



The  
University  
Of  
Sheffield.

### Access to Electronic Thesis

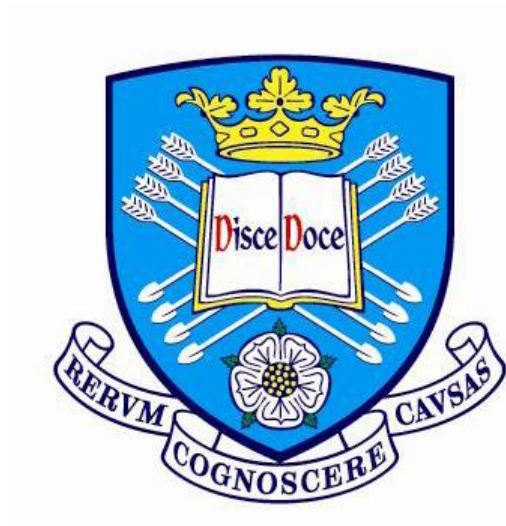
Author: Qian Zhang  
Thesis title: Nature-Inspired Multi-Objective Optimisation and Transparent Knowledge Discovery via Hierarchical Fuzzy Modelling  
Qualification: PhD  
Date awarded: 19 January 2009

**This electronic thesis is protected by the Copyright, Designs and Patents Act 1988. No reproduction is permitted without consent of the author. It is also protected by the Creative Commons Licence allowing Attributions-Non-commercial-No derivatives.**

# **Nature-Inspired Multi-Objective Optimisation and Transparent Knowledge Discovery via Hierarchical Fuzzy Modelling**

**Qian Zhang**

Department of Automatic Control and Systems Engineering  
The University of Sheffield



A thesis submitted in partial fulfilment of the requirements  
for the degree of Doctor of Philosophy

October 2008

# Abstract

Knowledge discovery is one of the most important human activities, which helps people recognise and understand some of the intricacies associated with the ancient and modern worlds. With the rapid development in the human capabilities to both generate and collect data, the discovery of knowledge from data has become a practical and popular research topic. In this thesis, knowledge discovery from data is conducted from the following two overarching viewpoints: first, developing prediction models using the data that represent input-output relationships; second, based on these developed prediction models, finding the optimal designs (solutions) from a set of predefined objectives. The theoretical aspects behind the previous two research facets are described and the associated experimental studies are carried out.

A particular focus of this thesis is on a cooperative fuzzy modelling framework, which integrates transparent (interpretable) fuzzy systems with robust evolutionary computing based algorithms involving several techniques, such as data clustering, data mining, and multi-objective optimisation. Evolutionary optimisation algorithms are also developed on the basis of nature and social inspired ideas. Optimisation forms an essential part of the modelling framework and is employed in the direct optimal design problems as well. The proposed cooperative fuzzy modelling methodology and the devised evolutionary optimisation algorithms are then applied to knowledge discovery in terms of systems modelling and control (static optimisation via reverse-engineering), using simulation platforms as well as real industrial data.

The experimental results show that the proposed modelling framework and optimisation algorithms outperform some of the other salient techniques; the proposed approaches can successfully work within the context of the high-dimensional industrial applications, including modelling and optimal design problems.

# Acknowledgements

I would like to thank everyone who has helped me over the last few years. Without their support, I would certainly not have completed this thesis.

First of all, I would like to express my sincere and great thanks to my excellent supervisor, Professor Mahdi Mahfouf, for his expert guidance, continuous support and constructive advice throughout the whole period of this PhD project.

I would also like to thank all my friends and colleagues in the Intelligent Systems Laboratory for their kind help and friendship.

Last but not least, I wish to thank my family, especially my parents, my sister and my grandparents for their support, encouragement and love throughout my PhD study. My family has always taken care of me and I love them all very much.

## Related Publications

- Zhang, Q., Mahfouf, M. (2008) “Mamdani-Type Fuzzy Modelling via Hierarchical Clustering and Multi-Objective Particle Swarm Optimisation (FM-HCPSO)” *International Journal of Computational Intelligence Research (IJCIR)*, Accepted
- Zhang, Q., Mahfouf, M. (2007) “Fuzzy predictive modelling using hierarchical clustering and multi-objective optimisation for mechanical properties of alloy steels” *Proceedings of the 12th IFAC Symposium on Automation in Mining, Mineral and Metal Processing, IFAC MMM’07*, pp. 427-432, 21-23 August 2007, Quebec City, Canada
- Zhang, Q., Mahfouf, M. (2007) “A new Reduced Space Searching Algorithm (RSSA) and its application in optimal design of alloy steels” *Proceedings of the 2007 IEEE Congress on Evolutionary Computation, CEC 2007*, pp. 1815-1822, 25-28 September 2007, Singapore
- Zhang, Q., Mahfouf, M. (2006) “A New Structure for Particle Swarm Optimization (nPSO) Applicable to Single Objective and Multiobjective Problems” *Proceedings of the 3rd International IEEE Conference on Intelligent Systems, IEEE IS’06*, pp. 176-181, 4-6 September 2006, London, UK
- Mahfouf, M., Chen, M.-Y., Zhang, Q., Linkens, D. A. (2006) “Adaptive Weighted Particle Swarm Multiobjective Optimisation and Societal Reasoning for the Design of Alloy” *Proceedings of the 1st IFAC Workshop on Applications of Large Scale Industrial Systems, IFAC ALSIS’06*, 30-31 August 2006, Helsinki, Finland
- Zhang, Q., Mahfouf, M. (2005) “Reduced Space Searching (RSS) Algorithm: A New Optimization Algorithm for Single-Objective and Multi-Objective Problems” *Proceedings of the 1st European Symposium on Nature-inspired Smart Information Systems, NiSIS’05*, 3-5 October 2005, Albufeira, Portugal

# Contents

## Chapter 1: Introduction

1.1 Background and Research Aims.....	1
1.2 Structure of the Thesis.....	7

## Chapter 2: Evolutionary Based Optimisation and Fuzzy Theories - A

### Background

2.1 Optimisation.....	11
2.1.1 Single Objective Optimisation.....	11
2.1.2 Multi-Objective Optimisation.....	12
2.2 Evolutionary Computation.....	14
2.2.1 Evolutionary Algorithms.....	15
2.2.1.1 Genetic Algorithms.....	16
2.2.1.2 Evolution Strategies.....	17
2.2.2 Swarm Intelligence.....	17
2.2.2.1 Particle Swarm Optimisation (PSO).....	18
2.2.3 Multi-Objective Optimisation Algorithms within Evolutionary Computation.....	19
2.2.3.1 Strength Pareto Evolutionary Algorithm.....	20
2.2.3.2 Pareto Archived Evolution Strategy.....	21
2.2.3.3 Nondominated Sorting Genetic Algorithm II.....	21
2.3 Fuzzy Sets and Fuzzy Systems.....	22
2.3.1 Fuzzy Sets.....	22

## Contents

2.3.2 Fuzzy Systems.....	24
2.4 Summary.....	26
<b>Chapter 3: A New Reduced Space Searching Algorithm with Applications to Single Objective and Multi-Objective Problems</b>	
3.1 Introduction to the Reduced Space Searching (RSS) Strategy.....	27
3.2 The Reduced Space Searching Algorithm (RSSA).....	30
3.2.1 The Basic Idea.....	30
3.2.2 Improvement of the Basic Idea.....	30
3.2.3 An Example of the Search Space Decreasing or Increasing.....	31
3.2.4 The Variation Operators.....	34
3.2.5 The RSSA Algorithm.....	34
3.3 Experimental Studies using RSSA.....	36
3.3.1 Benchmark Test Functions.....	36
3.3.2 Effect of the Decreasing Parameter ‘ $C_1$ ’.....	42
3.3.3 Effects of the Variation Strategies.....	44
3.3.4 A Comparison between RSSA and Other Evolutionary Algorithms.....	44
3.4 Extension of RSSA to Multi-Objective Optimisation Problems.....	49
3.4.1 The Random Weighted Aggregation Approach.....	49
3.4.2 Archive Design.....	50
3.4.3 The Multi-Objective Reduced Space Searching Algorithm (MO-RSSA).....	52
3.5 Experimental Studies using MO-RSSA.....	53
3.5.1 Comparison between MO-RSSA and Other Multi-Objective EAs.....	53
3.5.1.1 Problems Description.....	54
3.5.1.2 Performance Metrics.....	55
3.5.1.3 Experimental Setup.....	56
3.5.1.4 Experimental Results.....	57
3.5.2 Experiments Based on the DTLZ Series Problems.....	59
3.6 Summary.....	63





## Contents

5.3.1 Data Clustering.....	120
5.3.1.1 Basic Concepts.....	120
5.3.1.2 Types of Data Clustering Algorithms.....	122
5.3.1.3 Hierarchical Clustering Algorithms.....	123
5.3.1.4 The Proposed New Agglomerative Complete-Link Clustering Algorithm.....	127
5.3.2 Initial Fuzzy Model Construction.....	129
5.3.2.1 An Example of Constructing the Initial Fuzzy Model.....	129
5.3.2.2 Fuzzy System Definition and Notation.....	131
5.3.2.3 The Fuzzy Model Extraction Approach.....	131
5.3.3 Experimental Studies.....	133
5.3.3.1 Comparison of Clustering Algorithms for Fuzzy Modelling Problems.....	133
5.3.3.2 Effects of Various Weights of Distance Measure.....	138
5.4 Accuracy Optimisation and Data Selection.....	141
5.4.1 Accuracy Optimisation.....	141
5.4.1.1 The Validation Mechanism.....	142
5.4.2 Data Selection.....	144
5.4.2.1 Rough Data Selection.....	145
5.4.2.2 Missing Representative Data Selection.....	148
5.4.3 The Joint Mechanism of Accuracy Optimisation and Data Selection .....	151
5.5 Simultaneous Multi-Objective Optimisation of Accuracy and Interpretability.....	154
5.5.1 Interpretability Improvement.....	154
5.5.1.1 Removing Redundant Fuzzy Rules.....	154
5.5.1.2 Merging Similar Fuzzy Rules.....	157
5.5.1.3 Removing Redundant Fuzzy Sets.....	158
5.5.1.4 Merging Similar Fuzzy Sets.....	160
5.5.2 Experimental Studies.....	161
5.5.2.1 An Example of Using the Interpretability Improvement Approach.....	161

## Contents

5.5.2.2 Effects of the Thresholds of the Interpretability Improvement Approach.....	164
5.5.3 The Multi-Objective Optimisation Mechanism.....	169
5.5.4 The Designed Objective Functions for Fuzzy Modelling.....	172
5.6 Confidence Band Analysis.....	172
5.7 Summary.....	175
<b>Chapter 6: Application of FM-HCMO to Fuzzy Modelling Problems</b>	
6.1 Introduction.....	177
6.2 The Nonlinear Function Approximation.....	177
6.3 The Identification of a Dynamic System.....	186
6.4 Mechanical Property Prediction of Alloy Steels.....	196
6.4.1 Ultimate Tensile Strength (UTS).....	197
6.4.2 Reduction of Area (ROA).....	206
6.4.3 Elongation.....	212
6.4.4 Impact Energy.....	218
6.5 Summary.....	224
<b>Chapter 7: Application of RSSA and MO-RSSA for Alloy Steel Design Problems</b>	
7.1 Introduction.....	226
7.2 The Optimal Design of UTS for Alloy Steels.....	228
7.3 The Optimal Design of ROA.....	232
7.4 The Optimal Design of both UTS and ROA.....	236
7.4.1 Boundaries for the UTS and ROA Design.....	236
7.4.2 The Single Objective Optimisation.....	238
7.4.3 The Multi-objective Optimisation.....	239
7.5 The Optimal Alloy Design Considering both the Mechanical Properties and the Economical Factors.....	241
7.5.1 The Optimal Design Considering both UTS and the Cost.....	242
7.5.2 The Optimal Design Considering both ROA and the Cost.....	244
7.5.3 The Optimal Design Considering UTS, ROA and the Cost.....	246

## **Contents**

7.6 Summary.....	248
<b>Chapter 8: Conclusion and Future work</b>	
8.1 Summary of Main Results.....	249
8.2 Future work.....	252
8.3 Conclusion.....	256
<b>Bibliography.....</b>	<b>257</b>

## List of Figures

Figure		Page
2-1	An example of membership functions.....	23
2-2	The basic configuration of fuzzy systems.....	24
3-1	The RSS strategy for dealing with optimisation problems.....	29
3-2	An example of how to divide the search space in the case a two-dimensional problem.....	32
3-3	Flow chart of the RSSA algorithm.....	33
3-4	The 3-dimensional map for $f_1$ to $f_{14}$ in the 2-dimensional situation: (a) to (n).....	41
3-5	Pseudo-code for maintaining the archive.....	51
3-6	An example of non-dominated selection and diversity selection: (a) The solutions before the selections; (b) Non-dominated selection; (c) Cell selection; (d) The solutions after the selections.....	53
3-7	Pareto fronts obtained by MO-RSSA.....	59
3-8	The optimisation result of MO-RSSA on DTLZ1.....	60
3-9	The optimisation result of MO-RSSA on DTLZ2.....	61
3-10	The optimisation result of MO-RSSA on DTLZ3.....	61
3-11	The optimisation result of MO-RSSA on DTLZ5.....	62
3-12	The optimisation result of MO-RSSA on DTLZ6.....	62
3-13	The optimisation result of MO-RSSA on DTLZ7.....	63
4-1	An example of the particles' movements in two continuous iterations in PSO.....	67

## **List of Figures**

4-2	The mean number of function evaluations to find the optimum versus the population size with different problem dimension: (a) $f_1$ , (b) $f_2$ , (c) $f_3$ , (d) $f_7$ , (e) $f_8$ and (f) $f_{10}$ .....	85
4-3	Pareto fronts obtained by nMPSO.....	98
4-4	The optimisation result of nMPSO on DTLZ1.....	99
4-5	The optimisation result of nMPSO on DTLZ2.....	99
4-6	The optimisation result of nMPSO on DTLZ3.....	100
4-7	The optimisation result of nMPSO on DTLZ5.....	101
4-8	The optimisation result of nMPSO on DTLZ6.....	101
4-9	The optimisation result of nMPSO on DTLZ7.....	102
5-1	Modelling within the context of a Cartesian representation between accuracy and interpretability - the accuracy maximisation.....	113
5-2	Modelling within the context of a Cartesian representation between accuracy and interpretability - the optimisation of the weighted sum of objective functions.....	114
5-3	Modelling within the context of a Cartesian representation between accuracy and interpretability - the multi-objective optimisation.....	116
5-4	A framework for the proposed fuzzy modelling approach.....	119
5-5	Examples of data clustering.....	121
5-6	Data samples of a clustering problem.....	124
5-7	The hierarchical clustering dendrogram of the data samples shown in Figure 5-6.....	125
5-8	An example of the initial fuzzy model construction.....	130
5-9	<i>RMSE</i> of the validation data in the optimisation process.....	143
5-10	The initial fuzzy model's predicted UTS versus measured UTS: (a) all the training data (2820 data), (b) the selected training data (448 data).....	147
5-11	The trained fuzzy model's predicted UTS versus measured UTS: (a) all the training data (2820 data), (b) the selected training data (448 data).....	150
5-12	Examples of the missing representative data.....	151

## **List of Figures**

5-13	Flow chart of the mechanism for accuracy optimisation and data selection.....	152
5-14	The well-trained fuzzy model's predicted UTS versus measured UTS: (a) all the training data (2820 data), (b) the selected training data (480 data).....	153
5-15	An example of a redundant rule.....	155
5-16	An example of the membership function relates to a redundant fuzzy set.....	159
5-17	An example of membership functions of similar fuzzy sets.....	160
5-18	The membership functions of Inputs 14 and 15 before the interpretability improvement.....	163
5-19	The membership functions of Inputs 14 and 15 after the interpretability improvement.....	163
5-20	The performance of fuzzy models following the interpretability improvement with different $Th_1$ : (a) <i>RMSE</i> versus $Th_1$ ; (b) the number of rules versus $Th_1$ ; (c) the number of fuzzy sets versus $Th_1$ ; (d) the total length of rules versus $Th_1$ .....	165
5-21	The performance of fuzzy models following the interpretability improvement with different $Th_2$ : (a) <i>RMSE</i> versus $Th_2$ ; (b) the number of rules versus $Th_2$ ; (c) the number of fuzzy sets versus $Th_2$ ; (d) the total length of rules versus $Th_2$ .....	166
5-22	The performance of fuzzy models following the interpretability improvement with different $Th_3$ : (a) <i>RMSE</i> versus $Th_3$ ; (b) the number of rules versus $Th_3$ ; (c) the number of fuzzy sets versus $Th_3$ ; (d) the total length of rules versus $Th_3$ .....	167
5-23	The performance of fuzzy models following the interpretability improvement with different $Th_4$ : (a) <i>RMSE</i> versus $Th_4$ ; (b) the number of rules versus $Th_4$ ; (c) the number of fuzzy sets versus $Th_4$ ; (d) the total length of rules versus $Th_4$ .....	169
5-24	The framework of the proposed multi-objective optimisation mechanism.....	171

## **List of Figures**

5-25	Example of the $\alpha\%$ -range confidence band: (a) the prediction performance of the UTS model; (b) the $\alpha\%$ -range confidence band of the UTS model.....	175
6-1	The performance of the optimised Pareto-optimal fuzzy models for the nonlinear function approximation problem (6.1).....	179
6-2	The fuzzy models' predicted outputs versus the measured outputs with the nonlinear function approximation problem: (a) the initial model, (b) an optimised model with 8 rules, (c) an optimised model with 6 rules, and (d) an optimised model with 4 rules; the green and red lines represent the +10% and -10% error bands respectively.....	182
6-3	The fuzzy models' membership functions with the nonlinear function approximation problem: (a) the initial model, (b) an optimised model with 8 rules, (c) an optimised model with 6 rules, and (d) an optimised model with 4 rules.....	183
6-4	The optimised 8-rule fuzzy model for the nonlinear function approximation problem.....	185
6-5	Response surfaces for the nonlinear function approximation problem: (a) the actual system and (b) the optimised 8-rule fuzzy system.....	186
6-6	(a) The prediction performance and (b) the 5%-range confidence band of the optimised 8-rule fuzzy model for the nonlinear function approximation problem.....	186
6-7	The input $u(k)$ , unforced system $g(k)$ , and output $y(k)$ of the plant (6.3).....	188
6-8	The performance of the optimised Pareto-optimal models for the dynamical system identification problem.....	189
6-9	The fuzzy models' predicted outputs versus measured outputs with the dynamical system identification problem: (a) the initial model, (b) an optimised model with 6 rules, (c) an optimised model with 4 rules, and (d) an optimised model with 3 rules; the green and red lines represent the +10% and -10% error bands respectively.....	192

## **List of Figures**

6-10	The fuzzy models' membership functions with the dynamical system identification problem: (a) the initial model, (b) an optimised model with 6 rules, (c) an optimised model with 4 rules, and (d) an optimised model with 3 rules.....	193
6-11	The optimised 6-rule fuzzy model for the dynamical system identification problem.....	195
6-12	Response surfaces for the dynamical system identification problem: (a) the actual system and (b) the optimised 6-rule fuzzy model.....	195
6-13	Training and testing errors of the optimised 6-rule fuzzy model for the dynamical system identification problem.....	196
6-14	(a) The prediction performance and (b) the 5%-range confidence band of the optimised 6-rule fuzzy model for the dynamical system identification problem.....	196
6-15	The performance of the Pareto-optimal UTS models.....	199
6-16	The UTS models' predicted outputs versus measured outputs: (a) the initial model, (b) an optimised model with 13 rules, and (c) an optimised model with 10 rules; the green and red lines represent the +10% and -10% error bands respectively.....	201
6-17	The UTS models' membership functions: (a) the initial model, (b) an optimised model with 13 rules, and (c) an optimised model with 10 rules.....	202
6-18	Rules of the optimised 10-rule UTS model.....	204
6-19	Response surfaces of the optimised 10-rule UTS model.....	205
6-20	(a) The prediction performance and (b) the 5%-range confidence band of the optimised 10-rule UTS model.....	206
6-21	The performance of the Pareto-optimal ROA models.....	207
6-22	The ROA models' predicted outputs versus measured outputs: (a) the initial model, (b) an optimised model with 15 rules, and (c) an optimised model with 7 rules; the green and red lines represent the +10% and -10% error bands respectively.....	209



## **List of Figures**

6-23	The ROA models' membership functions: (a) the initial fuzzy model, (b) an optimised fuzzy model with 15 rules, and (c) an optimised fuzzy model with 7 rules.....	210
6-24	Response surfaces of the optimised 15-rule ROA model.....	211
6-25	(a) The prediction performance and (b) the 5%-range confidence band of the optimised 15-rule ROA model.....	212
6-26	The performance of the Pareto-optimal Elongation models.....	213
6-27	The Elongation models' predicted outputs versus measured outputs: (a) the initial model, (b) an optimised model with 10 rules, and (c) an optimised model with 8 rules; the green and red lines represent the +10% and -10% error bands respectively.....	215
6-28	The Elongation models' membership functions: (a) the initial fuzzy model, (b) an optimised fuzzy model with 10 rules, and (c) an optimised fuzzy model with 8 rules.....	216
6-29	Response surfaces of the optimised 10-rule Elongation model.....	217
6-30	(a) The prediction performance and (b) the 5%-range confidence band of the optimised 10-rule Elongation model.....	218
6-31	The performance of the Pareto-optimal Impact Energy models.....	219
6-32	The Impact Energy models' predicted outputs versus measured outputs: (a) the initial model, (b) an optimised model with 15 rules, and (c) an optimised model with 8 rules; the green and red lines represent the +10% and -10% error bands respectively.....	221
6-33	The Impact Energy models' membership functions: (a) the initial fuzzy model, (b) an optimised fuzzy model with 15 rules, and (c) an optimised fuzzy model with 8 rules.....	222
6-34	Response surfaces of the optimised 15-rule Impact Energy model.....	223
6-35	(a) The prediction performance and (b) the 5%-range confidence band of the optimised 15-rule Impact Energy model.....	224
7-1	The prediction performance of the UTS model and the ROA models used in this chapter; the red and green lines delimit the +10% and -10% confidence bands respectively.....	227

## **List of Figures**

7-2	Average fitness of 10 runs versus function evaluation for the UTS design problem with $UTS_{Target} = 900$ (MPa).....	229
7-3	Average fitness of 10 runs versus function evaluation for the UTS design problem with $UTS_{Target} = 1100$ (MPa).....	229
7-4	Average fitness of 10 runs versus function evaluation for the ROA design problem with $ROA_{Target} = 60$ (%).....	232
7-5	Average fitness of 10 runs versus function evaluation for the ROA design problem with $ROA_{Target} = 70$ (%).....	234
7-6	The maximum and minimum boundaries (Pareto fronts) for the problem of designing UTS and ROA simultaneously.....	237
7-7	The performance of the Pareto-optimal solutions for the design problem of $UTS_{Target} = 600$ (MPa) and $ROA_{Target} = 50$ (%) with respect to (a) the Objective 1 and the Objective 2 and (b) the UTS and the ROA.....	240
7-8	The performance of the Pareto-optimal solutions for the design problem of $UTS_{Target} = 900$ (MPa) and minimising the heat treatment cost with respect to (a) Objective 1 and Objective 2; (b) UTS and Cost.....	243
7-9	The performance of the Pareto-optimal solutions for the design problem of $ROA_{Target} = 60$ (%) and minimising the heat treatment cost with respect to (a) Objective 1 and Objective 2; (b) ROA and Cost.....	245
7-10	The performance of the Pareto-optimal solutions for the design problem of $UTS_{Target} = 900$ (MPa), $ROA_{Target} = 60$ (%) and minimising the heat treatment cost with respect to (a) Objective 1 and Objective 3; (b) Objective 2 and Objective 3; (c) UTS and Cost; and (d) ROA and Cost.....	247
8-1	The layout of the designed fuzzy modelling GUI.....	253
8-2	An example of the modelling process using the designed fuzzy modelling GUI: (a) loading data, (b) running the optimisation, (c) displaying the prediction performance of the training data and (d) output predicting from a set of provided discrete inputs.....	253

**List of Figures**

8-3 An example of analysing the loaded data and the obtained model using the designed fuzzy modelling GUI: (a) the training data with variable ID(s) being 1 and 3, (b) the histograms of the density of the training data with variable ID(s) being 1 and 3, (c) the measured outputs vs. the predicted outputs of the training data, and (d) the inputs and output sensitivity plot (response surface) with various variable ID(s) being 1 and 2..... 254

## List of Tables

Table		Page
3-1	The average number of function evaluations to find the optimum; the bold values represent the best results.....	43
3-2	Average performance of RSSA with different variation strategies in optimising $f_5$ to $f_{11}$ : The integer in every cell is the average function evaluation number in successful runs; the value between the parentheses is the average result in the unsuccessful runs; the percentage value in the square brackets indicates the percentage of the successful runs out of all the runs; the bold values represent the best results.....	45
3-3	Average performance of various algorithms in optimising $f_1$ to $f_{14}$ : The integer in every cell is the average function evaluation number in successful runs; the value between parentheses is the average result in the unsuccessful runs; the percentage value in the square brackets indicates the percentage of the successful runs out of all the runs; the bold values represent the best results.....	48
3-4	Mean and variance values of $GD$ ; the bold values represent the best results.....	58
3-5	Mean and variance values of $\Delta$ ; the bold values represent the best results.....	58
4-1	The average number of function evaluations to find the optimum on $f_1$ ; the bold values represent the best results.....	82

## **List of Tables**

4-2	The average number of function evaluations to find the optimum on $f_2$ ; the bold values represent the best results.....	83
4-3	The average number of function evaluations to find the optimum on $f_3$ ; the bold values represent the best results.....	83
4-4	The average number of function evaluations to find the optimum on $f_7$ ; the bold values represent the best results.....	83
4-5	The average number of function evaluations to find the optimum on $f_8$ ; the bold values represent the best results.....	84
4-6	The average number of function evaluations to find the optimum on $f_{10}$ ; the bold values represent the best results.....	84
4-7	Average performance of nPSO with various acceleration coefficients in optimising $f_1$ to $f_{14}$ : The integer value in every cell is the average function evaluation number in successful runs; the value in parenthesis is the average result in the unsuccessful runs; the percentage value in the square brackets indicates the percentage of the successful runs out of all the runs; the bold values represent the best results.....	87
4-8	Average performance of nPSO with different position parameter refresh mechanisms in optimising $f_5$ to $f_{11}$ : The integer in every cell is the average function evaluation number in successful runs; the value in parenthesis is the average result in the unsuccessful runs; the percentage value in the square brackets indicates the percentage of the successful runs out of all the runs; the bold values represent the best results.....	89
4-9	Average performance of various algorithms in optimising $f_1$ to $f_{14}$ : The integer in every cell is the average function evaluation number in successful runs; the value in parenthesis is the average result in the unsuccessful runs; the percentage value in the square brackets indicates the percentage of the successful runs out of all the runs; the bold values represent the best results.....	93
4-10	Mean and variance values of $GD$ ; the bold values represent the best results.....	97

## **List of Tables**

4-11	Mean and variance values of A; the bold values represent the best results.....	97
5-1	Comparative results of different clustering algorithms for Mamdani fuzzy systems construction with the function approximation problem	136
5-2	Comparative results of different clustering algorithms for Mamdani fuzzy systems construction with the ROA modelling problem.....	136
5-3	Comparative results of different clustering algorithms for Mamdani fuzzy systems construction with the UTS modelling problem.....	137
5-4	Comparative results of different distance weights for Mamdani fuzzy systems (with 8 rules) construction with the function approximation problem; the bold values represent the best results.....	139
5-5	Comparative results of different distance weights for Mamdani fuzzy systems (with 15 rules) construction with the ROA modelling problem; the bold values represent the best results.....	140
5-6	Comparative results of different distance weights for Mamdani fuzzy systems (with 30 rules) construction with the UTS modelling problem; the bold values represent the best results.....	140
5-7	The main parameters of the UTS fuzzy models following the different stages of the interpretability improvement.....	162
6-1	The performance comparison of various models for the nonlinear function approximation problem.....	181
6-2	The performance comparison of various models for the dynamical system identification problem.....	191
6-3	Main parameters of the obtained UTS models.....	200
6-4	Main parameters of the obtained ROA models.....	208
6-5	Main parameters of the obtained Elongation models.....	214
6-6	Main parameters of the obtained Impact Energy models.....	220
7-1	Optimisation solutions of 10 independent runs for the UTS design problem with $UTS_{Target} = 900$ (MPa).....	230
7-2	Optimisation solutions of 10 independent runs for the UTS design problem with $UTS_{Target} = 1100$ (MPa).....	231

## **List of Tables**

7-3	Optimisation solutions of 10 independent runs for the ROA design problem with $ROA_{Target} = 60$ (%).....	233
7-4	Optimisation solutions of 10 independent runs for the ROA design problem with $ROA_{Target} = 70$ (%).....	235
7-5	Optimisation solutions of 10 independent runs for the design problem with $UTS_{Target} = 900$ (MPa) and $ROA_{Target} = 70$ (%).....	238
7-6	Pareto-optimal solutions for the design problem with $UTS_{Target} = 600$ (MPa) and $ROA_{Target} = 50$ (%).....	240
7-7	Contribution of composites to the cost of heat treatment.....	242
7-8	Contribution of tempering (annealing) to the cost of heat treatment....	242
7-9	Ten of the Pareto-optimal solutions for the design problem of $UTS_{Target} = 900$ (MPa) and minimising the heat treatment cost.....	244
7-10	Ten of the Pareto-optimal solutions for the design problem of $ROA_{Target} = 60$ (%) and minimising the heat treatment cost.....	245
7-11	Ten of the Pareto-optimal solutions for the design problem of $UTS_{Target} = 900$ (MPa), $ROA_{Target} = 60$ (%) and minimising the heat treatment cost.....	248

## **Chapter 1**

# **Introduction**

### **1.1 Background and Research Aims**

Knowledge discovery is one of the most important activities that, we humans, undertake almost on a daily basis. It helps people recognise and understand some of the intricacies of the ancient and modern worlds. In computer science, knowledge discovery is one of the most desirable end-products of computing and is also one of the most difficult computing challenges to undertake.

For this purpose, one can identify two complementary approaches, which are knowledge acquisition from experts and knowledge discovery from data. “Knowledge acquisition from experts often includes discovery as a by-product, since the formalisation often uncovers new linkages. But that discovery also depends on human recognition of unexpected phenomena. Such discoveries must often be validated with broader tests, since a single expert typically has a narrow



## ***Chapter 1: Introduction***

---

view of the world.” [Fayyad *et al.* 1996]. In some of new and more complex systems, the knowledge cannot even be obtained from experts because of the lack of understanding of the systems themselves.

Discovering knowledge from data can help to overcome the above limitations. During the last two decades, one would have witnessed an explosive growth in humans’ capabilities to both generate and collect data. By analysing and summarising these data, one can extract ‘useful’ knowledge from such information. Experts may have known one part of the knowledge, but the other part is totally new to us humans.

To employ the idea of discovering knowledge from data, the format of knowledge expression must be determined first. In some particular domains, appropriate models exist and can be used in new knowledge expression. Experts can understand the new knowledge based on these models. But in some other domains, there are currently no models that can be used in a new knowledge expression. In this situation, fuzzy systems appear to be suitable for complex and uncertain environments.

Fuzzy systems are known to be universal approximators [Wang 1992] and good at modelling complex, nonlinear, or partially unknown systems [Babuska 1998; Passino & Yurkovich 1998; Wang 1997]. These systems are normally based on linguistic knowledge expressions that are easy to be understood not only by an expert but also by an even wider audience. The advantages of fuzzy systems in

## ***Chapter 1: Introduction***

---

both adaptability and transparency make them suitable for knowledge discovery in this current research project.

A Mamdani-type fuzzy system [Mamdani & Assilian 1975] is the first type of fuzzy systems, which is based on Zadeh's theories in human-machine interaction [Zadeh 1973]. It uses linguistic expressions in both the antecedent and consequent parts. Later, the TSK-type fuzzy system [Takagi & Sugeno 1985] was introduced by replacing the linguistic consequent parts of the Mamdani-type fuzzy system with mathematical functions. Because of their computational efficiency and their high accuracy, TSK-type fuzzy systems seem attractive and as a result abundant research has been carried-out based on their associated architectures in recent years. However, one cannot ignore the fact that on a linguistic level these numerically improved systems are somewhat meaningless to human operators.

In fact, the most attractive property of fuzzy systems lies in their ability of processing information in linguistic terms. But it can be argued now that this aspect is somewhat neglected and sacrificed to numerical accuracy. In this current research project, more attention will be paid to building more transparent (interpretable) fuzzy systems.

To implement a fuzzy system in a knowledge discovery context, it must cooperate with other paradigms that include some learning abilities. Two of the most successful attempts to hybridise fuzzy systems with learning and adaptation methods include neural fuzzy systems [Jang *et al.* 1997; Nauck *et al.* 1997; Fuller

## ***Chapter 1: Introduction***

---

1999] and genetic fuzzy systems [Pedrycz 1997; Cordon *et al.* 2001].

Neural fuzzy systems have been studied extensively in the last ten years while genetic fuzzy systems are still not fully developed. In recent years, better evolutionary computing based algorithms other than genetic algorithms (GAs) [Holland 1975] have been researched. These should be helpful in improving the learning abilities of fuzzy systems if used in cooperation with these algorithms. However, most of the current research still focuses on the hybridisation of fuzzy systems with only GAs. On the other hand, research about GA-based approaches focuses on simple problems and discovering knowledge at a low level, most of which are of two or three-input one-output systems. Only a few high-quality genetic fuzzy systems have been proposed and used in real industrial applications. Thus, in the research contained in this thesis, more emphasis will be directed forwards the cooperation of fuzzy systems with some effective evolutionary computing based optimisation algorithms.

It is worth nothing that data-driven modelling based on fuzzy systems possesses two conflicting requirements: accuracy (precision) and transparency (interpretability). Accuracy is easy to embrace as it relates to the capability of representing a real system faithfully, this representing a fundamental requirement for models. In contrast, interpretability means that human beings should be able to understand a fuzzy system's behaviour by inspecting its associated rule-base. The latter is crucial in the field of data mining and knowledge discovery, where information should be extracted from data bases and represented in a

## ***Chapter 1: Introduction***

---

comprehensible form, or for decision support systems where the reasoning process should be transparent to the users [Mikut *et al.* 2005]. To deal with this type of problems with conflicting objectives, multi-objective optimisation techniques [Deb 2001] becomes a natural choice. Another advantage of the multi-objective optimisation technique is that it will lead to a set of Pareto-optimal models with different accuracy and interpretability levels, instead of only one solution if using other learning techniques. This should provide more options and add more flexibility to users. Therefore, designing efficient optimisation algorithms, including multi-objective optimisation algorithms, will form another task within this research project.

Through the simulation of natural and social behaviours, researchers have succeeded in developing many successful approaches to solve complex problems. Artificial neural networks and evolutionary algorithms are two of the most salient representatives. Artificial neural networks imitate the structure of biological neural networks and mimic the process of human learning and memory to manage information. On the other hand, most of the introductions of evolutionary algorithms are also inspired by some natural or social behaviour. For example, Genetic Algorithms (GA) imitate the natural evolution, while Particle Swarm Optimisation (PSO) [Kennedy & Eberhart 1995] algorithms mimic the behaviour of birds 'flock'. In the same way, more effort will be devoted to develop nature-inspired or social-inspired methods for optimisation problems.

Once appropriate optimisation and modelling strategies have been designed, it is

## ***Chapter 1: Introduction***

---

necessary to validate them in real industrial problems. In this research project, the proposed methodologies will be applied to the modelling and design problems relating to the steel (metal) industry.

In materials engineering, it is important to establish appropriate and reliable mechanical property prediction models for materials design and development. In the past, several mechanical property models were developed, which were mainly based on linear regression methods [Pickering 1978] or artificial neural networks [Hodgson 1996; Chen *et al.* 1998; Bakshi & Chatterjee 1998]. Linear models are only designed for specific classes of steels and specific processing routes, and are not sophisticated enough to account for more complex interactions, while neural networks can be seen as black-box techniques and the knowledge behind this kind of models cannot be understood fully. Thus, developing a fast, efficient and transparent data-driven modelling framework for material property prediction is still needed. In this situation, fuzzy modelling provides an ideal approach because of its interpretable structure and its excellent ability of learning from data.

If intelligent models in the form of fuzzy systems were constructed to predict the mechanical test results for alloy steels, then these models can be implemented to facilitate the optimal design of alloy steels. In the steel industry, determining the optimal heat treatment regime and the required weight percentages for chemical composites to obtain the desired mechanical properties of the steel is always a challenging multi-objective optimisation problem. Usually, some objectives may conflict with each other, such as the Ultimate Tensile Strength (UTS) and the

## ***Chapter 1: Introduction***

---

ductility. By using the designed multi-objective optimisation algorithms and the developed prediction models, the design targets can be achieved.

Above all, in this project, a cooperative fuzzy modelling framework is investigated, which integrates such transparent fuzzy systems with the effective evolutionary computing based algorithms and involves many techniques, such as data clustering, data mining, as well as multi-objective optimisation. Evolutionary optimisation algorithms are also developed following the nature-inspired or the social-inspired ideas, which are an essential part of the modelling framework and are employed in the optimal design problems as well. This cooperative fuzzy modelling methodology and the proposed evolutionary optimisation algorithms are then applied to knowledge discovery, in terms of system modelling and optimal design, using simulation platforms as well as real industrial data.

### **1.2 Structure of the Thesis**

The next paragraphs will describe the key points covered by the various chapters in this thesis.

Chapter 2 will introduce some basic knowledge relating to this project. This will include the basic concepts relating to fuzzy sets, fuzzy systems, optimisation and multi-objective optimisation. The frameworks relating to single objective and multi-objective optimisation algorithms will also be described.

## ***Chapter 1: Introduction***

---

Chapter 3 will propose a new nature-inspired optimisation algorithm, Reduced Space Searching Algorithm (RSSA). This algorithm will be validated using a set of well-known benchmark problems and compared with some recently developed and most salient optimisation algorithms, the Covariance Matrix Adaptation Evolution Strategy (CMA-ES), the Differential Evolution (DE) and the Generalised Generation Gap model with the Parent-Centric Recombination operator (G3-PCX). In addition, various parameter settings will be explored and the appropriate parameter configurations will be outlined. Furthermore, this new algorithm will be extended to the multi-objective optimisation case (MO-RSSA), in which the Random Weighted Aggregation (RWA) technique will be employed and a new approach named ‘cell selection’ method will be introduced in order to keep a good diversity of the Pareto-optimal solutions. A comparative study between MO-RSSA and other MOEAs, such as the Pareto Archived Evolution Strategy (PAES), the Strength Pareto Evolutionary Algorithm (SPEA) and the Non-dominated Sorting Genetic Algorithm II (NSGA-II) will be carried-out based on a set of challenging problems, such as the ZDT and DTLZ series problems.

In Chapter 4, a new social-inspired algorithm, the new Particle Swarm Optimisation (nPSO), will be proposed, which will introduce a new ‘momentum term’ to replace the original inertia term of the standard PSO. This algorithm will be validated using a set of benchmark problems and will be compared with the standard PSOs and some other salient optimisation algorithms. In addition, various parameter settings will be explored in detail and the appropriate parameter

## ***Chapter 1: Introduction***

---

configurations will also be outlined. Furthermore, nPSO will be extended to the multi-objective optimisation case and the newly developed multi-objective PSO (nMPSO) will be compared with several well-known evolutionary multi-objective algorithms, such as PAES, SPEA and NSGA-II, via the ZDT and DTLZ series problems.

Chapter 5 will propose a new framework for data-driven fuzzy modelling, named the Fuzzy Modelling approach with a Hierarchical Clustering algorithm and a Multi-objective Optimisation mechanism (FM-HCMO), in order to construct linguistic fuzzy models considering both the accuracy and the interpretability of fuzzy systems. In this methodology, a new agglomerative complete-link clustering algorithm will be first developed and applied to construct an initial fuzzy model. A new data selection technique will then be proposed to select the representative training data used to improve the modelling efficiency. A multi-objective optimisation mechanism will then be developed for the improvement of modelling performance, which will take into account both the accuracy and interpretability attributes. Finally, a method for computing the confidence bands relating to the model prediction analysis will be described. All of these proposed techniques will be validated via a series of experiments using real industrial data from the steel industry.

In Chapter 6, the proposed modelling framework FM-HCMO will be validated as a whole. The test problems will include the benchmark problems relating to the identification of nonlinear, static and dynamic systems, as well as the modelling



## ***Chapter 1: Introduction***

---

problems of the mechanical properties for alloy steels, namely the Ultimate Tensile Strength (UTS), Reduction of Area (ROA), Elongation and Impact Energy.

In Chapter 7, RSSA and MO-RSSA will be applied to single objective and multi-objective optimal design of alloy steels. This research aims at determining the optimal heat treatment regime and the required weight percentages for chemical composites to obtain the desired mechanical properties of steel, such as UTS and ROA. In addition, the work will later be extended to include economic factors, such as the costs associated with the composites and the machining operation.

Finally, Chapter 8 will detail the conclusions resulting from the work within this project, together with future research directions.

## Chapter 2

# Evolutionary Based Optimisation and Fuzzy Theories - A Background

### 2.1 Optimisation

#### 2.1.1 Single Objective Optimisation

In mathematics, the term optimisation refers to the study of problems in which one seeks to minimise or maximise a real function by systematically choosing the values of real or integer variables from within an allowed set.

Normally, an optimisation problem can be represented in the following way:

Given a function  $f: A \rightarrow \mathcal{R}$ , seek a solution  $\vec{x}^* \in A$  such that  $f(\vec{x}^*) \leq f(\vec{x})$  for all  $\vec{x}$  in  $A$  (minimisation) or such that  $f(\vec{x}^*) \geq f(\vec{x})$  for all  $\vec{x}$  in  $A$  (maximisation).

## ***Chapter 2: A Background***

---

Typically,  $A$  is the subset of the Euclidean space  $R^n$  and is often specified by a set of constraints, equalities or inequalities that the members of  $A$  should satisfy. The domain  $A$  of  $f$  is called the search space, while the elements  $\vec{x}$  of  $A$  are called candidate solutions or feasible solutions. The function  $f$  is called an objective function, or cost function. The feasible solution that minimises (or maximises, if that is the goal) the objective function is called an optimal solution.

Generally, the optimisation problems may include some local minima or maxima, where a local minimum  $\vec{x}^l$  is defined as a point, for which there exists some  $\delta > 0$  so that for all  $\vec{x}$  where  $\|\vec{x} - \vec{x}^l\| \leq \delta$ , the expression  $f(\vec{x}^l) \leq f(\vec{x})$  holds. That is to say, in some region around  $\vec{x}^l$ , all of the function values are greater than or equal to the value at the  $\vec{x}^l$  point. Local maxima are defined similarly.

### **2.1.2 Multi-Objective Optimisation**

Multi-objective optimisation [Sawaragi *et al.* 1985; Steuer 1986], also known as multi-criteria optimisation, is the process of simultaneously optimising two or more conflicting objectives subject to certain constraints.

In mathematical terms, the multi-objective problem can be written as follows:

Find a vector  $\vec{x}^*$  that will optimise the following vector function:

$$f(\vec{x}) = [f_1(\vec{x}), f_2(\vec{x}), f_3(\vec{x}), \dots, f_k(\vec{x})]$$

## Chapter 2: A Background

---

subject to the  $m$  inequality constraints  $g_i(\vec{x}) \leq 0$ ,  $i = 1, 2, 3, \dots, m$ , and the  $p$  equality constraints  $h_j(\vec{x}) = 0$ ,  $j = 1, 2, 3, \dots, p$ , where  $\vec{x}$  is the vector of decision variables.

If a multi-objective problem is well formed, there should not be a single solution that simultaneously optimises each objective to its fullest. Meanwhile, multiple solutions exist, for which each objective has been optimised to the extent that if one tries to optimise it any further, then the other objective(s) will suffer as a result. These solutions are regarded as the answer to the multi-objective optimisation problem, which are called Pareto-optimal solutions or non-dominated solutions [Sawaragi *et al.* 1985; Steuer 1986; Deb 2001].

Pareto-optimal solutions (non-dominated solutions) are those for which improvement in one objective can only occur with the worsening of at least one other objective. They are defined as follows (for a minimisation problem):

One solution  $\vec{x}^* \in A$  is Pareto optimal, where  $A$  is the feasible solution space, if for every  $\vec{x} \in A$  either  $f_i(\vec{x}) = f_i(\vec{x}^*)$ ,  $\forall i \in k$ , where  $k$  is the number of objectives, or there is at least one  $i \in k$  such that  $f_i(\vec{x}) > f_i(\vec{x}^*)$ .

The above definition means that  $\vec{x}^*$  is Pareto optimal if there is no feasible vector  $\vec{x}$  that would decrease some objective values without causing a simultaneous increase in at least one other objective value [Coello Coello 1999].

## ***Chapter 2: A Background***

---

Multi-objective optimisation problems can be found in various fields such as product and process design, finance, aircraft design, the oil and gas industry, automobile design, or wherever optimal decisions need to be taken in the presence of trade-offs between two or more conflicting objectives, for instance, maximising the profit and minimising the cost of a product; maximising the performance and minimising the fuel consumption of a vehicle; and minimising the weight while maximising the strength of a particular component.

### **2.2 Evolutionary Computation**

Evolutionary computation [Eiben & Smith 2003; De Jong 2006] is a subfield of artificial intelligence that involves optimisation problems. It uses iterative progress with a growth or development in a population. This population is then selected in a guided random search using parallel processing to achieve the desired end. Such processes are often inspired by biological mechanisms of evolution.

Evolutionary techniques mostly involve evolutionary algorithms (comprising genetic algorithms [Mitchell 1996], evolutionary programming [Eiben & Smith 2003], evolution strategy [Beyer 2001], genetic programming [Langdon & Poli 2002] and learning classifier systems [Bull & Kovacs 2005]), swarm intelligence (comprising ant colony optimisation [Dorigo & Stutzle 2004] and particle swarm optimisation [Kennedy *et al.* 2001]), self-organising maps [Kohonen 2001], differential evolution [Price *et al.* 2005], and artificial immune systems [De Castro

## ***Chapter 2: A Background***

---

& Timmis 2002].

### **2.2.1 Evolutionary Algorithms**

In artificial intelligence, evolutionary algorithms (EA) are the generic population-based metaheuristic optimisation algorithms, which are a subset of evolutionary computation. An EA uses some mechanisms (operators) inspired by biological evolution: reproduction, mutation, recombination, and selection. Candidate solutions to the optimisation problem play the role of individuals in a population, and the cost function determines the environment within which the solutions ‘live’. Evolution of the population then takes place after the repeated application of the above operators.

Evolutionary algorithms consistently perform well in approximating solutions to all types of problems because they do not make any assumptions about the underlying fitness landscape. This generality is shown by successes in fields as diverse as engineering, art, biology, economics, marketing, genetics, operations research, robotics, social sciences, physics, politics, and chemistry.

#### **2.2.1.1 Genetic Algorithms**

Genetic Algorithms (GA) are the most widely known types of evolutionary

## ***Chapter 2: A Background***

---

algorithms. They are inspired by evolutionary biology such as inheritance, mutation, natural selection, and recombination (or crossover). Over the last two decades, GAs have been extensively used as search and optimisation tools in various problem domains, including science, commerce and engineering. The main reasons for their success lie in their broad applicability, ease of use and global perspective [Goldberg 1989].

The concept of a genetic algorithm was first proposed by John Holland of the University of Michigan in 1975 [Holland 1975]. Most of the initial research work can be found in various early international conference proceedings and several textbooks [Goldberg 1989; Holland 1975; Michalewicz 1992; Back *et al.* 1997]. Some journals are dedicated to promote research in evolutionary algorithms, certainly including GAs, such as ‘Evolutionary Computation Journal’ published by MIT Press, ‘Transactions on Evolutionary Computation’ published by IEEE and ‘Genetic Programming and Evolutionary Computation’ published by Kluwer Academic Publishers. New developments about GAs and other evolutionary algorithms can be found in these journals as well as in recent international conference proceedings.

### **2.2.1.2 Evolution Strategies**

The idea behind Evolution Strategies (ES) represents a joint development of Bienert, Rechenberg and Schwefel in the 1960s at the Technical University of

## ***Chapter 2: A Background***

---

Berlin (TUB) in Germany. The first application of ESs was experimental and attempted to solve the shape optimisation of a bended pipe, the drag minimisation of a jointed plate and the shape optimisation of a flashing nozzle. Thereafter, different versions of ESs were suggested, such as multi-membered ESs, recombinative ESs and self-adaptive ESs. More details can be found in [Michalewicz 1992; Schwefel & Rudolph 1995; Back *et al.* 1997].

Though the ESs' working principle is similar to that of a real-parameter GA used with selection and mutation operators only. The early ES is fundamentally different from the early binary GAs in mainly two ways: (1) ESs use real values for coding and (2) ESs do not use any crossover-like operator. In addition, the step size of ESs' mutation operator can adjust itself adaptively during the optimisation process. That gives ESs the capability of self-adaptation which GAs do not have. The latter is also the reason why there are still many researchers interested in ESs.

### **2.2.2 Swarm Intelligence**

Swarm intelligence (SI) is an artificial intelligence based on the collective behaviour of decentralised, self-organised systems. The expression was introduced by Gerardo Beni and Jing Wang in 1989, in the context of cellular robotic systems [Beni & Wang 1989].

SI systems are typically made up of a population of simple agents interacting



## ***Chapter 2: A Background***

---

locally with one another and with their environment. The agents follow very simple rules, and although there is no centralised control structure dictating how individual agents should behave, local interactions between such agents lead to the emergence of complex global behaviour. Natural examples of SI include ant colonies, bird flocking, animal herding, bacterial growth, and fish schooling. The most successful algorithms employing swarm intelligence are Ant Colony Optimisation (ACO) [Dorigo *et al.* 1996] and Particle Swarm Optimisation (PSO) [Kennedy *et al.* 2001].

### **2.2.2.1 Particle Swarm Optimisation (PSO)**

Particle swarm optimisation is a population-based evolutionary computing algorithm for problem solving. It is the type of swarm intelligence that is based on social-psychological principles and provides insights into social behaviour, as well as contributing to engineering applications. The particle swarm optimisation algorithm was first described in 1995 by James Kennedy and Russell C. Eberhart [Eberhart & Kennedy 1995; Kennedy & Eberhart 1995] and the techniques have evolved greatly since then.

Since its introduction in 1995, the PSO method has become very popular due to its simplicity of implementation and ability to quickly converge to a reasonably good solution. A fair amount of research results have been reported in the literature and the first book dedicated to PSO [Kennedy *et al.* 2001] has been published in 2001.

## ***Chapter 2: A Background***

---

### **2.2.3 Multi-Objective Optimisation Algorithms within Evolutionary**

#### **Computation**

Most real-world search and optimisation problems naturally involve multiple objectives. Knowledge discovery can also be seen as multi-objective. For example, one objective is that the knowledge discovered should be accurate and the other objective is that the knowledge should be transparent. These two objectives conflict with each other to a certain extent. With the accuracy of the knowledge increasing, the amount of the knowledge will also increase. This will make the knowledge less transparent for people to understand. If we want to decrease the amount of the knowledge for easier and better understanding, the accuracy should decrease. Zadeh termed this ‘principle of incompatibility’ [Zadeh 1973]. To solve multi-objective problems in knowledge discovery, multi-objective optimisation techniques are needed. Here, several well-known evolutionary multi-objective optimisation algorithms which have been developed in recent years will be discussed.

#### **2.2.3.1 Strength Pareto Evolutionary Algorithm**

The Strength Pareto Evolutionary Algorithm (SPEA) was introduced by Zitzler and Thiele [Zitzler & Thiele 1998] and was proposed as a way of integrating different Multi-Objective Evolutionary Algorithms (MOEAs). This algorithm introduces elitism by maintaining an archive to store non-dominated solutions

## ***Chapter 2: A Background***

---

previously found, which is called an ‘external non-dominated set’. At every generation, newly found non-dominated solutions are compared with the existing external population and the best non-dominated individuals are copied to the external non-dominated set. For each individual in the external set, a strength value is calculated, which is proportional to the number of solutions to which the certain individual dominates. The fitness of each individual in the current population is calculated according to the strength of all external non-dominated solutions that dominates it. To maintain diversity, a clustering technique ‘average linkage method’ is used.

A second version of the original algorithm, Strength Pareto Evolutionary Algorithm 2 (SPEA2), was proposed in 2001 [Zitzler *et al.* 2001]. It has three main differences with respect to its predecessor: (1) it improves the fitness assignment scheme which, for each individual, takes into account how many individuals that it dominates and it is dominated by; (2) it incorporates a nearest neighbour density estimation technique which gives a more precise guidance for the search process, and (3) it uses an enhanced archive truncation method to guarantee the preservation of boundary solutions.

### **2.2.3.2 Pareto Archived Evolution Strategy**

The Pareto Archived Evolution Strategy (PAES) was introduced by Knowles and Corne in 2000 [Knowles & Corne 2000]. In its simplest form, PAES consists of a

## ***Chapter 2: A Background***

---

(1+1) evolution strategy, which means that a single parent generates a single offspring in every generation. In PAES, a historical archive, which is the elitist mechanism of the algorithm, is used to record some of the non-dominated solutions previously found. This archive works as a reference set against which each mutated individual is being compared. To maintain diversity, the algorithm uses a mechanism that consists of a crowding procedure that divides objective space in a recursive manner. Each solution is located in a certain grid based on its objective value. The number of solutions that reside in each grid location is recorded in an external memory. This information is used for the selection and the archiving processes. This method has two advantages over other methods used in some multi-objective GAs: Its computational cost is lower; it is adaptive and does not need the critical setting of other extra parameters except for the number of divisions of the objective space.

### **2.2.3.3 Nondominated Sorting Genetic Algorithm II**

The Nondominated Sorting Genetic Algorithm II (NSGA-II) [Deb *et al.* 2002] was proposed by Deb *et al.*, which represents a revised version of the Nondominated Sorting Genetic Algorithm (NSGA) [Srinivas & Deb 1994] and is more efficient than its predecessor. It employs a crowded tournament selection operator to keep diversity. In the elitist mechanism of NSGA-II, it does neither use an external memory nor does it specify any extra niching parameters as most of the other algorithms have to do. Instead, the elitist mechanism consists of combining the

## ***Chapter 2: A Background***

---

best parents with the best offspring obtained. Because of NSGA-II's elitist approach and less parameters needed, NSGA-II has become one of the best multi-objective optimisation algorithms.

There are some other evolutionary multi-objective optimisation algorithms proposed in recent years. More details can be found in the following books and papers [Coello Coello 1999; Deb 2001; Zitzler *et al.* 2002; Coello Coello 2003; Zitzler *et al.* 2003].

### **2.3 Fuzzy Sets and Fuzzy Systems**

#### **2.3.1 Fuzzy Sets**

Fuzzy sets are sets whose elements have degrees of membership. Fuzzy sets have been introduced by Lotfi A. Zadeh [Zadeh 1965] at the University of California, Berkeley, as an extension of the classical notion of a 'set'. In classical set theory, the membership of elements in a set is assessed in binary terms according to a bivalent condition - an element either belongs or does not belong to the set. By contrast, fuzzy set theory permits the gradual assessment of the membership of elements in a set; this is described with the aid of a membership function valued in the real unit interval  $[0, 1]$ . Fuzzy sets generalise classical sets, since the indicator functions of classical sets are special cases of the membership functions of fuzzy sets, if the latter only take values 0 or 1 [Dubois & Prade 1988].

## Chapter 2: A Background

---

A fuzzy set is a pair  $(A, \mu_A)$  where  $A$  is a set and  $\mu_A: A \rightarrow [0, 1]$  is a membership function. For each  $x \in A$ ,  $\mu_A(x)$  is the grade of membership of  $x$ .  $x \in (A, \mu_A) \Leftrightarrow x \in A \wedge \mu_A(x) \neq 0$ . If  $A = \{z_1, \dots, z_n\}$ , the fuzzy set  $(A, \mu_A)$  can also be denoted as  $\{\mu_A(z_1) / z_1, \dots, \mu_A(z_n) / z_n\}$ .

An element mapping to the value '0' means that the member is not included in the fuzzy set, value '1' describes a fully included member. Values strictly between 0 and 1 characterise the fuzzy members.

Generally, the membership functions  $\mu_A(x)$  use Gaussian functions, triangular-shape functions or trapezoidal-shape functions, as shown in Figure 2-1.

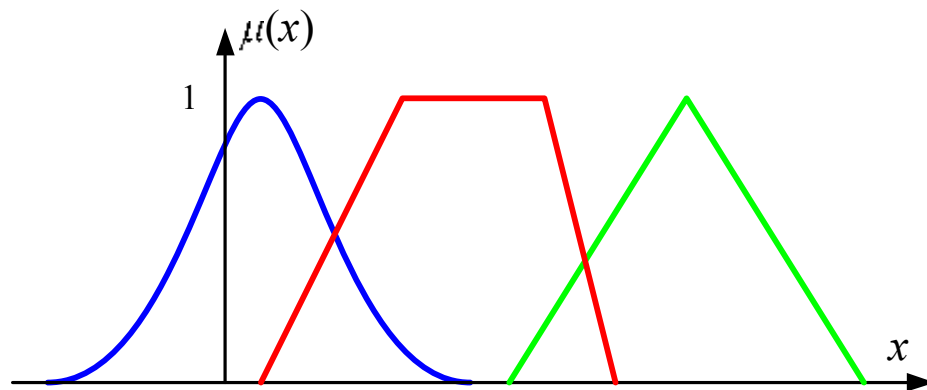


Figure 2-1. An example of membership functions

### 2.3.2 Fuzzy Systems

Fuzzy systems are knowledge-based or rule-based systems. They have been

## Chapter 2: A Background

---

applied to a wide variety of fields ranging from control, signal processing, communications, integrated circuit manufacturing, and expert systems to business, medicine, psychology, etc.

A fuzzy system basically consists of four components: fuzzy rule-base, fuzzy inference engine, fuzzifier and defuzzifier. Figure 2-2 shows the diagram of a fuzzy system.

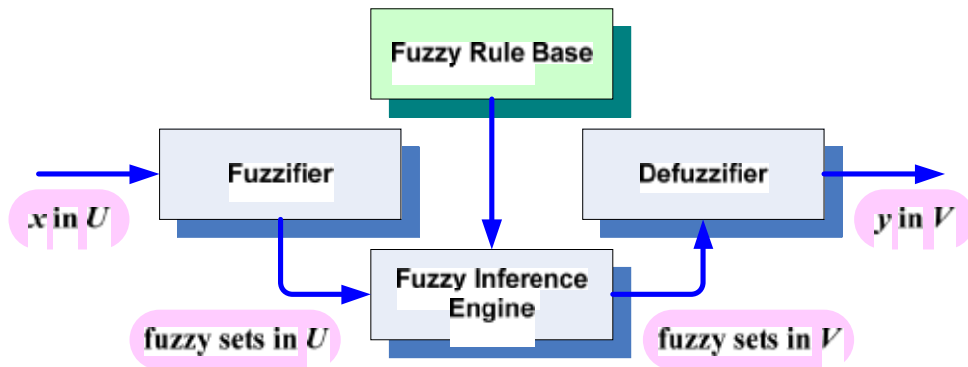


Figure 2-2. The basic configuration of fuzzy systems

The heart of a fuzzy system is the knowledge-base (rule-base) consisting of the so-called fuzzy IF-THEN rules and all other components are used to implement these rules in a reasonable and efficient manner. A fuzzy IF-THEN rule is an IF-THEN statement in which some words are characterised by continuous membership functions. Specifically, the fuzzy rule-base comprises the following fuzzy IF-THEN rules:

Rule $_j$ : **IF**  $x_1$  is  $A_1^l$  and ... and  $x_n$  is  $A_n^l$ , **THEN**  $y$  is  $B^l$ ,

where  $l = 1, 2, \dots, M$  and  $M$  is the number of rules in the fuzzy rule-base;  $A_i^l$  and  $B^l$

## **Chapter 2: A Background**

---

are fuzzy sets in  $U_i \subseteq R$  and  $V \subseteq R$ , respectively, and  $\mathbf{x} = (x_1, x_2, \dots, x_n)^T \in U$  and  $y \in V$  are the input and output (linguistic) variables of the fuzzy system, respectively.

The fuzzifier is defined as a mapping from a real-valued point  $\mathbf{x}^* \in U \subseteq R^n$  to a fuzzy set  $A^*$  in  $U$ . Normally, three types of fuzzifiers are used, which are singleton fuzzifier, Gaussian fuzzifier and triangular fuzzifier [Wang 1997]. The defuzzifier is defined as a mapping from a fuzzy set  $B^*$  in  $V \subseteq R$  (which is the output of the fuzzy inference engine) to a crisp point  $y^* \in V$ . Conceptually, the task of the defuzzifier is to specify a point in  $V$  that best represents the fuzzy set  $B^*$ . Three mostly used defuzzifiers are centre of gravity defuzzifier, centre average defuzzifier and maximum defuzzifier [Wang 1997].

In a fuzzy inference engine, fuzzy logic principles are used to combine the fuzzy IF-THEN rules in the fuzzy rule-base into a mapping from a fuzzy set  $A^*$  in  $U$  to a fuzzy set  $B^*$ . For more information and details about the fuzzy inference engine, please refer to the book [Wang 1997].

### **2.4 Summary**

In this chapter, some basic knowledge relating to this research project was introduced. This includes the basic concepts relating to optimisation, multi-objective optimisation, evolutionary computation, fuzzy sets, and fuzzy systems.



## ***Chapter 2: A Background***

---

Meanwhile, some important single objective and multi-objective optimisation algorithms based on evolutionary computation principles were also introduced, which include Genetic Algorithms, Evolution Strategies, Particle Swarm Optimisation, Strength Pareto Evolutionary Algorithms, Pareto Archived Evolution Strategy, and Nondominated Sorting Genetic Algorithm II. The next chapter presents the development of a new optimisation algorithm in its single objective format as well as its multi-objective extension.

## **Chapter 3**

# **A New Reduced Space Searching Algorithm with Applications to Single Objective and Multi-Objective Problems**

### **3.1 Introduction to the Reduced Space Searching (RSS) Strategy**

Inspired by natural and social behaviours, researchers have developed many successful optimisation algorithms. For example, the Genetic Algorithm (GA) [Holland 1975; Goldberg 1989] originates from the simulation of natural evolution, while the Particle Swarm Optimisation (PSO) [Kennedy and Eberhart 1995; Eberhart and Kennedy 1995] algorithm is motivated by the simulation of the social behaviour of birds flock. In the same way, a new search and optimisation algorithm Reduced Space Searching Algorithm (RSSA) is reported in this chapter, which is inspired by the simple human experience when looking for an ‘optimal’ solution.

### ***Chapter 3: A New Reduced Space Searching Algorithm***

---

If one approaches the optimisation issue from a totally unbiased angle it would be legitimate to postulate that ‘common’ sense should dictate that when searching for a candidate solution under predefined objectives, a relatively large search space area must be initially targeted. When clues are available that the objective may be met in a particular area, the initial search area is then justifiably reduced. This simple principle is being widely used in our every-day life and has proved to be effective. In the light of the above, a strategy of constructing a new optimisation algorithm, named Reduced Space Searching (RSS) throughout, is proposed here.

The ‘rationale’ behind this RSS strategy is as follows: given an optimisation problem, one should divide the initial search space into parts and rank these parts according to the probability of the candidates satisfying the objective(s). First, a search is conducted in the partial space where the probability is the highest followed by the one with the lowest probability. The diagram of Figure 3-1 illustrates the idea behind the RSS strategy.

Following this simple idea, a new optimisation algorithm, Reduced Space Searching Algorithm (RSSA), has been constructed. This algorithm has also been extended to include the multi-objective case. More details about the new algorithm and its experimental performance are given in the remaining parts of this chapter. Section 3.2 outlines the various steps included in the new proposed algorithm. Section 3.3 presents the results of applying the new algorithm to optimise some well-known single objective benchmark functions. A comparative study between RSSA and other three evolutionary algorithms, i.e. the Covariance Matrix

### Chapter 3: A New Reduced Space Searching Algorithm

Adaptation Evolution Strategy (CMA-ES) [Hansen *et al.* 2003], the Differential Evolution (DE) [Storn & Price 1995] and the Generalised Generation Gap model with a Parent-Centric Recombination operator (G3+PCX) [Deb *et al.* 2002], is also conducted. In Section 3.4, the algorithm is extended to solve the multi-objective optimisation problems. In Section 3.5, the proposed multi-objective optimisation algorithm is then compared to other salient multi-objective evolutionary algorithms in solving the well-known ZDT [Zitzler *et al.* 2000] and DTLZ [Deb *et al.* 2001] series test problems. Finally, summary remarks are given in Section 3.6.

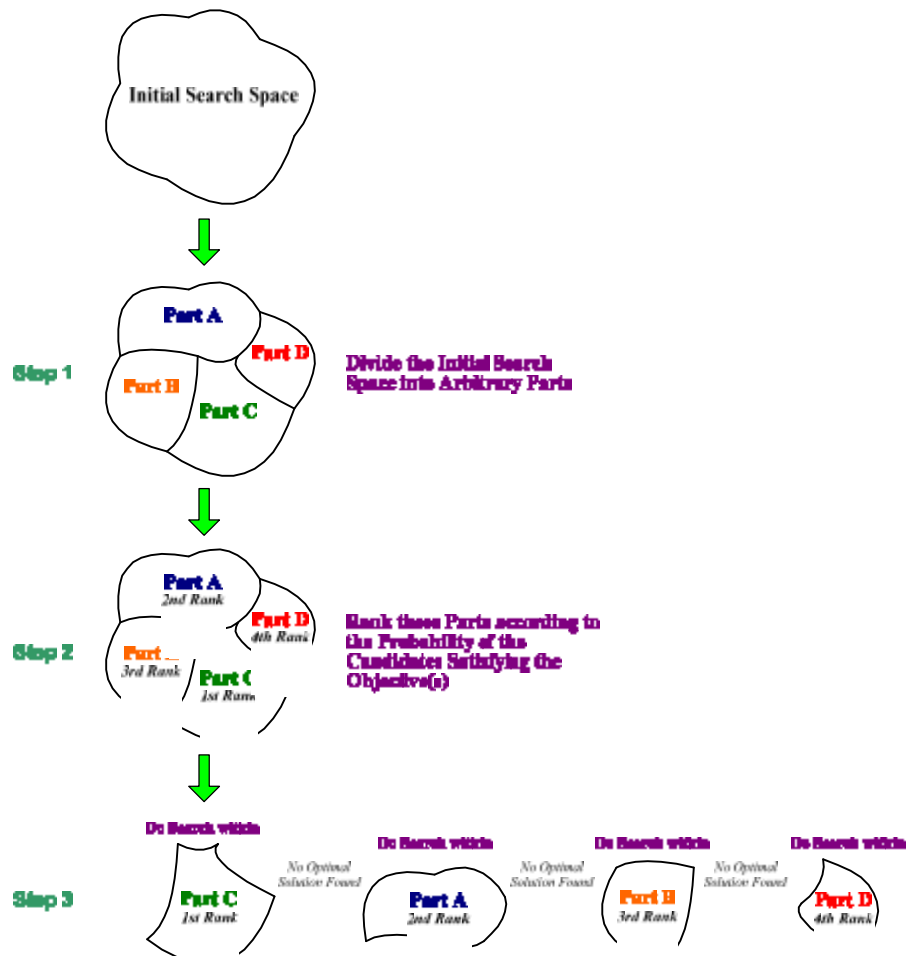


Figure 3-1. The RSS strategy for dealing with optimisation problems

## ***Chapter 3: A New Reduced Space Searching Algorithm***

---

### **3.2 The Reduced Space Searching Algorithm (RSSA)**

#### **3.2.1 The Basic Idea**

To develop an algorithm following the RSS strategy described in the previous section, one must first define how to divide the search space into parts and how to rank them. In this work, a simple (but not simpler) method to achieve this purpose is proposed. The basic idea is that the search space should be divided into two parts: one part is located around the best solution found so far and the other part is the space left. The partial space around the best solution should be ranked highly. To simplify the method, the marginal partial space can be neglected and only the space that includes the best solution is kept for search purposes. If the process of dividing the search space into smaller parts is repeated sufficiently enough, then a final relatively small search space as well as an ‘optimal’ solution to the problem will be obtained.

#### **3.2.2 Improvement of the Basic Idea**

It was found that reducing the search space all the time is not the most effective way of locating the optimal solution. Sometimes, a too-small search space will decrease the speed of solution convergence and at the same time will reduce the probability of the solution jumping out of the local optima. Thus, a search space ‘increase’ mechanism is proposed to cooperate with the original ‘decrease’

### ***Chapter 3: A New Reduced Space Searching Algorithm***

---

mechanism. In this new mechanism, if no better solution can be found in the optimisation search process, then the search space is reckoned to be too large for such a search and should be decreased to reinforce the local search. If better solutions can always be found in a particular reduced space, then the algorithm may certainly have got trapped in a local optimal area. Given this situation, the search space should be increased to reinforce the global search. This proposed method attempts to strike a balance between the global and the local searches to make the optimisation search process more adaptive.

#### **3.2.3 An Example of the Search Space Decreasing or Increasing**

Figure 3-2 shows an example of the size of the search space decreasing or increasing in a two-dimensional problem. In Figure 3-2(a), the rectangular Region 1 is the search space of an optimisation problem. Solution 'A' is the best solution hitherto located. If there are several continuous randomly selected candidate solutions worse than 'A' in the fitness to the optimisation problem, as is shown in Figure 3-2(b), then the size of the search space should be decreased around the best solution 'A'. The partial space (Region 2) containing 'A', as the centre, is set to the new space one should perform the search in. On the other hand, if there are several continuous randomly selected candidate solutions better than 'A' in the fitness, which is shown in Figure 3-2(c), then the size of the search space should be increased around the best solution 'B'. The increasing space (Region 3) containing 'B', as the centre, is set to the new space one should perform the search

### Chapter 3: A New Reduced Space Searching Algorithm

in. If there are better solutions (but not continuous) that can be found in the search space (shown in Figure 3-2(d)), then the size of the search space should not be changed and the centre of the search space (Region 4) will be moved to the new best solution 'C'.

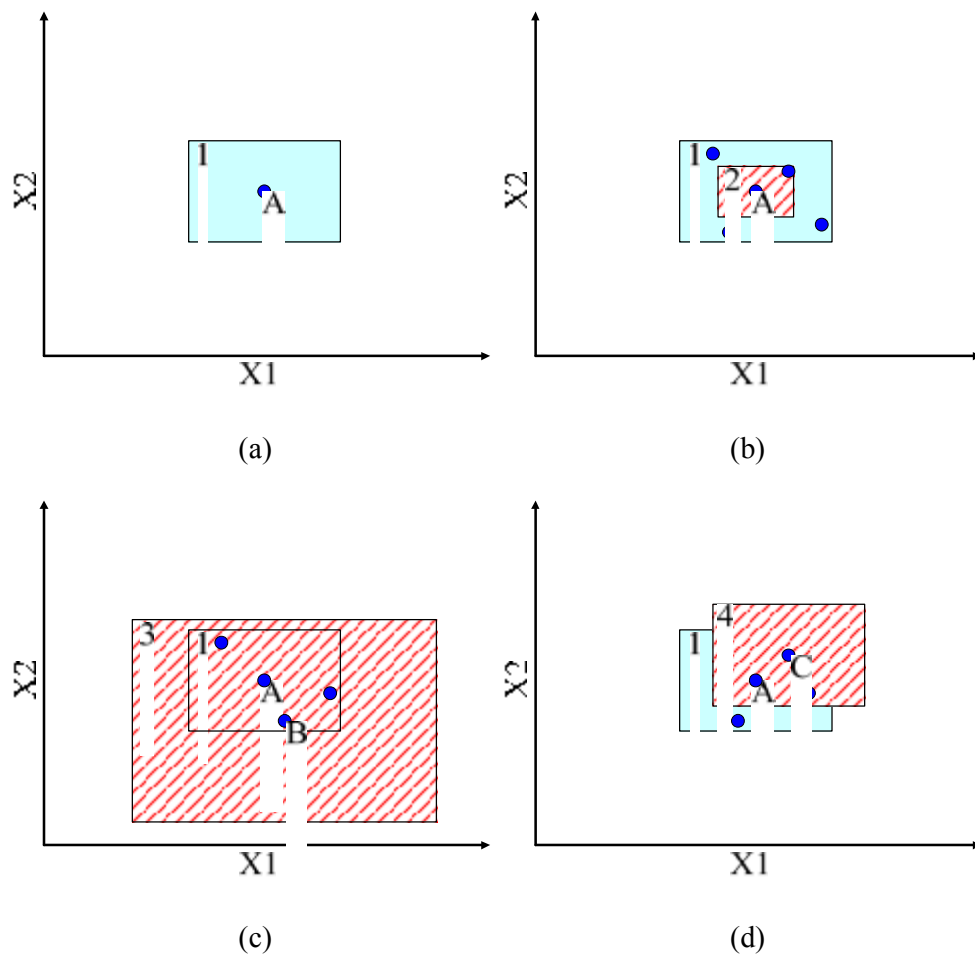


Figure 3-2. An example of how to divide the search space in the case a two-dimensional problem

In the light of the above considerations, one will obtain a good solution after a finite number of repetitions. However, it must be stressed that the above method

### Chapter 3: A New Reduced Space Searching Algorithm

may only find a local optimal solution rather than a global optimal one. To obtain a global optimal solution, a variation operator is employed to cooperate with the RSS operator. Figure 3-3 shows the flow chart of the whole RSSA algorithm.

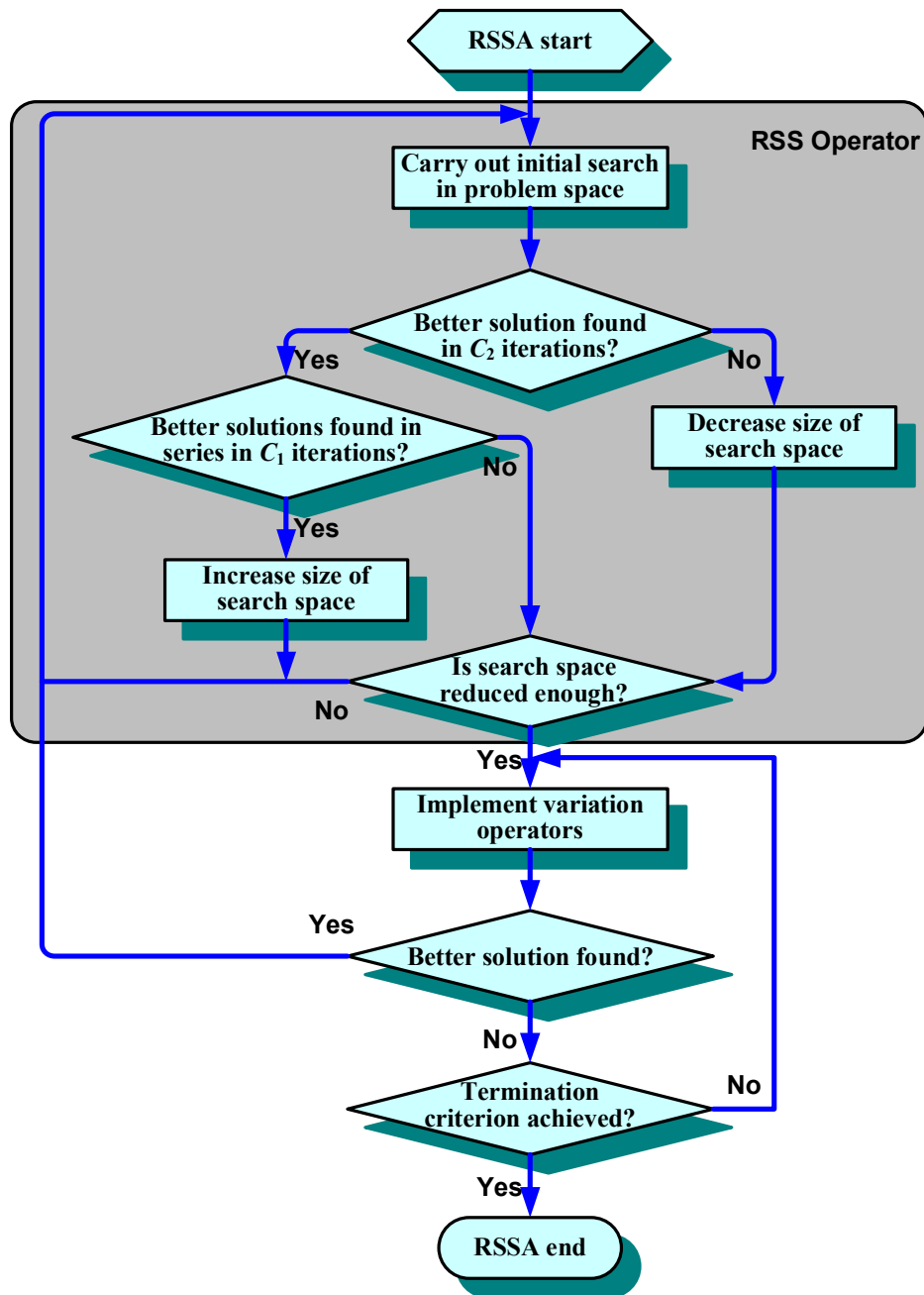


Figure 3-3. Flow chart of the RSSA algorithm



### ***Chapter 3: A New Reduced Space Searching Algorithm***

---

#### **3.2.4 The Variation Operators**

Three variation strategies are designed as follows:

1. **One-dimensional variation:** Only one element of the decision variable vector will be varied. The position of this element will be randomly chosen and the element will be set at a random value within the search bounds.
2. **Multi-dimensional variation:** The number of elements of the decision variable that will be varied and the positions of these elements will be randomly generated. These elements will then be set to some random values.
3. **All-dimensional variation:** All the elements of the decision variable vector will be randomly varied.

#### **3.2.5 The RSSA Algorithm**

Consider a single objective optimisation problem with  $N$  decision variables as follows: Minimise  $f(X)$ ,  $X \in [X_{\min_1}, X_{\max_1}] \times [X_{\min_2}, X_{\max_2}] \times \dots \times [X_{\min_N}, X_{\max_N}]$ .

The proposed RSSA algorithm can be summarised as follows:

### **Chapter 3: A New Reduced Space Searching Algorithm**

1. Randomly select one candidate solution  $Xa (x_1, x_2, \dots, x_N)$  in the original search space and save it as the best solution  $Xbest = Xa$ . Set  $n = 0$ , which is used to control the bounds of the search space.
  
2. Randomly select the candidate solutions  $Xb(s)$  in the current search space. If  $C_1$ -continuous  $Xb(s)$  satisfies  $f(Xb) < f(Xbest)$  and  $n > 1$ , then  $Xbest = Xb$  and  $n = n - 1$ . If  $C_2$ -continuous  $Xb(s)$  satisfies  $f(Xb) > f(Xbest)$ , then  $n = n + 1$ . If non-continuous  $Xb(s)$  satisfies  $f(Xb) < f(Xbest)$ , then  $Xbest = Xb$ .
  
3. Change the size of the search space using the ratio  $K$  ( $0 < K < 1$ , in this thesis  $K = 0.5$  without any loss of generality).  $Xbest$  is located at the centre of the new space.  $Ymin_i$  is the lower bound of the  $i$ th decision variable in the new search space and  $Ymax_i$  is the upper bound. To avoid the new bounds stepping outside the original constraints, the following equations are used to define the new bounds:

$$\begin{aligned}
 Ymin_i &= \max\{Xmin_i, Xbest(i) - K^n L(i)\}, \\
 Ymax_i &= \min\{Xmax_i, Xbest(i) + K^n L(i)\}.
 \end{aligned} \tag{3.1}$$

where  $i = 1, 2, \dots, N$ ;  $0 < n < m$ ;  $L(i) = Xmax_i - Xmin_i$ .  $m$  is a threshold value that depends on the precision needed and relates to the value of  $K$ . If  $K = 0.5$ , a value of  $m = 15$  to  $30$  should prove adequate.

4. Repeat Steps 2 and 3 until  $n = m$ .
  
5. Perform the variation operator on  $Xbest$  and obtain  $Xc$ . If  $f(Xc) < f(Xbest)$ ,

### ***Chapter 3: A New Reduced Space Searching Algorithm***

---

then  $X_{best} = X_c$ ,  $n = 0$  and repeat Steps 2 to 4. If  $C_3$ -continuous  $X_c(s)$  satisfies  $f(X_c) > f(X_{best})$ , then  $n = 0$  and repeat Steps 2 to 4.

6. Repeat Step 5 until the ‘optimal’ solution is found or the termination criterion is reached.

It is worth noting that the decreasing parameter  $C_1$  and the increasing parameter  $C_2$  play important roles in the RSSA algorithm. They are used to balance the global search as well as the local search in the optimisation process.

## **3.3 Experimental Studies using RSSA**

### **3.3.1 Benchmark Test Functions**

In the field of evolutionary computation, it is common to compare different algorithms using a large test set. When an algorithm is evaluated, one must look for the type of problems where its performance is good, in order to characterise the type of problems for which the algorithm is suitable. In this work, the test set with some well-characterised functions is used as it allows one to obtain and generalise the results regarding the kind of functions involved. All these functions are used as minimisation problems and the following shows their expressions and the summary of their features about separability and multimodality. Figures 3-4(a) to 3-4(n) show the 3-dimensional maps of these functions in the 2-dimensional case.

### **Chapter 3: A New Reduced Space Searching Algorithm**

---

1. **Sphere function** (Unimodal, Separable and  $D$ -dimensional):

$$f_1(x) = \sum_{i=1}^D x_i^2, x_i \in [-10, 10], \min(f_1) = f_1(0, \dots, 0) = 0.$$

2. **Schwefel's function 2.22** (Unimodal, Non-separable and  $D$ -dimensional):

$$f_2(x) = -\sum_{i=1}^D |x_i| + \prod_{i=1}^D |x_i|, x_i \in [-10, 10], \min(f_2) = f_2(0, \dots, 0) = 0.$$

3. **Schwefel's function 1.2** (Unimodal, Non-separable and  $D$ -dimensional):

$$f_3(x) = \sum_{i=1}^D \left( \sum_{j=1}^i x_j \right)^2, x_i \in [-10, 10], \min(f_3) = f_3(0, \dots, 0) = 0.$$

4. **Schwefel's function 2.21** (Unimodal, Non-separable and  $D$ -dimensional):

$$f_4(x) = \max_i \{ |x_i|, 1 \leq i \leq D \}, x_i \in [-10, 10], \min(f_4) = f_4(0, \dots, 0) = 0.$$

5. **Rosenbrock's function** (Multimodal, Non-separable and  $D$ -dimensional):

$$f_5(x) = \sum_{i=1}^{D-1} \left( 100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2 \right), x_i \in [-2, 2],$$

$$\min(f_5) = f_5(1, \dots, 1) = 0.$$

6. **Schwefel's function 2.26** (Multimodal, Separable and  $D$ -dimensional):

$$f_6(x) = -\sum_{i=1}^D \left( x_i \sin(\sqrt{|x_i|}) \right), x_i \in [-500, 500],$$

$$\min(f_6) = f_6(420.9687, \dots, 420.9687) = -12569.5.$$

### Chapter 3: A New Reduced Space Searching Algorithm

---

7. **Rastrigin's function** (Multimodal, Separable and  $D$ -dimensional):

$$f_7(x) = \sum_{i=1}^D (x_i^2 - 10 \cos(2\pi x_i) + 10), \quad x_i \in [-5, 5], \quad \min(f_7) = f_7(0, \dots, 0) = 0.$$

8. **Ackley's function** (Multimodal, Non-separable and  $D$ -dimensional):

$$f_8(x) = -20 \exp \left( -0.2 \sqrt{\frac{1}{D} \sum_{i=1}^D x_i^2} \right) - \exp \left( \frac{1}{D} \sum_{i=1}^D \cos(2\pi x_i) \right) + 20 - e,$$

$$x_i \in [-30, 30], \quad \min(f_8) = f_8(0, \dots, 0) = 0.$$

9. **Griewank's function** (Multimodal, Non-separable and  $D$ -dimensional):

$$f_9(x) = \frac{1}{4000} \sum_{i=1}^D x_i^2 - \prod_{i=1}^D \cos \left( \frac{x_i}{\sqrt{i}} \right) + 1, \quad x_i \in [-600, 600],$$

$$\min(f_9) = f_9(0, \dots, 0) = 0.$$

10. **Bohachevsky's function** (Multimodal, Separable and  $D$ -dimensional):

$$f_{10}(x) = \sum_{i=1}^{D-1} (x_i^2 - 2x_{i+1}^2 - 0.3 \cos(3\pi x_i) - 0.4 \cos(4\pi x_{i+1}) + 0.7),$$

$$x_i \in [-15, 15], \quad \min(f_{10}) = f_{10}(0, \dots, 0) = 0.$$

11. **Schaffer's function** (Multimodal, Non-separable and  $D$ -dimensional):

$$f_{11}(x) = \sum_{i=1}^{D-1} (x_i^2 + x_{i+1}^2)^{0.25} (\sin^2(50(x_i^2 + x_{i+1}^2)^{0.1}) - 1.0), \quad x_i \in [-100, 100],$$

$$\min(f_{11}) = f_{11}(0, \dots, 0) = 0.$$

### Chapter 3: A New Reduced Space Searching Algorithm

---

12. **Six-hump Camel-Back function** (Multimodal, Non-separable and 2-dimensional):

$$f_{12}(x) = 4x_1^2 - 2.1x_1^4 - \frac{1}{3}x_1^6 + x_1x_2 - 4x_2^2 + 4x_2^4, \quad x_i \in [-5, 5],$$

$$\min(f_{12}) \quad f_{12}(0.08983, -0.7126) \quad f_{12}(-0.08983, 0.7126) \quad 0.$$

13. **Branin function** (Multimodal, Non-separable and 2-dimensional):

$$f_{13}(x) = \left( x_2 - \frac{5.1}{4\pi^2}x_1^2 - \frac{5}{\pi}x_1 - 6 \right)^2 + 10 \left( 1 - \frac{1}{8\pi} \right) \cos x_1 + 10,$$

$$x_1 \in [-\pi, \pi], \quad x_2 \in [0, 15], \quad \min(f_{13}) \quad f_{13}(-3.142, 12.275)$$

$$f_{13}(3.142, 2.275) \quad f_{13}(9.425, 2.425) \quad 0.398.$$

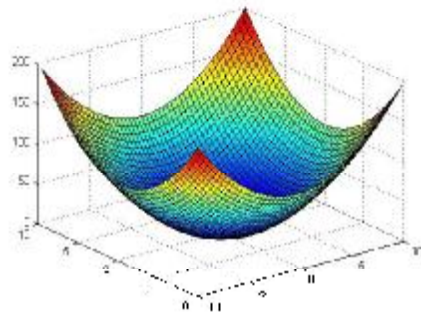
14. **Goldstein-Price function** (Multimodal, Non-separable and 2-dimensional):

$$f_{14}(x) = \left( 1 + (x_1 - x_2 - 1)^2 (19 - 14x_1 - 3x_1^2 - 14x_2 - 6x_1x_2 + 3x_2^2) \right) \\ \times \left( 30 - (2x_1 - 3x_2)^2 (18 - 32x_1 + 12x_1^2 + 48x_2 - 36x_1x_2 + 27x_2^2) \right),$$

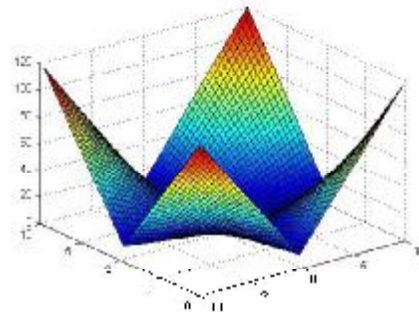
$$x_i \in [-2, 2], \quad \min(f_{14}) \quad f_{14}(0, -1) \quad 3.$$

**Chapter 3: A New Reduced Space Searching Algorithm**

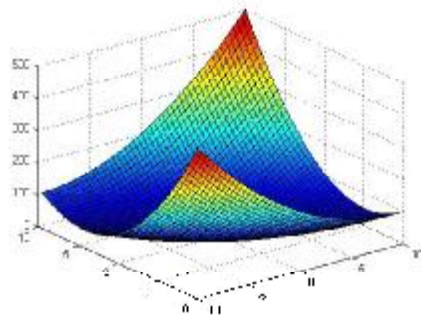
---



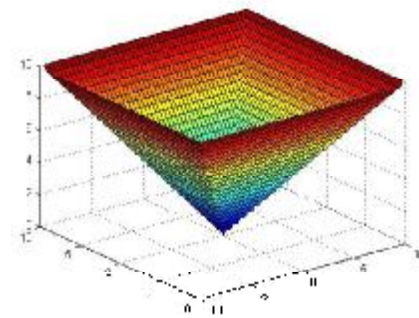
(a)



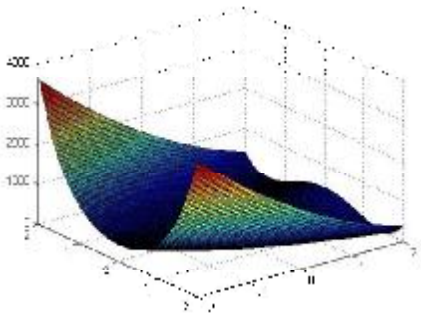
(b)



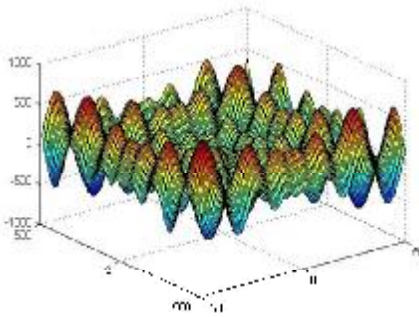
(c)



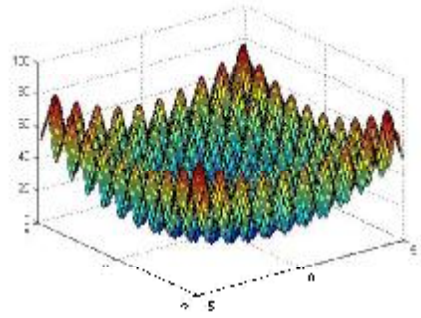
(d)



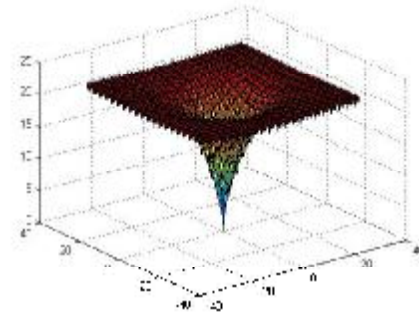
(e)



(f)



(g)



(h)

### Chapter 3: A New Reduced Space Searching Algorithm

---

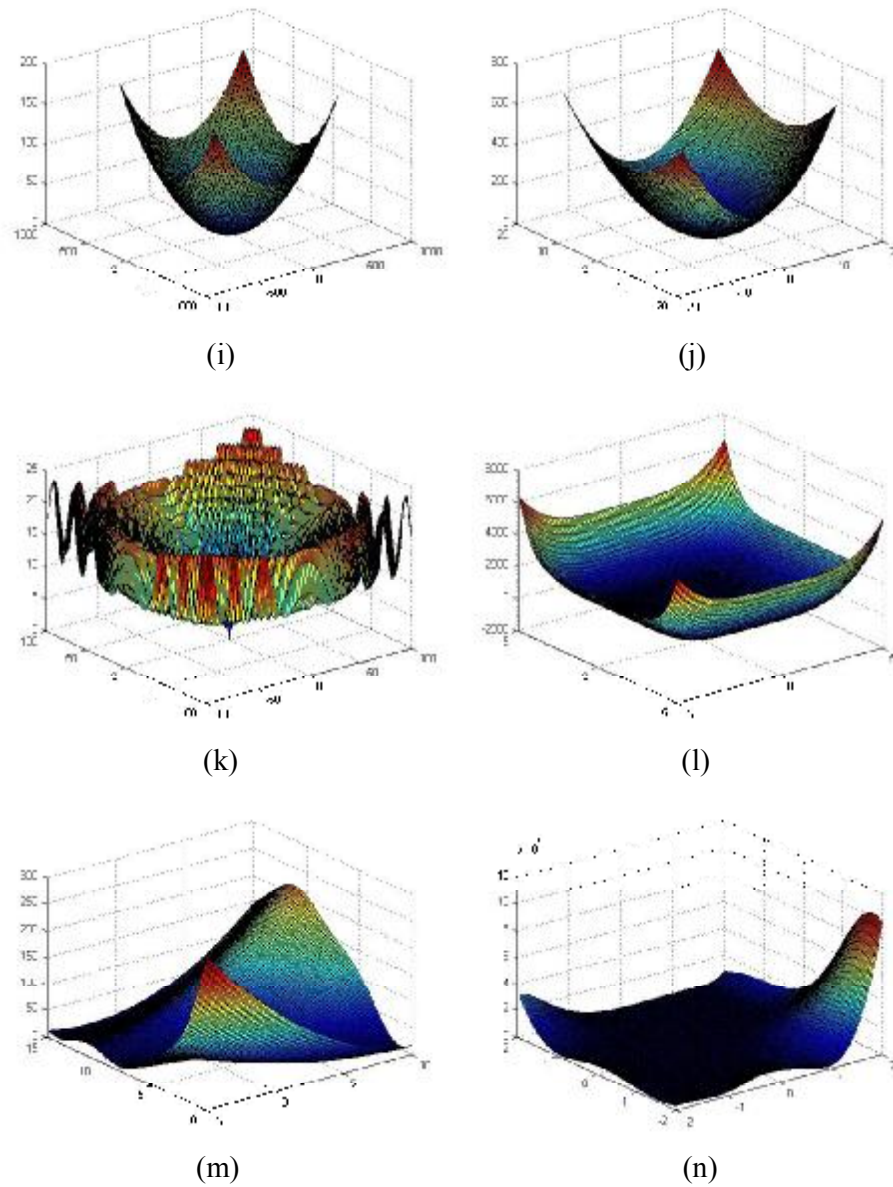


Figure 3-4. The 3-dimensional map for  $f_1$  to  $f_{14}$  in the 2-dimensional situation: (a) to (n)

A function of  $D$  variables is separable if it can be rewritten as a sum of  $D$  functions of just one variable. Non-separable functions are more difficult to optimise as the accurate search direction depends on two or more variables. On the other hand, separable functions can be optimised for each variable in turn. A function is



### **Chapter 3: A New Reduced Space Searching Algorithm**

multimodal if it has two or more local optima. The problem is more difficult if the function is multimodal. The search process must be able to avoid the regions around local optima in order to approximate, as far as possible, the global optimum.

#### **3.3.2 Effect of the Decreasing Parameter ‘ $C_1$ ’**

In this experiment, the RSSA algorithm was tested using the benchmark problems  $f_1, f_3, f_7$  and  $f_{10}$  with the settings of a constant increasing parameter ( $C_2 = 1$ ) and the various decreasing parameter ( $C_1 = 1, 2, 5, 10, 20$  and  $50$ ). For each benchmark problem, the dimension  $D$  was increased in the sequence to 2, 5, 10, 20 and 50, and for each setting, 20 runs were conducted. In each run, the maximal function evaluation number was set to  $10^6$  and the optimisation process was regarded as successful and stopped, when the best solution  $Fb$  satisfied the following condition:  $Fb < 10^{-5}$  if the true global minimum  $Gb = 0$  or  $|(Fb - Gb) / Gb| < 10^{-5}$  if  $Gb \neq 0$ .

In this case, the parameter  $m = 15$  and both the one-dimensional variation strategy (with the 50% probability of usage) and the multi-dimensional variation strategy (with the 50% probability of usage) were used.

Table 3-1 shows the performance of RSSA with various decreasing parameters in optimising the test problems with different dimensions. From this table, one can observe that, for one optimisation problem, with the increasing number of

### Chapter 3: A New Reduced Space Searching Algorithm

dimensions in the problem, the algorithm needs a larger decreasing parameter. As a result it is recommended that  $C_1 = C_2 \times (D/2 + 8)$ , where  $D$  is the dimension of the test problem.

Table 3-1. The average number of function evaluations to find the optimum; the bold values represent the best results.

Function	Dimension	$C_1=1$	$C_1=2$	$C_1=5$	$C_1=10$	$C_1=20$	$C_1=50$
$f_1$	$D=2$	4603	2838	<b>83</b>	161	262	732
	$D=5$	29289	25110	2094	<b>259</b>	468	1080
	$D=10$	145287	97843	13553	<b>466</b>	684	1538
	$D=20$	389010	293490	62555	<b>905</b>	1286	2246
	$D=50$	N/A	N/A	260492	13677	<b>2694</b>	4602
$f_3$	$D=2$	17074	2486	189	<b>164</b>	316	731
	$D=5$	388700	230640	40265	902	<b>569</b>	1096
	$D=10$	N/A	N/A	614729	98663	<b>1985</b>	2165
	$D=20$	N/A	N/A	N/A	N/A	10810	<b>6601</b>
	$D=50$	N/A	N/A	N/A	N/A	N/A	<b>44682</b>
$f_7$	$D=2$	26730	12690	<b>678</b>	862	1337	2000
	$D=5$	258280	91105	16135	<b>4456</b>	5368	10988
	$D=10$	821180	453360	99204	<b>14335</b>	19743	31215
	$D=20$	N/A	N/A	397555	280120	<b>37054</b>	92932
	$D=50$	N/A	N/A	908190	311700	<b>226977</b>	349717
$f_{10}$	$D=2$	30562	11527	<b>320</b>	418	571	1163
	$D=5$	398277	238140	71750	<b>1366</b>	2513	4501
	$D=10$	N/A	849800	140130	10785	<b>9058</b>	13389
	$D=20$	N/A	N/A	468950	43818	<b>30624</b>	34431
	$D=50$	N/A	N/A	N/A	280172	<b>126854</b>	152310

### ***Chapter 3: A New Reduced Space Searching Algorithm***

---

#### **3.3.3 Effects of the Variation Strategies**

Three types of variation operators were tested and compared in this experiment. For this purpose, the 30-dimensional multimodal benchmark problems  $f_5$  to  $f_{11}$  were used as test beds. The decreasing parameter  $C_1$  was set to be 23 and the increasing parameter  $C_2$  was set to be 1. Other settings were similar to those of the experiments conducted in Section 3.3.2.

From Table 3-2, it can be seen that the one-dimensional variation strategy performs best on the functions  $f_5, f_6, f_7, f_8, f_{10}$  and  $f_{11}$ , while the all-dimensional variation strategy performs best on the problems  $f_9$ . For a broad adaptation to various problems, it is recommended to use both the one-dimensional and multi-dimensional variation strategies simultaneously.

#### **3.3.4 A Comparison between RSSA and Other Evolutionary Algorithms**

In this section, experiments were carried-out between RSSA and other three salient evolutionary algorithms, which are the Covariance Matrix Adaptation Evolution Strategy (CMA-ES) [Hansen *et al.* 2003], the Differential Evolution (DE) [Storn & Price 1995] and the Generalised Generation Gap model with the Parent-Centric Recombination operator (G3+PCX) [Deb *et al.* 2002].

### Chapter 3: A New Reduced Space Searching Algorithm

Table 3-2. Average performance of RSSA with different variation strategies in optimising  $f_5$  to  $f_{11}$ : The integer in every cell is the average function evaluation number in successful runs; the value between the parentheses is the average result in the unsuccessful runs; the percentage value in the square brackets indicates the percentage of the successful runs out of all the runs; the bold values represent the best results.

Function	One-dimensional Variation	Multi-dimensional Variation	Multi-dimensional Variation
$f_5$	N/A <b>(0.0017)</b> [0%]	N/A (0.8025) [0%]	N/A (0.0043) [0%]
$f_6$	<b>76645</b> (N/A) [100%]	108630 (N/A) [100%]	N/A (-7712) [0%]
$f_7$	<b>88647</b> (N/A) [100%]	318860 (N/A) [100%]	N/A (1.7491e+2) [0%]
$f_8$	N/A <b>(1.2877e-5)</b> [0%]	N/A (1.7127e-5) [0%]	N/A (1.3796e+1) [0%]
$f_9$	2788 (0.0193) [40%]	2935 (0.0158) [50%]	<b>2753</b> <b>(0.0108)</b> [55%]
$f_{10}$	<b>46969</b> (N/A) [100%]	312660 (N/A) [100%]	N/A (1.7105e+1) [0%]
$f_{11}$	N/A <b>(1.7736e+2)</b> [0%]	N/A (1.8662e+2) [0%]	N/A (2.0294e+2) [0%]

The parameter settings for these algorithms are described as follows:

1. **RSSA:**  $C_1 = D/2 + 8$ ,  $C_2 = 1$ ,  $K = 0.5$ ,  $m = 30$ , where  $D$  is the dimension of the test problem. The variation operator worked as a combination of the

### ***Chapter 3: A New Reduced Space Searching Algorithm***

---

one-dimensional variation strategy (with the 50% probability of usage) and the multi-dimensional variation strategy (with the 50% probability of usage).

2. **CMA-ES**: There are 8 parameters to be predefined for this algorithm. All settings followed the instructions given in [Hansen 2007]. For instance, the population size  $\lambda = 4 + \text{floor}(3 \times \ln D)$ , the parent number  $\mu = \text{floor}(\lambda/2)$ , etc., where  $\text{floor}(x)$  is the function that allows to round-off  $x$  to the nearest integer towards  $-\infty$ .
3. **DE**: The DE/Rand/1 scheme was employed. The parameter settings followed the instructions in [Storn 1996]. The population size  $N = 10 \times D$ ; the crossover probability  $CR = 0.9$  and the weighting factor  $F = 0.8$ .
4. **G3+PCX**: Following the papers by [Deb *et al.* 2002; Deb 2005], the population size  $N = 10 \times D$ ; the parent size was set to 3; the offspring size was set to 2 and the replacement size was set to 2. For the PCX operator, the distribution parameter  $\sigma_c = 0.1$  and  $\sigma_\sigma = 0.1$ .

The optimisation process was regarded as successful and stopped when the best solution  $Fb$  satisfied the following condition:  $Fb < 10^{-5}$  if the true global minimum  $Gb = 0$  or  $|(Fb - Gb) / Gb| < 10^{-5}$  if  $Gb \neq 0$ . For every individual experiment, the result was obtained after 20 runs. For each run, the maximal function evaluation number was set to  $10^6$ .

### ***Chapter 3: A New Reduced Space Searching Algorithm***

---

Table 3-3 shows the optimisation results of different algorithms on various problems. From this table, one can observe the following:

1. For the unimodal problems  $f_1$  to  $f_4$ , CMA-ES performs best in most of the situations. RSSA performs best using the fewest function evaluation for  $f_1$ . For  $f_3$ , RSSA can achieve the minimum with a small function evaluation number, but it cannot obtain the optima of the problems  $f_2$  and  $f_4$ .
2. For the high-dimensional multimodal problems  $f_5$  to  $f_{11}$ , RSSA performs better than other algorithms. For instance, for  $f_7, f_8$  and  $f_{10}$ , RSSA is able to locate the global optimum with the fewest function evaluations; for  $f_6$  and  $f_9$ , RSSA performs better than the other algorithms. In most of the situations, RSSA can achieve the optima, while other algorithms often cannot find the ‘true’ optimal solutions.
3. For the low-dimensional multimodal problems  $f_{12}$  to  $f_{14}$ , RSSA is able to obtain the global optimum and needs fewer function evaluations, compared with other algorithms.

### Chapter 3: A New Reduced Space Searching Algorithm

Table 3-3. Average performance of various algorithms in optimising  $f_1$  to  $f_{14}$ : The integer in every cell is the average function evaluation number in successful runs; the value between parentheses is the average result in the unsuccessful runs; the percentage value in the square brackets indicates the percentage of the successful runs out of all the runs; the bold values represent the best results.

Function	RSSA	CMA-ES	DE	G3+PCX
$f_1$	<b>1806</b> (N/A) [100%]	3207 (N/A) [100%]	391770 (N/A) [100%]	7140 (N/A) [100%]
$f_2$	N/A (0.0038) [0%]	<b>11751</b> (N/A) [100%]	655110 (N/A) [100%]	N/A (12.0469) [0%]
$f_3$	24287 (N/A) [100%]	<b>10830</b> (N/A) [100%]	N/A (1.8527) [0%]	25937 (N/A) [100%]
$f_4$	N/A (0.0147) [0%]	<b>8929</b> (N/A) [100%]	N/A (0.2004) [0%]	117414 (N/A) [100%]
$f_5$	N/A (0.0074) [0%]	<b>46072</b> (N/A) [100%]	N/A (0.0158) [0%]	140430 (N/A) [100%]
$f_6$	<b>73451</b> (N/A) [100%]	N/A (-6665) [0%]	616080 (N/A) [100%]	N/A (-6878) [0%]
$f_7$	<b>94499</b> (N/A) [100%]	N/A (106.1617) [0%]	940560 (N/A) [100%]	N/A (142.8754) [0%]
$f_8$	<b>209440</b> (N/A) [100%]	8575 (19.3625) [40%]	694560 (N/A) [100%]	N/A; (3.1199) [0%]
$f_9$	<b>2717</b> ( <b>0.0112</b> ) [50%]	5586 (0.0100) [75%]	586740 (N/A) [100%]	10983 (0.0110) [65%]
$f_{10}$	<b>52774</b> (N/A) [100%]	N/A (2.2897) [0%]	510180 (N/A) [100%]	N/A (15.1530) [0%]
$f_{11}$	N/A (197.8232) [0%]	N/A (248.84) [0%]	N/A ( <b>0.1217</b> ) [0%]	N/A (184.4355) [0%]
$f_{12}$	<b>329</b> (N/A) [100%]	221 (-19.8160) [95%]	853 (N/A) [100%]	N/A (-0.4128) [0%]
$f_{13}$	322 (N/A) [100%]	<b>224</b> (N/A) [100%]	1182 (N/A) [100%]	N/A (0.8862) [0%]
$f_{14}$	<b>366</b> (N/A) [100%]	253 (141.0000) [95%]	777 (N/A) [100%]	N/A (35.3369) [0%]

### ***Chapter 3: A New Reduced Space Searching Algorithm***

---

#### **3.4 Extension of RSSA to Multi-Objective Optimisation Problems**

The RSSA algorithm proposed in this work seems an ideal candidate for multi-objective optimisation because of the good performance ascertained in the single objective optimisation case. To extend the RSSA to optimise multi-objective problems, the Random Weighted Aggregation (RWA) technique [Murata *et al.* 1996] is employed and an archive [Knowles & Corne 2000] is also included to preserve the Pareto-optimal solutions.

##### **3.4.1 The Random Weighted Aggregation Approach**

Assume a multi-objective problem that consists of finding a vector

$$X^* = (x_1^*, x_2^*, x_3^*, \dots, x_D^*) \quad (3.2)$$

that will optimise the following vector function:

$$\vec{f}(X) = [f_1(X), f_2(X), f_3(X), \dots, f_k(X)]. \quad (3.3)$$

The Weighted Aggregation is one of the most common approaches for solving multi-objective problems. In this type of approach, all the objectives are summed to a weighted combination as follows:

$$F = \sum_{i=1}^k w_i f_i(X), \quad \sum_{i=1}^k w_i = 1, \quad (3.4)$$

where  $w_i, i = 1, 2, \dots, k$ , are non-negative weights.



### ***Chapter 3: A New Reduced Space Searching Algorithm***

---

In the Conventional Weighted Aggregation (CWA) method, the above weights are fixed during the optimisation process. By using CWA, only a single Pareto-optimal solution can be obtained in every optimisation run. If one wishes to obtain different Pareto solutions, the algorithm has to be repeated several times with different weights settings. In addition, this method cannot locate the Pareto solutions when there are concave regions in the true Pareto front.

Random Weighted Aggregation (RWA) can overcome the limitations of CWA. In the RWA method, the weights are modified after every certain number of iterations during the optimisation. The weights are defined by the following equation:

$$w_i(t) = \begin{cases} \frac{rand_i(t)}{\sum_{j=1}^k rand_j(t)}, & \text{if } rem(t, H) = 1; \\ w_i(t-1), & \text{else.} \end{cases} \quad (3.5)$$

where  $t$  is the index of iteration and  $H$  is the frequency of the weight changing;  $rand_i(t)$  is a function to create a uniformly distributed random value in the range  $[0, 1]$ ;  $rem(t, H)$  is a function to obtain the remainder from dividing  $t$  by  $H$ .

#### **3.4.2 Archive Design**

In the RWA method, the population cannot keep all the found Pareto solutions. Thus, an archive is used to record the Pareto solutions found so far during the optimisation search. The pseudo-code for maintaining the archive is listed in

### ***Chapter 3: A New Reduced Space Searching Algorithm***

---

Figure 3-5. To update the archive with appropriate Pareto solutions, a non-dominated selection and a diversity selection mechanism are employed. The non-dominated selection aims to obtain the Pareto solutions from the candidates. This is easy to implement. The diversity selection tends to obtain the solutions with a good diversity from the candidates. In this section, a simple method named ‘cell selection’ is proposed to achieve this purpose.

```
For every iteration
    Add new solutions to the archive
    Non-dominated selection
    Diversity selection
End For
```

Figure 3-5. Pseudo-code for maintaining the archive

The cell selection method works as follows:

1. Divide the objective space, which includes the candidate solutions, into equal parts; every part is a ‘cell’. For example, in a two-objective problem, if one divides the objective space into  $g \times g$  grids, then  $g^2$  cells are formed.
2. For every candidate solution, identify the cell that contains this solution.
3. One cell is only permitted to involve one solution. If more than one solution are located in the same cell, the redundant solutions will be

### ***Chapter 3: A New Reduced Space Searching Algorithm***

---

randomly chosen and deleted.

This method can also restrict the number of solutions in the archive. For example, in a two-objective problem, if the cells in this method are formed through the division of a  $g \times g$  grid, then it can be seen that after the non-dominated selection and the cell selection,  $2g$  solutions, at most, will remain.

Figure 3-6 illustrates an example of how the selection methods work in the archive maintaining process. The ‘blue’ points in the figures represent the solutions contained in the archive. Figure 3-6(a) shows the situation after some newly generated solutions are added to the archive. In Figure 3-6(b), non-dominated (Pareto) solutions, which are represented as the red-circled points, are selected. As a result, these non-dominated solutions are selected again, using the cell selection approach, which is shown as Figure 3-6(c). Finally, the Pareto-optimal solutions with good diversity are obtained (see Figure 3-6(d)).

#### **3.4.3 The Multi-Objective Reduced Space Searching Algorithm (MO-RSSA)**

After applying the RWA method and maintaining an archive for preserving the Pareto-optimal solutions, the RSSA is now extended to the multi-objective optimisation case, which leads to the algorithm named throughout as Multi-Objective Reduced Space Searching Algorithm (MO-RSSA).

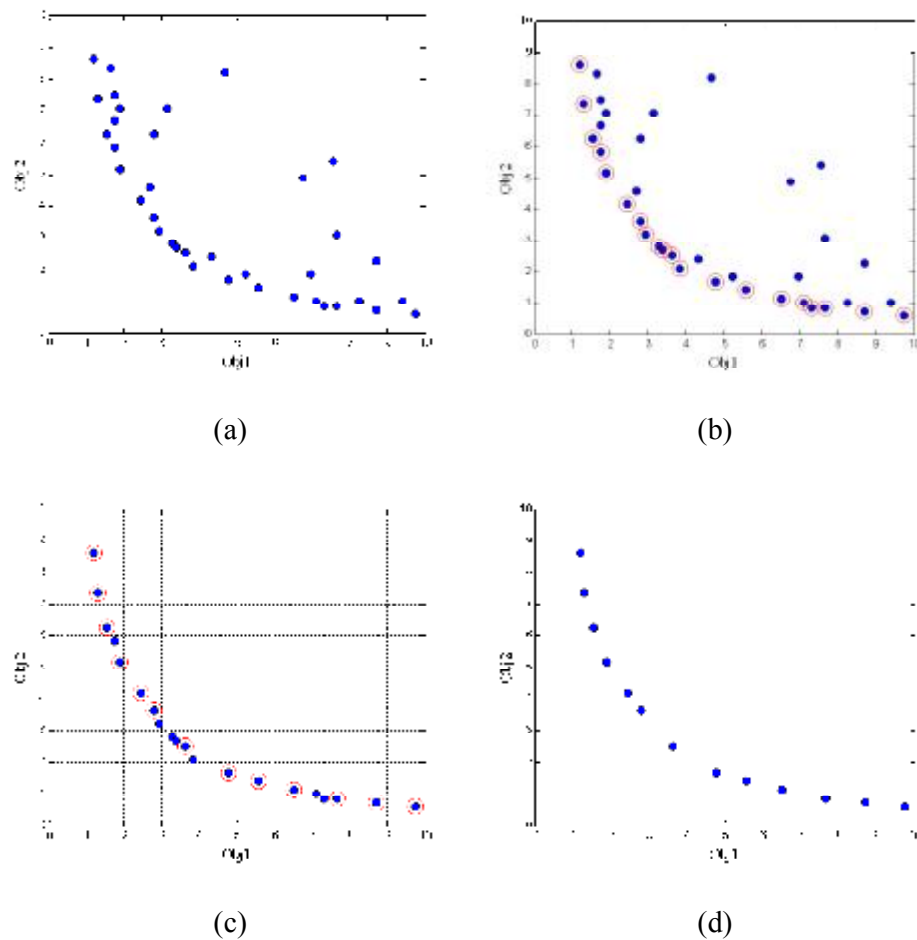


Figure 3-6. An example of non-dominated selection and diversity selection: (a) The solutions before the selections; (b) Non-dominated selection; (c) Cell selection; (d) The solutions after the selections

### 3.5 Experimental Studies using MO-RSSA

#### 3.5.1 Comparison between MO-RSSA and Other Multi-Objective EAs

A comparison between MO-RSSA and some salient multi-objective evolutionary algorithms, such as the Pareto Archived Evolution Strategy (PAES) [Knowles & Corne 2000], the Strength Pareto Evolutionary Algorithm (SPEA) [Zitzler &

### **Chapter 3: A New Reduced Space Searching Algorithm**

Thiele 1998] and the Non-dominated Sorting Genetic Algorithm II (NSGA-II) [Deb *et al.* 2002], was carried-out using the well-known multi-objective optimisation problems – the ZDT series benchmark problems [Zitzler *et al.* 2000].

#### **3.5.1.1 Problems Description**

The ZDT series benchmark functions are considered to be difficult to optimise, especially ZDT2, ZDT3 and ZDT4. All of these functions represent two-objective problems of minimising both  $f_1$  and  $f_2$ . They are described as follows [Zitzler *et al.* 2000]:

1. **ZDT1** (30-variable problem with a convex Pareto front):

$$f_1 = x_1, \quad g = 1 + \frac{9}{n-1} \sum_{i=2}^n x_i, \quad f_2 = g \left( 1 - \sqrt{f_1/g} \right), \quad 0 \leq x_i \leq 1, \quad n = 30.$$

2. **ZDT2** (30-variable problem with a concave Pareto front):

$$f_1 = x_1, \quad g = 1 + \frac{9}{n-1} \sum_{i=2}^n x_i, \quad f_2 = g \left( 1 - (f_1/g)^2 \right), \quad 0 \leq x_i \leq 1, \quad n = 30.$$

3. **ZDT3** (30-variable problem with a number of disconnected Pareto fronts):

$$f_1 = x_1, \quad g = 1 + \frac{9}{n-1} \sum_{i=2}^n x_i, \quad f_2 = g \left( 1 - \sqrt{f_1/g} - (f_1/g) \sin(10\pi f_1) \right),$$

$$0 \leq x_i \leq 1, \quad n = 30.$$

### Chapter 3: A New Reduced Space Searching Algorithm

---

4. **ZDT4** (10-variable problem with a Pareto front out of 100 local Pareto fronts):

$$f_1 = x_1, \quad g = 1 + 10(n-1) + \sum_{i=2}^n (x_i^2 - 10 \cos(4\pi x_i)), \quad f_2 = g \left(1 - \sqrt{f_1/g}\right),$$

$$0 \leq x_1 \leq 1, \quad -5 \leq x_i \leq 5, \quad i = 2, 3, \dots, n, \quad n = 10.$$

#### 3.5.1.2 Performance Metrics

The Generational Distance (*GD*) performance metrics measures the closeness of the obtained Pareto solution set  $Q$  from a known set of the Pareto-optimal set  $P$ . It is defined as follows [Deb 2001]:

$$GD = \frac{(\sum_{i=1}^{|Q|} d_i^m)^{1/m}}{|Q|} \quad (3.6)$$

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 are used to calculate the closeness metric *GD*.

The Spread  $\Delta$  measures the diversity of the solutions along the Pareto front in the final population and is defined as follows:

$$\Delta = \frac{\sum_{m=1}^M d_m^e + \sum_{i=1}^{|Q|} |d_i - \bar{d}|}{\sum_{m=1}^M d_m^e + |Q| \bar{d}} \quad (3.7)$$

where  $d_i$  is 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

### ***Chapter 3: A New Reduced Space Searching Algorithm***

---

of  $P$  and  $Q$  along the  $m$ th objective. It is worth nothing that for an ideal distribution of the solutions (uniform distribution)  $\Delta = 0$ .

#### **3.5.1.3 Experimental Setup**

For a meaningful comparison, the experiments configuration referred to the experiments in [Deb 2001]. The maximal function evaluation for every experiment was set to 25000. The result of every experiment was obtained after 10 independent runs. The parameter settings for different algorithms are listed as follows:

1. **MO-RSSA**: decreasing parameter  $C_1 = 3$ , increasing parameter  $C_2 = 1$ , changing ratio  $K = 0.5$ ,  $m = 15$ , frequency parameter  $H = 10000$  and a variation strategy of the combination of the one-dimensional variation (with the 75% probability of usage) and the multi-dimensional variation (with the 25% probability of usage).
2. **Pareto Archived Evolution Strategy (PAES)**: Population size 100, maximum generation 250, crossover probability 0.9 and mutation probability 0.01.
3. **Strength Pareto Evolutionary Algorithm (SPEA)**: Population size 80, external population size 20, maximum generation 250, crossover

### ***Chapter 3: A New Reduced Space Searching Algorithm***

---

probability 0.9 and mutation probability 1/80.

4. **Non-dominated Sorting Genetic Algorithm II (NSGA-II) (binary-coded)**: Population size 100, maximum generation 250, crossover probability 0.9 and mutation probability 1/(string-length). 30 bits were used to code each variable.

#### **3.5.1.4 Experimental Results**

The average  $GD$  and  $\Delta$  values of 10 runs and the corresponding variances  $\sigma^2$  are summarised in Tables 3-4 and 3-5 respectively. In these tables, the compared results for PAES, SPEA and NSGA-II are obtained from [Deb 2001]. It can be seen that MO-RSSA performs best in all the situations. It outperforms the other three well-known multi-objective optimisation algorithms in terms of both accuracy and diversity.

Figure 3-7 shows the graphical results produced by MO-RSSA. The true optimal Pareto fronts of the problems are represented with a continuous 'red' curve and the 'blue' points are the solutions obtained using MO-RSSA. It can be observed that the algorithm possesses very good convergence properties while maintaining a good diversity among the Pareto solutions.



### Chapter 3: A New Reduced Space Searching Algorithm

Table 3-4. Mean and variance values of  $GD$ ; the bold values represent the best results.

Problems	Mean value of $GD$ / (variance $\sigma^2$ )			
	MO-RSSA	PAES	SPEA	NSGA-II
ZDT1	<b>4.07e-4</b> (0)	8.21e-2 (8.68e-3)	1.25e-3 (0)	8.94e-4 (0)
ZDT2	<b>7.37e-4</b> (0)	1.26e-1 (3.69e-2)	3.04e-3 (2.00e-5)	8.24e-4 (0)
ZDT3	<b>4.07e-4</b> (0)	2.39e-2 (1.00e-5)	4.42e-2 (1.90e-5)	4.34e-2 (4.20e-5)
ZDT4	<b>1.82e-4</b> (0)	8.55e-1 (5.27e-1)	9.51e+0 (1.13e+1)	3.23e+0 (7.31e+0)

Table 3-5. Mean and variance values of  $\Delta$ ; the bold values represent the best results.

Problems	Mean value of $\Delta$ / (variance $\sigma^2$ )			
	MO-RSSA	PAES	SPEA	NSGA-II
ZDT1	4.68e-1 (1.66e-3)	1.23e+0 (4.84e-3)	7.30e-1 (9.07e-3)	<b>4.63e-1</b> <b>(4.16e-2)</b>
ZDT2	<b>3.94e-1</b> <b>(8.10e-4)</b>	1.17e+0 (7.68e-3)	6.78e-1 (4.48e-3)	4.35e-1 (2.46e-2)
ZDT3	<b>4.97e-1</b> <b>(5.30e-3)</b>	7.90e-1 (1.65e-3)	6.66e-1 (6.66e-4)	5.76e-1 (5.08e-3)
ZDT4	<b>4.78e-1</b> <b>(2.62e-3)</b>	8.70e-1 (1.01e-1)	7.32e-1 (1.13e-2)	4.79e-1 (9.84e-3)

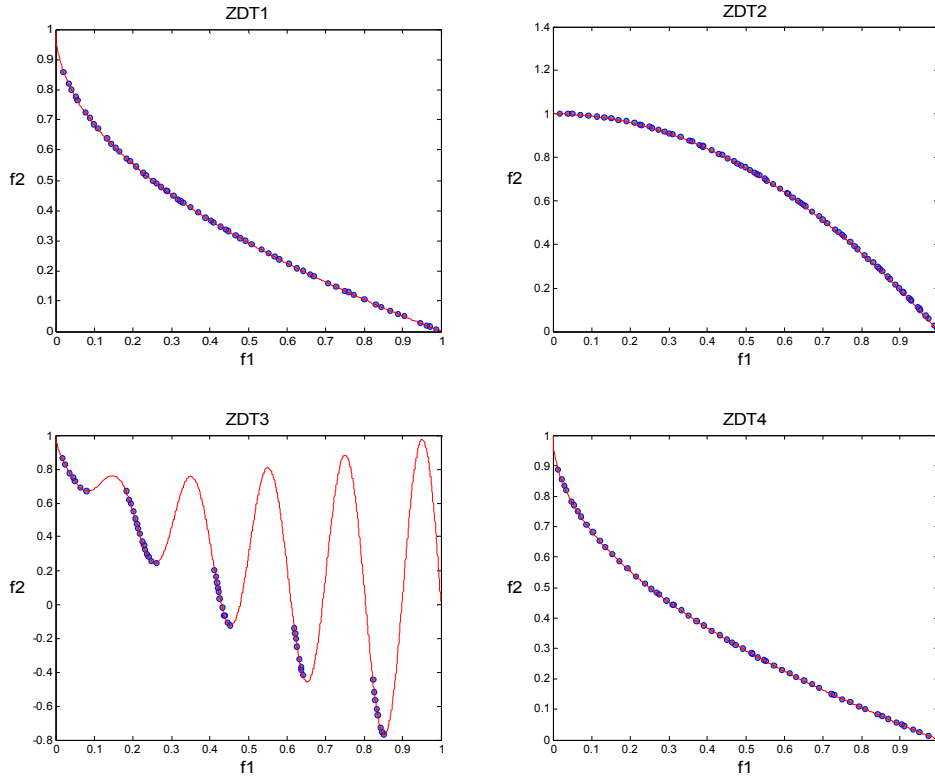


Figure 3-7. Pareto fronts obtained by MO-RSSA

### 3.5.2 Experiments Based on the DTLZ Series Problems

In this experiment, MO-RSSA was used to optimise the DTLZ series problems [Deb *et al.* 2001]. All the DTLZ problems were set so as to include three objectives. For a meaningful comparison with the optimisation results using NSGA-II and SPEA2 in [Deb *et al.* 2001], MO-RSSA used the same number of function evaluations as the experiments in the previous paper. The configuration of the MO-RSSA algorithm was the same as the one used for optimising ZDT problems, except the weight changing frequency parameter  $H$ , which is now taken

### Chapter 3: A New Reduced Space Searching Algorithm

---

to be 1000.

1. **DTLZ1**: The difficulty of this problem lies in the convergence to the hyper-plane. The search space contains  $(11^5-1)$  local Pareto-optimal fronts. Figure 3-8 shows the optimisation results using MO-RSSA in different angles of view. One can see that, MO-RSSA can obtain the Pareto-optimal solutions that possess a very good diversity.

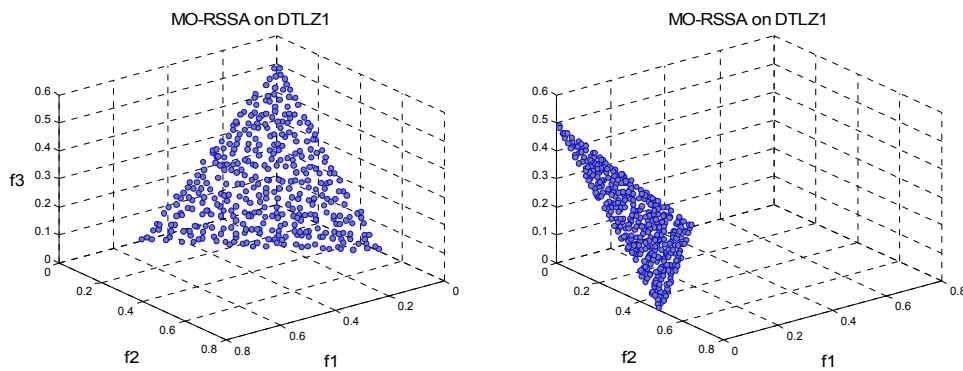


Figure 3-8. The optimisation result of MO-RSSA on DTLZ1

2. **DTLZ2**: The difficulty of this problem relates to its concave Pareto-optimal area. Figure 3-9 shows the optimisation results using the MO-RSSA algorithm. It can be seen that MO-RSSA can converge to the Pareto-optimal front with a good diversity.

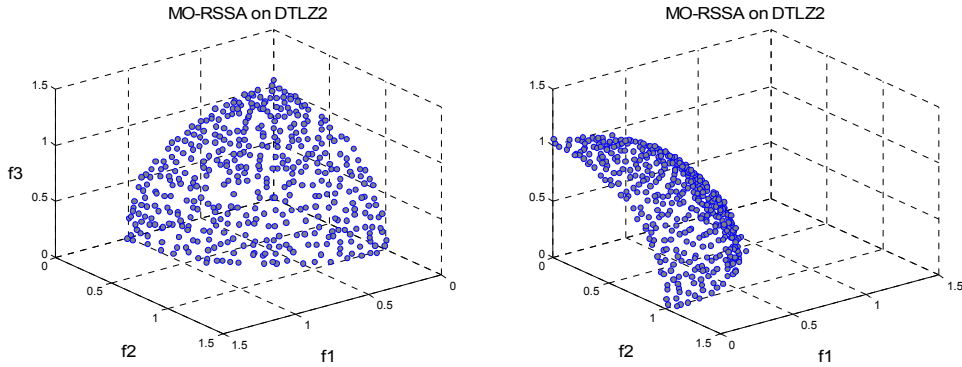


Figure 3-9. The optimisation result of MO-RSSA on DTLZ2

3. **DTLZ3**: The difficulty of this problem lies in the presence of several local Pareto-optimal fronts. This problem has  $3^{10}-1$  local Pareto-optimal fronts and one global Pareto-optimal front. All local Pareto-optimal fronts are parallel to the global Pareto-optimal front and a Multi-objective Evolutionary Algorithm (MOEA) can become trapped at any of these local Pareto-optimal fronts during the optimisation process. The optimisation result using MO-RSSA is shown in Figure 3-10. It can be seen that MO-RSSA can achieve the true Pareto-optimal front, while the results in [Deb *et al.* 2001] show that NSGA-II and SPEA2 cannot converge to the true Pareto front.

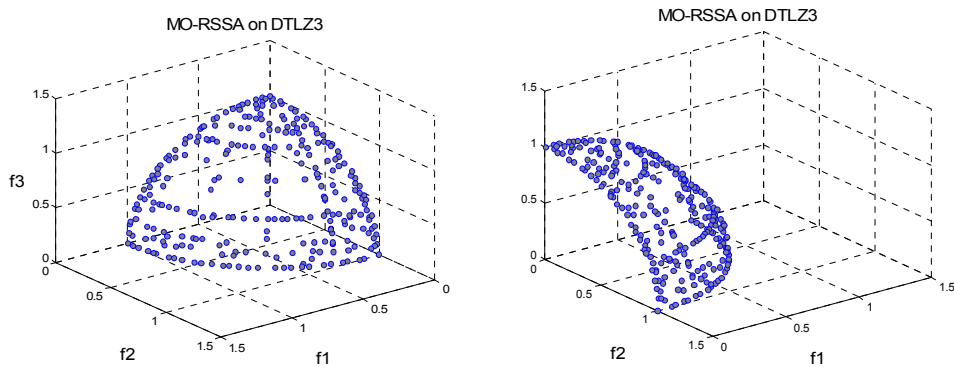


Figure 3-10. The optimisation result of MO-RSSA on DTLZ3

### Chapter 3: A New Reduced Space Searching Algorithm

4. **DTLZ5**: This problem will test MOEA's ability to converge to a curve. The optimisation result using MO-RSSA is shown in Figure 3-11. One can see that MO-RSSA is able to converge to the Pareto-optimal front with a good diversity. For this test problem, MO-RSSA can find the Pareto-optimal solutions in the early stages of the optimisation process.

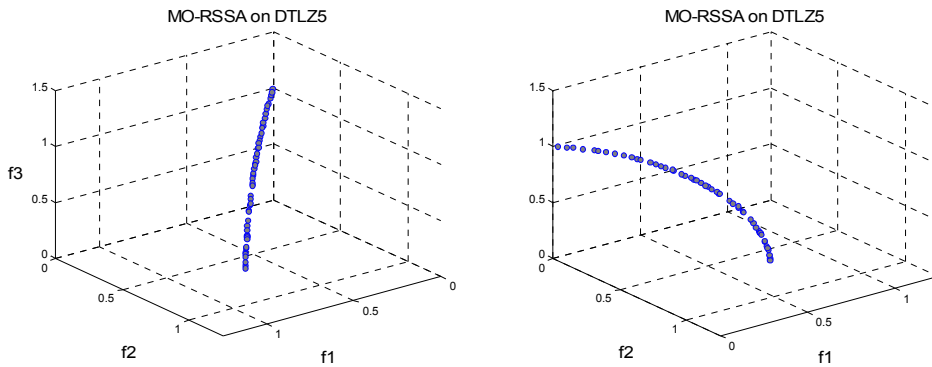


Figure 3-11. The optimisation result of MO-RSSA on DTLZ5

5. **DTLZ6**: This problem is a modified version of DTLZ5, which becomes more difficult to solve than DTLZ5. As the results in [Deb *et al.* 2001], MO-RSSA also cannot converge to the true Pareto-optimal front, which is shown in Figure 3-12.

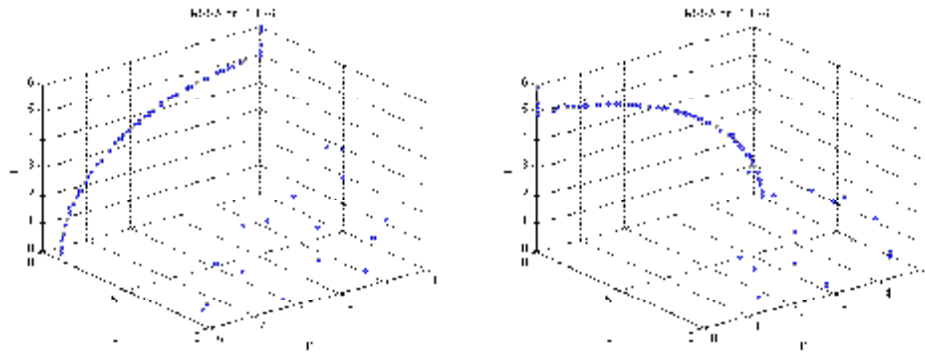


Figure 3-12. The optimisation result of MO-RSSA on DTLZ6

### Chapter 3: A New Reduced Space Searching Algorithm

---

6. **DTLZ7**: This problem has a disconnected set of Pareto-optimal regions and it will test the algorithm's ability to maintain a subpopulation in different Pareto-optimal regions. Figure 3-13 shows the optimisation result using MO-RSSA to optimise DTLZ7. From this figure, it can be observed that the new algorithm is able to find and maintain stable and distributed subpopulations in all four disconnected Pareto-optimal regions.

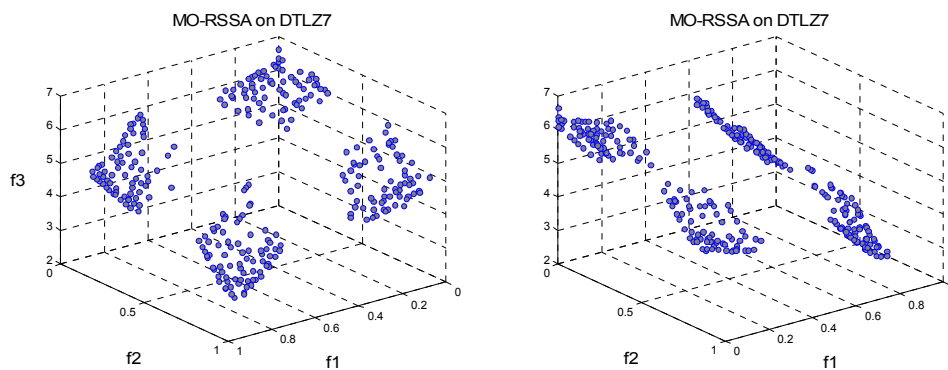


Figure 3-13. The optimisation result of MO-RSSA on DTLZ7

### 3.6 Summary

In this chapter, a new optimisation algorithm, Reduced Space Searching Algorithm (RSSA), was introduced, which is inspired from the simulation of the simple human societal behaviour when searching for optimal solutions in our daily routines. This new algorithm has been validated using a set of well-known benchmark problems. Compared with the recently developed and most salient optimisation algorithms, CMA-ES, DE and G3-PCX, RSSA performs as well as

### ***Chapter 3: A New Reduced Space Searching Algorithm***

---

and sometimes better than these algorithms. In addition, various parameter settings have also been explored and the appropriate parameter configurations were recommended.

Furthermore, RSSA was extended to the multi-objective optimisation case, in which the Random Weighted Aggregation (RWA) was employed. In addition, a new approach named ‘cell selection’ method was introduced in order to keep a good diversity of the Pareto-optimal solutions. After a comparison between the proposed Multi-Objective Reduced Space Searching Algorithm (MO-RSSA) and several well-known evolutionary multi-objective algorithms, such as PAES, SPEA and NSGA-II, which is based on a set of challenging problems ZDT and DTLZ series problems, it can be concluded that MO-RSSA shows noticeable improvements over these algorithms in terms of both accuracy and diversity of the Pareto solutions.

The next chapter will introduce a new optimisation algorithm based on Particle Swarm Optimisation [Kennedy & Eberhart 1995].

## **Chapter 4**

# **A New Structure for Particle Swarm Optimisation with Applications to Single Objective and Multi-Objective Problems**

### **4.1 Introduction to Particle Swarm Optimisation (PSO)**

Particle swarm optimisation (PSO) is a powerful evolutionary computation technique that was originally introduced by Kennedy and Eberhart [Kennedy & Eberhart 1995; Eberhart & Kennedy 1995]. It was developed by the simulation of a simplified animal social behaviour of birds flocking and fish schooling. This technique is initialised with a population of random solutions, called particles. Each particle flies through the search space with a velocity which is dynamically adjusted according to its own and its companions' historical behaviours. It is expected that the particles have a tendency to fly towards better search areas during the search process.



## ***Chapter 4: A New Structure for Particle Swarm Optimisation***

---

Since its introduction in 1995, the PSO technique is becoming very popular due to its simplicity of implementation and ability to quickly converge to a reasonably good solution. Though PSO has many advantages, it still suffers from premature convergence, by becoming trapped in local extrema. To solve this problem, a new structure, based on the original PSO algorithm, is proposed in this chapter, where a new component, named ‘momentum term’, is introduced to replace the ‘inertial term’ of the original PSO.

This chapter is organised as follows. The remaining part of Section 4.1 will provide a background knowledge about PSO. Section 4.2 will introduce the proposed new structure for PSO in details. In Section 4.3, sets of experiments will be carried-out to examine the optimisation performance of the new PSO. In Section 4.4, this new algorithm will be extended to include a multi-objective case and the results of applying the new multi-objective optimisation algorithm to some well-known benchmark problems, namely the ZDT series and the DTLZ series problems, will also be presented. Finally, summary remarks will be given in Section 4.5.

### **4.1.1 Basic Concepts of PSO**

In PSO, each particle represents an alternative solution in the multi-dimensional search space. Initially, a population of particles is generated with random positions and random velocities. Then, each particle flies through the search space with the

#### ***Chapter 4: A New Structure for Particle Swarm Optimisation***

---

velocity constantly updated according to its own flying experience and its companions' flying experience. Figure 4-1 shows an example of the particles' movement in a 2-dimensional search space. In this case, the 'blue' dots represent the particles' positions in a certain iteration; a 'yellow' dot represents the historical best position found hitherto for one particular particle; the 'red' dot is the best position found so far among all particles. In addition to these, the related positions are linked with the dashed lines; and the lines with the arrow marks indicate the possible movements of these particles. It is shown that, in one iteration, one particle has the tendency to fly towards its own best position that has been experienced ('yellow' dot) and also the particle tends to fly towards the best position experienced by all its companions ('red' dot). After one movement, all the personal previous best positions and the global best position for all the particles are evaluated and updated again.

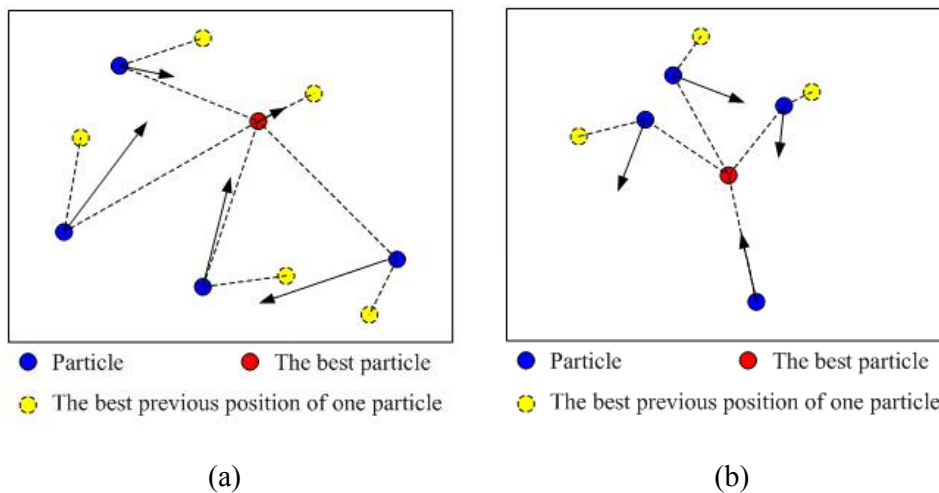


Figure 4-1. An example of the particles' movements in two continuous iterations in PSO

#### ***Chapter 4: A New Structure for Particle Swarm Optimisation***

---

Specifically, the position vector and velocity vector of the  $i$ th particle in a  $D$ -dimensional search space can be represented as  $x_i = [x_{i1}, x_{i2}, x_{i3}, \dots, x_{iD}]^T$  and  $v_i = [v_{i1}, v_{i2}, v_{i3}, \dots, v_{iD}]^T$  respectively. According to a predefined fitness function, the best previous position of the  $i$ th particle is  $p_i = [p_{i1}, p_{i2}, p_{i3}, \dots, p_{iD}]^T$  (its corresponding fitness value is named the ‘personal best’  $p_{best}$ ) and the fittest position among all the particles found so far is  $p_g = [p_{g1}, p_{g2}, p_{g3}, \dots, p_{gD}]^T$  (its corresponding fitness value is called the ‘global best’  $g_{best}$ ). The velocities and positions of the particles are updated according to the following equations [Shi & Eberhart 1998a]:

$$v_{id}(t+1) = w \times v_{id}(t) + c_1 \times r_{id}(t+1) \times (p_{id}(t) - x_{id}(t)) + c_2 \times R_{id}(t+1) \times (p_{gd}(t) - x_{id}(t)) \quad (4.1)$$

$$x_{id}(t+1) = x_{id}(t) + v_{id}(t+1) \quad (4.2)$$

where  $d = 1, 2, \dots, D$ ;  $t$  is the index of the iteration;  $w$  is the inertia weight;  $c_1$  and  $c_2$  are positive constants known as acceleration coefficients; and  $r_{id}(t)$  and  $R_{id}(t)$  are two uniformly distributed random variables in the range  $[0, 1]$ .

The first component of Equation (4.1) represents the previous velocity. It is used to model the tendency of the particle to fly in the same direction that it has been travelling. In this component, the inertia weight  $w$  controls the search behaviour of the particles. It can either be a fixed value or it can be dynamically changing [Eberhart & Shi 2000; Eberhart & Shi 2001b]. A higher inertia weight (for example 0.9) allows the particles to move freely and it is helpful for the particles to find the global optimum neighbourhood fast. A lower value of the inertia weight (for instance 0.4) can narrow the search scope and enable the particles to converge to local optima fast. In [Shi & Eberhart 1998b; Shi & Eberhart 1999], The authors

#### **Chapter 4: A New Structure for Particle Swarm Optimisation**

suggested employing a linearly decreasing inertia weight, which aims to execute a more global search (exploration) at the beginning and execute a more local search (exploitation) at the end. However, the main disadvantage of this method is that once the inertia weight is decreased, the swarm loses its ability to search new areas because it is not able to recover its exploration mode.

The second part of Equation (4.1) is viewed as the ‘memory’ component. It represents the personal thinking of one particle. This component attracts the particle to fly towards its own best positions so far found. The third part of Equation (4.1) is known as the ‘cooperation’ component, which represents the cooperative effect of the particles in the optimisation search. This component always encourages the particle to move towards the global best position. In these two components, the two acceleration coefficients  $c_1$  and  $c_2$  are often set to be 2, which is supposed to perform well [Eberhart *et al.* 2001a]. It is also important to note that  $c_1$  and  $c_2$  should not necessarily be equal because the weights for individual and group experience can vary according to the characteristics of different problems [Del Valle *et al.* 2008].

Generally, a maximum velocity vector  $v_{\max} = [v_{\max 1}, v_{\max 2}, v_{\max 3}, \dots, v_{\max D}]$  is defined and acts as the upper limit for the achievable velocity of the particles, where  $v_{\max d}$ ,  $d = 1, 2, \dots, D$ , are positive numbers. It works as follows:

$$\begin{aligned} &\text{If } v_{id} > v_{\max d}, \text{ then } v_{id} = v_{\max d}; \\ &\text{Else if } v_{id} < -v_{\max d}, \text{ then } v_{id} = -v_{\max d}. \end{aligned} \tag{4.3}$$

#### ***Chapter 4: A New Structure for Particle Swarm Optimisation***

---

It is important to note that the value of  $v_{\max}$  should not be too large, as the particles may move erratically and fly beyond a good solution; however, the value of  $v_{\max}$  should not be too small, as the particle's movement may be restricted to short steps and the optimal solution may be achieved after a long journey. Frequently, the  $v_{\max}$  value is confined within the range of search space and defined as follows:

$$v_{\max d} = (x_{\max d} - x_{\min d})/N, \quad (4.4)$$

where  $x_{\max d}$  and  $x_{\min d}$  are the maximum and minimum bands of the  $d$ th dimension of the search space;  $N$  is a positive number where  $1 \leq N \leq 10$ . In addition, research work performed by [Fan & Shi 2001] showed that an appropriate dynamically changing  $v_{\max}$  can also improve the performance of the PSO algorithm.

However, the particles may still occasionally fly to a position beyond the predefined search space and produce an invalid solution. In this case, a simple handling method is used in most of the PSO algorithms, which works as follows:

$$\begin{aligned} &\text{If } x_{id} > x_{\max d}, \text{ then } x_{id} = x_{\max d}; \\ &\text{Else if } x_{id} < x_{\min d}, \text{ then } x_{id} = x_{\min d}. \end{aligned} \quad (4.5)$$

Furthermore, other methods have also been proposed to solve the previous problem. For instance, it is suggested in [Robinson & Rahmat-Samii 2004] to enclose the search space with three different types of hypothetical boundaries each with its own boundary condition, namely, absorbing boundaries, reflecting boundaries and invisible boundaries. From this viewpoint, the method expressed in (4.5) can be viewed as the absorbing boundary condition. In [Huang & Mohan 2005], another method of damping boundary condition is proposed, which

#### **Chapter 4: A New Structure for Particle Swarm Optimisation**

---

combines the features of both the absorbing and reflecting boundaries.

In summary, the entire PSO algorithm can be described via the following procedure:

1. Initialise the swarm by assigning a random position to each particle within the problem hyperspace.
2. According to the predefined objective function, evaluate the fitness for each particle.
3. For each individual particle, compare the particle's fitness value with its  $p_{best}$ . If the current value is better than the  $p_{best}$ , then set the current value as the new  $p_{best}$  and set the current particle's position  $x_i$  as the new  $p_i$ .
4. Identify the particle that has the best fitness value. The value of its  $p_{best}$  is identified as  $g_{best}$  and its  $p_i$  is identified as the  $p_g$ .
5. Update the velocities and positions of all the particles using Equations (4.1) and (4.2).
6. If the velocity of one particle exceeds the upper limit  $v_{max}$ , then implement Method (4.3).

#### **Chapter 4: A New Structure for Particle Swarm Optimisation**

7. If the position of one particle exceeds the search bounds, then implement Method (4.4).
8. Repeat Step 2 to Step 7 until a stopping criterion (e.g., a maximum number of iterations or a sufficiently good fitness value) is achieved.

##### **4.1.2 New Developments in PSO**

In the open literature, several methods were suggested to tune the main parameters of PSO so as to obtain a good performance for the algorithm. For instance, some researchers added a random component to the inertia weight [Eberhart & Shi 2001b; Zhang & Hu 2003], and this was found to perform slightly better than the constant inertia weight. In [Shi & Eberhart 2001], a simple fuzzy system was applied to predict the appropriate inertia weight. Since the fuzzy rules and the parameters of the fuzzy system were obtained from the authors' own experience (hand-crafted rules), this fuzzy system cannot be universally adaptive for all the optimisation problems since different problems involve different characteristics. In [Doctor *et al.* 2004], the authors used a secondary PSO algorithm to find the optimal parameters of a primary PSO. Although this method may improve the performance of PSO, it brings more complexity in computation and the algorithm structure. In addition to the inertia weight factor, time-varying acceleration coefficients (TVAC) were introduced in [Ratnaweera *et al.* 2004].

#### ***Chapter 4: A New Structure for Particle Swarm Optimisation***

---

In [Richard & Ventura 2004], an initialisation method was proposed to produce the particles that are distributed as evenly as possible throughout the problem space. It was concluded that this method ensures a broad coverage of the search space and improves the performance of PSO compared with the original random initialisation.

The cooperative PSO (CPSO), proposed by Van den Bergh and Engelbrecht [Van den Bergh & Engelbrecht 2004], employs multiple swarms to optimise different components of the solution vector cooperatively. In CPSO, the search space is explicitly partitioned by splitting the solution vectors into smaller vectors. Two algorithms were proposed, namely CPSO- $S_k$  and CPSO- $H_k$ . In the CPSO- $S_k$  algorithm, a swarm with  $n$ -dimensional vectors is partitioned into  $k$ -swarms of smaller vectors, with each swarm attempting to optimise a small component of the solution vector. The advantage of the CPSO- $S_k$  approach is that only one component is modified at a time, therefore, many combinations are formed using different members from different swarms, yielding the desired fine-grained search and a significant increase in the solution diversity. On the other hand, CPSO- $H_k$  combines the two techniques of PSO and CPSO- $S_k$  by executing one iteration of CPSO- $S_k$  followed by one iteration of the standard PSO algorithm.

In addition, to the previous research activities, other researchers investigated the hybridisation by combining PSO with other search techniques to improve the performance of PSO. Evolutionary operators, such as selection and mutation, have been introduced within PSO to increase the diversity of the population and



#### ***Chapter 4: A New Structure for Particle Swarm Optimisation***

---

improve the ability to escape from local optima [Lovbjerg *et al.* 2001]. In [Lovbjerg *et al.* 2001], the swarm was divided into subpopulations, and a breeding operator was used within a subpopulation or between the subpopulations to increase the diversity of the population. Negative entropy was used to discourage the premature convergence in [Xie *et al.* 2002], while in [Parsopoulos & Vrahatis 2004], deflection, stretching, and repulsion techniques were used to find as many optima as possible by preventing the particles from moving to a previously discovered optimal region.

Improving PSO's performance by designing different types of topologies represents another active research direction. In most of these new approaches, one particle flies towards the neighbourhood best position instead of the global best position. Kennedy [Kennedy 1999; Kennedy & Mendes 2002] claimed that PSO with a small neighbourhood might perform better on complex problems, while the PSO with a large neighbourhood would perform better on simple problems. Suganthan [Suganthan 1999] applied a dynamically adjusted neighbourhood where the neighbourhood of a particle gradually increases until it includes all the particles. In [Hu & Eberhart 2002], the authors also used a dynamic neighbourhood where  $m$  closest particles in the performance space are selected to be its new neighbourhood in each generation. Peram *et al.* developed the fitness-distance-ratio-based PSO (FDR-PSO) with near neighbour interactions [Peram *et al.* 2003]. When updating each velocity dimension, the FDR-PSO algorithm selects one other particle  $nbest$ , which has a higher fitness value and is nearer to the particle being updated.

#### **Chapter 4: A New Structure for Particle Swarm Optimisation**

Recently, several multi-objective PSO algorithms have been developed based on the Pareto optimality concept. The main issue to be addressed is the selection of the cognitive and social leaders ( $p_{best}$  and  $g_{best}$ ) such that they can provide an effective guidance to reach the most promising Pareto front region but at the same time maintain the population diversity.

For the selection procedure, two typical approaches were suggested in the literature: a selection based on quantitative standards and random selection. In the first case, the leader is determined without any randomness involved, such as the Pareto ranking scheme [Ray 2002], the sigma method [Mostaghim & Teich 2003] or the dominated tree [Fieldsend *et al.* 2003]. However, in the random selection approach, the selection is stochastic and proportional to certain weights assigned to maintain the population diversity (crowding radius, crowding factor, niche count, etc.) [Hu 2006].

Coello and Lechuga [Coello & Lechuga 2002] have also incorporated the Pareto dominance into the PSO algorithm. In their method, the non-dominated solutions are stored in a secondary population. The primary population uses the neighborhood bests, which are randomly selected from this secondary population, to update the velocities.

## ***Chapter 4: A New Structure for Particle Swarm Optimisation***

---

### **4.1.3 Features of PSO**

PSO is a computational and intelligence-based technique, which has some advantages over other similar optimisation techniques, namely the following:

1. PSO is easier to implement and it includes fewer parameters to adjust.
2. PSO is not largely affected by the size and nonlinearity of the problem.
3. In PSO, every particle remembers its own previous best solution as well as its companions' information; therefore, it has a more effective memory capability.

### **4.2 A New Structure for Particle Swarm Optimisation (nPSO)**

Though PSO based algorithms have many advantages, they still suffer from premature convergence. To solve this problem, one new term, named the 'momentum term' is introduced in this chapter to replace the inertial term of the original PSO. This new component can help in avoiding such a premature convergence and in encouraging each particle to jump out of any local minimum. To provide the particles with more adaptability, a separate momentum weight is assigned to each particle as it dynamically adjusts itself according to the particle's own search experience.

## Chapter 4: A New Structure for Particle Swarm Optimisation

---

### 4.2.1 Algorithm Formulation

In this new PSO (nPSO), a new ‘momentum term’ is introduced to replace the original inertia term. As defined in the standard PSO algorithm, the position vector and velocity vector of the  $i$ th particle in the  $D$ -dimensional search space can be represented as  $x_i = [x_{i1}, x_{i2}, x_{i3}, \dots, x_{iD}]^T$  and  $v_i = [v_{i1}, v_{i2}, v_{i3}, \dots, v_{iD}]^T$  respectively. According to a predefined fitness function, the best previous position of the  $i$ th particle and the fittest position among all the particles found so far are represented as  $p_i = [p_{i1}, p_{i2}, p_{i3}, \dots, p_{iD}]^T$  and  $p_g = [p_{g1}, p_{g2}, p_{g3}, \dots, p_{gD}]^T$  respectively.  $v_{\max} = [v_{\max 1}, v_{\max 2}, v_{\max 3}, \dots, v_{\max D}]$  is an upper limit for the achievable velocity of the particles. The new algorithm can be described using the following equations:

$$v_{id}(t+1) = w_{id}(t-1) \times r1_{id}(t+1) \times v_{\max d} + c_1 \times r2_{id}(t+1) \times (p_{id}(t) - x_{id}(t)) + c_2 \times r3_{id}(t-1) \times (p_{gd}(t) - x_{id}(t)), \quad (4.6)$$

$$x_{id}(t+1) = x_{id}(t) + v_{id}(t+1), \quad (4.7)$$

where  $d = 1, 2, \dots, D$ ;  $t$  is the index of the iteration;  $w_{id}$  is the momentum weight of the  $i$ th particle in the  $d$ th search space dimension;  $c_1$  and  $c_2$  are positive constants known as acceleration coefficients;  $r1_{id}(t)$ ,  $r2_{id}(t)$  and  $r3_{id}(t)$  are three uniformly distributed random variables in the range  $[0, 1]$ .

For minimisation problems, the momentum weight  $w_{id}$  is varied in the optimisation process as follows:

#### Chapter 4: A New Structure for Particle Swarm Optimisation

---

$$w_{id}(t+1) = \begin{cases} 1, & \text{if } v_{id}(t) \leq \varepsilon \times v_{\max d} \text{ and} \\ & pos_{id}(t+1) = 1; \\ w_{id}(t) \times m_1, & \text{if } v_{id}(t) \geq \varepsilon \times v_{\max d} \text{ and} \\ & f(\mathbf{x}_i(t)) \geq f(\mathbf{p}_i(t-1)); \\ w_{id}(t) \times m_2, & \text{if } v_{id}(t) \geq \varepsilon \times v_{\max d} \text{ and} \\ & f(\mathbf{x}_i(t)) < f(\mathbf{p}_i(t-1)). \end{cases} \quad (4.8)$$

where  $\varepsilon$  is a small positive coefficient;  $m_1$  and  $m_2$  are two scaling parameters used to control the range of magnitudes of the momentum weight varying, where  $m_1$  is a positive coefficient smaller than 1 and  $m_2$  is a positive coefficient larger than 1;  $pos_{id}$  is a discrete variable used to control the direction in which the velocity of one particle should be refreshed, where its value is 0 or 1;  $f(x)$  is the fitness function of the minimisation problem.

The momentum term, which is the first component of Equation (4.6), aims at giving one particle extra adjustable momentum in its optimal search process to keep a balance between exploration (global search) and exploitation (local search). When a particle converges to one solution, which is judged by whether the velocity of the particle is too small or not, it may become trapped in a local minimum. In this situation, the momentum weight is set at a big value '1' to encourage the particle to jump out from the local minimum. When a particle does not converge, the momentum weight is dynamically adjusted according to the particle's own search experience: More specifically, if the particle can find a better solution in the previous generation, it may be in a local optimum space and the momentum should be increased to enhance the global search ability; if the particle cannot find a better solution in the previous generation, it may wander in a large space and the momentum should be reduced to enhance the local search ability.

#### **Chapter 4: A New Structure for Particle Swarm Optimisation**

This mechanism is proved to be effective in balancing the global search and local search and makes the optimisation search more adaptive.

Normally,  $\epsilon$  is set to  $10^{-10}$  without any loss of generality, which means that if the velocity of a particle is smaller than  $V_{\max} \times 10^{-10}$ , the particle is assumed to be converging. In addition,  $m_1$  is generally set to 0.5 and  $m_2$  is set to 2, which means that the momentum weight will reduce to the half of the previous value or increase to the double of the previous value. The position parameter  $pos_{id}$  is used to control the direction in which the velocity of the particle should be refreshed. In this work, three strategies for setting the value for  $Pos_i = (pos_{i1}, pos_{i2}, pos_{i3}, \dots, pos_{id})$  are proposed as follows:

1. **One-directional refresh mechanism:** In every iteration, for each particle, only one of the elements in the position parameter vector  $Pos_i$  will be randomly chosen and set to the value '1'; other elements will be set to the value '0'.
2. **Multiple-directional refresh mechanism:** The number of elements and the positions of elements that will be set to 1 are both randomly generated; other elements in the position parameter vector will be set to 0.
3. **All-directional refresh mechanism:** All the elements of  $Pos_i$  will be set to the value '1' all the time.

## ***Chapter 4: A New Structure for Particle Swarm Optimisation***

---

### **4.2.2 Advantages of nPSO**

The advantages of nPSO, compared with the standard PSO, lie mainly in the following three aspects:

1. Both the momentum term in nPSO and the inertia term in the original PSO aim to avoid premature convergence in the optimisation process. But the new term is more adaptive for different problems than the original one. For the standard PSO, different optimisation problems need different inertia weight settings, while in nPSO, the momentum weight can adjust itself adaptively. As it varies, it will find a suitable value for different optimisation problems in the whole optimisation process. This guarantees a fast convergence and concomitantly avoids premature convergence.
2. If the particles converge to local optima, the normal PSO cannot refresh the velocities of particles and cannot encourage the particles to jump out from local optima. But nPSO can do so through setting the momentum weight to a new larger value.
3. In the standard PSO, different particles share one inertia weight. When dealing with an optimisation problem, the change of inertia weight value may be suitable for one part of the particles to direct their optimisation search, but is unsuitable for the other part of particles; in nPSO, every particle has its own momentum weight. One particle adjusts its weight

## **Chapter 4: A New Structure for Particle Swarm Optimisation**

according to its own experience and situation. This enables the movements of particles to be efficient and hence giving the particles more adaptability.

### **4.3 Experimental Studies using nPSO**

In the following experiments, the benchmark test functions which have already been described in Section 3.3.1 are used.

#### **4.3.1 Effect of the Population Size**

In this experiment, the nPSO algorithm was tested using the benchmark problems  $f_1, f_2, f_3, f_7, f_8$  and  $f_{10}$  with various settings of the population size ( $Np = 2, 5, 10, 20, 50, 100, 200, 500$  and  $1000$ ). For each benchmark problem, the dimension  $D$  was increased in the sequence to  $2, 5, 10, 20$  and  $50$ ; and for each setting  $20$  runs were conducted. In each run, the maximum number of function evaluation was set to  $10^6$  and the optimisation process was regarded as successful and stopped, when the best solution  $Fb$  satisfied:  $Fb < 10^{-5}$  if the true global minimum  $Gb = 0$  or  $|(Fb - Gb) / Gb| < 10^{-5}$  if  $Gb \neq 0$ .

In this case, the acceleration coefficients  $c_1$  and  $c_2$  were set to  $1.5$ ; the scaling parameters  $m_1$  and  $m_2$  were set to  $0.5$  and  $2$  respectively;  $\epsilon = 10^{-10}$ ;  $v_{\max} = [v_{\max 1}, v_{\max 2}, \dots, v_{\max D}] = [0.5 \times (x_{\max 1} - x_{\min 1}), 0.5 \times (x_{\max 2} - x_{\min 2}), \dots, 0.5 \times (x_{\max D} - x_{\min D})]$ ,



### **Chapter 4: A New Structure for Particle Swarm Optimisation**

---

where  $x_{\max d}$  and  $x_{\min d}$  are the maximum and minimum bounds for the  $d$ th variable; The position parameter  $pos_{id}$  was updated using both the one-directional refresh mechanism (with the 70% probability of usage) and the multiple-directional refresh mechanism (with the 30% probability of usage), where  $i = 1, 2, \dots, Np$  and  $d = 1, 2, \dots, D$ .

Tables 4-1 to 4-6 and Figure 4-2 show the performance of nPSO with different population sizes in optimising the test problems with different dimensions. From these tables and figures, the following has been observed:

1. In most of the situations, nPSO performs best when the population size is relatively small (for example,  $Np = 5, 10$  and  $20$ ).
2. For low-dimensional problems ( $D = 2$ ), nPSO with large population size (for instance,  $Np = 100, 200$  and  $500$ ) performs better in the majority of cases.

Table 4-1. The average number of function evaluations to find the optimum on  $f_1$ ; the bold values represent the best results.

$D$	$Np=2$	$Np=5$	$Np=10$	$Np=20$	$Np=50$	$Np=100$	$Np=200$	$Np=500$	$Np=1000$
2	<b>743</b>	913	1165	1344	1500	2008	1348	2320	3980
5	661	<b>531</b>	675	1085	2321	4258	8168	18070	32740
10	808	<b>662</b>	962	1593	3405	6254	11584	26470	49980
20	<b>1257</b>	1281	1743	2772	5445	9568	17268	38510	71650
50	<b>3246</b>	3278	4307	6347	11643	19750	34220	72550	131300

### Chapter 4: A New Structure for Particle Swarm Optimisation

Table 4-2. The average number of function evaluations to find the optimum on  $f_2$ ; the bold values represent the best results.

$D$	$Np=2$	$Np=5$	$Np=10$	$Np=20$	$Np=50$	$Np=100$	$Np=200$	$Np=500$	$Np=1000$
2	12546	15040	15593	16640	18074	15732	8092	<b>2300</b>	4180
5	1983	<b>1473</b>	1571	2186	4464	8178	15236	32330	56400
10	8826	5696	3218	<b>3049</b>	6145	11450	21350	49600	94850
20	48405	26383	20193	<b>13142</b>	15320	17480	31460	71200	135000
50	281280	173840	123864	<b>100284</b>	150105	169840	117500	170250	270000

Table 4-3. The average number of function evaluations to find the optimum on  $f_3$ ; the bold values represent the best results.

$D$	$Np=2$	$Np=5$	$Np=10$	$Np=20$	$Np=50$	$Np=100$	$Np=200$	$Np=500$	$Np=1000$
2	1052	1013	1325	1776	1604	<b>942</b>	1512	2180	4080
5	3897	3063	2362	<b>1788</b>	2758	4594	7912	17990	32620
10	10371	5864	<b>5356</b>	6583	8668	12045	16890	32375	54050
20	21831	<b>19364</b>	21804	26159	39833	57410	88820	156300	232300
50	173390	<b>123337</b>	140094	178580	275580	385300	625120	N/A	N/A

Table 4-4. The average number of function evaluations to find the optimum on  $f_7$ ; the bold values represent the best results.

$D$	$Np=2$	$Np=5$	$Np=10$	$Np=20$	$Np=50$	$Np=100$	$Np=200$	$Np=500$	$Np=1000$
2	2163	2907	2938	2702	3229	1882	<b>1032</b>	2180	4020
5	<b>2258</b>	2410	2827	2702	5515	6510	11160	21600	35320
10	<b>11449</b>	11810	11847	15838	23713	31785	49890	93850	124900
20	54649	39779	<b>36474</b>	56296	75900	109940	157640	266700	438200
50	401260	204067	<b>171152</b>	187576	302690	451340	N/A	N/A	N/A

#### Chapter 4: A New Structure for Particle Swarm Optimisation

Table 4-5. The average number of function evaluations to find the optimum on  $f_8$ ; the bold values represent the best results.

$D$	$Np=2$	$Np=5$	$Np=10$	$Np=20$	$Np=50$	$Np=100$	$Np=200$	$Np=500$	$Np=1000$
2	33461	33021	40008	37442	36913	16085	8050	<b>2300</b>	4000
5	2413	<b>1827</b>	1875	2554	4898	8965	15270	39150	61150
10	8918	5551	4179	<b>3793</b>	6275	11680	21650	50400	95400
20	60416	31773	19523	<b>16213</b>	18679	27500	33380	69450	129350
50	511950	324120	169570	122964	<b>116950</b>	210635	256170	355250	487150

Table 4-6. The average number of function evaluations to find the optimum on  $f_{10}$ ; the bold values represent the best results.

$D$	$Np=2$	$Np=5$	$Np=10$	$Np=20$	$Np=50$	$Np=100$	$Np=200$	$Np=500$	$Np=1000$
2	2469	2464	2673	3235	3714	3187	<b>1908</b>	2265	4150
5	1174	1071	<b>1036</b>	1529	3107	5698	10540	23680	42580
10	6776	<b>5101</b>	5162	5332	8500	9712	15090	35075	65600
20	37936	21697	20854	<b>20718</b>	27503	38080	48260	92550	132200
50	239850	130176	<b>98510</b>	112520	122090	162780	224000	447200	618600

**Chapter 4: A New Structure for Particle Swarm Optimisation**

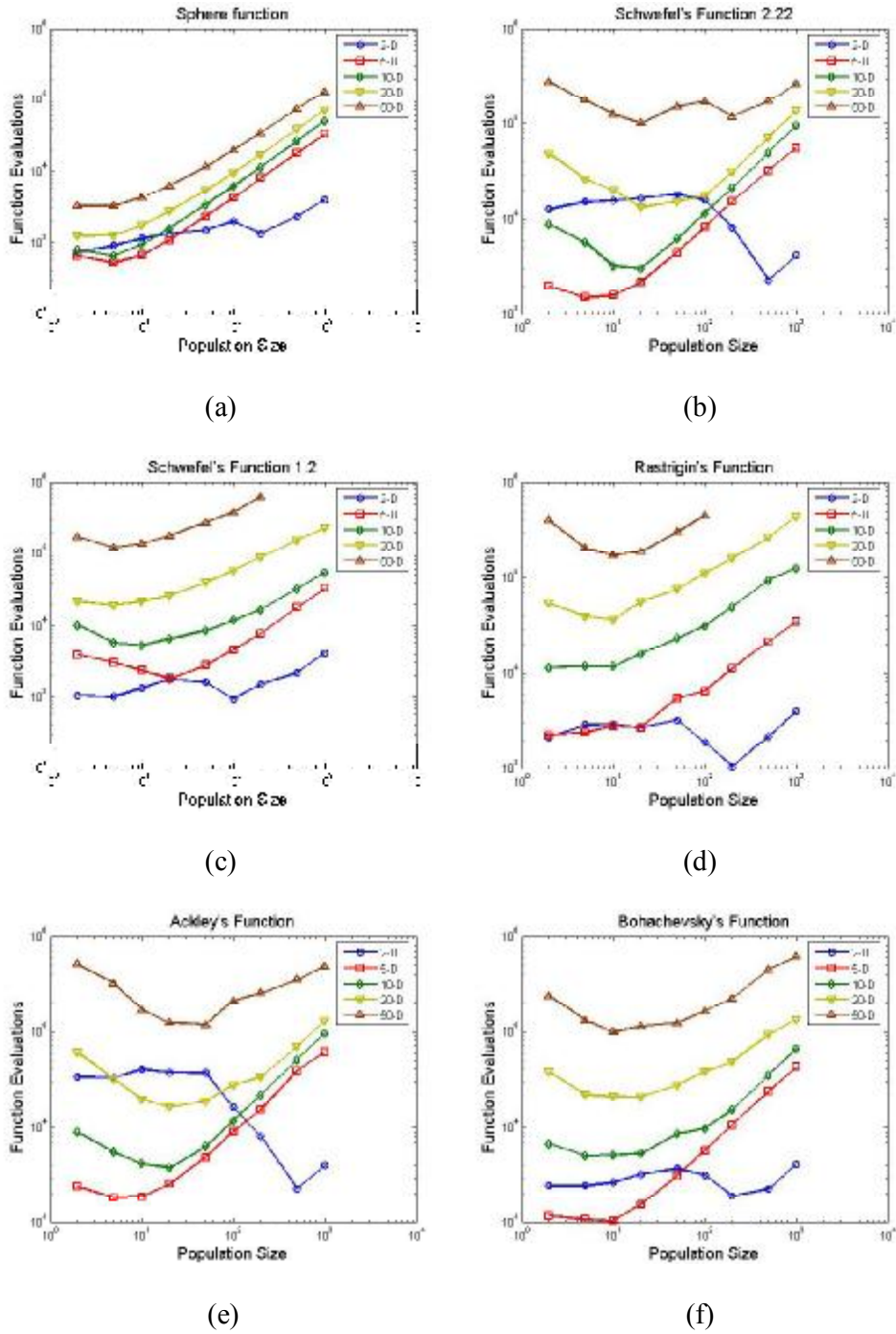


Figure 4-2. The mean number of function evaluations to find the optimum versus the population size with different problem dimension: (a)  $f_1$ , (b)  $f_2$ , (c)  $f_3$ , (d)  $f_7$ , (e)  $f_8$  and (f)  $f_{10}$ .

## ***Chapter 4: A New Structure for Particle Swarm Optimisation***

---

### **4.3.2 Effects of the Acceleration Coefficients**

In this experiment, the nPSO algorithm was tested using the benchmark problems  $f_1$  to  $f_{14}$  with various settings of  $c_1$  and  $c_2$  ( $c_1 = c_2 = 1, 1.1, 1.2, \dots, 2$ ). For problems  $f_1$  to  $f_{11}$ , the dimension  $D$  was set to 30. For each setting, 20 runs were conducted and averaged. For each run, the maximum number of function evaluation was set to  $10^6$ . The population size of particles  $Np$  was set to 10 and other settings were the same as the experiments in Section 4.3.1.

Table 4-7 shows the performance of nPSO with different acceleration coefficients.

It can be seen as follows:

1. Large acceleration coefficients (for example,  $c_1 = c_2 = 1.6$  to  $1.8$ ) perform well in difficult problems, such as the high-dimensional multimodal problems.
2. Small acceleration coefficients (for example,  $c_1 = c_2 = 1.0$  to  $1.2$ ) are more suitable for simple problems, such as the low-dimensional problems with few local extrema.

This means that large acceleration coefficients will enhance the exploration (global search) ability and small acceleration coefficients can increase the exploitation (local search) ability of the algorithm.

### Chapter 4: A New Structure for Particle Swarm Optimisation

Table 4-7. Average performance of nPSO with various acceleration coefficients in optimising  $f_1$  to  $f_{14}$ : The integer value in every cell is the average function evaluation number in successful runs; the value in parenthesis is the average result in the unsuccessful runs; the percentage value in the square brackets indicates the percentage of the successful runs out of all the runs; the bold values represent the best results.

Function	$c_1 = c_2$										
	1.0	1.1	1.2	1.3	1.4	1.5	1.6	1.7	1.8	1.9	2.0
$f_1$	2772 (N/A) [100%]	2544 (N/A) [100%]	<b>2464</b> (N/A) [100%]	2530 (N/A) [100%]	2553 (N/A) [100%]	2611 (N/A) [100%]	2805 (N/A) [100%]	3162 (N/A) [100%]	4403 (N/A) [100%]	13953 (N/A) [100%]	85080 (N/A) [100%]
$f_2$	99451 (N/A) [100%]	78157 (N/A) [100%]	69175 (N/A) [100%]	61803 (N/A) [100%]	52144 (N/A) [100%]	38744 (N/A) [100%]	35007 (N/A) [100%]	25606 (N/A) [100%]	<b>17164</b> (N/A) [100%]	33118 (N/A) [100%]	153340 (N/A) [100%]
$f_3$	56569 (N/A) [100%]	52503 (N/A) [100%]	49848 (N/A) [100%]	48483 (N/A) [100%]	<b>48108</b> (N/A) [100%]	49355 (N/A) [100%]	54028 (N/A) [100%]	74854 (N/A) [100%]	122618 (N/A) [100%]	291296 (N/A) [100%]	N/A (0.0130) [0%]
$f_4$	N/A (0.3167) [0%]	644324 (N/A) [100%]	506652 (N/A) [100%]	N/A (0.0032) [0%]	390060 (0.0087) [80%]	402694 (N/A) [100%]	335208 (N/A) [100%]	275428 (N/A) [100%]	<b>160232</b> (N/A) [100%]	385882 (N/A) [100%]	N/A (0.0032) [0%]
$f_5$	N/A (0.8108) [0%]	N/A (0.8042) [0%]	N/A (0.0039) [0%]	N/A (0.7994) [0%]	N/A (0.7987) [0%]	N/A (0.7982) [0%]	<b>N/A</b> ( $5.2941 \times 10^{-4}$ ) [0%]	N/A (0.7978) [0%]	N/A (0.0024) [0%]	N/A (1.4771) [0%]	N/A (43.097) [0%]
$f_6$	N/A (-10864) [0%]	N/A (-11077) [0%]	N/A (-10959) [0%]	N/A (-10698) [0%]	<b>N/A</b> (-11101) [0%]	N/A (-10864) [0%]	N/A (-10722) [0%]	N/A (-10817) [0%]	N/A (-10864) [0%]	N/A (-11042) [0%]	N/A (-10757) [0%]
$f_7$	407034 (N/A) [100%]	143905 (N/A) [100%]	104384 (N/A) [100%]	94201 (N/A) [100%]	96601 (N/A) [100%]	79670 (N/A) [100%]	75998 (N/A) [100%]	66297 (N/A) [100%]	<b>52531</b> (N/A) [100%]	96235 (N/A) [100%]	386905 (N/A) [100%]
$f_8$	82181 (N/A) [100%]	73467 (N/A) [100%]	53191 (N/A) [100%]	64546 (N/A) [100%]	51679 (N/A) [100%]	49585 (N/A) [100%]	48362 (N/A) [100%]	44917 (N/A) [100%]	<b>32738</b> (N/A) [100%]	47774 (N/A) [100%]	247168 (N/A) [100%]
$f_9$	5527 (0.0091) [35%]	5012 (0.0128) [50%]	4728 (0.0103) [40%]	4390 (0.0088) [55%]	<b>4228</b> (0.0099) [90%]	63587 (0.0116) [30%]	4240 (0.0154) [35%]	20453 (0.0127) [45%]	20484 (0.0196) [50%]	220577 (0.0439) [30%]	187890 (0.0228) [20%]
$f_{10}$	82150 (N/A) [100%]	74970 (N/A) [100%]	65383 (N/A) [100%]	55895 (N/A) [100%]	46030 (N/A) [100%]	38773 (N/A) [100%]	38623 (N/A) [100%]	30874 (N/A) [100%]	<b>17295</b> (N/A) [100%]	25213 (N/A) [100%]	132778 (N/A) [100%]
$f_{11}$	N/A (114.00) [0%]	N/A (158.85) [0%]	N/A (150.40) [0%]	N/A (124.93) [0%]	N/A (109.67) [0%]	N/A (122.68) [0%]	N/A (0.0010) [0%]	<b>N/A</b> ( $9.8807 \times 10^{-4}$ ) [0%]	N/A (0.0013) [0%]	N/A (0.0015) [0%]	N/A (0.0021) [0%]
$f_{12}$	1287 (N/A) [100%]	<b>1283</b> (N/A) [100%]	1384 (N/A) [100%]	1392 (N/A) [100%]	1467 (N/A) [100%]	1498 (N/A) [100%]	1854 (N/A) [100%]	1813 (N/A) [100%]	2933 (N/A) [100%]	3822 (N/A) [100%]	4950 (N/A) [100%]
$f_{13}$	<b>1472</b> (N/A) [100%]	1495 (N/A) [100%]	1507 (N/A) [100%]	1682 (N/A) [100%]	1893 (N/A) [100%]	1673 (N/A) [100%]	1835 (N/A) [100%]	2291 (N/A) [100%]	2573 (N/A) [100%]	3611 (N/A) [100%]	4382 (N/A) [100%]
$f_{14}$	<b>1458</b> (N/A) [100%]	1461 (N/A) [100%]	1704 (N/A) [100%]	1667 (N/A) [100%]	1796 (N/A) [100%]	1807 (N/A) [100%]	2092 (N/A) [100%]	2632 (N/A) [100%]	3177 (N/A) [100%]	3970 (N/A) [100%]	5024 (N/A) [100%]

## ***Chapter 4: A New Structure for Particle Swarm Optimisation***

---

### **4.3.3 Effects of Different Mechanisms for Position Parameters Refreshing**

As already stated, the position parameter is used to control the direction in which the velocity of one particle should be refreshed. Three strategies for setting the value of the position parameters  $Pos_i$  are compared here. In this experiment, the 30-dimension multimodal benchmark problems  $f_5$  to  $f_{11}$  were tested. The population size  $Np$  was set to 10 and the acceleration coefficients  $c_1$  and  $c_2$  were set to 1.8. Other settings were the same as those in the experiments of Section 4.3.1.

From Table 4-8, it can be seen that the one-directional refresh mechanism performs best on the functions  $f_6$ ,  $f_7$ ,  $f_8$  and  $f_{10}$ , while the multiple-directional refresh mechanism performs best on the problems  $f_5$ ,  $f_9$  and  $f_{11}$ . For a broad adaptation to various problems, it is recommended to use both the one-directional and multiple-directional refresh mechanisms simultaneously.

#### Chapter 4: A New Structure for Particle Swarm Optimisation

Table 4-8. Average performance of nPSO with different position parameter refresh mechanisms in optimising  $f_5$  to  $f_{11}$ : The integer in every cell is the average function evaluation number in successful runs; the value in parenthesis is the average result in the unsuccessful runs; the percentage value in the square brackets indicates the percentage of the successful runs out of all the runs; the bold values represent the best results.

Function	One-directional Refresh Mechanism	Multiple-directional Refresh Mechanism	All-directional Refresh Mechanism
$f_5$	N/A (1.2365) [0%]	<b>N/A</b> <b>(0.7989)</b> <b>[0%]</b>	N/A (1.5999) [0%]
$f_6$	<b>N/A</b> <b>(-10698)</b> <b>[0%]</b>	N/A (-10292) [0%]	N/A (-10504) [0%]
$f_7$	<b>46856</b> <b>(N/A)</b> <b>[100%]</b>	172372 (N/A) [100%]	163530 (33.165) [40%]
$f_8$	<b>27960</b> <b>(N/A)</b> <b>[100%]</b>	47216 (N/A) [100%]	47238 (N/A) [100%]
$f_9$	8387 (0.0130) [30%]	<b>14135</b> <b>(0.0240)</b> <b>[60%]</b>	7176 (0.0197) [30%]
$f_{10}$	<b>12856</b> <b>(N/A)</b> <b>[100%]</b>	28999 (N/A) [100%]	33483 (N/A) [100%]
$f_{11}$	N/A (0.0016) [0%]	<b>N/A</b> <b>(0.0012)</b> <b>[0%]</b>	N/A (39.907) [0%]

#### 4.3.4 Comparison between nPSO and Other Evolutionary Algorithms

In this section, experiments were carried-out using nPSO, two versions of normal



#### **Chapter 4: A New Structure for Particle Swarm Optimisation**

PSO, which apply a linearly decreasing inertia weight (PSO-LD) [Shi & Eberhart 1999] and a randomly varying inertia weight (PSO-RV) [Eberhart & Shi 2001b], and other three salient Evolutionary Algorithms, which are the Covariance Matrix Adaptation Evolution Strategy (CMA-ES) [Hansen *et al.* 2003], the Differential Evolution (DE) [Storn & Price 1995] and the Generalised Generation Gap model with the Parent-Centric Recombination operator (G3+PCX) [Deb *et al.* 2002].

The parameter settings for these algorithms were as follows:

1. **nPSO**:  $N_p = 10$ ;  $c_1 = c_2 = 1.8$ ;  $m_1 = 0.5$  and  $m_2 = 2$ ;  $\epsilon = 10^{-10}$ ;  $pos_{id}$  was updated using both the one-directional refresh mechanism (with the 50% probability of usage) and the multiple-directional refresh mechanism (with the 50% probability of usage).
2. **PSO-LD**: Population size  $N = 10 \times D$ , where  $D$  is the dimension of the test problem;  $c_1 = c_2 = 1.8$ ; the inertia weight  $w$  varied from 0.9 at the beginning of the search to 0.4 at the end of the search [Shi & Eberhart 1999].
3. **PSO-RV**:  $N = 10 \times D$ ;  $c_1 = c_2 = 1.8$ ;  $w$  changed according to the equation:  $w = 0.4 + rand() \times 0.5$ , where  $rand()$  is a uniformly distributed random number within the range  $[0, 1]$  [Eberhart & Shi 2001b].
4. **CMA-ES**: There are 8 parameters to be predefined for this algorithm. All

#### **Chapter 4: A New Structure for Particle Swarm Optimisation**

settings followed the instructions given in [Hansen 2007]. For instance, the population size  $\lambda = 4 + \text{floor}(3 \times \ln D)$ , the parent number  $\mu = \text{floor}(\lambda/2)$ , etc., where  $\text{floor}(x)$  is the function that allows to round-off  $x$  to the nearest integer towards  $-\infty$ .

5. **DE**: The DE/Rand/1 scheme was employed. The parameter settings followed the instructions in [Storn 1996]. The population size  $N = 10 \times D$ ; the crossover probability  $CR = 0.9$  and the weighting factor  $F = 0.8$ .
6. **G3+PCX**: Following the papers by [Deb *et al.* 2002; Deb 2005], the population size  $N = 10 \times D$ ; the parent size was set to 3; the offspring size was set to 2 and the replacement size was set to 2. For the PCX operator, the distribution parameter  $\sigma_x = 0.1$  and  $\sigma_v = 0.1$ .

For all the PSO algorithms, the maximum velocity vector  $v_{\max} = [v_{\max 1}, v_{\max 2}, \dots, v_{\max D}] = [0.5 \times (x_{\max 1} - x_{\min 1}), 0.5 \times (x_{\max 2} - x_{\min 2}), \dots, 0.5 \times (x_{\max D} - x_{\min D})]$ , where  $x_{\max d}$  and  $x_{\min d}$  are the maximal and minimal bounds for the  $d$ th variable.

For every individual experiment, the result was obtained after 20 runs. For each run, the maximal function evaluation number was set to  $10^6$  and the optimisation process was regarded as successful and stopped, when the best solution  $Fb$  satisfied:  $Fb < 10^{-5}$  if the true global minimum  $Gb = 0$  or  $|(Fb - Gb) / Gb| < 10^{-5}$  if  $Gb \neq 0$ .

#### **Chapter 4: A New Structure for Particle Swarm Optimisation**

Table 4-9 shows the optimisation results of different algorithms on various problems. From this table, one can observe the following:

1. For unimodal problems  $f_1$  to  $f_4$ , CMA-ES performs best with the fewest function evaluations; nPSO and G3+PCX perform second best among the algorithms.
2. For high-dimensional multimodal problems  $f_5$  to  $f_{11}$ , nPSO performs better than other algorithms in most of the situations. For instance, for  $f_7, f_8$  and  $f_{10}$ , nPSO is able to locate the global optimum with the fewest function evaluations; for  $f_5, f_6$  and  $f_{11}$ , though nPSO cannot find the global optimum, but among the unsuccessful algorithms, it still performs best with the minimal fitness values.
3. For low-dimensional multimodal problems  $f_{12}$  to  $f_{14}$ , nPSO is able to obtain the global optimum and performs moderate, compared with other algorithms.

Compared nPSO with RSSA, in general, nPSO executes faster while RSSA is more capable in finding global optimum.

### Chapter 4: A New Structure for Particle Swarm Optimisation

Table 4-9. Average performance of various algorithms in optimising  $f_1$  to  $f_{14}$ : The integer in every cell is the average function evaluation number in successful runs; the value in parenthesis is the average result in the unsuccessful runs; the percentage value in the square brackets indicates the percentage of the successful runs out of all the runs; the bold values represent the best results.

Function	nPSO	PSO-LD	PSO-RV	CMA-ES	DE	G3+PCX
$f_1$	4263 (N/A) [100%]	447150 (N/A) [100%]	101160 (N/A) [100%]	<b>3207</b> (N/A) [100%]	391770 (N/A) [100%]	7140 (N/A) [100%]
$f_2$	18660 (N/A) [100%]	503957 (13.3333) [70%]	180420 (N/A) [100%]	<b>11751</b> (N/A) [100%]	655110 (N/A) [100%]	N/A (12.0469) [0%]
$f_3$	125486 (N/A) [100%]	861350 (79.1666) [20%]	958580 (20.0000) [45%]	<b>10830</b> (N/A) [100%]	N/A (1.8527) [0%]	25937 (N/A) [100%]
$f_4$	248638 (N/A) [100%]	813120 (N/A) [100%]	873960 (N/A) [100%]	<b>8929</b> (N/A) [100%]	N/A (0.2004) [0%]	117414 (N/A) [100%]
$f_5$	N/A (0.0023) [0%]	N/A (24.4833) [0%]	N/A (31.2336) [0%]	<b>46072</b> (N/A) [100%]	N/A (0.0158) [0%]	140430 (N/A) [100%]
$f_6$	N/A (-10840) [0%]	N/A (-9340) [0%]	N/A (-10202) [0%]	N/A (-6665) [0%]	<b>616080</b> (N/A) [100%]	N/A (-6878) [0%]
$f_7$	<b>55341</b> (N/A) [100%]	N/A (46.0341) [0%]	N/A (30.8072) [0%]	N/A (106.1617) [0%]	940560 (N/A) [100%]	N/A (142.8754) [0%]
$f_8$	<b>37208</b> (N/A) [100%]	535500 (N/A) [100%]	193650 (N/A) [100%]	8575 (19.3625) [40%]	694560 (N/A) [100%]	N/A; (3.1199) [0%]
$f_9$	52024 (0.0222) [55%]	493425 (0.0193) [40%]	146983 (0.0148) [60%]	<b>5586</b> ( <b>0.0100</b> ) [75%]	586740 (N/A) [100%]	10983 (0.0110) [65%]
$f_{10}$	<b>23363</b> (N/A) [100%]	485389 (225.6000) [90%]	143130 (N/A) [100%]	N/A (2.2897) [0%]	510180 (N/A) [100%]	N/A (15.1530) [0%]
$f_{11}$	N/A (0.0012) [0%]	N/A (30.3990) [0%]	<b>228900</b> ( <b>21.8394</b> ) [10%]	N/A (248.84) [0%]	N/A (0.1217) [0%]	N/A (184.4355) [0%]
$f_{12}$	2715 (N/A) [100%]	15328 (N/A) [100%]	842 (N/A) [100%]	<b>221</b> ( <b>-19.8160</b> ) [95%]	853 (N/A) [100%]	N/A (-0.4128) [0%]
$f_{13}$	3601 (N/A) [100%]	10591 (N/A) [100%]	861 (N/A) [100%]	<b>224</b> (N/A) [100%]	1182 (N/A) [100%]	N/A (0.8862) [0%]
$f_{14}$	3373 (N/A) [100%]	14896 (N/A) [100%]	898 (N/A) [100%]	<b>253</b> ( <b>141.0000</b> ) [95%]	777 (N/A) [100%]	N/A (35.3369) [0%]

## ***Chapter 4: A New Structure for Particle Swarm Optimisation***

---

### **4.4 Extension of nPSO to Multi-objective Optimisation Problems**

To extend the nPSO algorithm for the multi-objective optimisation case, the same method, which has been introduced in Section 3.4 to organise MO-RSSA, is employed here. In this new developed algorithm, which is named the new multi-objective PSO (nMPSO), both the Random Weighted Aggregation (RWA) technique [Murata *et al.* 1996] and the designed archiving mechanism (see Section 3.4.2) are employed.

#### **4.4.1 Experiments Based on the ZDT Series Problems**

A comparative study between nMPSO and other multi-objective optimisation algorithms is conducted using the well-known multi-objective optimisation problems - ZDT series benchmark problems [Zitzler *et al.* 2000] (see Section 3.5.1.1). The other multi-objective algorithms related to the Multi-Objective Reduced Space Searching Algorithm (MO-RSSA) (see Section 3.4), the Pareto Archived Evolution Strategy (PAES) [Knowles & Corne 2000], the Strength Pareto Evolutionary Algorithm (SPEA) [Zitzler & Thiele 1998] and the Non-dominated Sorting Genetic Algorithm II (NSGA-II) [Deb *et al.* 2002].

To compare the performance of nMPSO with such algorithms, the two performance metrics, namely the Generational Distance ( $GD$ ) and the Spread ( $\Delta$ ) (see Equations (3.6) and (3.7) in Section 3.5.1.2), are used.

#### **Chapter 4: A New Structure for Particle Swarm Optimisation**

In the following experiments, the maximum function evaluation for every experiment was set to 25000. The result of every experiment was obtained after 10 independent runs. The parameter settings for different algorithms were listed as follows:

1. **nMPSO**: 10 particles, maximum generation 2500,  $\omega = 10^{-3}$ , scaling parameters  $m_1 = 0.5$  and  $m_2 = 2$ , acceleration coefficients  $c_1 = c_2 = 1.5$ , frequency parameter  $H = 2000$  and one-directional refresh strategy for  $Pos_i$  setting.
2. **MO-RSSA**: decreasing parameter  $C_1 = 3$ , increasing parameter  $C_2 = 1$ , changing ratio  $K = 0.5$ ,  $m = 15$ , frequency parameter  $H = 10000$  and a variation strategy of the combination of the one-dimensional variation (with the 75% probability of usage) and the multi-dimensional variation (with the 25% probability of usage).
3. **PAES**: Population size 100, maximum generation 250, crossover probability 0.9 and mutation probability 0.01.
4. **SPEA**: Population size 80, external population size 20, maximum generation 250, crossover probability 0.9 and mutation probability 1/80.
5. **NSGA-II (binary-coded)**: Population size 100, maximum generation 250, crossover probability 0.9 and mutation probability 1/(string-length). 30 bits

#### **Chapter 4: A New Structure for Particle Swarm Optimisation**

were used to code each variable.

The average  $GD$  and  $\Delta$  values of 10 runs and the corresponding variances  $\sigma^2$  are summarised in Tables 4-10 and 4-11 respectively. In these tables, the compared results for PAES, SPEA and NSGA-II are obtained from [Deb 2001]. The following remarks can be made:

1. nMPSO and MO-RSSA outperform the other three salient EAs in terms of both accuracy and diversity.
2. For ZDT1, ZDT2 and ZDT3, nMPSO can achieve more accurate Pareto solutions than MO-RSSA, while MO-RSSA performs better in optimising ZDT4.
3. Both nMPSO and MO-RSSA can obtain the Pareto solutions with a good diversity. Between these two algorithms, the results that nMPSO obtained show a better distribution.

Figure 4-3 shows the graphical result produced by nMPSO. The true Pareto fronts of the problems are represented with continuous curves and the 'round' dots are the solutions obtained using nMPSO. It can be seen that the nMPSO can locate the true Pareto fronts with very good diversities in a consistent way.

### Chapter 4: A New Structure for Particle Swarm Optimisation

Table 4-10. Mean and variance values of  $GD$ ; the bold values represent the best results.

Problems	Mean value of $GD$ / (variance $\sigma^2$ )				
	nMPSO	MO-RSSA	PAES	SPEA	NSGA-II
ZDT1	<b>3.29e-4</b> (0)	4.07e-4 (0)	8.21e-2 (8.68e-3)	1.25e-3 (0)	8.94e-4 (0)
ZDT2	<b>2.66e-4</b> (0)	7.37e-4 (0)	1.26e-1 (3.69e-2)	3.04e-3 (2.00e-5)	8.24e-4 (0)
ZDT3	<b>3.29e-4</b> (0)	4.07e-4 (0)	2.39e-2 (1.00e-5)	4.42e-2 (1.90e-5)	4.34e-2 (4.20e-5)
ZDT4	4.90e-4 (0)	<b>1.82e-4</b> (0)	8.55e-1 (5.27e-1)	9.51e+0 (1.13e+1)	3.23e+0 (7.31e+0)

Table 4-11. Mean and variance values of  $\Delta$ ; the bold values represent the best results.

Problems	Mean value of $\Delta$ / (variance $\sigma^2$ )				
	nMPSO	MO-RSSA	PAES	SPEA	NSGA-II
ZDT1	<b>3.96e-1</b> (1.48e-3)	4.68e-1 (1.66e-3)	1.23e+0 (4.84e-3)	7.30e-1 (9.07e-3)	4.63e-1 (4.16e-2)
ZDT2	<b>3.76e-1</b> (2.63e-3)	3.94e-1 (8.10e-4)	1.17e+0 (7.68e-3)	6.78e-1 (4.48e-3)	4.35e-1 (2.46e-2)
ZDT3	<b>4.84e-1</b> (2.40e-3)	4.97e-1 (5.30e-3)	7.90e-1 (1.65e-3)	6.66e-1 (6.66e-4)	5.76e-1 (5.08e-3)
ZDT4	<b>3.86e-1</b> (7.51e-4)	4.78e-1 (2.62e-3)	8.70e-1 (1.01e-1)	7.32e-1 (1.13e-2)	4.79e-1 (9.84e-3)



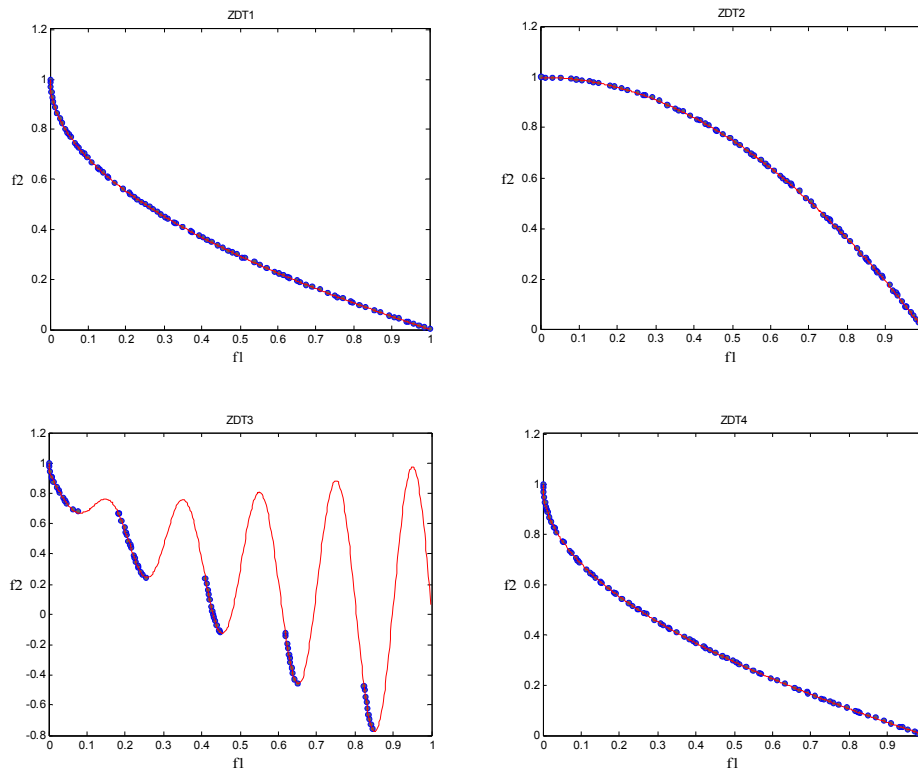


Figure 4-3. Pareto fronts obtained by nMPSO

#### 4.4.2 Experiments Based on the DTLZ Series Problems

In this section, nMPSO was used to solve the DTLZ series optimisation problems [Deb *et al.* 2001]. All the DTLZ problems were set to include three objectives and nMPSO used the same number of function evaluations as the experiments in [Deb *et al.* 2001]. The configuration of the nMPSO algorithm was set to be similar to the one used in optimising the ZDT problems, except the weight changing frequency parameter  $H = 1000$ .

## Chapter 4: A New Structure for Particle Swarm Optimisation

1. **DTLZ1:** The difficulty of this problem lies in the convergence to the hyper-plane. The search space contains  $(11^5-1)$  local Pareto-optimal fronts. Figure 4-4 shows the optimisation results using nMPSO in different angles of view. One can see that, nMPSO can obtain the Pareto-optimal solutions that possess a very good diversity.

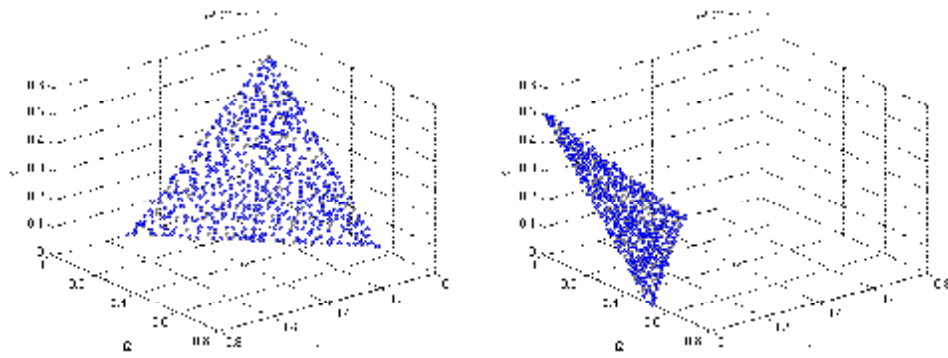


Figure 4-4. The optimisation result of nMPSO on DTLZ1

2. **DTLZ2:** The difficulty of this problem relates to its concave Pareto-optimal area. Figure 4-5 shows the optimisation results using the nMPSO algorithm. It can be seen that nMPSO can converge to the Pareto-optimal front with a good diversity.

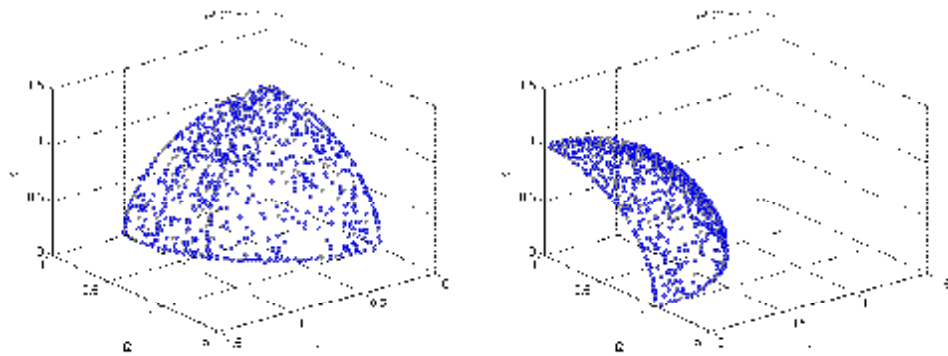


Figure 4-5. The optimisation result of nMPSO on DTLZ2

## ***Chapter 4: A New Structure for Particle Swarm Optimisation***

---

3. **DTLZ3**: The difficulty of this problem lies in the presence of several local Pareto-optimal fronts. This problem has  $3^{10}-1$  local Pareto-optimal fronts and a multi-objective optimisation algorithm can easily get stuck at any of these local Pareto-optimal fronts during the optimisation process. The results in [Deb *et al.* 2001] show that NSGA-II and SPEA2 can become trapped at some of these local Pareto-optimal fronts and cannot converge to the true Pareto front, while it can be seen from Figure 4-6 that nMPSO can achieve the true Pareto-optimal front.

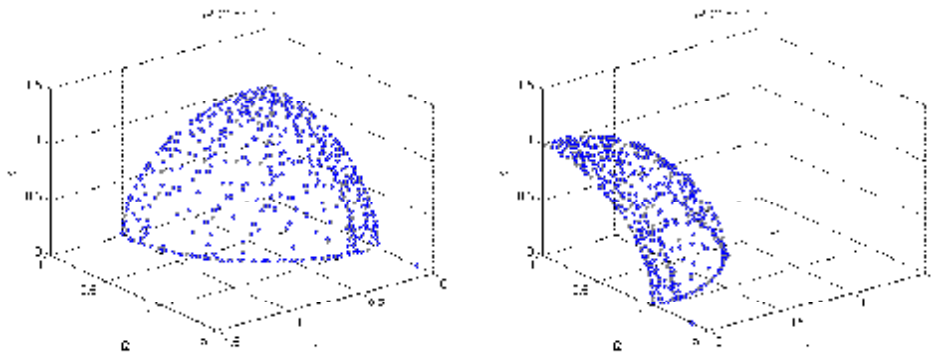


Figure 4-6. The optimisation result of nMPSO on DTLZ3

4. **DTLZ5**: This problem will test MOEA's ability to converge to a curve. From Figure 4-7, one can see that nMPSO is able to converge to the true Pareto-optimal front with a good diversity. For this problem, nMPSO can find the Pareto-optimal solutions in the early stage of the optimisation process.

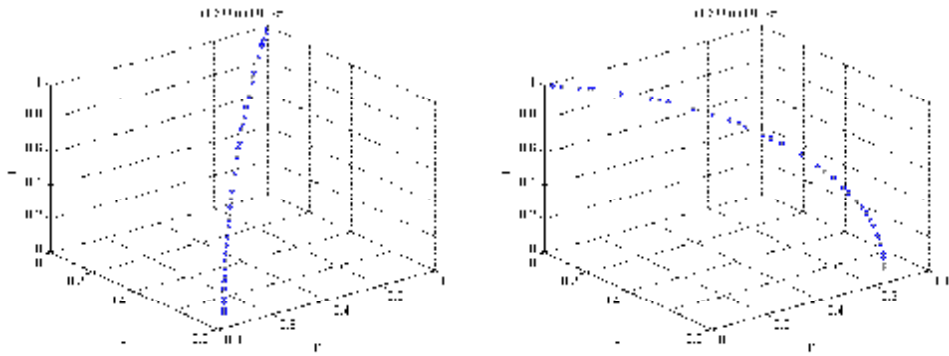


Figure 4-7. The optimisation result of nMPSO on DTLZ5

5. **DTLZ6:** This problem is a modified version of DTLZ5, which becomes more difficult to solve than DTLZ5. In [Deb *et al.* 2001], NSGA-II and SPEA2 cannot converge to the true Pareto-optimal front, as well as the proposed MO-RSSA (see Section 3.5.2). But nMPSO can achieve the true Pareto-optimal front, which is shown in Figure 4-8.

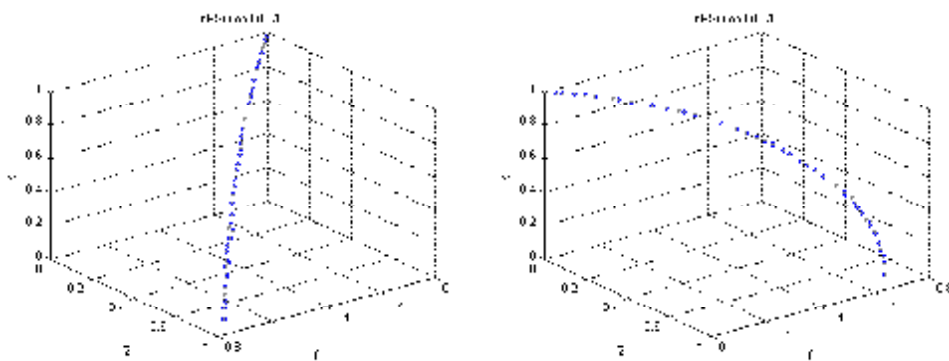


Figure 4-8. The optimisation result of nMPSO on DTLZ6

6. **DTLZ7:** This problem has a disconnected set of Pareto-optimal regions and it will test the algorithm's ability to maintain a subpopulation in

## ***Chapter 4: A New Structure for Particle Swarm Optimisation***

---

different Pareto-optimal regions. Figure 4-9 shows the optimisation result using nMPSO to optimise DTLZ7. It can be observed that the new algorithm is able to find and maintain stable and distributed subpopulations in all four disconnected Pareto-optimal regions.

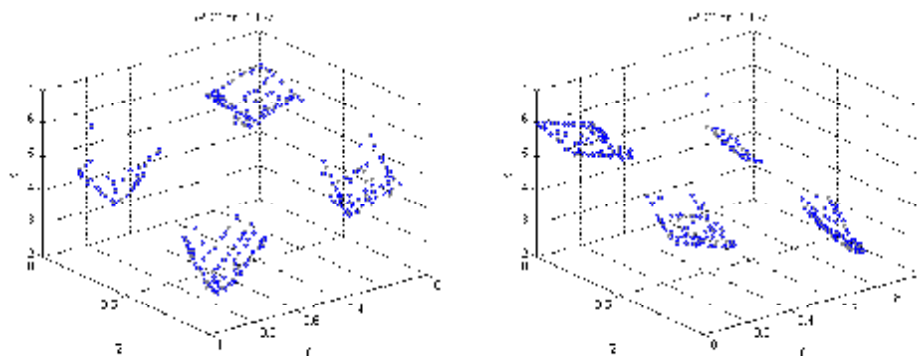


Figure 4-9. The optimisation result of nMPSO on DTLZ7

### **4.5 Summary**

In this chapter, a new structure for the particle swarm optimisation algorithm (nPSO) is proposed, which introduces a new ‘momentum term’ to replace the original inertia term. The advantages of nPSO compared with the standard PSOs lie in its ability of avoiding premature convergence and its adaptability in different situations. This algorithm was validated using a set of benchmark problems and is shown to lead to a better performance than the standard PSOs and some other salient optimisation algorithms. In addition, various parameter settings have been well explored and the appropriate parameter configurations were also

#### ***Chapter 4: A New Structure for Particle Swarm Optimisation***

---

recommended. Furthermore, the new algorithm was extended to become a multi-objective optimisation algorithm, nMPSO, by using the random weighted aggregation technique and maintaining an archiving mechanism. A comparative study between nMPSO and other MOEAs, such as PAES, SPEA and NSGA-II, was carried-out based on the ZDT series problems. The nMPSO was then employed to optimise the challenging DTLZ problems. The experimental results led to the conclusion that nMPSO is effective in finding the Pareto-optimal solutions and possesses advantages over some evolutionary algorithms in terms of the accuracy and the diversity of the final solutions.

The next chapter will propose a new modelling approach which will utilise multi-objective optimisation to reconcile model transparency with model accuracy.

## **Chapter 5**

# **A Fuzzy Modelling Approach with a Hierarchical Clustering Algorithm and a Multi-Objective Optimisation Mechanism (FM-HCMO)**

### **5.1 Introduction**

For modelling problems, accurate mathematical models do not always exist or it is difficult to derive them for all complex environments [Rojas *et al.* 2000], while at the same time, the available data that represent input-output relationships may be abundant. In this situation, model-free techniques such as artificial neural networks (NNs) and data-driven fuzzy modelling are suitable for model construction. Compared with black-box approaches such as artificial neural networks, fuzzy systems have white-box characteristics and are more transparent to humans to

## ***Chapter 5: FM-HCMO***

---

interpret. This is one of the most attractive aspects that distinguish fuzzy modelling from other black-box approaches. Nowadays, fuzzy rule-based systems are being applied to a growing number of significant applications in a wide variety of fields ranging from pattern recognition, data mining, classification, prediction, non-linear system approximation, and process control [Mamdani & Assilian 1974; Bezdek 1981; Takagi & Sugeno 1985; Jang 1993; Sugeno & Yasukawa 1993; Wang 1994; Marsili-Libelli & Muller 1996; Delgado *et al.* 1997].

In this chapter, a systematic data-driven fuzzy modelling methodology is proposed, which allows to construct Mamdani (linguistic) fuzzy models considering both accuracy (precision) and transparency (interpretability) of fuzzy systems. In this methodology, a hierarchical clustering algorithm is employed for the initial fuzzy model generation; a data selection mechanism is developed for selecting appropriate and efficient training data; and a multi-objective optimisation mechanism is developed for the fuzzy model improvement, which takes into account both accuracy and interpretability performance of fuzzy models.

This chapter is organised as follows. The remaining part of Section 5.1 provides a background knowledge about fuzzy modelling. Section 5.2 introduces the framework of the proposed modelling approach. Section 5.3 describes the proposed approach that concerns the generation of an initial fuzzy model using a hierarchical clustering algorithm. The related experimental studies are also presented. In Section 5.4, the proposed accuracy improvement and data selection mechanism are introduced, while in Section 5.5, a multi-objective optimisation



## ***Chapter 5: FM-HCMO***

---

algorithm, which includes both the accuracy and the interpretability performance of fuzzy models as attributes, is presented. In Section 5.6, a post-modelling technique for computing the model confidence bands is proposed. Finally, Section 5.7 concludes this chapter.

### **5.1.1 Fuzzy Systems**

The fundamental concept of fuzzy systems was first introduced by Zadeh in 1965 [Zadeh 1965] and later expanded upon in 1973 [Zadeh 1973]. Since then, fuzzy systems continued to receive more and more attention from a wide section of the research community. The main advantages of fuzzy systems consist of the following:

1. Fuzzy systems are interpretable (transparent). They include an explicit knowledge representation in the form of linguistic ‘If-Then’ rules, which can easily be understood and explained by humans to allow them to gain a deeper insight into complex and ill-defined systems.
2. Fuzzy systems are capable of handling complex, nonlinear, and uncertain problems and yet still able to exhibit robust behaviour. They have proven to be universal approximators able to perform nonlinear mappings between inputs and outputs [Wang & Mendel 1992].

3. Fuzzy systems are relatively easier to design and relatively inexpensive to implement.

### **5.1.2 Fuzzy Modelling**

Fuzzy modelling is a systems modelling with a fuzzy rule-based system. It is an approach that allows to model systems using a descriptive language (linguistic ‘If-Then’ rules, for instance) based on fuzzy logic and on fuzzy quantisation. Currently, fuzzy modelling is increasingly becoming a serious contender for the modelling of complex, nonlinear, and uncertain systems.

There are two complementary approaches in fuzzy modelling, namely knowledge acquisition from human experts and knowledge discovery from data. The knowledge acquisition approach lends itself to the design fuzzy models based on existing expert knowledge. This is the natural and classical method, however, it is not a trivial task for experts are not always available, and even if they are available, the cost of obtaining such expert knowledge may sometimes be too high or the knowledge itself is not consistent, and/or complete. On the other hand, knowledge discovery from data, i.e. ‘data-driven’ fuzzy modelling, will enable one to identify the structure and the parameters of fuzzy models from numerical data automatically. This can help to overcome the limitations of the ‘knowledge acquisition’ approach. The last two decades have witnessed an explosive growth in both the generation and the collection of data. By analysing and summarising these

## ***Chapter 5: FM-HCMO***

---

data, one can extract knowledge from such data-driven information. Experts may possess parts of the knowledge, but the other parts are totally new to humans. From this viewpoint, data-driven fuzzy modelling can be viewed as a process of discovering new knowledge expressed in the form of linguistic fuzzy statements (rules).

### **5.1.3 Data-Driven Fuzzy Modelling**

Broadly, the design of a data-driven fuzzy system can be a two-step process. The first step consists of generating a crude approximation of the fuzzy model that describes the system. This can be achieved via two methods: The first method uses a grid-partitioning of the multidimensional space. This partitioning can be generated from data or obtained from experts; it defines a number of fuzzy sets for each variable, which are interpreted as linguistic labels and shared by all the rules. Meanwhile, a training procedure optimises the grid structure, as well as the rule consequences, according to data samples. In contrast, the second method includes data clustering (grouping). In this method, the training data are gathered into homogeneous groups and a rule is associated to each group. The fuzzy sets are not shared by all the rules, but each set is mapped into one particular rule.

The second step consists of optimising the initial fuzzy sets and the initial rules to lead to a finally retained fuzzy model. The main techniques for this work include linear least-squares, nonlinear optimisation methods and machine learning based

## **Chapter 5: FM-HCMO**

---

techniques. Two of the most successful attempts to implement these learning and optimisation methods relate to neuro-fuzzy systems [Jang *et al.* 1997; Nauck *et al.* 1997; Fuller 1999] and evolutionary fuzzy systems [Pedrycz 1997; Sanchez *et al.* 1997; Cordon *et al.* 2001].

Neuro-fuzzy systems view fuzzy systems as a particular type of neural networks and employ related neural networks' training techniques, such as the Back-Propagation algorithm (BP) [Rumelhart *et al.* 1986], to improve the parameters of the fuzzy sets. On the other hand, evolutionary fuzzy systems employ evolutionary algorithms, such as Genetic Algorithms (GAs) and Evolution Strategies (ESs), to improve the initial fuzzy systems, because of their capability for searching relatively large multidimensional solution spaces. Unlike neuro-fuzzy systems, evolutionary fuzzy systems are able to realise improvements not only on the parameters of the fuzzy sets but also on the structure of the fuzzy rules. Moreover, multi-objective optimisation techniques within the evolutionary algorithms can prove very helpful in studying the trade-off between the accuracy and the interpretability of fuzzy models. Using a multi-objective evolutionary algorithm, the accuracy and the interpretability of models can be incorporated into several objectives in order to evaluate the final multiple Pareto-optimal solutions, each of which represents an individual fuzzy model.

### **5.1.4 Accuracy and Interpretability**

Fuzzy modelling may be conducted under two contradictory requirements: accuracy (significance) and interpretability (transparency). Accuracy is easy to come to grips with as it relates to the capability of representing the real system faithfully, this being a fundamental requirement for models. In contrast, interpretability means that human beings are able to understand a fuzzy system's behaviour by inspecting its associated rule-base(s). It is crucial in the field of data mining and knowledge discovery where information should be extracted from data bases and represented in a comprehensible form, or for decision support systems where the reasoning process should be transparent to the users [Mikut *et al.* 2005].

Whereas the definition of accuracy in a certain application is straightforward, the definition of interpretability is rather problematic. Most researchers and practitioners should agree on interpretability involving the following aspects [Mikut *et al.* 2005]:

1. The number of rules should be small enough to be comprehensible.
2. The rule-base should be formed from the rules describing the relevant local relationships. Furthermore, the rules should be consistent (similar premises lead to similar conclusions).
3. The rule premises should be simple in structure and contain only a few

## **Chapter 5: FM-HCMO**

---

features (input variables).

4. The linguistic terms should be intuitively comprehensible. The form and parameters of the membership functions should correspond to the understanding of the linguistic expressions.
5. The inference mechanism should produce technically and intuitively correct results.

It is obvious that the ideal scenario would be for the model to satisfy both the accuracy and the interpretability criteria to the highest possible degrees but, since these are contradictory attributes, this scenario is generally not possible. In recent years, many researches have been devoted to the study of the trade-offs between accuracy and interpretability [Guillaume 2001; Jimenez *et al.* 2001; Ishibuchi & Yamamoto 2003; Wang *et al.* 2005; Kim *et al.* 2006; Gonzalez *et al.* 2007].

### **5.1.5 Optimisation of Fuzzy Modelling**

From the viewpoint of optimisation, the design of a fuzzy model can be formulated as a search problem in a multidimensional space where each solution represents a possible fuzzy model with different rule structures, membership functions, and related parameters. According to specific performance criteria given by the designer, the performance of different fuzzy models forms a hypersurface. Thus,

## ***Chapter 5: FM-HCMO***

---

designing an optimal fuzzy model can be considered as finding the optimal location of this hypersurface.

In most of the papers relating to fuzzy modelling problems which were published in 1990's, only the parameters of fuzzy models were optimised while the model structure itself was fixed [Horikawa *et al.* 1992; Jang 1993]. Since then, some other approaches [Setnes *et al.* 1998; Yen *et al.* 1998; Jin 2000; Jimenez *et al.* 2001; Ishibuchi & Yamamoto 2004; Wang *et al.* 2005] have been developed, in which both the parameters and the model structures were improved simultaneously by encoding all the information simultaneously into the solutions.

The fuzzy modelling approaches [Horikawa *et al.* 1992; Jang 1993; Cordon *et al.* 2001] that optimise the parameters can only be viewed as optimising the following single-objective problem:

$$\text{Maximise: } Accuracy(S), \quad (5.1)$$

where  $Accuracy(S)$  is an accuracy measure of the fuzzy system.

Figure 5-1 illustrates this optimal search. In this figure, the search direction is represented by the arrow; a single fuzzy system, which is represented by a green dot, is obtained as the final solution by maximising the accuracy. It can be seen that the final solution performs well in 'accuracy' but badly in 'interpretability', because accuracy improvements are usually achieved at the expense of interpretability. More specifically, accuracy improvements often increase the complexity of systems and tune the topology of fuzzy sets so that they become

overlapping.

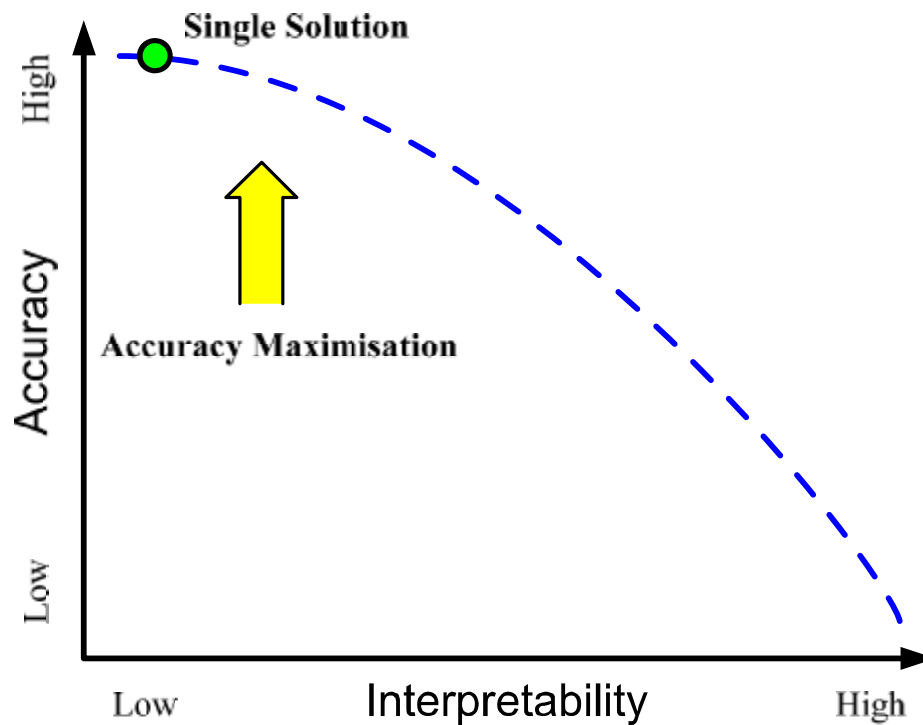


Figure 5-1. Modelling within the context of a Cartesian representation between accuracy and interpretability – the accuracy maximisation

In the approaches proposed in late 1990's [Setnes *et al.* 1998; Yen *et al.* 1998; Jin 2000], although both accuracy and interpretability were considered, these two objectives were combined to form one objective and the final solution will still represent a single fuzzy rule-based system. These studies can be viewed as optimising the following weighted sum of objective functions:

$$\text{Maximise: } w_1 \times \textit{Accuracy}(S) + w_2 \times \textit{Interpretability}(S), \quad (5.2)$$

where *Accuracy(S)* and *Interpretability(S)* are the accuracy and the interpretability measures of the fuzzy system;  $w_1$  and  $w_2$  are objective weights.



## Chapter 5: FM-HCMO

---

As shown in Figure 5-2, a single fuzzy model is obtained as the final solution by maximising the weighted sum of the objective functions. The search direction represented by the arrow in Figure 5-2 is specified by the weight vector  $[w_1, w_2]$ . In this situation, the specification of the weight vector is not easy and is problem-dependent. Thus, the decision maker must have prior knowledge of the priority of each objective before aggregating them into one single objective. If multiple solutions are needed, different priorities need to be assigned and the same problem needs to be solved again.

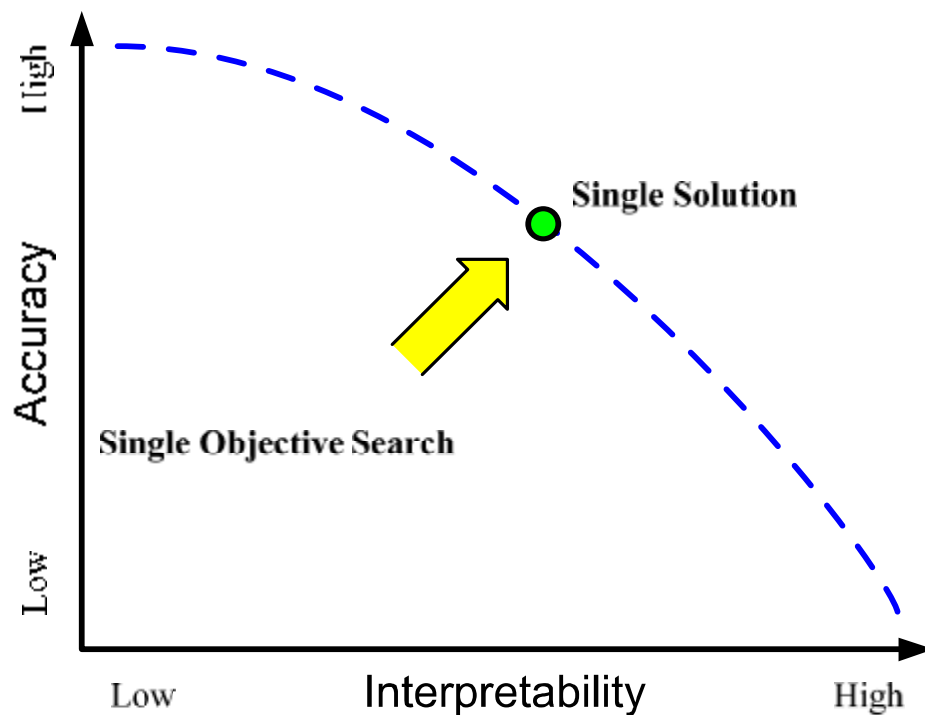


Figure 5-2. Modelling within the context of a Cartesian representation between accuracy and interpretability – the optimisation of the weighted sum of objective functions

In recent years, the trade-off between the accuracy and the interpretability of fuzzy

## **Chapter 5: FM-HCMO**

---

systems has been the subject of many published papers [Jimenez *et al.* 2001; Ishibuchi & Yamamoto 2004; Wang *et al.* 2005]. Because of such conflicting objectives, the fuzzy modelling problem has always been viewed as a multi-objective optimisation problem by its very nature. To this effect, multi-objective optimisation techniques, especially those allied to multi-objective evolutionary algorithms, have been considered as possible candidates for fuzzy modelling. These studies can be viewed as being equivalent to the following two-objective problem:

$$\begin{aligned} &\text{Maximise: } Accuracy(S) \text{ and,} \\ &\text{Maximise: } Interpretability(S). \end{aligned} \tag{5.3}$$

Figure 5-3 illustrates the multi-objective optimisation search process for a number of fuzzy models with various accuracy-interpretability trade-offs. The search direction can be in every feasible direction. The finally obtained solutions are called multiple Pareto-optimal (non-dominated) solutions. After obtaining these final solutions (models), the decision maker should be able to choose the most appropriate solution for the current situation. For example, some users may prefer a ‘simpler’ model with a high interpretability; other may prefer a ‘more complex’ one with a high accuracy. Moreover, the knowledge of several other Pareto-optimal solutions can also be useful for later use, when the current situation has changed and a new solution is required.

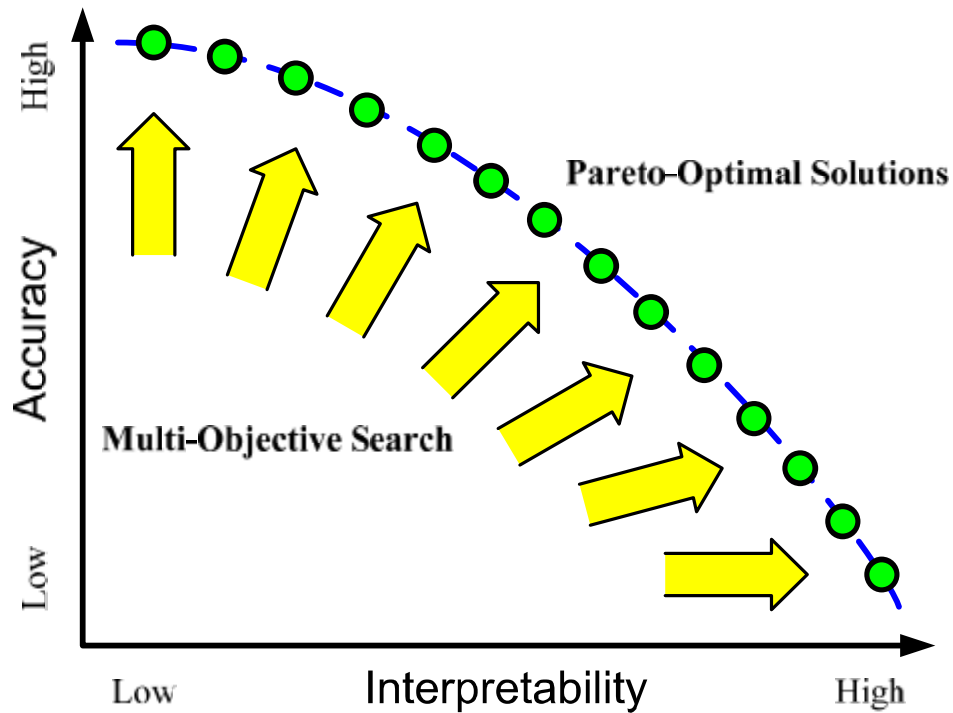


Figure 5-3. Modelling within the context of a Cartesian representation between accuracy and interpretability – the multi-objective optimisation

### 5.1.6 Features of the Proposed Modelling Methodology

The main aim of this chapter is to develop a systematic fuzzy modelling approach, which can be used for constructing a set of fuzzy models from sample data. The important features of this approach will consist of the following:

1. Mamdani fuzzy models (linguistic fuzzy models) are used in the proposed approach. Most recent research in the field of fuzzy modelling concentrated extensively on the use of Takagi-Sugeno-Kang (TSK) [Takagi & Sugeno 1985] fuzzy models because of their high accuracy. But compared to

## **Chapter 5: FM-HCMO**

---

Mamdani fuzzy models, TSK models are less transparent.

2. A hierarchical clustering algorithm, which is an agglomerative complete-link clustering algorithm, is employed to generate the initial fuzzy system. To reduce the computational complexity of this normally involved algorithm, a new version agglomerative complete-link clustering algorithm is devised. The algorithm should prove to be more efficient and perform better than other well-known clustering algorithms, such as the well-known fuzzy c-means (FCM) clustering algorithm [Dunn 1973; Bezdek 1981].
3. A data selection mechanism is proposed for selecting the appropriate data for training the models.
4. Both the accuracy and the interpretability of fuzzy models are included in the list of objective functions to be optimised. In this work, a multi-objective optimisation mechanism, which is based on the previously developed efficient optimisation algorithm nMPSO (see Chapter 4), is proposed and employed in order to obtain a set of Pareto-optimal fuzzy models with different accuracy and interpretability levels.
5. The proposed fuzzy modelling approach is developed to solve not only low-dimensional modelling problems but also high-dimensional modelling problems. The fuzzy modelling of high-dimensional systems is always

challenging because the model complexity grows exponentially as the input dimension increases. In this proposed approach, an efficient hierarchical clustering algorithm and a data selection mechanism are designed and a high-performance evolutionary computation based algorithm, nPSO (see Chapter 4), is also employed. All these paradigms cooperate together to tackle the high-dimensional modelling problem efficiently.

### **5.2 The Framework of the Proposed Modelling Methodology**

Figure 5-4 shows the flow chart of the proposed fuzzy modelling approach. This approach is named throughout the Fuzzy Modelling approach with a Hierarchical Clustering algorithm and a Multi-objective Optimisation mechanism (FM-HCMO). It can be divided into several parts and the execution steps can be described as follows:

1. **Data clustering:** A data clustering algorithm, the agglomerative complete-link clustering algorithm, is employed to process the training data in order to obtain the information of clusters.
2. **Initial model construction:** The information about clusters is then used to construct an initial fuzzy model.

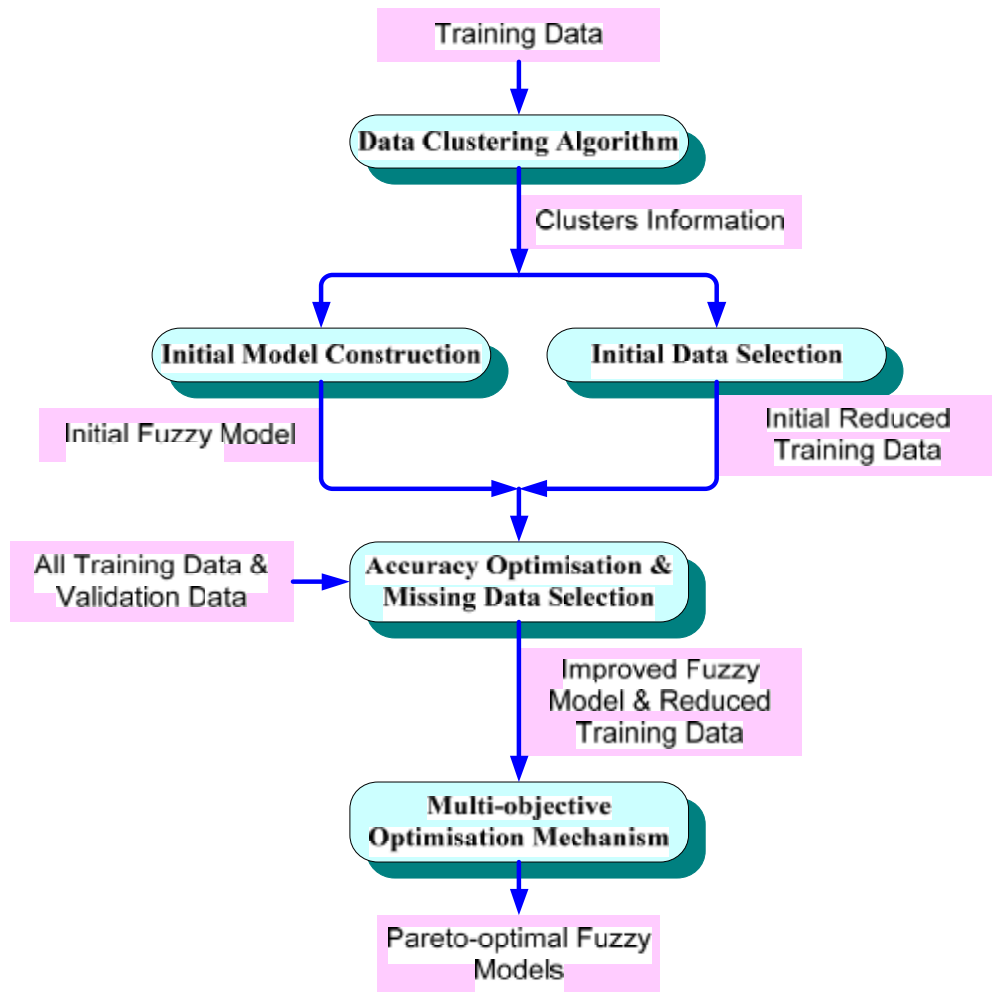


Figure 5-4. A framework for the proposed fuzzy modelling approach

3. **A crude data selection:** The information about clusters is also used for the selection of the training data. Following this operation, a representative training data set is selected.
4. **Accuracy optimisation and missing data selection:** In this step, the initial fuzzy model is improved in terms of accuracy and a further representative training data set is selected. After this step, an accurate fuzzy model is

obtained and a reduced training data set is formed.

5. **Multi-objective optimisation:** By using a multi-objective optimisation algorithm nMPSO, the previous fuzzy model is optimised according to the accuracy and interpretability objectives. Finally, a set of Pareto-optimal fuzzy models should be obtained.

### **5.3 Data Clustering and Initial Fuzzy Model Construction**

#### **5.3.1 Data Clustering**

##### **5.3.1.1 Basic Concepts**

Clustering is an unsupervised form of classification of data (patterns, observations, or feature vectors) into different clusters (groups). Each cluster consists of data that are similar between themselves and dissimilar to those data of other clusters. In other words, the data in one cluster share some common traits, which are often related to some defined distance measure. On the other hand, data clustering can also be viewed as a process of modelling data by its clusters. Representing many data by fewer clusters may lead to the loss of some details, but it achieves a simplification for data analysis.

An example of data clustering is depicted in Figure 5-5. In this example, the input

## Chapter 5: FM-HCMO

---

data are shown in Figure 5-5(a); the desired clusters are shown in Figure 5-5(b); the data points belonging to the same cluster are given the same label.

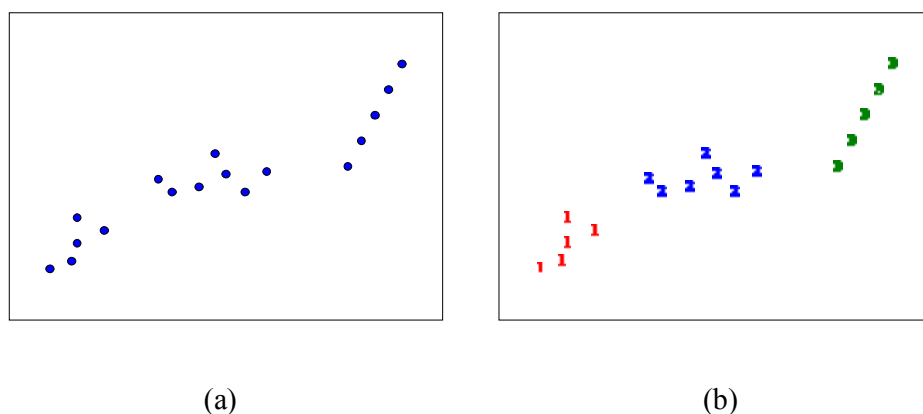


Figure 5-5. Examples of data clustering

It is important to get to grips with the difference between data clustering (unsupervised classification) and discriminant analysis (supervised classification) [Jain *et al.* 1999]. In supervised classification, a collection of labelled (pre-classified) patterns are provided; the problem is to label a newly encountered and unlabeled pattern. In this case, the given labelled patterns are used to learn the descriptions of classes which in turn are used to label a new pattern, while in the case of data clustering, the problem is to group a given collection of unlabelled patterns into meaningful clusters. In data clustering, labels are also associated with clusters, but these category labels are data driven; that is, they are obtained solely from the data.

In data clustering techniques, similarity is fundamental to the definition of a



cluster. The measure of similarity (distance measure) will influence the shape of the clusters [Jain *et al.* 1999]. Because of the variety of feature types and scales, the distance measure must be chosen carefully. Generally, the Euclidean distance (2-norm distance) is the most common distance measure and is widely used.

In recent years, data clustering has become a common technique for statistical data analysis and has been used in many fields, including data mining, machine learning, pattern recognition and image analysis.

### **5.3.1.2 Types of Data Clustering Algorithms**

Traditionally, clustering techniques have always been broadly divided into the hierarchical and the partitional categories. Hierarchical clustering is further subdivided as agglomerative or divisive. In hierarchical clustering, the clusters are built gradually as crystals are grown. More details about hierarchical clustering will be given in the next section.

On the other hand, partitional clustering infers the clusters directly and it determines all the clusters at once. In doing so, it either tries to discover clusters by iteratively relocating data between subsets, or attempts to identify them as areas which are highly populated with data. Algorithms of the first type (partition relocation clustering) concentrate on how well data fit into their clusters and tend to build clusters of proper convex shapes. Partitional clustering algorithms of the

## **Chapter 5: FM-HCMO**

---

second type (density-based clustering) try to discover dense connected components of data, which are flexible in terms of their shapes.

Hierarchical clustering based algorithms are more versatile than partitional clustering algorithms, whereas a typical partitional algorithm works well only on the data sets which include isotropic clusters [Nagy 1968]. For partition relocation clustering algorithms, they are simple and fast, but not accurate enough. For instance, they can not yield the same result with each run, since the resulting clusters depend on the initial random assignments. For density-based clustering algorithms, they usually work with low-dimensional data of numerical attributes, known as spatial data.

### **5.3.1.3 Hierarchical Clustering Algorithms**

Hierarchical clustering builds a cluster hierarchy or, in other words, a tree of clusters [Berkhin 2006]. Every cluster node contains child clusters and the sibling clusters classify the data points covered by their common parent. This tree can be cut at a desired similarity level to form a partition (clustering), which is identified by completely connected components in the corresponding tree. This approach allows exploring data on different levels of granularity.

For example in Figure 5-6 [Maksim 2006], suppose the data  $\{a\}$ - $\{f\}$  are to be clustered, and the Euclidean distance is the distance measure. The hierarchical

## Chapter 5: FM-HCMO

---

clustering dendrogram is as shown in Figure 5-7 [Sideris 2005]. For hierarchical clustering, cutting the tree at a given height (similarity) will give a clustering at a selected precision. In this example, cutting after the second row will yield the clusters  $\{a\}$   $\{b\ c\}$   $\{d\ e\}$   $\{f\}$ ; and cutting after the third row will obtain the clusters  $\{a\}$   $\{b\ c\}$   $\{d\ e\ f\}$ , which is a coarser clustering, with a smaller number of larger clusters.

Generally, hierarchical clustering methods can be categorised into agglomerative and divisive approaches. An agglomerative clustering method starts with one-point (singleton) clusters and recursively merges two or more most appropriate clusters. A divisive clustering method starts with one cluster of all data points and recursively splits the most appropriate cluster. The merging or splitting process continues until a stopping criterion (frequently, the requested number of clusters) is achieved.

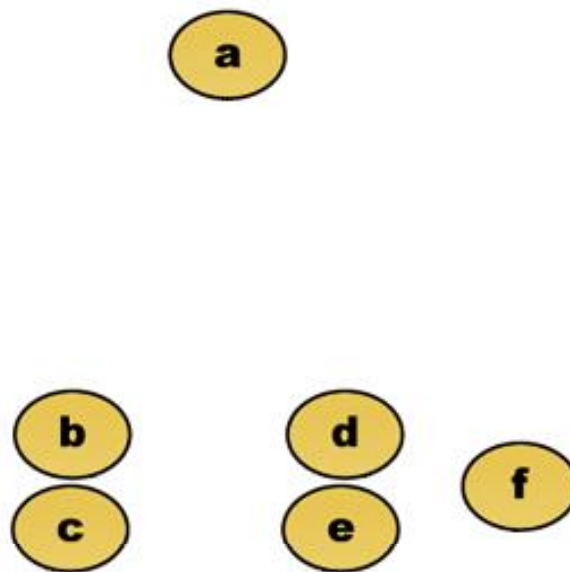


Figure 5-6. Data samples of a clustering problem

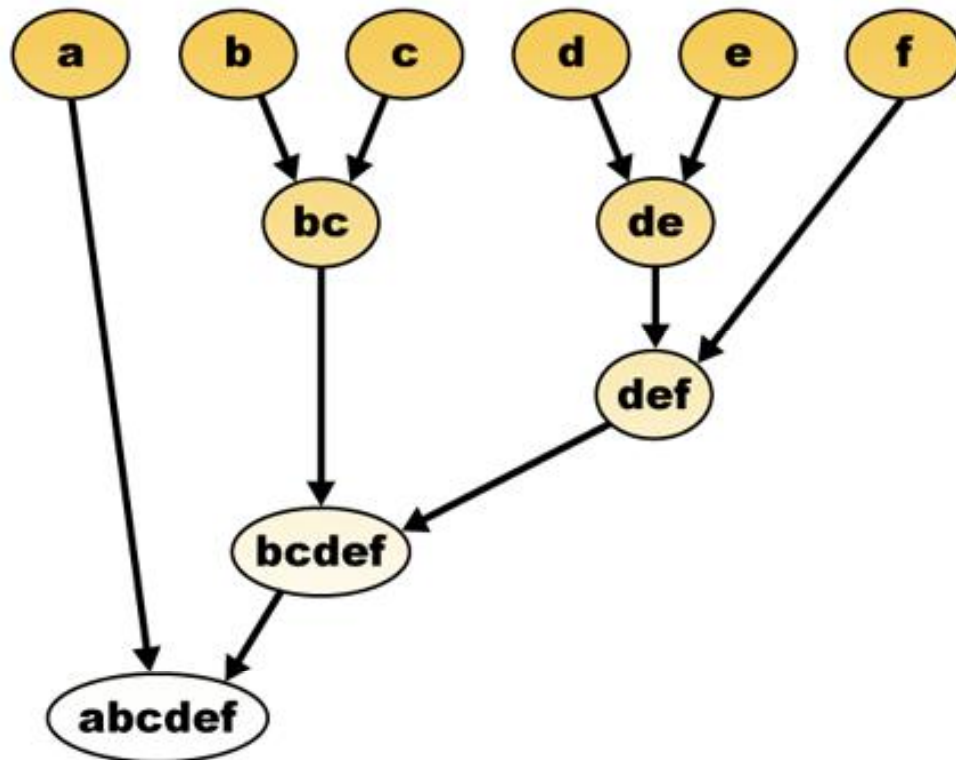


Figure 5-7. The hierarchical clustering dendrogram of the data samples shown in Figure 5-6

The hierarchical agglomerative clustering algorithm can be described as follows [Jain *et al.* 1999]:

1. Compute the proximity matrix containing the distance between each pair of patterns. Treat each pattern as a cluster.
2. Find the most similar pair of clusters using the proximity matrix. Merge these two clusters into one cluster. Update the proximity matrix after this merging operation.

## **Chapter 5: FM-HCMO**

---

3. If the stopping criterion is achieved, then stop; otherwise, go to Step 2.

In the light of the above, a variety of agglomerative algorithms can be designed. In particular, most hierarchical clustering algorithms can be divided into three categories, the single-link [Sneath & Sokal, 1973], complete-link [King 1967], and minimum-variance [Ward 1963; Murtagh 1984] algorithms, in which the single-link and complete-link algorithms are most popular. These algorithms differ in the way they characterise the similarity between a pair of clusters. For instance, in the single-link method, the distance between two clusters is the minimum of the distances between all pairs of patterns drawn from the two clusters (one pattern from the first cluster, the other from the second). In the complete-link algorithm, the distance between two clusters is the maximum of all pair wise distances between the patterns in the two clusters. In either case, two clusters are merged to form a larger cluster based on the minimum distance measure.

In [Baeza-Yates 1992], it is explained that the complete-link algorithm produces tightly bound or compact clusters. The single-link algorithm, by contrast, suffers from a chaining effect [Nagy. 1968]. It has a tendency to produce clusters that are straggly or elongated. From a pragmatic viewpoint, it has been observed that the complete-link algorithm produces more useful clusters in many applications than the single-link algorithm [Jain & Dubes 1988].

**5.3.1.4 The Proposed New Agglomerative Complete-Link Clustering**

**Algorithm**

As described in the previous section, the agglomerative complete-link clustering algorithm produces tight bounds and compact clusters. This characteristic makes the agglomerative complete-link algorithm more suitable for modelling purposes. Thus, this algorithm is employed to build the initial fuzzy model in this work.

However, hierarchical clustering algorithms, in general, suffer from a problem of high computational complexity. For example, if the number of training data is  $N$ , the computational complexity will be  $O(N^2)$ . To reduce the computational complexity of the original agglomerative complete-link algorithm, the following methodology is adopted:

1. Define the desired number of clusters  $N_c$  and a threshold  $N_{\max}$ . This threshold is an integer number used to estimate whether the training data set is too large for computation. To ensure that this method works adequately, the threshold is set as  $N_{\max} > N^{1/2}$ .
2. If  $N \leq N_{\max}$ , then apply the normal agglomerative complete-link clustering algorithm to classify the training data into  $N_c$  clusters, and terminate the whole clustering process; If  $N > N_{\max}$ , go to Step 3.
3. Divide the training data set randomly and equally into  $i$  groups.  $i = \text{Ceil}(N)$

## Chapter 5: FM-HCMO

---

$/ N_{\max}$ ), where  $Ceil(x)$  is a function that rounds off  $x$  to the nearest integer towards  $+\infty$ .

4. For every group out of  $i$  groups, classify the  $N/i$  training data into  $j$  sub-clusters using the normal agglomerative complete-link algorithm, where  $j = Floor(N_{\max} / i)$  and  $Floor(x)$  is a function that rounds off  $x$  to the nearest integer towards  $-\infty$ .
5. For every sub-cluster out of  $i \times j$  clusters, select the centric datum, which is closest to the cluster's centre among all the data in this sub-cluster, as the representative data.
6. Combine all the representative data to form a representative data set, in which the number of data is  $i \times j$  and smaller than  $N_{\max}$ .
7. Apply the normal agglomerative complete-link algorithm to clustering the representative data into  $N_c$  clusters.
8. Replace each representative datum with the original training data in its corresponding sub-cluster.

If the value of  $N_{\max}$  is smaller than  $N$ , the computational complexity of the proposed method becomes  $O(N \times N_{\max})$ , where  $O(N^{3/2}) < O(N \times N_{\max}) < O(N^2)$ . It can be seen that, by designing an adequate threshold value, the computational

complexity of this clustering method is as low as  $O(N^{3/2})$ . From this viewpoint, the proposed method should improve the clustering efficiency greatly; thus it will be used in the modelling process throughout.

### **5.3.2 Initial Fuzzy Model Construction**

By using the agglomerative complete-link clustering algorithm, a predefined number of clusters can be obtained from the training data. The information that these clusters will provide is then used to construct an initial fuzzy model. In this modelling approach, one cluster corresponds to one fuzzy rule; the centres of membership functions are defined using the information of their corresponding clusters' centre positions; other parameters relating to the membership functions are defined under the principle that one membership function must cover all the training data, which are included in its corresponding cluster.

#### **5.3.2.1 An Example of Constructing the Initial Fuzzy Model**

Figure 5-8 shows an example of how to construct initial fuzzy model from the information of clusters:

1. One fuzzy rule corresponds to one cluster. For instance, for the right upper cluster in Figure 5-8, the corresponding fuzzy rule should be: IF input is



## Chapter 5: FM-HCMO

---

$A_1^3$ , THEN output is  $A_2^3$ .

2. The membership functions' centres correspond to the clusters' centres. As shown in Figure 5-8, the green dashed-dotted lines represent the centres of clusters and they are used to define the membership functions' centres.
3. The membership functions' widths correspond to the clusters' widths, which are shown as red dashed lines in Figure 5-8. For the parameters that relate to the covering range of membership functions, they are defined under the clusters' width restriction.

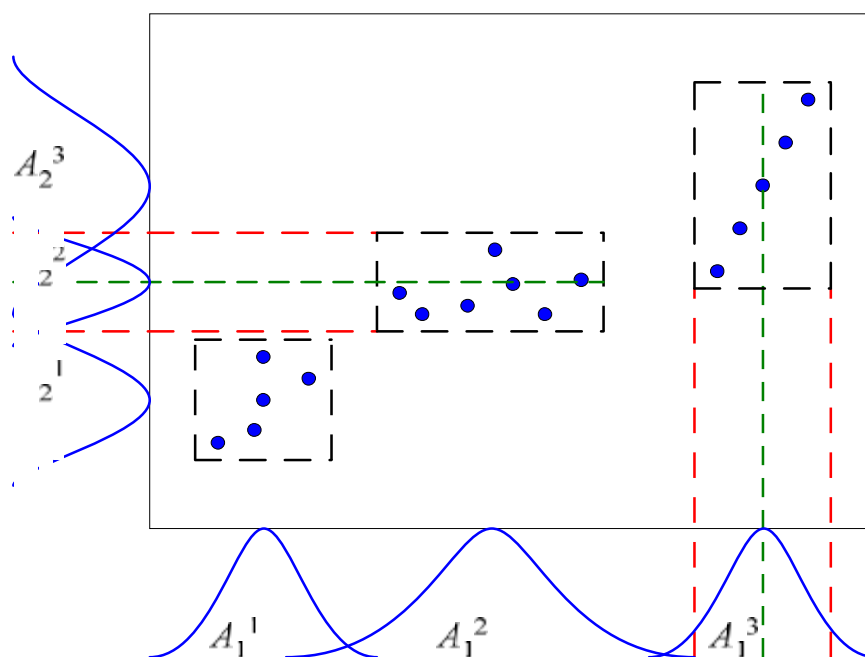


Figure 5-8. An example of the initial fuzzy model construction

## Chapter 5: FM-HCMO

---

### 5.3.2.2 Fuzzy System Definition and Notation

A generic multi-input and single-output (MISO) fuzzy model is represented as a collection of fuzzy rules in the following form:

Rule  $R_k$ : IF  $x_1$  is  $A_1^k$  and  $x_2$  is  $A_2^k$  ... and  $x_D$  is  $A_D^k$ , THEN  $y$  is  $B_k$ ,

where  $R_k$  is the label of the  $k$ th fuzzy rule;  $\mathbf{x} = [x_1, x_2, \dots, x_D]^T \in U_1 \times U_2 \times \dots \times U_D$  are input linguistic variables;  $A_l^k$  are the antecedent fuzzy sets of the universes of discourse  $U_l$ , where  $l = 1, 2, \dots, D$ ;  $y \in V$  is the output linguistic variable; and  $B_k$  is a consequent fuzzy set of the universe of discourse  $V$ .

In this work, Gaussian functions are chosen as the membership functions (without any loss of generality), i.e.:

$$\mu_A(x) = \exp\left(-\frac{(x-c)^2}{\sigma^2}\right), \quad (5.4)$$

where  $\mu_A$  is the membership function of  $x$  belonging to the fuzzy set  $A$ ; parameters  $c$  and  $\sigma$  represent the centre and the width of this membership function, where  $\sigma$  is a positive number. Besides this, the product inference engine and the centre average defuzzification method [Wang 1997] are also implemented in the fuzzy systems of this work.

### 5.3.2.3 The Fuzzy Model Extraction Approach

Assume a modelling problem being based on a collection of  $(D+1)$ -dimensional

## Chapter 5: FM-HCMO

---

input-output ( $D$ -input and 1-output) training data  $\{\mathbf{p}_1, \mathbf{p}_2, \dots, \mathbf{p}_N\}$ .  $\mathbf{p}_m = [x_1^m, x_2^m, \dots, x_D^m, y_m]^T$ , where  $m = 1, 2, \dots, N$ ;  $N$  is the number of training data.

By using the agglomerative complete-link clustering algorithm, a predefined number of clusters can be obtained from the training data. Let  $\mathbf{C}_n$  represent the  $n$ th cluster and  $\mathbf{C}_n = \{\mathbf{p}_{n1}, \mathbf{p}_{n2}, \dots, \mathbf{p}_{n(NDn)}\}$ , where  $n = 1, 2, \dots, N_c$  and  $NDn$  is the number of data in the  $n$ th cluster.

In this modelling approach, the rule-base is obtained and is composed of  $N_c$  fuzzy rules. The fuzzy rule corresponding to the cluster  $\mathbf{C}_n$  can be represented as follows:

$$R_n: \text{IF } x_1 \text{ is } A_1^n \text{ and } x_2 \text{ is } A_2^n \dots \text{ and } x_D \text{ is } A_D^n, \text{ THEN } y \text{ is } B_n.$$

where  $n = 1, 2, \dots, N_c$ ;  $\mathbf{x} = [x_1, x_2, \dots, x_D]^T$  are input linguistic variables;  $A_i^n$  are antecedent fuzzy sets, where  $i = 1, 2, \dots, D$ ;  $y$  is the output linguistic variable; and  $B_n$  is a consequent fuzzy set.

Considering one fuzzy set  $A_i^n$ , the Gaussian membership function of  $A_i^n$  includes two parameters  $c_i^n$  and  $\sigma_i^n$ .  $c_i^n$  can be calculated using the following equation:

$$c_i^n = \frac{\sum_{j=1}^{NDn} x_i^{nj}}{NDn}. \quad (5.5)$$

It is worth nothing that the membership function should cover all the training data contained in its corresponding cluster. In other words, for every data included in this cluster, its membership degree should be high enough to ensure the data maps into this rule. Based on this requirement, the membership parameter  $\sigma_i^n$  is designed

## Chapter 5: FM-HCMO

---

to satisfy the following equation:

$$\min_j \left( \mu_{A_i^n}(x_i^{nj}) \right) = \min_j \left[ \exp \left\{ - \frac{(x_i^{nj} - c_i^n)^2}{(\sigma_i^n)^2} \right\} \right] \geq Th, \quad (5.6)$$

where  $j = 1, 2, \dots, NDn$ . This equation means that, for all the data included in the  $n$ th cluster, the membership degrees are higher than a threshold  $Th$ . The value of  $Th$  is set to 0.5 in this work without any loss of generality. The previous equation can be rewritten as follows:

$$\sigma_i^n = \frac{\max_j (x_i^{nj} - c_i^n)}{\sqrt{-\ln(Th)}}, \quad (5.7)$$

where  $j = 1, 2, \dots, NDn$ . Using this equation, the parameter  $\sigma_i^n$  is determined.

### 5.3.3 Experimental Studies

#### 5.3.3.1 Comparison of Clustering Algorithms for Fuzzy Modelling Problems

For this comparison study, the fuzzy c-means (FCM) [Dunn 1973; Bezdek 1981] clustering algorithm, which has been used in many fuzzy modelling problems [Kim *et al.* 1997; Emami *et al.* 1998; Chen & Linkens 2001], and the agglomerative complete-link clustering algorithm proposed in this chapter, were applied to solve three modelling problems.

The first problem consists of approximating the following two-input and single-output nonlinear function, which is taken from [Sugeno & Yasukawa 1993]:

## **Chapter 5: FM-HCMO**

---

$$y = (1 + x_1^{-2} + x_2^{-1.5})^2, \quad (5.8)$$

where  $1 \leq x_1, x_2 \leq 5$ . The input data set consists of 50 data points  $[x_1^k, x_2^k]^T$ , where  $k = 1, 2, \dots, 50$ , and they are the same as the data points described in [Sugeno & Yasukawa 1993].

The second problem aims to model a mechanical property of alloy steels, the reduction of area (ROA). The ROA data include 15 inputs and one output and it is considered to be a high-dimensional problem for modelling purposes. In this experiment, 250 ROA data are used for modelling.

The third problem relates to modelling another mechanical property of alloy steels, the ultimate tensile strength (UTS). This problem is also a high-dimensional problem and includes 15 inputs and one output. In this experiment, 1000 UTS data are used in the modelling process.

In these experiments, the clustering algorithms were applied first to generate a number of clusters. This cluster information was then used to construct a set of simple Mamdani-type fuzzy systems using the approach described in Section 5.3.2.3. From the performance of these fuzzy models, one can judge the merits of all the clustering techniques.

In order to compare the performance of the clustering algorithms across different levels of granularity, the experiments were carried-out using various settings of the cluster number (fuzzy rule number). For each experiment, the average result of 10

## Chapter 5: FM-HCMO

---

runs is regarded as the final outcome.

To evaluate the performance of fuzzy models, the accuracy measure, represented by the Root Mean Square Error (*RMSE*), is used throughout. It is described as follows:

$$RMSE = \sqrt{\frac{\sum_{l=1}^N (y_l^m - y_l^p)^2}{N}}, \quad (5.9)$$

where  $y_l^m$  is the measured output data and  $y_l^p$  is the predicted output data,  $l = 1, 2, \dots, N$ ;  $N$  is the total number of data.

Tables 5-1, 5-2 and 5-3 list the modelling performance using these two clustering methods for the function approximation, ROA modelling and UTS modelling problems, respectively.

From Table 5-1, it can be seen that, for this function approximation problem, the agglomerative complete-link clustering algorithm performs better than FCM in all the situations, except when the cluster number is 4. Furthermore, with the increase of the cluster number, the modelling performance of the agglomerative complete-link algorithm is improved significantly compared to FCM.

**Chapter 5: FM-HCMO**

Table 5-1. Comparative results of different clustering algorithms for Mamdani fuzzy systems construction with the function approximation problem

Number of Clusters (Number of Rules)	<i>RMSE</i> of 50 Training Data	
	FCM	Agglomerative Complete-link Clustering
4	0.5600	0.5617
6	0.4855	0.4443
8	0.4482	0.3854
10	0.3721	0.3305
15	0.3772	0.2003
20	0.3172	0.1773
30	0.2323	0.1097

Table 5-2. Comparative results of different clustering algorithms for Mamdani fuzzy systems construction with the ROA modelling problem

Number of Clusters (Number of Rules)	<i>RMSE</i> of 250 Training Data (%)	
	FCM	Agglomerative Complete-link Clustering
6	6.1171	6.3490
8	6.1117	6.1331
10	6.1588	6.0361
15	6.2054	5.7806
20	6.0942	5.2121
30	6.0115	4.8238
50	5.9062	3.6695

## Chapter 5: FM-HCMO

Table 5-3. Comparative results of different clustering algorithms for Mamdani fuzzy systems construction with the UTS modelling problem

Number of Clusters (Number of Rules)	RMSE of 1000 Training Data (MPa)	
	FCM	Agglomerative Complete-link Clustering
8	136.3927	171.8730
10	135.5494	139.7599
15	136.5325	135.0249
20	137.2658	121.8931
30	139.3152	113.1180
50	138.6608	96.9054
100	137.4576	63.5138

From Table 5-2, it can be seen that, when modelling the ROA problem, the agglomerative complete-link algorithm performs better than FCM in most of the situations, while FCM performs better when the cluster number is 6 or 8. Furthermore, with the increase of the cluster number, the modelling performance of the agglomerative complete-link algorithm is improved significantly, as compared to FCM.

From Table 5-3, it can be seen that, for this UTS modelling problem, the agglomerative complete-link algorithm performs better when the cluster number is relatively large (15 to 100); and FCM performs better when the cluster number is relatively small (8 to 10). With the increase of the cluster number, the modelling performance of the agglomerative complete-link algorithm is still improved greatly, when compared to FCM.



## Chapter 5: FM-HCMO

---

From these above experiments, it can be concluded that the proposed hierarchical clustering algorithm can lead to better clustering information for fuzzy modelling problems than FCM and therefore it is more suitable for fuzzy modelling purposes.

### 5.3.3.2 Effects of Various Weights of Distance Measure

For a clustering algorithm, the distance measure should be defined first. Generally, the distance measure between two data points  $\mathbf{p}_m = [x_1^m, x_2^m, \dots, x_D^m, y_m]^T$  and  $\mathbf{p}_n = [x_1^n, x_2^n, \dots, x_D^n, y_n]^T$  can be defined using the following equation:

$$Dis = \sqrt{\sum_{i=1}^D (w_{I_i} \times (x_i^m - x_i^n))^2 + (w_O \times (y_m - y_n))^2}, \quad (5.10)$$

where  $\mathbf{w}_I = [w_{I1}, w_{I2}, \dots, w_{ID}]$  is the vector of the input distance weights and  $w_O$  is the output distance weight.

The distance weights  $w_{I1}, w_{I2}, \dots, w_{ID}$  and  $w_O$  can influence the clustering compactness in different dimensions and thus influence the generated fuzzy model's sensitivity to different input variables. Specifically, one high weight value means that the data in this dimension are more important than those data in other dimensions with small weight values. Normally, a high weight value for the output data is suitable for the modelling purpose.

Three experiments were carried-out to test the influence of different distance weights in fuzzy modelling. These experiments are based on the proposed

**Chapter 5: FM-HCMO**

---

agglomerative complete-link algorithm and use the same training data as those experiments in the previous section (Section 5.3.3.1). In the first experiment, the fuzzy systems (with 8 rules) are generated from 50 function approximation data. In the second experiment, the fuzzy systems (with 15 rules) are elicited from 250 ROA data. In the third experiment, the fuzzy systems (with 30 rules) are generated using 1000 UTS data. For these three experiments, different distance weights for input and output data are tested. The performances of the generated fuzzy models are shown in Tables 5-4, 5-5 and 5-6.

Table 5-4. Comparative results of different distance weights for Mamdani fuzzy systems (with 8 rules) construction with the function approximation problem; the bold values represent the best results.

Distance weights	$w_I = 1$ $w_O = 0.25$	$w_I = 1$ $w_O = 0.5$	$w_I = 1$ $w_O = 0.75$	$w_I = 1$ $w_O = 1$	$w_I = 1$ $w_O = 1.5$
RMSE of 50 Data	0.5605	0.5175	0.5240	0.3854	0.3958
Distance weights	$w_I = 1$ $w_O = 2$	$w_I = 1$ $w_O = 2.5$	$w_I = 1$ $w_O = 3$	$w_I = 1$ $w_O = 4$	$w_I = 1$ $w_O = 5$
RMSE of 50 Data	0.3712	<b>0.3673</b>	0.4347	0.4869	0.4562
Distance weights	$w_I = 1$ $w_O = 6$	$w_I = 1$ $w_O = 7$	$w_I = 1$ $w_O = 8$	$w_I = 1$ $w_O = 9$	$w_I = 1$ $w_O = 10$
RMSE of 50 Data	0.4577	0.4755	0.4755	0.5109	0.5138

**Chapter 5: FM-HCMO**

Table 5-5. Comparative results of different distance weights for Mamdani fuzzy systems (with 15 rules) construction with the ROA modelling problem; the bold values represent the best results.

Distance weights	$w_I = 1$ $w_O = 0.25$	$w_I = 1$ $w_O = 0.5$	$w_I = 1$ $w_O = 0.75$	$w_I = 1$ $w_O = 1$	$w_I = 1$ $w_O = 1.5$
RMSE of 250 Data (%)	5.9889	6.0353	6.0831	5.7806	5.0042
Distance weights	$w_I = 1$ $w_O = 2$	$w_I = 1$ $w_O = 2.5$	$w_I = 1$ $w_O = 3$	$w_I = 1$ $w_O = 4$	$w_I = 1$ $w_O = 5$
RMSE of 250 Data (%)	4.8186	4.1899	4.3433	<b>4.1449</b>	4.2940
Distance weights	$w_I = 1$ $w_O = 6$	$w_I = 1$ $w_O = 7$	$w_I = 1$ $w_O = 8$	$w_I = 1$ $w_O = 9$	$w_I = 1$ $w_O = 10$
RMSE of 250 Data (%)	4.7766	4.6331	4.5474	4.3009	4.5514

Table 5-6. Comparative results of different distance weights for Mamdani fuzzy systems (with 30 rules) construction with the UTS modelling problem; the bold values represent the best results.

Distance weights	$w_I = 1$ $w_O = 0.25$	$w_I = 1$ $w_O = 0.5$	$w_I = 1$ $w_O = 0.75$	$w_I = 1$ $w_O = 1$	$w_I = 1$ $w_O = 1.5$
RMSE of 1000 Data (MPa)	109.6063	110.1273	108.4278	113.1180	113.1682
Distance weights	$w_I = 1$ $w_O = 2$	$w_I = 1$ $w_O = 2.5$	$w_I = 1$ $w_O = 3$	$w_I = 1$ $w_O = 4$	$w_I = 1$ $w_O = 5$
RMSE of 1000 Data (MPa)	97.4785	81.9490	86.8525	81.2002	80.7164
Distance weights	$w_I = 1$ $w_O = 6$	$w_I = 1$ $w_O = 7$	$w_I = 1$ $w_O = 8$	$w_I = 1$ $w_O = 9$	$w_I = 1$ $w_O = 10$
RMSE of 1000 Data (MPa)	83.4777	<b>63.2239</b>	80.6791	84.5728	84.9728

From these tables, it can be seen that a large distance weight for the output variable is more suitable for fuzzy modelling; it is worth nothing that the weight

## **Chapter 5: FM-HCMO**

---

should not be too large. In the first experiment, good performances are observed when  $w_O$  lies in the range between 1 and 2.5 and the best performance is registered when  $w_O$  is 2.5. In the second experiment, the model performs well when  $w_O$  is in the range from 2.5 to 5 and the best performance appears when  $w_O$  is 4. In the third experiment, good performances can be observed when  $w_O$  is between 4 and 8 and the model performs best when  $w_O$  is 7. It is suggested that, for low-dimensional problems,  $w_O$  value should be selected between 1 and  $Dim$ , where  $Dim$  is the dimension number of the modelling problem. For high-dimensional problems,  $w_O$  value should be selected between 2 and  $Dim/2$ .

### **5.4 Accuracy Optimisation and Data Selection**

#### **5.4.1 Accuracy Optimisation**

The initial fuzzy model elicited by a clustering method generally shows a poor performance in accuracy. To make the initial fuzzy model meaningful and effective for application, one should concentrate on the improvement of accuracy.

To the question of why not use a multi-objective optimisation for optimising both accuracy and interpretability, one would answer that at this stage as follows: if a multi-objective optimisation is employed, then this would inevitably increase the computational complexity and render the optimisation inefficient. Another reason for using accuracy optimisation at this stage is that this procedure can cooperate

## **Chapter 5: FM-HCMO**

---

with a data selection approach to choose the most representative training data set.

In this work, the single objective evolutionary algorithm, nPSO (see Chapter 4), is employed. This algorithm is applied here to search for the optimal parameters of the membership functions in combination with the performance index of the Root Mean Square Error (*RMSE*).

### **5.4.1.1 The Validation Mechanism**

To avoid the problem of overtraining (or overfitting) [Tetko *et al.* 1995] in the optimisation process, a validation mechanism is designed for this phase. One part of the data is separated and is used for validation only. In particular, the validation mechanism works as follows: In the optimisation process, after a certain number of function evaluations (*NEv*), the model will be evaluated once, using the validation data; Compared with the previous model performance (which is  $T \times NEv$  function evaluations before, where  $T$  is a small integer), if the performance of the present model on the validation data is improved, then the optimisation will be allowed to continue, otherwise, the optimisation procedure will be terminated.

The following is an example of how this validation mechanism works. In the previously mentioned UTS modelling problem (15-input and 1-output), there are a total of 2820 available data vectors and 10% of the data are separated and used for validation. For this example,  $NEv = 1000$  and  $T = 5$ . Thus, in the optimisation process, after every 1000 function evaluations, the fuzzy model is examined once,

## Chapter 5: FM-HCMO

using the validation data. If the performance of the present model is worse than that of the former model (5000 function evaluations in the past), the optimisation process will be terminated. The model that performs best on the validation data is viewed as the optimisation solution. Figure 5-9 shows the model performance throughout the whole optimisation process based on the validation data. At the time that is identified as a red square in this figure, the *RMSE* of the validation data is larger than that at the time of 5000 function evaluations before, which is shown as a green round mark in the figure. Following the principle of the validation mechanism, the optimisation algorithm stops at the time of the square mark and the solution at the time of the round mark is the final solution for this optimisation.

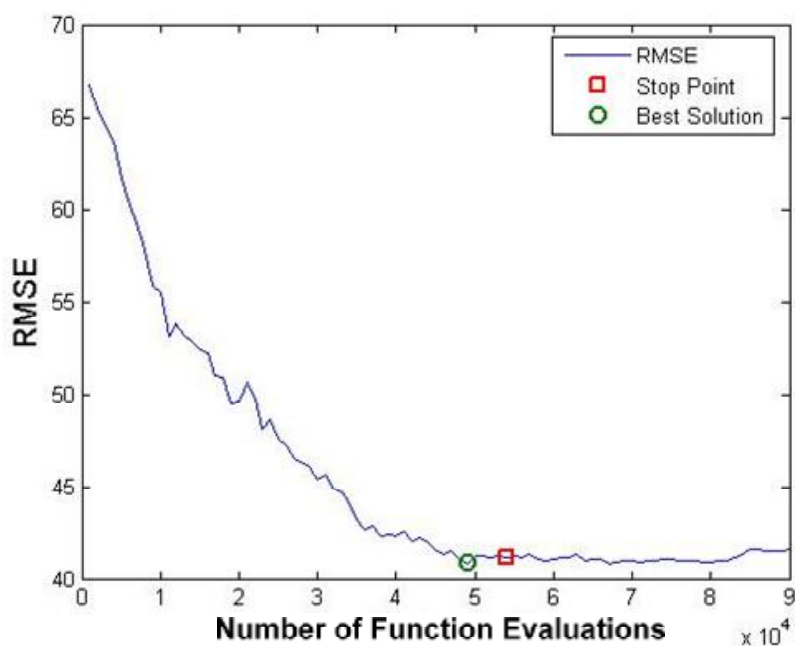


Figure 5-9. *RMSE* of the validation data in the optimisation process

**5.4.2 Data Selection**

It is well-known that more training data will not necessarily lead to a better performance for data-driven models. Normally, one can identify two typical scenarios:

1. The training data are not enough: There are only a few data points (vectors) available for model construction. In this situation, more training data will provide more information to the modelling algorithm and the constructed model will be more accurate.
2. The training data are abundant: These data points are concentrated in a small area of the input/output space. In this situation, if all the data are used in the training phase, then the extracted model will be very sensitive in the areas that include these training data and not sensitive in the areas that include only a few of these training data, which means that the extracted model will be accurate in some areas but not accurate in other areas. To avoid this situation, it is required to select parts of these data that are representative and make use of them in the training phase.

Scenario 2 can also be regarded as a problem of different data densities in different areas. The areas with high data densities will be trained well and the areas with low data densities will be trained less so. To balance the training in different areas, one needs to reduce the training data in the areas with high data densities and some

## ***Chapter 5: FM-HCMO***

---

of the most representative data should be held and used later.

Obviously, another important advantage of the data selection mechanism is that it will save effort and time for training, since the data selection will reduce the size of the training data set.

### **5.4.2.1 Rough Data Selection**

As already stated, the selected training data should be representatives of all the training data. They should include all the important information on the training data. From this viewpoint, the clustering technique may prove helpful. In data clustering, all the data are classified into several clusters with different features. In other words, the data in different clusters contain different information. Thus, the representative data should be selected from each cluster. To balance the influence of different clusters, the number of the selected data from each cluster should be approximately equal.

For every cluster, the data with a minimal or maximal value in any dimension are very important. They provide one with the information of the cluster boundaries. The generated model using this type of data can avoid the problem associated with generalisation. Therefore, these data should be included in the selected training data set.



## **Chapter 5: FM-HCMO**

---

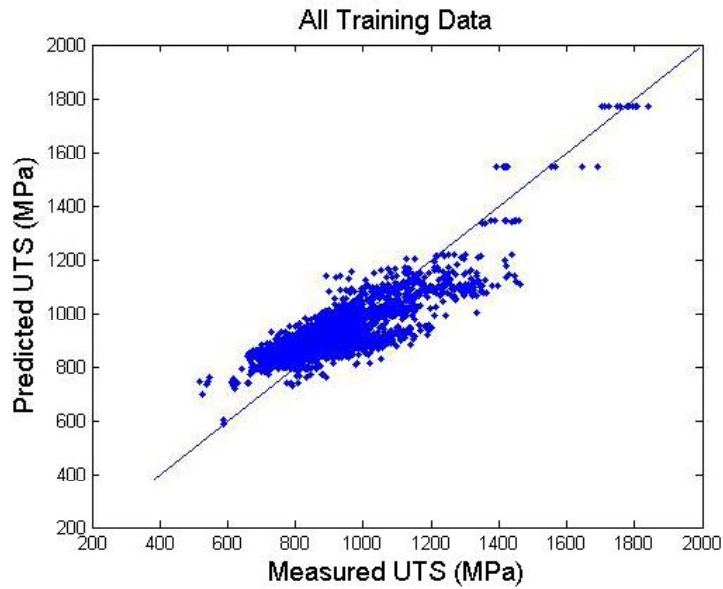
In the previous initial fuzzy model extraction approach, the hierarchical clustering algorithm has been employed. The clustering result can be applied to select the training data directly and the clustering algorithm does not need to be executed again.

In summary, this selection method can be described as follows: For every cluster, the data including the minimal or maximal value in any one input or output dimension are selected as the training data. If the data including the minimal or maximal value in one particular dimension is more than one, only one data point (vector) is randomly chosen and kept in the training data set. As a result, if the number of clusters is  $N_c$  and the dimension of the problem is  $D+1$  ( $D$ -input and 1-output), then the number of the selected training data will be less than  $2 \times N_c \times (D+1)$ .

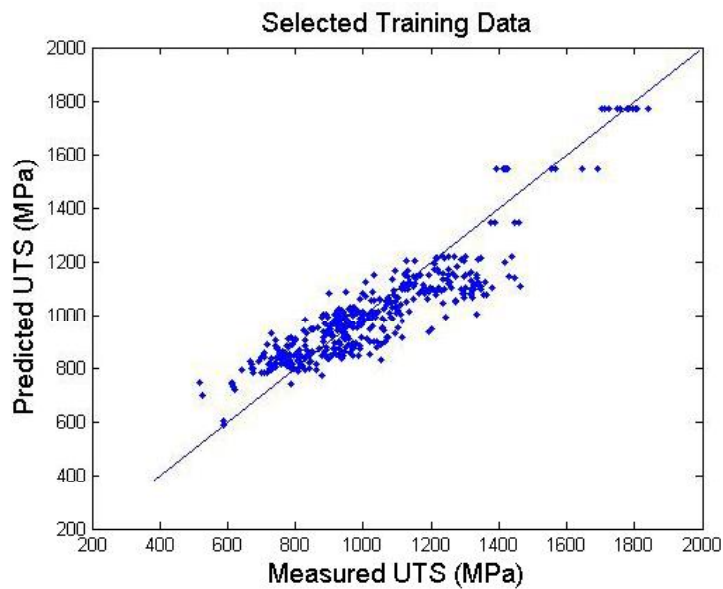
An experiment which consists of modelling the mechanical property UTS of steel is described next. The UTS data include 15 inputs and one output, and the initial number of fuzzy rules is set to 20. In this experiment, if the data selection method were not to be used, the number of training data would be 2820. If the previous described data selection method used them, the number of training data is reduced to 448. Figure 5-10 shows the prediction results of all the training data and the selected training data using the initial elicited fuzzy model. Comparing Figure 5-10 (b) with Figure 5-10 (a), one can see that: first, the data distribution in Figure 5-10 (b) is better than that in Figure 5-10 (a); second, the data in Figure 5-10 (b) may be good representatives of the data in Figure 5-10 (a). In this case, the *RMSE* of all

## Chapter 5: FM-HCMO

the training data is 88.72 MPa and the *RMSE* of the selected training data is 105.27 MPa.



(a)



(b)

Figure 5-10. The initial fuzzy model's predicted UTS versus measured UTS: (a) all the training data (2820 data), (b) the selected training data (448 data)

**5.4.2.2 Missing Representative Data Selection**

The training data selection method in Section 5.4.2.1 is able to find a set of training data with some representative features, but it may still miss some important data. First, if more than one datum includes the minimal or maximal value in one specific input or output dimension, then only one datum is randomly selected and retained for future use. The neglected data may however contain some important information. Second, the data located inside the clusters, which do not have any minimal or maximal value, are also likely to contain some useful information for modelling.

Compared with the data that have already been selected, the missing data representative must possess some different features. Thus, the prediction model, which is trained based on the data that have already been selected, must be inaccurate as far as the missing data are concerned. As a result, the following method is proposed which is used to detect the missing representative data to be added to the training data set:

1. Train the initial fuzzy model using the data selected by the method of Section 5.4.2.1. This model does not need to be well trained.
2. Calculate the output prediction of all the available training data using the trained model. Find a set of data with the biggest differences between the predicted output value and the true output value.

## **Chapter 5: FM-HCMO**

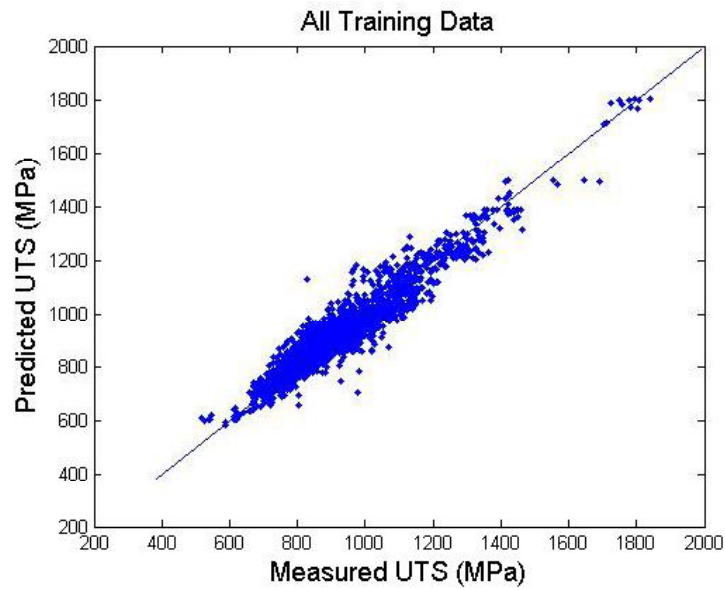
---

3. The data found in Step 2 are added to the training data set and the new training data set is used to improve the existing fuzzy model.

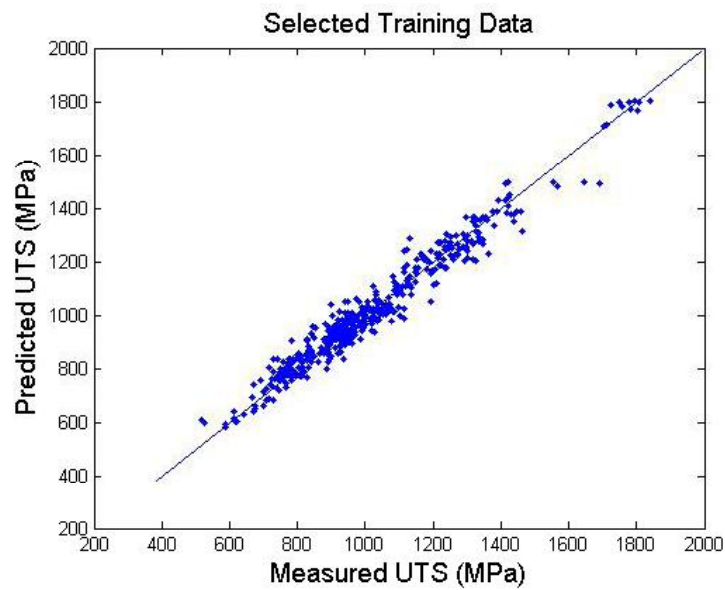
Following the UTS modelling experiment in Section 5.4.2.1, the initial fuzzy model is trained using the roughly selected training data. The nPSO optimisation algorithm is used for model training with 20,000 function evaluations. Figure 5-11 shows the prediction results of all the training data and the selected training data using the improved fuzzy model, respectively. From this figure, it can be seen that even though this trained model performs very well for the selected training data, it cannot predict some training data correctly out of all the possible training data, which means that the selected training data set misses some important and representative data including useful information for model construction. In this case, the *RMSE* of all the training data is 47.70 and the *RMSE* of the roughly selected training data is 47.65.

In the light of the above, the data selection method described in this section can be applied to detect the missing representative data. As shown in Figure 5-12, the circled data are some examples of the ones with the biggest differences between the predicted output and the truly measured output. They are viewed as the missing representative data and should be added to the selected training data set.

It should be noted that the above missing data selection procedure may need to be repeated several times to ensure that all the representative data are included in the final selected training data set.



(a)



(b)

Figure 5-11. The trained fuzzy model's predicted UTS versus measured UTS: (a) all the training data (2820 data), (b) the selected training data (448 data)

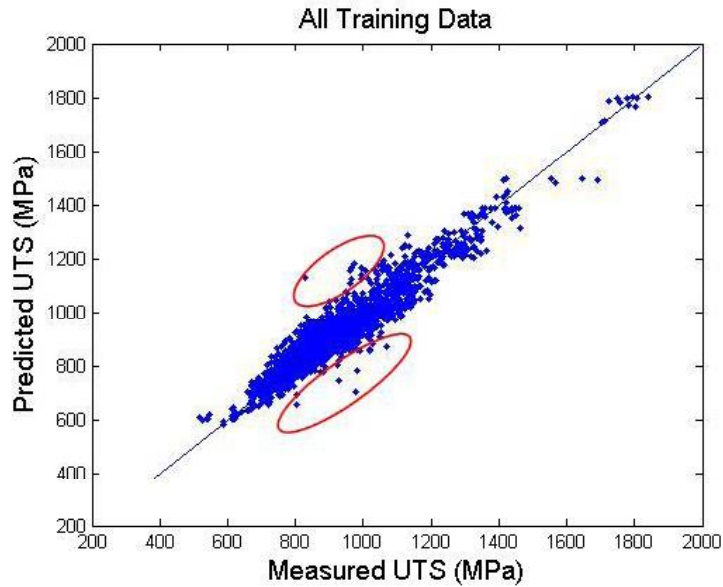


Figure 5-12. Examples of the missing representative data

### **5.4.3 The Joint Mechanism of Accuracy Optimisation and Data Selection**

By combining the initially selected training data and the subsequently detected training data, one can obtain the final training data, which are the representatives of all the training data which can then be used in the next model improvement stage. Figure 5-13 shows the flow chart of the joint mechanism for accuracy optimisation and data selection. Normally, the termination criterion for this mechanism is designed so that the loop iteration achieves a predefined number.

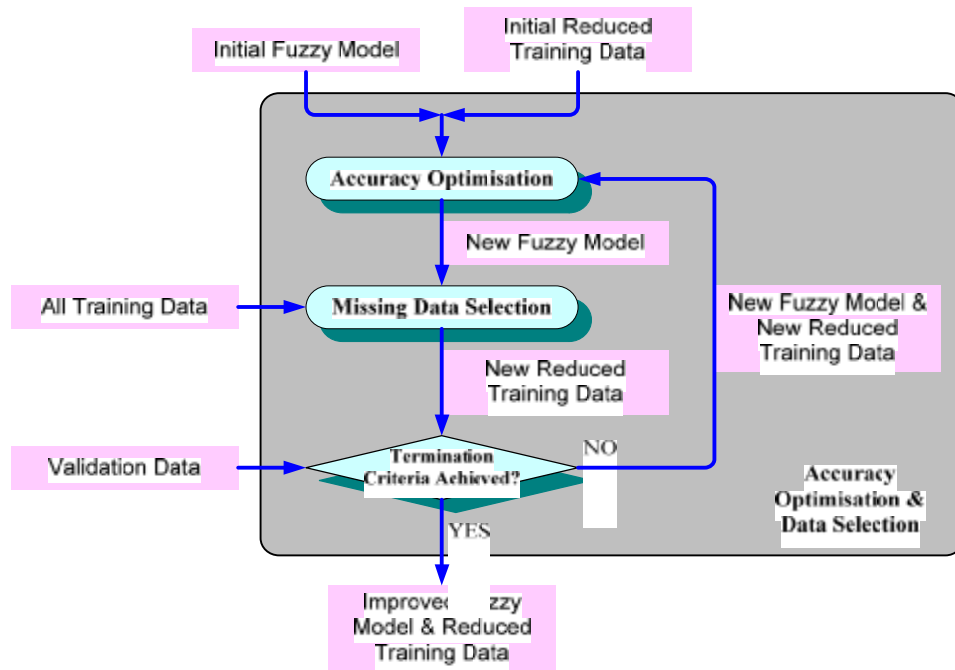
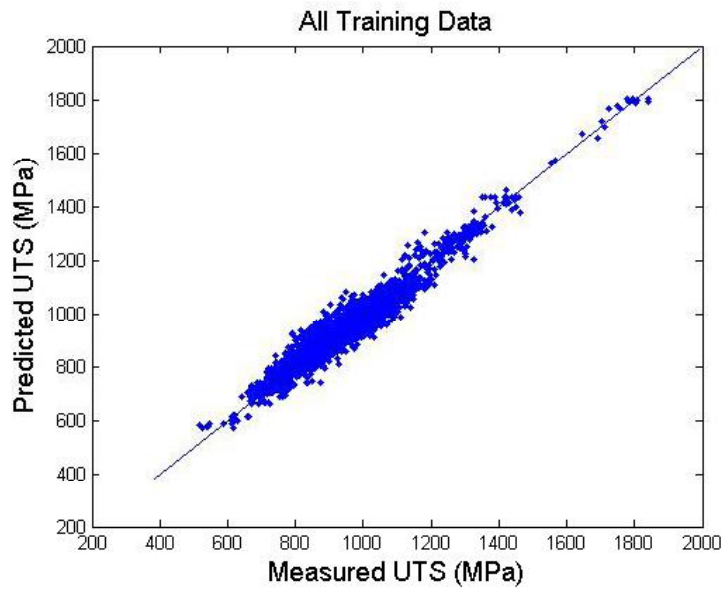
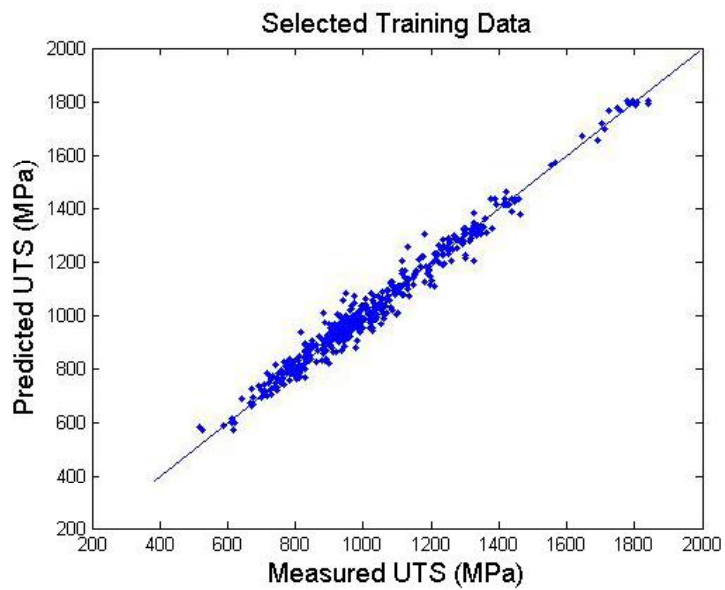


Figure 5-13. Flow chart of the mechanism for accuracy optimisation and data selection

Following the previous UTS modelling experiment described in Section 5.4.2.2, the fuzzy model, which has been initially trained, is then trained using the newly formed training data. nPSO is employed for this training procedure and 70,000 function evaluations are used. Figure 5-14 shows the prediction results of all the training data and the final selected training data using the well-trained fuzzy model. From this figure, it can be seen that the trained model performs very well for all the training data while only parts of these training data are truly involved in the training process. Following this experiment, the achieved *RMSE* for all the training data is 38.21 and the *RMSE* for the selected training data is 37.65.



(a)



(b)

Figure 5-14. The well-trained fuzzy model's predicted UTS versus measured UTS: (a) all the training data (2820 data), (b) the selected training data (480 data)



## **5.5 Simultaneous Multi-Objective Optimisation of Accuracy and Interpretability**

After the accuracy optimisation and data selection, an accurate fuzzy model and a reduced number of training data are obtained. The focus is shifted next to improving the interpretability (transparency) of the fuzzy system while maintaining a good accuracy.

### **5.5.1 Interpretability Improvement**

The improvement of the interpretability of fuzzy systems is tantamount to reducing the number of fuzzy rules, reducing the length of fuzzy rules, reducing the number of fuzzy sets, and adjusting these sets to be evenly distributed along the universes of discourse. These tasks can be achieved using the following four-step operation:

#### **5.5.1.1 Removing Redundant Fuzzy Rules**

This operation can reduce the number of fuzzy rules. At the same time, some fuzzy sets, which are only involved in these redundant rules, are also removed. For instance, in the case of Figure 5-15, a redundant rule exists which is: IF *input* is  $A_1^3$ , THEN *output* is  $A_2^1$ .

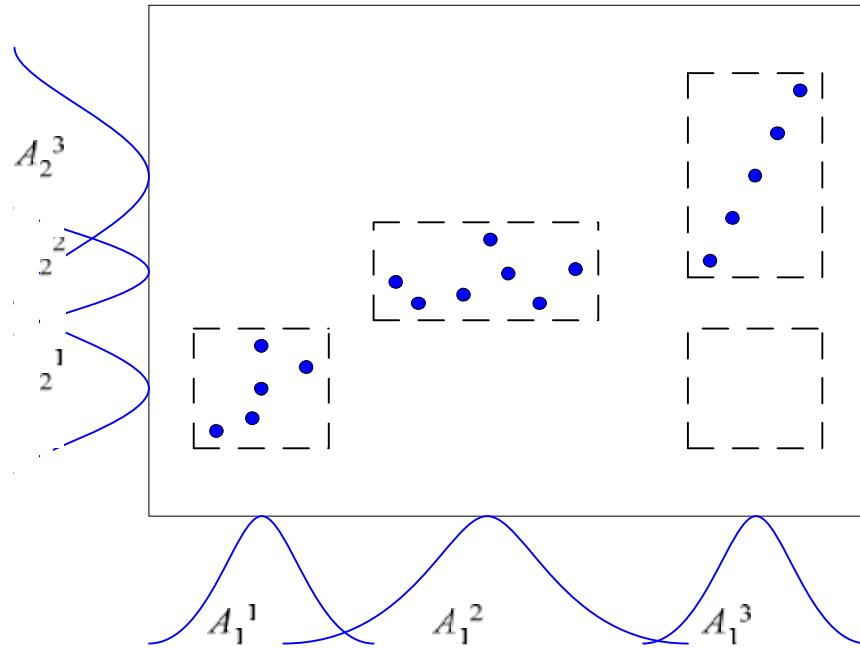


Figure 5-15. An example of a redundant rule

To evaluate whether a fuzzy rule is redundant or not, two evaluation measures are used, namely *confidence* and *support* [Ishibuchi *et al.* 2001]. In the area of data mining, these two measures have been used for evaluating association rules in the pattern classification problem [Ishibuchi & Yamamoto 2004]. Let  $\mathcal{C}$  be the set of  $N$  training patterns  $\mathbf{p}_i$ , where  $\mathbf{p}_i = [\mathbf{x}_i, y_i]^T = [x_1^i, x_2^i, \dots, x_D^i, y_i]^T$  and  $i = 1, 2, \dots, N$ . The *confidence* of rule  $A \rightarrow B$  is defined as follows:

$$\text{conf}(A \rightarrow B) = \frac{|C(A) \cap C(B)|}{|C(A)|}, \quad (5.11)$$

where  $|C(A)|$  is the number of training patterns that are compatible with the antecedent  $A$ , and  $|C(A) \cap C(B)|$  is the number of training patterns that are compatible with both the antecedent  $A$  and the consequent  $B$ . The *confidence*

## Chapter 5: FM-HCMO

---

indicates that  $conf \times 100\%$  of the training patterns that are compatible with the antecedent  $A$  are also compatible with the consequent  $B$ . In [Ishibuchi *et al.* 2001], the *confidence* measure is extended to the fuzzy case and can be rewritten as follows:

$$conf(A \rightarrow B) = \frac{|C(A) \cap C(B)|}{|C(A)|} \frac{\sum_{i=1}^N (\mu_A(x_i) \times \mu_B(y_i))}{\sum_{i=1}^N \mu_A(x_i)}, \quad (5.12)$$

where  $\mu_A(x_i)$  is the compatibility grade of the input vector  $x_i$  with the antecedent part  $A = [A_1, A_2, \dots, A_D]^T$  of the fuzzy rule  $R$ , and  $\mu_B(y_i)$  is the compatibility grade of the output value  $y_i$  with the consequent part  $B$  of  $R$ .  $\mu_A(x_i)$  is usually defined by the minimum operator or the product operator. Such as:

$$\mu_A(x_i) = \min(\mu_{A_1}(x_1^i), \mu_{A_2}(x_2^i), \dots, \mu_{A_D}(x_D^i)) \quad \text{or} \quad (5.13)$$

$$\mu_A(x_i) = \mu_{A_1}(x_1^i) \times \mu_{A_2}(x_2^i) \times \dots \times \mu_{A_D}(x_D^i), \quad (5.14)$$

where  $\mu_{A_j}(x_j^i)$  is the membership function of the antecedent fuzzy set  $A_j$ .

On the other hand, the *support* of  $A \rightarrow B$  is defined as follows:

$$supp(A \rightarrow B) = \frac{|C(A) \cap C(B)|}{|C|}, \quad (5.15)$$

where  $|D| = N$ . It indicates that  $supp \times 100\%$  of all training patterns are compatible with the association rule  $A \rightarrow B$ . Similarly to the *confidence*, the *support* of a fuzzy rule can be rewritten as follows:

$$supp(A \rightarrow B) = \frac{|C(A) \cap C(B)|}{|C|} \frac{\sum_{i=1}^N (\mu_A(x_i) \times \mu_B(y_i))}{N}. \quad (5.16)$$

## **Chapter 5: FM-HCMO**

---

In this work, the product of *support* and *confidence* is used as the criterion for the fuzzy rule selection. A threshold  $Th_1$  for this rule selection is also defined. If the product criterion of one rule is smaller than the threshold  $Th_1$ , then this fuzzy rule is deemed redundant, and as a consequence the fuzzy rule and the fuzzy sets that are only included by this redundant rule are removed. Generally,  $Th_1$  is defined in the range  $[0, 0.01]$ .

### **5.5.1.2 Merging Similar Fuzzy Rules**

This operation can reduce the number of fuzzy rules. At the same time, the fuzzy sets involved within similar rules are also merged. For example, the following two fuzzy rules may be thought of as similar:

$R_1$ : IF  $x_1$  is “*small*” and  $x_2$  is “*big*” and  $x_3$  is “*small*”, THEN  $y$  is “*big*”;

$R_2$ : IF  $x_1$  is “*small*” and  $x_2$  is “*big*” and  $x_3$  is “*medium*”, THEN  $y$  is “*big*”.

These can be merged into one rule as follows:

$R_{new}$  : IF  $x_1$  is “*small*” and  $x_2$  is “*big*” and  $x_3$  is “*slightly small*”, THEN  $y$  is “*big*”.

To decide whether two fuzzy rules are similar enough for combination or not, one only needs to evaluate the similarity of the antecedent parts of the rules. Two fuzzy rules with very similar antecedents but different consequents usually indicate that these two rules conflict with each other. Therefore, we should either merge these rules into one new rule or delete one of them.

## Chapter 5: FM-HCMO

---

To calculate the similarity degree of the antecedents of two fuzzy rules, the similarity of every fuzzy set pair should be checked. For the  $k$ th fuzzy rule  $R_k$ , the corresponding preconditions are  $A_1^k, A_2^k, \dots, A_D^k$ . Similarly, the corresponding antecedents of the  $l$ th rule  $R_l$  are  $A_1^l, A_2^l, \dots, A_D^l$ . Thus, the similarity measure can be characterised as follows:

$$S_R(R_k, R_l) = \prod_{m=1}^D S(A_m^k, A_m^l), \quad (5.17)$$

where  $S(A_m^k, A_m^l)$  is the similarity of two fuzzy sets  $A_m^k$  and  $A_m^l$  and it is defined in Section 5.5.1.4.

Once  $S_R(R_k, R_l)$  reaches a threshold value  $Th_2$ , then these two fuzzy rules as well as the fuzzy set pairs of these two rules are considered to be similar. The two fuzzy rules are then merged into a new rule  $R_{new}$ . The new antecedents and consequent of  $R_{new}$  are obtained by merging the fuzzy sets (see Section 5.5.1.4). Normally, threshold  $Th_2$  is defined in the range  $[0.01, 1]$ .

### 5.5.1.3 Removing Redundant Fuzzy Sets

This operation can reduce the number of fuzzy sets and remove some fuzzy sets that cover others. In addition, this operation can shorten the length of fuzzy rules because some of their premises, which include redundant fuzzy sets, should also be removed from the fuzzy rules simultaneously. Figure 5-16 shows an example of the membership function relating to a redundant fuzzy set.

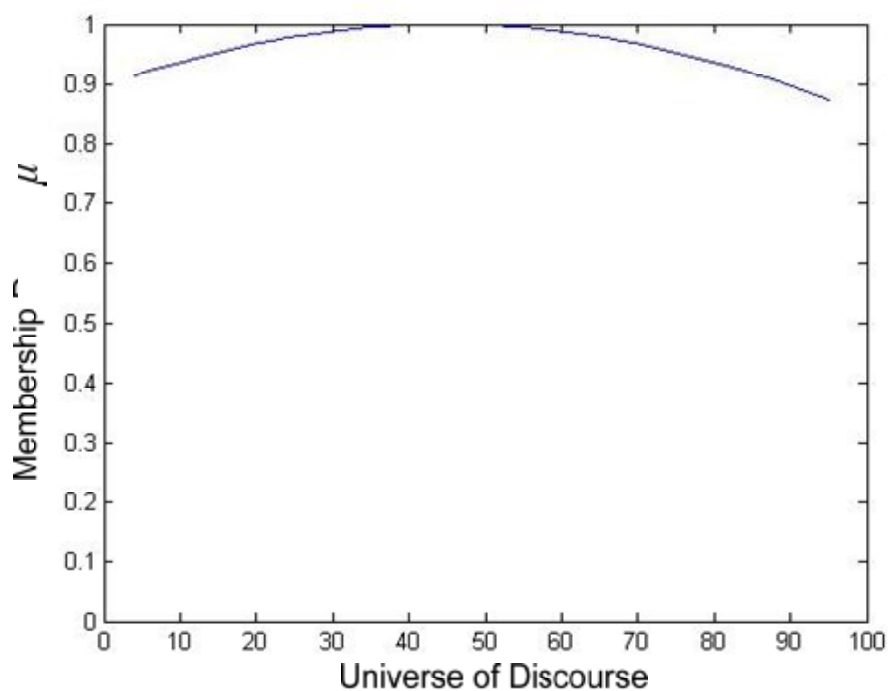


Figure 5-16. An example of the membership function relates to a redundant fuzzy set

In this method, the similarity for each fuzzy set  $A_n$  to the universal set  $U (\mu_U(x)=1)$  is calculated. If the similarity value is greater than a threshold value  $Th_3$ , then this fuzzy set is counted as a redundant fuzzy set. As a result, the associated fuzzy set should be removed. If Gaussian membership functions are involved, then the similarity of one fuzzy set to the universal set can be represented using the parameter  $\sigma_n$ . Generally,  $Th_3$  is defined in the range  $[0.5, 2]$ .

**5.5.1.4 Merging Similar Fuzzy Sets**

This operation can keep the number of fuzzy sets low and also tune the fuzzy sets so as not to overlap. Figure 5-17 shows an example of similar fuzzy sets. It can be seen that the two blue membership functions are very similar in location and extension range. To simplify the fuzzy model and improve the interpretability, these functions are merged into one membership function, which can represent the characteristics of both of them. The red dashed membership function shows a possible solution following such a merger.

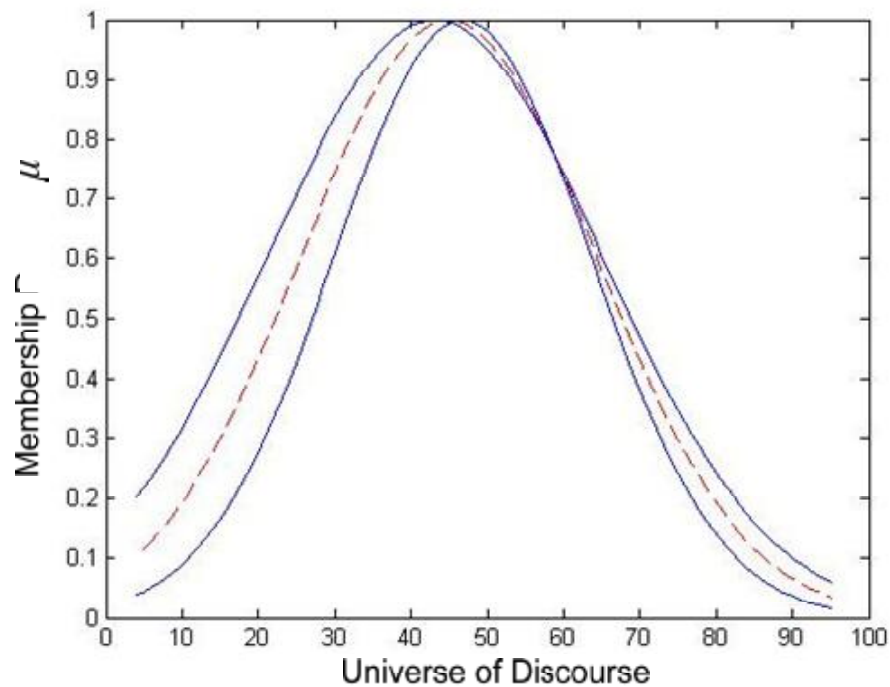


Figure 5-17. An example of membership functions of similar fuzzy sets

From an evaluation viewpoint, a similarity measure should be defined. There are

## Chapter 5: FM-HCMO

---

several fuzzy similarity measures [Cross 1993; Setnes 1995; Setnes *et al.* 1998], one of which is based on the following distance measure:

$$S(A_1, A_2) = \frac{1}{1 + d(A_1, A_2)}, \quad (5.18)$$

where  $d(A_1, A_2)$  is the distance between two fuzzy sets  $A_1$  and  $A_2$ .

If Gaussian membership functions are used, then the following simple expression can be used to approximate the distance between the two fuzzy sets:

$$d(A_1, A_2) = \sqrt{(c_1 - c_2)^2 + (\sigma_1 - \sigma_2)^2}. \quad (5.19)$$

A threshold  $Th_4$  for merging similar fuzzy sets is then defined, where  $Th_4 \in (0, 1]$ . If  $S(A_1, A_2) > Th_4$ , i.e., the fuzzy sets  $A_1$  and  $A_2$  are highly overlapping, then these two fuzzy sets should be merged into one new fuzzy set  $A_{new}$ , where  $c_{new} = (c_1 + c_2) / 2$  and  $\sigma_{new} = (\sigma_1 + \sigma_2) / 2$ . Because the fuzzy sets in the antecedent part and the fuzzy sets in the consequent part have a different influence on the performance of the fuzzy model, different thresholds  $Th_4$  and  $Th_5$  should be predefined for the antecedent part and the consequent part respectively. Normally,  $Th_4$  and  $Th_5$  are set to values in the range  $[0.8, 1]$ .

### 5.5.2 Experimental Studies

#### 5.5.2.1 An Example of Using the Interpretability Improvement Approach

To validate the effectiveness of the proposed strategy for improving



**Chapter 5: FM-HCMO**

interpretability, it was applied to the problem of modelling the mechanical property UTS of steel (15-input and 1-output; 2820 data). The initial number of fuzzy rules is set to 20 and  $Th_1 - Th_5$  are set to 0.001, 0.1, 0.6, 0.9 and 0.95, respectively.

Table 5-7 shows the main parameters of the fuzzy models, which were obtained following the different operation steps during the interpretability improvement process. In this table, rule length refers to the total number of antecedent conditions; it can be seen that the interpretability improvement approach succeeded in reducing the number of fuzzy rules, the number of fuzzy sets and in generally simplifying the structure of the fuzzy rules.

Table 5-7. The main parameters of the UTS fuzzy models following the different stages of the interpretability improvement

Fuzzy model	Number of fuzzy rules	Number of fuzzy sets in every input and output dimension	Rule length of every fuzzy rule	RMSE of training data
Before the interpretability improvement	15	Inputs: [15; 15; 15; 15; 15; 15; 15; 15; 15; 15; 15; 15; 15; 15] Output: 15	[15; 15; 15; 15; 15; 15; 15; 15; 15; 15; 15; 15; 15; 15; 15; 15; 15]	31.04
After Step 5.5.1.2	12	Inputs: [12; 12; 12; 12; 12; 12; 12; 12; 12; 12; 12; 12] Output: 12	[15; 15; 15; 15; 15; 15; 15; 15; 15; 15; 15; 15; 15]	35.87
After Step 5.5.1.3	12	Inputs: [10; 10; 10; 11; 11; 9; 9; 12; 10; 10; 9; 8; 10; 7; 10] Output: 12	[12; 9; 13; 14; 11; 15; 13; 9; 13; 12; 12; 13]	47.01
After Step 5.5.1.4	12	Inputs: [4; 6; 6; 7; 6; 6; 3; 8; 7; 7; 4; 4; 7; 7; 5] Output: 8	[12; 9; 13; 14; 11; 15; 12; 9; 13; 12; 12; 13]	66.75

**Chapter 5: FM-HCMO**

---

Figure 5-18 shows the membership functions of Input 14 and Input 15 before the interpretability improvement, and Figure 5-19 shows the same membership functions after the interpretability improvement. Comparing these figures, it can be seen that the distributions of membership functions have improved significantly.

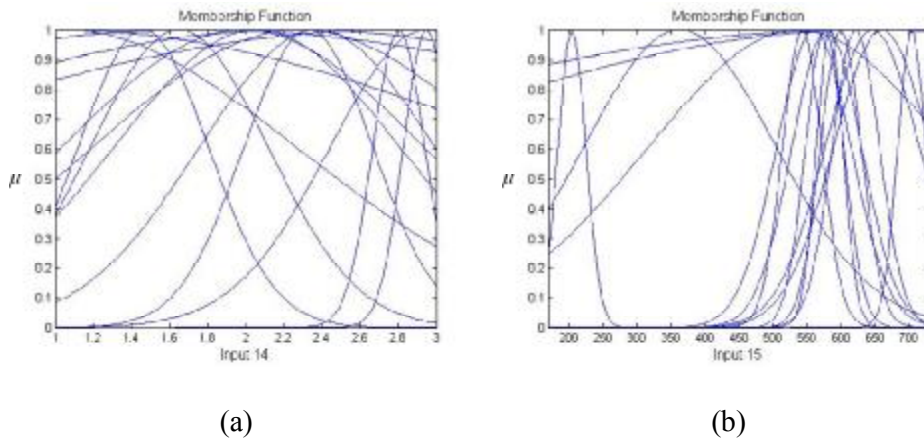


Figure 5-18. The membership functions of Inputs 14 and 15 before the interpretability improvement

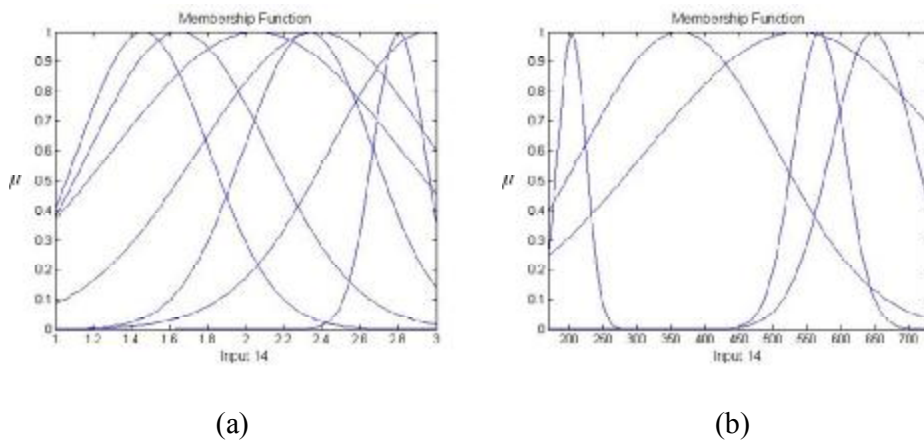


Figure 5-19. The membership functions of Inputs 14 and 15 after the interpretability improvement

**5.5.2.2 Effects of the Thresholds of the Interpretability Improvement**

**Approach**

In the proposed interpretability improvement approach, there are five thresholds that need to be set in the 4-step operation. To inspect their effects on the system performance, a set of experiments have been carried out. These experiments are based on the UTS data (2820 data) with the initial number of fuzzy rules being set to 12.

1. The first step of the interpretability improvement is to remove the redundant fuzzy rules.  $Th_1$  is used to define whether one rule is redundant or not.  $Th_1$  was set to be variable in the range  $0 \leq Th_1 \leq 0.05$  and other thresholds and parameters were fixed at  $Th_2 = 1$ ;  $Th_3 = 1$ ;  $Th_4 = 1$ ;  $Th_5 = 1$ . Figure 5-20 shows the system performance with different  $Th_1$  values.

From Figure 5-20, it can be seen that, with the increase of  $Th_1$ , the *RMSE* of the obtained system increases while the number of rules, the number of fuzzy sets and the total rule length decrease. This means that, with the increase of  $Th_1$ , the resulting model accuracy decreases and its interpretability increases.

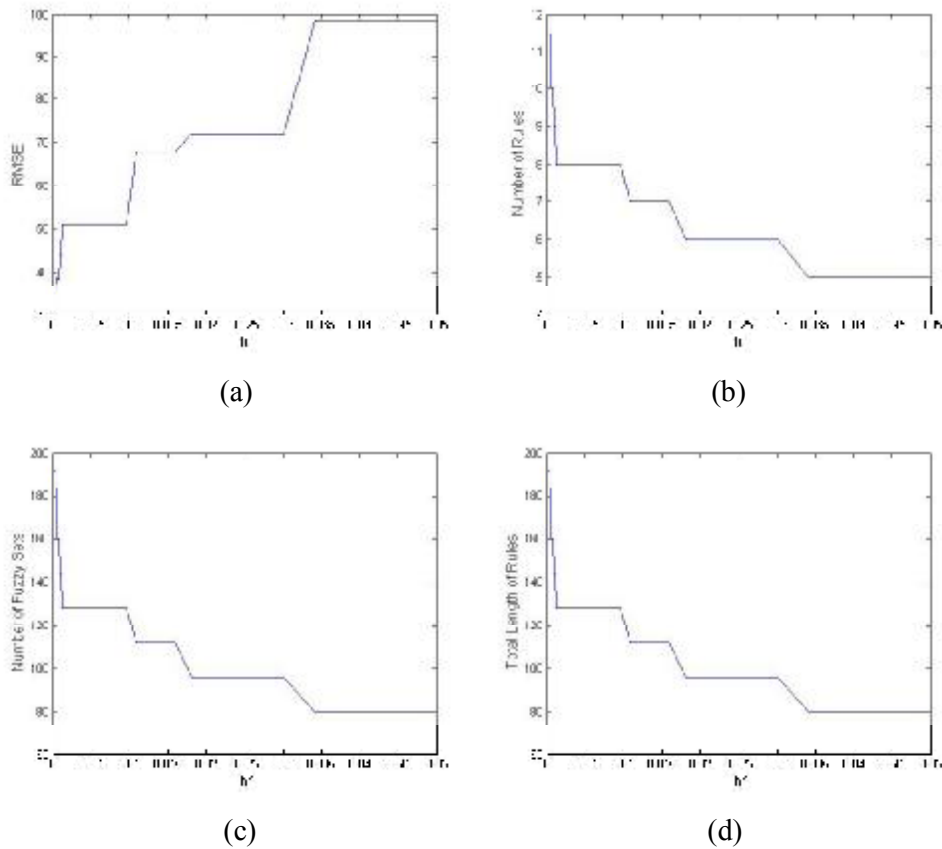


Figure 5-20. The performance of fuzzy models following the interpretability improvement with different  $Th_1$ : (a)  $RMSE$  versus  $Th_1$ ; (b) the number of rules versus  $Th_1$ ; (c) the number of fuzzy sets versus  $Th_1$ ; (d) the total length of rules versus  $Th_1$

2. The second step of the interpretability improvement is to merge similar fuzzy rules.  $Th_2$  is used to define whether two rules are similar enough to be merged. In this experiment,  $Th_2$  was set to be variable in the range  $0.01 \leq Th_2 \leq 1$  and other thresholds and parameters were fixed at  $Th_1 = 0$ ;  $Th_3 = 1$ ;  $Th_4 = 1$ ;  $Th_5 = 1$ . Figure 5-21 shows the model performance with various  $Th_2$  values.

From Figure 5-21, it can be seen that, with the increase of  $Th_2$ , the  $RMSE$

## Chapter 5: FM-HCMO

values of the obtained fuzzy model tend to decrease while the number of rules, the number of fuzzy sets and the total rule length tend to increase, which means that the obtained model accuracy tends to increase and its interpretability tends to decrease.

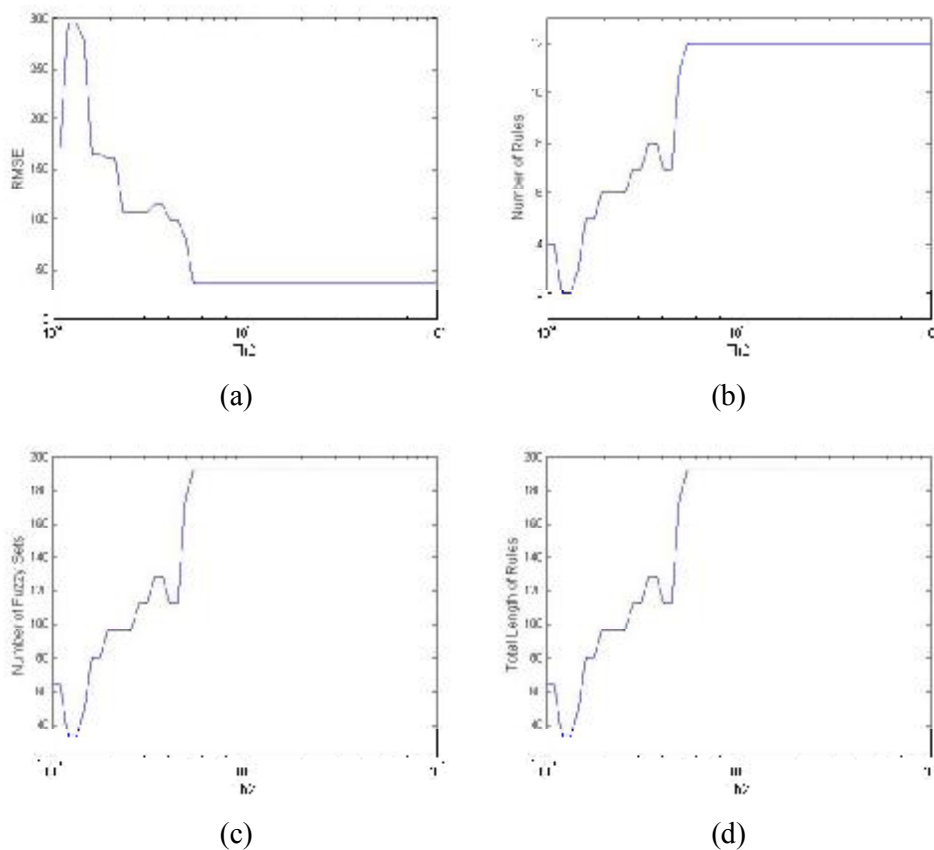


Figure 5-21. The performance of fuzzy models following the interpretability improvement with different  $Th_2$ : (a)  $RMSE$  versus  $Th_2$ ; (b) the number of rules versus  $Th_2$ ; (c) the number of fuzzy sets versus  $Th_2$ ; (d) the total length of rules versus  $Th_2$

3. The third step of the interpretability improvement is to remove redundant fuzzy sets.  $Th_3$  is used to define whether one fuzzy set is redundant. In this experiment,  $Th_3$  was set to be variable in the range  $0.5 < Th_3 < 1$  and other

## Chapter 5: FM-HCMO

thresholds and parameters were fixed at  $Th_1 = 0$ ;  $Th_2 = 1$ ;  $Th_4 = 1$ ;  $Th_5 = 1$ .

Figure 5-22 shows the system performance with different  $Th_3$ .

From this figure, it can be seen that, with the increase of  $Th_3$ , the *RMSE* of the result model tends to decrease; the number of rules does not change; the number of fuzzy sets and the total rule length increase. This means that the obtained model accuracy tends to increase while its interpretability decreases.

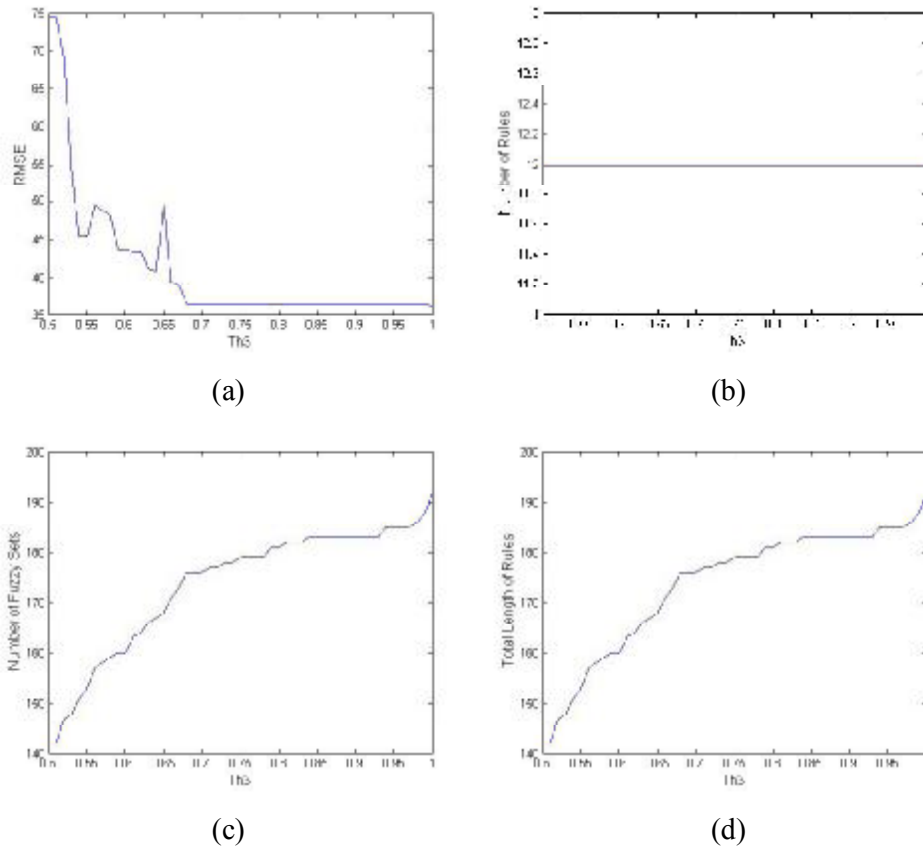


Figure 5-22. The performance of fuzzy models following the interpretability improvement with different  $Th_3$ : (a) *RMSE* versus  $Th_3$ ; (b) the number of rules versus  $Th_3$ ; (c) the number of fuzzy sets versus  $Th_3$ ; (d) the total length of rules versus  $Th_3$

## Chapter 5: FM-HCMO

---

4. The last step of interpretability improvement is to merge similar fuzzy sets.  $Th_4$  and  $Th_5$  are used to define whether two fuzzy sets are similar enough to be merged. Because  $Th_4$  and  $Th_5$  have the same effect, only  $Th_4$  was tested in this experiment. Also, it was set to be variable in the range  $0.8 < Th_4 < 1$  and other thresholds and parameters were fixed at  $Th_1 = 0$ ;  $Th_2 = 1$ ;  $Th_3 = 1$ ;  $Th_5 = 1$ . Figure 5-23 shows the model performance with different  $Th_4$ .

From this figure, it can be seen that, with the increase of  $Th_4$ , the *RMSE* of the obtained model has the tendency of decreasing; the number of rules and the total rule length do not change; the number of fuzzy sets increases, which means that the obtained model accuracy tends to increase while its interpretability decreases.

From these experiments, it can be seen that the thresholds  $Th_1 \sim Th_5$  can greatly affect the system performance in terms of accuracy as well as interpretability.

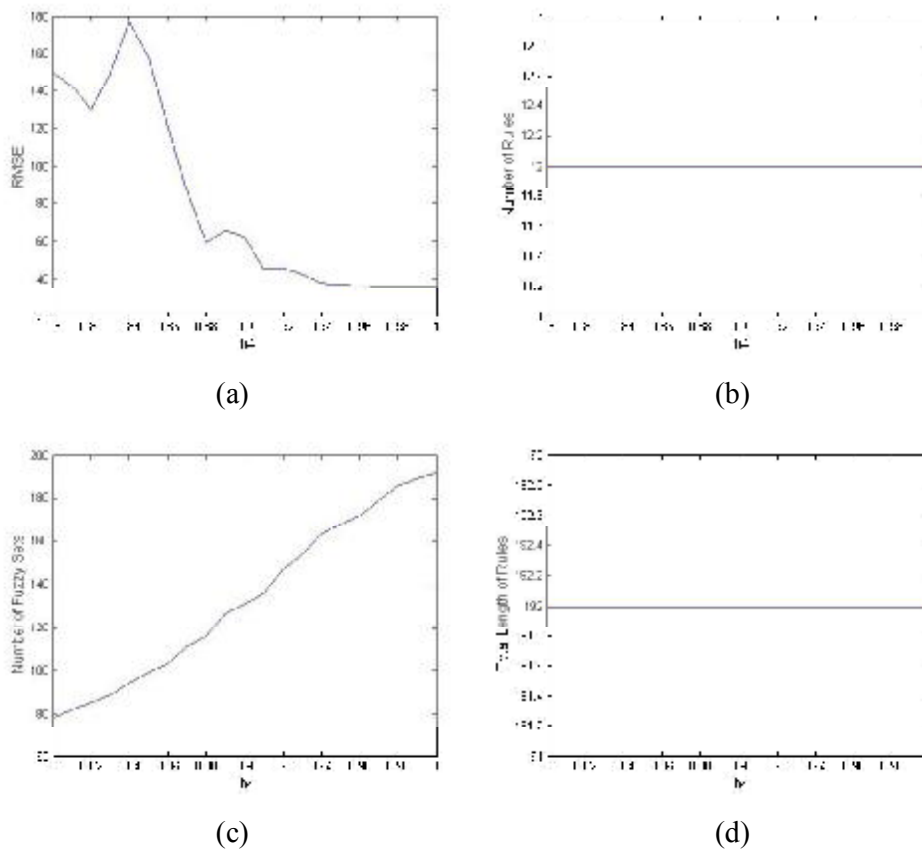


Figure 5-23. The performance of fuzzy models following the interpretability improvement with different  $Th_4$ : (a)  $RMSE$  versus  $Th_4$ ; (b) the number of rules versus  $Th_4$ ; (c) the number of fuzzy sets versus  $Th_4$ ; (d) the total length of rules versus  $Th_4$ .

### 5.5.3 The Multi-Objective Optimisation Mechanism

Based on the proposed four-step interpretability improvement approach, a multi-objective optimisation mechanism, which is intended to optimise both the accuracy and the interpretability of fuzzy systems, is developed. Figure 5-24 outlines the steps behind the proposed multi-objective optimisation mechanism. It works according to the following steps:



## **Chapter 5: FM-HCMO**

---

1. **Initial threshold values generation:** Randomly generate the thresholds' values within the predefined bounds.
2. **Interpretability improvement:** Based on the reduced training data, improve the previous fuzzy model in interpretability using the proposed 4-step improvement approach. In this step, the input rule-base is fixed and remains as such while the parameters of the membership functions and the thresholds vary after each loop. Following this step, a new fuzzy model is elicited.
3. **Performance evaluation:** The new fuzzy model is evaluated using the designed fitness functions (objective functions).
4. **Pareto-optimal fuzzy models preservation:** Compare the fitness of every generated model, preserve the adequate Pareto-optimal models via the archive mechanism in nMPSO.
5. **New parameters and thresholds generation:** This task is accomplished by the nMPSO algorithm based on some particular principles, which are related to the fitness values and the location of individual solutions.
6. **Termination estimation:** If the termination criteria are achieved, stop the mechanism and return the final Pareto-optimal fuzzy models; otherwise, replace the old membership function's parameters and threshold values



**5.5.4 The Designed Objective Functions for Fuzzy Modelling**

The multi-objective optimisation includes two aspects: One is the accuracy of the fuzzy model, which can be evaluated using the Root Mean Square Error (*RMSE*) index. This index has already been described in Section 5.3.3.1. Another aspect is the interpretability of the fuzzy model, which is affected by the number of fuzzy rules (*Nrule*), the number of fuzzy sets (*Nset*) and the total length of fuzzy rules (*Lrule*).

To normalise these two objectives and make them similar and comparable in scale, they are formulated as follows:

$$\begin{aligned} \text{Objective 1: } & \frac{RMSE}{RMSE_I}; \\ \text{Objective 2: } & \frac{Nrule}{Nrule_I} + \frac{Nset}{Nset_I} + \frac{Lrule}{Lrule_I}; \end{aligned} \tag{5.20}$$

where  $RMSE_I$  is the root mean square error of the fuzzy model that is not optimised using the multi-objective optimisation mechanism;  $Nrule_I$ ,  $Nset_I$  and  $Lrule_I$  represent the number of fuzzy rules, the number of fuzzy sets and the total rule length of this fuzzy model, respectively.

**5.6 Confidence Band Analysis**

Once the final fuzzy models have been elicited, confidence bands relating to predictions are computed. In other words, when given a prediction by a data-

## Chapter 5: FM-HCMO

---

driven fuzzy model, one wishes to know how confident can one be in such a prediction.

Normally, the standard deviation of the prediction errors of all the training data is computed in order to represent the confidence band. But this can only inform on a generalised view about the model. Specifically, it cannot provide particular guidance for one specific prediction.

In this work, a confidence band named  $\alpha\%$ -range confidence band is designed. It is calculated as follows:

1. When given a prediction value  $y^p$ , define a prediction scope  $S$  where the lower bound is  $y^p - 0.005 \times \alpha \times L_p$  and the upper bound is  $y^p + 0.005 \times \alpha \times L_p$ , with  $L_p$  being the total range of the prediction values and it equals to the maximal prediction value minus the minimal prediction value.
2. From all the training data, find the ones  $p_i$  with their prediction output values  $y_i^p$  including in the scope  $S$ , which is  $y_i^p \in S$ , where  $i = 1, 2, \dots, N_s$  and  $N_s$  is the number of training data  $p_i$ .
3. The  $\alpha\%$ -range confidence band  $CB$  is defined using the standard deviation of the prediction errors of the training data  $p_i$ :

$$CB = Std(Error) = \sqrt{\frac{\sum_{j=1}^{N_s} (error_j - \overline{error})^2}{N_s}}, \quad (5.21)$$

## Chapter 5: FM-HCMO

---

where  $Error = \{error_1, error_2, \dots, error_{N_s}\}$ ;  $error_j = y_j^p - y_j^m$ ;  $y_j^m$  are the measured output values of  $p_i$ ;  $i = 1, 2, \dots, N_s$ .

For an obtained model, it is not realistic to calculate the  $\alpha\%$ -range confidence band for every possible prediction. Generally, some averagely distributed prediction values are selected to provide some confidence bands which will be viewed as the representatives of all the possible prediction values.

Figure 5-25 shows an example of the  $\alpha\%$ -range confidence band, which is based on a fuzzy model of UTS prediction (12 rules). In this case,  $\alpha$  is set to 5 and 200 evenly distributed prediction values are chosen and calculated for the confidence bands. Figure 5-25(a) shows the prediction performance of the fuzzy model for all the training data (2820 data). Figure 5-25(b) gives the confidence bands of this model. From this figure, one can infer more details of how confident one can be about a prediction. For instance, when a prediction is 1500, which is shown with the red dashed line in the figures, its confidence band is around 108; while the prediction 800 has a confidence band of only 32, which is shown with the green dashed-dotted line.

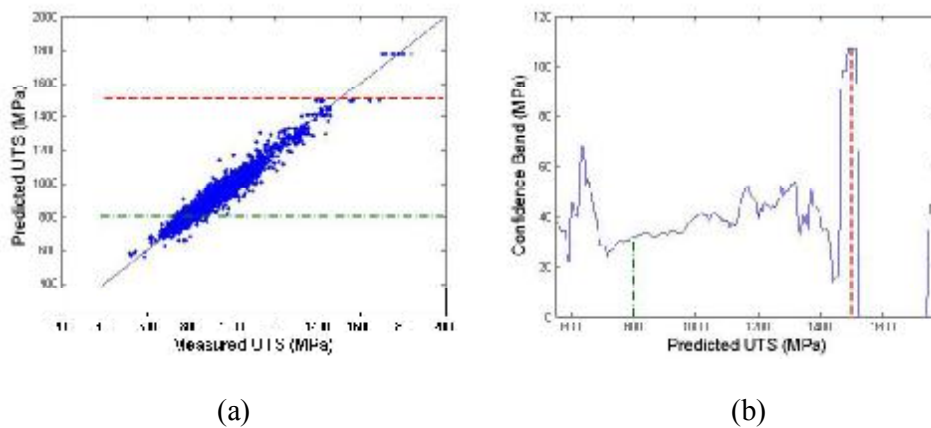


Figure 5-25. Example of the  $\alpha$ %-range confidence band: (a) the prediction performance of the UTS model; (b) the  $\alpha$ %-range confidence band of the UTS model.

### 5.7 Summary

In this chapter, a framework for data-driven fuzzy modelling is proposed in order to construct linguistic fuzzy models considering both accuracy and interpretability of fuzzy systems. In this methodology, a new agglomerative complete-link clustering algorithm is first developed and applied to construct the initial fuzzy model. A new data selection technique is then proposed to select representative training data used to improve the modelling efficiency. A multi-objective optimisation mechanism is developed for the performance improvement of such models, which takes into account both the accuracy and the interpretability attributes. Finally, a method for computing the confidence band relating to the model prediction analysis is proposed. All the above proposed techniques were validated via a series of experiments using real data, from the steel industry. The

## ***Chapter 5: FM-HCMO***

---

next will focus specifically on how such techniques can be applied to modelling problems.

## **Chapter 6**

# **Application of FM-HCMO to Fuzzy Modelling Problems**

### **6.1 Introduction**

In order to validate the effectiveness of the proposed modelling strategy named FM-HCMO (see Chapter 5), the associated algorithm was applied to the modelling of two benchmark problems, one is a problem of static nonlinear system approximation and the other is a dynamical system identification problem. Furthermore, FM-HCMO was applied to the modelling of the mechanical properties of alloy steels using real industrial data.

### **6.2 The Nonlinear Function Approximation**

In this experiment, the proposed fuzzy modelling approach was used to



## **Chapter 6: Application of FM-HCMO in Modelling Problems**

approximate the following two-input-single-output nonlinear static system, which is also introduced in [Sugeno & Yasukawa 1993]:

$$y = (1 + x_1^{-2} + x_2^{-1.5})^2, \quad 1 \leq x_1, x_2 \leq 5. \quad (6.1)$$

In order to establish a quantitative comparison with the results obtained in various papers, the training data set was selected as being the same as the one described in [Sugeno & Yasukawa 1993], which consists of 50 data points. Furthermore, another 50 randomly generated data points were used for model testing.

In this case, the initial fuzzy model was obtained using 8 clusters, resulting in a model with 8 rules and 24 fuzzy sets; the maximum number of function evaluation for nPSO and nMPSO were set to 20,000 and 20,000 respectively. Two optimisation objectives used by nMPSO have already been introduced in Section 5.5.4, which is as follows:

$$\begin{aligned} \text{Objective 1: } & \frac{RMSE}{RMSE_I}; \\ \text{Objective 2: } & \frac{Nrule}{Nrule_I} + \frac{Nset}{Nset_I} + \frac{Lrule}{Lrule_I}; \end{aligned} \quad (6.2)$$

where  $RMSE_I$  is the root mean square error of the fuzzy model that is not optimised using the multi-objective optimisation mechanism;  $Nrule_I$ ,  $Nset_I$  and  $Lrule_I$  represent the number of fuzzy rules, the number of fuzzy sets and the total rule length of this fuzzy model, respectively.

After the final multi-objective optimisation, a set of Pareto-optimal fuzzy models

## Chapter 6: Application of FM-HCMO in Modelling Problems

was obtained. Figure 6-1 shows their performances with respect to various indices, including the root mean square error, the number of fuzzy rules, the number of fuzzy sets and the length of the fuzzy rules.

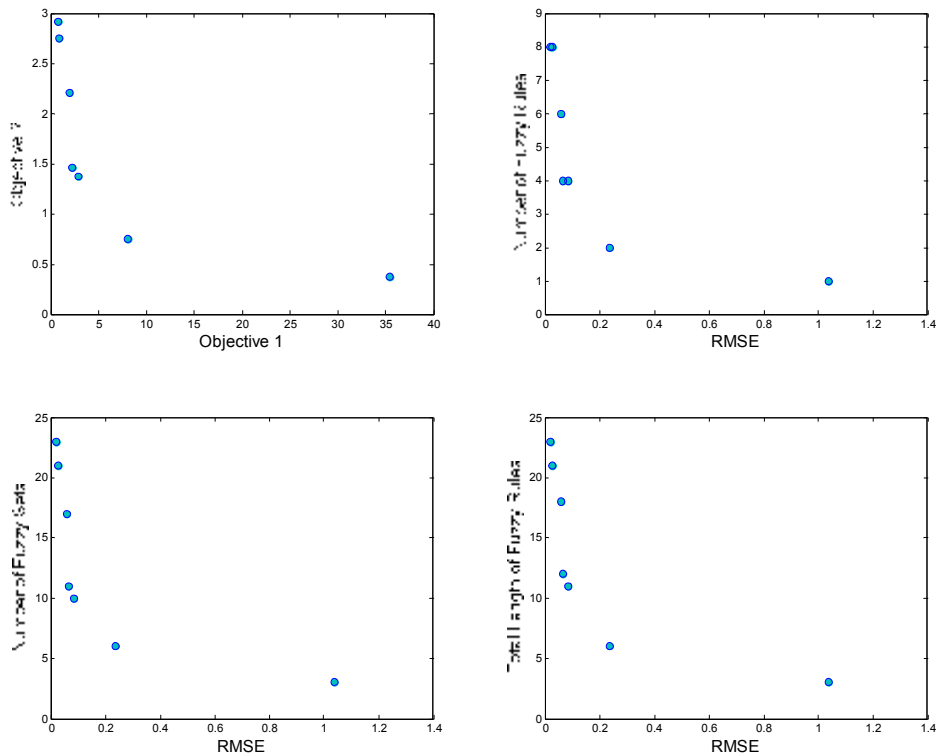


Figure 6-1. The performance of the optimised Pareto-optimal fuzzy models for the nonlinear function approximation problem (6.1)

Table 6-1 describes the experimental results compared with those published via other research studies. Three models out of all the Pareto-optimal models, which include 8, 6 and 4 rules respectively, are chosen as the representatives and are listed in this table. It can be seen that FM-HCMO performs better than the other methods, whose strategy is based on linguistic fuzzy systems [Sugeno & Yasukawa 1993]; for the method based on singleton fuzzy systems [Rojas *et al.* 2000], it

## ***Chapter 6: Application of FM-HCMO in Modelling Problems***

---

needed more fuzzy rules to reach the same accuracy level as that of FM-HCMO. For the methods based on TSK fuzzy systems [Kim *et al.* 1997; Chen & Linkens 2004, Wang *et al.* 2005], sometimes, they may perform slightly better than FM-HCMO in accuracy. But compared with linguistic fuzzy models, TSK fuzzy models are not transparent and interpretable enough, since linear equations, instead of the linguistic terms, operate as the consequent part of the fuzzy rules.

Figure 6-2 shows the prediction performance of the initial as well as the three selected fuzzy models and Figure 6-3 illustrates the distribution of their membership functions relating to two inputs ( $x_1$  and  $x_2$ ). It can be seen that, for these optimised models, more rules and more parameters will bring more accuracy while the models with fewer rules and parameters are simpler in structure and easier to understand.

## Chapter 6: Application of FM-HCMO in Modelling Problems

Table 6-1. The performance comparison of various models for the nonlinear function approximation problem

Fuzzy model	Number of fuzzy rules	Number of fuzzy sets for input and output parts	Total rule length	RMSE of training data	RMSE of testing data
[Sugeno & Yasukawa 1993], Linguistic models	6 (initial)	Input: 12 Output: 6	12	0.564	N/A
	6 (optimised)	Input: 12 Output: 6	12	0.281	N/A
[Kim <i>et al.</i> 1997], TSK model	3	Input: 6	6	0.140	N/A
[Rojas <i>et al.</i> 2000], Singleton models	9 (case 1)	Input: 6	18	0.513	N/A
	16 (case 2)	Input: 8	32	0.176	N/A
	25 (case 3)	Input: 10	50	0.066	N/A
[Chen and Linkens 2004], TSK models	4 (initial)	Input: 8	N/A	N/A	0.066
	4 (optimised)	Input: 5	N/A	N/A	0.088
	3 (optimised)	Input: 5	N/A	N/A	0.138
	2 (optimised)	Input: 4	N/A	N/A	0.275
[Wang <i>et al.</i> 2005], TSK models	6 (initial)	Input: 12	12	0.176	N/A
	7 (optimised)	Input: 6	14	0.030	N/A
	4 (optimised)	Input: 3	6	0.052	N/A
	3 (optimised)	Input: 2	4	0.072	N/A
FM-HCMO	8 (initial)	Input: 16 Output: 8	16	0.527	0.527
	8 (optimised)	Input: 14 Output: 7	13	0.026	0.082
	6 (optimised)	Input: 11 Output: 6	12	0.059	0.114
	4 (optimised)	Input: 7 Output: 4	8	0.084	0.162

**Chapter 6: Application of FM-HCMO in Modelling Problems**

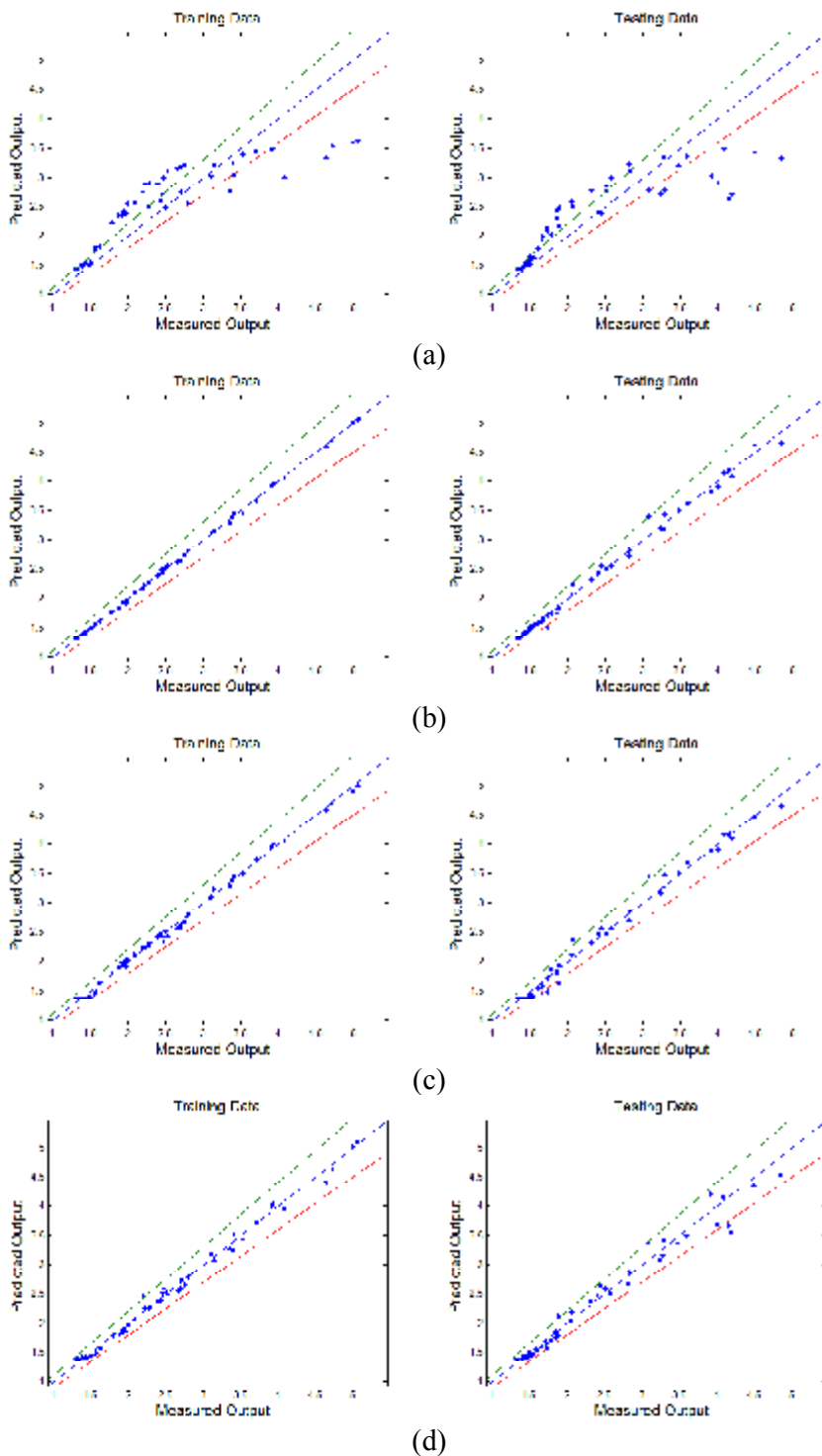


Figure 6-2. The fuzzy models' predicted outputs versus the measured outputs with the nonlinear function approximation problem: (a) the initial model, (b) an optimised model with 8 rules, (c) an optimised model with 6 rules, and (d) an optimised model with 4 rules; the green and red lines represent the +10% and -10% error bands respectively.

**Chapter 6: Application of FM-HCMO in Modelling Problems**

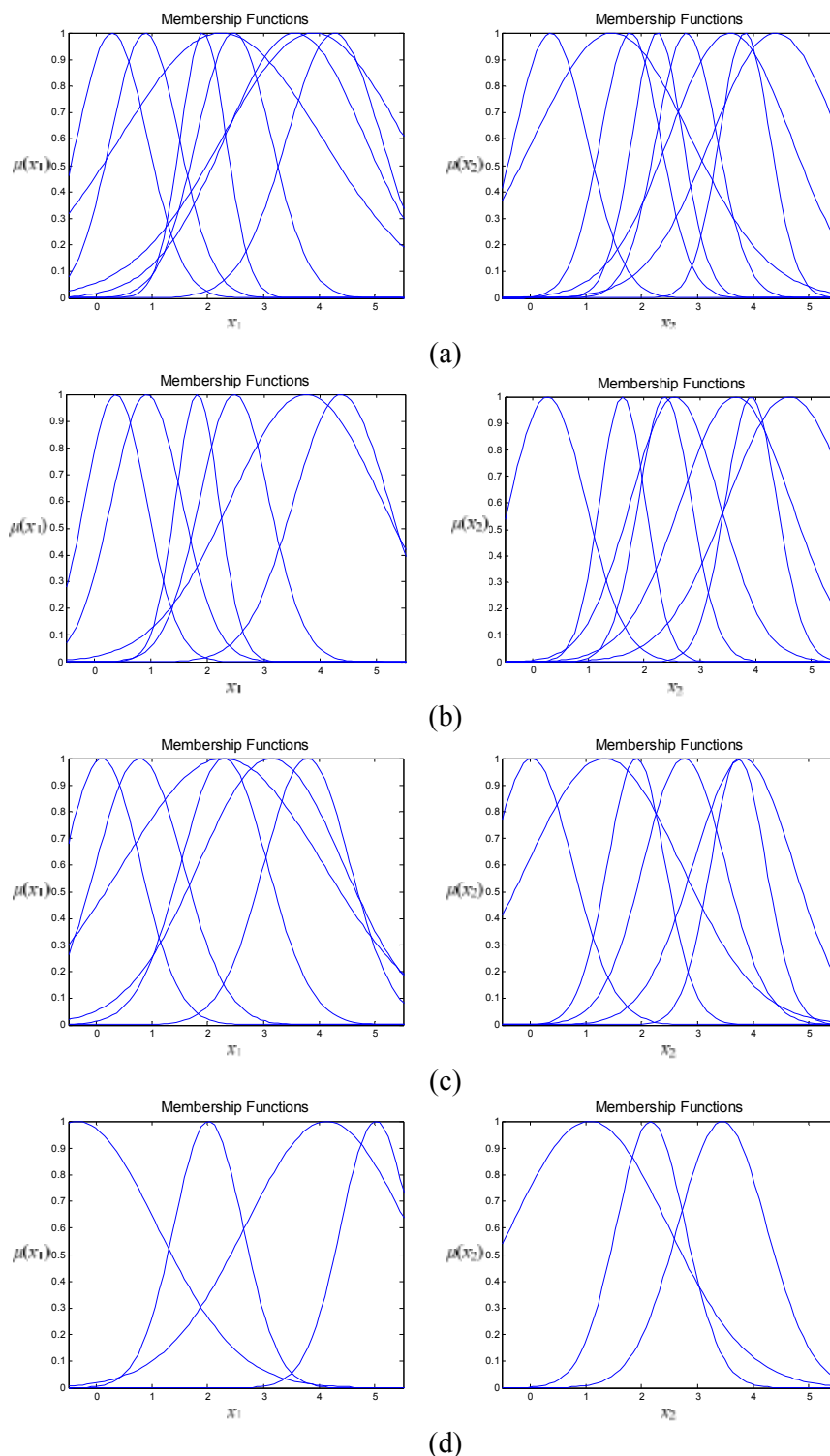


Figure 6-3. The fuzzy models' membership functions with the nonlinear function approximation problem: (a) the initial model, (b) an optimised model with 8 rules, (c) an optimised model with 6 rules, and (d) an optimised model with 4 rules

## **Chapter 6: Application of FM-HCMO in Modelling Problems**

To provide more details about these Pareto-optimal models, Figure 6-4 shows the rule-base relating to the optimised system, which is the one associated with 8 rules, its other information being included in Table 6-1, Figures 6-2(b) and 6-3(b). For this fuzzy model, the linguistic hedges approach [Zadeh 1972; Fukuyama & Sugeno 1989; Chen & Linkens 2001; Chen & Linkens 2004] can be employed to derive the corresponding linguistic rules as follows:

R<sub>1</sub>: IF  $x_2$  is *small*, THEN  $y$  is *quite large*.

R<sub>2</sub>: IF  $x_1$  is *small* AND  $x_2$  is *large*, THEN  $y$  is *medium*.

R<sub>3</sub>: IF  $x_1$  is *more or less quite large* AND  $x_2$  is *slightly small*, THEN  $y$  is *slightly small*.

R<sub>4</sub>: IF  $x_1$  is *small*, THEN  $y$  is *large*.

R<sub>5</sub>: IF  $x_1$  is *medium* AND  $x_2$  is *quite large*, THEN  $y$  is *small*.

R<sub>6</sub>: IF  $x_1$  is *slightly small* AND  $x_2$  is *medium*, THEN  $y$  is *slightly small*.

R<sub>7</sub>: IF  $x_2$  is *large*, THEN  $y$  is *small*.

R<sub>8</sub>: IF  $x_1$  is *large* AND  $x_2$  is *medium*, THEN  $y$  is *small*.

Figure 6-5 shows the three-dimensional input/output surfaces of the actual system. The 8-rule fuzzy system and its 5%-range confidence band (see Section 5.6) are displayed in Figure 6-6.

**Chapter 6: Application of FM-HCMO in Modelling Problems**

---

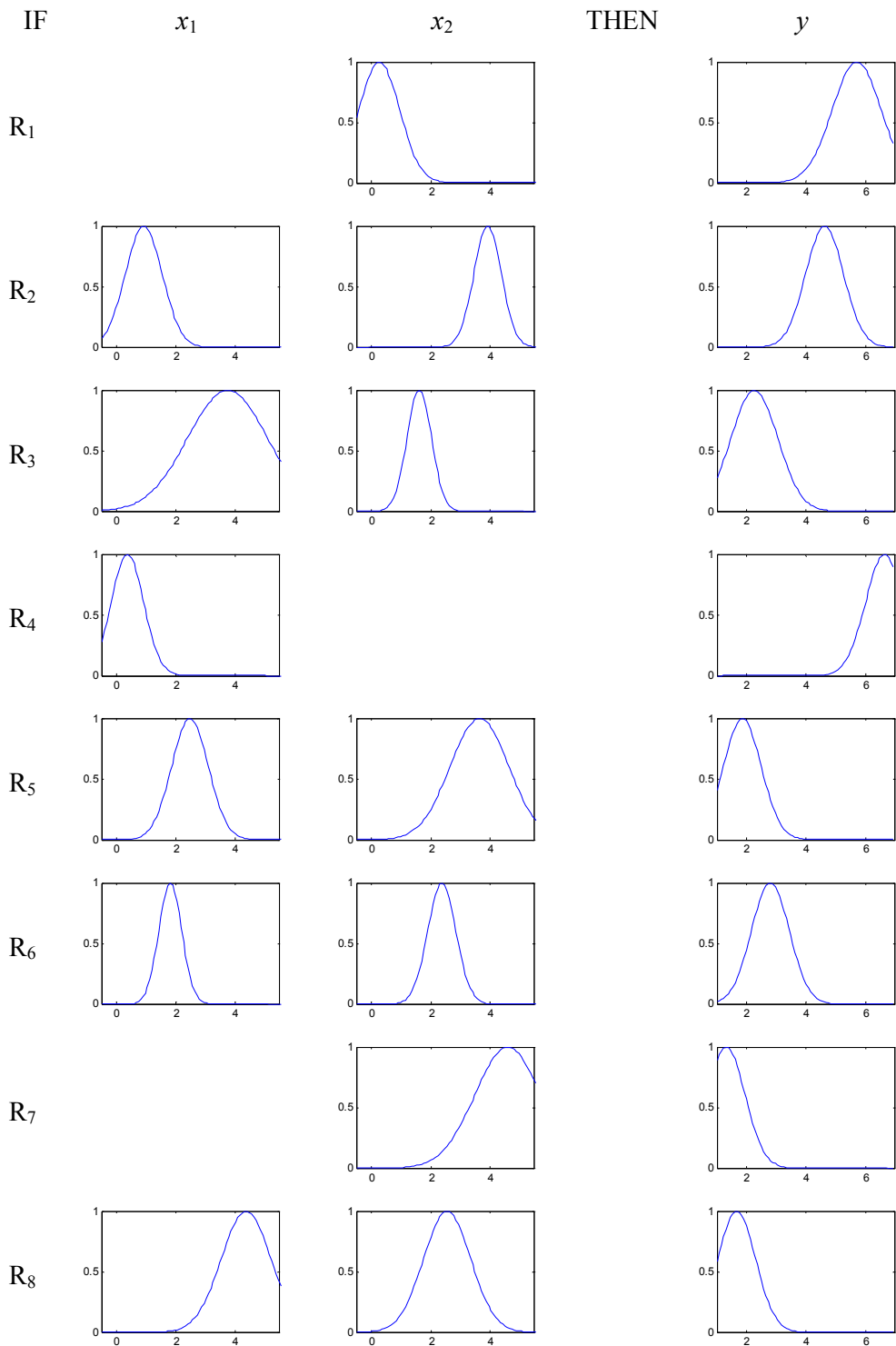


Figure 6-4. The optimised 8-rule fuzzy model for the nonlinear function approximation problem



## Chapter 6: Application of FM-HCMO in Modelling Problems

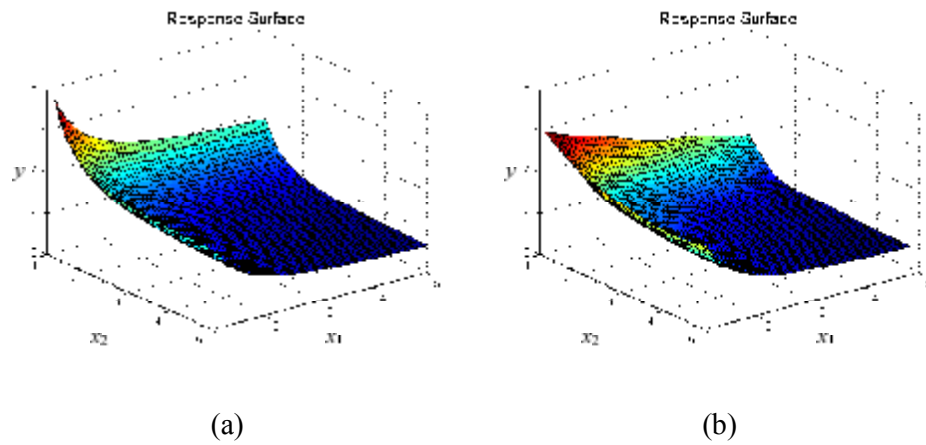


Figure 6-5. Response surfaces for the nonlinear function approximation problem: (a) the actual system and (b) the optimised 8-rule fuzzy system

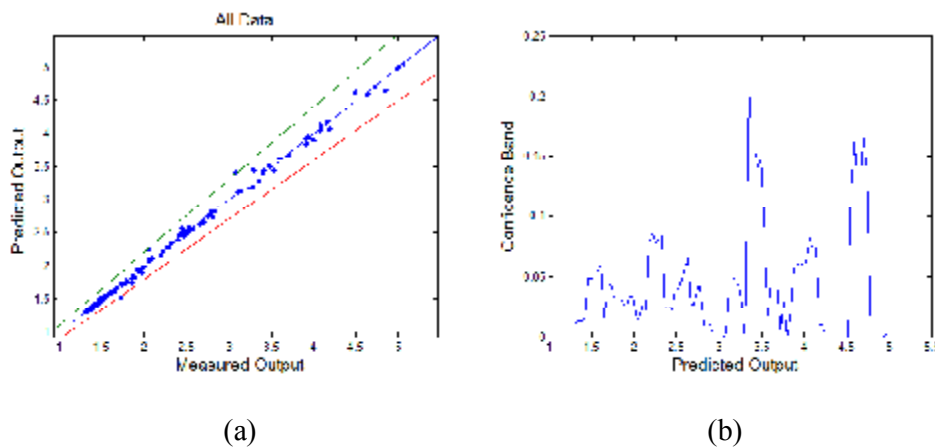


Figure 6-6. (a) The prediction performance and (b) the 5%-range confidence band of the optimised 8-rule fuzzy model for the nonlinear function approximation problem

### 6.3 The Identification of a Dynamic System

In this problem, the modelling target is a nonlinear second-order plant, which has

## ***Chapter 6: Application of FM-HCMO in Modelling Problems***

---

been studied in [Yen & Wang 1998; Yen & Wang 1999; Wang & Yen 1999; Setnes & Roubos 2000],

$$y(k) = g(y(k-1), y(k-2)) + u(k), \quad (6.3)$$

where

$$g(y(k-1), y(k-2)) = \frac{y(k-1)y(k-2)(y(k-1) - 0.5)}{1 + y^2(k-1) - y^2(k-2)}. \quad (6.4)$$

where  $y()$  is the output of the system;  $g()$  is a nonlinear component;  $u()$  is the input signal;  $k$  is the index of the input signals.

The output of this system depends on both its past values and the current input. The modelling goal is to approximate the nonlinear component  $g(y(k-1), y(k-2))$ , which is usually called the ‘unforced system’ in the control literature. Similarly to the settings reported in [Setnes & Roubos 2000], 400 simulated data points were generated from the plant model (6.3). With the starting equilibrium state (0, 0), 200 samples of training data were obtained by using a random input signal  $u(k)$  that is uniformly distributed in the interval [-1.5, 1.5], while the rest 200 samples of testing data were generated using a sinusoidal input signal  $u(k) = \sin(2\pi k/25)$ . These 400 simulated data points are shown in Figure 6-7.

## Chapter 6: Application of FM-HCMO in Modelling Problems

---

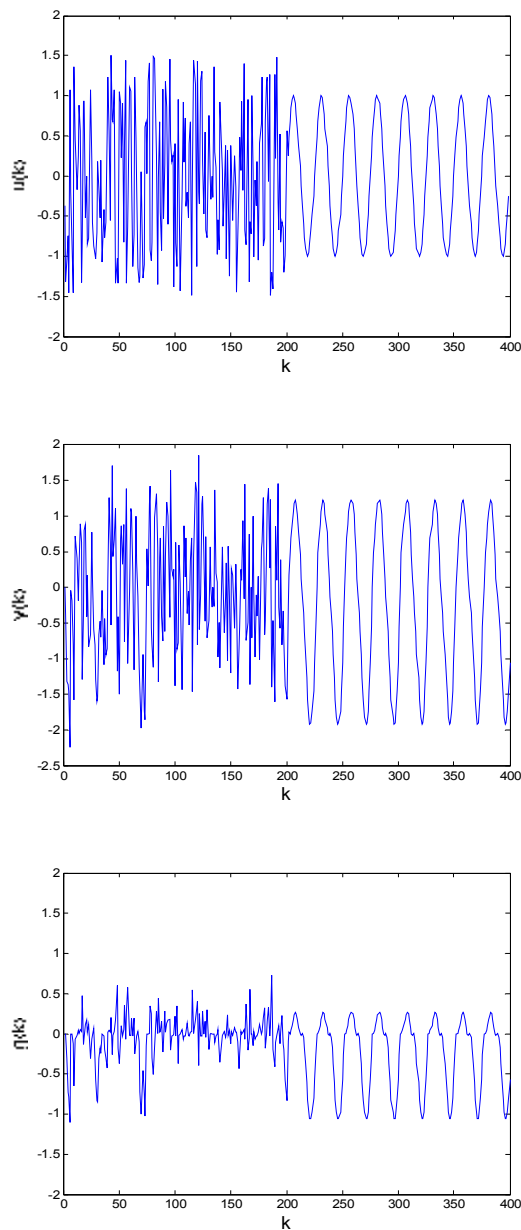


Figure 6-7. The input  $u(k)$ , forced system  $g(k)$ , and output  $y(k)$  of the plant (6.3)

In this case, the initial fuzzy model was also obtained with 8 clusters; the maximal function evaluation numbers for nPSO and nMPSO were both set to be 20,000. After the optimisation scheme, 13 non-dominated solutions (fuzzy models) were

## Chapter 6: Application of FM-HCMO in Modelling Problems

obtained. Figure 6-8 demonstrates the trade-offs among the multiple objectives and criteria within these non-dominated fuzzy system solutions. The upper left figure gives the trade-off relationship between the designed accuracy objective and interpretability objective (see Section 5.5.4); the upper right figure illustrates the trade-off between the accuracy criterion (RMSE) and the fuzzy rules number; the lower left figure is for the trade-off between the accuracy and the number of fuzzy sets and the lower right one shows the relationship between the accuracy and the total length of fuzzy rules.

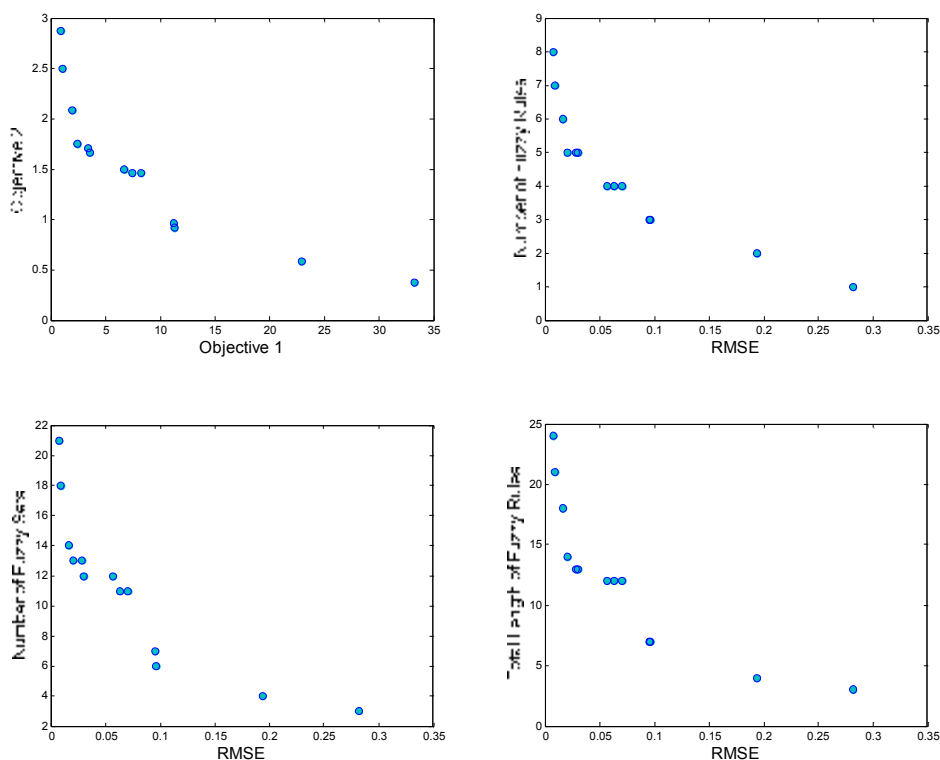


Figure 6-8. The performance of the optimised Pareto-optimal models for the dynamical system identification problem

Table 6-2 compares the experimental results with some other studies previously

## ***Chapter 6: Application of FM-HCMO in Modelling Problems***

---

reported in the literature [Yen & Wang 1998; Wang & Yen 1999; Yen & Wang 1999; Setnes & Roubos 2000]. Three optimised models out of all the Pareto-optimal models, which include 6, 4 and 3 rules respectively, together with the initial generated model are listed in this table. It can be seen that FM-HCMO is able to produce more compact and simpler models compared to the other methods, since the modelling strategies reported in [Yen & Wang 1998; Wang & Yen 1999; Yen & Wang 1999] needed more fuzzy rules and fuzzy sets to achieve the same accuracy level as that of FM-HCMO. In other words, this proposed approach strikes a good balance between numerical accuracy and model simplicity, compared to the above mentioned fuzzy modelling methods.

The prediction performance of the initial and the three selected fuzzy models is shown in Figure 6-9. The distribution of their membership functions relating to the inputs  $y(k - 1)$  and  $y(k - 2)$  is displayed in Figure 6-10. It can be observed that these Pareto-optimal models exhibit fuzzy sets pattern behaviour, which means that they provide a wider choice of different solutions to users.

**Chapter 6: Application of FM-HCMO in Modelling Problems**

Table 6-2. The performance comparison of various models for the dynamical system identification problem

Fuzzy model	Number of fuzzy rules	Number of fuzzy sets for input and output part	Total rule length	RMSE of training data	RMSE of testing data
[Yen & Wang 1998], Singleton models	36 (initial)	Input: 12	72	0.005	0.072
	23 (optimised)	Input: 12	46	0.006	0.038
[Yen & Wang 1999], Singleton models	25 (initial)	Input: 25	50	0.015	0.020
	20 (optimised)	Input: 20	40	0.026	0.015
[Wang & Yen 1999], Singleton models	40 (initial)	Input: 40	80	0.018	0.026
	28 (optimised)	Input: 28	56	0.018	0.024
[Setnes & Roubos 2000], Singleton models	7 (initial)	Input: 14	14	0.126	0.035
	7 (optimised)	Input: 14	14	0.055	0.022
FM-HCMO	8 (initial)	Input: 16 Output: 8	16	0.171	0.292
	6 (optimised)	Input: 8 Output: 6	12	0.017	0.028
	4 (optimised)	Input: 7 Output: 4	8	0.063	0.084
	3 (optimised)	Input: 3 Output: 3	4	0.096	0.114

**Chapter 6: Application of FM-HCMO in Modelling Problems**

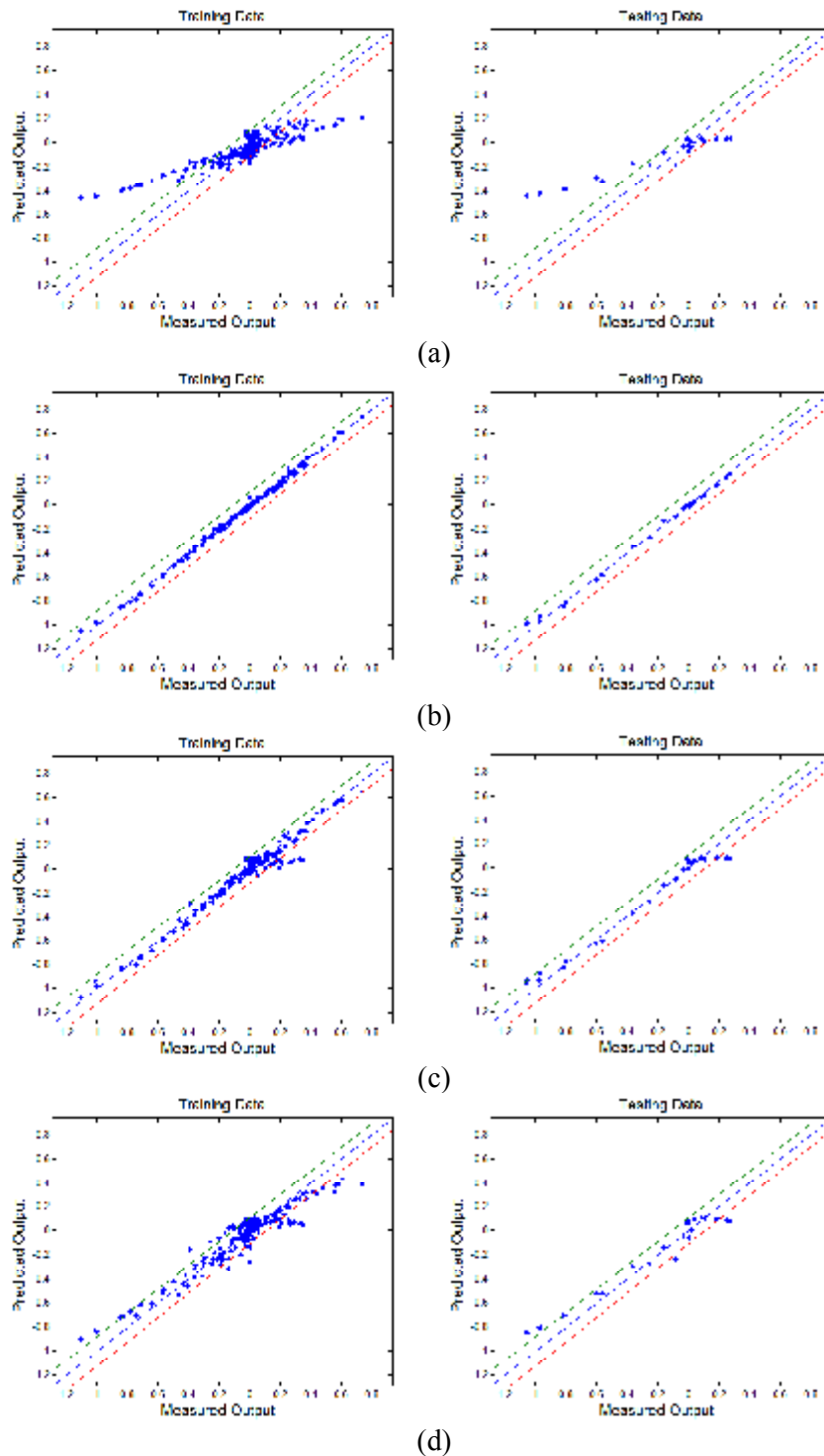


Figure 6-9. The fuzzy models' predicted outputs versus measured outputs with the dynamical system identification problem: (a) the initial model, (b) an optimised model with 6 rules, (c) an optimised model with 4 rules, and (d) an optimised model with 3 rules; the green and red lines represent the +10% and -10% error bands respectively.

**Chapter 6: Application of FM-HCMO in Modelling Problems**

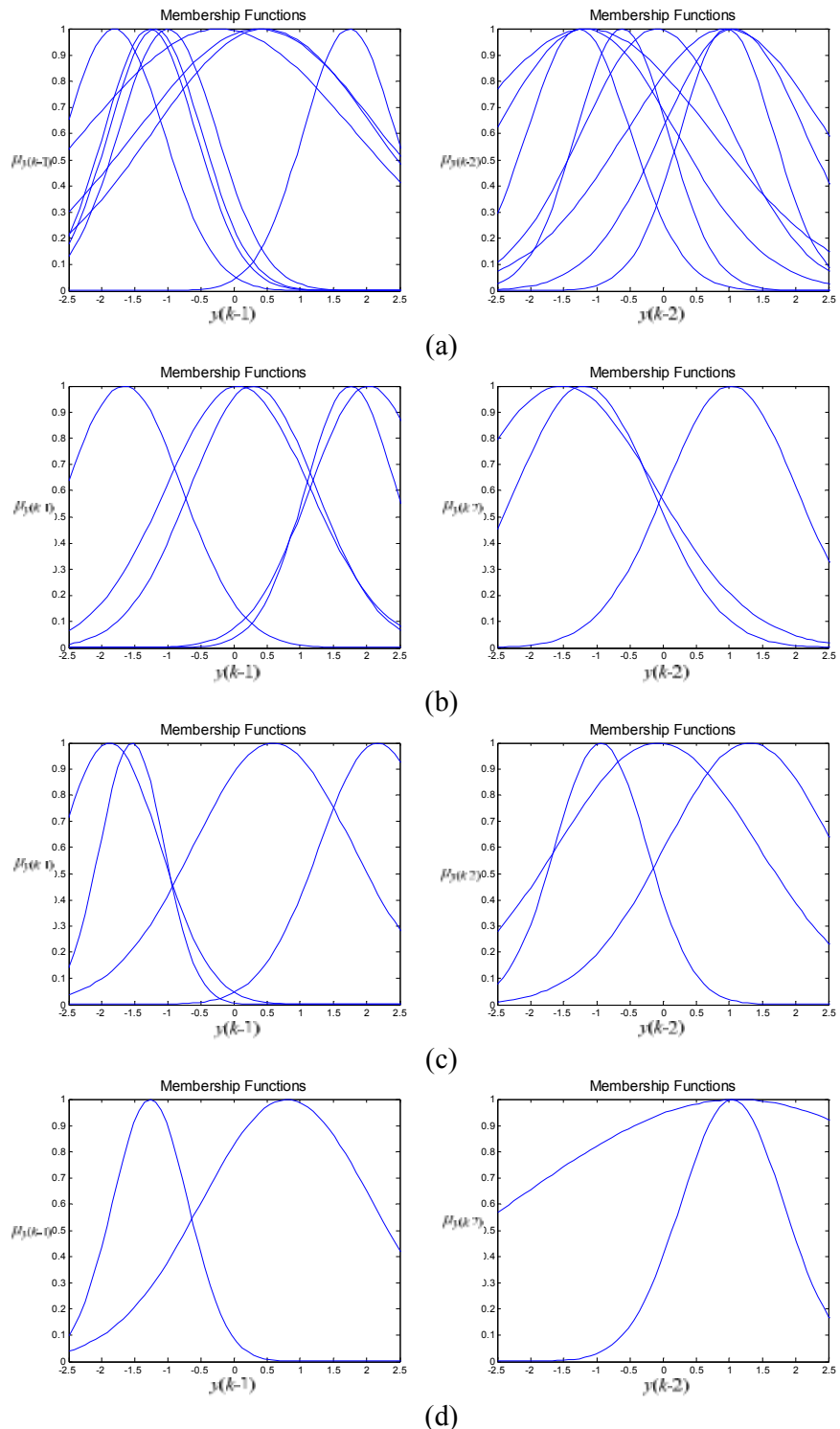


Figure 6-10. The fuzzy models' membership functions with the dynamical system identification problem: (a) the initial model, (b) an optimised model with 6 rules, (c) an optimised model with 4 rules, and (d) an optimised model with 3 rules



## **Chapter 6: Application of FM-HCMO in Modelling Problems**

To provide more details about these non-dominated models, the fuzzy rule-base of an optimised model, which includes 8 rules, is shown in Figure 6-11. Other details about this model can also be found in Table 6-2, Figures 6-9(b) and 6-10(b). For this fuzzy model, the following linguistic rules can be derived by using the linguistic hedges approach [Zadeh 1972; Fukuyama & Sugeno 1989; Chen & Linkens 2001; Chen & Linkens 2004]:

R<sub>1</sub>: IF  $y(k-1)$  is *small* AND  $y(k-2)$  is *quite large*, THEN  $g(k)$  is *large*.

R<sub>2</sub>: IF  $y(k-1)$  is *large* AND  $y(k-2)$  is *quite large*, THEN  $g(k)$  is *quite large*.

R<sub>3</sub>: IF  $y(k-1)$  is *small* AND  $y(k-2)$  is *slightly small*, THEN  $g(k)$  is *small*.

R<sub>4</sub>: IF  $y(k-1)$  is *medium* AND  $y(k-2)$  is *more or less small*, THEN  $g(k)$  is *medium*.

R<sub>5</sub>: IF  $y(k-1)$  is *large* AND  $y(k-2)$  is *slightly small*, THEN  $g(k)$  is *slightly small*.

R<sub>6</sub>: IF  $y(k-1)$  is *medium* AND  $y(k-2)$  is *quite large*, THEN  $g(k)$  is *medium*.

Figure 6-12 shows the three-dimensional response surfaces of the actual system and the optimised 6-rule fuzzy system. It can be observed that these two surfaces are perfectly matched. The 6-rule model's training and testing errors and its 5%-range confidence band are displayed in Figures 6-13 and 6-14, respectively.

**Chapter 6: Application of FM-HCMO in Modelling Problems**

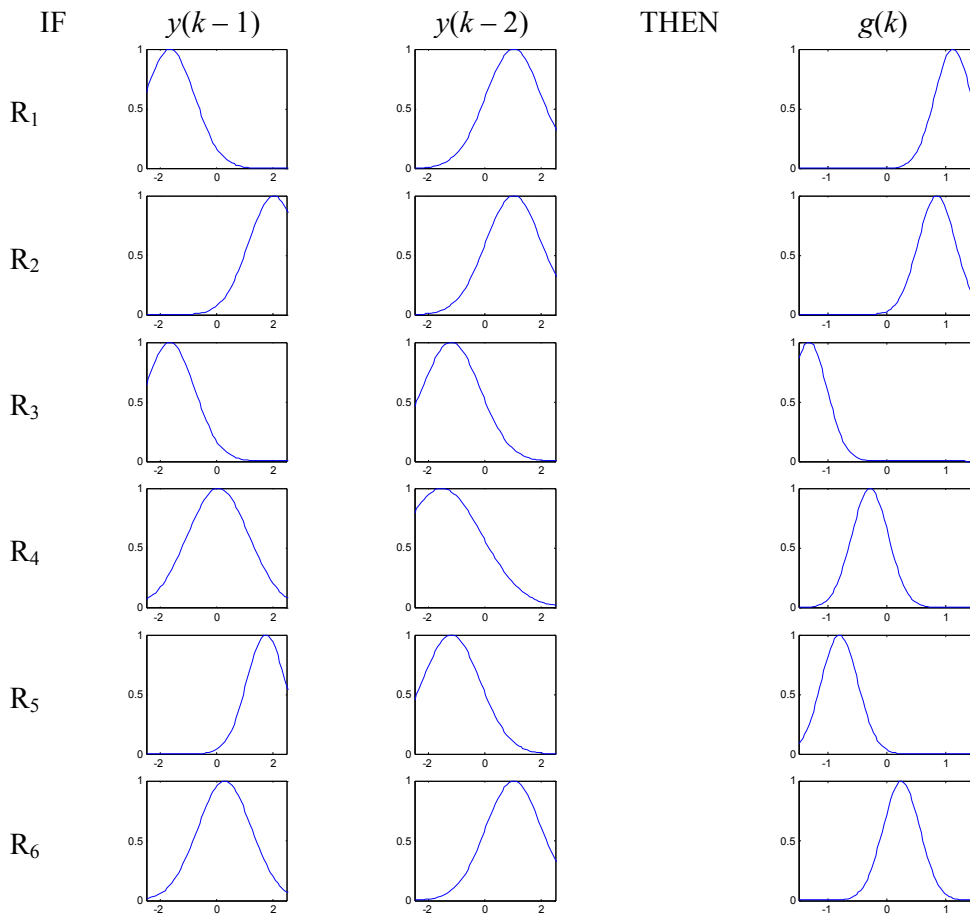


Figure 6-11. The optimised 6-rule fuzzy model for the dynamical system identification problem

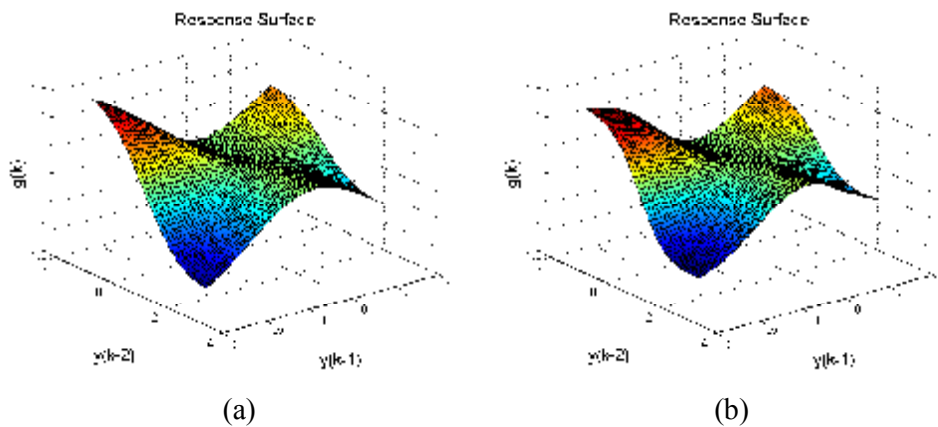


Figure 6-12. Response surfaces for the dynamical system identification problem: (a) the actual system and (b) the optimised 6-rule fuzzy model

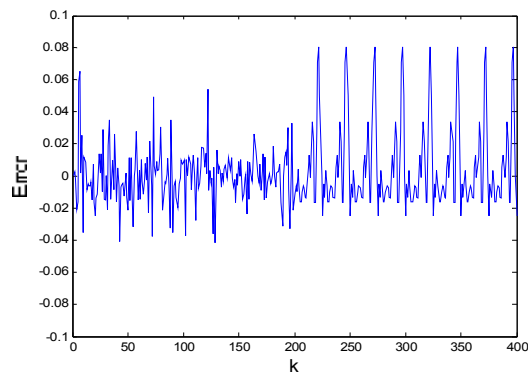


Figure 6-13. Training and testing errors of the optimised 6-rule fuzzy model for the dynamical system identification problem

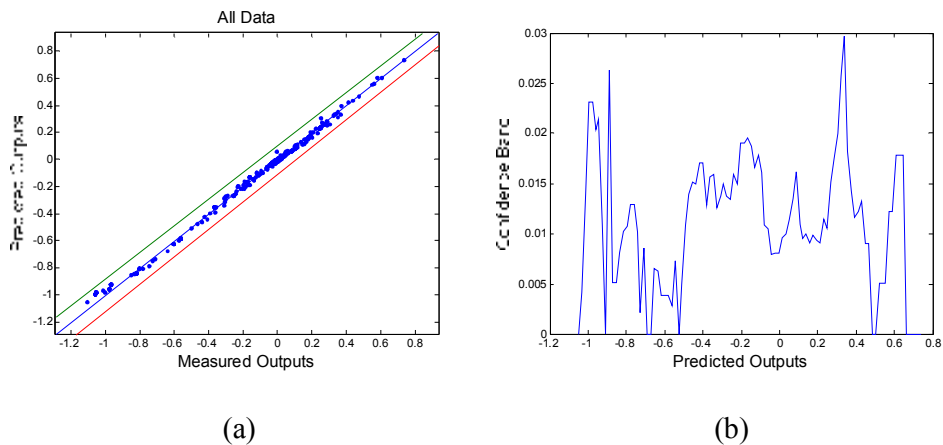


Figure 6-14. (a) The prediction performance and (b) the 5%-range confidence band of the optimised 6-rule fuzzy model for the dynamical system identification problem

#### 6.4 Mechanical Property Prediction of Alloy Steels

In material engineering, it is important to establish appropriate and reliable property prediction models for materials design and development. In the past, several mechanical property models were developed which were mainly based on

## ***Chapter 6: Application of FM-HCMO in Modelling Problems***

---

linear regression methods [Pickering 1978] or artificial neural networks [Hodgson 1996; Chen *et al.* 1998; Bakshi & Chatterjee 1998]. 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 techniques and the knowledge behind this kind of models cannot be understood fully. Thus, developing a fast, efficient and transparent data-driven modelling framework for material property prediction is still needed. In this situation, fuzzy modelling, such as the proposed FM-HCMO, provides an ideal approach because of its interpretable structure and its excellent ability of learning from data.

In general, the problem of modelling the properties of metal materials can be broadly stated as follows: given a certain material which undergoes a specified set of manufacturing processes, what are the final mechanical properties of this material [Chen & Linkens 2001]? Typical final mechanical properties that one may be interested in are Ultimate Tensile Strength (UTS), Reduction of Area (ROA), Elongation, and Impact Energy.

### **6.4.1 Ultimate Tensile Strength (UTS)**

In this modelling case, 3760 UTS data were used, 75% of the data were used for training, 10% of the data were used for validation (for validation mechanism, see Section 5.4.1.1) and the remaining 15% were used for final testing. These UTS

## ***Chapter 6: Application of FM-HCMO in Modelling Problems***

---

data include 15 inputs and one output, which is considered to be a high-dimensional problem for modelling purposes. The inputs consist of the weight percentages for the chemical composites, namely Carbon (C), Silica (Si), Manganese (Mn), Sulphur (S), Chromium (Cr), Molybdenum (Mo), Nickel (Ni), Aluminium (Al) and Vanadium (V), the test depth, size and the site where the processing of the alloy steel took place, the cooling medium, as well as the hardening and tempering temperatures.

In this experiment, the initial number of clusters was set to 15, which means that the initial fuzzy model was generated using 15 rules. For the optimisation algorithms nPSO and nMPSO, the maximum numbers of function evaluation were both set to 50,000. After the operation of the training data selection mechanism (see Section 5.4.2), 440 data points out of 2820 data points (all the training data) were selected. They were used in the following training process and worked as the representatives of all the training data.

Finally, a set of non-dominated solutions (fuzzy models) were obtained. Figure 6-15 demonstrates the trade-offs among the multiple objectives and criteria, including the RMSE, the number of fuzzy rules, the number of fuzzy sets and the total length of fuzzy rules, within these Pareto-optimal fuzzy models.

## Chapter 6: Application of FM-HCMO in Modelling Problems

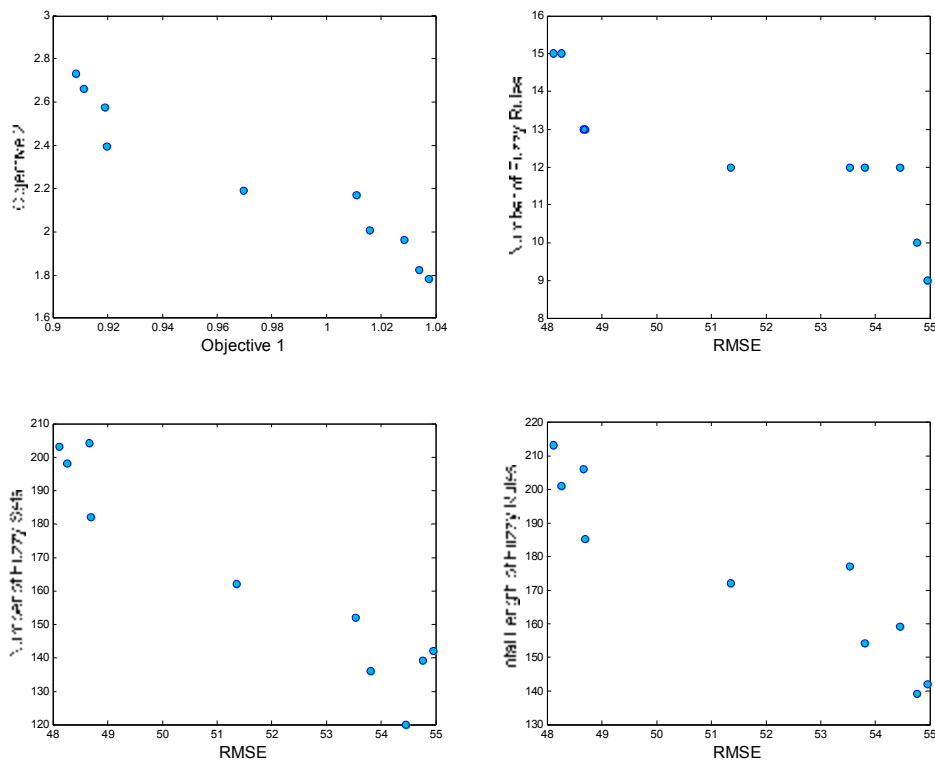


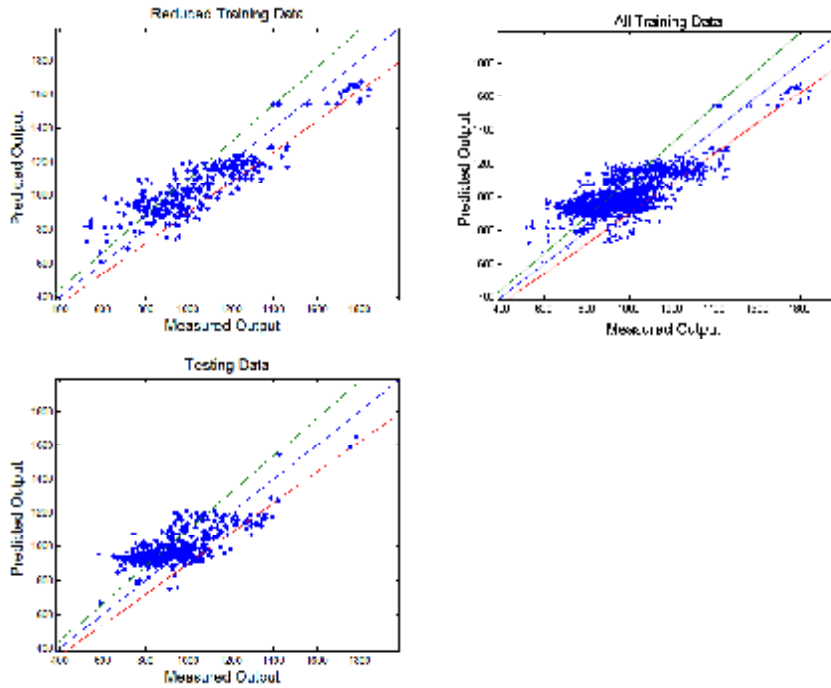
Figure 6-15. The performance of the Pareto-optimal UTS models

Table 6-3 includes the main parameters of the initial model and the two optimised models, which are selected from all the Pareto-optimal models with 13 and 10 rules respectively. Figure 6-16 shows the prediction performance of these models and Figure 6-17 shows the distribution of the membership functions relating to Input 4 (weight percentage for Carbon), which is an instance of all the membership functions. From Figure 6-16, it can also be seen that the selected training data work well as the representatives of all the training data. By using these reduced data instead of all the training data, much time and effort can be saved.

**Chapter 6: Application of FM-HCMO in Modelling Problems**

Table 6-3. Main parameters of the obtained UTS models

Fuzzy model	Number of fuzzy sets for every input and output	Rule length of every fuzzy rule	RMSE of reduced training data	RMSE of training data	RMSE of testing data
Initial model with 15 rules	Inputs: [15; 15; 15; 15; 15; 15; 15; 15; 15; 15; 15; 15; 15; 15; 15] Output: 15	[15; 15; 15; 15; 15; 15; 15; 15; 15; 15; 15; 15; 15; 15; 15]	124.04	117.32	119.58
Optimised model with 13 rules	Inputs: [13; 11; 10; 11; 10; 12; 12; 13; 12; 11; 9; 11; 11; 10; 13] Output: 13	[15; 13; 11; 12; 11; 11; 15; 15; 13; 14; 13; 14]	48.27	42.59	46.26
Optimised model with 10 rules	Inputs: [10; 8; 7; 7; 8; 9; 9; 10; 10; 9; 7; 9; 9; 7; 10] Output: 10	[15; 12; 11; 12; 11; 15; 12; 14; 13; 14]	54.76	44.65	46.03



(a)

**Chapter 6: Application of FM-HCMO in Modelling Problems**

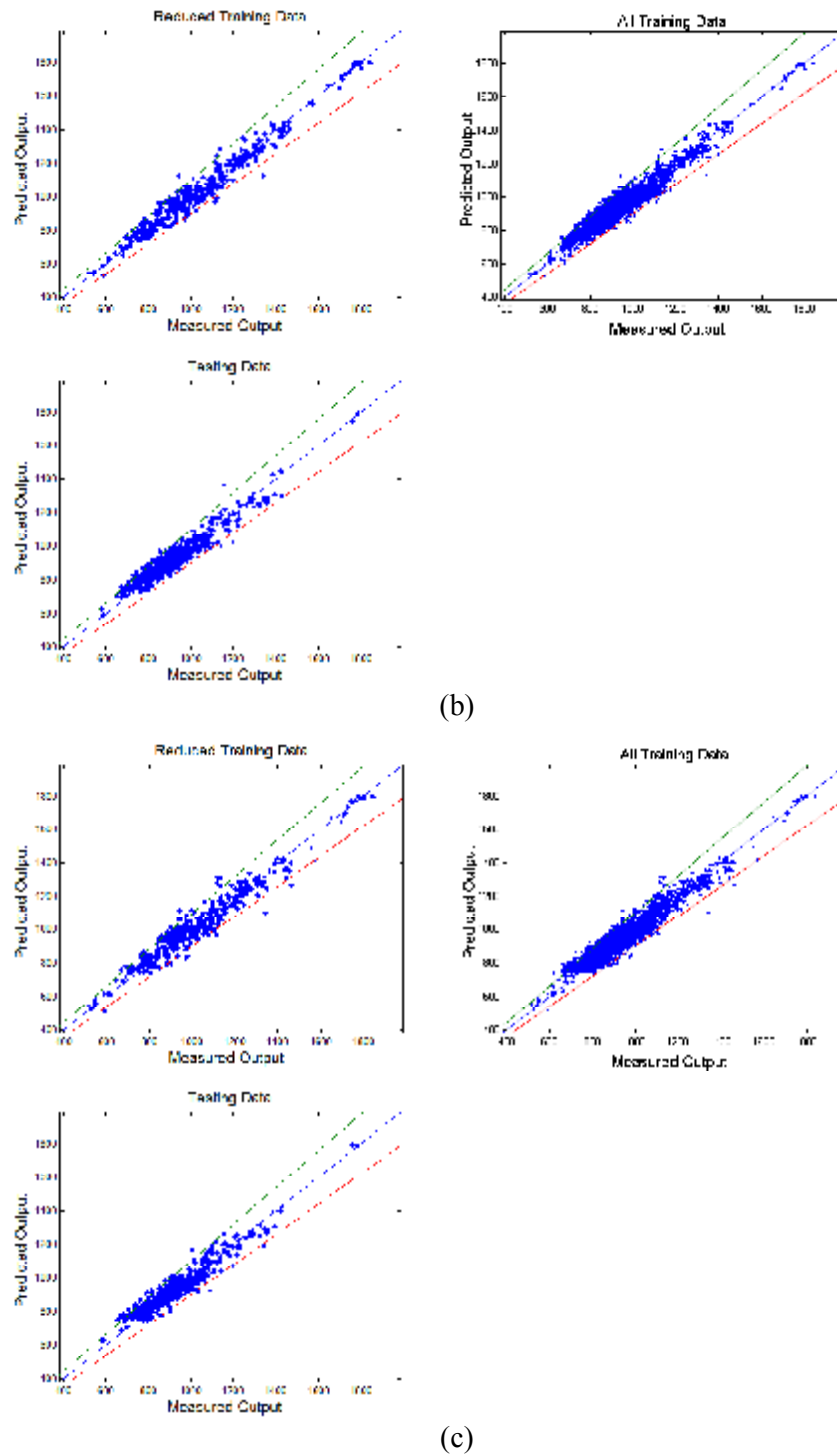


Figure 6-16. The UTS models' predicted outputs versus measured outputs: (a) the initial model, (b) an optimised model with 13 rules, and (c) an optimised model with 10 rules; the green and red lines represent the +10% and -10% error bands respectively.



**Chapter 6: Application of FM-HCMO in Modelling Problems**

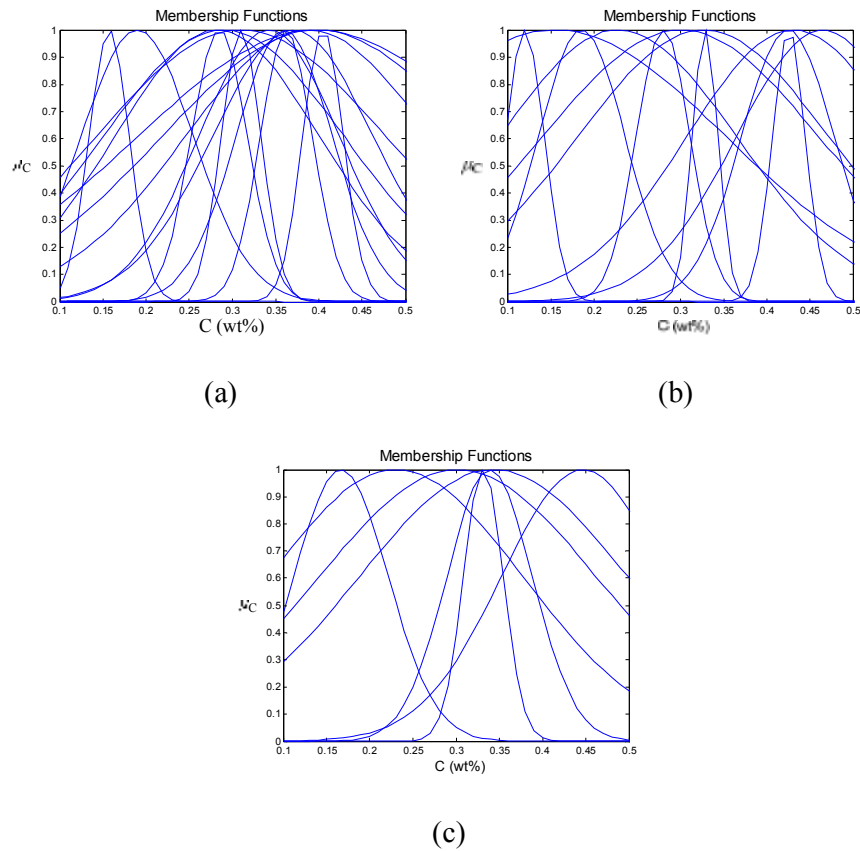


Figure 6-17. The UTS models' membership functions: (a) the initial model, (b) an optimised model with 13 rules, and (c) an optimised model with 10 rules

To provide more details about these Pareto-optimal UTS models, Figure 6-18 shows two rules (the 4<sup>th</sup> rule and the 7<sup>th</sup> rule) out of the rule base of the optimised 10-rule model. For these fuzzy rules, they can be rewritten as the following linguistic rules using the linguistic hedges approach:

R<sub>4</sub>: IF Test Depth is *slightly small* AND Size is *medium* AND C is *small*  
 AND Mn is *medium* AND S is *more or less small* AND Cr is  
*slightly small* AND Mo is *more or less small* AND Ni is *very small*

## **Chapter 6: Application of FM-HCMO in Modelling Problems**

AND Al is *very small* AND V is *small* AND Hardening Temperature is *medium* AND Tempering Temperature is *large*, THEN UTS is *very small*.

R<sub>7</sub>: IF Test Depth is *slightly small* AND Site Number is *quite large* AND C is *medium* AND Si is *medium* AND Mn is *more or less small* AND Cr is *more or less large* AND Mo is *slightly small* AND Ni is *medium* AND V is *medium* AND Hardening Temperature is *more or less slightly small* AND Cooling Medium Number is *large* AND Tempering Temperatures is *large*, THEN UTS is *quite large*.

To verify the physical interpretation of the obtained models, Figure 6-19 shows the three-dimensional response surfaces of the 10-rule UTS model. These surfaces are achieved by plotting two varying input variables against the output while keeping other input variables constant. The constant variables are set to the average values of the dominant steel grade, which is the 1%CrMo steel grade [Tenner 1999]. These plots in Figure 6-19 are consistent with those variable effect plots in [Tenner 1999], which have been verified to follow the expected behaviour as predicted by theory or by expert knowledge. This 10-rule model's 5%-range confidence band is shown in Figure 6-20.

**Chapter 6: Application of FM-HCMO in Modelling Problems**

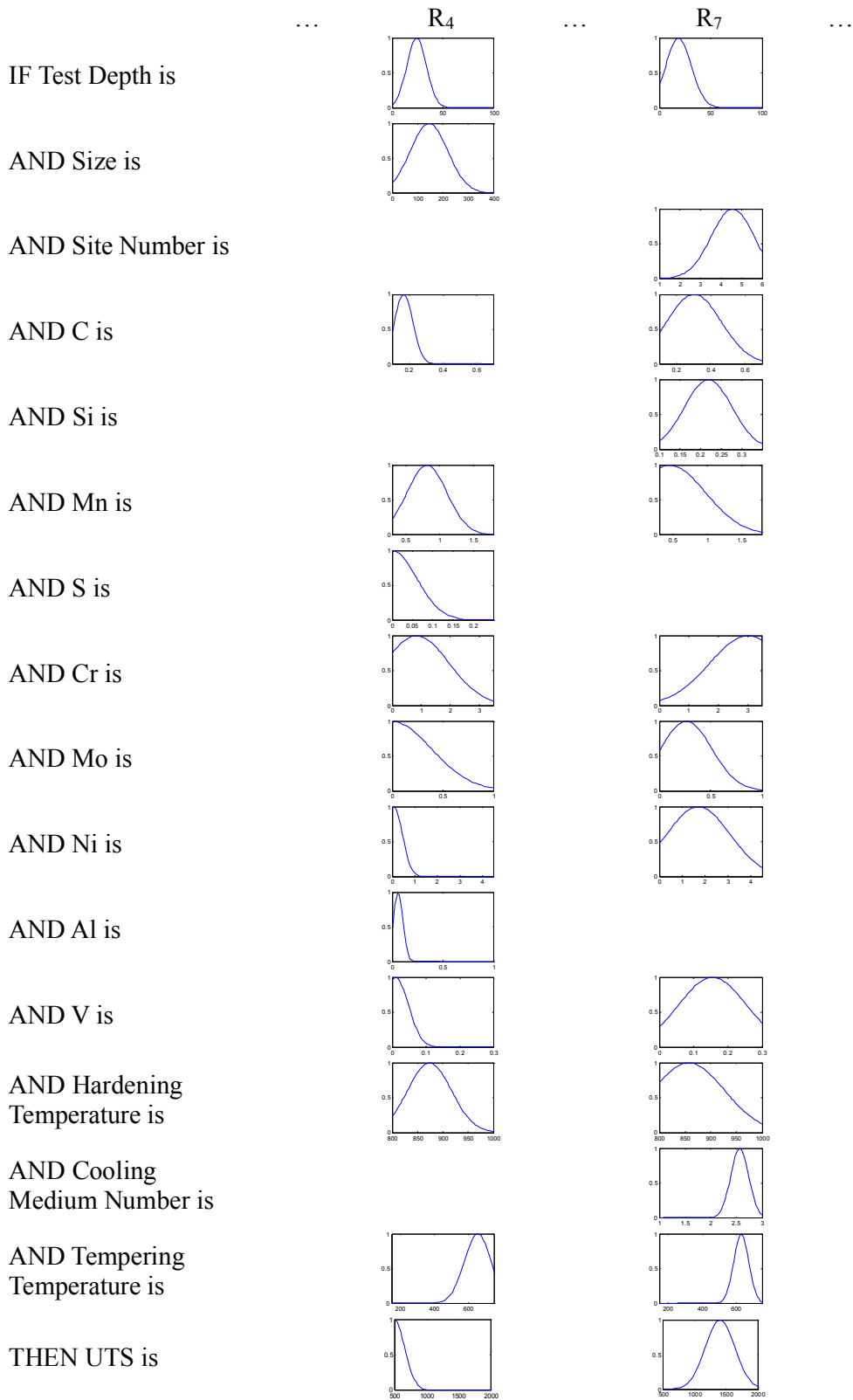


Figure 6-18. Rules of the optimised 10-rule UTS model

**Chapter 6: Application of FM-HCMO in Modelling Problems**

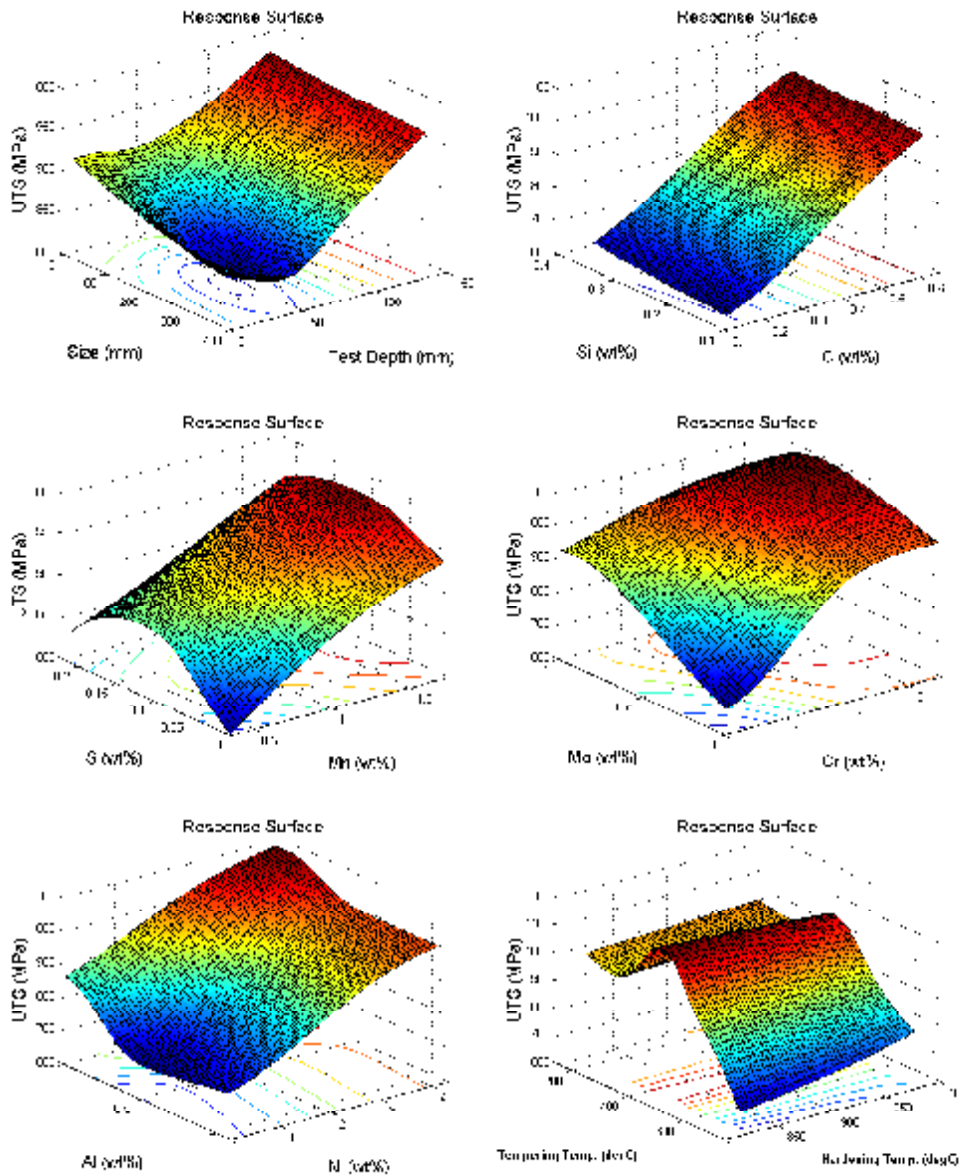


Figure 6-19. Response surfaces of the optimised 10-rule UTS model

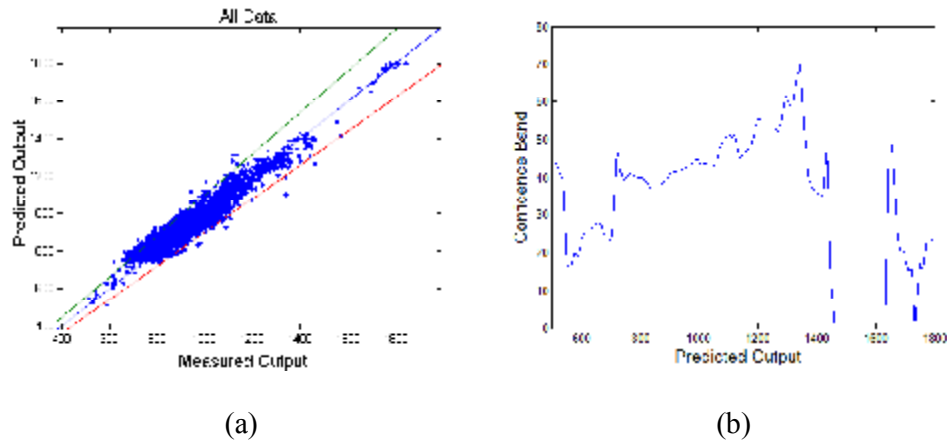


Figure 6-20. (a) The prediction performance and (b) the 5%-range confidence band of the optimised 10-rule UTS model

#### **6.4.2 Reduction of Area (ROA)**

In this experiment, 3710 ROA data were used, 75% of the data were used for training, 10% of the data were used for validation and the remaining 15% were used for testing. The ROA data include 15 inputs and one output, which are the same as those already reviewed for the UTS data (see Section 6.4.1).

In this case, the initial fuzzy model was obtained using 20 clusters; the maximum number of function evaluation for nPSO and nMPSO were both set to be 50,000. After the operation of the data selection mechanism (see Section 5.4.2), only 630 data points out of 2783 data points (all the training data) were selected as the representative training data. After the optimisation scheme, 12 Pareto-optimal models were obtained. Figure 6-21 shows the models' performance using various indices relating to the objective functions, the RMSE, the number of fuzzy sets, the

## Chapter 6: Application of FM-HCMO in Modelling Problems

number of fuzzy rules, and the length of the fuzzy rules.

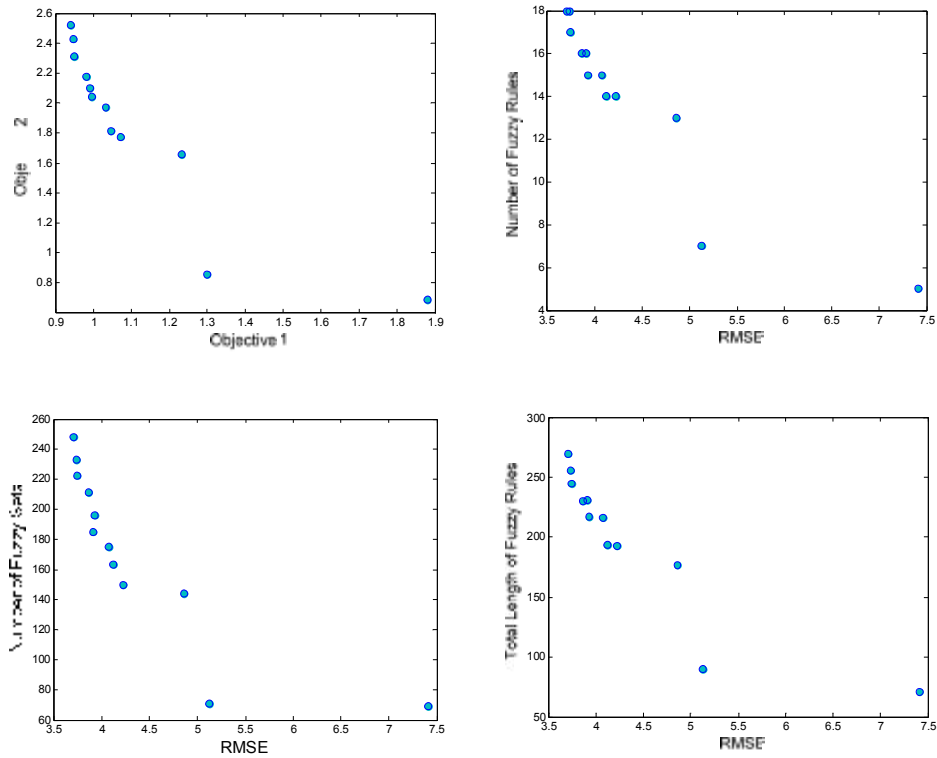


Figure 6-21. The performance of the Pareto-optimal ROA models

Table 6-4 shows the main parameters of the initial ROA model and two optimised ROA models, which are selected from all the Pareto-optimal models and include 15 and 7 rules respectively. Figure 6-22 shows the prediction performance of these models. Figure 6-23 illustrates the distribution of the membership functions relating to Input 4 (weight percentage for Carbon).



## Chapter 6: Application of FM-HCMO in Modelling Problems

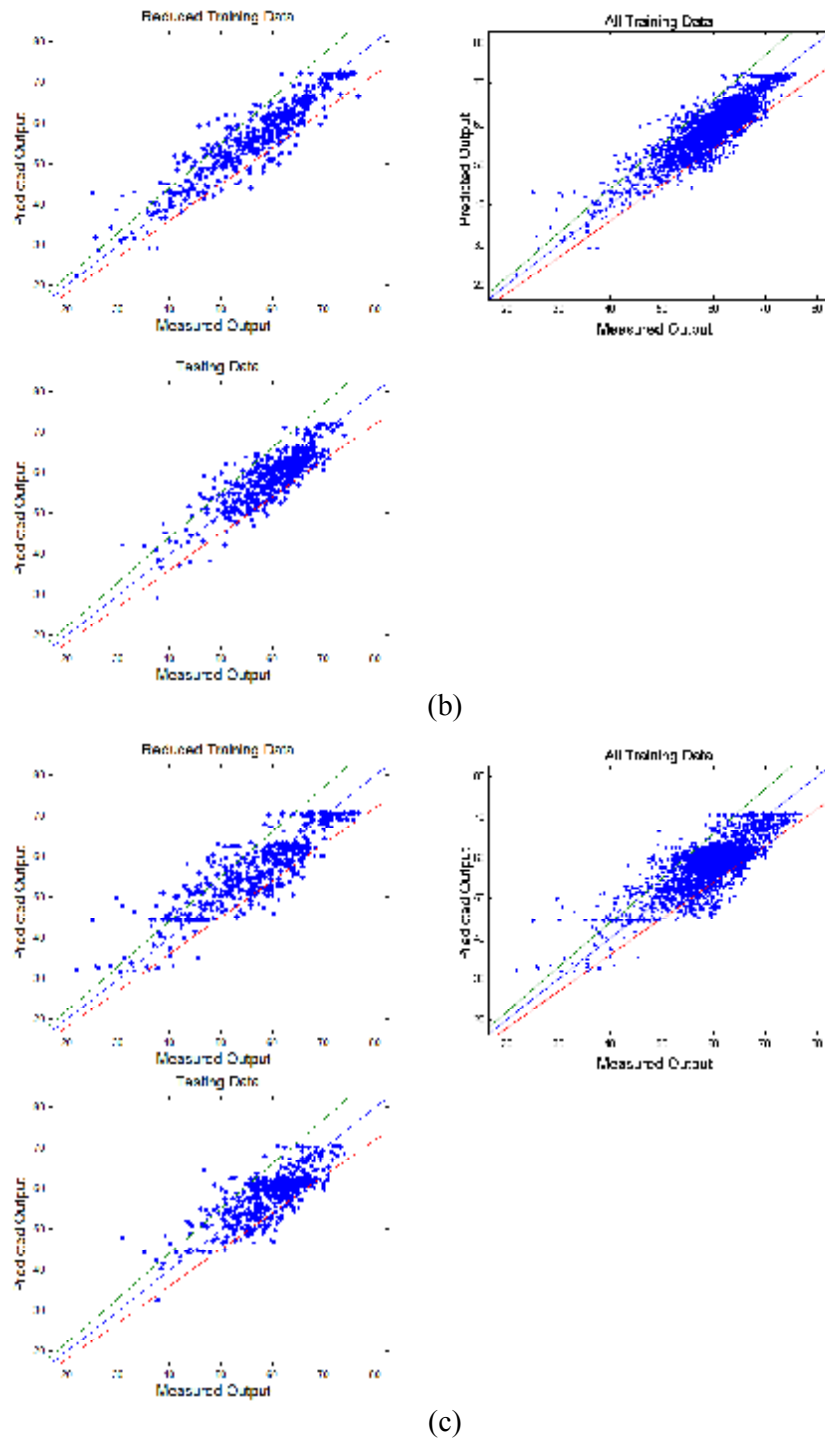


Figure 6-22. The ROA models' predicted outputs versus measured outputs: (a) the initial model, (b) an optimised model with 15 rules, and (c) an optimised model with 7 rules; the green and red lines represent the +10% and -10% error bands respectively.



**Chapter 6: Application of FM-HCMO in Modelling Problems**

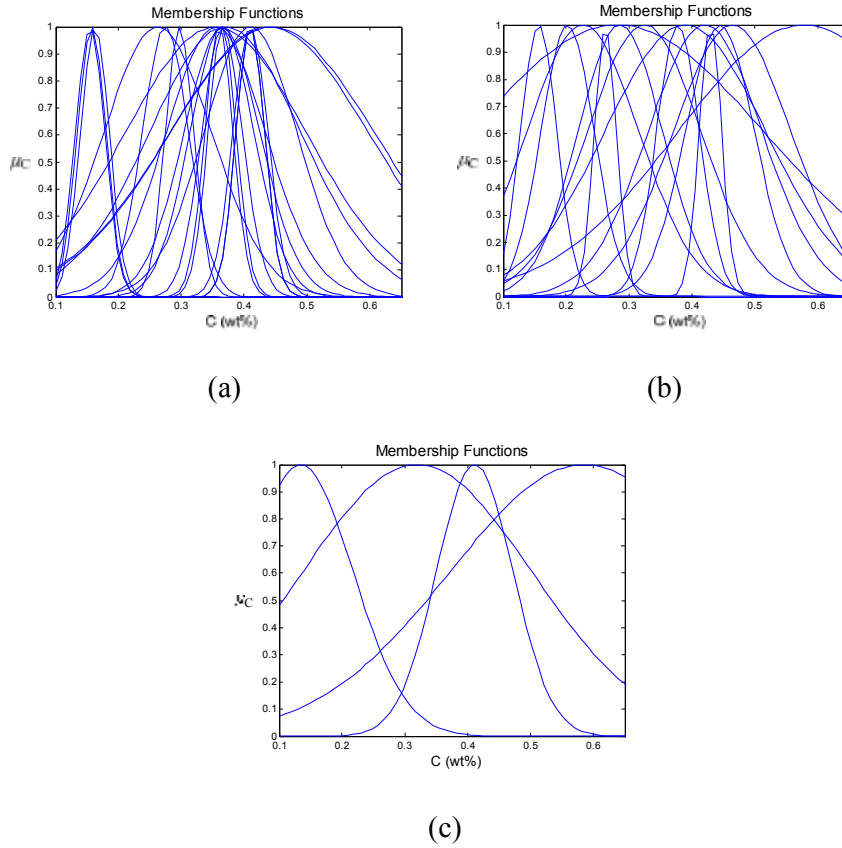


Figure 6-23. The ROA models' membership functions: (a) the initial fuzzy model, (b) an optimised fuzzy model with 15 rules, and (c) an optimised fuzzy model with 7 rules

Figure 6-24 displays the three-dimensional response surfaces of the 15-rule ROA model by setting two of the input variables to be varying and other input variables to be the constant average values of the dominant steel grade, which is the 1%CrMo steel grade [Tenner 1999]. These surfaces are consistent with the variable effect plots in [Tenner 1999], and this means that the models built by FM-HCMO follow the theoretical and experts' expectation in this example. For this 15-rule model, its 5%-range confidence band is shown in Figure 6-25.

**Chapter 6: Application of FM-HCMO in Modelling Problems**

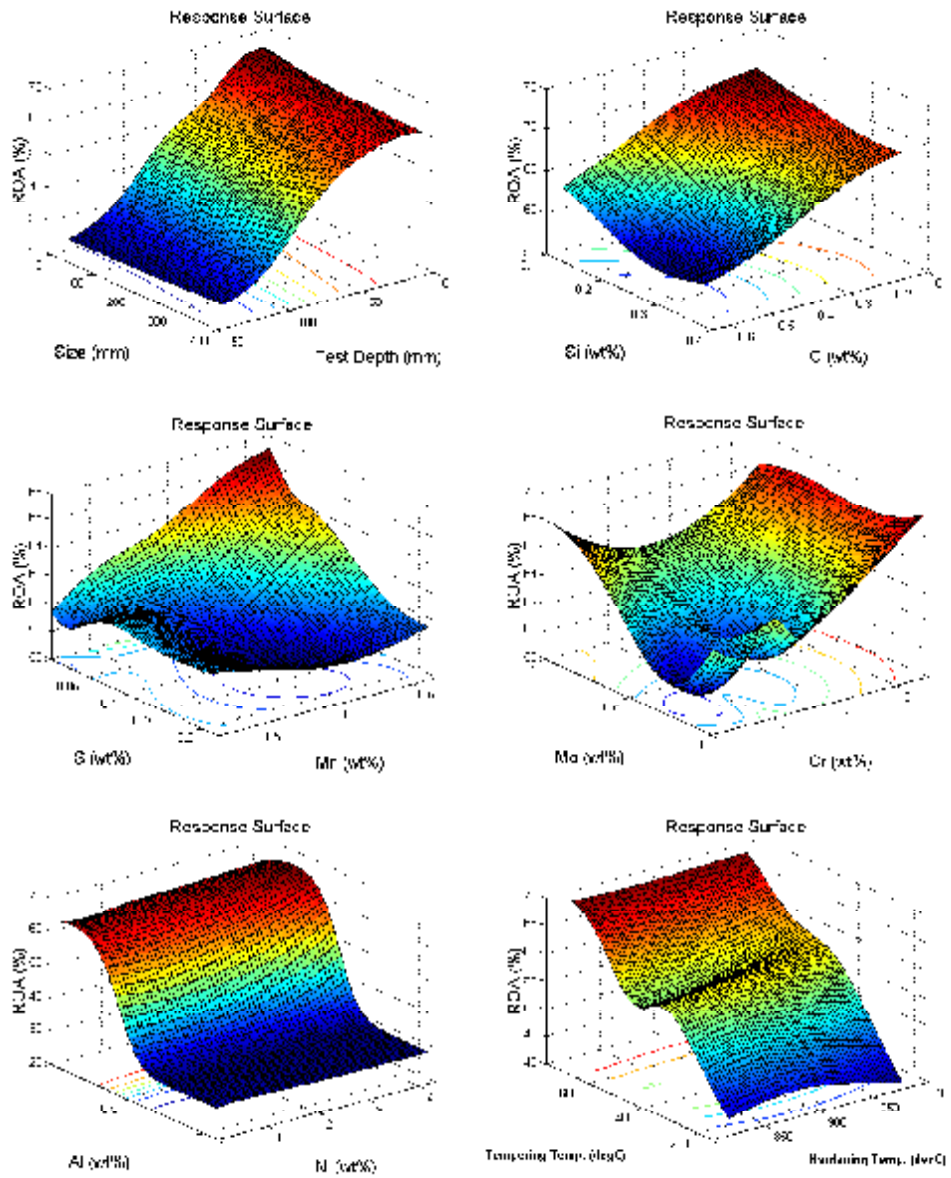


Figure 6-24. Response surfaces of the optimised 15-rule ROA model

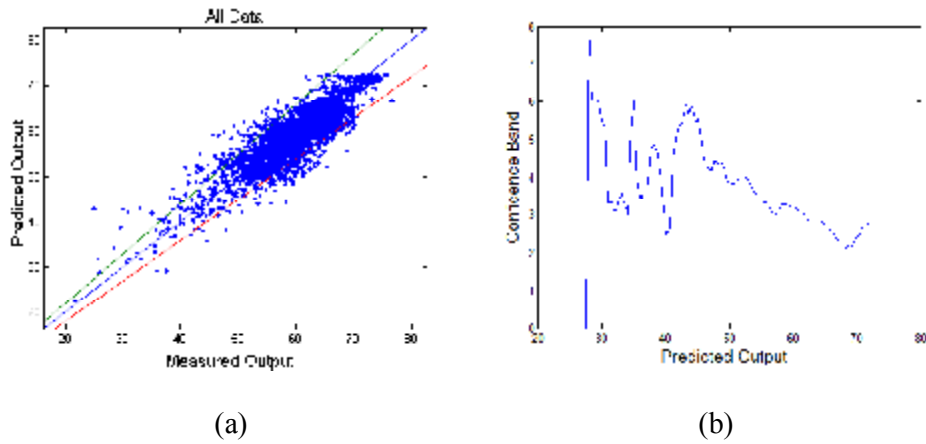


Figure 6-25. (a) The prediction performance and (b) the 5%-range confidence band of the optimised 15-rule ROA model

### 6.4.3 Elongation

3804 Elongation data were used in this case, 75% of the data were used for training, 10% of the data were used for validation and the remaining 15% were used for final testing. These data include 16 inputs, which consist of the weight percentages for the chemical composites, for instance Carbon (C), Silica (Si), Manganese (Mn), Chromium (Cr), Molybdenum (Mo), etc., the test depth, size and site of the alloy steel, the cooling medium, as well as the hardening and tempering temperatures.

In this case, the initial fuzzy model was obtained with the setting of 15 clusters; the maximum number of function evaluation for nPSO and nMPSO were both set to be 50,000. After data selection, 500 representative data points out of 2853 data points were selected and then used in the following training process. Finally, a set

## Chapter 6: Application of FM-HCMO in Modelling Problems

of Pareto-optimal Elongation models were constructed. Figure 6-26 shows the trade-offs among the multiple criteria within these non-dominated fuzzy models.

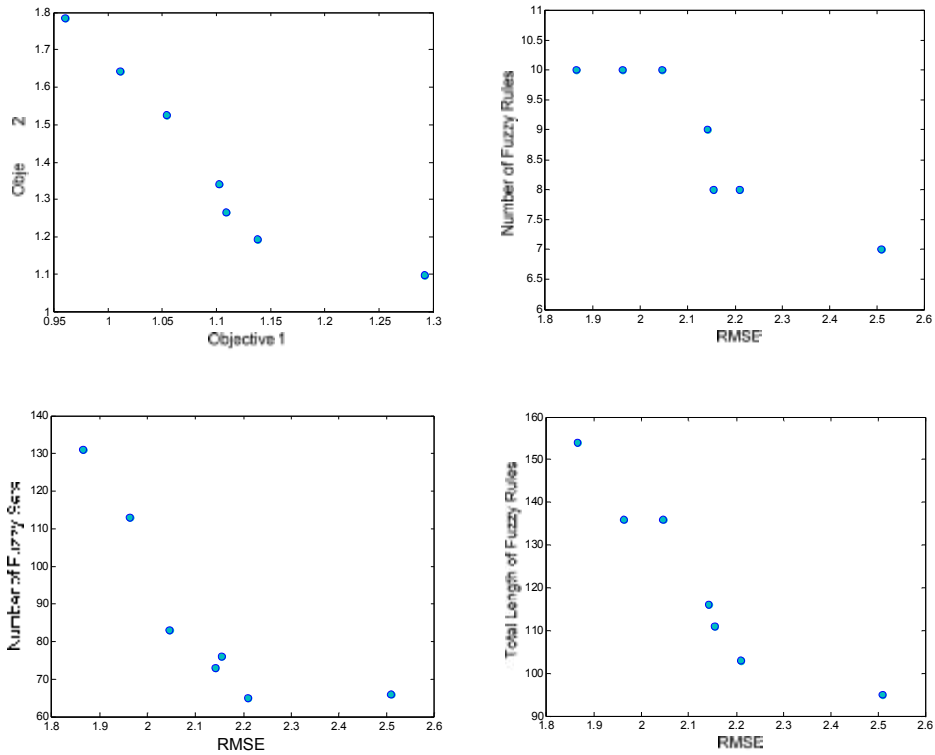


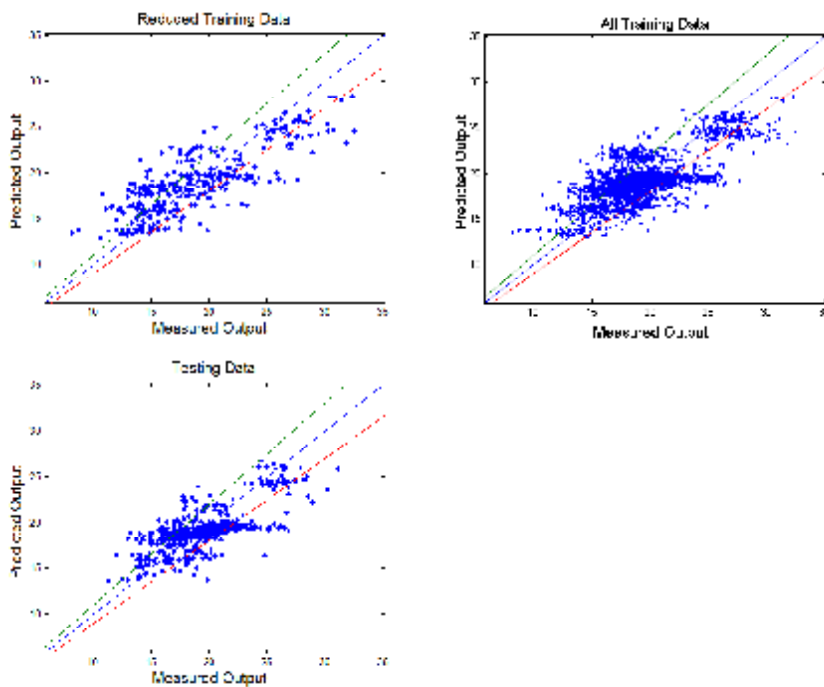
Figure 6-26. The performance of the Pareto-optimal Elongation models

Table 6-5 describes the main parameters of the initial Elongation model and two optimised Elongation models with 10 and 8 rules respectively. Figure 6-27 shows the prediction performance of these models and Figure 6-28 illustrates the distribution of the membership functions for Input 16 (tempering temperature), which is an instance of all other membership functions.

**Chapter 6: Application of FM-HCMO in Modelling Problems**

Table 6-5. Main parameters of the obtained Elongation models

Fuzzy model	Number of fuzzy sets for every input and output	Rule length of every fuzzy rule	RMSE of reduced training data	RMSE of training data	RMSE of testing data
Initial Model (15 rules)	Inputs: [15; 15; 15; 15; 15; 15; 15; 15; 15; 15; 15; 15; 15; 15; 15] Output: 15	[16; 16; 16; 16; 16; 16; 16; 16; 16; 16; 16; 16; 16; 16; 16]	2.90	2.39	2.23
Optimised model with 10 rules	Inputs: [8; 6; 9; 7; 8; 9; 9; 3; 9; 9; 7; 6; 5; 9; 9; 9] Output: 9	[16; 15; 15; 15; 15; 14; 16; 13; 13; 12]	1.87	1.78	1.76
Optimised model with 8 rules	Inputs: [5; 4; 5; 2; 5; 5; 6; 3; 4; 4; 5; 2; 4; 5; 5; 5] Output: 7	[13; 14; 13; 12; 16; 12; 13; 10]	2.15	1.78	1.65



(a)

## Chapter 6: Application of FM-HCMO in Modelling Problems

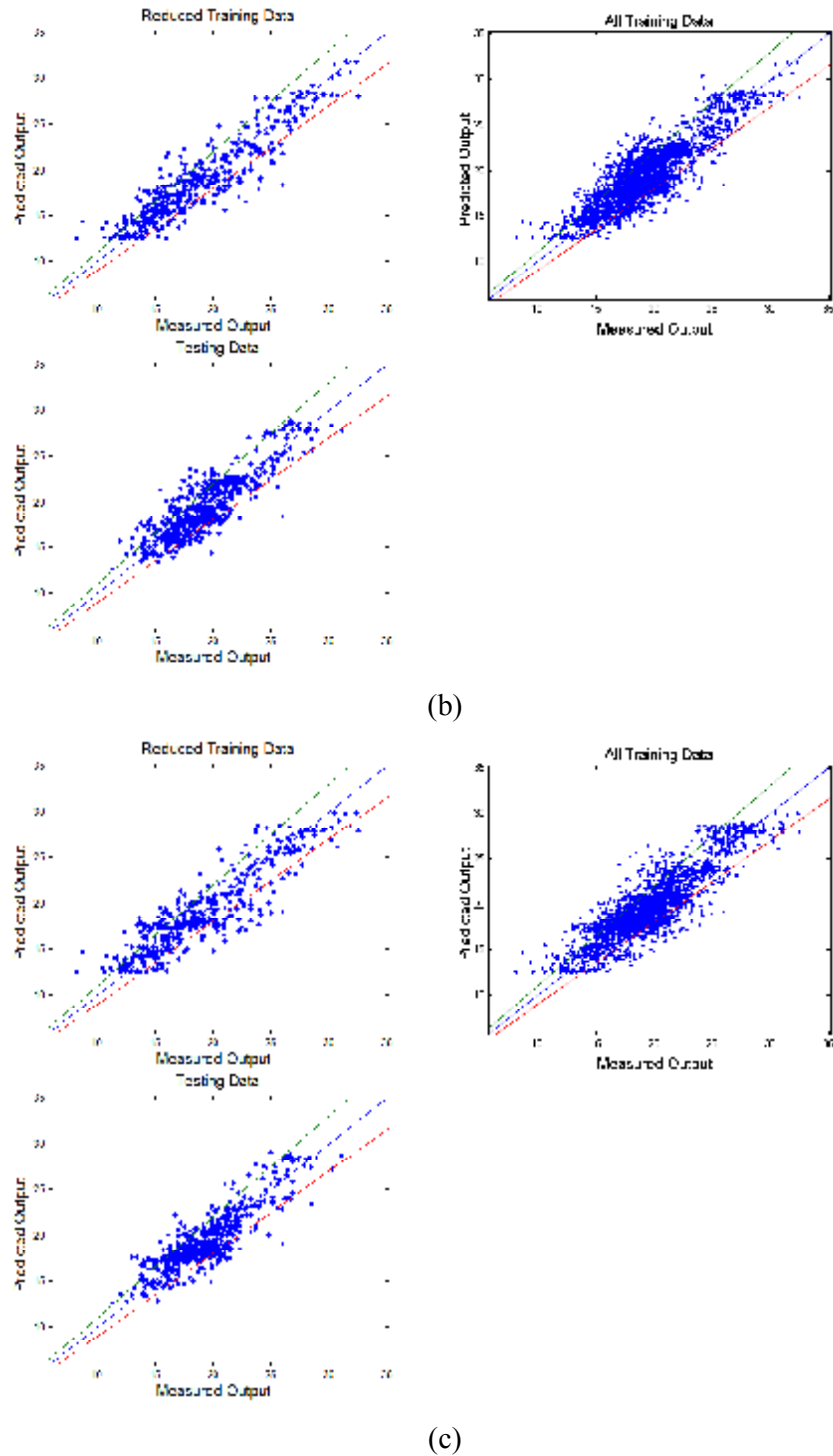


Figure 6-27. The Elongation models' predicted outputs versus measured outputs: (a) the initial model, (b) an optimised model with 10 rules, and (c) an optimised model with 8 rules; the green and red lines represent the +10% and -10% error bands respectively.

## Chapter 6: Application of FM-HCMO in Modelling Problems

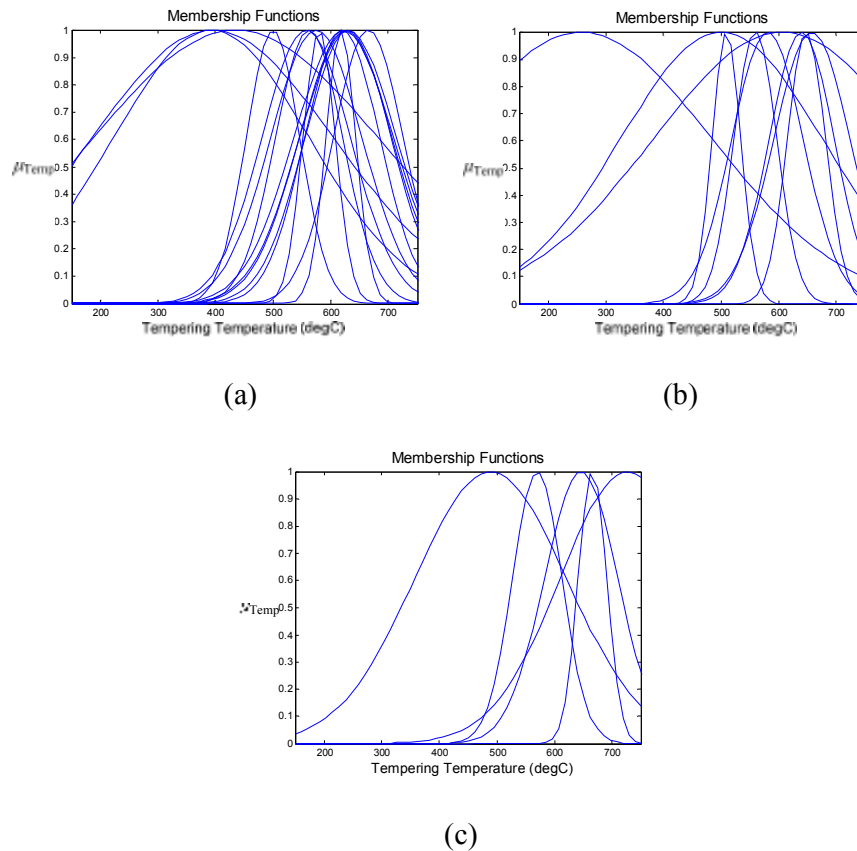


Figure 6-28. The Elongation models' membership functions: (a) the initial fuzzy model, (b) an optimised fuzzy model with 10 rules, and (c) an optimised fuzzy model with 8 rules

The response surfaces of the 10-rule Elongation model are shown in Figure 6-29. The constant variables are set to be the average values of the 1%CrMo steel grade. These surfaces reveal a consistent match with the variable effect plots in [Tenner 1999], and this means the constructed models follow the expected behaviour as predicted by theory or by expert knowledge. The 5%-range confidence band of this 10-rule model is displayed in Figure 6-30.

**Chapter 6: Application of FM-HCMO in Modelling Problems**

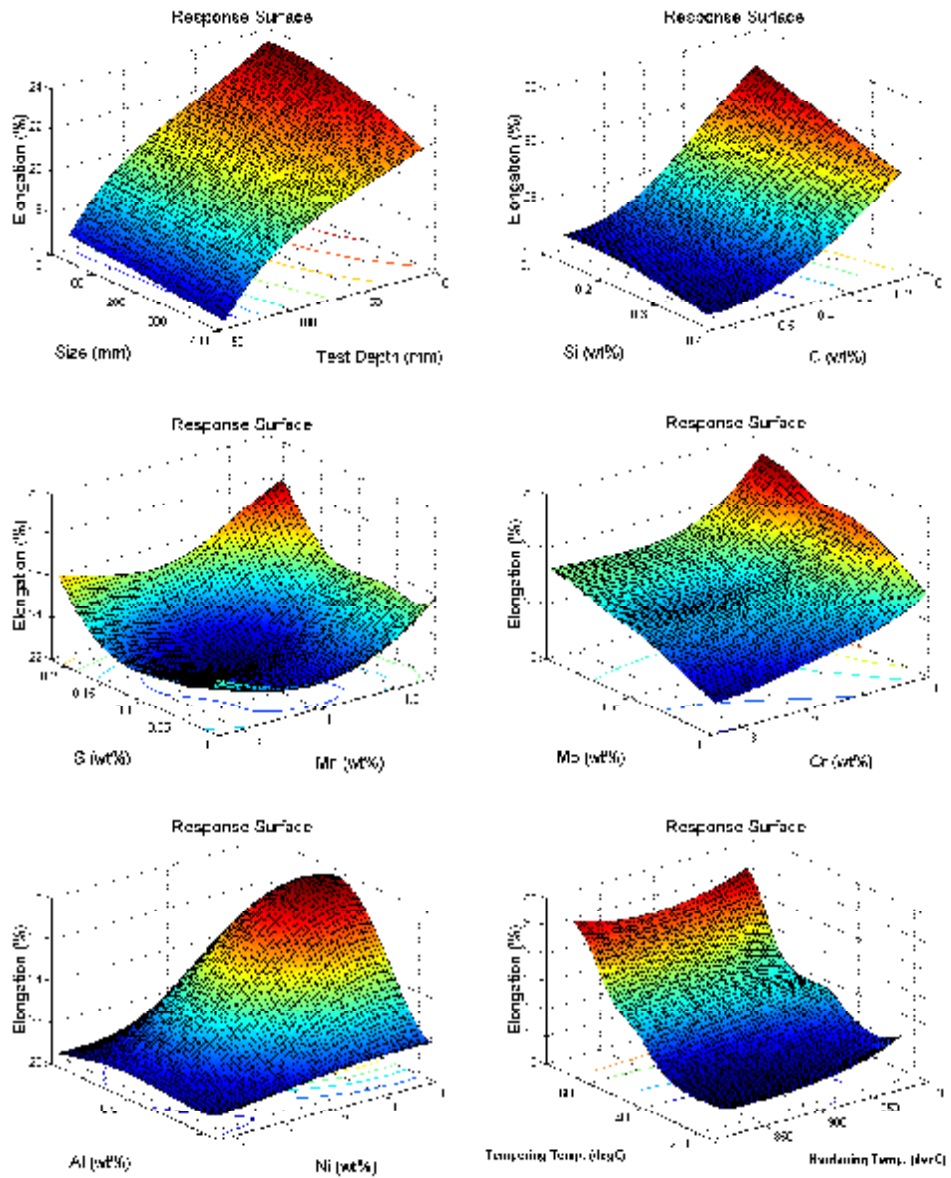


Figure 6-29. Response surfaces of the optimised 10-rule Elongation model



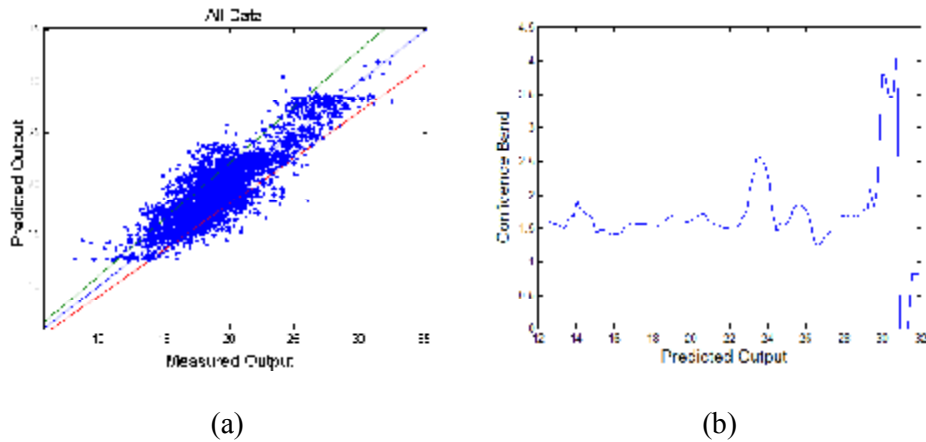


Figure 6-30. (a) The prediction performance and (b) the 5%-range confidence band of the optimised 10-rule Elongation model

#### 6.4.4 Impact Energy

In this example, 1661 Impact Energy data were employed, 75% of the data were used for training, 10% of the data were used for validation and the remaining 15% were used for final testing. 16 inputs and one output are included in these data. The inputs consist of the weight percentages for the chemical composites, namely Carbon (C), Silica (Si), Manganese (Mn), Sulphur (S), Chromium (Cr), Molybdenum (Mo), Nickel (Ni), Aluminium (Al) and Vanadium (V), the test depth, size and site of the alloy steel, the cooling medium, as well as the hardening, tempering and impact test temperatures.

In this experiment, the initial fuzzy model was obtained with 15 clusters; the maximal function evaluation numbers for nPSO and nMPSO were both set to be 50,000. Following a data selection exercise, 455 data points out of 1246 data

## Chapter 6: Application of FM-HCMO in Modelling Problems

points were selected for the following model training. After the optimisation, some non-dominated solutions were obtained. Figure 6-31 shows the trade-offs among the multiple objectives and criteria within these non-dominated fuzzy system solutions.

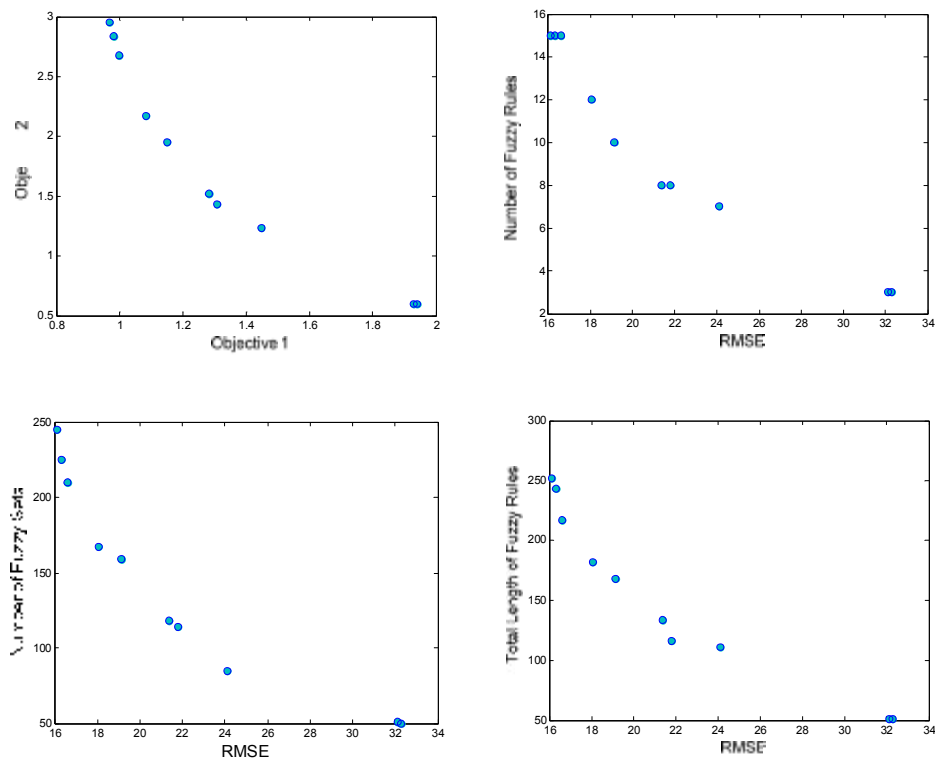


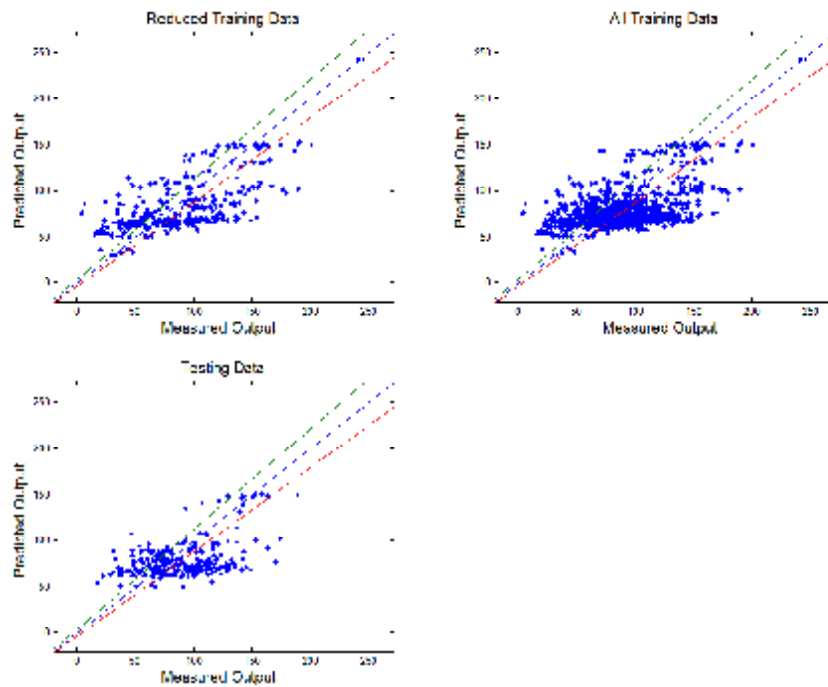
Figure 6-31. The performance of the Pareto-optimal Impact Energy models

Table 6-6 shows the main parameters of the initial Impact Energy model and two optimised Impact Energy models with 15 and 8 rules respectively. Figure 6-32 shows the prediction performance of these models and Figure 6-33 shows the membership functions of Input 5 (weight percentage for Silica).

**Chapter 6: Application of FM-HCMO in Modelling Problems**

Table 6-6. Main parameters of the obtained Impact Energy models

Fuzzy model	Number of fuzzy sets for every input and output	Rule length of every fuzzy rule	RMSE of reduced training data	RMSE of training data	RMSE of testing data
Initial Model (15 rules)	Inputs: [15; 15; 15; 15; 15; 15; 15; 15; 15; 15; 15; 15; 15; 15; 15] Output: 15	[16; 16; 16; 16; 16; 16; 16; 16; 16; 16; 16; 16; 16; 16; 16]	31.56	30.54	31.44
Optimised model with 15 rules	Inputs: [12; 15; 14; 14; 13; 15; 13; 12; 14; 13; 12; 13; 15; 13; 11; 15] Output: 11	[16; 16; 15; 16; 16; 16; 13; 14; 14; 16; 16; 15; 16; 13; 16]	16.32	14.35	17.10
Optimised model with 8 rules	Inputs: [8; 8; 8; 7; 6; 7; 7; 8; 7; 7; 7; 5; 7; 7; 4; 7] Output: 8	[16; 16; 16; 15; 15; 16; 16; 16]	21.36	17.85	19.03



## Chapter 6: Application of FM-HCMO in Modelling Problems

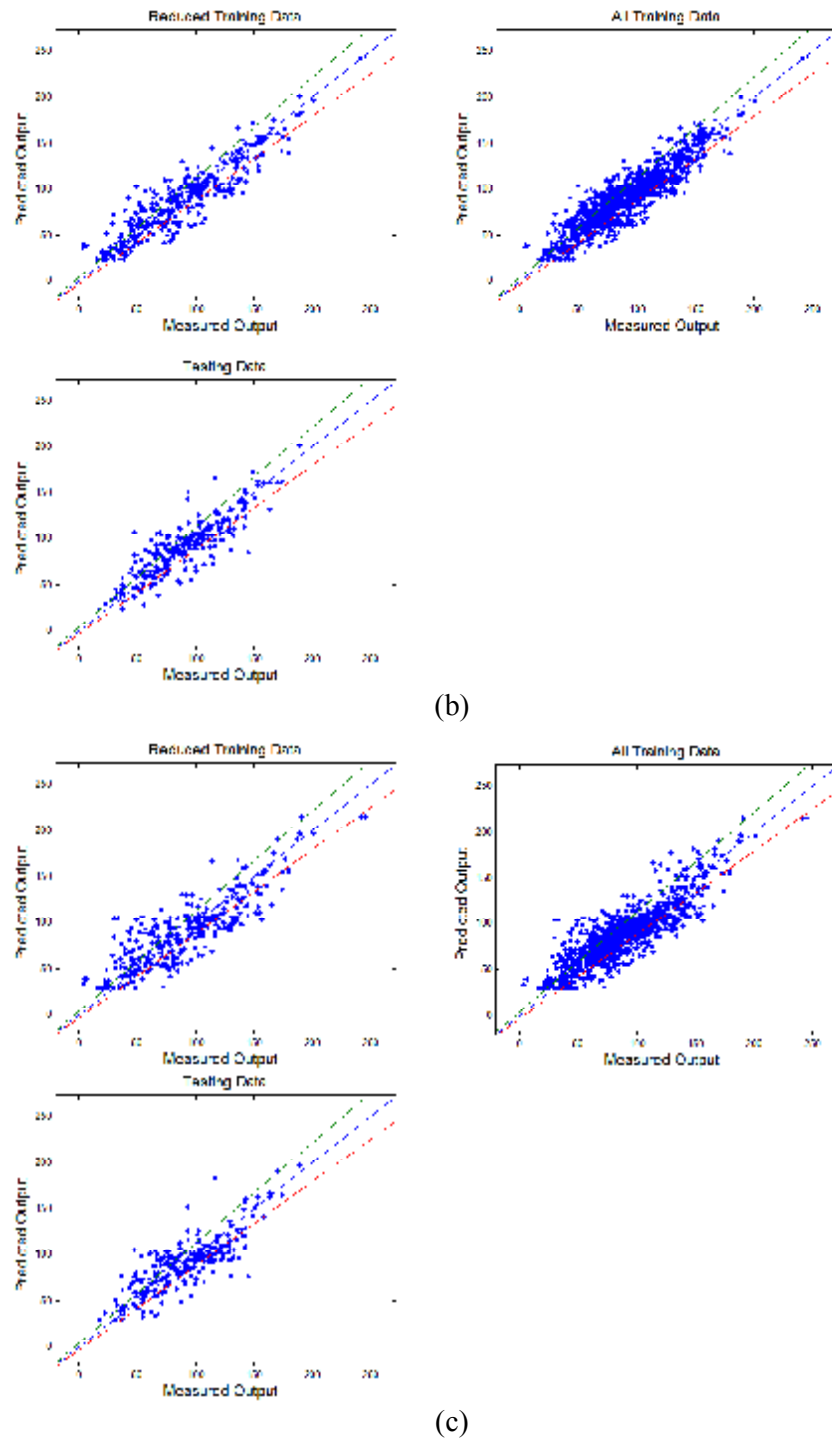


Figure 6-32. The Impact Energy models' predicted outputs versus measured outputs: (a) the initial model, (b) an optimised model with 15 rules, and (c) an optimised model with 8 rules; the green and red lines represent the +10% and -10% error bands respectively.

## Chapter 6: Application of FM-HCMO in Modelling Problems

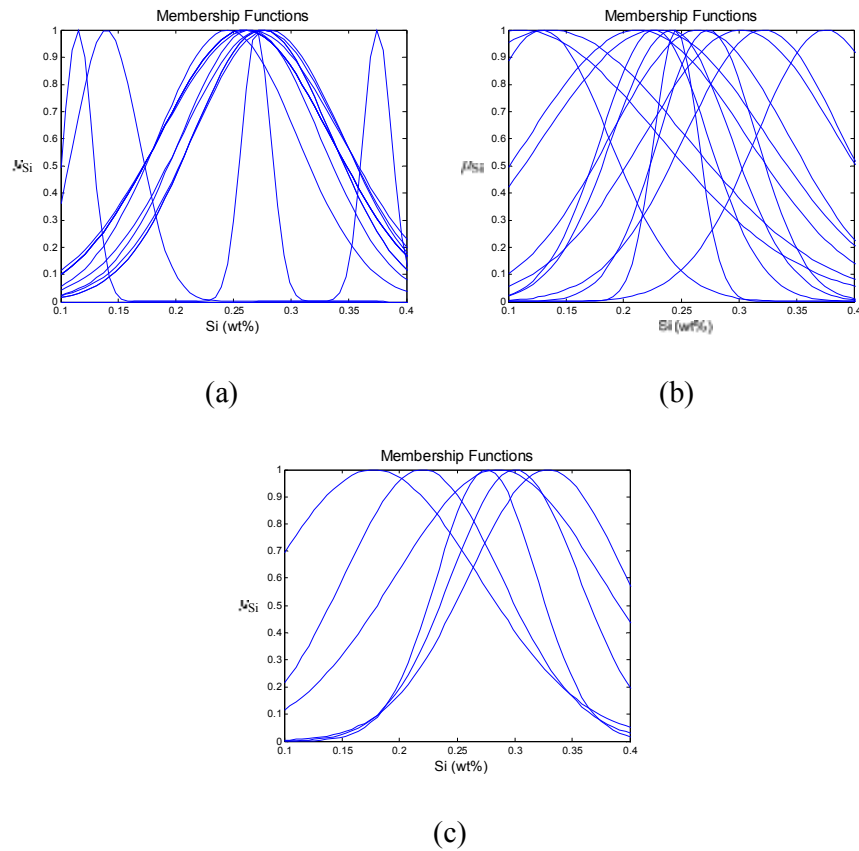


Figure 6-33. The Impact Energy models' membership functions: (a) the initial fuzzy model, (b) an optimised fuzzy model with 15 rules, and (c) an optimised fuzzy model with 8 rules

Figure 6-34 shows the three-dimensional response surfaces of the 15-rule Impact Energy model. The constant variables are set to the average values of the 1%CrMo steel grade. These figures reveal a consistent match with the variable effect plots in [Tenner 1999], which have been verified to follow the theoretical or expert knowledge. In Figure 6-35, the 5%-range confidence band of this 15-rule model is shown.

**Chapter 6: Application of FM-HCMO in Modelling Problems**

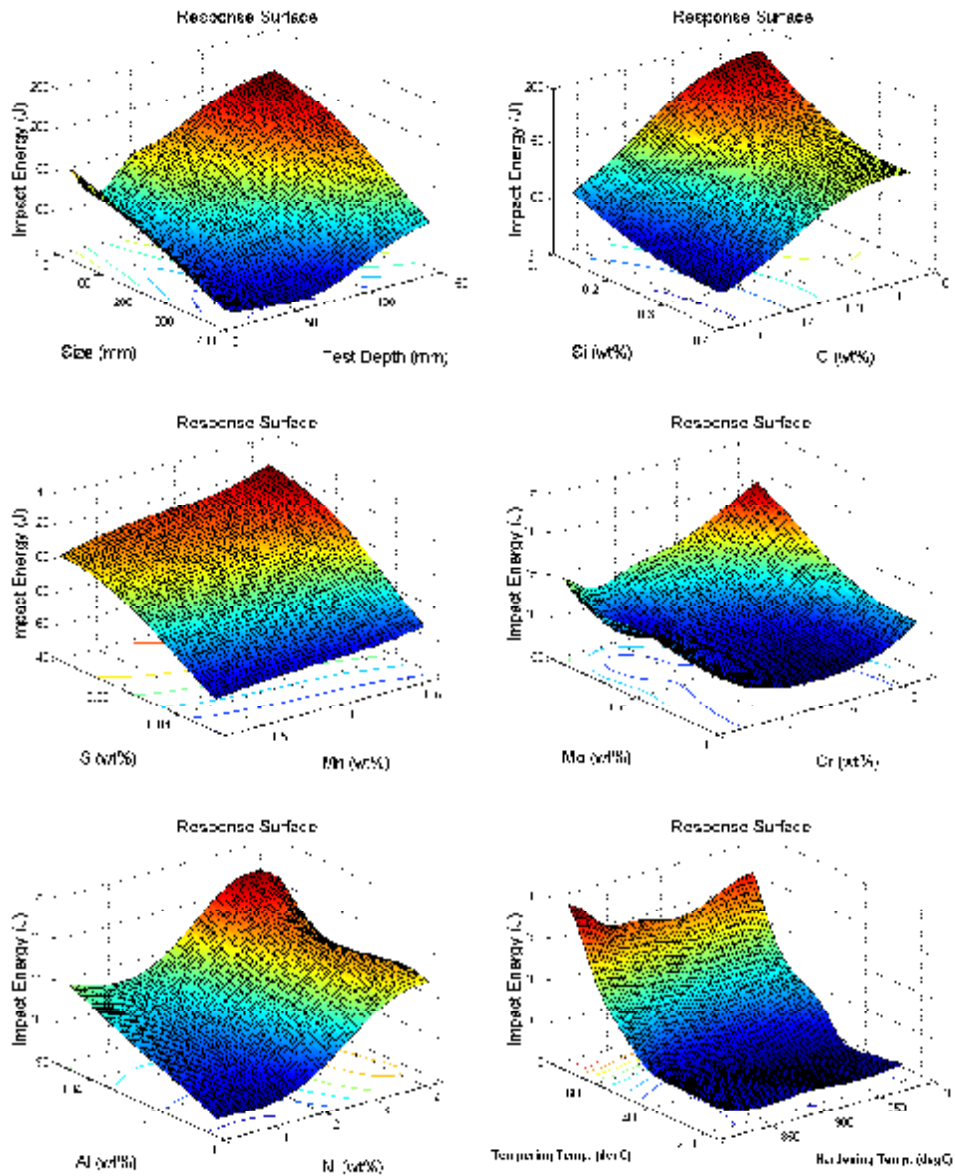


Figure 6-34. Response surfaces of the optimised 15-rule Impact Energy model

## Chapter 6: Application of FM-HCMO in Modelling Problems

---

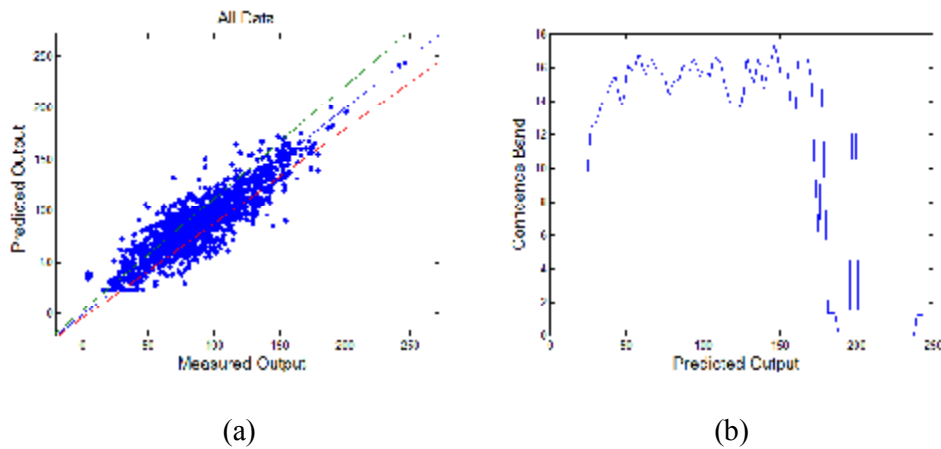


Figure 6-35. (a) The prediction performance and (b) the 5%-range confidence band of the optimised 15-rule Impact Energy model

### 6.5 Summary

The experimental validation carried-out in this chapter shows that the proposed approach FM-HCMO works effectively in eliciting accurate and interpretable models. Initially, FM-HCMO was applied to the modelling of two benchmark problems, a static nonlinear system approximation problem and a dynamical system identification problem. The experimental results revealed that, compared to other modelling methods, FM-HCMO is able to produce more compact and simpler models; the obtained Pareto-optimal models using FM-HCMO provide a wider choice of different solutions to users. Furthermore, the modelling of the mechanical properties for alloy steels demonstrated that this proposed approach works well within the context of a high-dimensional industrial application. The physical interpretation of the obtained models has been shown to be consistent with the expected behaviour as predicted by theory or by expert knowledge. In

## ***Chapter 6: Application of FM-HCMO in Modelling Problems***

---

addition to the above, it is worth noting that the linguistic models constructed using FM-HCMO has a good generalisation ability, which is evidenced by the smooth input/output response surfaces obtained using the elicited models. Thus, it can be concluded that FM-HCMO provides a simple and effective framework for system identification and prediction. The next chapter will highlight how such robust prediction models can be exploited in a reverse-engineering fashion to identify ‘optimal’ recipes for system design.



## **Chapter 7**

# **Application of RSSA and MO-RSSA for Alloy Steel Design Problems**

### **7.1 Introduction**

In the steel industry, determining the optimal heat treatment regime and the required weight percentages for the chemical composites to obtain the desired mechanical properties of the steel is always a challenging multi-objective optimisation problem. Usually, some objectives may conflict with each other, such as the ultimate tensile strength (UTS) and the ductility. The steel ductility can also be reflected by its Reduction of Area (ROA). In this chapter, details relating to the optimisation of UTS and ROA using the RSSA and MO-RSSA algorithms (see Chapter 4) are presented and discussed.

In Chapter 6, the intelligent models based on fuzzy systems had been developed to predict the mechanical test results for the steels covered by a wide range of

## Chapter 7: Application of RSSA and MO-RSSA

training data. These models can be used to facilitate the findings relating to the optimal heat treatment regime and the weight percentages for the chemical composites to obtain the desired mechanical properties. Figure 7-1 shows the prediction results of one UTS model and one ROA model, whose  $RMSE(s)$  are 34.59 and 2.86 respectively. In the following studies, all alloy design experiments are conducted based on these two developed fuzzy models.

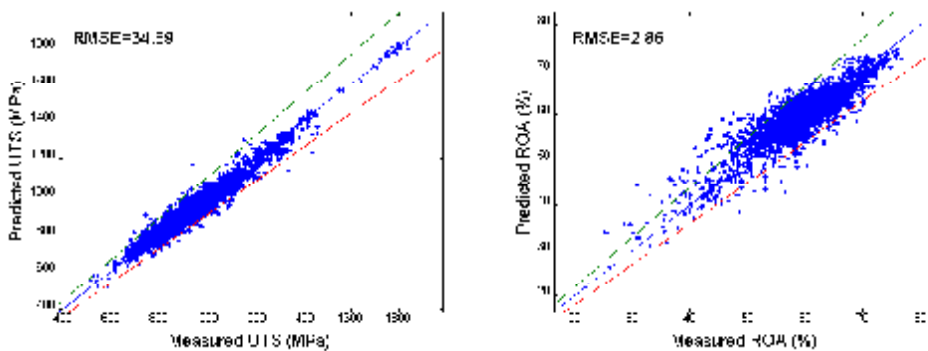


Figure 7-1. The prediction performance of the UTS model and the ROA models used in this chapter; the red and green lines delimit the +10% and -10% confidence bands respectively.

In this chapter, all parameter settings for RSSA and MO-RSSA (see Chapter 3) assumed the following parameters throughout: the decreasing parameter  $C_1 = 3$ , the increasing parameter  $C_2 = 1$ , the threshold value  $m = 15$ ; the variation operator works as a combination of the one-dimensional variation strategy (with the 75% probability of usage) and the multi-dimensional variation strategy (with the 25% probability of usage) (see Section 4.2.4); for MO-RSSA, the weight changing frequency parameter  $H = 10000$ .

## ***Chapter 7: Application of RSSA and MO-RSSA***

---

### **7.2 The Optimal Design of UTS for Alloy Steels**

In this case, the aim is to find the optimal solution for achieving a predefined target UTS value. The decision vector consists of weight percentages for the chemical composites, namely Carbon (C), Silica (Si), Manganese (Mn), Sulphur (S), Chromium (Cr), Molybdenum (Mo), Nickel (Ni), Aluminium (Al) and Vanadium (V), the test depth, the size and the site of the alloy steel, the cooling medium, as well as the hardening and tempering temperatures.

The objective function was designed to be as follows:

$$\text{Minimise } J_1 = \left( \frac{UTS - UTS_{Target}}{900} \right)^2 \quad (7.1)$$

where  $UTS_{Target}$  is the target UTS value.

In the first experiment, the  $UTS_{Target}$  was set to 900 MPa. Figure 7-2 shows the optimisation process and Table 7-1 provides the optimisation results relating to 10 different runs. The average function evaluation number used in the RSSA algorithm is only 36. From Table 7-1, it can be seen that the differences between the 10 solutions are somewhat stark, which means that there are many possible solutions satisfying the same defined objective.

In the second experiment, the  $UTS_{Target}$  was set to 1100 MPa. Figure 7-3 shows the fitness variation in the optimisation process and Table 7-2 shows the optimisation results in 10 different runs. For this experiment, the average function evaluation

**Chapter 7: Application of RSSA and MO-RSSA**

---

number used in the RSSA algorithm is 34. From Table 7-2, it can be seen that there are more additional feasible solutions to this optimisation problem and RSSA can find these very quickly.

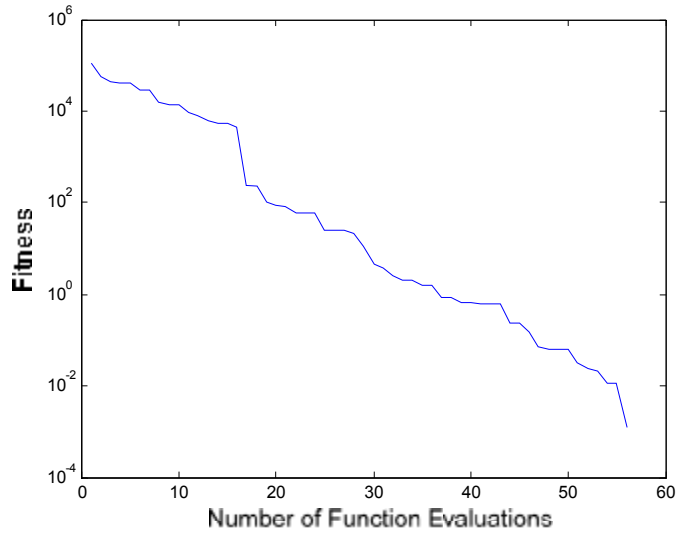


Figure 7-2. Average fitness of 10 runs versus function evaluation for the UTS design problem with  $UTS_{Target} = 900$  (MPa)

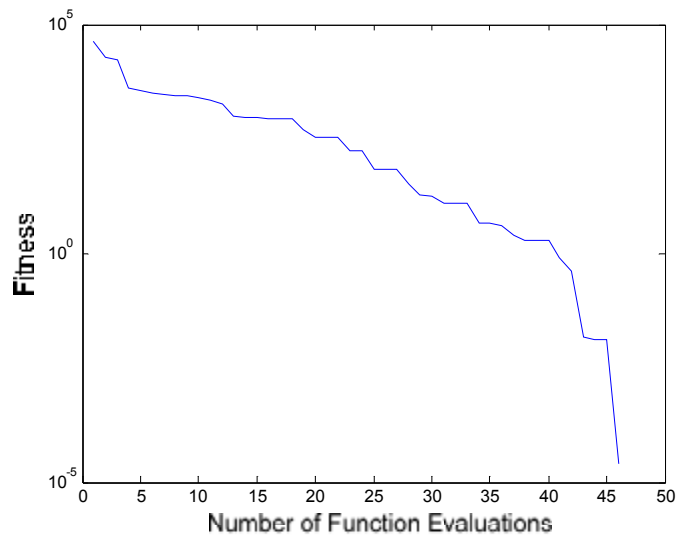


Figure 7-3. Average fitness of 10 runs versus function evaluation for the UTS design problem with  $UTS_{Target} = 1100$  (MPa)

## Chapter 7: Application of RSSA and MO-RSSA

Table 7-1. Optimisation solutions of 10 independent runs for the UTS design problem with  $UTS_{Target} = 900$  (MPa)

Solutions	1	2	3	4	5	6	7	8	9	10
Test Depth (mm)	61.8	67.8	111.4	41.9	129.6	58.1	78.6	18.8	93.9	74.2
Size (mm)	268.9	88.1	283.2	41.5	271.3	136.6	206.7	137.6	279.6	254.9
Site Number	2	5	5	2	5	3	6	3	3	4
C (wt%)	0.364	0.440	0.503	0.182	0.354	0.203	0.496	0.220	0.413	0.354
Si (wt%)	0.112	0.235	0.216	0.270	0.285	0.174	0.272	0.289	0.204	0.319
Mn (wt%)	1.554	1.189	0.939	0.954	1.397	0.644	0.521	0.488	0.742	0.940
S (wt%)	0.100	0.096	0.127	0.169	0.080	0.112	0.066	0.036	0.148	0.132
Cr (wt%)	0.263	0.589	3.025	0.613	2.733	0.615	0.790	0.650	0.140	0.489
Mo (wt%)	0.079	0.735	0.780	0.157	0.111	0.659	0.094	0.335	0.231	0.327
Ni (wt%)	0.609	2.069	0.241	0.379	3.765	0.312	2.967	2.557	2.003	1.023
Al (wt%)	0.641	0.028	0.029	0.842	0.190	0.093	0.086	0.253	0.260	0.495
V (wt%)	0.163	0.149	0.095	0.181	0.047	0.225	0.043	0.030	0.203	0.077
Hardening Temperature (°C)	970.3	971.9	908.6	979.4	860.2	907.6	889.1	975.2	933.1	923.8
Cooling Medium Number	2	3	2	3	2	1	2	1	1	2
Tempering Temperature (°C)	497.5	644.8	590.6	475.8	596.1	660.8	625.3	704.0	629.7	651.9
UTS (MPa)	900.1	899.9	899.9	899.9	900.0	899.6	899.9	899.8	899.9	900.0

## Chapter 7: Application of RSSA and MO-RSSA

Table 7-2. Optimisation solutions of 10 independent runs for the UTS design problem with  $UTS_{Target} = 1100$  (MPa)

Solutions	1	2	3	4	5	6	7	8	9	10
Test Depth (mm)	97.8	61.9	135.1	9.2	74.4	107.3	62.1	22.6	93.4	78.2
Size (mm)	15.9	44.7	106.2	226.7	251.2	166.2	269.8	235.5	29.6	87.8
Site Number	4	2	3	1	2	2	3	4	4	3
C (wt%)	0.194	0.306	0.551	0.618	0.186	0.409	0.236	0.161	0.228	0.341
Si (wt%)	0.194	0.161	0.128	0.149	0.234	0.349	0.330	0.272	0.250	0.203
Mn (wt%)	0.718	0.565	1.118	0.455	0.500	1.123	1.274	1.153	0.600	0.956
S (wt%)	0.049	0.144	0.143	0.184	0.056	0.144	0.205	0.095	0.142	0.177
Cr (wt%)	0.485	0.208	1.793	0.389	0.236	0.095	1.627	0.252	0.351	0.981
Mo (wt%)	0.318	0.044	0.648	0.082	0.622	0.114	0.087	0.442	0.337	0.381
Ni (wt%)	1.148	2.364	0.726	1.915	0.475	2.905	1.466	1.070	3.155	1.733
Al (wt%)	0.570	0.185	0.187	0.524	0.550	0.622	0.700	0.060	0.688	0.049
V (wt%)	0.105	0.212	0.229	0.140	0.130	0.104	0.098	0.267	0.036	0.240
Hardening Temperature (°C)	890.6	897.7	824.0	913.0	841.6	887.3	853.7	877.1	953.5	940.1
Cooling Medium Number	2	2	1	2	2	1	2	1	3	3
Tempering Temperature (°C)	362.3	353.8	597.9	677.8	280.9	360.6	637.1	523.9	587.8	710.7
UTS (MPa)	1100.0	1099.7	1100.1	1100.0	1099.9	1099.8	1099.8	1099.8	1100.0	1100.1

### 7.3 The Optimal Design of ROA

In this section, details relating to finding the optimal solution for achieving a predefined target ROA value are presented. In this case, the decision vector is the same as the one used for the UTS design problem in Section 7.2. The optimisation objective function was designed as follows:

$$\text{Minimise } J_2 \left( \frac{ROA - ROA_{Target}}{60} \right)^2 \quad (7.2)$$

where  $ROA_{Target}$  is the target ROA value.

In the first experiment, the  $ROA_{Target}$  was set to 60%. Table 7-3 provides the optimisation results for 10 different runs and Figure 7-4 shows the variation of the average fitness of these 10 runs during the optimisation process. The average function evaluation number used in the RSSA algorithm is only 28.

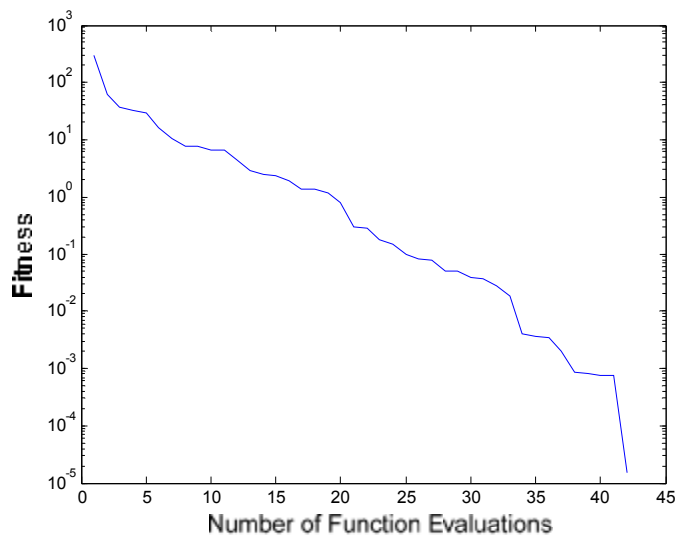


Figure 7-4. Average fitness of 10 runs versus function evaluation for the ROA design problem with  $ROA_{Target} = 60$  (%)

### Chapter 7: Application of RSSA and MO-RSSA

Table 7-3. Optimisation solutions of 10 independent runs for the ROA design problem with  $ROA_{Target} = 60$  (%)

Solutions	1	2	3	4	5	6	7	8	9	10
Test Depth (mm)	62.4	102.1	21.0	137.8	59.7	20.0	110.7	59.2	71.8	68.8
Size (mm)	74.4	351.6	251.9	71.5	101.0	310.7	250.6	94.6	303.6	276.9
Site Number	1	3	4	4	4	4	3	4	1	2
C (wt%)	0.434	0.249	0.246	0.248	0.517	0.191	0.243	0.204	0.240	0.239
Si (wt%)	0.297	0.295	0.129	0.226	0.222	0.193	0.154	0.227	0.157	0.281
Mn (wt%)	1.321	1.339	1.164	0.805	0.823	0.809	1.156	0.391	1.191	1.141
S (wt%)	0.033	0.041	0.128	0.208	0.114	0.158	0.181	0.189	0.012	0.095
Cr (wt%)	1.874	1.952	1.794	2.293	1.645	2.830	2.462	2.315	1.549	1.468
Mo (wt%)	0.207	0.747	0.384	0.151	0.024	0.152	0.335	0.667	0.955	0.797
Ni (wt%)	0.317	3.024	3.116	2.525	1.138	1.699	0.959	2.926	0.131	2.323
Al (wt%)	0.262	0.339	0.491	0.983	0.419	0.706	0.479	0.018	0.121	0.164
V (wt%)	0.180	0.079	0.139	0.233	0.096	0.045	0.172	0.187	0.237	0.177
Hardening Temperature (°C)	924.4	958.2	901.6	885.2	823.9	893.6	880.0	936.9	962.2	962.3
Cooling Medium Number	1	1	2	2	1	2	1	2	1	1
Tempering Temperature (°C)	534.2	300.4	413.1	513.2	640.4	316.6	383.1	595.2	289.0	680.1
ROA (%)	60.07	60.06	59.96	59.92	60.019	59.93	59.97	59.96	60.01	59.98



## Chapter 7: Application of RSSA and MO-RSSA

---

In the second experiment, the  $ROA_{Target}$  was set to 70%. Figure 7-5 shows the optimisation process and Table 7-4 lists the optimisation results relating to 10 different runs. The average function evaluation number employed in locating the optimal solutions is 28.

From the results of the above two experiments, it can be seen that, for the optimisation problem of achieving a predefined target ROA, there are many feasible solutions and the RSSA algorithm can find these optimal solutions in different runs using only a few function evaluations.

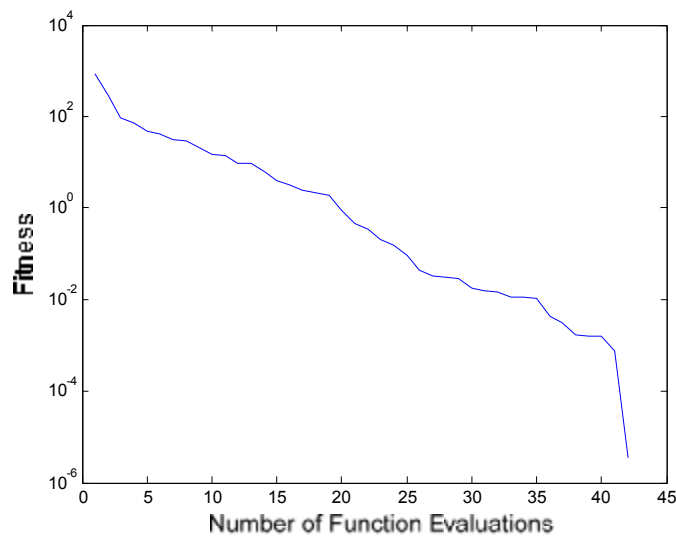


Figure 7-5. Average fitness of 10 runs versus function evaluation for the ROA design problem with  $ROA_{Target} = 70$  (%)

## Chapter 7: Application of RSSA and MO-RSSA

Table 7-4. Optimisation solutions of 10 independent runs for the ROA design problem with  $ROA_{Target} = 70$  (%)

Solutions	1	2	3	4	5	6	7	8	9	10
Test Depth (mm)	125.3	76.9	113.4	57.3	9.2	34.1	21.2	81.5	110.4	130.7
Size (mm)	238.2	136.4	318.9	43.7	162.0	219.1	248.5	204.3	287.5	82.6
Site Number	1	5	4	5	5	5	3	2	3	5
C (wt%)	0.309	0.450	0.536	0.571	0.304	0.360	0.424	0.346	0.538	0.148
Si (wt%)	0.251	0.138	0.336	0.189	0.331	0.244	0.312	0.120	0.334	0.136
Mn (wt%)	1.229	0.383	1.322	0.643	1.074	0.645	0.356	0.695	0.943	0.425
S (wt%)	0.032	0.152	0.046	0.108	0.153	0.030	0.170	0.103	0.120	0.056
Cr (wt%)	1.677	2.212	1.868	3.435	2.584	1.371	2.188	2.850	2.590	0.094
Mo (wt%)	0.905	0.856	0.271	0.067	0.100	0.773	0.938	0.210	0.430	0.955
Ni (wt%)	2.405	0.730	1.708	3.689	2.213	0.797	0.534	1.273	1.275	1.640
Al (wt%)	0.290	0.912	0.612	0.258	0.730	0.191	0.628	0.542	0.241	0.749
V (wt%)	0.084	0.143	0.030	0.173	0.219	0.194	0.213	0.128	0.133	0.097
Hardening Temperature (°C)	921.2	968.9	943.2	938.6	937.6	926.2	931.6	881.3	948.4	863.5
Cooling Medium Number	3	2	1	3	2	1	2	2	2	2
Tempering Temperature (°C)	371.8	567.6	719.8	492.0	326.8	708.4	669.2	279.4	670.2	385.5
ROA (%)	70.09	69.96	69.99	69.99	69.904	70.05	70.03	69.94	69.92	69.92

#### **7.4 The Optimal Design of both UTS and ROA**

In the design of alloy steels, sometimes it is required to achieve a predefined target UTS value and a predefined target ROA value simultaneously. For this problem, one should first judge whether such requirements are possible. If the answer is 'yes', then the problem can be solved as a single objective optimisation problem by combining these two objectives into a weighted sum formulation. However, if the answer is 'no', then the problem should be solved using the multi-objective optimisation technique, which is able to offer a set of approximate candidate solutions (Pareto-optimal solutions). In order to ascertain both scenarios, the achievable minimum and maximum boundaries are needed.

In this section, the decision vector of these design problems consists of weight percentages of Carbon (C), Manganese (Mn), Chromium (Cr), Molybdenum (Mo), and tempering temperature.

##### **7.4.1 Boundaries for the UTS and ROA Design**

To obtain the mechanical property boundaries for alloy steels design, the multi-objective optimisation technique was employed. Two distinct relevant multi-objective optimisation problems were defined as follows:

1. Minimising UTS and ROA simultaneously, i.e.:

Objective 1: Minimise UTS

## Chapter 7: Application of RSSA and MO-RSSA

Objective 2: Minimise ROA (7.3)

2. Maximising UTS and ROA simultaneously, i.e.:

Objective 1: Maximise UTS

Objective 2: Maximise ROA (7.4)

The MO-RSSA algorithm was employed to optimise the above problems and the maximum function evaluations number was set to 10,000. The obtained Pareto fronts using MO-RSSA are displayed in Figure 7-6. The region between the two fronts is where one can design the properties (UTS and ROA).

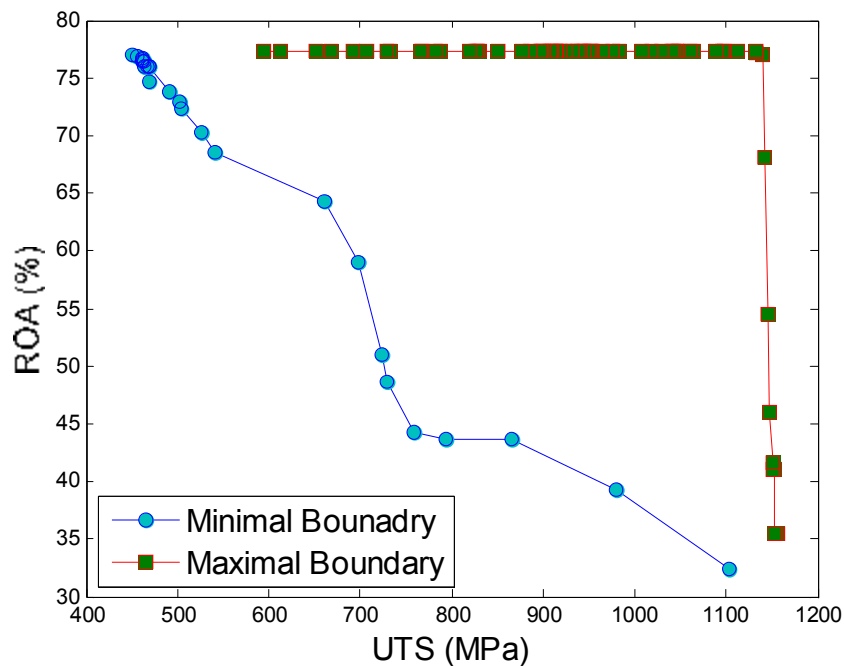


Figure 7-6. The maximum and minimum boundaries (Pareto fronts) for the problem of designing UTS and ROA simultaneously

## Chapter 7: Application of RSSA and MO-RSSA

### 7.4.2 The Single Objective Optimisation

If the target UTS and ROA are located between the design boundaries, then the single objective optimisation technique can be used to obtain the desired solution by optimising the following objective function:

$$\text{Minimise } J_3 = \left( \frac{UTS - UTS_{Target}}{900} \right)^2 + \left( \frac{ROA - ROA_{Target}}{60} \right)^2 \quad (7.5)$$

where  $UTS_{Target}$  is the target UTS value and  $ROA_{Target}$  is the target ROA value.

For instance, if  $UTS_{Target}$  is 900 MPa and  $ROA_{Target}$  is 60%, it can be seen from Figure 7-6 that the targets are located between the design boundaries. Table 7-5 shows the results of applying RSSA to optimise Problem (7.5) for 10 different runs. The average number of function evaluations needed for these 10 runs is 133.

Table 7-5. Optimisation solutions of 10 independent runs for the design problem with  $UTS_{Target} = 900$  (MPa) and  $ROA_{Target} = 70$  (%)

Solutions	1	2	3	4	5	6	7	8	9	10
C (wt%)	0.427	0.530	0.516	0.453	0.438	0.503	0.404	0.436	0.426	0.502
Mn (wt%)	1.642	1.484	1.160	1.281	1.511	0.362	1.189	1.561	0.692	1.047
Cr (wt%)	1.341	0.136	0.436	0.583	1.186	1.116	0.316	1.196	0.431	0.639
Mo (wt%)	0.886	0.230	0.208	0.252	0.884	0.233	0.633	0.879	0.848	0.183
Tempering Temperature (°C)	868.7	940.3	917.3	906.0	897.5	925.4	945.0	890.8	888.5	936.3
UTS (MPa)	900.0	900.6	899.5	900.3	900.5	899.9	899.1	900.6	900.2	900.3
ROA (%)	59.94	60.03	59.99	59.94	59.99	60.00	59.98	59.99	60.04	60.02

### 7.4.3 The Multi-objective Optimisation

If the target UTS and ROA are located outside the design boundaries, then no precise solutions can be found to satisfy the desired targets. In this case, the multi-objective optimisation technique can be used to obtain a set of Pareto-optimal solutions, which are regarded as the possible candidate solutions.

The design problem can be described as follows:

$$\begin{aligned} \text{Objective 1: Minimise } J_1 & \left( \frac{UTS - UTS_{Target}}{900} \right)^2 \\ \text{Objective 2: Minimise } J_2 & \left( \frac{ROA - ROA_{Target}}{60} \right)^2 \end{aligned} \quad (7.6)$$

where  $UTS_{Target}$  is the target UTS value and  $ROA_{Target}$  is the target ROA value.

For example, if the design targets  $UTS_{Target}$  is 600 MPa and  $ROA_{Target}$  is 50%, then from Figure 7-6 it can be seen that the targets are beyond the lower design boundary. In this type of a situation, the multi-objective optimisation algorithm MO-RSSA should suitably be employed to optimise the above Problem (7.6) with a maximum function evaluations number being set to 10,000 for instance. The obtained Pareto-optimal solutions are shown in Figure 7-7 and Table 7-6 provides details of 10 of these solutions.

**Chapter 7: Application of RSSA and MO-RSSA**

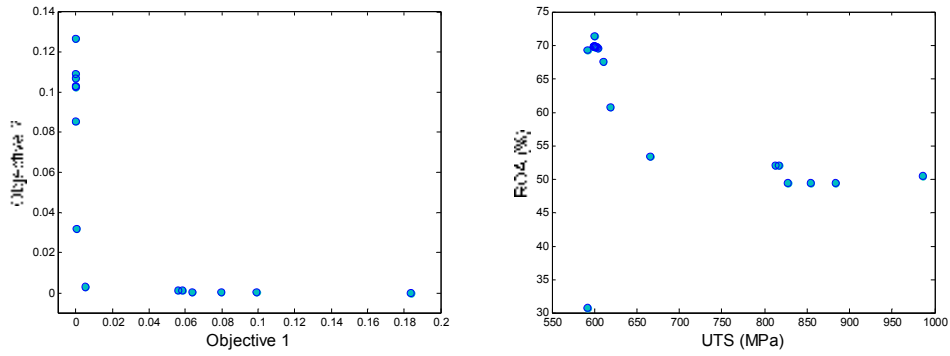


Figure 7-7. The performance of the Pareto-optimal solutions for the design problem of  $UTS_{Target} = 600$  (MPa) and  $ROA_{Target} = 50$  (%) with respect to (a) the Objective 1 and the Objective 2 and (b) the UTS and the ROA

Table 7-6. Pareto-optimal solutions for the design problem with  $UTS_{Target} = 600$  (MPa) and  $ROA_{Target} = 50$  (%)

Solutions	1	2	3	4	5	6	7	8	9	10
C (wt%)	0.528	0.211	0.474	0.408	0.235	0.209	0.225	0.224	0.226	0.136
Mn (wt%)	0.523	1.485	1.668	1.535	1.157	1.535	0.731	1.128	0.474	0.634
Cr (wt%)	1.706	3.130	1.874	2.158	1.218	2.158	0.253	0.278	0.257	1.933
Mo (wt%)	0.751	0.679	0.987	0.988	0.897	0.988	0.662	0.063	0.056	0.890
Tempering Temperature (°C)	969.3	903.5	978.9	941.5	880.2	839.8	847.0	847.1	847.0	914.4
UTS (MPa)	985.9	853.9	817.4	812.9	665.6	619.0	610.8	600.0	599.9	591.8
ROA (%)	50.43	49.41	52.02	52.02	53.29	60.70	67.52	69.80	71.35	69.22

From the experimental results in this section, the following can be observed:

1. For an optimal design problem with two conflicting targets, MO-RSSA is able to find the design boundaries, which is used to ascertain two

## ***Chapter 7: Application of RSSA and MO-RSSA***

---

different design scenarios.

2. If the target values locate between the design boundaries, RSSA can be used to obtain the desired precise solutions successfully.
  
3. If the target values are located outside the design boundaries, MO-RSSA can be used to obtain a set of approximate candidate solutions (Pareto-optimal solutions) successfully.

### **7.5 The Optimal Alloy Design Considering both the Mechanical Properties and the Economical Factors**

This study consists of finding the optimal chemical compositions and heat-treatment process parameters in order to obtain the required UTS and ROA while minimising the production costs. The production costs of heat-treated steels include the costs of the addition of alloying elements, such as Cr, Mo, V, etc. and the costs of energy consumption during the heat-treatment process.

In this experiment, five decision variables, C, Mn, Cr, Mo and Tempering Temperature, have been considered although other composites and temperatures could also be included. The factors contributing to the cost of heat treatment operation are summarised in Tables 7-7, 7-8 [Mahfouf *et al.* 2002].



## Chapter 7: Application of RSSA and MO-RSSA

Table 7-7. Contribution of composites to the cost of heat treatment

Composite	Cost (US\$ per tonne)
Manganese	18
Chromium	42
Molybdenum	52

Table 7-8. Contribution of tempering (annealing) to the cost of heat treatment

Item	Cost (US\$: 1.3GJ/tonne at 600 ° C)
Annealing (tempering)	4.88

### 7.5.1 The Optimal Design Considering both UTS and the Cost

According to the contribution of the chemical composites and the tempering process to the cost of heat-treated steels, a new objective function to reflect such costs was introduced as follows:

$$J_{\text{cost}} = \left( \frac{18Mn + 42Cr + 52Mo + 4.88Temp / 600}{100} \right)^2 \quad (7.7)$$

By taking into account such economic consideration, the problem of designing an alloy steel with a predefined target UTS property becomes a two-objective optimisation problem described as follows:

$$\text{Objective 1: Minimise } J_1 = \left( \frac{UTS - UTS_{\text{target}}}{900} \right)^2$$

## Chapter 7: Application of RSSA and MO-RSSA

$$\text{Objective 2: Minimise } J_{\text{cost}} \quad (7.8)$$

Figure 7-8 displays the obtained Pareto-optimal solutions in the objective space with the UTS target value  $UTS_{\text{target}} = 900$  (MPa). Ten (10) various solutions around the UTS target value are selected from the Pareto-optimal solutions and listed in Table 7-9.

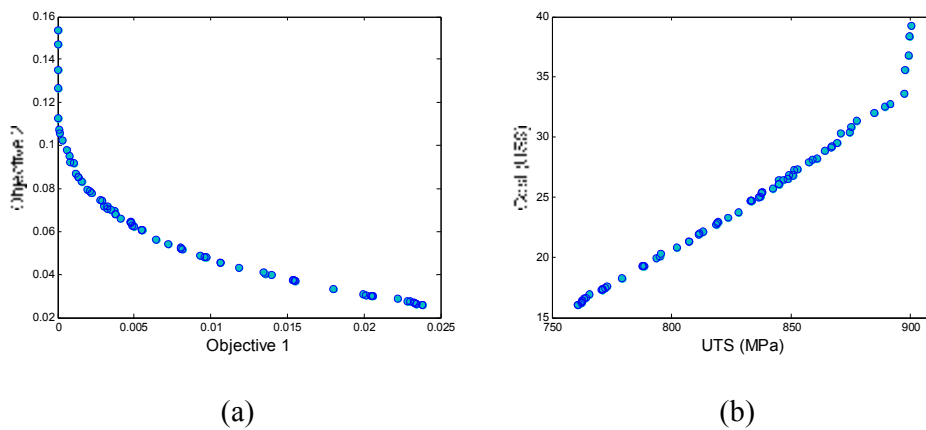


Figure 7-8. The performance of the Pareto-optimal solutions for the design problem of  $UTS_{\text{target}} = 900$  (MPa) and minimising the heat treatment cost with respect to (a) Objective 1 and Objective 2; (b) UTS and Cost

## Chapter 7: Application of RSSA and MO-RSSA

Table 7-9. Ten of the Pareto-optimal solutions for the design problem of  $UTS_{Target} = 900$  (MPa) and minimising the heat treatment cost

Solutions	1	2	3	4	5	6	7	8	9	10
C (wt%)	0.619	0.618	0.619	0.618	0.619	0.619	0.619	0.619	0.619	0.619
Mn (wt%)	1.661	0.738	1.101	1.031	0.921	0.853	0.846	0.799	0.734	0.694
Cr (wt%)	0.050	0.050	0.051	0.061	0.050	0.050	0.060	0.050	0.060	0.051
Mo (wt%)	0.010	0.205	0.050	0.047	0.053	0.051	0.017	0.010	0.010	0.010
Tempering Temperature (°C)	821.9	822.4	821.6	823.6	823.1	821.7	821.7	821.6	821.3	821.9
UTS (MPa)	900.0	891.5	877.4	870.8	860.6	850.7	838.0	827.8	819.3	811.3
Cost (US\$)	39.22	32.75	31.30	30.33	28.20	26.81	25.38	23.73	22.96	21.88

### 7.5.2 The Optimal Design Considering both ROA and the Cost

By considering both the ROA and the economical factors, the following two-objective optimisation problem can be set:

$$\text{Objective 1: Minimise } J_2 \left( \frac{ROA - ROA_{Target}}{60} \right)^2$$

$$\text{Objective 2: Minimise } J_{cost} \quad (7.9)$$

Figure 7-9 shows the obtained Pareto-optimal solutions in the objective space, where the ROA target value  $ROA_{Target}$  is 60%. Ten different solutions around the ROA target value are selected from the Pareto-optimal solutions and listed in Table 7-10.

**Chapter 7: Application of RSSA and MO-RSSA**

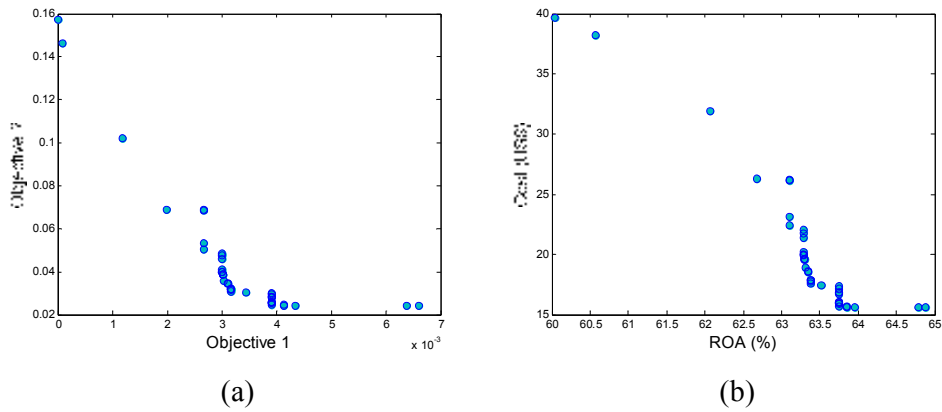


Figure 7-9. The performance of the Pareto-optimal solutions for the design problem of  $ROA_{Target} = 60$  (%) and minimising the heat treatment cost with respect to (a) Objective 1 and Objective 2; (b) ROA and Cost

Table 7-10. Ten of the Pareto-optimal solutions for the design problem of  $ROA_{Target} = 60$  (%) and minimising the heat treatment cost

Solutions	1	2	3	4	5	6	7	8	9	10
C (wt%)	0.436	0.611	0.467	0.599	0.607	0.607	0.614	0.562	0.562	0.562
Mn (wt%)	0.839	0.820	0.995	0.454	0.597	0.579	0.448	0.351	0.351	0.351
Cr (wt%)	0.242	0.149	0.089	0.050	0.113	0.076	0.050	0.050	0.050	0.050
Mo (wt%)	0.126	0.194	0.058	0.169	0.010	0.025	0.010	0.010	0.010	0.010
Tempering Temperature (°C)	960.2	870.7	888.6	882.3	868.6	867.0	830.3	862.7	820.8	820.1
ROA (%)	60.04	60.57	62.07	62.67	63.10	63.28	63.51	63.75	63.95	64.87
Cost (US\$)	39.65	38.23	31.94	26.27	23.11	22.03	17.44	15.96	15.62	15.61

## ***Chapter 7: Application of RSSA and MO-RSSA***

---

### **7.5.3 The Optimal Design Considering UTS, ROA and the Cost**

Taking into account all the three factors, i.e. UTS, ROA and the cost of the heat treatment, the problem of designing an alloy steel can be described as follows:

$$\begin{aligned} \text{Objective 1: Minimise } J_1 & \left( \frac{UTS - UTS_{Target}}{900} \right)^2 \\ \text{Objective 2: Minimise } J_2 & - \left( \frac{ROA - ROA_{Target}}{60} \right)^2 \\ \text{Objective 3: Minimise } J_{cost} & \end{aligned} \quad (7.10)$$

An optimisation experiment has been conducted based on the above objectives where the target values  $UTS_{Target} = 900$  (MPa) and  $ROA_{Target} = 60$  (%). The result of this experiment is shown in Figure 7-10. Ten (10) solutions out of all the obtained Pareto-optimal solutions are selected and listed in Table 7-11.

From the above experiments, it can be seen that, for the optimal design problems that consider both the mechanical properties and the economical factors, MO-RSSA is able to obtain a set of optional solutions (Pareto-optimal solutions), which are close to the predefined UTS and/or ROA targets while providing various levels of heat treatment costs.

## Chapter 7: Application of RSSA and MO-RSSA

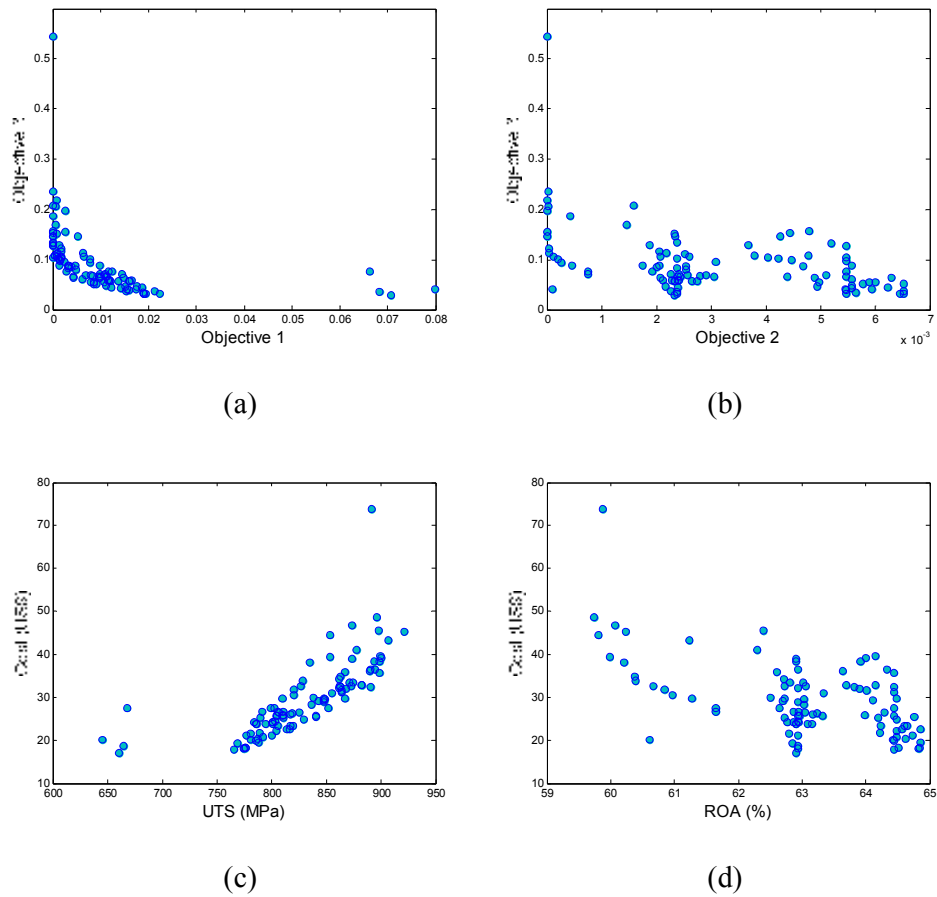


Figure 7-10. The performance of the Pareto-optimal solutions for the design problem of  $UTS_{Target} = 900$  (MPa),  $ROA_{Target} = 60$  (%) and minimising the heat treatment cost with respect to (a) Objective 1 and Objective 3; (b) Objective 2 and Objective 3; (c) UTS and Cost; and (d) ROA and Cost

## Chapter 7: Application of RSSA and MO-RSSA

Table 7-11. Ten of the Pareto-optimal solutions for the design problem of  $UTS_{Target} = 900$  (MPa),  $ROA_{Target} = 60$  (%) and minimising the heat treatment cost

Solutions	1	2	3	4	5	6	7	8	9	10
C (wt%)	0.612	0.602	0.604	0.598	0.441	0.613	0.606	0.536	0.531	0.619
Mn (wt%)	0.608	0.740	1.332	0.796	0.701	0.903	0.458	0.811	0.795	0.998
Cr (wt%)	0.357	0.295	0.050	0.050	0.878	0.050	0.366	0.244	0.208	0.050
Mo (wt%)	0.233	0.195	0.118	0.253	0.325	0.143	0.199	0.287	0.276	0.012
Tempering Temperature (°C)	892.1	895.2	849.3	840.9	898.8	852.2	862.4	849.6	856.5	831.9
UTS (MPa)	921.4	906.8	900.6	894.5	891.9	882.5	877.4	873.9	853.3	851.7
ROA (%)	60.24	61.23	64.00	64.32	59.87	63.69	62.29	60.07	59.80	64.44
Cost (US\$)	45.34	43.21	39.14	36.45	73.75	32.73	41.05	46.71	44.44	27.51

### 7.6 Summary

In this chapter, RSSA and MO-RSSA have been successfully applied to single objective and multi-objective optimal design of alloy steels. This research aims at determining the optimal heat treatment regime and the required weight percentages for the chemical composites to obtain the desired mechanical properties of steel such as UTS and ROA. In addition, the work was later extended to include economic factors, such as the costs associated with the composites and the tempering operation. The simulation results showed that MO-RSSA is able to produce a range of well-spread optional solutions around the property targets while maintaining reasonable production cost.

## **Chapter 8**

# **Conclusion and Future work**

### **8.1 Summary of Main Results**

In this thesis, a new nature-inspired optimisation algorithm, Reduced Space Searching Algorithm (RSSA) was proposed. The inspiration behind this optimisation algorithm originated from the simulation of a simple human societal behaviour when searching for optimal solutions in our daily routines. This new algorithm was validated using a set of well-known challenging benchmark problems. Compared with the recently developed and most salient optimisation algorithms, CMA-ES, DE and G3-PCX, RSSA performed as well as and sometimes better than these algorithms. Furthermore, RSSA was extended to the multi-objective optimisation case (MO-RSSA) by using the Random Weighted Aggregation (RWA) technique and maintaining an archiving mechanism for the solutions. A comparative study based on the ZDT and DTLZ series problems showed that MO-RSSA is effective in finding the Pareto-optimal solutions and



## ***Chapter 8: Conclusion and Future work***

---

possesses advantages over some evolutionary algorithms, such as PAES, SPEA and NSGA-II, in terms of accuracy and diversity of the final solutions.

Another new optimisation algorithm, nPSO, was then proposed. It introduced a new ‘momentum term’ to replace the original inertia term of the standard PSO. The advantages of nPSO lie in its ability to avoid premature convergence and its adaptability in different situations. This algorithm has been validated using a set of benchmark problems and was shown to lead to a better performance than the standard PSOs and some other salient optimisation algorithms. Furthermore, nPSO was extended to the multi-objective optimisation case, in which the RWA technique was employed and a new approach named ‘cell selection’ method was introduced in order to keep a good diversity of the Pareto-optimal solutions. After a comparison between the new multi-objective PSO (nMPSO) and several well-known evolutionary multi-objective algorithms, such as PAES, SPEA and NSGA-II, it was concluded that nMPSO shows noticeable improvements over these algorithms in terms of both accuracy and diversity of the Pareto solutions when using the set of challenging benchmarks ZDT and DTLZ series problems.

Based on the developed optimisation algorithms, the framework for data-driven fuzzy modelling, FM-HCMO, was proposed in order to construct linguistic fuzzy models considering both the accuracy and the interpretability of fuzzy systems. In this methodology, a new agglomerative complete-link clustering algorithm was first developed and applied to construct the initial fuzzy model. A new data selection technique was then proposed to select representative training data used to

## ***Chapter 8: Conclusion and Future work***

---

improve the modelling efficiency. Furthermore, a multi-objective optimisation mechanism was developed for the performance improvement of the initial model, which takes into account both the accuracy and the interpretability attributes. Finally, a method for computing the confidence bands relating to the model prediction analysis was also proposed.

FM-HCMO was validated via some benchmark problems, which include the identification of nonlinear, static or dynamic systems. It can be concluded that FM-HCMO provides a simple and effective framework for system identification and prediction and the linguistic models constructed using it lead to good generalisation properties.

FM-HCMO was then applied to model the mechanical properties of alloy steels using the real data from the steel industry. It was demonstrated that this proposed approach works effectively in eliciting accurate and interpretable models within the context of high-dimensional industrial applications.

Finally, RSSA and MO-RSSA were successfully applied to the single objective and the multi-objective optimal design of alloy steels. This research aimed at determining the optimal heat treatment regime and the required weight percentages for the chemical composites to obtain the desired mechanical properties of steel such as TS and ROA. In addition, this work was extended to include economic factors, such those costs associated with the composites and the tempering operation. The simulation results showed that MO-RSSA was able to produce a

## ***Chapter 8: Conclusion and Future work***

---

range of well-spread optional solutions around the property targets while maintaining reasonable production costs. These findings indicate that the algorithms, RSSA and MO-RSSA, can effectively be used for industrial optimisation problems and will be beneficial for the steel industry as a whole.

### **8.2 Future Work**

With the availability of the data-driven modelling methodology and the optimisation algorithms, one can apply them into various applications. To facilitate the usage of these modelling and optimisation tools, a Graphical User Interface (GUI), which will allow the information extraction and analysis through simple operations, is needed.

Figure 8-1 shows a tentative version of the fuzzy modelling tool, which has integrated and implemented parts of the proposed techniques. This GUI tool can be used to construct a fuzzy model from a set of loaded data, which is shown in Figure 8-2. In addition, as shown in Figure 8-3, the basic analysis functions for the loaded data and the obtained models are also provided.

Future developments will focus on the design and implementation of this flexible GUI software tool that embodies all the proposed algorithms and methods with possibility of further extension to include newly devised techniques. This tool will allow the application of the proposed techniques for solving the modelling and

## Chapter 8: Conclusion and Future work

optimisation problems emanating from the real world in a more convenient way.

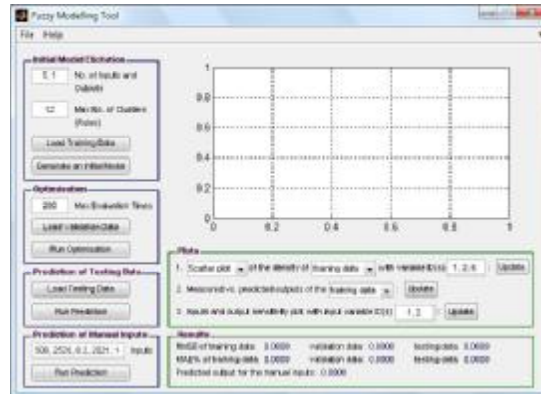


Figure 8-1. The layout of the designed fuzzy modelling GUI

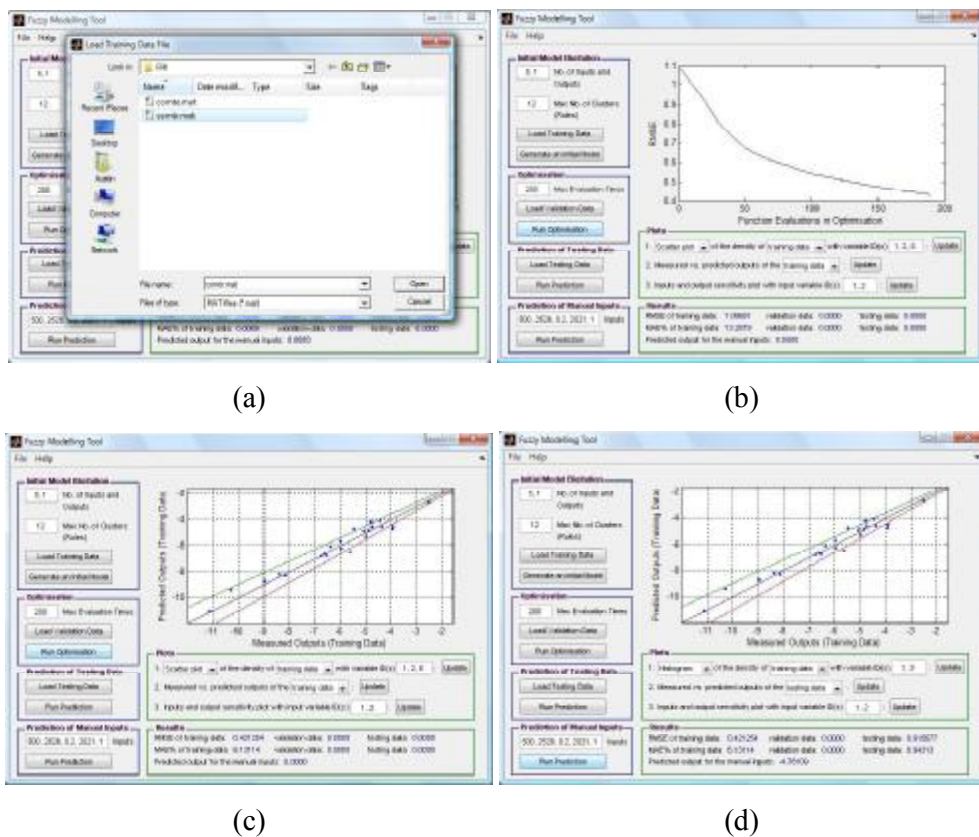


Figure 8-2. An example of the modelling process using the designed fuzzy modelling GUI: (a) loading data, (b) running the optimisation, (c) displaying the prediction performance of the training data and (d) output predicting from a set of provided discrete inputs

## Chapter 8: Conclusion and Future work

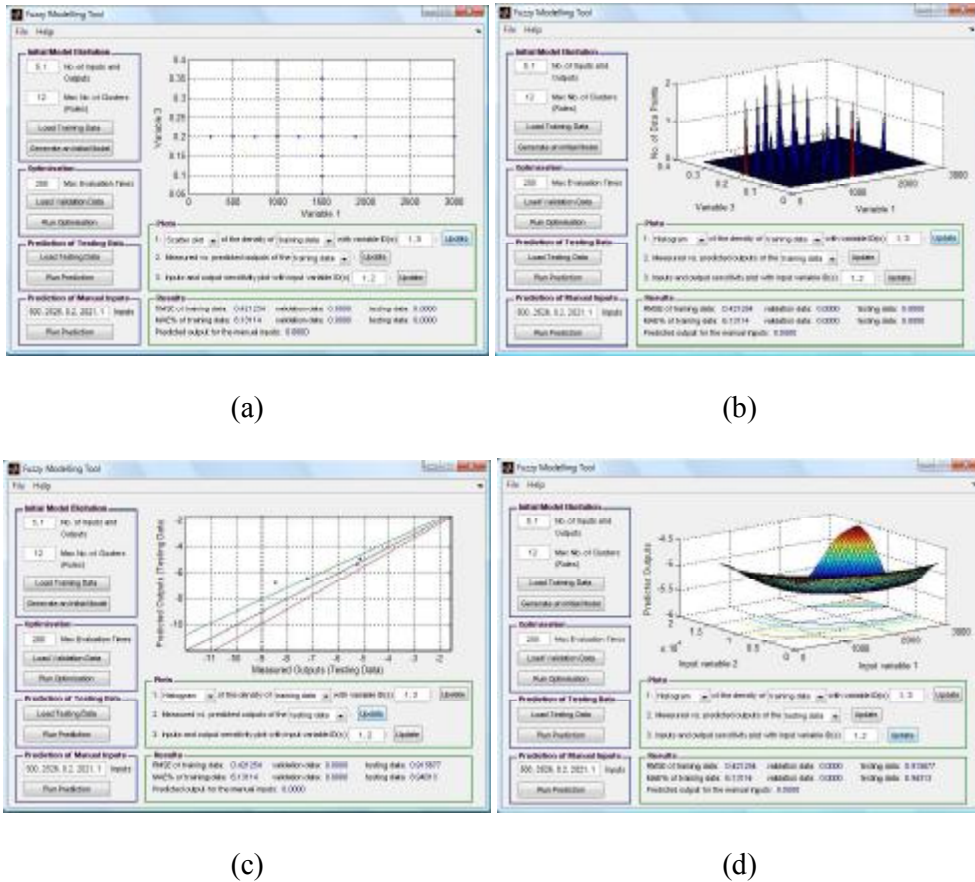


Figure 8-3. An example of analysing the loaded data and the obtained model using the designed fuzzy modelling GUI: (a) the training data with variable ID(s) being 1 and 3, (b) the histograms of the density of the training data with variable ID(s) being 1 and 3, (c) the measured outputs vs. the predicted outputs of the training data, and (d) the inputs and output sensitivity plot (response surface) with various variable ID(s) being 1 and 2

Besides the above work, the following paragraphs summarise some most remarkable research trends in the field of evolutionary optimisation and fuzzy modelling (from the author's perspective):

1. **Theoretical foundations for EAs (including MOEAs):** In the future, more research work should be carried-out to develop the theoretical

## ***Chapter 8: Conclusion and Future work***

---

foundations of EAs. The theoretical aspects of EAs are important, which can help to understand why they work. The generation of new algorithms and associated operators can also take the advantage of the theoretical understanding of EAs. So far, only little research have been carried out in this area; for instance, Rudolph and Agapie [2000] have proved the convergence properties of some EAs using Markov Chains.

2. **The integration of domain knowledge into modelling and optimisation:** In the modelling and/or optimisation problems that include two or more conflicting objectives, providing all the Pareto-optimal solutions may not be the most effective way of solving the problems. In this situation, the problem domain information, which may be represented as the users' preference [Deb & Sundar 2006; Ishibuchi & Nojima 2007], can be introduced into the modelling and/or optimisation procedure. This integration has the potential to improve the effectiveness and the accuracy in modelling and optimisation.
  
3. **The extension of fuzzy modelling with the inclusion of stochastic modelling:** With the inclusion of stochastic reasoning, some successful optimisation algorithms, such as the Estimation of Distribution Algorithms (EDAs) [Larranaga & Lozano 2001] and the RSSA algorithm (see Chapter 3) were developed. These stochastic processes provide the algorithms with more adaptation abilities. In the same way, some stochastic techniques and statistical learning theories [Cherkassky

## ***Chapter 8: Conclusion and Future work***

---

& Mulier 1998], such as Bayesian networks [Jensen 2001], may cooperate with fuzzy modelling (a deterministic method). This will improve the generalisation ability of fuzzy systems and will offer a better adaptation to fuzzy modelling in dealing with some complex problems, especially those characterised by high degrees of uncertainty.

### **8.3 Conclusion**

In this research project, two evolutionary optimisation algorithms, namely the Reduced Space Searching Algorithm (RSSA) and the new Particle Swarm Optimisation algorithm (nPSO), were developed. Based on these proposed optimisation algorithms, a data-driven modelling methodology, named the Fuzzy Modelling approach with a Hierarchical Clustering algorithm and a Multi-objective Optimisation mechanism (FM-HCMO), was designed. The FM-HCMO approach was successfully implemented within an industrial application, which consists of eliciting prediction models of the mechanical properties of alloy steels using real industry data. Using these developed models, the proposed optimisation algorithms were successfully applied to single objective and multi-objective optimal designs of alloy steels.

## Bibliography

- Alcala, R., Alcala-Fdez, J., Herrera, F., Otero, J. (2007) “Generic learning of accurate and compact rule based systems based on the 2-tuples linguistic representation” *International Journal of Approximate Reasoning*, vol. 44, pp. 45-64
- Alcala, R., Cano, J. R., Cordon, O., Herrera, F., Villar, P., Zwir, I. (2003) “Linguistic modeling with hierarchical systems of weighted linguistic rules” *International Journal of Approximate Reasoning*, vol. 32, pp. 187-215
- Babuska, R. (1998) *Fuzzy Modeling for Control*, Boston: Kluwer Academic Publishers
- Back, T. (1996) *Evolutionary Algorithms in Theory and Practice: Evolution Strategies, Evolutionary Programming, Genetic Algorithms*, New York: Oxford University Press
- Back, T., Fogel, D. Michalewicz, Z. (1997) *Handbook of Evolutionary Computation*, Bristol: Institute of Physics Publishing and New York: Oxford University Press
- Baeza-Yates, R. A. (1992) “Introduction to data structures and algorithms related to information retrieval” *Information Retrieval: Data Structures and Algorithms*, Frakes, W. B., Baeza-Yates, R. (Eds.) NJ: Prentice-Hall, pp. 13-27
- Bakshi B. R., Chatterjee, R. (1998) “Unification of neural and statistical methods as applied to materials structure-property mapping” *Journal of Alloys Compounds*, vol. 279, no. 1, pp. 39-46
- Beni, G., Wang, J. (1989) “Swarm Intelligence in Cellular Robotic Systems” *Proceedings of the NATO Advanced Workshop on Robots and Biological Systems*, Tuscany, Italy



## Bibliography

- Berkhin, P. (2006) "A Survey of Clustering Data Mining Techniques" *Grouping Multidimensional Data: Recent Advances in Clustering*, Kogan, J., Nicholas, C., Teboulle, M. (Eds.) Berlin: Heidelberg: Springer, pp. 25-71
- Beyer, H.-G. (2001) *The Theory of Evolution Strategies*, Springer
- Bezdek, J. C. (1981) *Pattern Recognition with Fuzzy Objective Function Algorithms*, New York: Plenum Press
- Bull, L., Kovacs, T. (2005) *Foundations of Learning Classifier Systems*, Springer
- Casillas, J., Cordon, O., Del Jesus, M. J., Herrera, F. (2005) "Genetic Tuning of Fuzzy Rule Deep Structures Preserving Interpretability and Its Interaction With Fuzzy Rule Set Reduction" *IEEE Transactions on Fuzzy Systems*, vol. 13, no. 1, pp. 13-29
- Castillo, L., Gonzalez, A., Perez, R. (2001) "Including a simplicity criterion in the selection of the best rule in a genetic fuzzy learning algorithm" *Fuzzy Sets and Systems*, vol. 120, pp. 309-321
- Chen C. L. P., Yang, C., LeClair, S. R. (1998) "Materials structure-property prediction using a self-architecting neural network" *Journal of Alloys Compounds*, vol. 279, no. 1, pp. 30-38
- 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 - Part B: Cybernetics*, vol. 31, no. 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
- Cherkassky, V., Mulier, F. (1998) *Learning from Data: Concepts, Theory, and Methods*, New York: John Wiley & Sons
- Coello Coello, C. A. (1999) "An Updated Survey of Evolutionary Multiobjective Optimization Techniques: State of the Art and Future Trends" *Proceedings of the 1999 IEEE Congress on Evolutionary Computation*, vol. 1, pp. 3-13
- Coello Coello, C. A. (2003) "Evolutionary Multiobjective Optimization: Current and Future Challenges" *Advances in Soft Computing-Engineering: Design and Manufacturing*, Benitez, J., Cordon, O., Hoffmann, F., Roy, R. (Eds.), pp. 243-256, Springer-Verlag
- Coello Coello, C. A., Lechuga, M. (2002) "MOPSO: A proposal for multiple objective particle swarm optimization" *Proceedings of the 2002 IEEE Congress on Evolutionary Computation*, vol. 2, pp. 1051-1056

## Bibliography

- Coello Coello, C. A., Pulido, G., Lechuga, M. (2004) "Handling multiple objectives with particle swarm optimization" *IEEE Transactions on Evolutionary Computation*, vol. 8, no. 3, pp. 256-279
- 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. (1997) "A Three-Stage Evolutionary Process for Learning Descriptive and Approximate Fuzzy Logic Controller Knowledge Bases from Examples" *International Journal of Approximate Reasoning*, vol. 17, no. 4, pp. 369-407
- Cordon, O., Herrera, F., Hoffmann, F., Magdalena, L. (2001) *Genetic Fuzzy Systems – Evolutionary Tuning and Learning of Fuzzy Knowledge Bases*, Singapore: World Scientific
- Cordon, O., Herrera, F., Zwir, I. (2003) "A hierarchical knowledge-based environment for linguistic modeling: models and iterative methodology" *Fuzzy Sets and Systems*, vol. 138, pp. 307-341
- Cross, V. V. (1993) *An analysis of fuzzy set aggregators and compatibility measures*, Ph.D. thesis, Wright State University, Dayton, OH
- De Castro, L. N., Timmis, J. (2002) *Artificial Immune Systems: A New Computational Intelligence Approach*, Springer
- De Jong, K. A. (2006) *Evolutionary computation: a unified approach*, Cambridge MA: MIT Press
- Deb, K. (2001) *Multi-objective Optimization Using Evolutionary algorithms*, New York: Chichester: Wiley
- Deb, K. (2005) "A population-based algorithm-generator for real-parameter optimisation" *Soft Computation*, vol. 9, no. 4, pp. 236-253
- Deb, K., Anand, A., Joshi, D. (2002) "A computationally efficient evolutionary algorithm for real-parameter optimisation" *Evolutionary Computation*, vol. 10, no. 4, pp. 371-395
- Deb, K., Pratap, A., Agarwal, S., Meyarivan, T. (2002) "A Fast and Elitist Multiobjective Genetic Algorithm: NSGA-II" *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 2, pp. 182-197
- Deb, K., Sundar, J. (2006) "Reference point based multi-objective optimization using evolutionary algorithms" *Proceedings of GECCO 2006*, pp. 635-642

## Bibliography

- Deb, K., Thiele, L., Laumanns, M., Zitzler, E. (2001) "Scalable Test Problems for Evolutionary Multi-Objective Optimisation" Technical Report 112, Computer Engineering and Networks Laboratory, Swiss Federal Institute of Technology
- Del Valle, Y., Venayagamoorthy, G. K., Mohagheghi, S., Hernandez, J.-C., Harley, R. G., (2008) "Particle swarm optimization: basic concepts, variants and applications in power systems" *IEEE Transactions on Evolutionary Computation*, vol. 12, no. 2, pp. 171-195
- Delgado, M, Gomez-Skarmeta, A. F., Martin, F. (1997) "A fuzzy clustering-based rapid prototyping for fuzzy rule-based modelling" *IEEE Transactions on Fuzzy Systems*, vol. 5, no. 2, pp. 223-233
- Doctor, S., Venayagamoorthy, G., Gudise, V. (2004) "Optimal PSO for collective robotic search applications" *Proceedings of the 2004 IEEE Congress on Evolutionary Computation*, vol. 2, pp. 1390-1395
- Dorigo, M., Maniezzo, V., Colorni, A. (1996) "Ant System: Optimization by a Colony of Cooperating Agents" *IEEE Transactions on Systems, Man, and Cybernetics – Part B*, vol. 26, no. 1, pp. 29–41
- Dorigo, M., Stutzle, T. (2004) *Ant Colony Optimization*, Cambridge, MA: MIT Press
- Dubois, D., Prade, H. (1988) *Fuzzy Sets and Systems*, New York: Academic Press
- Dunn, J. C. (1973) "A Fuzzy Relative of the ISODATA Process and Its Use in Detecting Compact Well-Separated Clusters" *Journal of Cybernetics*, vol. 3, pp. 32-57
- Eberhart, R. C., Kennedy, J. (1995) "A New Optimizer Using Particle Swarm Theory" *Proceedings of the 6th International Symposium on Micro Machine and Human Science*, pp. 39-43
- Eberhart, R. C., Shi, Y. (2000) "Comparing inertia weights and constriction factors in particle swarm optimization" *Proceedings of the 2000 IEEE Congress on Evolutionary Computation*, vol. 1, pp. 84-88
- Eberhart, R. C., Shi, Y. (2001a) "Particle swarm optimization: developments, applications and resources" *Proceedings of the 2001 IEEE Congress on Evolutionary Computation*, vol. 1, pp. 81-86
- Eberhart, R. C., Shi, Y. (2001b) "Tracking and optimizing dynamic systems with particle swarms" *Proceedings of the 2001 IEEE Congress on Evolutionary Computation*, pp. 94-97

## **Bibliography**

- Eberhart, R. C., Shi, Y., Kennedy, J. (2001) *Swarm Intelligence*, San Mateo, CA: Morgan Kaufmann
- Eiben, A. E., Smith, J. E. (2003) *Introduction to Evolutionary Computing*, Springer
- Emami, M. R., Turksen, I. B., Goldenberg, A. A. (1998) "Development of a systematic methodology of fuzzy logic modeling" *IEEE Transactions on Fuzzy Systems*, vol. 6, no. 3, pp. 346-361
- Fan, H., Shi, Y. (2001) "Study on Vmax of particle swarm optimization" *Proceedings of the Workshop on Particle Swarm Optimization*, Purdue School of Engineering and Technology, Indianapolis, IN, April 2001
- Fayyad, U. M., Piatetsky-Shapiro, G., Smyth P., Uthurusamy, R. (1996) *Advances in Knowledge Discovery and Data Mining*, Cambridge, MA: AAAI/MIT Press
- Fieldsend, J. E., Everson, R. M., Singh, S. (2003) "Using unconstrained elite archives for multiobjective optimization" *IEEE Transactions on Evolutionary Computation*, vol. 7, no. 3, pp. 305-323
- Fonseca, C. M., Fleming, P. J. (1993) "Genetic Algorithms for Multiobjective Optimization: Formulation, Discussion, and Generalization" *Proceedings of the Fifth International Conference on Genetic Algorithms*, pp. 416-423
- Fukuyama, Y., Sugeno, M. (1989) "A new method of choosing the number of clusters for the fuzzy c-means method" *Proceedings of the 5th Fuzzy System Symposium*, pp. 247-250
- Fuller, R. (1999) *Introduction to Neuro-Fuzzy System*, Wurzburg: Physica-Verlag
- Goldberg, D. E. (1989) *Genetic Algorithms in Search, Optimization and Machine Learning*, Reading, MA: Addison Wesley
- Gonzalez, A., Perez, R. (1999) "SLAVE: A Genetic Learning System Based on an Iterative Approach" *IEEE Transaction of Fuzzy Systems*, vol. 7, no. 2, pp. 176-191
- Gonzalez, J., Rojas, I., Pomares, H., Herrera, L. J., Guillen, 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, pp. 32-44
- Guillaume, S. (2001) "Designing Fuzzy Inference Systems from Data: An Interpretability-Oriented Review" *IEEE Transactions on Fuzzy Systems*, vol. 9, no. 3, pp. 426-443

## **Bibliography**

- Guillaume, S., Charnomordic, B. (2004) "Generating an Interpretable Family of Fuzzy Partitions From Data" *IEEE Transactions on Fuzzy Systems*, vol. 12, no. 3, pp. 324-335
- Hansen, N. (2007) "The CMA evolution strategy: a tutorial" Available from: <http://www.bionik.tu-berlin.de/user/niko/cmatutorial.pdf>
- Hansen, N., Muller, S. D., Koumoutsakos, P. (2003) "Reducing the time complexity of the derandomized evolution strategy with covariance matrix adaptation (CMA-ES)" *Evolutionary Computation*, vol. 11, no. 1, pp. 1-18
- Hodgson, P. D. (1996) "Microstructure modeling for property prediction and control" *Journal of Materials Processing Technology*, vol. 60, pp. 27-33
- Holland, J. H. (1975) *Adaptation in Natural and Artificial Systems*, Ann Arbor, MI: The University of Michigan Press
- Holland, J. H., Reitman, J. S. (1978) "Cognitive Systems Based on Adaptive Algorithms" *Pattern-Directed Inference Systems*, Waterman, D. A., Hayes-Roth, F. (Eds.), New York: Academic Press, pp. 313-329
- Horikawa, S., Furuhashi, T., Uchikawa, Y. (1992) "On fuzzy modeling using fuzzy neural networks with the backpropagation algorithm" *IEEE Transactions on Neural Networks*, vol. 3, no. 5, pp. 801-806
- Hu, X. (2006) "Particle swarm optimization" *Tutorial of the 2006 IEEE Swarm Intelligent Symposium*
- Hu, X., Eberhart, R. C. (2002) "Multiobjective optimization using dynamic neighborhood particle swarm optimization" *Proceedings of the 2002 IEEE Congress on Evolutionary Computation*, pp. 1677-1681
- Huang, T., Mohan, A. S. (2005) "A hybrid boundary condition for robust particle swarm optimization" *IEEE Antennas and Wireless Propagation Letters*, vol. 4, pp. 112-117
- Ishibuchi, H. (2007) "Evolutionary Multiobjective Design of Fuzzy Rule-Based Systems" *Proceedings of the 2007 IEEE Symposium on Foundations of Computational Intelligence*, pp. 9-16
- Ishibuchi, H., Nojima, Y. (2007) "Optimization of scalarizing functions through evolutionary multiobjective optimization" *Lecture Notes in Computer Science 4403: Evolutionary Multi-Criterion Optimization - EMO 2007*, pp. 51-65

## Bibliography

- Ishibuchi, H., Yamamoto, T. (2003) "Interpretability Issues in Fuzzy Genetics-Based Machine Learning for Linguistic Modelling" *Modelling with Words*, Lawry, J., Shanahan, J., Ralescu, A. (Eds.) Berlin: Heidelberg: Springer, pp. 209-228
- 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., Yamamoto, T., Nakashima, T. (2001) "Fuzzy data mining: effect of fuzzy discretization" *Proceedings of the First IEEE International Conference on Data Mining*, pp. 241-248
- Jain, A. K., Dubes, R. C. (1988) *Algorithms for Clustering Data*, Prentice-Hall advanced reference series, NJ: Prentice-Hall
- Jain, A. K., Murty, M. N., Flynn, P. J. (1999) "Data Clustering: A Review" *ACM Computing Surveys*, vol. 31, no. 3, pp. 264-323
- Jang, J. R., (1993) "ANFIS: adaptive network-based fuzzy inference system" *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 23, no. 4, pp. 665-684
- Jang, J.-S. R., Sun, C.-T., Mizutani, E. (1997) *Neuro-Fuzzy and Soft Computing*, Englewood Cliffs, NJ: Prentice-Hall
- Jimenez, F., Gomez-Skarmeta, A. F., Roubos, H., Babuska, R. (2001) "Accurate, Transparent, and Compact Fuzzy Models for Function Approximation and Dynamic Modeling through Multi-objective Evolutionary Optimization" *Proceedings of the First International Conference on Evolutionary Multi-criterion Optimization (EMO 2001)*, pp.653-667
- Jin, Y., von Seelen, W., Sendhoff, B. (1999) "On Generating FC<sup>3</sup> Fuzzy Rule Systems from Data Using Evolution Strategies" *IEEE Transactions on Systems, Man, and Cybernetics - Part B: Cybernetics*, vol. 29, no. 6, pp. 829-845
- Johansen, T. A., Babuska, R. (2003) "Multiobjective Identification of Takagi-Sugeno Fuzzy Models" *IEEE Transactions on Fuzzy Systems*, vol. 11, no. 6, pp. 847-860
- Karr, C. (1991) "Genetic Algorithms for Fuzzy Controllers" *AI Expert*, vol. 6, no. 2, pp. 26-33
- Kennedy, J. (1999) "Small worlds and mega-minds: Effects of neighborhood topology on particle swarm performance" *Proceedings of the 1999 IEEE Congress on Evolutionary Computation*, pp. 1931-1938

## Bibliography

- Kennedy, J., Eberhart, R. C. (1995) "Particle Swarm Optimization" *Proceedings of the IEEE International Conference On Neural Networks*, pp. 1942-1948
- Kennedy, J., Eberhart, R. C., Shi, Y. H. (2001) *Swarm Intelligence*, San Mateo, CA: Morgan Kaufmann
- Kennedy, J., Mendes, R. (2002) "Population structure and particle swarm performance" *Proceedings of the 2002 IEEE Congress on Evolutionary Computation*, pp. 1671-1676
- Kim, E., Park, M., Ji, S., Park, M. (1997) "A new approach to fuzzy modeling" *IEEE Transactions on Fuzzy Systems*, vol. 5, no. 3, pp. 328-337
- Kim, M.-S., Kim, C.-H., Lee, J.-J. (2006) "Evolving Compact and Interpretable Takagi-Sugeno Fuzzy Models With a New Encoding Scheme" *IEEE Transactions on Systems, Man, and Cybernetics - Part B: Cybernetics*, vol. 36, no. 5, pp. 1006-1023
- King, B. (1967) "Step-wise clustering procedures" *Journal of the American Statistical Association*, vol. 62, no. 317, pp. 86-101
- Knowles, J. D., Corne, D. W. (2000) "Approximating the Nondominated Front Using the Pareto Archived Evolution Strategy" *Evolutionary Computation*, vol. 8 no. 2 pp. 149-172
- Kohonen, T. (2001) *Self-Organizing Maps*, Springer
- Langdon, W. B., Poli, R. (2002) *Foundations of Genetic Programming*, Springer-Verlag
- Larranaga, P., Lozano, J. A. (2001) *Estimation of Distribution Algorithms: A New Tool for Evolutionary Computation*, Springer
- Lovbjerg, M., Rasmussen, T. K., Krink, T. (2001) "Hybrid particle swarm optimizer with breeding and subpopulations" *Proceedings of the third Genetic and Evolutionary Computation Conference*, pp. 469-476
- Mahfouf, M., Linkens, D. A., Abbod, M.F. (2002) "Optimisation of steel production incorporating economic factors" *Proceedings of the 15th IFAC World Congress on Automatic Control*
- Mahfouf, M., Chen, M.-Y., Zhang, Q., Linkens, D. A. (2006) "Adaptive Weighted Particle Swarm Multiobjective Optimisation and Societal Reasoning for the Design of Alloy Steels" *Proceedings of the 1st IFAC Workshop on Applications of Large Scale Industrial Systems*
- Maksim, (diagram), Available from: <http://upload.wikimedia.org/wikipedia/commons/c/ce/Clusters.PNG>

## Bibliography

- Mamdani, E. H., Assilian, S. (1974) "Applications of fuzzy algorithms for control of simple dynamic plant" *Proceedings of IEE*, vol. 121, 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
- Marsili-Libelli, S., Muller, A. (1996) "Adaptive fuzzy pattern recognition in the anaerobic digestion process" *Pattern Recognition Letter*, vol. 17, no. 6, pp. 651-659
- Michalewicz, Z. (1992) *Genetic Algorithms + Data Structures = Evolution Programs*, Berlin: Springer-Verlag
- Mikut, R., Jakel, J., Groll, L. (2005) "Interpretability issues in data-based learning of fuzzy systems" *Fuzzy Sets and Systems*, vol. 150, pp. 179-197
- Mitchell, M. (1996) *An Introduction to Genetic Algorithms*, Cambridge, MA: MIT Press
- Mostaghim, S., Teich, J. (2003) "Strategies for finding good local guides in multi-objective particle swarm optimization (MOPSO)" *Proceedings of the 2003 IEEE Swarm Intelligent Symposium*, pp. 26-33
- Murata, T., Ishibuchi, H., Tanaka, H. (1996) "Multi-objective genetic algorithm and its applications to flowshop scheduling" *Computers & Industrial Engineering*, vol. 30, no. 4, pp. 957-968
- Murtagh, F. (1983) "A survey of recent advances in hierarchical clustering algorithms which use cluster centers" *Journal of Computation*, vol. 26, pp. 354-359
- Nagy, G. (1968) "State of the art in pattern recognition" *Proceedings of the IEEE*, vol. 56, no. 5, pp. 836-863
- Nauck, D., Klawoon, F., Kruse, R. (1997) *Foundations of Neuro-Fuzzy Systems*, New York: Wiley
- Ng, K. C., Li, Y. (1994) "Design of Sophisticated Fuzzy Logic Controllers Using Genetic Algorithms" *Proceedings of the 3rd IEEE International Conference on Fuzzy Systems*, vol. 3, pp. 1708-1712
- Novák, V. (1999) *Mathematical principles of fuzzy logic*, Dodrecht: Kluwer Academic
- Papadakis, S. E., Theocharis, J. B. (2002) "A GA-based fuzzy modeling approach for generating TSK models" *Fuzzy Sets and Systems*, vol. 131, pp. 121-152



## Bibliography

- Park, D., Kandel, A., Langholz, G. (1994) "Genetic-based New Fuzzy Reasoning Models with Application to Fuzzy Control" *IEEE Transaction of Systems Man Cybernet*, vol. 24, no. 1, pp. 39-47
- Parsopoulos, K. E., Vrahatis, M. N. (2004) "On the computation of all global minimizers through particle swarm optimization" *IEEE Transactions on Evolutionary Computation*, vol. 8, pp. 211-224
- Passino K. M., Yurkovich, S. (1998) *Fuzzy Control*, Menlo Park, CA: Addison Wesley Longman
- Pedrycz, W. (Ed.) (1997) *Fuzzy Evolutionary Computation*, Dordrecht: Kluwer Academic Publishers
- Peram, T., Veeramachaneni, K., Mohan, C. K. (2003) "Fitness-distance-ratio based particle swarm optimization" *Proceedings of the IEEE 2003 Swarm Intelligence Symposium*, pp. 174-181
- Pickering, F. B. (1978) *Physical Metallurgy and the Design of Steels*, London: Applied Science
- Price, K. V., Storn, R. M., Lampinen, J. A. (2005) *Differential evolution: a practical approach to global optimization*, Birkhauser
- Ratnaweera, A., Halgamuge, S. K., Watson, H. C. (2004) "Self-organizing hierarchical particle swarm optimizer with time-varying acceleration coefficients" *IEEE Transactions on Evolutionary Computation*, vol. 8, no. 3, pp. 240-255
- Ray, T. (2002) "Constrained robust optimal design using a multiobjective evolutionary algorithm" *Proceedings of the 2002 IEEE Congress on Evolutionary Computation*, vol. 1, pp. 419-424
- Richards, M., Ventura, D. (2004) "Choosing a starting configuration for particle swarm optimization" *Proceedings of the 2004 IEEE International Joint Conference on Neural Networks*, vol. 3, pp. 2309-2312
- Robinson, J., Rahmat-Samii, Y. (2004) "Particle swarm optimization in electromagnetics" *IEEE Transactions on Antennas Propagation*, vol. 52, no. 2, pp. 397-407
- Rojas, I., Pomares, H., Ortega, J., Prieto, A. (2000) "Self-Organized Fuzzy System Generation from Training Examples" *IEEE Transactions on Fuzzy Systems*, vol. 8, no. 1, pp. 23-36
- Rudolph, G. U., Agapie, A. (2000) "Convergence properties of some multi-objective evolutionary algorithms" *Proceedings of the IEEE Congress on Evolutionary Computation (CEC 2000)*, pp. 1010-1016

## Bibliography

- Rumelhart, D. E., Hinton, G. E., Williams, R. J., (1986) "Learning internal representation by error propagation" *Parallel Distributed Processing*, Cambridge, MA: MIT Press
- Sanchez, E., Shibata, T., Zadeh L. (Eds.) (1997) *Genetic Algorithms and Fuzzy Logic Systems*, Soft Computing Perspectives, Singapore: World Scientific
- Sawaragi, Y., Nakayama, H., Tanino, T. (1985) *Theory of Multiobjective Optimization*, vol. 176 of Mathematics in Science and Engineering, Orlando, FL: Academic Press
- Schwefel H.-P. Rudolph, G. (1995) "Contemporary Evolution Strategies" *Proceedings of the Third European Conference on Advances in Artificial Life*, pp. 893-907
- Setnes, M. (1995) *Fuzzy rule-base simplification using similarity measures*, M.Sc. thesis, Department of Electrical Engineering, Control Laboratory, Delft University Technology, Holland
- Setnes, M., Babuska, R., Kaymak, U., Van Nauta Lemke, H. R. (1998) "Similarity Measures in Fuzzy Rule Base Simplification" *IEEE Transactions on Systems, Man, and Cybernetics - Part B: Cybernetics*, vol. 28, no. 3, pp. 376-386
- Setnes, M., Babuska, R., Verbruggen, B. (1998) "Rule-based modeling: Precision and transparency" *IEEE Transactions on Systems, Man, and Cybernetics - Part C*, vol. 28, no. 1, pp. 165-169
- Setnes, M., Roubos, H. (2000) "GA-fuzzy modeling and classification: complexity and performance" *IEEE Transactions on Fuzzy Systems*, vol. 8, no. 5, pp. 509-522
- Shi, Y. (2004) "Feature article on particle swarm optimization" IEEE Neural Network Society, Feature Article, pp. 8-13
- Shi, Y., Eberhart, R. C. (1998a) "A Modified Particle Swarm Optimizer" *Proceedings of the 1998 IEEE International Conference on Evolutionary Computation*, pages 69-73.
- Shi, Y., Eberhart, R. C. (1998b) "Parameter selection in particle swarm optimization" *Proceedings of the 7th International Conference on Evolutionary Programming*, pp. 591-600
- Shi, Y., Eberhart, R. C. (1999) "Empirical study of particle swarm optimization" *Proceedings of the 1999 IEEE Congress on Evolutionary Computation*, vol. 3, pp. 1945-1950

## Bibliography

- Shi, Y., Eberhart, R. C. (2001) "Fuzzy adaptive particle swarm optimization" *Proceedings of the 2001 IEEE Congress on Evolutionary Computation*, vol. 1, pp. 101-106
- Sideris, S. (diagram) Available from: [http://upload.wikimedia.org/wikipedia/commons/a/ab/Hierarchical\\_clustering\\_diagram.png](http://upload.wikimedia.org/wikipedia/commons/a/ab/Hierarchical_clustering_diagram.png)
- Smith, S. F. (1998) *A Learning System Based on Genetic Adaptive Algorithms*, Doctoral Dissertation, Department of Computer Science, University of Pittsburgh
- Sneath, P. H. A., Sokal, R. R. (1973) *Numerical Taxonomy*, London: Freeman
- Srinivas, N., Deb, K. (1994) "Multiple Objective Optimization with Vector Evaluated Genetic Algorithms" *Evolutionary Computation*, vol. 2, no. 3, pp. 221-248
- Steuer, R. E. (1986) *Multiple Criteria Optimization: Theory, Computations, and Application*, New York: John Wiley & Sons
- Storn, R. (1996) "On the usage of differential evolution for function optimisation" *Proceedings of the 1996 Biennial Conference of the North American, Fuzzy Information Processing Society*, pp. 519-523
- Storn, R., Price, K. (1995) "Differential Evolution – a simple and efficient adaptive scheme for global optimisation over continuous spaces" Technical Report, TR-95-012, ICSI, Berkeley, CA, USA
- Suganthan, P. N. (1999) "Particle swarm optimizer with neighborhood operator" *Proceedings of the 2002 IEEE Congress on Evolutionary Computation*, pp. 1958-1962
- Sugeno, M., Yasukawa, T. (1993) "A Fuzzy-Logic-Based Approach to Qualitative Modeling" *IEEE Transactions on Fuzzy Systems*, vol. 1, no. 1, pp. 7-31
- Takagi, T., Sugeno, M. (1985) "Fuzzy Identification of Systems and Its Applications to Modeling and Control" *IEEE Transaction Systems, Man and Cybernetics*, vol. 15, no. 1, pp. 116-132
- Tenner, J. (1999) "Optimisation of the heat treatment of steel using neural networks" PhD Thesis, The Department of Automatic Control and Systems Engineering, The University of Sheffield, UK
- Tetko, I. V., Villa, A. E., Livingstone, D. J. (1995) "Neural network studies. 1. Comparison of overfitting and overtraining" *Journal of Chemical Information and Computer Sciences*, vol. 35, pp. 826-833

## Bibliography

- Thrift, P., Hayashi, I. (1991) "Fuzzy Logic Synthesis with Genetic Algorithms" *Proceedings of the Fourth International Conference on Genetic Algorithms*, pp. 509-513
- Valenzuela-Rendon, M. (1991) "The Fuzzy Classifier System: Motivations and First Results" *Proceedings of the First International Conference on Parallel Problem Solving from Nature-PPSN I*, pp. 753-758
- Van den Bergh, F., Engelbrecht, A. (2004) "A cooperative approach to particle swarm optimization" *IEEE Transactions on Evolutionary Computation*, vol. 8, no. 3, pp. 225-239
- Velasco, J. R. (1998) "Genetic-based On-line Learning for Fuzzy Process Control" *International Journal of Intelligent System*, vol. 13, no. 10-11, pp. 891-903
- Venturini, G. (1993) "SIA: A Supervised Inductive Algorithm with Genetic Search for Learning Attribute Based Concepts" *Proceedings of the European Conference on Machine Learning*, pp. 280-296
- Wang, H., Kwong, S., Jin, Y., Wei, W., Man, K.F. (2005) "Multi-objective hierarchical genetic algorithm for interpretable fuzzy rule-based knowledge extraction" *Fuzzy Sets and Systems*, vol. 149, pp. 149-186
- Wang, H., Kwong, S., Jin, Y., Wei, W., Man, K.F. (2005) "Agent-Based Evolutionary Approach for Interpretable Rule-Based Knowledge Extraction" *IEEE Transactions on Systems, Man, and Cybernetics - Part C: Applications and Reviews*, vol. 35, no. 2, pp. 143-155
- Wang, L.-X. (1992) "Fuzzy Systems Are Universal Approximators" *Proceedings of the 1<sup>st</sup> IEEE Conference On fuzzy Fuzzy Systems*, pp. 1163-1170
- Wang, L.-X. (1994) *Adaptive Fuzzy Systems and Control*, Englewood Cliffs, NJ: Prentice-Hall
- Wang, L.-X. (1997) *A Course in Fuzzy Systems and Control*, Englewood Cliffs, NJ: Prentice-Hall
- Wang, L.-X., Mendel, J. M. (1992) "Fuzzy basis functions, universal approximation, and orthogonal least-squares learning" *IEEE Transactions on Neural Networks*, vol. 3, no. 5, pp. 807-814
- Wang, L., Yen, J. (1999) "Extracting fuzzy rules for system modeling using a hybrid of genetic algorithms and Kalman filter" *Fuzzy Sets and Systems*, vol. 101, pp. 353-362
- Ward, Jr., J. H. (1963) "Hierarchical grouping to optimize an objective function" *Journal of the American Statistical Association*, vol. 58, no. 301, pp. 236-244

## Bibliography

- Xie, X., Zhang, W., Yang, Z. (2002) "A dissipative particle swarm optimization" *Proceedings of the 2002 IEEE Congress on Evolutionary Computation*, pp. 1456-1461
- Yen, J., Wang, L. (1998) "Application of statistical information criteria for optimal fuzzy model construction" *IEEE Transactions on Fuzzy Systems*, vol. 6, no. 3, pp. 362-372
- Yen, J., Wang, L. (1999) "Simplifying fuzzy rule-based models using orthogonal transformation methods" *IEEE Transactions on Systems, Man, and Cybernetics - Part B: Cybernetics*, vol. 29, no. 1, pp. 13-24
- Yen, J., Wang, L., Gillespie, G. W. (1998) "Improving the interpretability of TSK fuzzy models by combining global learning and local learning" *IEEE Transactions on Fuzzy Systems*, vol. 6, no. 4, pp. 530-537
- Zadeh, L. A. (1965) "Fuzzy sets" *Information and Control*, vol. 8, no. 3, pp. 338-353
- Zadeh, L. A. (1972) "A fuzzy-set-theoretic interpretation of linguistic hedges" *Journal of Cybernetics*, vol. 2, pp. 4-34
- Zadeh, L. A. (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, L. H., Hu, S. (2003) "A new approach to improve particle swarm optimization" *Lecture Notes in Computer Science*, vol. 2723, pp. 134-139
- Zhang, Q., Mahfouf, M. (2005) "Reduced Space Searching (RSS) Algorithm: A New Optimization Algorithm for Single-Objective and Multi-Objective Problems," *Proceedings of the 1st European Symposium on Nature-inspired Smart Information Systems*
- Zhang, Q., Mahfouf, M. (2006) "A New Structure for Particle Swarm Optimization (nPSO) Applicable to Single Objective and Multiobjective Problems" *Proceedings of the 3rd International IEEE Conference on Intelligent Systems*, pp. 176-181
- Zhang, Q., Mahfouf, M. (2007) "A new Reduced Space Searching Algorithm (RSSA) and its application in optimal design of alloy steels" *Proceedings of the IEEE Congress on Evolutionary Computation (CEC 2007)*, pp. 1815-1822
- Zhang, Q., Mahfouf, M. (2007) "Fuzzy predictive modelling using hierarchical clustering and multi-objective optimisation for mechanical properties of alloy steels" *Proceedings of Twelfth IFAC Symposium on Automation in Mining, Mineral and Metal Processing*

## **Bibliography**

- Zhang, Q., Mahfouf, M. (2008) "Mamdani-Type Fuzzy Modelling via Hierarchical Clustering and Multi-Objective Particle Swarm Optimisation (FM-HCPSO)" *International Journal of Computational Intelligence Research (IJCIR)*, Accepted
- Zitzler, E., Deb, K., Thiele, L. (2002) "Comparison of Multiobjective Evolutionary Algorithm: Empirical Results" *Evolutionary Computation*, vol. 8, no. 2, pp. 173-195
- Zitzler, E., Laumanns M., Thiele, L. (2001) "SPEA2: Improving the Strength Pareto Evolutionary Algorithm" Technical Report 103, Zurich, Switzerland: Computer Engineering and Networks Laboratory (TIK), Swiss Federal Institute of Technology (ETH)
- Zitzler, E., Laumanns M., Bleuler, S. (2003) "A Tutorial on Evolutionary Multiobjective Optimization" Workshop on Multiple Objective Metaheuristics, Berlin: Springer-Verlag
- Zitzler E., Thiele, L. (1998) "An Evolutionary Algorithm for Multiobjective Optimization: The Strength Pareto Approach" Technical Report 43, Computer Engineering and Networks Laboratory (TIK), Swiss Federal Institute of Technology (ETH), Swiss