systems, Man and Cybernetics*, vol. 23(3), pp. 665-685.

Jang J.-S.R., Sun CT. (1993) "Functional Equivalence between Radial Basis Function Networks and Fuzzy Inference Systems", *IEEE Transactions on Neural Networks*, vol. 4 (1), pp. 156-159.

Jerne N. K. (1974) "Towards a Network Theory of the Immune System", *Ann. Immunology (Inst. Pasteur)*, vol. 125C, pp. 373-389.

Jiao L. C., Gong M. G., Shang R. H. (2005) "Clonal Selection with Immune Dominance and Anergy Based Multiobjective Optimisation", in *C. A. Coello Coello, et al. (Eds.): Proc. of the Third International Conference on Evolutionary Multi-Criterion Optimisation (EMO'2005), LNCS 3410*, pp. 474-489, 2005.

Jiménez F., Gómez-Skarmeta A. F., Roubos H., Babuška R. (2001) "Accurate, Transparent, and Compact Fuzzy Models for Function Approximation and Dynamic Modeling through Multi-Objective Evolutionary Optimisation", in *E. Zitzler et al. (Eds.): EMO 2001, LNCS* 1993, pp. 653-667.

Jiménez F., Sánchez G., Gómez-Skarmeta A. F., Roubos H., Babuška R. (2002) "Fuzzy Modeling with Multi-Objective Neuro-Evolutionary Algorithms", *IEEE International Conference on Systems, Man, and Cybernetics*, vol. 3.

Jin Y. (2000) "Fuzzy Modeling of High-Dimensional Systems: Complexity Reduction and Interpretability Improvement", *IEEE Transactions on Fuzzy Systems*, vol. 8 (2), pp. 212-221.

Jin Y., Olhofer M., Sendhoff B. (2001) "Dynamic Weighted Aggregation for Evolutionary Multi-Objective Optimisation: Why Does It Work and How?", *Proc. GECCO 2001 Conf.*, pp. 1042-1049.

Jin Y., Von Seelen W., Sendhoff B. (1999) "On Generating FC^3 Fuzzy Rule Systems From Data Using Evolution Strategies", *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 29 (6), pp. 829-845.

Karr C. L. (1991) “Genetic Algorithms for Fuzzy Controllers”, *AI Expert*, vol. 6 (2), pp. 26-33.

Kelsey J., Timmis J. (2003) “Immune Inspired Somatic Contiguous Hyper-mutation for Function Optimisation”, in *E. Cantupaz, et al. (Eds.): Proc. of Genetic and Evolutionary Computation Conference (GECCO), LNCS 2723*, pp. 207-218.

Kennedy J., Eberhart R. (1995) “Particle Swarm Optimisation”, *IEEE International Conference on Neural Networks*, vol. 4, pp. 1942-1948.

Knowles J. D., Corne D. W. (2000) “Approximating the Nondominated Front Using the Pareto Archived Evolution Strategy”, *Evolutionary Computation*, vol. 8(2), pp. 149-172.

Kosko B. (1994) “Fuzzy Systems as Universal Approximators”, *IEEE Transactions on Computers*, vol. 43(11), pp. 1329-1333.

Koza J. R. (1999) *Genetic Programming III: Darwinian Invention and Problem Solving*, Morgan Kaufman, San Francisco, CA.

Krishna K., Narasimha Murty M. (1999) “Genetic K-Means Algorithm”, *IEEE Transactions on Systems, Man, and Cybernetics-Part B: Cybernetics*, vol. 29(3), pp. 433-439.

Krogh A., Vedelsby J. (1995) “Neural Network Ensembles, Cross Validation, and Active Learning”, in *G. Tesauro, D. Touretsky, T. Leen (Eds.): Advances in Neural Information Processing Systems*, vol. 7, Cambridge, Mass.: MIT Press.

Lin Y. H., Cunningham III G. A., Coggeshall S. V. (1997) “Using Fuzzy Partitions to Create Fuzzy Systems from Input-Output Data and Set the Initial Weights in a Fuzzy Neural Network”, *IEEE Transactions on Fuzzy Systems*, vol. 5 (4), pp. 614-621.

Magdalena L. (1998) ‘Crossing Unordered Sets of Rules in Evolutionary Fuzzy Controllers’, International Journal of Intelligent Systems, vol. 13 (10/11), pp. 993-1010.

Mamdani E. H. (1974) “Applications of Fuzzy Algorithm for Control a Simple Dynamic Plant”, *Proc. Inst. Electr. Eng.*, vol. 121 (12), pp. 1585-1588.

Mamdani E. H., Assilian S. (1975) “An Experiment in Linguistic Synthesis with a Fuzzy Logic Controller”, *International Journal Man-Machine Studies*, vol. 7, pp. 1-13.

Maulik U., Bandyopadhyay S. (2000) “Genetic Algorithm-based Clustering Technique”, *Pattern Recognition*, vol. 33(9), pp. 1455-1465.

McCulloch W. S., Pitts W. (1943) “A Logical Calculus of the Ideas Immanent in Nervous Activity”, *Bulletin of Mathematical Biology*, vol. 5(4), pp. 115-133.

Mencar C., Castellano G., Fanelli A. M. (2005) “Some Fundamental Interpretability Issues in Fuzzy Modelling”, *in Joint EUSFLAT-LFA 2005*, pp. 100-105.

Miller G. A. (1956) “The Magical Number Seven, Plus or Minus Two: Some Limits on Our Capacity for Processing Information”, *The Psychological Review*, vol. 63(2), pp. 81-97.

Murthy C. A., Chowdhury N. (1996) “In Search of Optimal Clusters Using Genetic Algorithms”, *Pattern Recognition Letters*, vol. 17, pp. 825-832.

Nelder J. A., Mead R. (1965) “A Simplex Method for Function Minimization”, *The Computer Journal*, vol. 7(4), pp. 308-313.

Panoutsos G., Mahfouf M. (2008) "Modelling Imprecise and Scattered Multidimensional Data Using Granular Data Compression and Multiple Granularity Modelling", *Proc. of the IEEE International Conference on Granular Computing (GrC2008)*, pp. 512-517.

Passino K. M., Yurkovich S. (1998) *Fuzzy Control*, MA: Addison-Wesley.

Perelson A. S. (1989) "Immune Network Theory", *Immunological Review*, vol. 110, pp. 5-36.

Pickering F. B. (1978) *Physical Metallurgy and Design of Steels*, Applied Science, Barking, U.K.

Roubos H., Setnes M. (2001) "Compact and Transparent Fuzzy Models and Classifiers Through Iterative Complexity Reduction", *IEEE Transactions on Fuzzy Systems*, vol. 9 (4), pp. 516-524.

Setnes M. (2000) "Supervised Fuzzy Clustering for Rule Extraction", *IEEE Transactions on Fuzzy Systems*, vol. 8(4), pp. 416-424.

Schaffer J. D., Grefenstette J. J. (1985) "Multi-objective Learning via Genetic Algorithms", *Proc. of the Ninth International Joint Conference on Artificial Intelligence*, pp. 593-595.

Schwefel H. P. (1981) *Numerical Optimisation of Computer Models*, Wiley, Chichester.

Setnes M., Babuška R., Kaymak U., Lemke H. (1998) "Similarity Measures in Fuzzy Rule Base Simplification", *IEEE Transactions on Systems, Man, and Cybernetics-Part B*, vol. 28 (3), pp. 376-386.

Setnes M., Roubos H. (2000) "GA-Fuzzy Modeling and Classification: Complexity and Performance", *IEEE Transactions on Fuzzy Systems*, vol. 8 (5), pp. 509-522.

Sheng W., Liu X. (2006) "A Genetic K-medoids Clustering Algorithm", *Journal of Heuristics*, vol. 12, pp. 447-466.

Sneath P. H. A. (1957) "The Application of Computers to Taxonomy", Journal of General Microbiology, vol. 17, pp. 201-226.

Sollich P., Krogh A. (1996) "Learning with Ensembles: How over-fitting can be useful", in *D. S Touretzky, M. C. Mozer, M. E. Hasselmo (Eds.): Advances in Neural Information Processing Systems*, vol. 8, MIT Press.

Sokal R. R., Michener C. D. (1958) "A Statistical Method for Evaluating Systematic Relationships", *University of Kansas Scientific Bulletin*, vol. 28, pp. 1409-1438.

Sorensen T. (1948) "A Method of Establishing Groups of Equal Amplitude in Plant Sociology Based on Similarity of Species Content and Its Application to Analyzes of the Vegetation on Danish Commons", *Biologiske Skrifter*, vol. 5, pp. 1-34.

Stone M. H. (1948) "The Generalized Weierstrass Approximation Theorem", *Mathematics Magazine*, vol. 21(5), pp. 237-254.

Sugeno M., Yasukawa T. (1993) "A Fuzzy-Logic-Based Approach to Qualitative Modeling", *IEEE Transactions on Fuzzy Systems*, vol. 1 (1), pp. 7-31.

Takagi T., Sugeno M. (1985) "Fuzzy Identification of Systems and Its Applications to Modelling and Control", *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 15, pp. 116-132.

Tenner J. (1999) *Optimisation of the Heat Treatment of Steel using Neural Networks*, Ph.D. Thesis, Department of Automatic Control and Systems Engineering, University of Sheffield, U.K.

Ting K. M., Witten I. H. (1999) "Issues in Stacked Generalization", *Journal of Artificial Intelligence Research (JAIR)*, vol. 10, pp. 271-289.

Wang H. L., Kwong S., Jin Y. C., Wei W., Man K. F. (2005) "Multi-objective Hierarchical Genetic Algorithm for Interpretable Fuzzy Rule-based Knowledge Extraction", *Fuzzy Sets and Systems*, Vol. 149 (1), pp. 149-186.

Wang L. X., Mendel J. M. (1992) "Generating Fuzzy Rules by Learning from Examples", *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 22(6), pp. 1414-1427.

Wang X., Gao X. Z., Ovaska S. J. (2004) "Artificial Immune Optimisation Methods and Applications-A Survey", *2004 IEEE International Conference on Systems, Man and Cybernetics*, vol. 4, pp. 3415-3420.

Wang X. L., Mahfouf M. (2005) "ACSAMO: An Adaptive Multiobjective Optimisation Algorithm using the Clonal Selection Principle", *1st European Symposium on Nature-inspired Smart Information Systems*, Albufeira, Portugal.

Ward J. J. JR. (1963) "Hierarchical Grouping to Optimise an Objective Function", *J. Am. Stat. Assoc.* vol. 58, pp. 236-244.

Wolpert D. H. (1992) "Stacked Generalization", *Neural Networks*, vol. 5(2), pp. 241-260.

Wong C. C., Chen CC. (1999) "A Hybrid Clustering and Gradient Descent Approach for Fuzzy Modelling", *IEEE Transactions on Systems, Man, and Cybernetics-Part B: Cybernetics*, vol. 29(6), pp. 686-693.

Xie X. L., Beni G. (1991) "A Validity Measure for Fuzzy Clustering", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 13(8), pp. 841-847.

Xu R., Wunsch II D. (2005) "Survey of Clustering Algorithms", IEEE Transactions on Neural Networks, vol. 16(3), pp. 645-678.

Yager R. R., Filev D. P. (1994) "Approximate Clustering Via the Mountain Method", *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 24(8), pp. 1279-1284.

Yoo J., Hajela P. (1999) "Immune Network Simulations in Multicriterion Design", *Structural Optimisation*, vol. 18, pp. 85-94.

Yoshinari Y., Pedrycz W., Hirota K. (1993) "Construction of Fuzzy Models Through Clustering Techniques", *Fuzzy Sets and Systems*, vol. 54, pp. 157-165.

Zadeh L. (1965) "Fuzzy Sets", *Information and control*, vol. 8(3), pp. 338-353.

Zadeh L. (1973) "Outline of A New Approach to the Analysis of Complex Systems and Decision Processes", *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 3, pp. 28-44.

Zhang Q., Mahfouf M. (2007) "Fuzzy Predictive Modeling Using Hierarchical Clustering and Multi-Objective Optimisation for Mechanical Properties of Alloy Steels", in *Proceedings of the 12th IFAC Symposium on Automation in Mining, Mineral and Metal Processing.*

Zhang Q. (2009) *Nature-Inspired Multi-Objective Optimisation and Transparent Knowledge Discovery via Hierarchical Fuzzy Modelling*, Ph.D. Thesis, Department of Automatic Control and Systems Engineering, The University of Sheffield, U.K.

Zitzler E., Laumanns M., Thiele L. (2001) “SPEA2: Improving the Strength Pareto Evolutionary Algorithm”, *TIK-Report 103*, Zurich: Swiss Federal Institute of Technology (ETH), Computer Engineering and Networks Laboratory (TIK).

Appendix A

The Back-Error-Propagation Algorithm for Mamdani FRBS

According to Eqs. 5.5 and 5.6, a defuzzified Mamdani FRBS can be described as follows:

$$y^{crisp}(X|\theta) = \frac{\sum_{i=1}^{k} b_i \cdot \mu_i(X) \cdot g(b_i, \sigma_i^y)}{\sum_{i=1}^{k} \mu_i(X) \cdot g(b_i, \sigma_i^y)} \tag{A.1}$$

There are four parameters to update, namely $\theta = (b_i, \sigma_i^y, c_i^j, \sigma_i^j)$. Before the detailed deductions of each update laws, following denotation is adopted:

$$e_m = \frac{1}{2} \cdot [y^{crisp}(X_m|\theta) - y_m]^2 \tag{A.2}$$

Where X_m is the input vector of the mth data sample; y_m is the actual output of the mth data sample. Using the chain rule, the general gradient based update law for the parameters has the form shown in Eq. A.3. The update formula for each parameter is obtained by replacing $\theta_{(\cdot)}$ with the corresponding parameter.

$$\theta_{(\cdot)}(t+1) = \theta_{(\cdot)}(t) - \lambda \cdot \frac{\partial e_m}{\partial \theta_{(\cdot)}}|_t \tag{A.3}$$

1. Centre of the Consequents Updating Law

$$b_i(t+1) = b_i(t) - \lambda_1 \cdot \frac{\partial e_m}{\partial b_i}|_t \tag{A.4}$$

$$\text{Here, } \frac{\partial e_m}{\partial b_i}|_t = (y^{crisp}(X_m|\theta) - y_m) \cdot \frac{\partial y^{crisp}(X_m|\theta)}{\partial b_i}|_t \tag{A.5}$$

$$\text{Let: } \varepsilon_m \triangleq y^{crisp}(X_m|\theta) - y_m \tag{A.6}$$

$$since,\quad \frac{\partial y^{crisp}(X_m|\theta)}{\partial b_i}|_t$$

$$= \frac{\left[b_i \cdot \mu_i(X_m) \cdot g(b_i, \sigma_i^y)\right]'_{b_i} \cdot \sum_{i=1}^{k} \mu_i(X_m) \cdot g(b_i, \sigma_i^y) - \left[\mu_i(X_m) \cdot g(b_i, \sigma_i^y)\right]'_{b_i} \cdot \sum_{i=1}^{k} b_i \cdot \mu_i(X_m) \cdot g(b_i, \sigma_i^y)}{\left[\sum_{i=1}^{k} \mu_i(X_m) \cdot g(b_i, \sigma_i^y)\right]^2}|_t$$

$$and,\quad g'\left(b_i, \sigma_i^y\right)|_{b_i} \triangleq g'(b_i) = \frac{1}{1+\left(\frac{y_L - b_i}{\sigma_i^y}\right)^2} - \frac{1}{1+\left(\frac{y_U - b_i}{\sigma_i^y}\right)^2} \tag{A.7}$$

$$\therefore \frac{\partial y^{crisp}(X_m|\theta)}{\partial b_i}|_t$$

$$= \frac{\left[\mu_i(X_m) \cdot g\left(b_i, \sigma_i^y\right) + b_i \cdot \mu_i(X_m) \cdot g'(b_i)\right] \cdot \sum_{i=1}^{k} \mu_i(X_m) \cdot g\left(b_i, \sigma_i^y\right) - \mu_i(X_m) \cdot g'(b_i) \cdot \sum_{i=1}^{k} b_i \cdot \mu_i(X_m) \cdot g\left(b_i, \sigma_i^y\right)}{\left[\sum_{i=1}^{k} \mu_i(X_m) \cdot g\left(b_i, \sigma_i^y\right)\right]^2}|_t$$

$$\Rightarrow \frac{\partial y^{crisp}(X_m|\theta)}{\partial b_i}|_t = \frac{\mu_i(X_m) \cdot g\left(b_i, \sigma_i^y\right) + b_i \cdot \mu_i(X_m) \cdot g'(b_i) - \mu_i(X_m) \cdot g'(b_i) \cdot y^{crisp}(X_m|\theta)}{\sum_{i=1}^{k} \mu_i(X_m) \cdot g\left(b_i, \sigma_i^y\right)}|_t \tag{A.8}$$

Substituting Eqs. A.5 and A.8 into A.4 gives the update law for the output centres:

$$b_i(t+1) = b_i(t) - \lambda_1 \cdot \varepsilon_m(t) \cdot \frac{\mu_{i(t)}(X_m) \cdot \left[g\left(b_{i(t)}, \sigma_{i(t)}^y\right) + b_{i(t)} \cdot g'\left(b_{i(t)}\right) - g'\left(b_{i(t)}\right) \cdot y^{crisp}\left(X_m|\theta(t)\right)\right]}{\sum_{i=1}^{k} \mu_{i(t)}(X_m) \cdot g\left(b_{i(t)}, \sigma_{i(t)}^y\right)} + \beta_1 \cdot \Delta b_i(t-1) \tag{A.9}$$

Where, $\beta_1 \cdot \Delta b_i(t-1)$ is the momentum term which can sometimes improve the speed of convergence (Passino, 1997, p. 246-252).

2. Spread of the Consequents Updating Law

$$\sigma_i^y(t+1) = \sigma_i^y(t) - \lambda_2 \cdot \frac{\partial e_m}{\partial \sigma_i^y}|_t \tag{A.10}$$

$$\text{Here, } \frac{\partial e_m}{\partial \sigma_i^y}|_t = \varepsilon_m \cdot \frac{\partial y^{crisp}(X_m|\theta)}{\partial \sigma_i^y}|_t \tag{A.11}$$

$$since,\quad \frac{\partial y^{crisp}(X_m|\theta)}{\partial \sigma_i^y}|_t$$

$$= \frac{\left[b_i \cdot \mu_i(X_m) \cdot g(b_i, \sigma_i^y)\right]'_{\sigma_i^y} \cdot \sum_{i=1}^{k} \mu_i(X_m) \cdot g(b_i, \sigma_i^y) - \left[\mu_i(X_m) \cdot g(b_i, \sigma_i^y)\right]'_{\sigma_i^y} \cdot \sum_{i=1}^{k} b_i \cdot \mu_i(X_m) \cdot g(b_i, \sigma_i^y)}{\left[\sum_{i=1}^{k} \mu_i(X_m) \cdot g(b_i, \sigma_i^y)\right]^2}|_t$$

$$and,\ \ g'\left(b_i,\sigma_i^y\right)|_{\sigma_i^y} \triangleq g'\left(\sigma_i^y\right)$$

$$= \left[arctan\left(\frac{y_U - b_i}{\sigma_i^y}\right) - arctan\left(\frac{y_L - b_i}{\sigma_i^y}\right)\right] + \sigma_i^y$$

$$\cdot \left[\frac{1}{1+\left(\frac{y_U - b_i}{\sigma_i^y}\right)^2} \cdot \left(-\frac{y_U - b_i}{\left(\sigma_i^y\right)^2}\right) - \frac{1}{1+\left(\frac{y_L - b_i}{\sigma_i^y}\right)^2} \cdot \left(-\frac{y_L - b_i}{\left(\sigma_i^y\right)^2}\right)\right]$$

$$\Rightarrow g'\left(\sigma_i^y\right) = \frac{1}{\sigma_i^y} \cdot \left[g\left(b_i,\sigma_i^y\right) + \frac{y_L - b_i}{1+\left(\frac{y_L - b_i}{\sigma_i^y}\right)^2} - \frac{y_U - b_i}{1+\left(\frac{y_U - b_i}{\sigma_i^y}\right)^2}\right] \tag{A.12}$$

$$\therefore\ \frac{\partial y^{crisp}(X_m|\theta)}{\partial \sigma_i^y}|_t$$

$$= \frac{b_i \cdot \mu_i(X_m) \cdot g'\left(\sigma_i^y\right) \cdot \sum_{i=1}^{k} \mu_i(X_m) \cdot g\left(b_i,\sigma_i^y\right) - \mu_i(X_m) \cdot g'\left(\sigma_i^y\right) \cdot \sum_{i=1}^{k} b_i \cdot \mu_i(X_m) \cdot g\left(b_i,\sigma_i^y\right)}{\left[\sum_{i=1}^{k} \mu_i(X_m) \cdot g\left(b_i,\sigma_i^y\right)\right]^2}|_t$$

$$\Rightarrow \frac{\partial y^{crisp}(X_m|\theta)}{\partial \sigma_i^y}|_t = \frac{b_i \cdot \mu_i(X_m) \cdot g'\left(\sigma_i^y\right) - \mu_i(X_m) \cdot g'\left(\sigma_i^y\right) \cdot y^{crisp}(X_m|\theta)}{\sum_{i=1}^{k} \mu_i(X_m) \cdot g\left(b_i,\sigma_i^y\right)}|_t \tag{A.13}$$

Substituting A.11, A.12 and A.13 into A.10 gives the update law for the output spread:

$$\sigma_i^y(t+1) = \sigma_i^y(t) - \lambda_2 \cdot \varepsilon_m(t) \cdot \frac{\mu_{i(t)}(X_m) \cdot g'\left(\sigma_{i(t)}^y\right) \cdot \left[b_{i(t)} - y^{crisp}(X_m|\theta(t))\right]}{\sum_{i=1}^{k} \mu_{i(t)}(X_m) \cdot g\left(b_{i(t)},\sigma_{i(t)}^y\right)} + \beta_2 \cdot \Delta\sigma_i^y(t-1) \tag{A.14}$$

3. Centre of the Premise Updating Law

$$c_i^j(t+1) = c_i^j(t) - \lambda_3 \cdot \frac{\partial e_m}{\partial c_i^j}|_t \tag{A.15}$$

$$\text{Here,}\quad \frac{\partial e_m}{\partial c_i^j}|_t = \varepsilon_m \cdot \frac{\partial y^{crisp}(X_m|\theta)}{\partial \mu_i(X_m)} \cdot \frac{\partial \mu_i(X_m)}{\partial c_i^j}|_t \tag{A.16}$$

$$since,\ \ \frac{\partial y^{crisp}(X_m|\theta)}{\partial \mu_i(X_m)}|_t$$

$$= \frac{\left[b_i \cdot \mu_i(X_m) \cdot g\left(b_i,\sigma_i^y\right)\right]'_{\mu_i} \cdot \sum_{i=1}^{k} \mu_i(X_m) \cdot g\left(b_i,\sigma_i^y\right) - \left[\mu_i(X_m) \cdot g\left(b_i,\sigma_i^y\right)\right]'_{\mu_i} \cdot \sum_{i=1}^{k} b_i \cdot \mu_i(X_m) \cdot g\left(b_i,\sigma_i^y\right)}{\left[\sum_{i=1}^{k} \mu_i(X_m) \cdot g\left(b_i,\sigma_i^y\right)\right]^2}|_t$$

$$\Rightarrow \frac{\partial y^{crisp}(X_m|\theta)}{\partial \mu_i(X_m)}|_t = \frac{g\left(b_i,\sigma_i^y\right) \cdot \left(b_i - y^{crisp}(X_m|\theta)\right)}{\sum_{i=1}^{k} \mu_i(X_m) \cdot g\left(b_i,\sigma_i^y\right)}|_t \tag{A.17}$$

$$also, \quad \frac{\partial \mu_i(X_m)}{\partial c_i^j}|_t = \mu_i(X_m) \cdot \left(\frac{x_m^j - c_i^j}{\left(\sigma_i^j\right)^2} \right)|_t \tag{A.18}$$

Substituting A.16, A.17 and A.18 into A.15 gives the following update law:

$$c_i^j(t+1) = c_i^j(t) - \lambda_3 \cdot \varepsilon_m(t) \cdot \frac{g\left(b_{i(t)}, \sigma_{i(t)}^y\right) \cdot [b_{i(t)} - y^{crisp}(X_m|\theta(t))]}{\sum_{i=1}^k \mu_{i(t)}(X_m) \cdot g(b_{i(t)}, \sigma_{i(t)}^y)} \cdot \mu_{i(t)}(X_m) \cdot \left[\frac{x_m^j - c_{i(t)}^j}{(\sigma_{i(t)}^j)^2} \right] + \beta_3 \cdot$$

$$\Delta c_i^j(t-1) \tag{A.19}$$

4. Spread of the Premise Updating Law

$$\sigma_i^j(t+1) = \sigma_i^j(t) - \lambda_4 \cdot \frac{\partial e_m}{\partial \sigma_i^j}|_t \tag{A.20}$$

$$\text{Here,} \quad \frac{\partial e_m}{\partial \sigma_i^j}|_t = \varepsilon_m \cdot \frac{\partial y^{crisp}(X_m|\theta)}{\partial \mu_i(X_m)} \cdot \frac{\partial \mu_i(X_m)}{\partial \sigma_i^j}|_t \tag{A.21}$$

$$also, \quad \frac{\partial \mu_i(X_m)}{\partial \sigma_i^j}|_t = \mu_i(X_m) \cdot \left(\frac{\left(x_m^j - c_i^j\right)^2}{\left(\sigma_i^j\right)^3} \right)|_t \tag{A.22}$$

Substituting A.17, A.21 and A.22 into A.20 gives the following update law:

$$\sigma_i^j(t+1) = \sigma_i^j(t) - \lambda_4 \cdot \varepsilon_m(t) \cdot \frac{g\left(b_{i(t)}, \sigma_{i(t)}^y\right) \cdot [b_{i(t)} - y^{crisp}(X_m|\theta(t))]}{\sum_{i=1}^k \mu_{i(t)}(X_m) \cdot g(b_{i(t)}, \sigma_{i(t)}^y)} \cdot \mu_{i(t)}(X_m) \cdot \left[\frac{(x_m^j - c_{i(t)}^j)^2}{(\sigma_{i(t)}^j)^3} \right] + \beta_4 \cdot$$

$$\Delta \sigma_i^j(t-1) \tag{A.23}$$

Appendix B

Modelling Results of Impact Energy Data Using IMOFM_S

Figure B.1 shows the predictive performances of the three stages for the impact energy training and testing data sets using IMOFM_S.

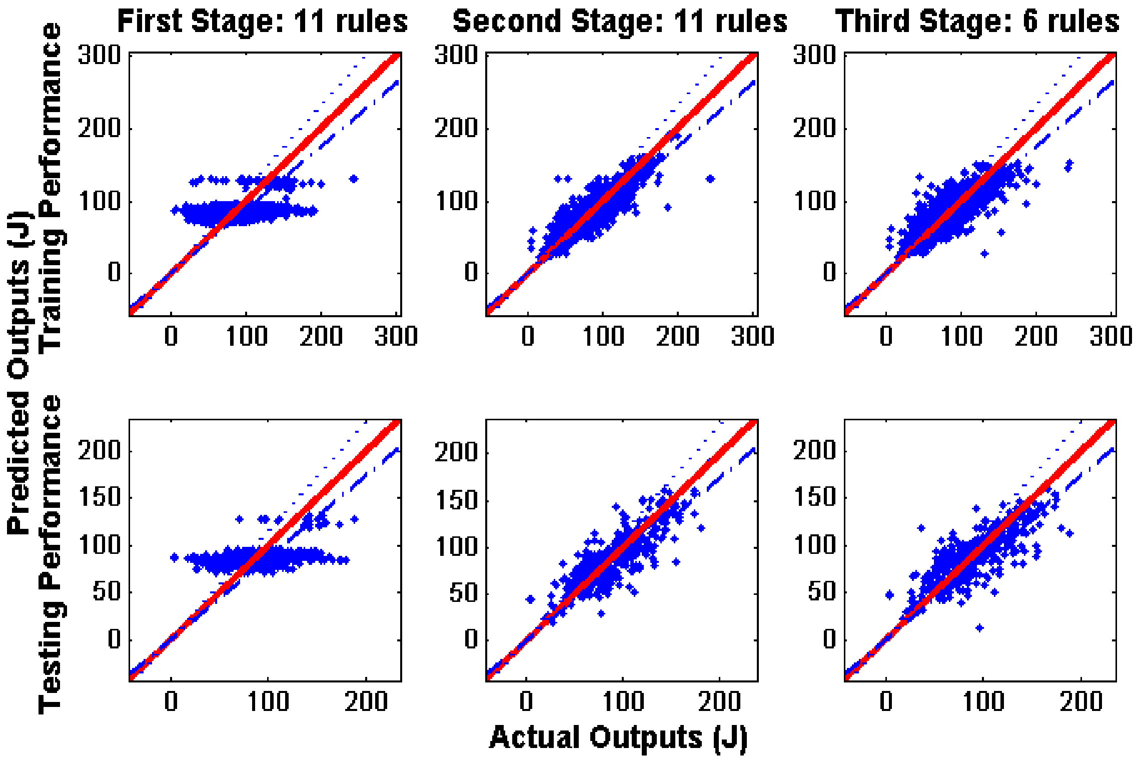


Figure B.1 The prediction performances of the three stages for the impact energy training and testing data using IMOFM_S.

Figure B.2 show the Pareto fronts of the impact energy modelling problem.

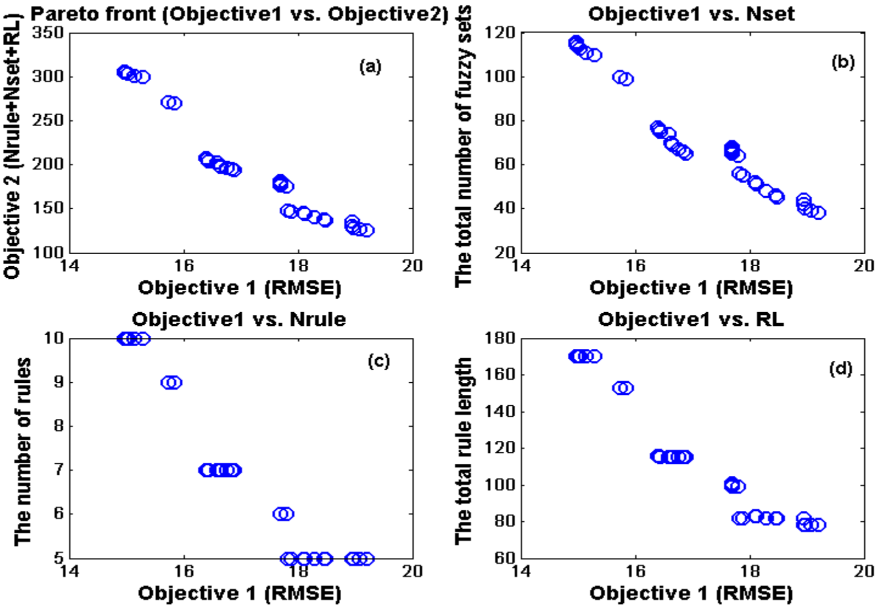


Figure B.2 The Pareto fronts obtained using IMOFM_S from the third modelling procedure for the impact energy modelling problem: (a) Objective1 vs. Objective2; (b) Objective1 vs. Nset; (c) Objective1 vs. Nrule; (d) Objective1 vs. RL.

Figures B.3~B.5 show the distribution of some membership functions of the fuzzy models elicited in different modelling stages.

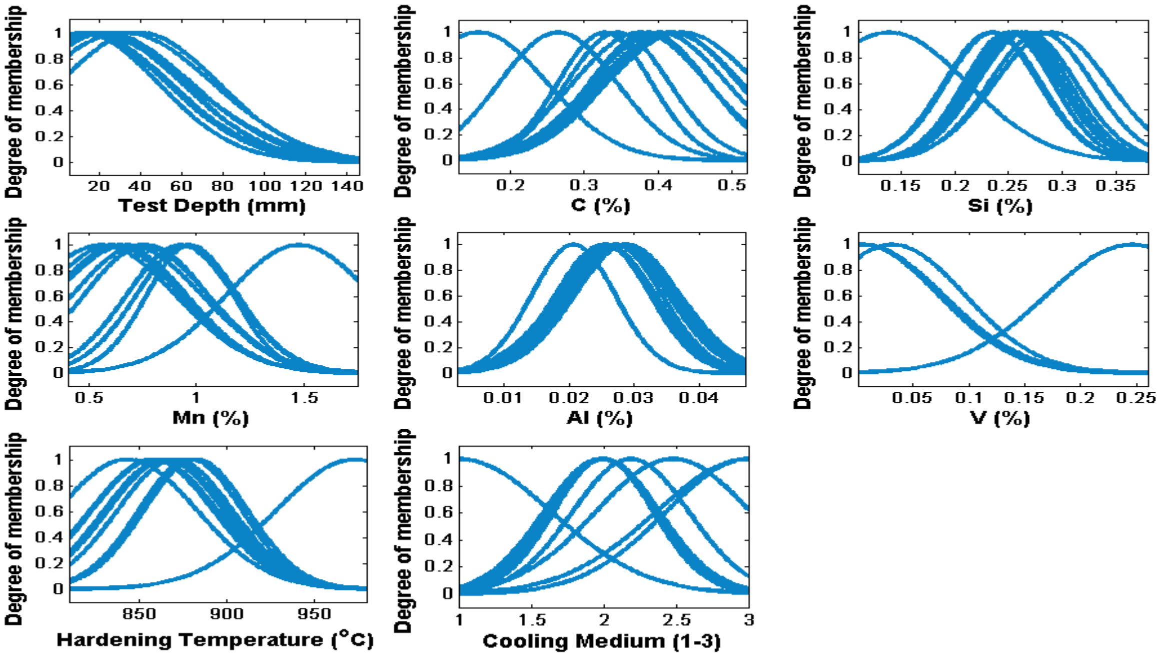


Figure B.3 The distribution of some membership functions of the 11-rule initial Singleton FRBS (from the first modelling stage) for impact energy modelling.

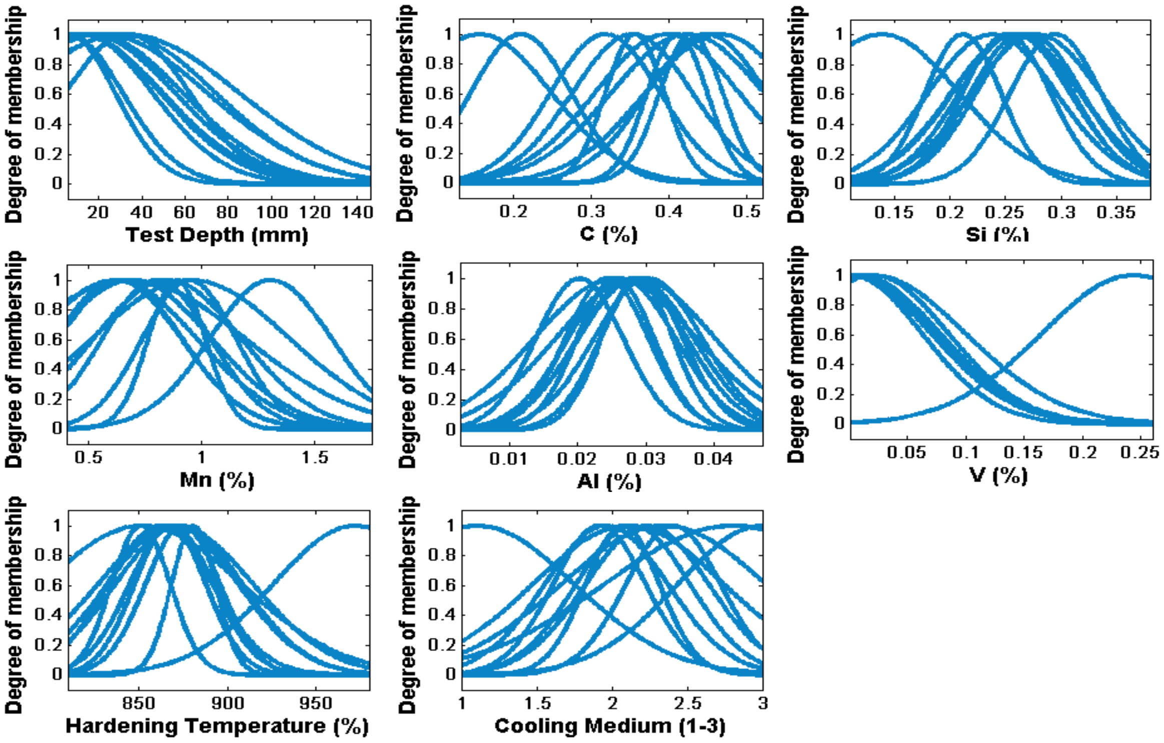


Figure B.4 The distribution of some membership functions of the 11-rule refined Singleton FRBS (from the second modelling stage) for impact energy modelling.

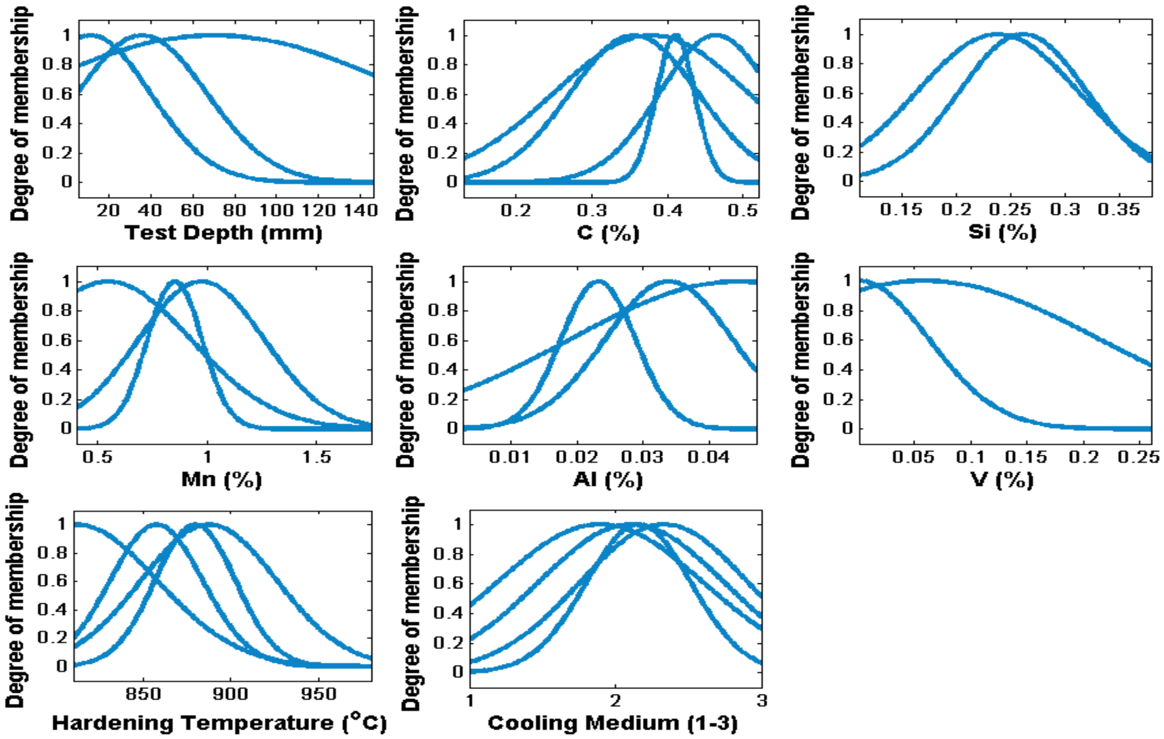


Figure B.5 The distribution of some membership functions of the 6-rule simplified Singleton FRBS (from the third modelling stage) for impact energy modelling.