

System Architecture Design Using Multi-Criteria Optimization



The
University
Of
Sheffield.

Rajesh Kudikala

Department of Automatic Control and Systems Engineering

The University of Sheffield

A thesis submitted for the degree of

Doctor of Philosophy

March, 2015

I would like to dedicate this thesis to my beloved parents
Satyanarayana and *Bhagya Lakshmi*
and to my beloved wife *Kiranmai*.

Acknowledgements

I would like to express my deepest and sincere gratitude to my supervisor, Professor Peter J Fleming, for the help and support he has given me throughout my Ph.D. research. I feel very privileged to have him as my supervisor. This thesis would have not been possible without his vision, motivation, guidance and supervision. I am very thankful to him for his time, support, patience and encouragement.

I would like to take this opportunity to thank Rolls-Royce plc and EPSRC for the financial support through a Dorothy Hodgkin Post-graduate Award (DHPA).

I would also like to express my sincere thanks to system engineering experts from Rolls-Royce plc, Graham Tanner, Jonathan Holt, Dr. Ian Griffin, Dr. Vijay Patel, Adam McLoughlin, and Peter Beecroft for providing necessary models and information needed for my research work and for their reviews and support.

I am very thankful to Professor Haydn A Thompson, Professor Visakan Kadirkamanathan, Dr. Andrew R Mills, Dr. Robin C Purshouse, Dr. Korak Gosh and Dr. Ioannis Giagkiozis for their invaluable advice and comments throughout my Ph.D. research work. I am extremely grateful to all the people that I have had the pleasure of working with in Rolls-Royce UTC, Rui, Vignesh, Martha, Maszatul, Renata, Mathew and to all my friends, who have been very helpful and friendly to me throughout the time of my study. During my research work I received many helps from colleagues and staff at the University of Sheffield. I thank the University of Sheffield for giving me this great opportunity. I feel proud to have been part of this excellent institution.

I want to express my sincere thanks to Dr. Andrew J Chipperfield and Professor Tony Dodd for their positive comments and corrections in my thesis.

I would also like to express my thanks to Professor Kalyanmoy Deb and Professor Bishakh Bhattacharya from IIT Kanpur for inspiring me to pursue Ph.D. research study, and to Dr. Chandra Shekhar Nyalapogula for providing a great encouragement during my Ph.D.

Many thanks to my friends Ramesh, Rajesh, Purna, Ravikiran, Somu, Chandu, Pawan, RamaRao, Roopa, Pradeep, Ravi, Dattu, Sujan, Chenna, Ravitej, Sunith, Eashwar, Anil, Monalisa, Rathod, Lucky and Deva. I feel very lucky to have them in my life. Special thanks to Rajendra and Keerthi, who helped me and made my stay in Sheffield so enjoyable. I would also like to express my gratitude to Rajesh, Vishnuji and members of JET UK for sharing with me, their invaluable insight into Vedic principles and ethical values.

Deeply from my heart with love, I would like to thank my beloved parents Satyanarayana and Bhagya Lakshmi for their encouragement and prayers throughout my life. My dearest love and thanks to my wife Kiranmai for her patience, support, love and for being there in my life. I could not have wished for a more understanding and more loving wife than her. My profound thanks to my dear parents-in-law Sudhakar, Lalitha, my dear brother Harish and brother-in-law Aashish for their unconditional support. I am truly indebted to all their patience, support and understanding. Thank you all very much!

All praise and glory is due to Almighty Lord Sri Laxmi Narayana, the lord of the universe, whose mercy and blessings have been bestowed constantly upon me, along with mangalashasanams from Acharya His Holiness Sri Sri Sri Tridandi Srimannarayana Ramanuja Chinna Jeeyar Swamiji.

“Serve all beings as service to God. Worship your own..Respect all.”
..HH Sri Chinna Jeeyar Swamiji.

Abstract

System architecture is defined as the description of a complex system in terms of its functional requirements, physical elements and their interrelationships. Designing a complex system architecture can be a difficult task involving multi-faceted trade-off decisions. The system architecture designs often have many project-specific goals involving mix of quantitative and qualitative criteria and a large design trade space. Several tools and methods have been developed to support the system architecture design process in the last few decades. However, many conventional problem solving techniques face difficulties in dealing with complex system design problems having many goals.

In this research work, an interactive multi-criteria design optimization framework is proposed for solving many-objective system architecture design problems and generating a well distributed set of Pareto optimal solutions for these problems. The optimization framework supports the decision maker by providing a facility for progressive preference articulation, empowering closely coupled user and optimization process interaction. A novel Pareto estimation (PE) method is introduced for increasing the density of the available Pareto optimal solutions with reduced computational expense and further extended to multi-objective multi-modal problems.

System architecture design using multi-criteria optimization is demonstrated using a real-world application of an aero engine health management (EHM) system. A design process is presented for the optimal deployment of the EHM system functional operations over physical architecture subsystems. The EHM system architecture design problem

is formulated as a multi-criteria optimization problem. The proposed methodology successfully generates a well distributed family of Pareto optimal architecture solutions for the EHM system, which provides valuable insights into the design trade-offs. Uncertainty analysis is implemented using an efficient polynomial chaos approach and robust architecture solutions are obtained for the EHM system architecture design. Performance assessment through evaluation of benchmark test metrics demonstrates the superior performance of the proposed methodology.

Contents

Contents	vi
List of Figures	x
List of Tables	xiv
Nomenclature	xviii
1 Introduction	1
1.1 Introduction and Motivation	1
1.2 Research Objectives	3
1.3 Outline of the Thesis	4
1.4 Research Contributions	7
1.5 Publications and Research Reports	8
2 Literature Review	10
2.1 Introduction	10
2.2 System Architecture Design	10
2.3 Introduction to Multi-Objective Optimization (MOO)	11
2.3.1 Goals of Multi-Objective Optimization	13
2.3.2 General Optimization Process	15
2.3.3 Classical Optimization Approaches	16
2.3.4 Evolutionary Computation Methods	17
2.4 Review of System Architecture Design using Optimization Methods	18
2.5 Many-Objective Optimization	23

2.5.1	Background	23
2.5.2	Methods for Increasing Selective Pressure	24
2.5.2.1	Modification of Pareto-dominance	24
2.5.2.2	Assignment of different ranks	25
2.5.3	Indicator Based Methods	26
2.5.4	Decomposition Based Methods	27
2.5.5	Preference Articulation Methods	27
2.5.5.1	<i>A-priori</i> Preference Articulation	27
2.5.5.2	<i>A-posteriori</i> Preference Articulation	28
2.5.5.3	Progressive Preference Articulation	28
2.5.6	A Co-Evolutionary Approach using Preferences	29
2.5.7	Multi-Objective Multi-Modal Optimization	29
2.5.8	Visualization Methods	31
2.5.8.1	The Cartesian Co-ordinates Plot and Scatter-Plot	32
2.5.8.2	The Parallel Co-ordinates Plot	32
2.6	Uncertainty Analysis and Robust Design Optimization	33
2.7	Research Gaps	33
2.8	Summary	36
3	Interactive Multi-Criteria Design Optimization Framework	37
3.1	Introduction	37
3.2	Interactive Multi-Criteria Design Optimization Framework	38
3.3	Pareto Estimation Method	44
3.3.1	Motivation	44
3.3.2	Radial Basis Function Neural Networks	45
3.3.3	Pareto Estimation - General Procedure	47
3.3.4	Limitations of Pareto Estimation Method	49
3.4	Extended Pareto Estimation Method with Clustering	50
3.5	Visual Assessment of Cluster Tendency (VAT) Algorithm	52
3.6	Experimental Setup	55
3.6.1	Test problem 1	56
3.6.1.1	CASE I	56
3.6.1.2	CASE II	60

3.6.2	Test problem 2	63
3.7	Summary	70
4	System Architecture Design: An	
	Aero Engine Health Management System Case Study	72
4.1	Introduction	72
4.2	Aero Engine Health Management System Architecture Design . .	73
4.2.1	Stakeholder Identification and Requirement	
	Analysis	74
4.2.2	EHM System Functional Decomposition	76
4.2.3	EHM System Physical Architecture Decomposition	78
4.2.4	EHM System Physical Architecture	
	Subsystems Limitations	80
4.2.5	EHM System Functional Operations Deployment	82
4.3	Multi-Criteria Optimization Formulation for	
	EHM System Architecture Design Problem	83
4.3.1	Decision Variables	84
4.3.2	Constraints	84
4.3.3	Objective Functions	85
4.3.4	Optimization Problem Formulation	85
4.3.5	Integration of the Architecture Models into	
	the Optimization Framework	86
4.3.6	Evaluation of Criteria Values for a Sample EHM System	
	Deployment Solution	87
4.4	Multi-Criteria Optimization of EHM	
	System Architecture Design Problem	90
4.5	Cluster Analysis and Pareto Estimation	94
4.6	Statistical Performance Evaluation	98
4.7	Exploring What-If Design Scenarios	106
4.8	Summary	107
5	Uncertainty Analysis and Robust Optimization of EHM System	
	Architecture Design	108

CONTENTS

5.1	Introduction	108
5.2	Uncertainties in EHM System Architecture Design	109
5.3	Robustness Metric	110
5.4	Polynomial Chaos Expansion Approach	111
5.5	Uncertainty Analysis and Robust Optimization of EHM System Architecture Design	113
5.6	Validation of the EHM System Architecture Design Optimization	118
5.7	Summary	125
6	Conclusions and Future Work	126
6.1	Research Objectives	126
6.2	Research Contributions	127
6.3	Future Work	132
	References	134

List of Figures

2.1	Pareto optimal solutions on a Pareto Front for a two objective minimization problem.	13
2.2	Goals of a multi-objective optimization algorithm.	14
2.3	Components of an optimization problem.	15
2.4	Regions of non-domination in a two-objective minimization problem.	23
2.5	Multi-modal Pareto optimal solutions in a multi-objective optimization problem.	30
3.1	Ranking of solutions with (i)an unattainble goal and (ii)an attainble goal, using the progressive preference articulation PPA _{FF} (Fonseca & Fleming, 1998a) technique.	39
3.2	Flowchart of multi-objective genetic algorithm (MOGA) with PPA _{FF} and Crowding distance operator.	41
3.3	Interactive optimization framework with progressive preference articulation technique.	42
3.4	MOGA trade-off graph with progressive preference articulation window.	44
3.5	The Pareto estimation method (Giagkiozis & Fleming, 2012). . .	47
3.6	The normalization and projection of a two-objective Pareto front (Giagkiozis & Fleming, 2012).	48
3.7	A many to one mapping of decision vectors to Pareto front in multi-objective multi-modal problems (Kudikala <i>et al.</i> , 2013b). . .	50
3.8	The extended Pareto estimation method. Clusters the Pareto optimal decision variable vectors and identifies a mapping for every cluster (Kudikala <i>et al.</i> , 2013b).	51

LIST OF FIGURES

3.9	Flowchart of the proposed interactive multi-criteria design optimization framework incorporating the MOGA optimizer, the clustering analysis and the Pareto estimation method.	54
3.10	Case I: Non-dominated solutions obtained for the test problem 1 (equation 3.10) with 2 variables using NSGA-II, Omni-Optimizer, MOGA and RANDOM search methods.	57
3.11	Case I: Non-dominated solutions obtained for the test problem 1 (equation 3.10) with 2 variables using the MOGA optimizer (top) and the Pareto estimation method (bottom) in objective space, decision variable space and image of clusters.	58
3.12	Case II: Non-dominated solutions obtained for the test problem (equation 3.10) with 3 variables using NSGA-II, Omni-Optimizer, MOGA and RANDOM search methods.	61
3.13	Case II: Non-dominated solutions obtained for the test problem (equation 3.10) with 3 variables using the MOGA optimizer (top) and the Pareto estimation method (bottom) in objective space, decision variable space and image of clusters.	62
3.14	Non-dominated solutions obtained for the DTLZ3 test problem with 3 objectives from NSGA-II, Omni-Optimizer, MOGA and RANDOM search methods.	65
3.15	Non-dominated solutions obtained for the DTLZ3 test problem with 4 objectives from NSGA-II, Omni-Optimizer, MOGA and RANDOM search methods.	66
3.16	Non-dominated solutions obtained for the DTLZ3 test problem with 6 objectives from NSGA-II, Omni-Optimizer, MOGA and RANDOM search methods.	67
3.17	Non-dominated solutions obtained for DTLZ3 problem with 3 objectives using the MOGA optimizer (top) and the Pareto estimation methods (bottom) in objective space, decision variable space and image of clusters.	68

LIST OF FIGURES

3.18	Non-dominated solutions obtained for DTLZ3 problem with 4 objectives using the MOGA optimizer (top) and the Pareto estimation methods (bottom) in objective space, decision variable space and image of clusters.	68
3.19	Non-dominated solutions obtained for DTLZ3 problem with 6 objectives using the MOGA optimizer (top) and the Pareto estimation methods (bottom) in objective space, decision variable space and image of clusters.	69
4.1	Aero Engine Health Management System.	73
4.2	Stakeholders and use-cases identified in EHM system design (Tanner 2010).	75
4.3	EHM system functional decomposition and operational attributes.	76
4.4	EHM system functions and functional operations.	77
4.5	EHM system physical architecture decomposition and subsystems.	79
4.6	EHM system physical architecture subsystem limitations elicited from interviews with GTE System experts.	80
4.7	EHM system architecture multi-criteria optimization problem components: Decision variables, OPs Deployment Model, Constraints and Objective functions.	83
4.8	Integrating the system architecture SysML model to optimization platform.	86
4.9	Evaluation of constraints and objective functions values for a sample solution.	88
4.10	EHM system architecture solutions in MOGA parallel coordinates trade-off graph with preference articulation.	91
4.11	VAT clusters image with four clusters (dark blocks) in EHM system non-dominated solutions.	94
4.12	Clusters of non-dominated solutions after the Pareto estimation for EHM system architecture design.	96
4.13	VAT Image of clusters of non-dominated solutions after the Pareto estimation for EHM system architecture design.	96

LIST OF FIGURES

4.14	Comparison of non-dominated solutions obtained from NSGA-II, Omni-Optimizer, MOGA and RANDOM search methods.	99
4.15	Box plots of the (i) HV and (ii) IGD metric values computed for the solutions obtained in 25 runs by the optimizers.	103
4.16	Box plots of the (i) S_R and (ii) C -metric values computed for the solutions obtained in 25 runs by the optimizers.	103
4.17	Trade-off solutions with increased dataflow rate to the on-ground station.	105
4.18	Trade-off solutions with increased processor limit on the EMU.	105
4.19	Optimal deployment solution with four objectives satisfied.	105
5.1	EHM system architecture Pareto optimal solutions with robustness metric values.	115
5.2	Robust EHM system architecture Pareto optimal solutions obtained by expressing preferences on criticality and robustness metric objectives.	116
5.3	Criteria scores for robust EHM system architecture Pareto optimal solutions.	117
5.4	Robust solution 1 EHM system architecture with functional deployment.	119
5.5	Robust solution 2 EHM system architecture with functional deployment.	120
5.6	Baseline EHM system architecture with functional deployment.	121
5.7	Criteria scores for the baseline EHM system architecture.	122
5.8	Baseline EHM system architecture criteria scores on MOGA parallel co-ordinates graph.	122
5.9	Comparison of Robust EHM system architecture solutions obtained from the optimization process with the baseline EHM system architecture.	122

List of Tables

4.1	Hypervolume (HV) and $D(A, \mathcal{P}^*)$ values of the obtained solutions by the optimizers, where A is the obtained set in each run, \mathcal{P}^* is the considered PF.	102
4.2	$S_R(A, \mathcal{P}^*)$ and C -Metric values of the solutions obtained by the optimizers, where A is the obtained set in each run, \mathcal{P}^* is the considered PF.	102

Nomenclature

Symbols

A, B	Sets of Non-dominated Solutions Obtained from an Optimizer
α	Coefficients of Polynomial Chaos Expansions
C	C-metric Value
C_m	Clusters of Multi-Modal Decision Vectors
\mathcal{D}	Pareto Optimal Decision Vectors
\mathcal{D}_ε	Estimated Decision Vectors from Pareto Estimation Method
\bar{d}_h	Mean Distance of Solutions to their Nearest Neighbour
D_r	Data Flow Rate Resource Constraints
E	Attribute Excess Requirement Objectives for EHM System
ε	Uniformly Distributed Points on $(k - 1)$ -Dimensional Simplex
\mathbf{f}	Vector of Objective Functions
h	Order of Polynomial Chaos Expansion
k	Number of Objectives
μ	Mean
N	Number of Samples in Polynomial Chaos Expansion
\mathcal{P}	Pareto Optimal Objective Vectors
p	Uncertain Design Parameter
P_E	Projection Matrix
ϕ_i	Basis Function in Radial Basis Function Neural Networks
π	$\simeq 3.14 \dots$
$\tilde{\mathcal{P}}$	Transformed Pareto Optimal Objective Vectors
$P_i(x)$	Orthogonal Polynomials
P_s	Size of Pareto Set
P_r	Processing Resource Constraints

R_F	Robustness Metric
S	Feasible Set in Decision Space
σ	Standard Deviation
S_R	Mean Neighbourhood Distance Metric
u	Number of Uncertain Parameters
w	Weights of Neurons in Neural Networks
\mathbf{x}, \mathbf{y}	Decision Variable Vectors
Z	Feasible Set in Objective Space
$\bar{\mathbf{z}}$	Dominated Reference Objective Vector
\mathbf{z}^{nd}	Nadir Objective Vector
\mathbf{z}^*	Ideal Objective Vector

Acronyms

<i>ACARS</i>	Aircraft Communications, Addressing and Reporting System
<i>AEDN</i>	Architecture Design Environment
<i>BB</i>	Broadband
<i>DACS</i>	Data Accumulation and Counting System
<i>DCS</i>	Distributed Control System
<i>DHPA</i>	Dorothy Hodgkin Postgraduate Award
<i>DM</i>	Decision Maker
<i>DTLZ</i>	Deb, Thiele, Laumanns and Zizler Test Problems
<i>EA</i>	Enterprise Architect software
<i>EAs</i>	Evolutionary Algorithms
<i>EEC</i>	Engine Electronic Controller
<i>EHM</i>	Engine Health Management
<i>EMO</i>	Evolutionary Multi-objective Optimization
<i>EMU</i>	Engine Monitoring Unit
<i>FD</i>	Feature Detector
<i>FFT</i>	Fast Fourier Transform
<i>GAs</i>	Genetic Algorithms
<i>GTE</i>	Gas Turbine Engine
<i>HV</i>	Hypervolume
<i>HypE</i>	Hypervolume Based Evolutionary Algorithm
<i>IBEA</i>	Indicator Based Evolutionary Algorithm

NOMENCLATURE

<i>IGD</i>	Inverted Generational Distance
<i>IP</i>	Intellectual Property
<i>LRU</i>	Line Replaceable Units
<i>MATE</i>	Multi Attribute Trade Space Exploration
<i>MC</i>	Monte Carlo Method
<i>MCDM</i>	Multi-Criteria Decision Making
<i>MFLOPS</i>	Mega Floating-point Operations Per Second
<i>MOEA</i>	Multi-Objective Evolutionary Algorithm
<i>MOEA/D</i>	Multi-objective Evolutionary Algorithm based on Decomposition
<i>MOGA</i>	Multi-Objective Genetic Algorithm
<i>MOGAC</i>	Multi-Objective Genetic Algorithm for Hardware Software Co-synthesis
<i>MOO</i>	Multi-Objective Optimization
<i>MSOPS</i>	Multiple Single Objective Pareto Sampling
<i>NN</i>	Neural Network
<i>NPV</i>	Net Present Value
<i>NSGA</i>	Non-Dominated Sorting Genetic Algorithm
<i>NSGAI</i>	Non-Dominated Sorting Genetic Algorithm - II
<i>ODMS</i>	Oil Debris Monitoring System
<i>OEM</i>	Original Equipment Manufacturer
<i>OPs</i>	Operations
<i>PCE</i>	Polynomial Chaos Expansions
<i>PCM</i>	Palladio Component Model
<i>PE</i>	Pareto Estimation
<i>PICEAs</i>	Preference Inspired Co-Evolutionary Algorithms
<i>PPA</i>	Progressive Preference Articulation
<i>PPA_{FF}</i>	Progressive Preference Articulation method of Fonseca and Fleming (1998a)
<i>QFD</i>	Quality Function Deployment
<i>RBFNN</i>	Radial Basis Function Neural Network
<i>ROI</i>	Region of Interest
<i>SPEA2</i>	Strength Pareto Evolutionary Algorithm 2
<i>SysML</i>	System Modelling Language
<i>TO</i>	Tracked Order

NOMENCLATURE

<i>UTC</i>	University Technology Centre
<i>VAT</i>	Visual Assessment of Cluster Tendency Method
<i>VEGA</i>	Vector Evaluated Genetic Algorithm
<i>VOES</i>	Vector Optimized Evolutionary Algorithm
<i>XML</i>	Extensible Mark-up Language

Chapter 1

Introduction

1.1 Introduction and Motivation

This chapter provides a brief introduction followed by motivation for the research work presented in this thesis.

System architecture is defined as the description of a complex system in terms of its functional requirements, physical elements, and their interrelationships. System architecture design is concerned with exploring the trade space of early, high-level, system design decisions. System architecture designs often have many goals: performance, reliability, cost, flexibility, security. Design requirements also include safety, functional and non-functional requirements, mix of qualitative and quantitative objectives, project-specific goals and through-life costs. A large number of stakeholders and design experts are involved in the design process.

Since the goal of a system engineer is to find the best possible system architecture, the architecture design process can be formulated as an optimization problem, where the variables are the alternative architectural options, and the objective functions capture design requirements and project goals. Constraints can also be added to capture the various resource limitations. In the last few years, several tools and methods have been developed to support the system architecture design process. Unfortunately many conventional problem solving

techniques have difficulty when dealing with complex system design problems having many goals. In particular, optimization problems that appear in system architecture design are typically very hard to solve because:

- (1) they are non-convex problems and they have multiple local and global optimal solutions.
- (2) they are integer, mixed-integer, or most often combinatorial problems.
- (3) they are non-linear, both in the objective functions and in the constraints.
- (4) they may be computationally expensive and large-scale high dimensional problems.

Many real-world engineering problems often have multiple competing objectives to be optimized within the constraints of their resource limitations (e.g. Fleming *et al.*, 2005). For these multi-objective optimization problems, there is no one single optimal solution. Instead, there is a family of solutions, where each solution represents a compromise, or trade-off, between the competing objectives, and which cannot be improved upon with respect to all objectives. Historically, multi-objective optimization problems were solved using classical optimization methods, e.g. gradient based weighted sum approaches. In these methods the multiple objectives were aggregated to form a single objective using weights. However, these approaches have several weaknesses. They are very sensitive to the choice of weights and starting point. They have tendency to get stuck in local optima. They require multiple runs to generate a family of trade-off solutions as required for a multi-objective optimization (Deb, 2001).

In recent times, Evolutionary Algorithms (EAs) have been developed to solve multi-objective optimization problems. EAs work with a population of solutions and mimic the principles of biological evolution, i.e., survival of the fittest. They can search for global optima without getting stuck in local optima and find a diverse set of optimal trade-off solutions called a Pareto optimal front in a single run of the optimization algorithm. EAs have mainly been applied to optimization problems having single, two or three objective functions. The Pareto optimal solutions can be visualized using simple scatter plots to assist in selecting a preferred

solution for these problems. However, many real-world engineering problems can comprise significantly more than three objective functions to be optimized. These are called many-objective optimization problems. EAs face several difficulties and challenges for solving many-objective optimization problems, most importantly the inefficient search and convergence of solutions towards the Pareto front. The optimal trade-off surface of a multi-objective optimization problem can contain a potentially infinite number of Pareto-optimal solutions. For solving many-objective optimization problems, EAs require large population sizes which lead to increased computational expense. Visualization of the high dimensional Pareto optimal solutions for many-objective functions becomes difficult (Fleming *et al.*, 2005). In the case of multi-modal multi-objective problems, there exist multiple local and global optima. EAs have issues in finding multi-modal Pareto sets and fail to maintain diversity in both objective and decision variable space. Finding these multi-modal Pareto sets would give the decision maker a greater choice when choosing between optimal solutions.

This thesis addresses this research gap in system architecture design by proposing an interactive multi-criteria design optimization framework which can solve many-objective optimization problems and find diversified multi-modal Pareto optimal solutions. The optimization framework supports the decision maker by providing a facility for progressive preference articulation, empowering closely coupled user and optimization process interaction. The proposed methodology is applied to several real-world system architecture design applications related to aero gas turbine engines.

1.2 Research Objectives

The main goals of this research work are:

- To develop a methodology for system architecture design that addresses multiple design criteria, and
- To demonstrate the application of the methodology on a real-world system architecture design.

In order to achieve the goals, this research work has accomplished the following objectives:

- Development of a multi-criteria design optimization framework, which can solve many-objective system architecture design problems and facilitate a decision maker to interact with the optimization process and express design preferences.
- Development of a methodology for increasing the density of optimal solutions in a decision maker's (DM's) preferred region in the objective space with reduced computational cost.
- Application of multi-criteria design optimization framework to many-objective real-world system architecture design problem.
- Investigate different groups of solutions available in the obtained Pareto optimal solutions using clustering analysis revealing important design features for the system architecture design.
- Perform uncertainty analysis for the real-world system architecture design problem, considering the uncertainties involved in the architecture design process in order to find less sensitive robust optimal system architecture solutions.

1.3 Outline of the Thesis

The previous section outlined the overall objectives of the research. Next, the structure of the thesis is presented.

Chapter 2 provides a critical review of the literature relevant to this research. An introduction to the system architecture design process is presented. A brief history of the development of the evolutionary computing and multi-objective optimization techniques is given. Based on the review of the literature, weaknesses of existing approaches in solving many-objective optimization problems are summarized and research opportunities are discussed.

In Chapter 3, an interactive multi-criteria design optimization framework for solving many-objective optimization problems is proposed. The framework uses a multi-objective genetic algorithm with a unique progressive preference articulation technique. The optimization algorithm is further enhanced with archiving and crowding distance operators to maintain the diversity in both objective and decision vector spaces in order to find multi-modal Pareto optimal solutions. A novel Pareto estimation method for increasing the available Pareto solutions is presented. The method uses an efficient clustering algorithm and neural networks to estimate a large number of multi-modal global optimal solutions with reduced computation cost. A rigorous performance evaluation of the proposed methodology is described by solving a variety of multi-objective test problems that are commonly used in the literature.

In Chapter 4, the system architecture design process is illustrated with a case study of an aero gas turbine engine health management (EHM) system. Decomposition of the EHM system functional operations and physical architecture subsystems limitations is presented. The system architecture design problem involving deployment of EHM system functional operations onto physical architecture components nodes is formulated as a multi-criteria optimization problem. Functional operations need to be deployed in order to satisfy operational attribute requirements within the constraints of resource limitations. Excess requirements, in terms of operational attributes, are considered as multiple objective functions/criteria to be minimized in the optimization process. Challenges in solving the system architecture design optimization problem are discussed. The proposed methodology is applied to the EHM system architecture design problem and optimized with respect to many qualitative criteria in terms of operational attributes within the constraints of resource limitations. The optimization algorithm has produced a family of Pareto solutions which provide valuable insight into design trade-offs. Using the progressive preference articulation technique, the optimization search is focused for the industrial decision maker, onto a region of interest in the objective space. Using this approach it was possible to identify the most significant design constraints (“hot spots”) and the opportunities afforded by either the relaxation or the tightening of these constraints, along with

their performance implications. Performance of the optimization framework is assessed by evaluating test metrics and comparing with popular multi-objective optimization algorithms using established statistical methods.

Chapter 5 presents an uncertainty analysis performed for the EHM system architecture design using the polynomial chaos expansion method for estimating the sensitivities of objectives. In the EHM architecture design study, in order to integrate the EHM architecture models into the optimization platform and to evaluate the various objective functions, the qualitative functional attribute requirements are transformed to suitable numeric values. The obtained Pareto optimal solutions are sensitive with respect to the chosen numerical parameter values. An efficient uncertainty quantification technique using polynomial chaos expansions is integrated within the optimization process. The approach generates a design of experiments with respect to the parameter uncertainties. A robustness metric is calculated using the variations of the criteria values with respect to variations in the parameter values. A parallel processing facility available in MATLAB is utilized for evaluating the robustness metric for each candidate architecture solution over the number of samples from the design of experiments. In the optimization framework using, progressive preference articulation, the decision maker can specify the level of sensitivity that can be allowed in the design. Finally, a set of robust non-dominated solutions are obtained using the proposed approach for the EHM system architecture design. The EHM system architecture design process using the proposed multi-criteria optimization methodology is further validated by comparing with a baseline aero engine EHM system architecture. This demonstrates that the proposed multi-criteria optimization approach for system architecture design is successful in obtaining the best Pareto optimal architecture solutions for the EHM system architecture.

Finally, Chapter 6 provides conclusions and summarizes the work presented in the thesis and the results obtained. Some advantages and shortcomings of the methods described in the thesis are discussed and opportunities for future work are highlighted.

1.4 Research Contributions

This thesis describes the following research contributions:

- 1. Development of a methodology for system architecture design** using a multi-criteria optimization formulation with many conflicting objective functions and constraints. Integration of system models into an optimization framework and evaluation of architecture alternatives using attributes information. Investigation of clusters of solutions resulted in identifying the most significant design features and constraints and the opportunities afforded by either the relaxation or the tightening of these constraints along with their performance implications. This contribution is demonstrated in Chapter 4 of the thesis by applying developed methodology on a real-world aero engine health management system architecture design.
- 2. An interactive multi-criteria design optimization framework** with a progressive preference articulation technique and a crowding distance operator which can handle many-objective optimization problems and maintain diversity in objective space and decision vectors space in order to find diverse sets of multi-modal Pareto optimal solutions. This contribution is presented in Chapter 3 of the thesis and it is used for solving several test problems and a real-world aero engine health management system architecture design. Evaluation of the performance test metrics demonstrated the superior performance of the proposed methodology with good convergence, diversity and repeatability.
- 3. A novel methodology to find the mapping between Pareto-optimal objective vectors and decision vectors** using neural networks and an efficient clustering algorithm, which has been proven useful in increasing the density of multi-modal Pareto solutions in the preferred regions to the decision maker with reduced computation effort. The contribution is presented in Chapter 3 of the thesis and it is used for estimating a large number of multi-modal Pareto solutions for several test problems and a real-world aero engine health management system architecture design.

- 4. Sensitivity analysis approach to system architecture design** which takes into account the uncertainties involved in parameter assumptions and expert judgements for criteria evaluations of the system architecture alternatives in the optimization process. An efficient uncertainty quantification technique using polynomial chaos expansions is integrated into the optimization process. The polynomial chaos expansions technique generates a design of experiments with respect to parameter uncertainties and computes the robustness metrics with lower computational cost compared to standard approaches such as Monte Carlo sampling technique. This contribution is presented in Chapter 5 of the thesis and it is applied finding robust solutions for a real-world aero engine health management system architecture design. A validation process using a baseline system demonstrates success of the proposed methodology for the system architecture design.

1.5 Publications and Research Reports

Material from this thesis has formed the basis for the following Rolls-Royce University Technology Centre (UTC) research reports and international conference publications:

1. Kudikala R., Mills A.R. and Fleming P.J. System Architecture Design of an Aero Engine Health Management (EHM) System. *Rolls-Royce University Technology Centre (UTC) Research Report*, Number: RRUTC/Shef/R/10602, Issue 1, 2010.
2. Kudikala R., Mills A.R. and Fleming P.J. Multi-Criteria System Architecture Design Optimization: an Engine Health Management (EHM) System Case Study. *Rolls-Royce University Technology Centre (UTC) Research Report*, Number: RRUTC/Shef/R/10902, Issue 1, 2010.
3. Kudikala R., Mills A.R. and Fleming P.J. A Proposed Interactive Multi-Criteria Optimization Framework for Gas Turbine Engine System Architecture Design Optimization. *Rolls-Royce University Technology Centre (UTC) Research Report*, Number: RRUTC/Shef/R/11601, Issue 1, 2011.

4. Kudikala R., Mills A.R., Fleming P.J., Tanner G.F. and Holt J.E. An Engine Health Monitoring System Architecture Design using Multi-Criteria Optimization Techniques. *International Workshop on Health Monitoring Systems and Multi-Criteria Optimization (sponsored by UK-India Education and Research Initiative (UKIERI))*, Indian Institute of Technology Kanpur, India, 21-34, 2012.
5. Kudikala R., Giagkiozis I. and Fleming P.J. Increasing the Density of Multi-Objective Multi-Modal Solutions using Clustering and Pareto Estimation Techniques. In *Proceedings of the 10th International Conference on Genetic and Evolutionary Methods GEM 2013*, Las Vegas, USA, 10-16, 2013.
6. Kudikala R., Giagkiozis I. and Fleming P.J. Estimation of Pareto Optimal Solutions for Multi-Objective Multi-Modal Problems. In *Proceedings of 19th International Conference on Soft Computing - MENDEL2013*, Brno, Czech Republic, 25-30, 2013.
7. Kudikala R., Mills A.R., Fleming P.J., Tanner G.F. and Holt J.E. An Aero Engine Health Management System Architecture Design Using Multi-Criteria Optimization. In *Proceedings of the Genetic and Evolutionary Computation Conference - GECCO 2013*, ACM, Amsterdam, The Netherlands, 180-186, 2013.
8. Kudikala R., Mills A.R., Fleming P.J., Tanner G.F. and Holt J.E. Real World System Architecture Design Using Multi-Criteria Optimization: A Case Study. *Advances in Intelligent Systems and Computing, EVOLVE - A Bridge between Probability Set Oriented Numerics and Evolutionary Computation IV, LEIDEN, The Netherlands, Springer, Vol. 227, pp 245-260, 2013.*

Chapter 2

Literature Review

2.1 Introduction

The purpose of this chapter is to review the relevant technical literature to establish the ‘state-of-the-art’ and in particular, to identify any existing weaknesses. This will provide a reference framework against which the value of the research can be assessed.

In this Chapter, initially a definition of system architecture is presented and the process of system architecture design optimization is described. An introduction to multi-objective optimization and the goals of a multi-objective optimizer are presented. In the following section a review of the literature relevant to the architecture design optimization methods is presented. Then there is a discussion of current issues in high dimensional or many-objective optimization. This is followed by sections describing methods for addressing the many-objective optimization problems that arise in system architecture design.

2.2 System Architecture Design

System architecture is defined as “an abstract description of the entities of a system and the relationships between those entities, where the entities could be physical components or functions of the system” by Crawley *et al.* (2004). The

architecture of a complex system can be described in terms of its functional requirements, physical elements, and element interrelationships. The architecture of a system represents a blueprint of the overall system or its highest level of design. Designing a complex system architecture is a difficult task involving multi-faceted trade-off decisions. The design process often needs to consider experience, models and data from many design disciplines.

System architecture design is generally described by large multidisciplinary design phases. A typical architecture design process starts by identifying the main functional requirements and follows a process of decomposition. The top level system functional requirements (use cases) are divided into several sub-functions. The physical form of the system is also divided into sub-systems and components. Designers try to map the functionalities onto the physical hardware components. Designers then iterate between the upper and lower levels of the system decomposition to optimize the deployment of functional operations onto physical systems. However, due to the large and discontinuous design search space and the many qualitative and quantitative criteria or objectives, it is difficult for the designer to obtain optimal architecture designs. Designers have considered various optimization methods and tools for exploring the trade-offs in the system architecture design space.

2.3 Introduction to Multi-Objective Optimization (MOO)

Real-world engineering problems often involve several competing objectives and constraints (e.g. Fleming *et al.*, 2005). When there are conflicting objectives improving one objective will lead to the worsening of one or more of remaining objectives. Thus, for problems with multiple conflicting objectives, there is no one single optimal solution. Instead, there is a family of solutions, where each solution represents a compromise, or trade-off, between the competing objectives, and which cannot be improved upon with respect to all objectives. To distinguish these solutions from inferior ones, many multi-objective optimization algorithms

make use of the principle of Pareto-dominance, which is initially introduced by Edgeworth (1881) and further studied by Pareto (1896).

A general definition of a multi-objective optimization problem is defined as:

$$\begin{aligned} \min_{\mathbf{x}} \quad & \mathbf{f}(\mathbf{x}) = (f_1(\mathbf{x}), f_2(\mathbf{x}), \dots, f_k(\mathbf{x})), \\ \text{subject to } & \mathbf{x} \in S, \end{aligned} \tag{2.1}$$

where k describes the number of objectives f_i in an objective vector $\mathbf{f}(\mathbf{x})$ and \mathbf{x} is a decision vector within a feasible region S in the decision space.

Definition 1. *In a minimization problem, a decision vector $\mathbf{x} \in S$ feasible decision space, is said to **dominate** a decision vector \mathbf{y} if and only if $f_i(\mathbf{x}) < f_i(\mathbf{y})$, $\forall i \in \{1, 2, \dots, k\}$.*

Definition 2. *In a minimization problem, a decision vector $\mathbf{x} \in S$ feasible decision space, is said to **weakly dominate** a decision vector \mathbf{y} if and only if $f_i(\mathbf{x}) \leq f_i(\mathbf{y})$, $\forall i \in \{1, 2, \dots, k\}$ and $f_i(\mathbf{x}) < f_i(\mathbf{y})$, for at least one $i \in \{1, 2, \dots, k\}$.*

Definition 3. *In a minimization problem, a decision vector $\mathbf{x} \in S$ feasible decision space is called a **Pareto optimal solution** if and only if, it satisfies that $f_i(\mathbf{x}) \leq f_i(\mathbf{y})$, $\forall i \in \{1, 2, \dots, k\}$ and $f_i(\mathbf{x}) < f_i(\mathbf{y})$, for at least one $i \in \{1, 2, \dots, k\}$ for all other decision vectors \mathbf{y} and there exist no other decision vector which **dominates** \mathbf{x} in the feasible decision space S . The set of **Pareto optimal solutions** is called a **Pareto-optimal surface** or **Pareto Front (PF)**.*

A Pareto front for a two-objective minimization problem is shown in Figure 2.1. Two solutions \mathbf{x} , \mathbf{y} and their objective vectors $\mathbf{f}(\mathbf{x})$, $\mathbf{f}(\mathbf{y})$ are shown as two examples of non-dominated solutions on the Pareto-optimal surface. Neither solution is preferred to the other. Solution \mathbf{x} has a smaller value of objective function $f_1(\mathbf{x})$ than solution \mathbf{y} , but a larger value of objective function $f_2(\mathbf{x})$. Correspondingly, solution \mathbf{y} has a smaller value of $f_2(\mathbf{y})$ than solution \mathbf{x} , but a larger value of objective function $f_1(\mathbf{y})$. Neither solution \mathbf{x} nor solution \mathbf{y} dominate each other and there are no other solutions which dominate solutions \mathbf{x}

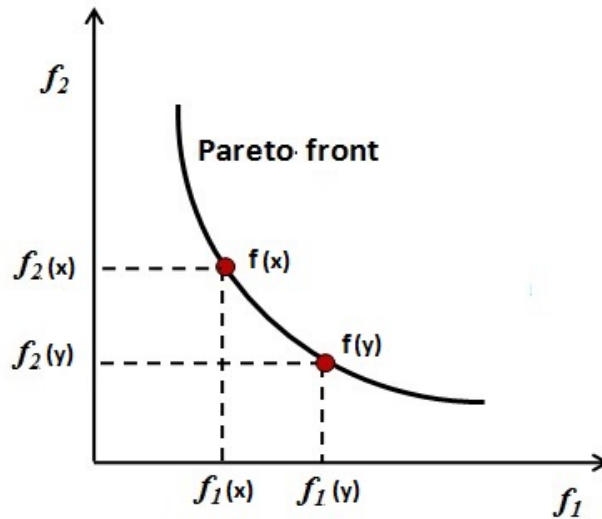


Figure 2.1: Pareto optimal solutions on a Pareto Front for a two objective minimization problem.

and \mathbf{y} on the Pareto-optimal front. These are Pareto-optimal (or non-dominant) solutions, that is, solutions which dominate all other candidate solutions in the decision space.

2.3.1 Goals of Multi-Objective Optimization

In solving the multi-objective optimization problems using an optimizer, three main goals of optimization need to be achieved by the optimizer (Purshouse, 2003 and Fleming et al., 2005). The goals of multi-objective optimization are shown in Figure 2.2 and described below:

- **Convergence:** The approximated non-dominated solutions set obtained by the optimizer is required to be as close as possible to the true Pareto-optimal front of the multi-objective optimization problem.
- **Diversity:** In a multi-objective optimization with many competing objectives there is no ideal single optimal solution for the problem, and the global trade-off surface can potentially have an infinite number of solutions.

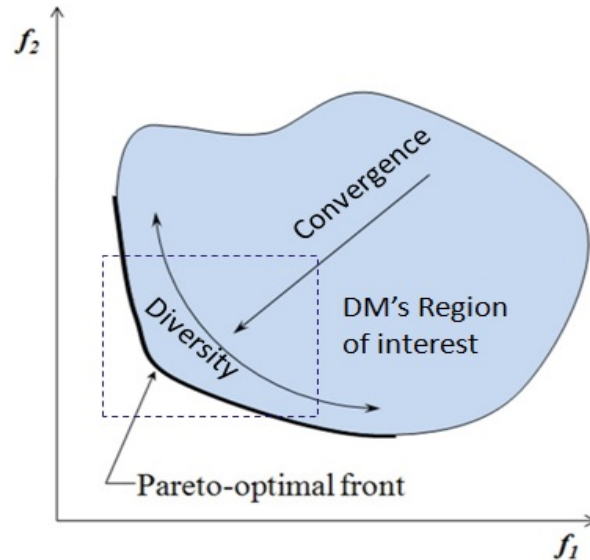


Figure 2.2: Goals of a multi-objective optimization algorithm.

The obtained Pareto optimal solutions are required to be well spread and uniformly covering wide areas of the Pareto front. Diversity is conventionally preferred in the objective space in order to present the DM with a well-distributed set of solutions to choose from, based on certain preferences such as objective priorities or region of interest (ROI). Diversity of solutions is however not restricted to the objective space; it is a desired requirement in the decision space also.

- **Pertinence** (Purshouse, 2003): As the number of objectives in the multi-objective problem increases, the visualization of the optimal solutions becomes difficult. In the decision-making process a decision-maker (DM) is usually interested in sub-regions of the search space which makes the optimization process more practical and efficient. Therefore, the convergence and the diversity of the solutions are particularly required in the pertinent areas of the decision space, i.e., in the decision-makers's region of interest.

2.3.2 General Optimization Process

Optimization, by search, is an iterative process. Optimization of a system/model design involves searching for optimum solutions which will maximize or minimize certain objectives by varying system parameters iteratively. In general, to formulate an optimization problem it is necessary to identify the following key components: a model of the system to be optimized; its decision variables; the objective functions to be optimized; and the system constraints.

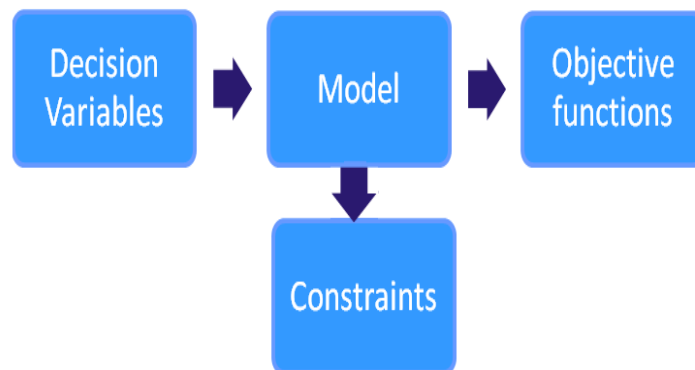


Figure 2.3: Components of an optimization problem.

In the system architecture design process, formulation of an optimization problem begins by identifying the decision variables which are key parameters of the system or model available for variation. A search is conducted over the decision variable space which, typically, have lower and upper bounds. In system architecture design constraints associated with the optimization problem must be identified. These constraints include the resource limitations of the system. The goals of system design optimization are formulated as objective functions. The components of an optimization problem: a model of the system, its decision variables, the objective functions and the system constraints are shown in Figure 2.3. An optimization algorithm generates candidate solutions at each iteration by selecting different values for the decision variables. The constraints and objective functions are evaluated for the values of decision variables of the candidate solutions chosen by the optimization algorithm. If a candidate solution satisfies the

system constraints, then that solution is treated as a feasible solution. If it does not satisfy the constraints, then the solution is treated as infeasible solution. At each iteration a suitable approach is utilized in which the solutions are sorted in terms of their objective function values and constraint violation values and ranked accordingly. The information from the best solutions in each iteration tend to be used to progress towards new solutions for the next iteration.

2.3.3 Classical Optimization Approaches

Classical optimization approaches have been used to solve multi-objective optimization problems in the past (Deb, 2001). Usually in these approaches, the multiple objective functions were aggregated using weights to form a single objective function. These weights are chosen according to the relative importance of the objective functions. Examples include the weighted sum approach, as used in linear quadratic regulator design (Athans & Falb, 1966; Rockafellar, 1987), gradient based methods (Curry, 1944; Fletcher, 1980; Snyman, 2005) and the goal attainment method (Gembicki, 1974). An extensive review of classical optimization methods is given in Rao (2009) and Ravindran *et al.* (2006). Integer linear programming (Heady & Candler, 1963; Murty, 1983; Nemhauser & Wolsey, 1988) is another classical approach used for solving convex integer optimization problems. These techniques require linearization of the objective functions and constraints. Few examples for linear programming approaches are simplex-based methods (Murty, 1983), cutting planes method (Kelley, 1960), and Lagrangian relaxation method (Fisher, 2004).

However, classical optimization approaches have a number of weaknesses for solving multi-objective optimization problems with conflicting objectives:

- These approaches work with a single solution, not multiple solutions, in their iterations to find the optimal solutions.
- The choice of starting position may have a strong influence on convergence to an optimal solution. A poor choice may result in the optimizer getting stuck at local optima, which often exist in engineering problems.

- Aggregation methods can be very sensitive to the chosen weights and starting point in the iterative search.
- These methods require multiple runs of the optimizer to generate sufficient solutions to adequately represent the Pareto-optimal front.
- There is no guarantee with such approaches that multiple runs with various aggregations of objectives will produce the desired diversity in the Pareto-optimal front.

2.3.4 Evolutionary Computation Methods

An alternative approach to solve the multi-objective optimization problems is the class of algorithms known as Evolutionary Algorithms (EAs). They are founded on the principles of natural selection (Darwin, 1858), biological evolution (Holland, 1975; Rechenberg, 1973) and population genetics (Fisher, 1930). EAs are population-based meta-heuristic optimization algorithms. They comprise a population of solutions, which is evolved or improved over a number of generations to search for the non-dominated trade-off solutions and generate the Pareto-optimal front in a single run of the algorithm. Furthermore, the population-based approach can be used to emphasize all the solutions equally and thus produce a diverse set of optimal solutions. Evolutionary algorithms (EAs) are very popular in solving complex design problems. They are global, parallel, search and optimization methods. In addition, various preference articulation methods have been developed and applied to allow the decision-maker to focus the population on the ROI. Multi-Objective Evolutionary Algorithms (MOEAs) are thus well suited to meet the goals of a multi-objective optimization algorithm. There has been much research in the field of evolutionary multi-objective optimization (Coello-Coello *et al.*, 2007; Deb, 2001).

One of the most popular evolutionary computation methods is Genetic algorithms (GAs) popularized by Goldberg (1989). GAs have found applications and very useful in solving problems which are non-convex, discontinuous, multi-modal and difficult to formulate mathematically. These problems are difficult to

solve using conventional numerical optimizers. GAs can handle problems with non-numeric and mixed-type variables. GAs support parallel computing, objective functions can be evaluated on parallel connected processors, thus minimizing the total elapsed computational time. These algorithms apply select, crossover and mutation operator to create new solutions in the non-standard combinatorial search spaces taking into account domain specific characteristics of the problem. These features confer considerable flexibility for addressing the complex requirements of system architecture design.

There are other groups of meta heuristic methods based on biology available in addition to GAs. They are ant colony optimization (Dorigo, 1992; Dorigo *et al.*, 1991) and particle swarm optimization (Kennedy & Eberhart, 1995) based on swarm intelligence, developed for combinatorial optimization. These methods model the behaviour of animals or insects for finding the optimal solutions. A good survey of population based meta-heuristics for multi-objective optimization can be found in (Bianchi *et al.*, 2009; Giagkiozis *et al.*, 2013).

2.4 Review of System Architecture Design using Optimization Methods

Complex trade-offs between various conflicting design objectives exist in the system architecture design. Designers have considered different optimization techniques for exploring these trade-offs in the design space (Gries, 2004; Künzli, 2006; Thiele *et al.*, 2002).

Over a decade earlier, Thompson *et al.* (1999) explored the architecture design optimization of deployment of smart sensors and actuators in future distributed control system (DCS) architectures for aero-engines. Fonseca & Fleming (1993) had introduced the first multi-objective evolutionary algorithm in the early 1990s. Their multi-objective genetic algorithm (MOGA) was used by Thompson *et al.* (1999), to generate and assess candidate architectures. Competing designs were evaluated with respect to a number of parameters, such as risk, weight, acqui-

sition cost, complexity, diagnostic capability and availability. Baseline architectures were defined so that changes made could be compared in a relative manner. In particular, this approach enabled justification to be provided that an optimal system architecture solution was found to be better than the previously existing solution.

Blickle *et al.* (1998) explored an approach to system-level synthesis for optimally mapping an algorithm-level specification onto a software architecture. In the initial step, a hardware architecture is selected with a set of processors, memories and buses. In the second step, in order to map the algorithm onto the selected architecture, the design space is explored with an MOEA with the goal of finding architecture implementations that satisfy a number of constraints on cost and performance. The key outcome of this paper is that it demonstrates that evolutionary algorithms are well suited for system architecture design, in particular, for the selection of architectures and binding the specification to the hardware architecture.

Dick & Jha (1997) presented a multi-objective genetic algorithm for hardware-software co-synthesis (MOGAC). The application of a multi-objective optimization strategy allows a single co-synthesis run to produce multiple designs that trade off different architectural features. Price and power consumption of the hardware system are optimized while constraints are satisfied.

Armstrong *et al.* (2008) developed a tool-set for function-based architecture design exploration. This included function decomposition, adaptive function mapping and complex inter-relations between architecture elements. This information aids in architectural definition and trading of architecture alternatives using an MOEA. They created an Architecture Design Environment (ADEN) to manage the complex inter-relationships between architectural elements, which guides the designer in evaluating the performance of candidate architectures.

Martens *et al.* (2010), developed an evolutionary multi-criteria optimization approach to generate optimal software architectures with respect to goals of per-

formance, reliability, and cost. They applied a multi-criteria optimization algorithm based on genetic algorithms for the software architecture design. Here, the software architecture is modelled with the Palladio Component Model (PCM) (Reussner *et al.*, 2007). Starting with a given initial architectural model, the multi-criteria genetic algorithm iteratively modifies architectural models and evaluates them in pursuit of better performance measures. This approach proved to be successful in systematically exploring the design space spanned by different design options. The PCM approach could be useful in creating software architectures and evaluation of various success criteria for real-world system architecture design problems.

Osekita and Holt (2009) presented a system architecture trade study which considered the integration of an engine health management system as part of a gas turbine engine control system. Eleven candidate architectures were considered for the trade study. They considered four architecture metrics: recurring cost; non-recurring cost; weight; and reliability as objective functions for the trade study and estimated the values of objectives for a baseline architecture, and converted these metrics into a single aggregated cost objective. The values of the objective metrics for the candidate architectures are estimated in a relative manner comparing with the baseline architecture in terms of the changes and complexity of development needed in the candidate architectures. They performed a sensitivity analysis with respect to the weight exchange rate and fleet size. However, the result of this trade study varies with respect to the changes in the weight exchange rate value. A key limitation identified in this study is its focus on a single criterion of aggregated cost objective; this runs the risk of masking the effects of the different elements which contribute to that criterion. In a multi-criteria optimization framework the multiple criteria can be considered individually without aggregating them into a single criterion, and the trade-off of the candidate architectures in terms of the different criteria can be explored. Using the multi-criteria approach, for example, the sensitivity of the design with respect to chosen aggregation factors can be eliminated.

Selva & Crawley (2010) studied the trade-offs between different satellite plat-

forms carrying a single instrument versus those carrying multiple instruments. They proposed a methodology for the analysis of design trade-offs in the satellite system architecture problems. Their architecture design methodology involves a historical study to identify specific requirements and learning points from previous architecture designs. The requirements of the stakeholders, expert systems engineers, customers and program managers are captured through several interviews and qualitative studies. Using this information a quantitative system model is created in order to perform architecture design optimization. They encoded the architecture options and evaluated the candidate architecture alternatives in terms of four objective functions: life-cycle cost, schedule, risk and performance. They performed a multi-attribute trade space exploration (MATE) by using an exhaustive search method and selecting a small number of well balanced architecture alternatives. The key areas of interest in this paper are in the formulation of the objective functions for a complex satellite platform system design and the methodology is applicable to other complex system architecture design problems. However, for a large design space exploration, an exhaustive search method is time-consuming and may not be viable.

Value-Driven Design provides a framework to enhance the systems engineering processes for the design of large systems. By employing economics in decision making, Value-Driven Design enables rational decision making in terms of the optimum business and technical solution at every level of engineering design. Cheung *et al.* (2012), applied Value-Driven Design to the aero-engine system through two case studies, which were conducted through Workshops under the Rolls-Royce plc Advanced Cost Modeling Methodologies project. In a value-driven design approach (Collopy & Hollingsworth, 2011), the values of objective functions are converted to a common score using *a-priori* chosen conversion factors and aggregated by assigning weights to a single score, called Net Present Value (NPV) of the system. The solution arising from this approach, however, is sensitive to the choice of conversion factors and weights.

Bourne *et al.* (2011) developed a framework for designing a distributed control system (DCS) architecture for gas turbine engines using genetic algorithms in

tandem with systems engineering processes. They developed a database containing information about engine dimensions, components and various sub-systems. In the optimization process, genetic algorithm uses the information stored in the decision variables to build the distributed architectures from the functional information, the properties of the engine and available components stored in the database. They evaluated candidate architectures from three viewpoints which are deemed to be most important to system designers and commercial managers:

- Architectural view
- Commercial view
- Life-cycle view

The architectural view considers the structural and technical form of the system. Evaluation functions within this view will analyze the placement of nodes, harness routing, hardware allocation and weight of the system. In the commercial view, the acquisition cost and non-recurring cost of the system are evaluated. The life-cycle view aims to capture and evaluate the costs and added value of the control system throughout its life-cycle based on reliability data and the number of serviceable components. They proved that a substantial systems modelling method with a Multi-Criteria Decision Making (MCDM) approach has the potential to obtain trade-off solutions for the complex system architecture design problems.

From the above review, most of the studies have mainly focused on system architecture design optimization with two or three objectives or design criteria. The optimization algorithms utilized in these studies are suitable for handling multi-objective optimization problems consisting of two to three objectives. The resulting Pareto solutions can be visualized using Cartesian plots or scatter plots and a preferred solution for the optimization problem can be selected from these plots. However, many real-world problems often have more than three objectives. These optimization problems are termed as “many-objective optimization” problems (Farina & Amato, 2002).

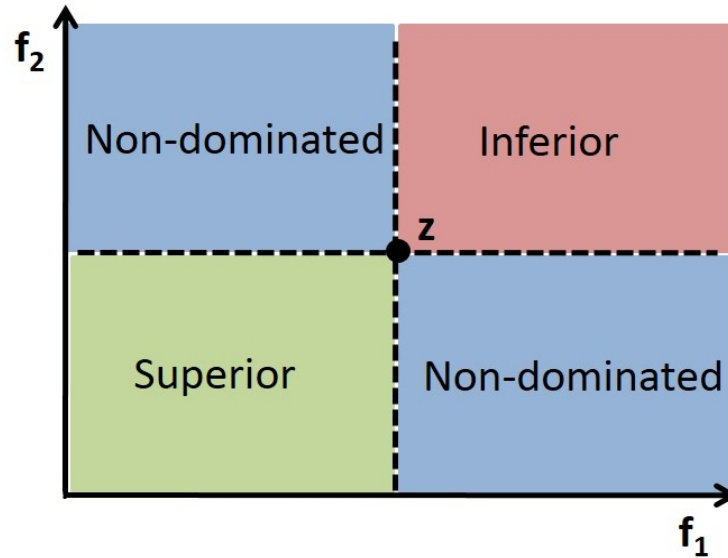


Figure 2.4: Regions of non-domination in a two-objective minimization problem.

2.5 Many-Objective Optimization

2.5.1 Background

Optimization approaches face a number of difficulties in solving many-objective optimization problems.

- In a many-objective optimization problem a large proportion of the population becomes non-dominated in the objective space. Figure 2.4 shows the regions of non-domination with respect to an objective vector z in a two-objective minimization problem. The solutions in the green region have lower objective values compared to z , hence they are superior solutions to z . Whereas, the solutions in the magenta region have higher objective values compared to z , hence they are inferior solutions to z . However, solutions in the blue regions have at least one objective value lower than z . In a two-objective optimization problem, there are 2 out of 4 ($= 2^2$) regions that contain non-dominated solutions. In case the of a three-objective optimization problem, there are 6 out of 8 ($= 2^3$) regions that contain non-dominated solutions. In a k -objective optimization problem, there are $(2^k - 2)$ out of

2^k regions that contain non-dominated solutions. Due to this, there is insufficient selection pressure to progress the search (Khare *et al.*, 2003; Purshouse & Fleming, 2003) and the ability of the optimizer to search towards the Pareto-optimal front can be compromised.

- Large population sizes are required for approximating the Pareto front in many-objective optimization.
- The computational expense of optimization also increases due to an exponential increase in the number of objective function evaluations.
- Visualization of optimal solutions in many dimensions becomes difficult.

In the recent past, several approaches are proposed for improving the convergence of the optimizers for many-objective optimization (Ishibuchi *et al.*, 2008; Zou *et al.*, 2008). In the following sections, potential approaches suitable for many-objective optimization problems are described.

2.5.2 Methods for Increasing Selective Pressure

In evolutionary multi-objective optimization (EMO) problems the lack of effective search, and hence convergence to the Pareto-optimal front, has been attributed to those algorithms based on Pareto-dominance selection (Kukkonen & Lampinen, 2007), e.g. NSGAII (Deb *et al.*, 2002a) and SPEA2 (Zitzler *et al.*, 2001).

2.5.2.1 Modification of Pareto-dominance

The purpose of this approach is to reduce the number of non-dominated solutions in the population by modifying the dominance relation. In the approach of Sato *et al.* (2007), the dominance relation was changed to allow the user to contract or expand the dominance area. This was achieved with a user-specified scaling of the fitness value for each objective to weaken or strengthen selection. Another implementation was that of α -dominance (Ikeda *et al.*, 2001) where a solution \mathbf{x} dominates another \mathbf{y} such that \mathbf{x} can be slightly inferior (by an amount controlled by α) to \mathbf{y} in one objective and superior to \mathbf{y} in other objectives.

Both approaches are examples of the application of weak dominance, which may increase the selective pressure and hence progress of the search towards the Pareto- optimal front. However, both researchers have observed that the diversity of the resulting population is reduced. In addition, both methods require the user to specify parameters to control the strictness of the dominance relation. In Zou *et al.* (2004), L-dominance is defined. This is an extended form of Pareto-dominance taking account of the number and value of improved objectives. For most of the DTLZ problems (Deb, Thiele, Laumanns and Zitzler, 2002) tested it outperforms the state-of-the-art algorithms, IBEA (Zitzler & Künzli, 2004) and NSGAII (Deb *et al.*, 2002a) in terms of convergence to and diversity in, the Pareto-optimal front. However, it does rely on the additional task of normalizing all objectives within the feasible set of solutions, i.e. the maximum and minimum of each objective must be found.

2.5.2.2 Assignment of different ranks

In these approaches different ranking methods are employed to reduce the non-dominated solutions in the population. In the approach proposed by Drechsler *et al.* (2001), they use of a relation called as ‘favour’ to compare solutions. Here, a solution \mathbf{x} is favoured to another solution \mathbf{y} if the number of objectives in which \mathbf{x} is better than \mathbf{y} , is greater than the number of objectives for which \mathbf{y} is better than \mathbf{x} . A more sophisticated version of the relation ‘favour’ called, ϵ -preferred was developed by Sülflow *et al.* (2007) to incorporate a user-specified difference in objective values, ϵ , between compared solutions and so address a variability issue.

Corne & Knowles (2007) compared different ranking methods and conclude that an average ranking method, such as Weighted Average Ranking (Bentley & Wakefield, 1998), outperformed other algorithms, except for problems comprising many objectives where significant objective conflict exists. Ranking solutions and then aggregating them, described as *ranking-dominance*, is used to replace non-dominated sorting based on Pareto-dominance in Kukkonen & Lampinen (2007). Two aggregation functions are tested, the sum and the minimum of ranks, on the DTLZ problems on up to fifty objectives. For four of the six problems,

ranking-dominance showed much improved convergence to the Pareto-optimal front. However, for the remaining two problems, ranking-dominance performed worse than Pareto-dominance. This was due to the fact that ranking-dominance generated solutions in which some objectives were deteriorated while others were improved. Another observation was that in some cases, while convergence was much improved, diversity was significantly compromised.

2.5.3 Indicator Based Methods

As an alternative to Pareto-dominance, various fitness evaluation functions have been developed and applied. Indicator-based algorithms are one such approach, in which a single-objective indicator of a desired property of the population is optimized, e.g. the Indicator-Based Evolutionary Algorithm (IBEA) (Zitzler & Künzli, 2004). A popular indicator is hypervolume, which measures the volume of objective space dominated by a population (Zitzler & Thiele, 1999). On selected many-objective DTLZ test problems, IBEA is shown to perform well (Wagner *et al.*, 2007). However, it has the drawback of being computationally expensive to compute hypervolume values and there have been various efforts to address this. Two examples are the iterative approach of Ishibuchi *et al.* (2007) and the fast hypervolume algorithm (HypE) of Bader & Zitzler (2011). The former only generates one solution per run and so multiple runs are required to generate a Pareto-optimal set of solutions. The latter proposes a fast search algorithm using Monte Carlo approximation of the hypervolume. When evaluated for a constant period of time, HypE outperforms popular MOEAs such as NSGA-II in terms of hypervolume, even though each generation takes longer and hence it processes half the number of generations. The approximation of the Pareto-optimal front depends on the sampling of the Monte Carlo approach. More samples results in a more accurate approximation, but require a longer execution time.

Another fitness evaluation mechanism is that of scalarizing or aggregation functions, which combine multiple objectives into a single objective, often with some form of weighting function. One such approach is Multiple Single Objective Pareto Sampling (MSOPS) by Hughes (2003). It comprises multiple single

objective searches run in parallel, each with a different aggregation of objectives using weight vectors. While this algorithm does not rely on Pareto-dominance to rank solutions and provide selective pressure, it does require specification of weight vectors *a-priori*.

2.5.4 Decomposition Based Methods

Another promising alternative approach to the Pareto-dominance relation is the weighted Tchebycheff scalarizing function based fitness evaluation approach, e.g. MoEA/D (Zhang & Li, 2007). In this a scalarizing function projects the entire objective space onto a line. Here the multi-objective problem is solved as a collection of a number of single-objective problems with different weight vectors defined by the scalarizing function. It has been demonstrated in the literature (Li & Zhang, 2009; Zhang & Li, 2007; Zhang *et al.*, 2009) that MOEA/D has a high search capability for continuous optimization.

2.5.5 Preference Articulation Methods

In multi-objective optimization, preferences can be used to restrict the search space of the optimizer to find the solutions in a certain region of interest on the Pareto optimal front. In Multi-Criteria Decision Making (MCDM), various approaches have been used for this purpose such as specification of weights, goals and aspirations for the criteria (objectives) and constraints for the decision variables (Hwang & Masud, 1979; Fleming *et al.*, 2005; Rachmawati & Srinivasan, 2006; Adra *et al.*, 2007).

Different preference articulation techniques used in the multi-objective optimization are described below:

2.5.5.1 *A-priori* Preference Articulation

In this approach, preferences are specified by the DM before starting the optimization process. A common approach is to specify an aggregating function, which converts individual objectives into a single utility function resulting in a single

objective optimization problem. A few examples of the *a-priori* approaches are the weighted-sum method (Hwang & Masud, 1979), Goal Programming method (Ignizio, 1976), the utility function approach (Greenwood *et al.*, 1996) and the weighting function approaches (Bentley & Wakefield, 1998). These approaches find only a single optimum solution in a single run of the optimizer. Thus, multiple runs are required for generating a well-distributed set of Pareto-optimal solutions. Furthermore, these approaches have difficulties in finding optimum solutions for non-convex problems.

2.5.5.2 *A-posteriori* Preference Articulation

In *a-posteriori* preference articulation approach, the DM specifies preferences after the optimization to identify a preferred solution. Examples include the Pareto optimization approaches NSGA (Srinivas & Deb, 1994), MOGA (Fonseca & Fleming, 1993), and SPEA (Zitzler & Thiele, 1998), non-Pareto approaches such as VEGA (Schaffer, 1985) and VOES (Kursawe, 1991).

When applied to EMO problems, common issues with *a-posteriori* methods are:

- The algorithm may have some difficulty in generating an adequate Pareto front in terms of its diversity of solutions and proximity to the true Pareto front (Purshouse, 2003).
- It may be computationally infeasible for the algorithm to generate Pareto optimal solutions for many objectives, particularly if, in the case of evolutionary algorithms, large populations are used. As the DM is usually only interested in a subset or region of interest (ROI) of the Pareto front, then many of the Pareto optimal solutions may be redundant.

2.5.5.3 Progressive Preference Articulation

In progressive or interactive articulation of preferences approach, the DM can interact with the optimization to provide revised preferences as more information progressively becomes available, in order to resume the optimization to search

for better solutions (Fonseca and Fleming, 1998a; Branke *et al.*, 2001; Branke & Deb, 2004; Cvetković & Coello-Coello, 2005).

Thus, the DM acquires knowledge of the problem as the optimization progresses and very little pre-requisite information such as, goal values, objective ranges and individual objective optima is required. The DM can change the goals and preferences over time, as trade-off solutions emerge in the optimization process and experience of the problem is gained. Compared to a-posteriori method, a progressive preference articulation process can be used to zoom in the ROI on the Pareto front, provide more and better distributed Pareto-optimal solutions and better proximity to the true Pareto front with the same computational cost.

Hence, progressive preference articulation (PPA) can be a promising technique for solving the many-objective optimization problems in order to find Pareto-optimal solution in the DMs region of interest.

2.5.6 A Co-Evolutionary Approach using Preferences

Co-evolving a family of preferences simultaneously with the population of candidate solutions has the potential to be another promising concept for solving many-objective problems. Recently, preference-inspired co-evolutionary algorithms (PICEAs) (Purshouse *et al.*, 2011; Wang *et al.*, 2012) have been shown to be able to achieve better convergence towards true Pareto fronts on many-objective problems and outperform other state-of-the-art algorithms. In these algorithms, a family of preferences are co-evolved with candidate solutions; the preferences gain higher fitness by being satisfied by fewer candidate solutions, and the candidate solutions gain fitness by meeting as many preferences as possible.

2.5.7 Multi-Objective Multi-Modal Optimization

In the case of non-convex multi-objective multi-modal problems, there exist multiple decision vectors which map to identical objective vectors on Pareto front Figure 2.5.

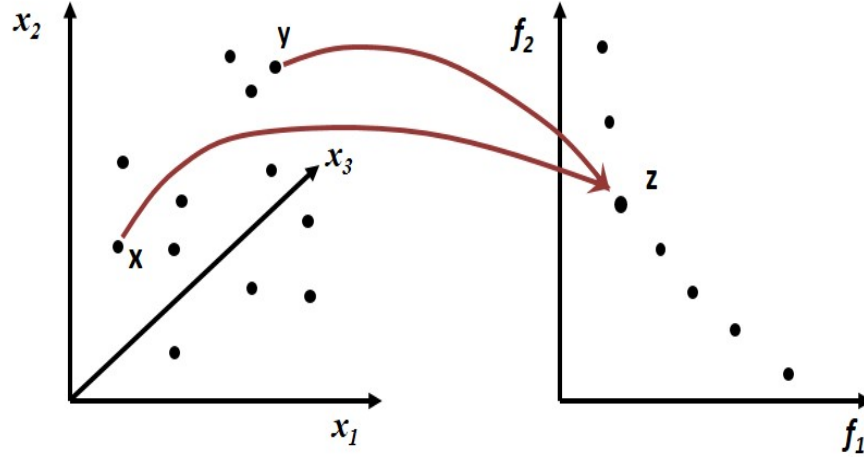


Figure 2.5: Multi-modal Pareto optimal solutions in a multi-objective optimization problem.

Definition 4. In a minimization optimization problem, two decision vectors $\mathbf{x}, \mathbf{y} \in S$ in the feasible decision space, are called **multi-modal Pareto optimal solutions**, if they satisfy $\mathbf{x} \neq \mathbf{y}$, and $f_i(\mathbf{x}) = f_i(\mathbf{y})$, $\forall i \in \{1, 2, \dots, k\}$ and there exist no other decision vector $\mathbf{q} \in S$, such that $f_i(\mathbf{q}) < f_i(\mathbf{x})$, $\forall i \in \{1, 2, \dots, k\}$.

Many multi-objective evolutionary algorithms fail to find and preserve all of the multi-modal solutions in the non-dominated solutions set (Preuss *et al.*, 2006; Rudolph *et al.*, 2007). Due to the incorporation of diversity operators in MOEA, they will assign low fitness values to solutions that are densely clustered in objective space, which will eventually lead to their elimination from the population. Hence they can converge to any one of the global optima out of multiple global optima present in the multi-objective multi-modal problems.

Goldberg & Richardson (1987), introduced a niche preserving technique for obtaining multi-modal solutions in a single-objective optimization, followed by other niching techniques include crowding (Mahfoud, 1992), restricted tournament selection (Harik, 1995), and clustering (Yin & Gernay, 1993) proposed for evolutionary algorithms. A new diversity preserving mechanism is presented in genetic diversity evolutionary algorithm (Toffolo & Benini, 2003) which considers a distance based diversity measure as an additional objective to improve decision

space diversity. Another MOEA approach by Chan & Ray (2005), uses hypervolume and neighborhood counting measures for maintaining diversity in objective and decision spaces, which is computationally expensive to implement on high dimensional problems. (Tarafder *et al.*, 2007) highlighted the importance of finding multi-modal Pareto solutions for a chemical process and proposed modifications to selection operator for NSGAI (Deb *et al.*, 2002a) algorithm for finding these solutions. Deb & Tiwari (2008), developed a generic evolutionary algorithm: Omni-optimizer for solving single and multi-objective multi-modal optimization problems. This optimizer incorporates restricted selection and a crowding distance measure utilizing both objective and decision space information to maintain diversity in both spaces and preserve a well distributed multi-modal solutions. A niche based evolution strategy approach is proposed by (Shir *et al.*, 2009) for enhancing the decision space diversity multi-objective optimization problems. They demonstrated methodology on several two-objective multi-modal problems. However, for many-objective optimization problems these approaches have difficulties in converging towards the true Pareto front. In system architecture design optimization, finding the multi-modal Pareto solutions would allow the decision maker a greater choice when choosing between solutions.

2.5.8 Visualization Methods

Much research on evolutionary multi-objective optimization has concentrated on two or three objective problems (Coello-Coello *et al.*, 2007; Deb, 2001). When there are two or three objective functions in the optimization, it is straightforward to visualize the obtained Pareto-optimal solutions using a scatter plot or Cartesian co-ordinates plot with two or three axes. A decision-maker can select a preferred solution conveniently from the displayed Pareto-optimal front. However, when the optimization problem has more than three objective functions, visualizing the Pareto-optimal solutions is difficult. It is also more challenging for the decision-maker to select a preferred solution for the optimal design. The commonly used visualization approaches in multi-objective optimization are reviewed below; more extensive surveys can be found in Miettinen (1999) and Deb (2001).

2.5.8.1 The Cartesian Co-ordinates Plot and Scatter-Plot

In the multi-objective optimization Cartesian co-ordinates plots are used for visualizing non-dominated solutions having two or three objective functions on the axes and solutions as points at respective objective values. In a Scatter-plot a matrix of pairwise objective plots are used for visualizing the two objective functions in each plot (Cleveland, 1993). The scatter plot is also very useful to show how the objective functions agree with each other and the non-linear relationships between them.

Although scatter-plots are widely used for visualizing the results of 2-objective optimization problems, they can become increasingly difficult to interpret for optimization problems having higher number of objectives. Visually presenting Pareto-optimal solutions for many-objective optimization problems using a matrix of two dimensional diagrams is not very informative for the decision maker.

2.5.8.2 The Parallel Co-ordinates Plot

In the case of a multi-objective evolutionary optimization, the parallel coordinate representation, as originated by Inselberg (1985), involves plotting the normalized objective values of the resulting non-dominated alternative solutions onto a number of parallel axes, one per normalized objective. The objective components for each individual solution are joined by a line. Crossing lines indicate that the objectives are in conflict and parallel lines indicate that the objectives are in harmony with each other. The degree to which the lines cross indicates how strong the trade-off is between adjacent objectives. The ordering of the objective axes is important to reveal the presence and degree of any conflict. The ability to easily re-order or re-scale axes can substantially change the view of the resulting objective space and offer new insights into the structure of the objective data. The parallel coordinates plots allow a multi-dimensional objective space to be represented in a two-dimensional diagram.

Example applications of the use of parallel coordinates are the interactive multi-objective optimization shown in Fonseca & Fleming (1998b) and Fleming

et al. (2005). Here, the decision maker visualize the progress of the optimization on a parallel co-ordinates graph and express his preference in terms of goals on the criteria scores displayed on the parallel co-ordinates trade-off graph in order to steer the search towards his region of interest.

2.6 Uncertainty Analysis and Robust Design Optimization

Uncertainty analysis and robust design optimization has become an important topic in real world design applications (Ben-Tal & Nemirovski, 2002; Beyer & Sendhoff, 2007; Deb & Gupta, 2006). A comprehensive review of robust optimization can be found in (Beyer & Sendhoff, 2007), where a classification of the uncertainty factors involved into systems design optimization is presented. In real-world design problems, several uncertainties related to different environmental conditions, experimental errors and evaluation errors may be introduced in the optimization problem. A widely used sampling method for performing uncertainty analysis is the Monte Carlo(MC) method (Kroese *et al.*, 2011; Robert & Casella, 1999). However, the MC method typically requires a large number of simulation runs that would be computationally very expensive. The main goal of the robust design optimization is to identify a solution, which may or may not be a global optimum solution, that is stable over small changes in design parameters due to uncertainties caused by underlying assumptions.

2.7 Research Gaps

Based on the review of the relevant literature in the previous sections several limitations and research gaps of the currently available multi-objective optimization tools are identified and enumerated below:

- In general, most real-world system design optimization problems will have significantly many objective functions which are non-convex, discontinuous, and multi-modal problems consisting of non-numeric and mixed-type decision variables. In these design problems, resource limitations of the system

are incorporated as design constraints. These problems are difficult to formulate mathematically and difficult to solve using conventional numerical optimizers. The research gap identified here is the lack of a methodology for formulating the system architectures design problems in terms of their qualitative and quantitative criteria, development of the models for evaluating the performance of the candidate architecture alternatives. This research gap is addressed in Chapter 4 of this thesis by developing a methodology for system architecture design using a case study of a real-world aero engine health management system.

- A multi-objective optimization problem having more than 3 objectives can be called a many-objective optimization problem (Fleming *et al.*, 2005). Due to the exponential increase of the number of non-dominant solutions and the consequent reduction in selection pressure, most optimization algorithms fail to converge towards a true Pareto optimal surface (Ishibuchi *et al.*, 2008; Purshouse & Fleming, 2007). Visualization of the Pareto optimal surface also becomes difficult for many-objective problems. System architecture design optimization problems often have multiple decision vectors mapping to same objective vectors on the Pareto front; these are multi-modal Pareto optimal solutions corresponding to different alternative solutions having similar objective values. Finding multi-modal solutions can increase the available choices to the decision maker for selecting a suitable architecture solution. Most popular optimizers fail to find and preserve all of the multi-modal solutions available in the decision search space of the optimization problem. This research gap is addressed in Chapter 3 of this thesis, by proposing an interactive multi-criteria design optimization framework which can solve many-objective optimization problems and find diversified multi-modal Pareto optimal solutions. The optimization framework supports the decision maker by providing a facility for progressive preference articulation, empowering closely coupled user and optimization process interaction.
- Most optimization algorithms attempt to find an approximated set of non-dominated solutions close to the true Pareto front. However, sometimes the

decision maker (DM) is not satisfied with the available set of non-dominated solutions, and wants to have more solutions in a specific region of interest (ROI) on the Pareto front. In such cases, the whole optimization process has to be repeated with more population members in order to increase the density of the approximated non-dominated solutions which leads to an increase in the computational expense. However, an alternative model can be developed using the information from the available Pareto optimal solutions, which can be utilized for estimating new non-dominated solutions in the DM's preferred ROI on the Pareto front. This gap in the research is addressed in the Chapter 3 of this thesis by introducing a Pareto estimation method and an efficient clustering technique to identify one-to-one mapping between the available Pareto optimal objective vectors and decision variables in order to generate a large number of multi-modal Pareto solutions.

- The system architecture design process consists of different models, which can be system modeling language (SysML) models, scoring tables, and meta-models representing expert judgements. These models can involve many assumptions and uncertain parameters, which need to be considered in the optimization process to find robust solutions for the system architecture problem. There is a need for the development of approaches for performing uncertainty analysis and estimating the sensitivities of the solutions with respect to the variations in the uncertain design parameters efficiently. This research gap is addressed in Chapter 5 of this thesis by integrating an efficient uncertainty quantification technique using polynomial chaos expansions within the optimization process. A robustness metric is calculated using the variations of the criteria values with respect to variations in the uncertain design parameter values. In the optimization framework the progressive preference articulation technique facilitates the decision maker to specify the level of sensitivity that can be allowed in the design. The proposed methodology is applied to a real-world system architecture design application of an aero gas turbine engine and robust solutions for an engine health management system architecture design are obtained.

2.8 Summary

In this chapter a brief review of the system architecture design process and previously implemented methods are described.

A review of multi-objective optimization methods was introduced. The concept of Pareto dominance and its utility for multi-objective optimization was presented. The three essential requirements for multi-objective optimizers were described. These are correspondingly the convergence of the approximation set towards the trade-off surface (proximity), the diversity of the approximation set across the trade-off surface, and the pertinence of approximation set to the decision maker. The classical approaches and evolutionary algorithms for solving multi-objective problems were described alongside their limitations for many-objective optimization. In the remainder of this thesis, potential countermeasures and remedial approaches proposed in the literature for improving the convergence of the optimizers are presented. Various preference articulation techniques and visualization techniques are described. Research gaps and goals of this thesis are highlighted. In the next chapter, a proposed interactive design optimization framework for solving many-objective optimization problems with diversified multi-modal Pareto optimal solutions is presented.

Chapter 3

Interactive Multi-Criteria Design Optimization Framework

3.1 Introduction

In this chapter an interactive multi-criteria design optimization framework is proposed for solving the many-objective system architecture design problems. The interactive design optimization framework consists of a multi-objective evolutionary algorithm (MOEA): multi-objective genetic algorithm (MOGA) with a unique progressive preference articulation (PPA) technique. The MOGA is updated with a crowding distance operator in order to identify Pareto optimal solutions with good solution diversity in both objective space and decision variable space.

Next, a novel Pareto estimation (PE) method is introduced for estimating the mapping between the obtained Pareto objective vectors and decision vectors, in order to increase the density of the available Pareto optimal solutions with reduced computational cost. Limitations of the PE method are discussed for multi-objective problems having many-to-one mappings between the decision vectors and Pareto objective vectors. An extension to the PE method is presented using an efficient clustering algorithm to identify groups of decision vectors. The performances of the proposed optimization framework and the extended PE method are assessed by running tests on different benchmark problems.

3.2 Interactive Multi-Criteria Design Optimization Framework

System architecture design problems often involve many conflicting objectives to be simultaneously optimized.

In this research work, an interactive multi-criteria design optimization framework based on a multi-objective genetic algorithm (MOGA) (Fonseca & Fleming, 1993) is proposed. The algorithm uses a non-dominated classification of the population of solutions and rank-based fitness sharing techniques in evolving multi-objective optimization solutions. Every individual in the population, after evaluation of the objective function, is ranked using the following relation:

$$rk_i = 1 + dm_i \quad (3.1)$$

where dm_i is the number of individuals dominating the decision vector \mathbf{x}_i . The idea behind this method of ranking is that misrepresented sections of the Pareto front (PF) will increase the selection pressure to that direction of the front. This information is used to induce better spread in the objective vectors that are not part of the current PF approximation so as to maintain a relatively even supply of objective vectors in all regions of the PF.

MOGA is incorporated with an efficient progressive preference articulation technique (Fonseca & Fleming, 1998a) (PPA_{FF}). This denotes the process of introducing, incorporating and modifying designer preferences in an interactive and progressive way at any time during the optimization search process; this is a key feature for multi-criteria decision making (MCDM). This enables the decision-maker (DM) to set goal values (design requirements or aspirations) for the objectives being optimized and to introduce and change priorities for objectives in a progressive fashion at any time during the optimization process. This enhanced version of MOGA also allows the designer to specify constraints together with a set of priority levels for design objectives in a transparent and consistent manner.

The PPA_{FF} facility is implemented by incorporating a goal attainment method in the ranking procedure. In the goal attainment method, let $\mathbf{g} = \{g_1, \dots, g_k\}$ be

3. Interactive Multi-Criteria Design Optimization Framework

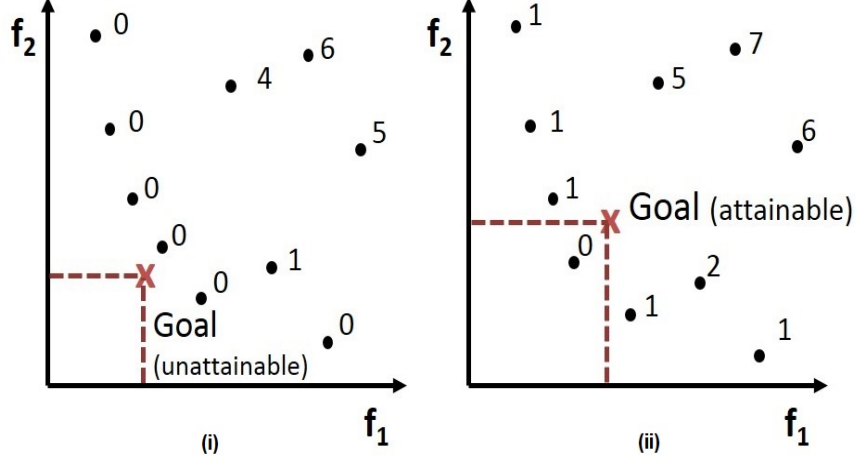


Figure 3.1: Ranking of solutions with (i) an unattainable goal and (ii) an attainable goal, using the progressive preference articulation PPA_{FF} (Fonseca & Fleming, 1998a) technique.

a goal vector and $\mathbf{z}_1 = \{z_1, \dots, z_k\}$ and \mathbf{z}_2 be two objective vectors. A preference relation can be expressed such that (as described in Giagkiozis *et al.* (2013)), when $k - s$ of the k objectives are met, then \mathbf{z}_1 is preferable to \mathbf{z}_2 if and only if.

$$\begin{aligned}
 z_{(1,1\dots k-s)} &\leq z_{(2,1\dots k-s)} \quad \mathbf{or} \\
 \{ &(z_{(1,1\dots k-s)} = z_{(2,1\dots k-s)}) \wedge \\
 ((z_{(1,k-s+1\dots k)} &\leq z_{(2,1\dots k-s+1\dots k)}) \vee \\
 (z_{(2,k-s+1\dots k)} &\not\leq g_{(k-s+1\dots k)}) \}
 \end{aligned} \tag{3.2}$$

Figure 3.1, shows the ranking of solutions in MOGA with (i) an unattainable goal and (ii) an attainable goal, using the PPA_{FF} technique. When the goal is unattainable all the solutions are ranked based on the number of solutions dominating the solution being ranked (equation 3.1). When the goal is attained by few solutions, these solutions are initially ranked as per equation (3.1) within the goal region, then for the solutions outside the goal region their ranks are added to the highest rank of the solutions within the goal region.

3. Interactive Multi-Criteria Design Optimization Framework

In the multi-objective optimization, there exist multi-modal Pareto solutions in decision space, i.e., multiple solutions with equivalent objective function values for the solutions in the objective space. In order to find and preserve all the sets of multi-modal Pareto solutions in the decision space, an MOEA should be able to maintain diversity of solutions found in both objective space and decision space. An MOEA also needs to incorporate elitism, by preserving the best solutions found in initial generations and avoiding them being replaced with inferior solutions in subsequent generations. In the proposed optimization framework, the MOGA optimizer is further enhanced with an efficient crowding operator (Deb *et al.*, 2002a) for maintaining the diversity of the non-dominated solutions both in objective space and decision space, along with an archiving technique for incorporating elitism in the optimizer. A “parallel coordinates” graph in the MOGA software suite facilitates visualization of the interplay between the different objectives.

Number of steps involved in the MOGA with PPA_{FF} and Crowding distance operator, as shown in Figure 3.2, are listed below:

- 1 Create the initial random population for optimization.
- 2 Integrate the system models into the optimization framework and evaluate the objective functions and constraints for design optimization problem.
- 3 Rank the population members using the PPA_{FF} techniques considering the preferred goal values set by the decision maker (DM). The PPA_{FF} technique gives DM feasibility to interact with the optimization process.
- 4 Compute the crowding distance metric values for the non-dominated solutions in the objective space and decision space separately.
- 5 Combine the normalized crowding distance metrics of objective space and decision space for each solution. Sort the solutions in the decreasing order of the combined crowding distance value. The solutions having higher crowding distance value are preferred to maintain diversity of the solutions in both the objective and decision spaces.

3. Interactive Multi-Criteria Design Optimization Framework

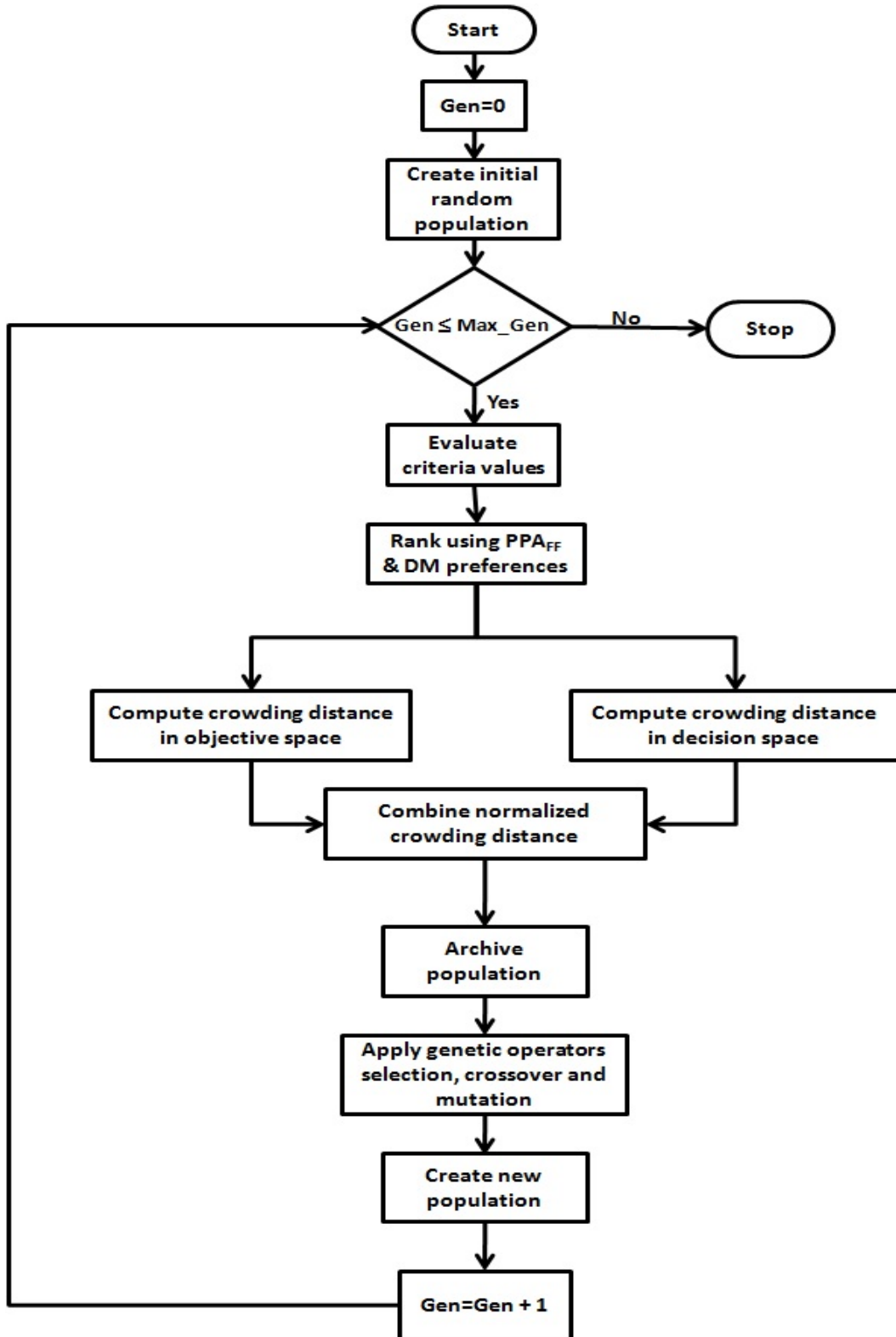


Figure 3.2: Flowchart of multi-objective genetic algorithm (MOGA) with PPA_{FF} and Crowding distance operator.

3. Interactive Multi-Criteria Design Optimization Framework

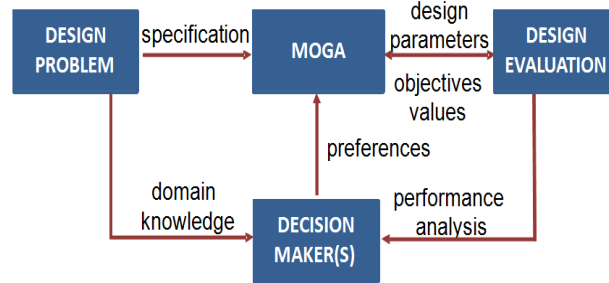


Figure 3.3: Interactive optimization framework with progressive preference articulation technique.

- 6 Archive the non-dominated solutions to maintain elitism in the optimization process.
- 7 Apply the genetic operators selection, crossover and mutation for the population.
- 8 Create new population and evaluate the objectives and constrains for the new populations.
- 9 Repeat the iterations until the maximum generations are reached. However, PPA_{FF} technique enable the DM to stop and start the optimization process in order to express his preferences and guide the optimization to DM's preferred region of interest.

The interactive optimization framework with progressive preference articulation technique is shown in Figure 3.3.

- Initially, the design problem is formulated as a multi-criteria optimization problem. The specification is passed to MOGA.
- MOGA generates initial candidate solutions and passes them to the model for evaluation of the design objective functions.
- The candidate solutions are ranked and crowding distance metric values are computed in both objective space and decision spaces for each solution. The non-dominated solutions are sorted in the decreasing order of crowding distance values. Solutions having higher crowding distance value are

3. Interactive Multi-Criteria Design Optimization Framework

preferred to maintain diversity of solutions in both objective and decision spaces.

- MOGA iterates the optimization process and creates new solutions using the genetic operators and displays the non-dominated trade-off solutions.
- While MOGA evolves new non-dominated solutions, the decision maker (DM) can observe progress of the optimization process and analyze the performance improvements.
- The decision maker can interact with MOGA using the progressive preference articulation approach. Using domain knowledge the decision maker can express preferences and steer the optimization into the design region of interest (ROI).

MOGA trade-off graph with non-dominated solutions for a multi-objective design optimization problem is shown in Figure 3.4. In MOGA, the trade-off values of all objective functions are shown in a “parallel coordinates” graph. In the case of many-objective optimization, the parallel coordinates graph aids the visualization of all constraints and objective functions values on a single plot. This approach places all the axes parallel to each other thus allowing any number of axes to be shown in a 2-D representation. On the ‘x-axis’ objectives are displayed and on the ‘y-axis’ corresponding criteria values are displayed. Each connected line in the trade-off graph represents a Pareto optimal solution for the multi-objective optimization problem.

The bottom window in Figure 3.4 shows the “preference articulation” facility. In this window, all the objective functions of the multi-objective design problem are listed. The decision maker can set goal values for each objective by moving the sliders between maximum and minimum bounds at any time during the optimization process. Goal points for each of the objectives are marked with an “**x**” in the trade-off graph. As the decision maker exercises progressive articulation of preferences, the optimization process will steer the search towards the preferred region of interest (ROI) in the feasible objective space and try to minimize

3. Interactive Multi-Criteria Design Optimization Framework

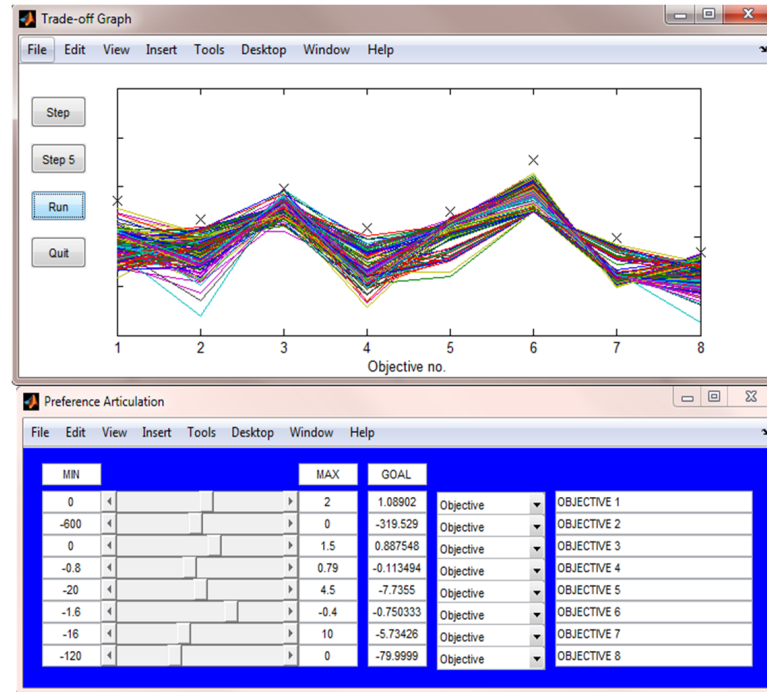


Figure 3.4: MOGA trade-off graph with progressive preference articulation window.

the objective functions values within the specified goals to find Pareto optimal solutions.

3.3 Pareto Estimation Method

Pareto estimation (PE) method was recently introduced by Giagkiozis & Fleming (2012) for increasing the number of available Pareto optimal solutions in the specific regions of the front that are of interest to the decision maker.

3.3.1 Motivation

At the end of an optimization run on a multi-objective optimization problem, the optimizer finds a set of solutions that approximate the Pareto optimal front. Subsequently, these solutions are presented to a decision maker (DM) who can identify a few candidate solutions that are of interest. However, in some cases,

3. Interactive Multi-Criteria Design Optimization Framework

the DM would prefer a solution in the vicinity of the aforementioned solutions which is not present in the given set of solutions. In this case, the analyst does not have many options and would either restart the optimization in the hope that the preferred solution of the decision maker is obtained. This presents a number of difficulties of which the most obvious one is that the computational load is increasing disproportionately to the expected gain. This consideration may lead the decision maker to abandon all the above scenarios and simply select one solution from the already existing Pareto set approximation.

The Pareto estimation (PE) method (Giagkiozis & Fleming, 2012) resolves this issue by allowing the decision maker to explore more solutions in the vicinity of already obtained ones without resorting to further optimization. A major motivation for the introduction of PE has been that Pareto optimal solutions can be obtained in specific regions of the Pareto front without the need to resort to additional optimization runs. Given a set of Pareto optimal solutions, obtained by any optimization algorithm, specific solutions on the Pareto front can be obtained that are not part of the initially obtained Pareto set using the PE method.

The PE method depends on the ability to identify a relationship (mapping) from Pareto optimal solutions in objective space to decision space. In the PE method (Giagkiozis & Fleming, 2012), this mapping is identified using a radial basis function neural network (RBFNN) (Baxter, 1992; Bishop, 1995). This relationship can then be manipulated to produce solutions in specific parts of the Pareto front. We refer to this mapping as, $F_{\mathcal{P}}$, whose domain of definition is the set of Pareto optimal objective vectors, \mathcal{P} , and its range their corresponding decision variable vectors \mathcal{D} .

$$F_{\mathcal{P}} : \mathcal{P} \rightarrow \mathcal{D}. \quad (3.3)$$

3.3.2 Radial Basis Function Neural Networks

In the PE method (Giagkiozis & Fleming, 2012), a specific type of neural network, radial basis function neural networks (RBFNN) (Baxter, 1992; Bishop, 1995; Bors & Pitas, 1996), are used to build the models to identify the mapping from Pareto

3. Interactive Multi-Criteria Design Optimization Framework

optimal solutions in objective space to decision space and for predicting the decision variables of the new solutions in specific regions of the Pareto front. RBFNN have a single hidden layer and an output layer. The output layer is often comprised of linear functions since this guarantees a unique solution to the weights w (Bishop, 1995).

RBFNNs usually employ basis functions that are radially symmetric about their centres μ , for the chosen norm, and decreasing as \mathbf{x} drifts away from μ . A commonly used basis function is the Gaussian (Bishop, 1995), which is given as,

$$\phi_i(\mathbf{x}) = \exp\left(\frac{\|\mathbf{x} - \mu_i\|^2}{2\sigma_i^2}\right). \quad (3.4)$$

In RBFNN the output layer is comprised of linear functions and the hidden layer is highly non-linear in the parameters μ and σ . Various techniques are suggested in the literature (Bishop, 1995), for the selection of their optimal values. In the PE method (Giagkiozis & Fleming, 2012), all the training data is used as centres for the radial basis functions, ϕ_i . Therefore the number of basis functions is equal to the number of training vectors used. Additionally, a uniform value for the parameter, σ , is used for all basis functions, and it is set to $5\bar{d}_h$, where \bar{d}_h is the mean distance of solutions in $\tilde{\mathcal{P}}$ to their nearest neighbour.

Then equation (3.4) becomes,

$$\phi_i(\mathbf{x}) = \exp\left(\frac{\|\mathbf{x} - \tilde{\mathcal{P}}_i\|^2}{2(5\bar{d}_h)^2}\right), \quad (3.5)$$

The output of a RBFNN is a linear combination of the basis function $\phi_i(\cdot)$,

$$y_m(\mathbf{x}) = \sum_{i=0}^{|\tilde{\mathcal{P}}|} w_{m,i} \phi_i(\mathbf{x}), \quad (3.6)$$

where $\phi_0(\cdot) = 1$ is the output layer bias term and $m \in \{1, \dots, n\}$, where n , is the number of outputs which is equal to the number of decision variables.

To validate the created neural network, $(n - 1)$ -cross validation was used as suggested in Jones *et al.* (1998). For a Pareto set of size (P_s, P_s) , RBFNN models

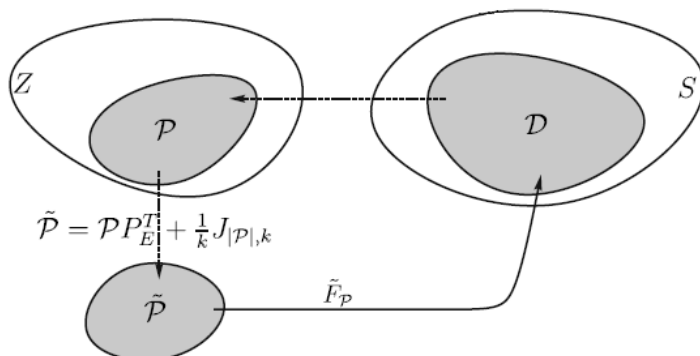


Figure 3.5: The Pareto estimation method (Giagkiozis & Fleming, 2012).

are created using $(P_s - 1)$ samples for the training set and the remaining sample was used to estimate the generalization error. This procedure is repeated until all the solutions in the Pareto set have been used as a test sample and then the mean square error is calculated. The input data, i.e. Pareto objective vectors in each cluster, is passed through the input layer then processed by the radial basis functions of the hidden layer. The outputs of the hidden layer units are then linearly combined and processed in the output layer to produce the output of decision vectors. RBF neural networks do not suffer from the problem of getting stuck at local minima in the parameter space because of their quadratic error function, thus global minima can be easily found.

3.3.3 Pareto Estimation - General Procedure

The process of Pareto estimation method is visualized in Figure 3.5, and the main steps involved in the Pareto estimation method can be summarized as follows:

- The Pareto optimal solutions in the objective space \mathcal{P} are normalized and transformed using the mapping, $\Pi^{-1} : \mathcal{P} \rightarrow \tilde{\mathcal{P}}$, where $\tilde{\mathcal{P}}$ is a transformed Pareto objective vectors. In this step a projection of a normalized version of the Pareto set approximation onto the $(k - 1)$ -dimensional simplex (CH_I) is performed, as shown in Figure 3.6. This step simplifies the task of

3. Interactive Multi-Criteria Design Optimization Framework

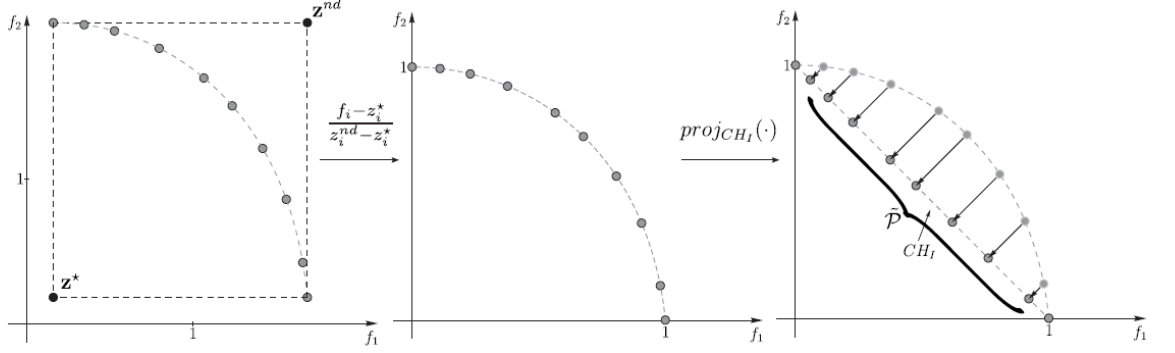


Figure 3.6: The normalization and projection of a two-objective Pareto front (Giagkiozis & Fleming, 2012).

generating a set of uniformly distributed points \mathcal{E} , as it is easy to produce these points on the $(k - 1)$ -simplex in contrast with generating points on the Pareto front.

The Pareto optimal objective vectors \mathcal{P} is normalized according to:

$$\tilde{f}_i = \frac{f_i - \mathbf{z}_i^*}{\mathbf{z}_i^{nd} - \mathbf{z}_i^*}. \quad (3.7)$$

The vectors \mathbf{z}^* and \mathbf{z}^{nd} , ideal and nadir objective vectors are estimated from the Pareto optimal set \mathcal{P} . This normalization restricts the Pareto optimal objective vectors, \mathcal{P} , in the range $[0, 1]$.

The normalized objective vectors projected onto the CH_I using:

$$\tilde{\mathcal{P}} = \mathcal{P}P_E^T + \frac{1}{k}J_{|\mathcal{P}|,k}. \quad (3.8)$$

where $\tilde{\mathcal{P}}$ is the transformed Pareto objective vectors, the matrix $J_{|\mathcal{P}|,k}$ is a unit matrix of size $|\mathcal{P}| \times k$ and P_E is a projection matrix obtained as:

$$P_E = H(H^T H)^{-1}H^T, \quad (3.9)$$

$$H = \left(\mathbf{e}_1 - \frac{1}{k}\mathbf{1} \cdots \mathbf{e}_{k-1} - \frac{1}{k}\mathbf{1} \right),$$

3. Interactive Multi-Criteria Design Optimization Framework

where \mathbf{e}_i is a vector of zeros with its i^{th} element set to 1.

- Subsequently a radial basis function neural network (RBFNN) is trained to identify the relationship, $\tilde{F}_{\tilde{\mathcal{P}}} : \tilde{\mathcal{P}} \rightarrow \mathcal{D}$. \mathcal{D} and $\tilde{\mathcal{P}}$ represent Pareto optimal decision vectors and projected objective vectors, respectively. Once the RBFNN is trained, it is used for estimating a large number of decision vectors whose objective vectors are very likely to be in the targeted regions on the PF.
- The generation of a set, $\mathcal{E} \subseteq CH_I$, and its use with the $F_{\tilde{\mathcal{P}}}$ mapping to generate a set of estimated decision vectors, $\mathcal{D}_{\mathcal{E}}$.

The selection of a set, \mathcal{E} , which controls the region on the Pareto optimal front interested to the DM, is supplied as an input to the RBFNN model and the corresponding objective vectors will be generated in the decision space. The set \mathcal{E} can be generated in one of two regions:

- In a specific region (to be selected by the decision maker).
- The entire CH_I (where the entire Pareto front will be covered, given the Pareto estimation method is successful).

3.3.4 Limitations of Pareto Estimation Method

One of the assumptions in the Pareto estimation method in Giagkiozis & Fleming (2012), is that the objective function is one-to-one, or at least that this condition obtains for the mapping between the Pareto set in decision and objective space. If this condition does not hold, then the artificial neural network used will face difficulties as for the same objective vector it would have to produce two or more output vectors simultaneously.

System architecture design problems objective often result in the multi-modal problems. In the case of multi-objective multi-modal problems, there exist multiple decision vectors which result in identical objective vectors on Pareto front as shown in Figure 3.7 (Kudikala *et al.*, 2013b). The two sets C_1 and C_2 map to the same Pareto optimal solutions in the Pareto optimal set \mathcal{P} . Z is the feasible

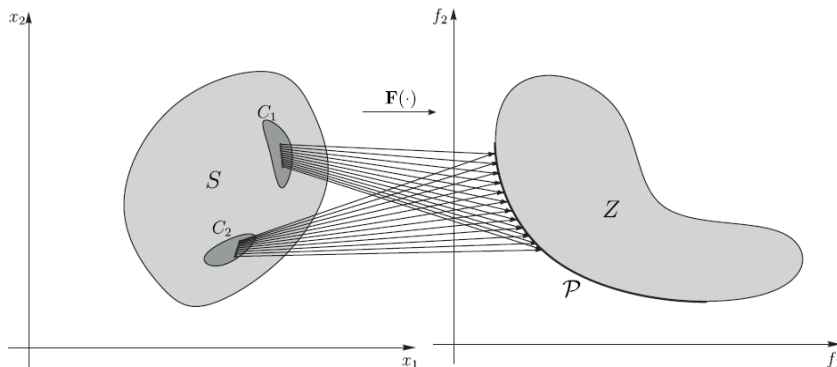


Figure 3.7: A many to one mapping of decision vectors to Pareto front in multi-objective multi-modal problems (Kudikala *et al.*, 2013b).

set in objective space (right) and S is the feasible set in decision space (left). This corresponds to the many-to-one mapping of the multiple decision vectors in \mathcal{D} to the objective vectors in \mathcal{P} . The decision vectors corresponding to each multi-modal optimal points originate from different clusters C_m in decision variable space \mathcal{D} . The RBFNN relationship will fail to produce the one-to-many mapping of $\tilde{F}_{\tilde{\mathcal{P}}} : \tilde{\mathcal{P}} \rightarrow \mathcal{D}$. It will generate any one, but not all, of the multi-modal solutions.

3.4 Extended Pareto Estimation Method with Clustering

In order to overcome the problem revealed in Section 3.3.4, the different clusters C_m of multi-modal solutions present in the non-dominated set can be identified and separated using a clustering algorithm. The obtained clusters of decision vectors C_m and corresponding objective vectors in \mathcal{P} will have a one-to-one mapping between decision variable space and objective space for the Pareto front (Kudikala *et al.*, 2013a,b). Once the different clusters of decision vectors C_m are separated, the RBFNN can be trained for the individual cluster of solutions C_m and $\tilde{\mathcal{P}}$ as shown in Figure 3.8 and identify number of one-to-one mappings $\tilde{F}_{\tilde{\mathcal{P}}_{C_m}} : \tilde{\mathcal{P}} \rightarrow C_m$.

3. Interactive Multi-Criteria Design Optimization Framework

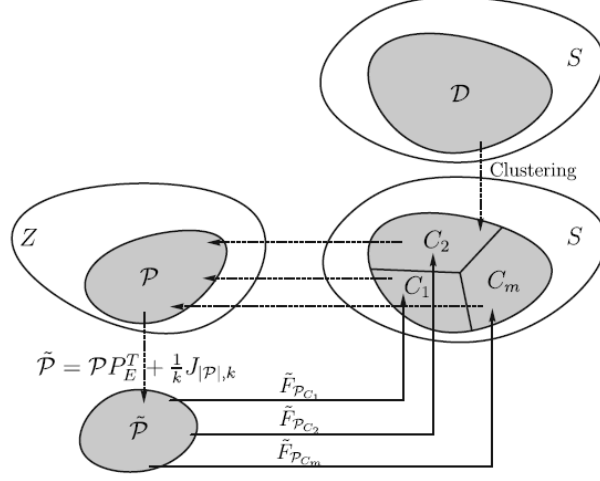


Figure 3.8: The extended Pareto estimation method. Clusters the Pareto optimal decision variable vectors and identifies a mapping for every cluster (Kudikala *et al.*, 2013b).

The steps involved in clustering and Pareto estimation are summarized as follows:

- Step 1** Extract, \mathcal{P} , the non-dominated individuals obtained from an optimization algorithm, and, \mathcal{D} the associated decision vectors.
- Step 2** Perform clustering analysis on the obtained decision variable vectors \mathcal{D} using a clustering algorithm.
- Step 3** For each individual cluster normalize \mathcal{P} using equation (3.7).
- Step 4** Project the normalized \mathcal{P} onto the $k - 1$ dimensional hyperplane defined by the set of vectors $\{\mathbf{e}_1, \dots, \mathbf{e}_{k-1}\}$ using equation (3.9) and equation (3.8), to produce $\tilde{\mathcal{P}}$.
- Step 5** Identify the mapping $\tilde{F}_{\tilde{\mathcal{P}}_{C_m}} : \tilde{\mathcal{P}} \rightarrow C_m$ using $\tilde{\mathcal{P}}$ and C_m as inputs and outputs, respectively, and use these to train a RBFNN.

3. Interactive Multi-Criteria Design Optimization Framework

Step 6 Create a set \mathcal{E} . In this work this is a set of evenly spaced convex vectors on the $k - 1$ dimensional hyper plane.

Step 7 Use the set \mathcal{E} as inputs to the RBFNN created in Step 5, to obtain estimates of decision vectors $C_{\mathcal{E}}$.

Step 8 All the sets $C_{\mathcal{E}}$ can be used with the objective function $\mathbf{f}(\cdot)$ to verify that the produced solutions are non-dominated and acceptable.

3.5 Visual Assessment of Cluster Tendency (VAT) Algorithm

Most clustering algorithms (Jain *et al.*, 1999; MacQueen *et al.*, 1967) need the number of output clusters to be pre-specified as an input to the algorithm. In general we do not know *a-priori* the number of clusters available in the Pareto optimal solutions obtained from the optimization algorithm.

Bezdek & Hathaway (2002) developed a visual assessment of cluster tendency (VAT) method, to identify potential clusters in a data set. Here the pair-wise dissimilarities between the n individuals of the data set are estimated and re-ordered, so that all the neighbouring individuals are consecutively ordered. The reordered $n \times n$ matrix of pair-wise dissimilarities is displayed as an intensity image with $n \times n$ pixels. Clusters are indicated by dark blocks of pixels along the diagonal of the image. However, the VAT method is too computationally costly for larger data sets. Wang *et al.* (2010), proposed an improved VAT (iVAT) and an automated VAT (aVAT) method to automatically determine the number of clusters and cluster separation based on the difference between diagonal blocks and off-diagonal blocks in the image of the reordered dissimilarity matrix.

In this research work, the iVAT method (Wang *et al.*, 2010) is used for identifying different clusters of decision vectors C_m in \mathcal{D} . The iVAT algorithm will estimate and reorder the pairwise dissimilarities between the decision vectors and extract different clusters available in the decision vectors \mathcal{D} . If there are multi-

3. Interactive Multi-Criteria Design Optimization Framework

modal optimal solutions present in the non-dominated solutions found by the optimizer, then the clustering will separate the decision variable vectors corresponding to the multi-modal solutions into different clusters.

A flowchart of the proposed interactive multi-criteria design optimization framework incorporating the MOGA optimizer, the clustering analysis and the Pareto estimation method is illustrated in Figure 3.9. Pareto estimation method and clustering analysis are described in the following sections. Number of steps involved in the proposed optimization framework are listed below:

- 1 Create the initial random population for optimization.
- 2 Integrate the system models into the optimization framework and evaluate the objective functions and constraints for design optimization problem.
- 3 Rank the population members using the PPA_{FF} techniques considering the preferred goal values set by the decision maker (DM). The PPA_{FF} technique gives the DM feasibility to interact with the optimization process.
- 4 Compute the crowding distance metric values for the non-dominated solutions in the objective space and decision space separately.
- 5 Combine the normalized crowding distance metrics of objective space and decision space for each solution. Sort the solutions in the decreasing order of the combined crowding distance value. The solutions having higher crowding distance value are preferred to maintain diversity of the solutions in both the objective and decision spaces.
- 6 Archive the non-dominated solutions to maintain elitism in the optimization process.
- 7 Apply the genetic operators selection, crossover and mutation for the population.
- 8 Create new population and evaluate the objectives and constrains for the new populations.

3. Interactive Multi-Criteria Design Optimization Framework

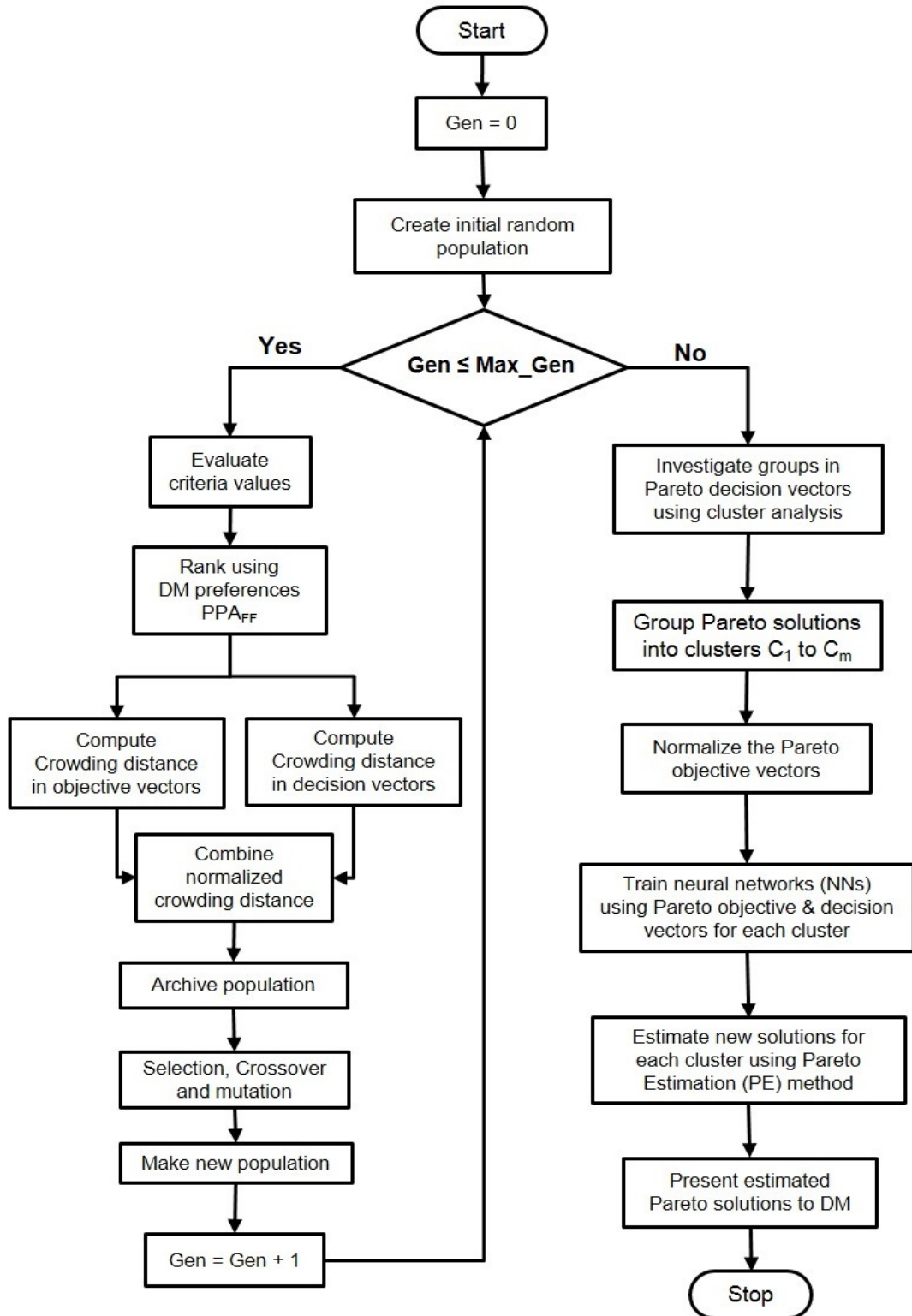


Figure 3.9: Flowchart of the proposed interactive multi-criteria design optimization framework incorporating the MOGA optimizer, the clustering analysis and the Pareto estimation method.

3. Interactive Multi-Criteria Design Optimization Framework

- 9 Repeat the iterations until the maximum generations are reached. However, PPA_{FF} technique enable the DM to stop and start the optimization process in order to express his preferences and guide the optimization to DM's preferred region of interest.
- 10 Obtain the Pareto optimal objective vectors and decision vectors at the end of optimization and investigate the presence of groups in the obtained decision vectors using an efficient clustering technique.
- 11 After the clustering analysis, group the decision vectors and corresponding objective vectors into clusters C_1 to C_m .
- 12 Normalize and transform the objective vectors in individual clusters.
- 13 Train radial basis function neural networks using the objective vectors and decision vectors for each cluster of solutions.
- 14 Estimate new solutions for each cluster using Pareto estimation (PE) method.
- 15 Present the obtained Pareto solutions from PE method to the decision maker.

In the next section the performance of the proposed framework is evaluated using the number test cases of multi-objective test functions.

3.6 Experimental Setup

In order to challenge the search capacities of MOEAs, a set of multi-objective scalable test functions (Deb *et al.*, 2002b) in terms of the number of objectives and decision variables were introduced. These test functions have characteristics, such as multi-modality and discontinuity, which are known to arise in system architecture design problems and generally cause difficulties to most MOEAs. The scalable test functions were intended for the evaluation of the performance of MOEAs when dealing with an increased number of competing objectives. These test functions are widely used as benchmark problems in the EMO community.

3. Interactive Multi-Criteria Design Optimization Framework

In this research work, several test cases of the multi-objective optimization problems with different numbers of variables and with different objective numbers in the optimization are considered. These problems are solved using the proposed MOGA with PPA and crowding distance operators, for 25 times using a different seed for the random number generator in each run. For the purpose of comparison, the test problems are also solved 25 times using NSGA-II (Deb *et al.*, 2002a), Omni-optimizer (Deb & Tiwari, 2008) and random search methods.

3.6.1 Test problem 1

The following multi-objective multi-modal problem is employed as test problem 1.

$$\begin{aligned}
 \min_{\mathbf{x}} \quad & \mathbf{f}(\mathbf{x}) = (f_1(\mathbf{x}), f_2(\mathbf{x})) \\
 = \quad & \left(\sum_{i=1}^n \sin(\pi x_i), \sum_{i=1}^n \cos(\pi x_i) \right) \\
 & x_i \in [0, 6], i = 1, 2, \dots, n.
 \end{aligned} \tag{3.10}$$

The above objective functions, as seen in Deb & Tiwari (2008), are chosen since both objectives are in conflict with each other and will have a trade-off in the objective space. For the minimization case, the above 2-objective optimization problem will have a known Pareto front which varies between $[-\sum_{i=1}^n i, 0]$, where i is the number of decision variables chosen. The above problem is a multi-objective multi-modal problem. The two objective functions are periodic functions with a period of 2. They will have efficient frontiers which correspond to the Pareto-optimal solutions for all the decision variable values varying in the ranges $x_i \in [2r + 1, 2r + 3/2]$, where r is an integer.

3.6.1.1 CASE I

In Case I, two decision variables are chosen in the variable ranges of $x_i \in [0, 6], i = 1, 2$, for the optimization test problem. Within the range of these two variables, the test problem 1 (equation 3.10), will have nine multi-modal Pareto optimal fronts with decision vectors in the ranges of $x_i \in [2r + 1, 2r + 3/2]$, where $i = 1, 2$

3. Interactive Multi-Criteria Design Optimization Framework

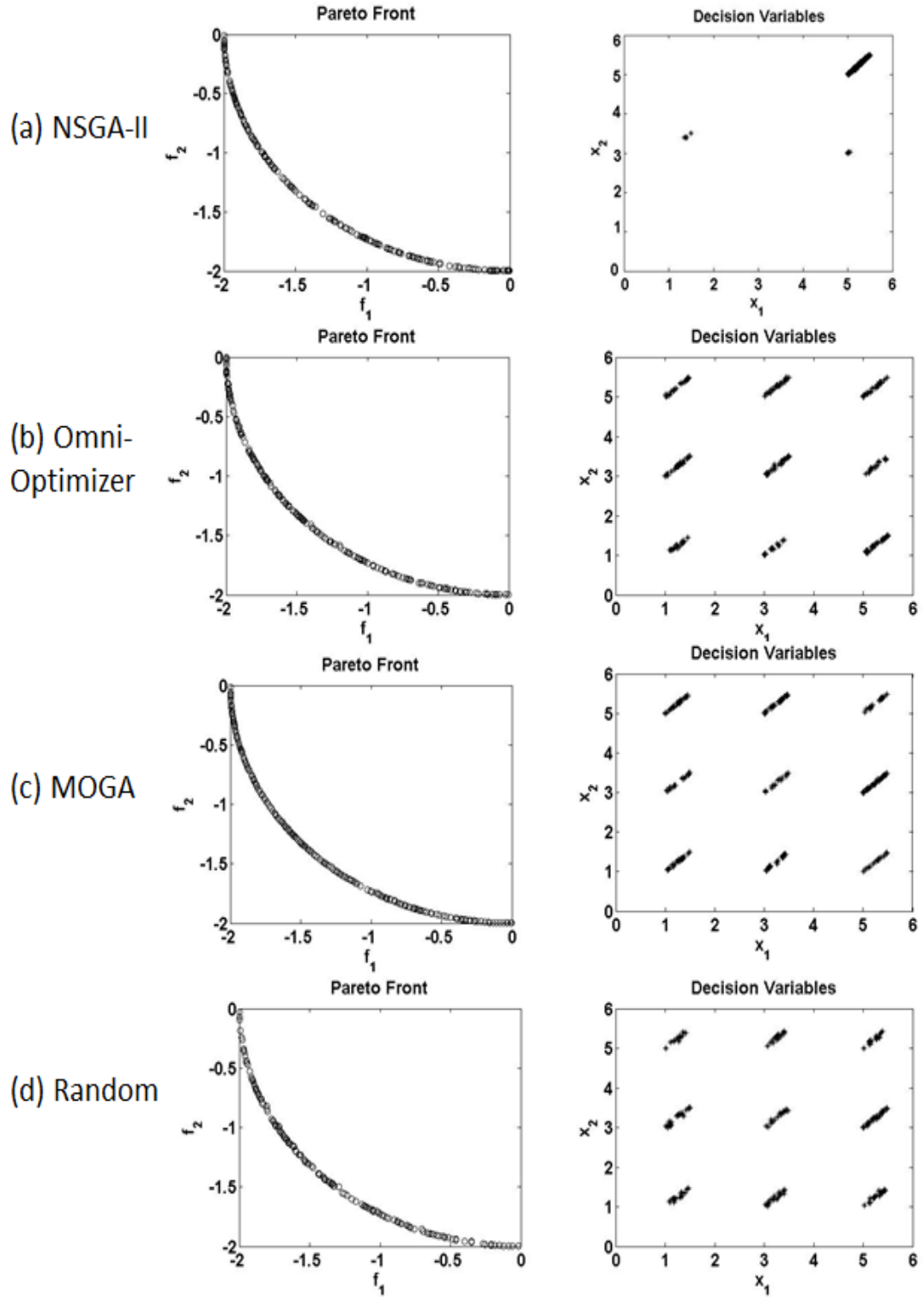


Figure 3.10: Case I: Non-dominated solutions obtained for the test problem 1 (equation 3.10) with 2 variables using NSGA-II, Omni-Optimizer, MOGA and RANDOM search methods.

3. Interactive Multi-Criteria Design Optimization Framework

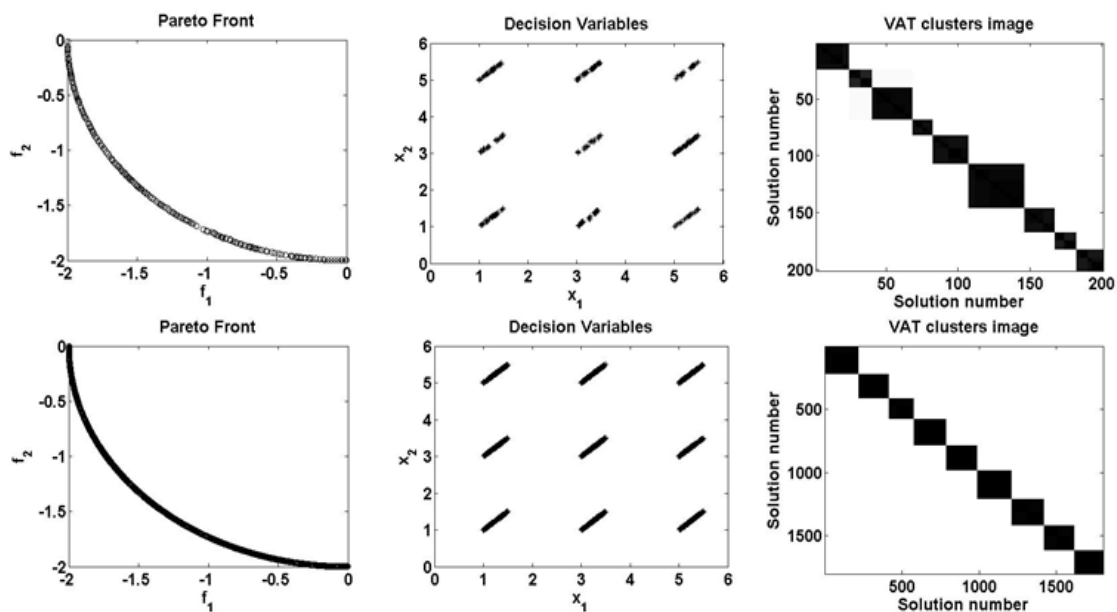


Figure 3.11: Case I: Non-dominated solutions obtained for the test problem 1 (equation 3.10) with 2 variables using the MOGA optimizer (top) and the Pareto estimation method (bottom) in objective space, decision variable space and image of clusters.

and $r = 0, 1, 2$. For an optimizer, finding the Pareto optimal is not so difficult in this problem. However, finding all the multi-modal Pareto optimal solutions with a good distribution in corresponding decision vector ranges is very difficult.

The following parameters are set for the optimizers, which are found to be suitable for the test problem through multiple runs.

- Population size = 200
- Maximum number of generations = 500
- Probability of crossover = 0.8
- Probability of mutation = 0.2

Figure 3.10 shows the obtained Pareto optimal solutions for optimization test problem 1 (equation 3.10) from NSGA-II, Omni-optimizer, MOGA and RANDOM search methods. In this case, NSGA-II, Omni-optimizer and MOGA are converged to the true Pareto front. They have found the Pareto solutions with a

3. Interactive Multi-Criteria Design Optimization Framework

good solution distribution in the objective space. However, in the decision variable space, NSGA-II is not able to find and preserve all the sets of multi-modal Pareto optimal solutions. Whereas the random search found solutions close to Pareto front in the objective space and from all the nine groups of multi-modal Pareto solutions in decision space. In the CASE I, it is evident that NSGA-II performed poorer than the random search with regard to finding diversified multi-modal Pareto solutions in the decision space. It can be seen from Figure 3.10, that proposed MOGA optimization framework with crowding distance operator is able to maintain a good diversity in objective space and decision space and successfully found all the sets of multi-modal Pareto solutions.

In order to demonstrate the extended Pareto estimation (PE) method, the Pareto solutions with a good diversity obtained from the MOGA optimizer are selected. Cluster analysis is performed on the Pareto solutions in the decision space, using the iVAT clustering method as described in Section 3.5. The iVAT clustering method successfully segregated the clusters of all the nine groups of multi-modal Pareto solutions. The reordered dissimilarity matrix of 200 decision vectors is displayed as 200 x 200 image with gray scaling in top right hand side sub-plot of Figure 3.11. The dark blocks appearing on the diagonal of the image represent individual clusters; the size of each dark block, represent number of individuals present in each cluster. It can be seen from this plot, that smallest cluster has 10 optimal solutions and largest cluster has 46 optimal solutions.

Now in each cluster of solutions, the solutions have a one-to-one mapping from decision space to objective space. Using the extended PE method an individual RBFNN model is trained for each cluster of solutions and the mapping function between objective space to decision space is estimated. Then these RBFNN models are used to estimate around 300 solutions from each model. After combining all the solutions obtained from individual Pareto estimations from each RBFNN model, non-dominated sorting is performed on the combined 2700(= 300 × 9) solutions to identify and remove any dominated solutions from the set. Around 600 solutions found to be dominated and around 2100 non-dominated solutions obtained for the optimization problem (equation 3.10). In Figure 3.11 the bottom

3. Interactive Multi-Criteria Design Optimization Framework

sub-plots show plots for objective vectors, decision vectors and gray scale image of the dissimilarity matrix of estimated Pareto solutions. It can be seen from these sub-plots, that extended Pareto estimation along with clustering is successfully able to find many solutions for the multi-objective multi-modal problem.

The benefit of using the Pareto estimation method is reduced computational effort in order to find a greater number of Pareto optimal solutions. Here the computation expense is computed as the total of function evaluations required to generate the Pareto optimal solutions. Using the Pareto estimation method, around 2100 Pareto optimal solutions are generated with the total number of function evaluations, including the initial optimization runs of 500 generations with population of 200, $(200pop \times 500gen) + 2700 = 10.2 \times 10^4$. Whereas, in the case of re-running the optimization process to obtain 2100 Pareto optimal solutions, we need to consider a population size of at least 2100 or more, and run the optimization for 500 generations, which leads to a total number of function evaluations of $2100pop \times 500gen = 10.5 \times 10^5$, which is an order of magnitude greater.

3.6.1.2 CASE II

In Case II, the same objective functions with three decision variables are chosen with variable bounds as $x_i \in [0, 6], i = 1, 2, 3$. Within the range of these three variables, the optimization problem (equation 3.10), will have 27 sets of multi-modal Pareto optimal solutions, each one corresponding to $x_i \in [2r + 1, 2r + 3/2]$, where $i = 1, 2, 3$ and $r = 0, 1, 2$.

The following parameters are set for the optimizers, which are found to be suitable for the test problem through multiple runs. Here the population size is increased to 500 in order to improve the density of the solutions in all of the multi-modal Pareto sets.

- Population size = 500
- Maximum number of generations = 500

3. Interactive Multi-Criteria Design Optimization Framework

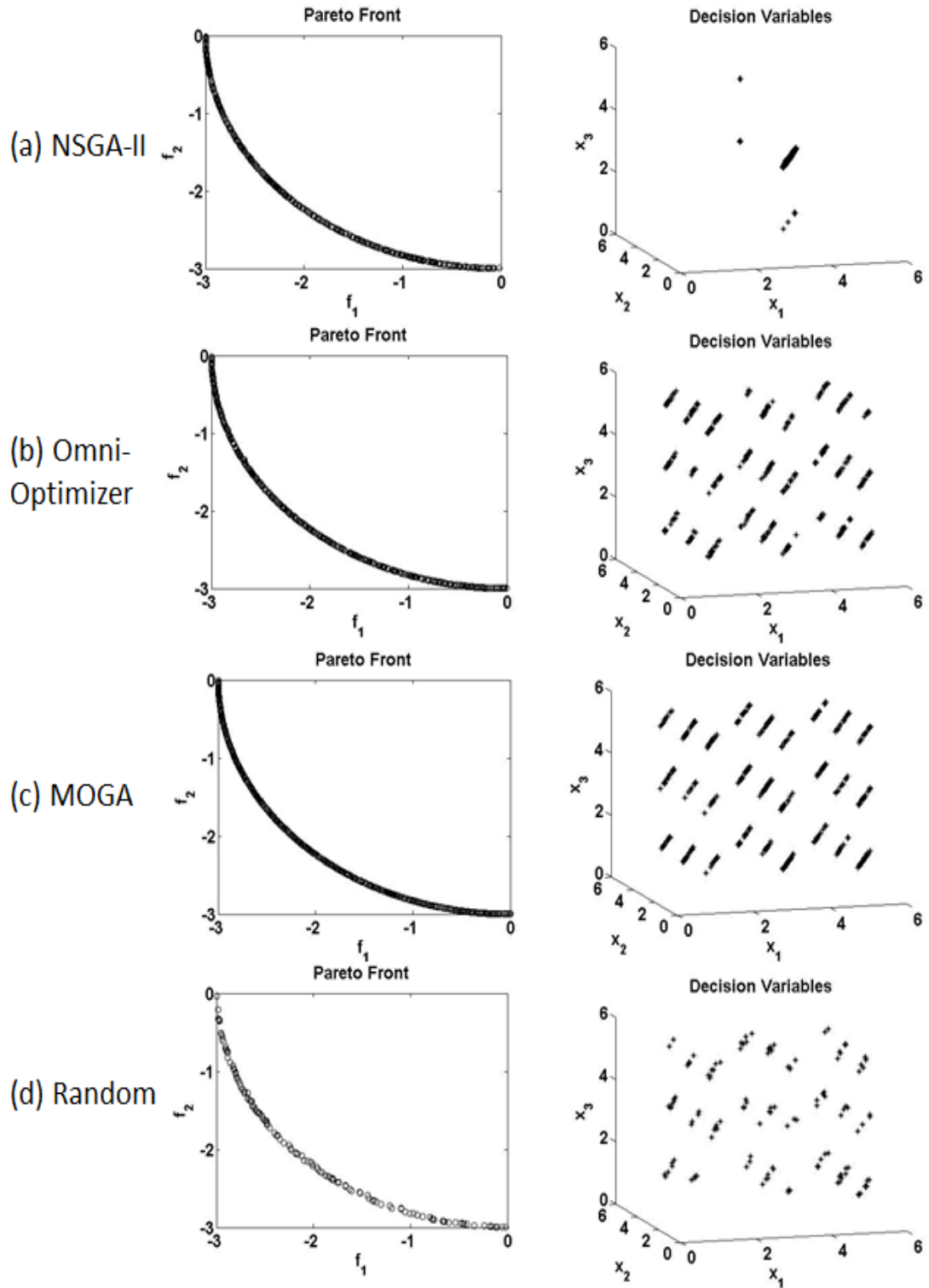


Figure 3.12: Case II: Non-dominated solutions obtained for the test problem (equation 3.10) with 3 variables using NSGA-II, Omni-Optimizer, MOGA and RANDOM search methods.

3. Interactive Multi-Criteria Design Optimization Framework

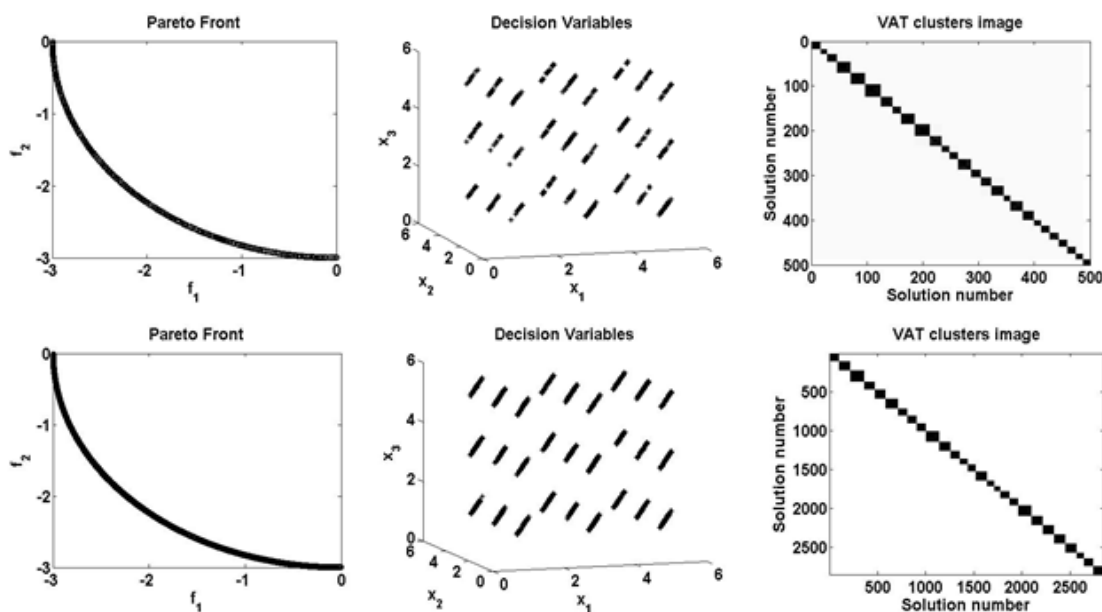


Figure 3.13: Case II: Non-dominated solutions obtained for the test problem (equation 3.10) with 3 variables using the MOGA optimizer (top) and the Pareto estimation method (bottom) in objective space, decision variable space and image of clusters.

- Probability of crossover = 0.8
- Probability of mutation = 0.4

The obtained non-dominated solutions from the four optimizers, are shown in the Figure 3.12. In this case also, all the optimizers found Pareto solutions with good convergence and good solution diversity in objective space. However, maintaining diversity in decision space is a difficult task, since there are 27 groups of multi-modal Pareto solutions available in the decision space. Here, NSGA-II is converged to only one group of solutions in the decision space. Where as, the Omni-optimizer and MOGA have found all 27 sets of multi-modal Pareto solutions with a good distribution of solutions in the objective space and decision space. In the random search only 218 non-dominated solutions are found. However, it is not able to obtain a good distribution of solutions in the decision space.

The extended Pareto estimation method is applied to each cluster of solutions obtained from MOGA. Each cluster has around 16 to 24 solutions, which are

3. Interactive Multi-Criteria Design Optimization Framework

used for the training the 27 RBFNN models. After the training 120 solutions are estimated from each RBFNN model. Non-dominated sorting is performed on the combined 3240 solutions estimated from the 27 RBFNN models and around 485 solutions found to be dominated. After removing the dominated solutions, around 2755 non-dominated multi-modal Pareto solutions are obtained from the Pareto estimation method. In Figure 3.13 the sub-plots show the Pareto solutions, in objective space, decision variable space and image of clusters estimated from the PE method. It can be seen that a very good distribution of non-dominated solutions in both objective and decision spaces are obtained from the extended Pareto estimation method.

3.6.2 Test problem 2

The DTLZ3 (Deb *et al.*, 2002b) problem has been considered as another test problem. The DTLZ3 test problem is scalable to any number of objectives with known Pareto fronts and will have multi-modal Pareto optima for different ranges of decision variables.

$$\begin{aligned}
 \min_{\mathbf{x}} \mathbf{f}(\mathbf{x}) &= (f_1(\mathbf{x}), f_2(\mathbf{x}), \dots, f_k(\mathbf{x})) \\
 \text{where } f_1(\mathbf{x}) &= (1 + g(\mathbf{x}_k)) \cos(x_1\pi/2) \cdots \cos(x_{k-1}\pi/2), \\
 f_2(\mathbf{x}) &= (1 + g(\mathbf{x}_k)) \cos(x_1\pi/2) \cdots \sin(x_{k-1}\pi/2), \\
 &\vdots \\
 f_k(\mathbf{x}) &= (1 + g(\mathbf{x}_k)) \sin(x_1\pi/2), \\
 g(\mathbf{x}_k) &= \left(|\mathbf{x}_k| + \sum_{x_i \in \mathbf{x}_k} (x_i)^2 - \cos(\pi(x_i)) \right)
 \end{aligned} \tag{3.11}$$

subject to $0 \leq x_i \leq 4$, for $i = 1, 2$ & $0 \leq x_i \leq 1$, for $i = 3, 4, \dots, n$.

In the current research, three cases of the DTLZ3 problem are considered with three, four and six objectives, having twelve decision variables with bounds, $x_1, x_2 \in [0, 4]$, and remaining $x_i \in [0, 1], i = 3, 4, \dots, 12$. Within the range of these twelve decision variables, the DTLZ3 problem will have two sets of known multi-modal Pareto optimal solutions. For solving the DTLZ3 optimization problem,

3. Interactive Multi-Criteria Design Optimization Framework

we have set the following parameters for the optimizers, which are found to be suitable for the DTLZ3 problem through multiple runs.

- Population size = 500
- Maximum number of generations = 500
- Probability of crossover = 0.9
- Probability of mutation = 0.1

For the current DTLZ3 problem test cases with 3, 4 and 6 objectives, challenging tasks for an optimizer are converging towards the Pareto front and finding all the multi-modal Pareto optimal solutions with a good distribution in both objective space and decision space.

Non-dominated solutions obtained from NSGA-II, Omni-optimizer, MOGA and random search methods for the DTLZ3 problem test cases with 3, 4 and 6 objectives are shown in Figures 3.14, 3.15 and 3.16 respectively. Since, there are 12 decision variables and more than 3 objectives present in the DTLZ3 problem test cases, here the decision variable plots and the objective plots are shown in “parallel co-ordinates” plot, representing each vertical axis as a corresponding decision variable and objective respectively.

In the test case with 3 objectives, the three optimizers NSGA-II, Omni-optimizer, and MOGA converged to true Pareto front for the given problem as shown in Figure 3.14. However, NSGA-II is able to find only one set of Pareto optimal solutions in the decision space. Where as, the Omni-optimizer and the MOGA found two groups of multi-modal Pareto optimal solutions. In the random search the solutions obtained are sub-optimal in all the three test cases of DTLZ3 problem, however they are diversified in the decision space.

It can be observed from Figures 3.15 and 3.16, as the number of objectives increases to 4 and 6 respectively, the convergence of the NSGA-II and Omni-optimizer is reduced significantly. This is due to the increased number of non-dominated solutions in early generations and reduced selective pressure in the

3. Interactive Multi-Criteria Design Optimization Framework

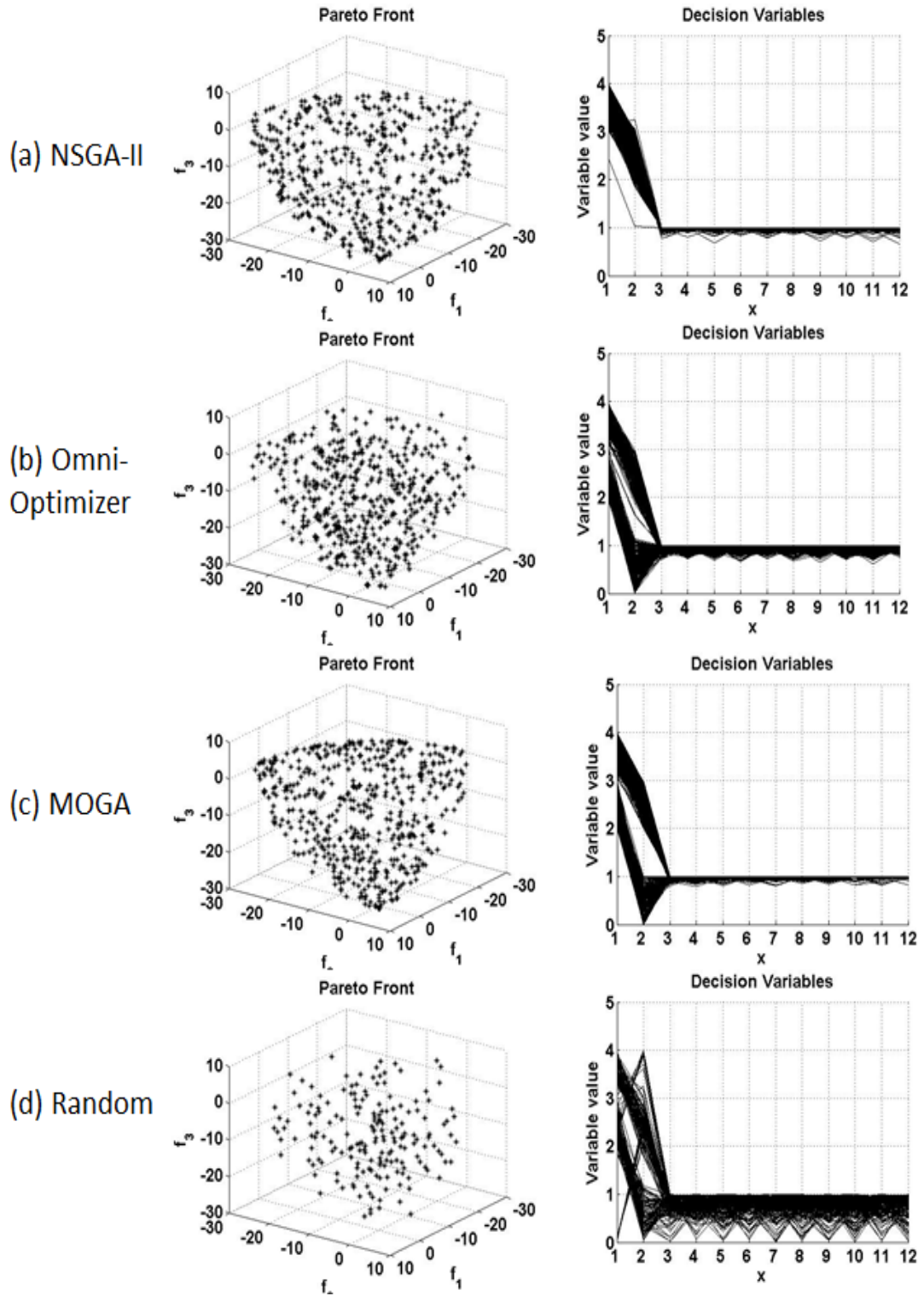


Figure 3.14: Non-dominated solutions obtained for the DTLZ3 test problem with 3 objectives from NSGA-II, Omni-Optimizer, MOGA and RANDOM search methods.

3. Interactive Multi-Criteria Design Optimization Framework

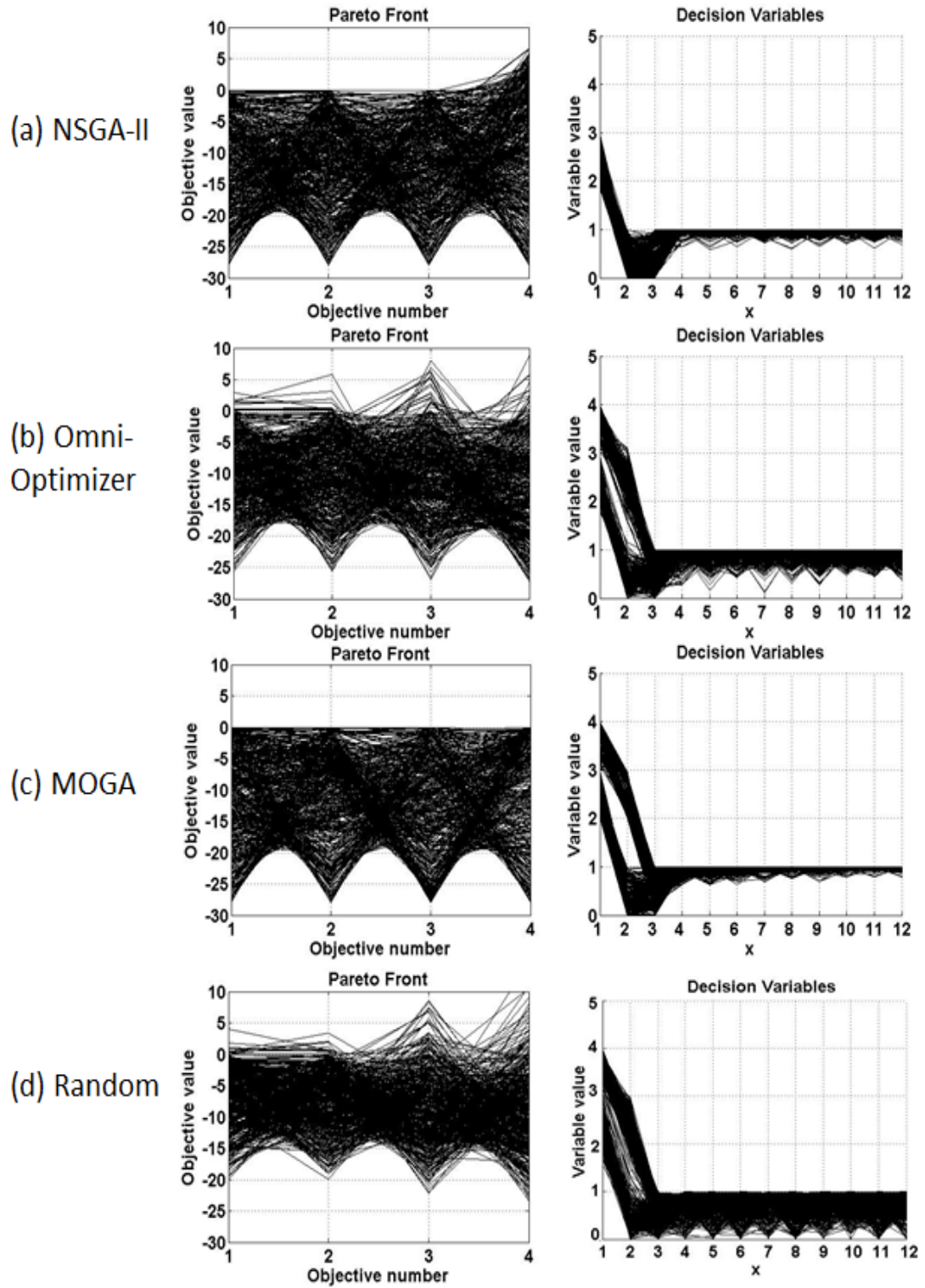


Figure 3.15: Non-dominated solutions obtained for the DTLZ3 test problem with 4 objectives from NSGA-II, Omni-Optimizer, MOGA and RANDOM search methods.

3. Interactive Multi-Criteria Design Optimization Framework

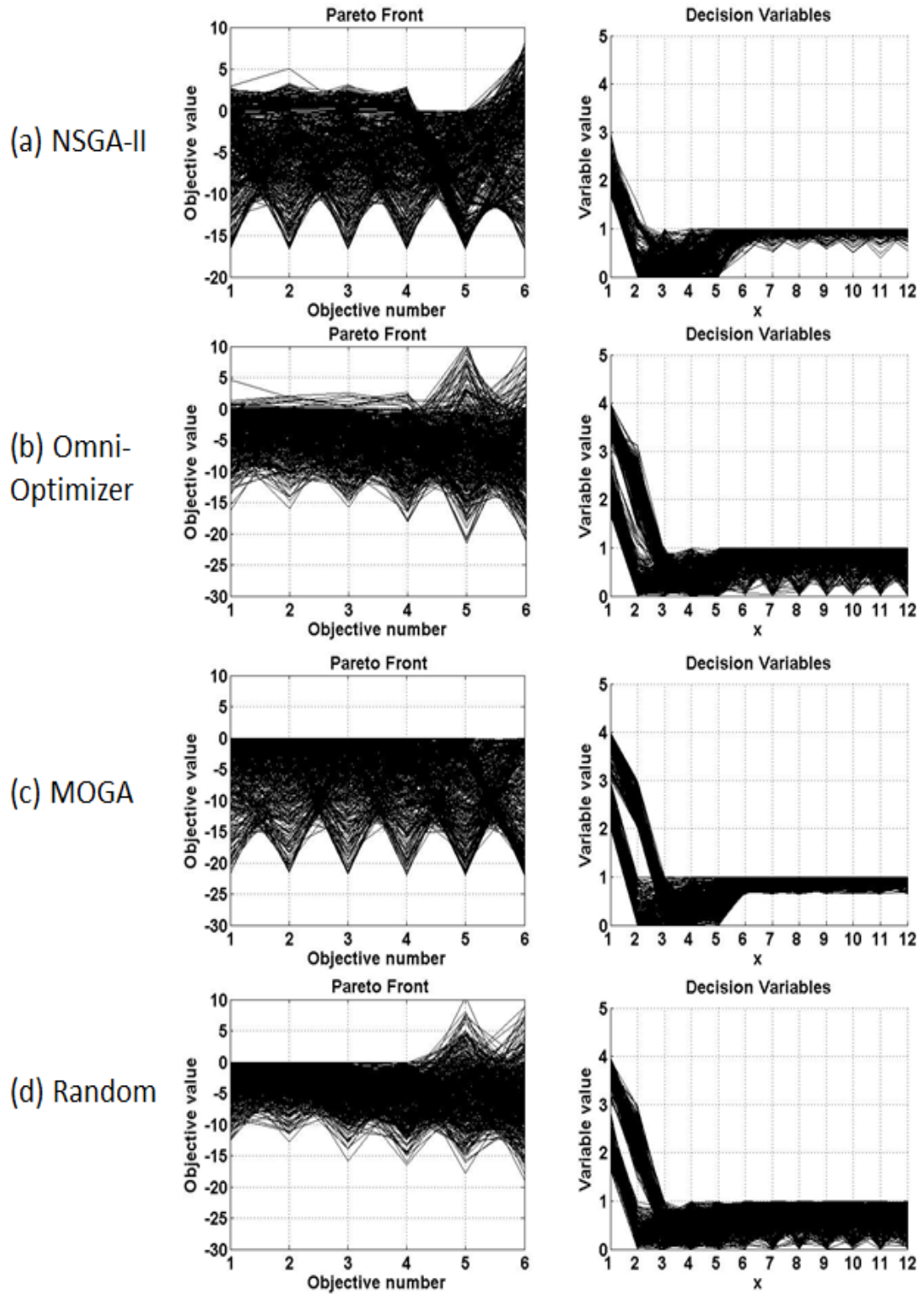


Figure 3.16: Non-dominated solutions obtained for the DTLZ3 test problem with 6 objectives from NSGA-II, Omni-Optimizer, MOGA and RANDOM search methods.

3. Interactive Multi-Criteria Design Optimization Framework

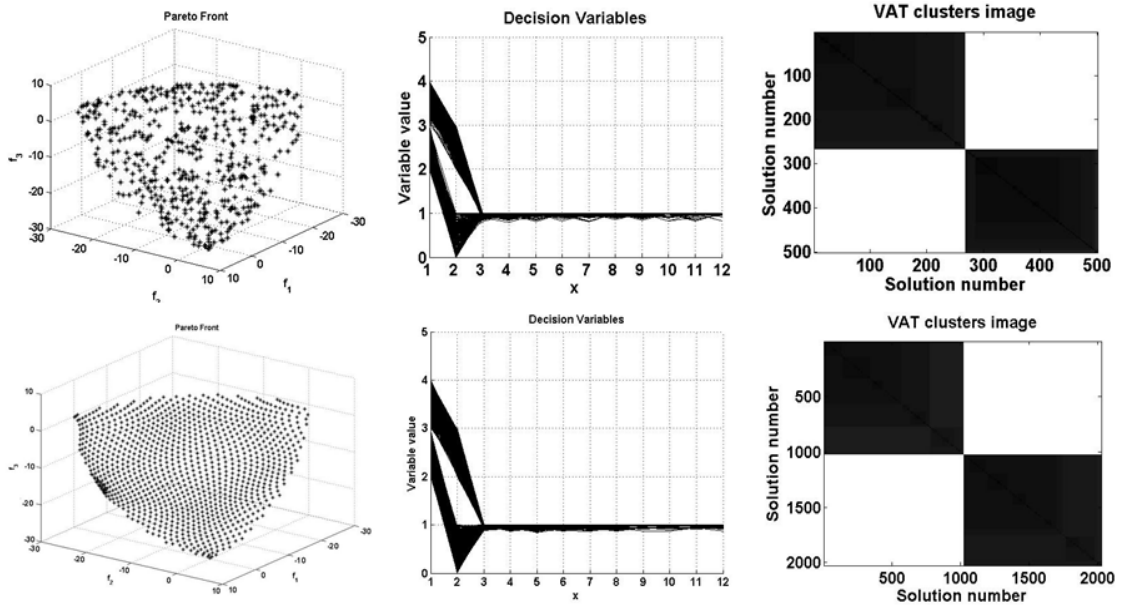


Figure 3.17: Non-dominated solutions obtained for DTLZ3 problem with 3 objectives using the MOGA optimizer (top) and the Pareto estimation methods (bottom) in objective space, decision variable space and image of clusters.

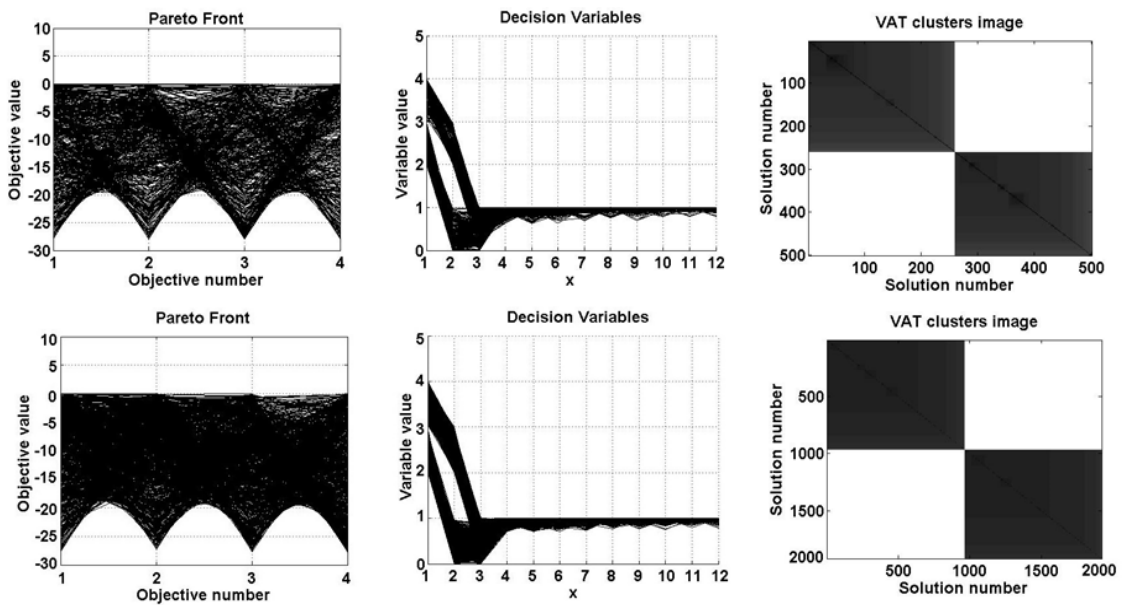


Figure 3.18: Non-dominated solutions obtained for DTLZ3 problem with 4 objectives using the MOGA optimizer (top) and the Pareto estimation methods (bottom) in objective space, decision variable space and image of clusters.

3. Interactive Multi-Criteria Design Optimization Framework

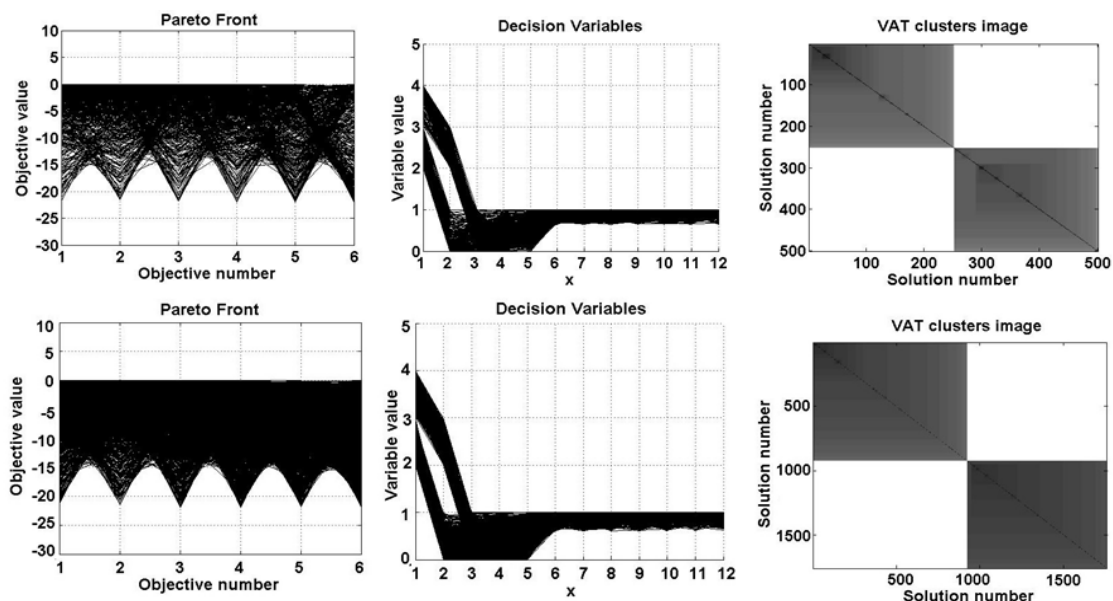


Figure 3.19: Non-dominated solutions obtained for DTLZ3 problem with 6 objectives using the MOGA optimizer (top) and the Pareto estimation methods (bottom) in objective space, decision variable space and image of clusters.

optimization process in order to progress towards the true Pareto front. However, in MOGA using progressive preference articulation technique preferences are expressed on the objectives during the optimization process in a progressive manner interactively. This will enable the MOGA optimizer to steer the optimization process towards the true Pareto front. Here, a well distributed multi-modal Pareto optimal solutions are obtained using MOGA in both the test cases with 4 and 6 objectives. The obtained results demonstrate the capability of the proposed methodology in finding multi-modal Pareto optimal solutions with a better convergence and good diversity in both objective and decision spaces.

Cluster analysis using the iVAT method is performed for the obtained Pareto optimal solutions from MOGA optimizer for the DTLZ3 problem test cases with 3, 4 and 6 objectives. The reordered dissimilarity matrix of 500 decision vectors is displayed as a 500×500 image showing the two dark blocks in the respective sub-plots of three test cases. Figures 3.17, 3.18 and 3.19, show the estimated Pareto solutions and iVAT clusters images for the DTLZ3 problem test cases

3. Interactive Multi-Criteria Design Optimization Framework

with 3, 4 and 6 objectives using the extended Pareto estimation method. It can be seen from this plot that there are two clusters available in the decision vectors corresponding to the two global optima. After separating these clusters of decision vectors, for each cluster, the Pareto estimation method is applied to find the one-to-one mapping between objective vectors and decision vectors for the three test cases. Then this mapping is used to estimate 1000 solutions in each cluster. After the estimation of new solutions, we combine all the solutions obtained from Pareto estimations for each cluster, and perform non-dominated sorting to remove any dominated solutions from the set for the individual test case. In the end around 2000 non-dominated solutions are estimated using the extended Pareto estimation method for each of the DTLZ3 problem test cases with 3, 4 and 6 objectives. The quality of mapping estimated by RBFNN is highly dependent on the supplied training decision vectors. If the training data has a sufficient number of vectors, well distributed, then the RBFNN will estimate a better mapping, otherwise, the mapping estimated by RBFNN will be deceptive and may not generate good solutions in the Pareto estimation process. Due to this reason, there are less estimated solutions found to be non-dominant in the test case with 6 objectives. In the lower sub-plots Figures 3.17, 3.18 and 3.19, show the estimated non-dominated solutions, in objective space, decision variable space and gray scale image of the dissimilarity matrix of decision vectors showing clusters.

It can be seen that, using the Pareto estimation approach, a very good distribution of non-dominated solutions is obtained for all the benchmark multi-objective multi-modal test problems with reduced computational expense. In the design optimization process a decision maker can chose to estimate the Pareto solutions in a preferred region of interest in objective space and decision space.

3.7 Summary

In this chapter, an interactive multi-criteria design optimization framework is proposed for solving the many-objective system architecture design problems. The interactive design optimization framework consists of an evolutionary multi-

3. Interactive Multi-Criteria Design Optimization Framework

objective genetic algorithm (MOGA) with a unique progressive preference articulation (PPA) technique. The MOGA is further enhanced with a crowding distance operator in order to maintain a good diversity in both objective space and decision space and find multi-modal Pareto solutions. A Pareto estimation (PE) method is introduced and an extended version of Pareto estimation (PE) method is proposed, to increase the density of the multi-objective multi-modal Pareto solutions. The method uses an efficient clustering technique to identify and separate different clusters in the decision variable space which correspond to the multi-modal Pareto optimal solutions. Then PE method is employed to estimate a large number of Pareto optimal solutions, thereby increasing the density of available non-dominated solutions in the multi-objective problems. The proposed method has been tested on several benchmark test problems, with different case studies. In all cases, the proposed optimization methodology has successfully found multi-modal Pareto solutions with a good convergence and diversity in objective space and decision space.

In the next Chapter, the proposed interactive multi-criteria design optimization framework is applied to a real-world aero gas turbine engine health management system architecture design problem. Formulation of the system architecture design multi-criteria problem and integration of system architecture models to optimization framework is presented.

Chapter 4

System Architecture Design: An Aero Engine Health Management System Case Study

4.1 Introduction

In this Chapter, the process of system architecture design is demonstrated using a case study of a real-world application of an engine health management (EHM) system for an aero gas turbine engine (GTE). The EHM system functional requirements are captured and decomposed into a number of functional operations. In the EHM system architecture design problem the system functional operations are to be optimally deployed on physical architecture subsystem locations in order to satisfy operational attribute requirements within the constraints of the physical architecture limitations. A multi-criteria optimization problem formulation for the EHM system architecture design is presented. Deployment locations of the operations are considered as decision variables and requirement violations, in terms of operational attributes, are considered as multiple objective functions to be minimized in the optimization process. Integration of the system architecture model into the optimization framework and evaluation of architecture criteria values is described. The EHM system architecture design optimization problem is solved using the proposed interactive multi-criteria optimization framework.

Clustering analysis is performed to identify different groups of architecture solutions highlighting useful design features. For each cluster of architecture solutions the Pareto estimation method is applied and additional non-dominated solutions are generated. Performance of the proposed optimization framework is assessed by evaluating test metrics and comparing the results with different optimizers.

4.2 Aero Engine Health Management System Architecture Design

An engine health management system has become an essential part of aero gas turbine engines in recent times, in order to minimize the cost of operation of the gas turbine engine and its associated equipment. The cost of operation includes fuel, scheduled maintenance and unforeseen events that result in the engine not being available for service to the customer. An EHM system can help to reduce costs due to unanticipated disruptions to service. Health status reports from the EHM system will improve the proper scheduling of the maintenance process, resulting from the greater knowledge base of the engine component failures detection, identification and prognosis of remaining life of the components.

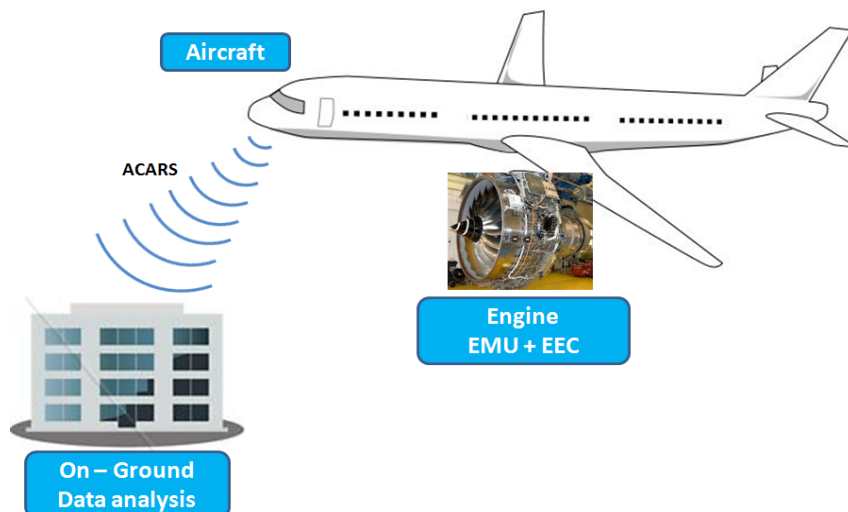


Figure 4.1: Aero Engine Health Management System.

The EHM system in an aero engine forms part of the engine electronic control (EEC) system, with an engine monitoring unit (EMU) at its center. The EMU is mounted along with the EEC on the fan case of the aero engine. Figure 4.1 shows an EHM system for an aero engine. The main aim of an EHM system is to perform real-time parameter analysis and anomaly detection of the aero engine. Output from on-board analysis can be passed to an on-ground computer resource using an aircraft communications addressing and reporting system (ACARS) over satellite for further analysis to predict, classify and locate developing engine faults and anomalies. The optimum combination of on-engine and on-ground computational resources for the EHM system will deliver the benefits of reducing the engine life-time operation and maintenance costs (Tanner & Crawford, 2003).

4.2.1 Stakeholder Identification and Requirement Analysis

The system architecture design process starts with the identification of key customers and stakeholders. It is an essential task for capturing all the stakeholder requirements. Different techniques are used for capturing requirements, such as stakeholder interviews, surveys, customer feedback forms, historical data and maintenance reports from other engine fleet, project specific goals etc.

In the current research work, for designing the EHM system architecture for Rolls-Royce aero gas turbine engine, all the stakeholders are identified and their requirements are captured using requirements analysis and flow-down techniques, for example, Quality Function Deployment (QFD) (Mizuno *et al.*, 1994), and represented as EHM system use cases as shown in Figure 4.2. Some of the typical EHM system functional requirements include: engine vibration monitoring and reporting, oil debris monitoring, asset (engine) performance monitoring, engine mechanical fault or novelty detection and engine/EEC abnormal incident reporting.

4. EHM System Architecture Design

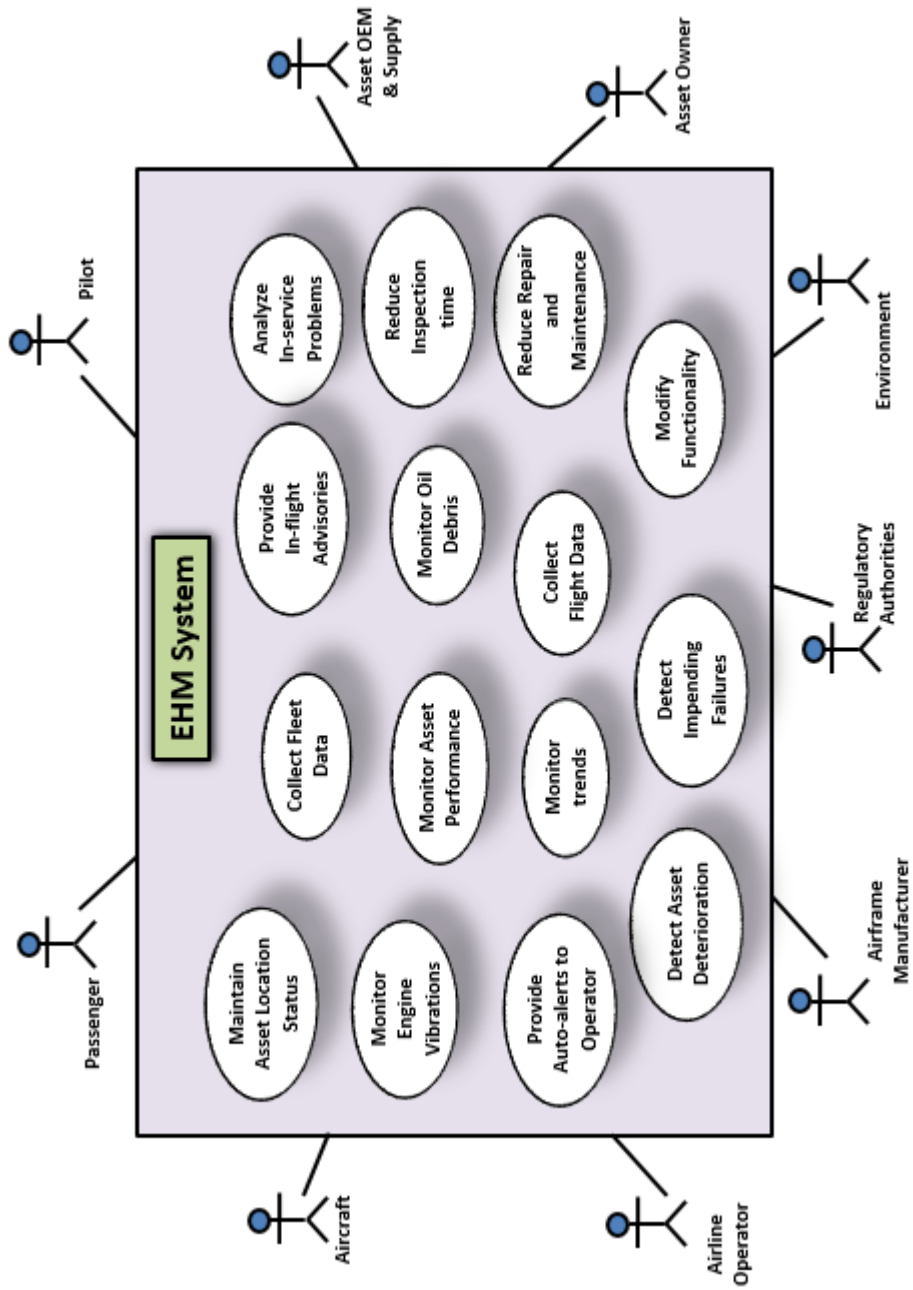


Figure 4.2: Stakeholders and use-cases identified in EHM system design (Tanner 2010).

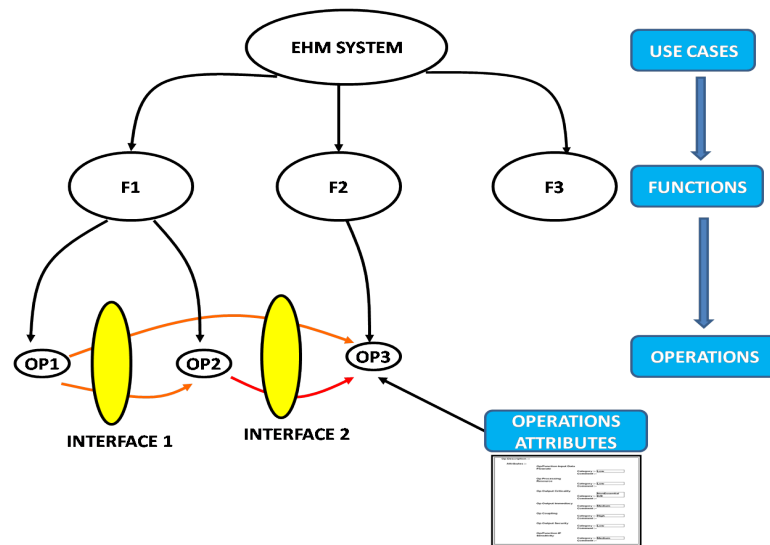


Figure 4.3: EHM system functional decomposition and operational attributes.

4.2.2 EHM System Functional Decomposition

The EHM system primary use cases are decomposed into several EHM system functions and these functions are further decomposed into a number of EHM system functional operations (OPs). Figure 4.3 illustrates the functional decomposition of the EHM system into functions and operations. After the functional decomposition the total number of EHM system functional operations are found to be 74 OPs. A systems modeling language (SysML) model (Friedenthal *et al.*, 2006; Hause, 2006; Weilkiens, 2011) for a baseline EHM system has been developed by system engineers using Enterprise Architect (EA) software (Sparx Systems, 2007). Figure 4.4 shows a list of EHM system functions and 74 functional operations. For each functional operation, several operational attributes are defined, which indicate the specific requirements of that operation in terms of its input data flowrate, processing power, immediacy and security etc., In the SysML model, for all the EHM system functional operations, these attribute requirements are expressed by system designers mainly in terms of different qualitative levels: 'high', 'medium' and 'low'. The operational attributes are described below:

4. EHM System Architecture Design

EHM System Functions	Operation Number	EHM System Functional Operations (OP)
F1 Engine Vibration Monitoring and Flight Deck Display	OP1	Acquire and process vibration/ speed sensor inputs
	OP2	Output Spectra, TOs and phase
	OP3	Format and scale for display
	OP4	Output Flight Deck Display
F2 Fan Trim Balance	OP5	Acquire and store Vibration TO, amplitude and Phase
	OP6	Interface to cockpit commands
	OP7	Calculate balance solution and output
F3 ETRECS Report Generation	OP8	Capture control system data
	OP9	Input ODMS data
	OP10	Capture vibration data
	OP11	Capture pressure data
	OP12	Perform event report collation
	OP13	Perform summary report collation
	OP14	Output event report - Small packet
	OP15	Output event report - Large Packet
	OP16	Output summary report - Small Packet
	OP17	Output summary report - Large Packet
F4 ETRECS Report Analysis	OP18	Input and format report
	OP19	Input relational data
	OP20	Process, Analyse and format data for display
	OP21	Output visual displays
	OP22	Apply auto-alerting logic
	OP23	Output alert
F5 Capture and Store Bulk Data	OP24	Capture vibration and pressure FFT data
	OP25	Capture control system data
	OP26	Synchronise data sources into bulk data stream
	OP27	Compress data stream
	OP28	Manage and store bulk data stream
F6 Oil Quantity Level Monitoring	OP29	Capture oil level data
F7 EHM System Enabling and Management	OP30	Retrieve archived data
	OP31	Display trend plots
	OP32	Archive flight reports for fleet trending
	OP33	Reprogram Off-board Functions
F8 Life Cycle Monitoring	OP34	Capture and transfer DACS
	OP35	Capture life cycle data file
	OP36	Calculate life consumed for each component
	OP37	Validate DAC life cycle calculations
	OP38	Output life consumed report
F9 Compressor Damage Monitoring	OP39	Capture P26 dynamic data
	OP40	Extract P26 TO and BB amplitudes
	OP41	Input P26 data into ETRECS
	OP42	Input P26 spectrum for bulk storage
F10 Capture Performance/ Control System Data	OP43	Acquire performance/control data - Basic EHM
	OP44	Acquire EMU Pressure Data
	OP45	Acquire Oil Quantity data
	OP46	Verify Basic EHM data
	OP47	Select data window (if required)
	OP48	Create Mode Reports
	OP49	Create Event Report
	OP50	Store reports for wireless transmission
	OP51	Output performance reports
F11 Engine Performance Health Analysis	OP52	Input and format snapshot reports
	OP53	Input Archived data
	OP54	Process reports
	OP55	Apply auto-alerting logic
	OP56	Output alert (if required)
F12 Advanced Vibration Acoustic Analysis	OP57	Input vibration spectral TO and phase data
	OP58	Perform feature detector (FD) analysis
	OP59	Output FD scores
	OP60	Extract harmonics and Broadband powers
F13 Oil Quality Monitoring	OP61	Capture oil debris count/acidity data
	OP62	Input ODMS into ETRECS summary report
F14 Fan Blade Damage Flutter Detection	OP63	Capture P160 dynamic data
	OP64	Extract P160 TO and BB amplitudes
	OP65	Input P160 data into ETRECS
	OP66	Analyse fan blade health
F15 On board Reprogramming	OP67	Prepare new program for upload
	OP68	Overwrite existing on-board configuration tables
	OP69	Overwrite existing on-board programs
F16 Bulk data Analysis	OP70	Download bulk data from Aircraft
	OP71	Receive and decompress bulk data
	OP72	Display formatted bulk data
F17 LRU Anomaly detection	OP73	Capture Control System data
	OP74	Analyse LRU data

Figure 4.4: EHM system functions and functional operations.

- **Input data flow rate:** an estimate of the input information / data flow rate (in Kbits/sec) required for the operation.
- **Processing power:** a measure of the processing power (in MFLOPS: mega floating-point operations per second) necessary to perform the operation.
- **Criticality:** a judgement of the level of safety criticality of the function output, from Level A (most safety critical to the engine safety) to Level E (non-critical function) (Blanquart *et al.*, 2012).
- **Immediacy:** a measure of how quickly the operation output needs to be acted upon by downstream OPs to fulfil the functional requirements.
- **Coupling:** a design judgement concerning whether there is a high sequence dependency on a preceding functional operation or external system events.
- **Security:** a judgement on the data security level required for the output of a functional operation.
- **IP sensitivity:** a judgement on the intellectual property sensitivity of a functional operation.
- **Flexibility:** a judgement on the level of modifiability of a functional operation in the future upgrades.

In order to enable the full functionality of the EHM system the OPs attribute requirements need to be satisfied. These operation attribute requirements are treated as the main rationale for deployment of the OPs on different physical architecture subsystem locations.

4.2.3 EHM System Physical Architecture Decomposition

The EHM system physical architecture for an aero engine is decomposed into a number of subsystems, as shown in Figure 4.5. It is mainly divided into on-board and off-platform systems. The on-board system is comprised of an on-engine subsystem and an on-aircraft subsystem. Further, the on-engine subsystem consists of an engine monitoring unit (EMU) and an engine electronic controller (EEC)

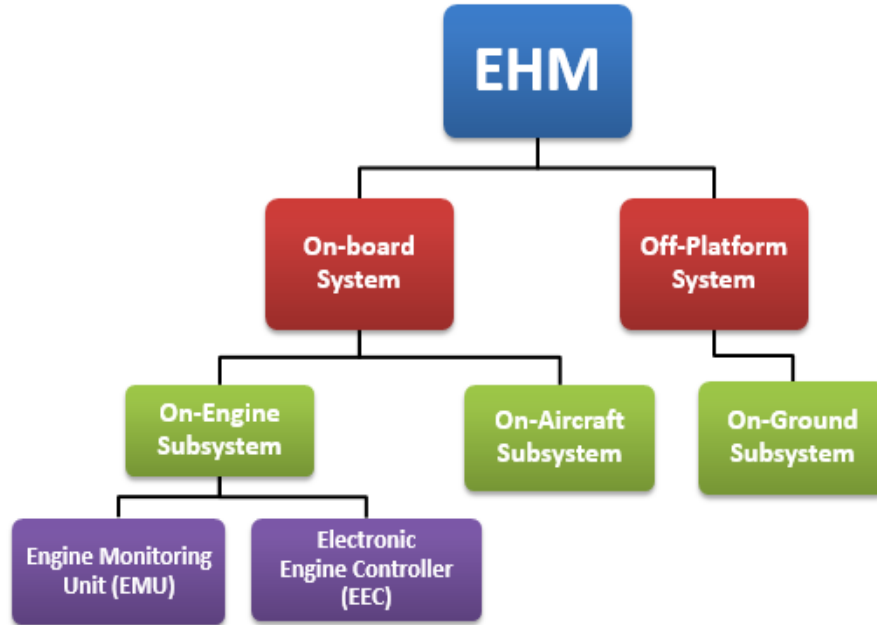


Figure 4.5: EHM system physical architecture decomposition and subsystems.

both mounted on the engine fan case. The EMU collects data from a number of sensors mounted on the various engine components and sub-systems of the engine. The EEC, along with the EMU, uses data from the sensors on engines to monitor numerous critical engine characteristics such as temperatures, pressures, speeds and vibration levels to ensure they are within known tolerances and to highlight when they are not. The EMU transmits applicable EHM system data to an on-aircraft system for reporting and storage on the maintenance server. From the aircraft system, potentially useful EHM system data can be compressed and transmitted in real-time to an on-ground station using an aircraft communications addressing and reporting system (ACARS) over satellite or radio.

Further detailed and complex data analysis necessary for monitoring the engine health status can be performed at the on-ground station using a substantial computational resource and a knowledge base accumulated from all the other engines in the fleet. The EHM system data analysis will highlight any changes in the engine component characteristics. Expert knowledge is used for diagnosis and

4. EHM System Architecture Design

Functional Operations Attributes	Physical Architecture Limitations			
	On-Engine		On-Aircraft	On-Ground
	EMU	EEC		
Interface Data Flowrate	16 Mbits/sec	32 Kbits/sec	10 Mbits/sec	100 bits/sec
Processing Capacity	1600 MFLOPS	200 MFLOPS	2000 MFLOPS	4000 MFLOPS (Limitless)
Criticality	Medium-Low (Level C/D/E)	High (Level A/B)	Low (Level D/E)	High-Medium-Low (Level A/B/C/D/E)
Immediacy	High	High	Medium	Low
Coupling	Coupling limitation on all locations is considered as Low. Coupling requirement is satisfied when coupled operations are deployed on the same physical architecture locations.			
Security	High	High	Low	High
IP Sensitivity	High	High	Low	High
Flexibility	Medium	Low	Medium	High

Figure 4.6: EHM system physical architecture subsystem limitations elicited from interviews with GTE System experts.

prognosis of the developing engine faults and to generate necessary maintenance reports.

4.2.4 EHM System Physical Architecture Subsystems Limitations

The EHM system physical architecture subsystems have certain limitations in terms of EHM system operational attributes. The subsystem limitations are elicited from interviews with gas turbine engine system experts. The limitations are given in Figure 4.6. The EMU is mounted on the engine fan case along with the EEC and acts as the centre of the EHM system. The close proximity of the EMU to the engine sensors helps to capture and process high-bandwidth signals without any degradation due to data transmission and reduces the significant weight of long cabling. The EMU contains two processing modules:

4. EHM System Architecture Design

(i) A signal processing module which takes the analog signals from a number of engine sensors and performs time to frequency domain transformation.

(ii) A main processing module which uses the frequency-domain data and performs the various EHM system feature-detection (FD) algorithms for detection of the engine component failures.

The combined processing capacity of the EMU is limited to 1600 MFLOPS. The main functionality of the EEC is to control the fuel flow to generate the required speed and thrust of the engine. A small portion of the EEC data flow and processing capacity is utilized for the purpose of the EHM system. EEC is a high ‘criticality’ (Level A) system, and for all the high ‘criticality’ functions it has to undergo extensive testing for safety certification. Hence it is a low ‘flexibility’ system for frequent modifications and upgrades. On the other hand, the EMU can accommodate medium and low ‘criticality’ functions and it is a medium ‘flexibility’ system. Both EMU and EEC can quickly respond to operational outputs and provide operator alerts. Both can be secured from third party visibility other than original equipment manufacturer (OEM) and hence they can provide high ‘security’ and ‘IP protection’ for the functions deployed on these subsystems. The limitation on the ‘coupling’ attribute is considered as ‘Low’ on all the physical subsystems. To satisfy the ‘coupling’ attribute requirement, all the operations having high coupling sequence dependency among them need to be deployed on the same physical architecture location. If the operations are separated by deploying on different locations their ‘coupling’ requirement is treated as not satisfied.

Data can be transmitted from on-engine to on-aircraft using a high bandwidth data interface system for significant processing, report generation and storing on the aircraft maintenance server. Since the on-aircraft system is also utilized for other aircraft monitoring functions not limited to engine related functions, it is considered as a medium ‘immediacy’, low ‘security’, low ‘IP-sensitivity’ and medium ‘flexibility’ system for the EHM system functionality. From the aircraft

system potentially important data is sent to the on-ground system with a very limited data flow rate of 100 bits/sec through the ACARS wireless transmission system. The ‘processing capacity’ of the on-ground system is considered as 4000 MFLOPS, although the OEM can provide limitless computing resources at the ground centre. The on-ground system can accommodate operations with all levels of ‘criticality’, and can also provide high ‘security’ and ‘IP protection’ for proprietary algorithms and techniques. The on-ground system is a high ‘flexibility’ system, which can be easily modified and upgraded.

4.2.5 EHM System Functional Operations Deployment

The next step in the EHM system architecture design is the deployment of the functional operations on to four physical architecture subsystems. For the current aero engine EHM system, there are 74 functional OPs with different operational attribute requirements in terms of their processing power, criticality, immediacy and flexibility etc., The OPs have high coupling and input data flow dependency on other functional operations. These operational attribute requirements for each of the functional operation is specified by the system experts in terms of ‘High’, ‘Medium’, and ‘Low’ qualitative attribute levels. The deployment process will need to account for a variety of factors to determine how best to allocate functionality to the various physical architecture subsystem components with in their resource limitations. Number of qualitative and quantitative criteria need to be formulated and models need to be developed in order to evaluate the performance of the deployment solutions. The functional operations deployment activity is a highly iterative process. For each functional operation there are 4 possible deployment options and for all the OPs there are $4^{74} = 3.6 \times 10^{44}$ total possible deployment options. This is a very large search space for manual exhaustive search for finding the best functional deployment solution for the EHM system. The design problem is discontinuous, non-convex over a large discrete variable space. There can be multiple local and global optimal solutions which can contain multiple deployment solutions resulting in similar performance in terms of design criteria. In such cases, optimization approaches have proven beneficial for efficiently searching the solution space to find the best solutions within the

feasible region.

In the next sections the EHM system architecture design problem is formulated as a multi-criteria optimization problem to find the optimal deployment solutions for EHM system functional operations within the constraints of physical subsystem resource limitations.

4.3 Multi-Criteria Optimization Formulation for EHM System Architecture Design Problem

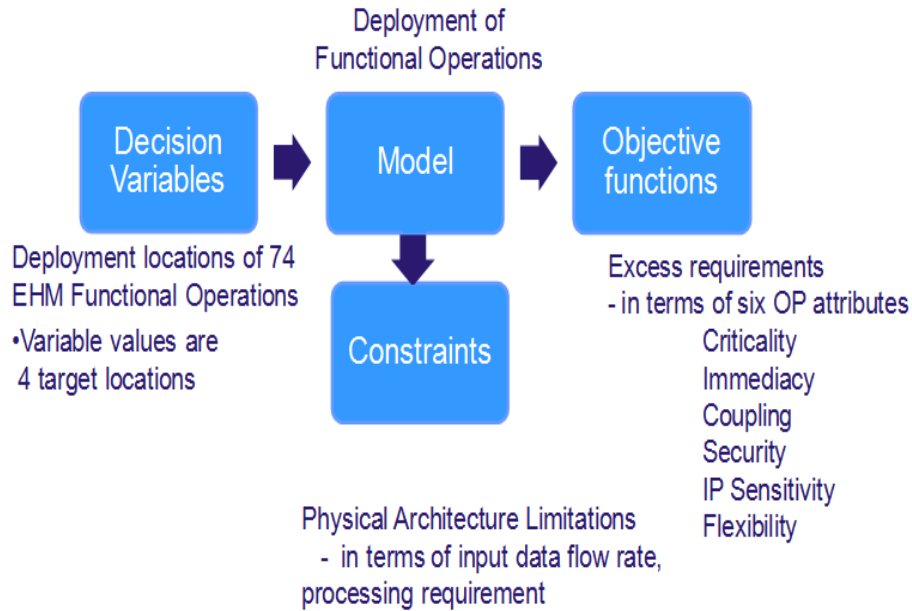


Figure 4.7: EHM system architecture multi-criteria optimization problem components: Decision variables, OPs Deployment Model, Constraints and Objective functions.

In the current EHM system architecture design, the EHM system functional operations (OPs) need to be optimally deployed onto the EHM system physical architecture components such that the OPs attribute requirements are satisfied within the limitations of the physical components. In order to find the optimal deployment of the EHM system functional operations, in this research study, the

EHM system architecture design problem is formulated as a multi-criteria optimization problem. Figure 4.7 shows the EHM system multi-criteria optimization problem components: decision variables, constraints and objective functions along with the model of the EHM system architecture which are further described in the following subsections.

4.3.1 Decision Variables

The EHM system functional operations need to be deployed onto the four target physical architecture component locations. Hence the locations of the deployment for each of the functional operations are treated as the decision variables for optimization. As there are 74 EHM system functional operations, 74 decision variables are chosen to represent the deployed location number of each functional operation. Each of these decision variables can have a value of 1, 2, 3 or 4, representing a physical architecture location: ‘1: EMU’, ‘2: EEC’, ‘3: on-aircraft’, and ‘4: on-ground’, respectively.

4.3.2 Constraints

The EHM system has certain resource limitations for the physical architecture components in terms of their capability to handle the functional operations attribute requirements. The EHM system functional deployment solutions should both satisfy all the limitations of these physical resources and enable full functionality of the EHM system. Hence, the hardware limitations of ‘data flow rate’ and ‘processing capacity’ on each of the four physical architecture component locations are imposed as eight constraints in the optimization process. In the optimization process, the functional operations deployed at each location are separated and grouped. For a subsystem, the total data flow requirement is computed for those operations receiving data through interfaces from the operations deployed on the other subsystems. The requirement of an operation receiving data from a preceding operation deployed on the same subsystem is neglected in the computation. The total data flow rate and processing power required for all the operations deployed in each subsystem location are computed and compared with the corresponding component limitations/ constraints. If these requirements

are satisfied, then the solution is considered as a feasible solution, otherwise it is treated as an infeasible solution.

4.3.3 Objective Functions

In the current design process, in order to facilitate the search for finding the best deployment solutions that satisfy the EHM system functional OP attribute requirements, violations are permitted in terms of the remaining six OP attributes on physical architecture components. These attribute requirement violations are treated as “excess requirements” in that OP attribute. The total excess requirements in terms of the six operational attributes: ‘**Criticality**’, ‘**Immediacy**’, ‘**Coupling**’, ‘**Security**’, ‘**IP sensitivity**’ and ‘**Flexibility**’, are considered as six individual objective functions to be minimized in the optimization process. Since the design problem has more than three objective functions, it is classified as a many-objective optimization problem (Fleming *et al.*, 2005).

4.3.4 Optimization Problem Formulation

The multi-criteria optimization problem for the EHM system architecture design can be formulated as below:

$$\min_{\mathbf{x}} \mathbf{f}(\mathbf{x}) = (f_1(\mathbf{x}), \dots, f_6(\mathbf{x})), \quad (4.1)$$

$$\text{where } f_k(\mathbf{x}) = \sum_{i=1}^{74} E^2_{ik}(\mathbf{x}), \text{ for } k = 1, \dots, 6$$

$$\text{subject to } \sum_{i=1}^{74} d_{ir} \leq D_r, \quad r = \{1, 2, 3, 4\} \quad (4.2)$$

$$\sum_{i=1}^{74} p_{ir} \leq P_r, \quad r = \{1, 2, 3, 4\} \quad (4.3)$$

$$\mathbf{x} = [x_1, \dots, x_i, \dots, x_{74}], \quad x_i \in \{1, 2, 3, 4\} \quad (4.4)$$

where, $\mathbf{f}(\mathbf{x})$ is a vector of six objective functions, each representing the sum

4. EHM System Architecture Design

of squares of total excess requirements $E_{ik} = (OP \text{ requirement} - resource \text{ limit})$ of 74 EHM system functional OPs. D_r and P_r are the constraint limitations of data flowrate and processing capacity on the four physical architecture locations, and, d_{ir} and p_{ir} are individual attribute requirement measures of each operation deployed at the corresponding location. The decision variables, x_i , can have values $\{1, 2, 3, 4\}$ to represent the deployment locations of the corresponding operation.

4.3.5 Integration of the Architecture Models into the Optimization Framework

An EHM system architecture for the aero gas turbine engine was developed by Rolls-Royce system engineer Tanner (2010), as a SysML model using Enterprise Architect (EA) software (Sparx Systems, 2007). The SysML model has information concerning all the EHM system functions and their operational attribute requirements. The information available in the SysML model has to be transferred to the platform used by the optimization algorithm.

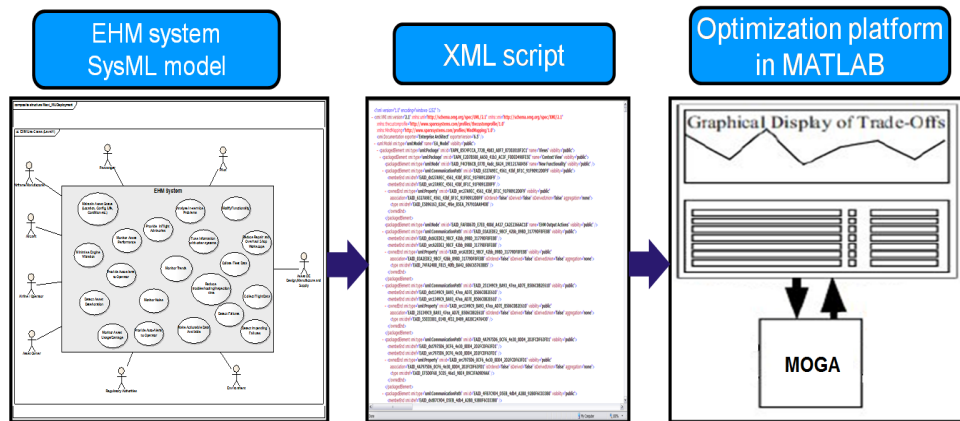


Figure 4.8: Integrating the system architecture SysML model to optimization platform.

In order to integrate the SysML model into the optimization platform, the SysML model is exported as an XML script file. From this XML script file all the attributes information of the functional operations is decoded and converted into a format suitable for use within the optimization platform using an XML toolbox

for Matlab (Molinari, 2004). The process of integrating the SysML model and importing data into the optimization platform in Matlab is depicted in Figure 4.8. Data for all the functional operations and their attribute levels are exported from the SysML model to the optimization platform in Matlab. Then, for ease of manipulation in the optimization, the qualitative attribute levels for the operations, i.e. ‘High’, ‘Medium’, and ‘Low’, are transformed to numerical values 9, 4, and 1 (3^2 , 2^2 and 1^2), respectively. These values are selected from commonly used numerical representations for attribute levels in engineering applications using qualitative and quantitative criteria (Andrews & Moss, 2002; Henley & Kumamoto, 1981). From the operations sequence diagrams in the SysML model, an input data dependency matrix and a coupling dependency matrix for all operations are created by identifying the preceding operations/sensor interfaces from which the operations are receiving the input data and their corresponding coupling dependency on the other functional operations. This data is further used for the evaluation of objective functions values for the candidate architecture alternatives and for finding the optimal architecture solutions for the EHM system.

4.3.6 Evaluation of Criteria Values for a Sample EHM System Deployment Solution

The process for calculation of constraints and objective function values for a sample candidate solution is shown in Figure 4.9. For the EHM system multi-criteria optimization problem, a sample solution consists of 74 decision variables having the deployment locations of the EHM system functional operations. Initially the OPs are grouped according to their deployment locations, then the constraint values and objective function values are computed.

In the illustrated solution in Figure 4.9, 39 OPs are deployed on the EMU, 2 OPs are deployed on the EEC, 18 OPs are deployed on the on-aircraft system and 15 OPs are deployed on the on-ground system. The two operations deployed on the EEC: ‘OP34- Capture and transfer life cycle DACS (Data Accumulation and Counting System) data’ and ‘OP59 - Perform feature detector analysis’, are shown in Figure 4.9. For these OPs the total requirements for all attributes are calculated. A table showing the OP attribute requirements and physical architec-

4. EHM System Architecture Design

ture limitations for each of the attributes is given. For the two OPs on the EEC, the total interface data flow rate requirement is computed to be 5.5 Kbits/sec, and the total processing resource requirement is 53 MFLOPS. It can be seen that, the two constraints: (2) data flow requirement on the EEC and (6) processing resource requirement on the EEC are satisfied, as the total requirements are within the constraint limitations on the EEC. For the criticality attribute, one low criticality (Level D/E = 1) operation (OP59) is deployed on the EEC which is a high criticality (Level A = 9) subsystem. All functions deployed on the EEC have to undergo extensive testing for safety certification. For this reason, the deployment of low criticality OPs on EEC is considered as an excess requirement objective function. The square of the difference in requirement levels $[(1 - 9)^2 = 64]$ is computed as the criticality excess requirement. For immediacy, security and IP sensitivity attributes, all the OPs requirements are satisfied. For the coupling attribute, OP34 has high coupling with the EEC for capturing DACS data so its requirement is satisfied with this deployment. However the OP59 has high coupling sequence dependency with another OP57 which is deployed on EMU. Hence its coupling requirement is not satisfied. The square of the difference in requirement levels $[(9 - 1)^2 = 64]$ is computed as the coupling excess requirement. For the flexibility attribute, one medium level (4) OP and one high level (9) OP are deployed on the EEC which is a low flexibility(1) resource. The sum of squares of differences in requirement levels $[(4 - 1)^2 + (9 - 1)^2 = 73]$ is computed as the flexibility excess requirement. Similar calculations are carried out for the other resources: EMU, on-aircraft and on-ground. Excess requirements on all of these resources for the OP attributes are aggregated to form individual objective functions.

For the current sample solution, the total values for the eight constraints (shown in green) and six objective functions (shown in blue) are given in Figure 4.9. It can be observed that all the constraint values are within the physical architecture limitations given in the Section 4.2.4. However, the remaining attribute requirements of OPs in terms of ‘Criticality’, ‘Immediacy’, ‘Coupling’, ‘Security’, ‘IP sensitivity’ and ‘Flexibility’ are not completely satisfied. In the optimization process, the optimizer tries to minimize the excess requirement ob-

jective function values and find the optimal deployment solutions for the EHM system.

4.4 Multi-Criteria Optimization of EHM System Architecture Design Problem

It can be seen from the previous section that the EHM system architecture design optimization problem has 74 decision variables which can take any one value from {1, 2, 3 and 4} representing the respective deployment location of the EHM system functional operation. The total number of possible deployment solutions are $4^{74} = 3.6 \times 10^{44}$. This is a very large discontinuous search space and becomes impractical for the exhaustive search to find the optimal solutions. In such cases, iterative MOEAs are employed which can search through the large design search space with a set of population and converge to optimal solutions space using the information contained in best solutions found in previous iterations.

The EHM system many-objective optimization problem is solved using the proposed multi-criteria optimization framework, in the MATLAB environment. For solving the current EHM system architecture optimization problem, the following parameters are set in MOGA, which are found to be suitable for the current problem:

- population size = 500,
- maximum number of generations = 500,
- probability of crossover = 0.7, and
- probability of mutation = 0.1

Stochastic universal sampling selection, single point binary crossover and mutation genetic operators are used in the MOGA algorithm. In this study elitism is incorporated in MOGA by maintaining an archive of non-dominated solutions of fixed size, which will keep the best non-dominated solutions found so far in

4. EHM System Architecture Design

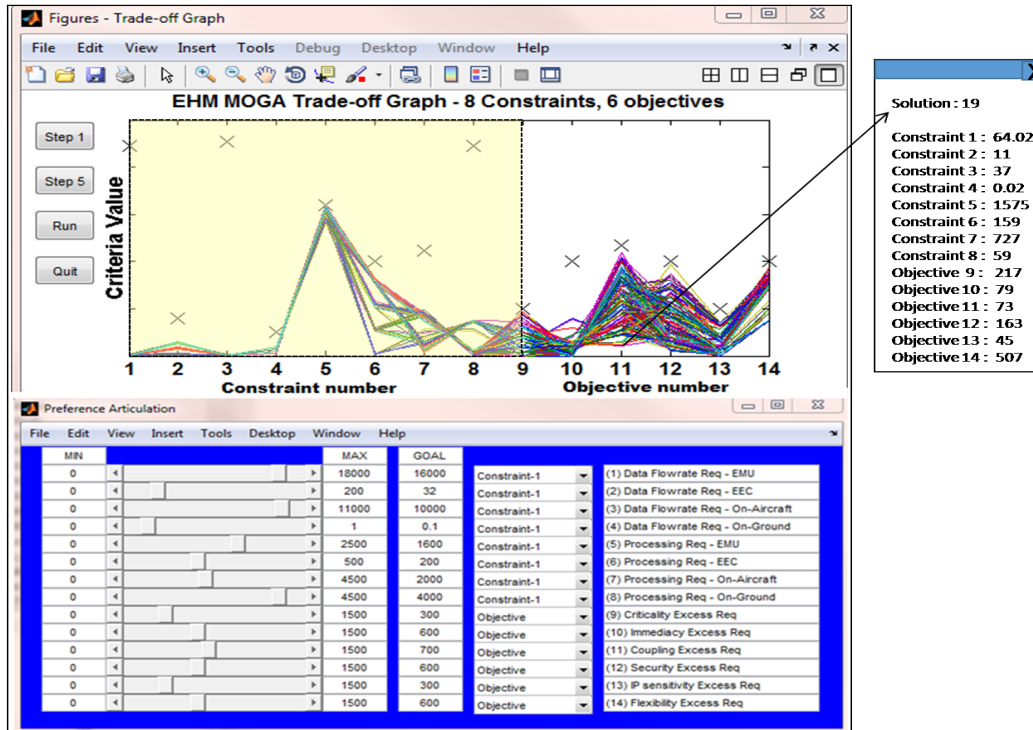


Figure 4.10: EHM system architecture solutions in MOGA parallel coordinates trade-off graph with preference articulation.

all the generations. With a view to increasing the confidence in the Pareto solutions, MOGA was run for 25 times using a different seed for the random number generator in each run and various performance metrics were also evaluated.

In the optimization process, a candidate architecture solution has 74 decision variables each representing the deployment resource location number for its corresponding operation. Then the operations are separated and grouped according to their deployment location. The solution should satisfy all the constraints. The total data flow rate and processing resource requirements on each location, i.e., values of eight constraints, are estimated and compared with the corresponding constraint limitations. For the six qualitative criteria: criticality, immediacy, coupling, security, IP sensitivity and flexibility, if the requirements of operations are not satisfied, then the excess requirements are estimated for all OPs, using the numerical transformation described in the Section 4.3.5. The total excess

4. EHM System Architecture Design

requirements in each criteria are considered as six individual objective functions to be minimized in the optimization process.

Due to the many objective functions and constraints in the optimization problem, non-dominated solutions predominate in each generation. This decreases the selection pressure towards the true Pareto optimal surface, thus reducing the performance of the optimization algorithm. Using the PPA technique, the decision maker's preferences are expressed progressively to reduce the objective search space, and steer the optimization into region of interest. The additional preference operator used by PPA helps restore selection pressure. Several Pareto optimal solutions are obtained for the EHM system architecture design using MOGA. The obtained Pareto optimal solutions for the EHM system architecture design are shown in Figure 4.10. In MOGA, the trade-off values of all constraints and objective functions are shown in a "parallel coordinates" graph. In the case of many-objective optimization, the parallel coordinates graph aids the visualization of all constraints and objective functions values on a single plot. In Figure 4.10, the top window shows the MOGA trade-off graph for the 14 criteria of the current EHM system architecture design, where criteria 1 to 8 are constraints on each physical architecture location in terms of data flow rate and processing resource. Criteria 9 to 14 are the six objective functions: 'criticality', 'immediacy', 'coupling', 'security', 'IP sensitivity' and 'flexibility' excess requirements. In the Figure 4.10, for clarity, the constraints and objective functions are separated using a dashed line and constraints are shown in shaded region and objectives are shown in unshaded region. On the 'y-axis' corresponding criteria values are displayed. Each connected line in the trade-off graph represents a Pareto optimal solution for the EHM system architecture design. With a mouse click on a solution, a pop-up notification window will display the corresponding solution number and all the constraints and objective functions values for that Pareto solution.

The bottom window shows the "preference articulation" facility for the EHM system architecture design problem. In this window, all the constraints and objective functions of the EHM system multi-criteria optimization problem are listed.

4. EHM System Architecture Design

A decision maker can set goal values for each objective by moving the sliders between maximum and minimum bounds at any time during the optimization process. For the EHM system architecture design, the goal points representing the decision maker's preferences for each of the objectives are marked with an "x" in the trade-off graph. As the decision maker exercises progressive articulation of preferences, the proposed MOGA optimizer directed the search towards the preferred region of interest in the feasible objective space and minimized the objective functions values within the specified goals and obtained the Pareto optimal solutions shown in the trade-off graph. In the trade-off plot, it can be observed that, crossing lines between criteria 9 and 10, criteria 10 and 11, and criteria 11 and 12, demonstrate that the objectives 'criticality', 'immediacy', 'coupling' and 'security' are in conflict with each other, while concurrent lines between criteria 12 and 13 demonstrate that the objectives 'security' and 'IP sensitivity' are in relative harmony with each other. A limitation of the parallel coordinates representation is that only adjacent objectives can be easily compared. However, in MOGA, there is a facility to interactively switch the order of representation of the objectives.

It can be seen from the trade-off graph, that the data flow requirements of deployed OPs on EMU (1) and on-aircraft (3), the processing resource requirements on-aircraft (7) and off-board (8) are far below the goals/limitations. These constraints are thus satisfied very easily. Whereas, the data flow requirements on EEC (2) and on-ground (4), the processing resource requirements on EMU (5) and EEC (6) are close to the goals/limitations. These constraints are tightly satisfied. These constraints can be identified as the most significant design constraints ("hot spots"). There are several Pareto optimal architecture solutions with zero values for one of the objective functions: criticality (9), immediacy (10) and IP sensitivity (13). Hence, in these deployment solutions, the the operations attribute requirements for any one of the attributes criticality, immediacy and IP sensitivity can be completely satisfied. It is shown that other OP attributes, coupling (11) and flexibility (14) requirements, are not satisfied completely in any Pareto optimal architecture solution within the given constraints of resource limitations.

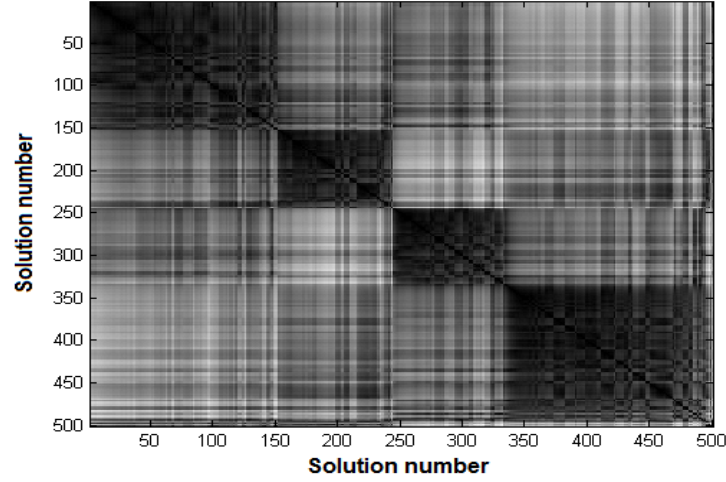


Figure 4.11: VAT clusters image with four clusters (dark blocks) in EHM system non-dominated solutions.

4.5 Cluster Analysis and Pareto Estimation

Cluster analysis is performed using the VAT algorithm on the decision variable space for the non-dominated solutions obtained from the MOGA optimizer for the EHM system architecture design. The reordered image of dissimilarity matrix of non-dominated decision vectors highlighting the clusters in the solutions is shown in Figure 4.11. It can be observed that there are four clusters present in the decision space of the non-dominated solutions for the EHM system architecture design. However, there are several gray shady regions on the image of clusters. This represents that the clusters are not clearly separated in the high dimensional decision space. After identifying the different cluster of solutions, the Pareto estimation method is applied to each cluster of solutions. Here several radial basis function neural network (RBFNN) models are trained with objective vectors as inputs and decision vectors as outputs for each cluster of solutions and in order to find the mapping between Pareto objective vectors and decision vectors. Then, we tried to estimate 300 solutions in each cluster using the RBFNN models and observed that several estimated solutions are becoming in-feasible due to very low data flow rate constraint on the ground subsystem. The Pareto estimation method is able to increase the density of the available Pareto solu-

4. EHM System Architecture Design

tions to a total number of 824 solutions for the EHM system architecture design. Figure 4.12 shows the objective space trade-off graph with the four clusters of non-dominated solution generated from the PE method. Here, the plot of the decision variable space is not shown, as there are 74 decision variables in decision space and the plot is not clearly observable with 74 dimensions. Figure 4.13 shows the VAT image of dissimilarity matrix of non-dominated solutions after the Pareto estimation, highlighting the four clusters of Pareto optimal solutions.

The cluster analysis revealed several important insights into the deployment of the EHM system functional operations. In these clusters, most of the solutions are mapped to the same objective vectors on the Pareto front indicating multi-modal Pareto solutions. The following observations are made from the architecture solutions for the EHM system OP deployment.

In cluster 1 architecture solutions (black), the EHM system functional operations related to engine vibration monitoring, fan balance, ETRECS (Engine Trending and Event Capture System) report generation and analysis and compressor damage monitoring functions are deployed on the EMU, which has close proximity to the numerous sensors mounted on the engine and has the sufficient processing resource for the analysis. The high criticality operation for capturing life cycle DAC (Direct Accumulation Counting) system data, Oil quantity and quality monitoring operations are deployed on the EEC. The functional operations related to the output of the ETRECS event report, mode report, summary report and performance reports and bulk data storage are deployed on the aircraft system. Operations for EHM system management, engine performance and health analysis, advanced vibration analysis, bulk data analysis and LRU (Line Replaceable Units) data analysis are deployed on the ground subsystem, where more processing resource is available for complex analysis required for these functions.

In cluster 2 architecture solutions (yellow), EHM system functional operations of capturing vibration, pressure and temperature sensors data and performance measuring operations are deployed on the EMU. High criticality life cycle mon-

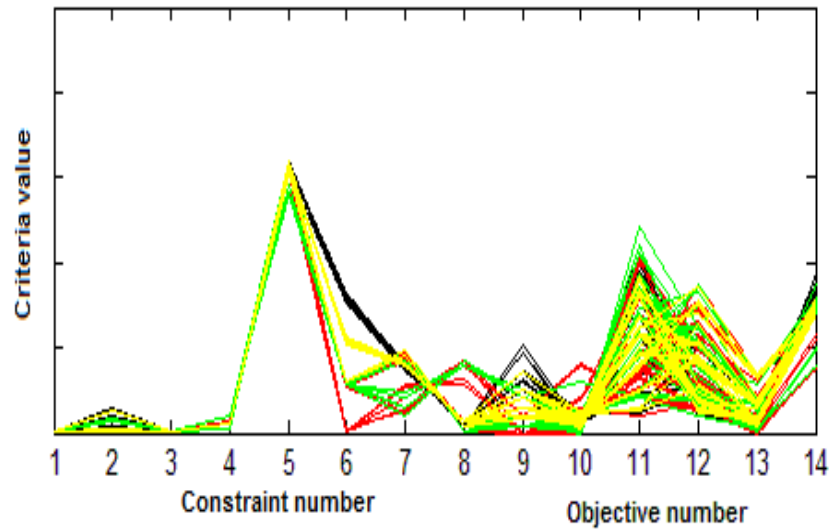


Figure 4.12: Clusters of non-dominated solutions after the Pareto estimation for EHM system architecture design.

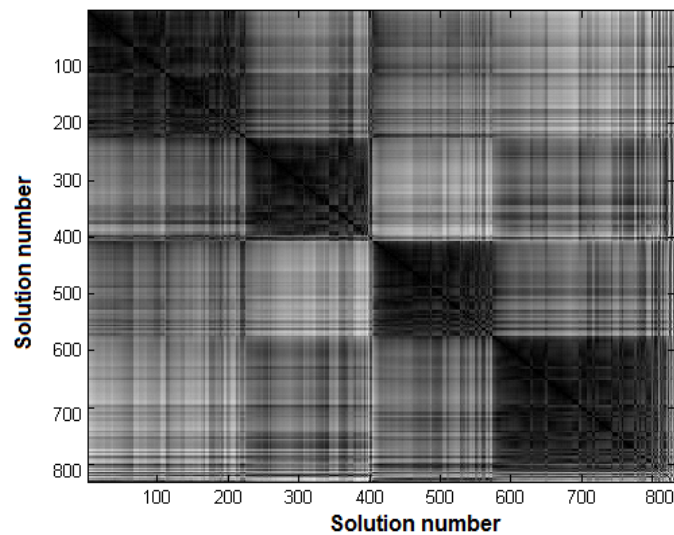


Figure 4.13: VAT Image of clusters of non-dominated solutions after the Pareto estimation for EHM system architecture design.

4. EHM System Architecture Design

itoring operations are deployed on the EEC. Operations for engine performance report generation and bulk data storage are deployed on the aircraft system. Operations for ETRECS report generation input ODMS (Oil Debris Monitoring System) data operation and capture control system data for LRU anomaly detection are deployed to the aircraft system. Here the deployment of these operations changed from EMU to the aircraft subsystem. However, their attribute requirements are satisfied both on EMU and aircraft system. Operations for EHM system enabling and management, advanced vibration analysis, bulk data analysis and LRU data analysis are deployed on the ground subsystem.

In cluster 3 architecture solutions (green), the EHM system functional operations related to engine vibration monitoring, fan balance and compressor damage monitoring functions, advanced vibration analysis, are deployed on the EMU. The high criticality operation for capturing life cycle DAC system data, Oil quantity and quality monitoring operations deployed on the EEC. The functional operations related to the output of the ETRECS event report, mode report, summary report and performance reports, through wireless transfer and bulk data storage are deployed on the aircraft system. Operations for ETRECS report analysis and life-cycle data analysis, are deployed from the EMU to the ground system. EHM system enabling and management, bulk data analysis and LRU data analysis are deployed on the ground subsystem.

In cluster 4 architecture solutions (red), the deployment of the EHM system functional operations similar to the cluster 3 architecture solutions except few operations. Here, operations for input ODMS data, capturing control system performance and verifying basic EHM data and capture LRU control system data are deployed on the EMU compared to the solutions in architecture 3 in which these operations are deployed on the aircraft. These operations have low level attribute requirements and they are satisfied in both the EMU and the aircraft subsystems. They are resulting in equal objective values similar to the solutions in cluster 3. They are representing the multi-modal Pareto solutions mapping to the same objective vectors. However, the data flow rate requirement and processing resource constraint values are not equal and they are changing

according to the number of operations deployed on the four subsystems for these solutions.

4.6 Statistical Performance Evaluation

To evaluate the performance of the proposed optimization framework, the EHM system architecture design optimization problem is solved 25 times with different random initial populations in each run and various test metrics are computed for the results obtained. For the purpose of comparison, the EHM system architecture design optimization problem is also solved using NSGA-II (Deb *et al.*, 2002a) and Omni-optimizer (Deb & Tiwari, 2008), 25 times, by setting the optimization parameter values similar to the MOGA. Random search is also performed for the EHM system architecture design problem by taking 500 x 500 random solution vectors in the decision space and evaluating the constraints and objective functions for these solutions. Out of these solutions, very few solutions are found to be within the constraint limitations on data flow rate and processing capacities of resources. Non-dominated solutions are then obtained from these solutions by performing non-dominated sorting. The process is repeated for 25 times for the Random search.

The non-dominated solutions obtained from one of the runs by NSGA-II, Omni-optimizer, MOGA and random search are shown in Figure 4.14. Here only the values of six objective functions are plotted on parallel co-ordinates graphs showing the comparison of non-dominated solutions obtained from the four optimizers. It can be observed that both the optimizers NSGA-II and Omni-optimizer are not able to converge to the Pareto front compare to the solutions obtained from the MOGA optimizer. Where as, the random search generated few solutions in the feasible space satisfying the constraint limitations, out of them very few solutions are found to be non-dominated. However, they are sub-optimal solutions compared to the solutions obtained from the NSGA-II, Omni-optimizer and MOGA.

Various test metrics employed to measure the performance of the optimizers are described below:

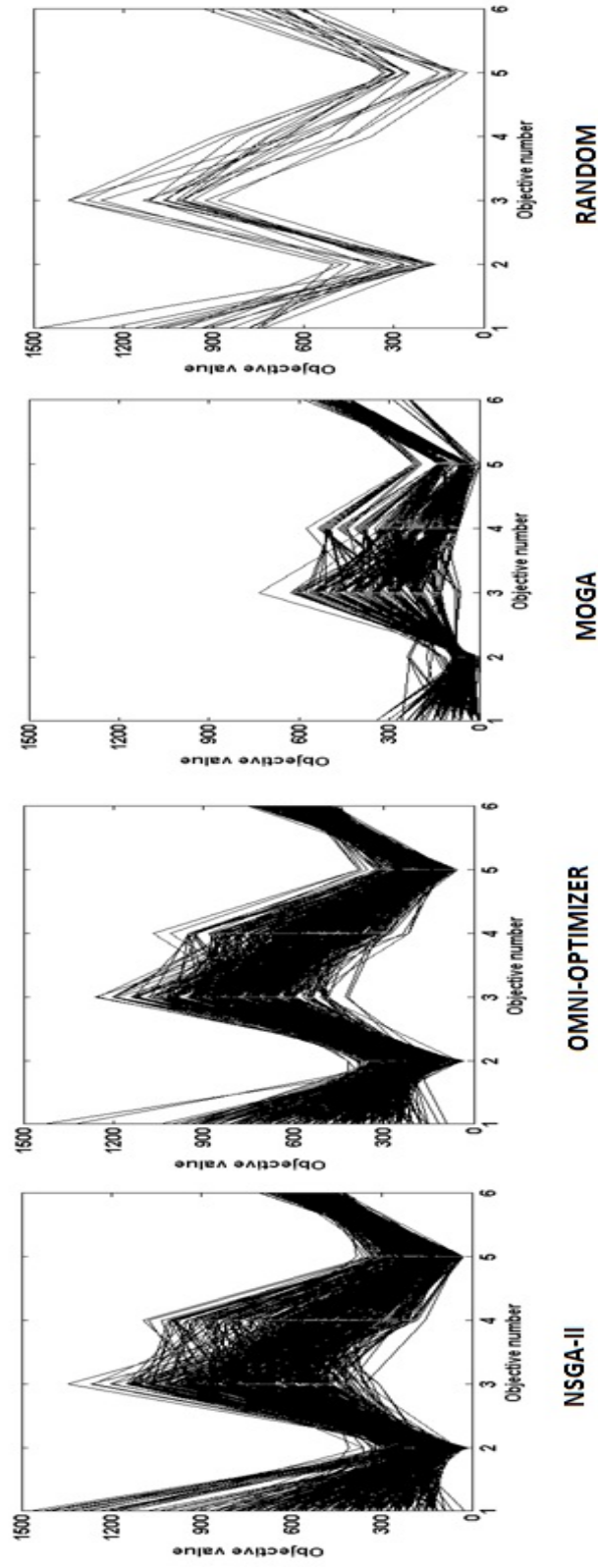


Figure 4.14: Comparison of non-dominated solutions obtained from NSGA-II, Omni-Optimizer, MOGA and RAN-
DOM search methods.

- **Hypervolume(HV)** is measured as the total volume of the k -objective dimensional region enclosed by the obtained non-dominated solutions and a dominated reference point. Among the solutions sets from different optimizers the higher the hypervolume value, the better the performance of the optimization algorithm. In this research work, the hypervolume metric is computed using the method by Fonseca *et al.* (2006). For the current minimization optimization problem of the EHM system architecture design, the dominated reference point in terms of the six objective functions is considered as (1500, 1500, 1500, 1500, 1500, 1500) solution, which used for normalizing the non-dominated objective vectors obtained from the optimizer. The hypervolume (HV) metric is computed as the total volume enclosed by a set of normalized non-dominated solutions A , obtained in each run of the optimizer and a reference point \bar{z} , which is a vector of (1, 1, 1, 1, 1, 1).

$$\text{HV}(A) = \mathbf{VOL}\{A, \bar{z}\}, \quad (4.5)$$

- **Inverted Generational Distance (IGD)**, introduced in Van Veldhuizen (1999),

$$D(A, \mathcal{P}^*) = \frac{\sum_{s \in \mathcal{P}^*} \min\{\|A_1 - s\|_2, \dots, \|A_N - s\|_2\}}{|\mathcal{P}^*|}, \quad (4.6)$$

where $|\mathcal{P}^*|$ is the cardinality of the set \mathcal{P}^* , which is true Pareto front (PF) and A is an approximated non-dominated set of the PF obtained from optimization algorithm. The IGD metric measures the distance of the elements in the set A from the nearest point of the true Pareto front \mathcal{P}^* .

In this case, since the true Pareto front (PF) for the EHM system architecture design problem is unknown, we have considered the best non-dominated solutions found so far out of all combined solutions from 25 runs to be the Pareto front \mathcal{P}^* and the non-dominated set obtained in each run of the optimizer as A .

- **Mean Nearest Neighbour Distance,**

$$S(A) = \frac{\sum_{i=1}^{|A|} d_i}{|A|}, \quad (4.7)$$

where d_i is,

$$d_i = \min_j \{\|F(x_i) - F(x_j)\|_2\}.$$

$$S_R(A, \mathcal{P}^*) = \frac{S(A)}{S(\mathcal{P}^*)}. \quad (4.8)$$

The ratio of the mean nearest neighbour distance metric $S_R(A, \mathcal{P}^*)$ is a measure of the density of solutions compared to the true Pareto front of the problem. The ratio of this metric compares a set of solutions obtained from one optimizer run with the true Pareto front solution set.

- **Coverage Metric (C-Metric)**

$$C(A, B) = \frac{|\{u \in B | \exists v \in A : v \preceq u\}|}{|B|}, \quad (4.9)$$

The coverage metric is introduced in the work done by Zitzler & Thiele (1999). The C-metric compares two sets of solutions and gives an indication of the ratio of the number of solutions in one set being dominated by the other set of solutions. A C-metric value $C(A, B) = 0$ is interpreted as: there is no solution in A that completely dominates any solution in B ; $C(A, B) = 1$ is interpreted as all the solutions in B are dominated by at least one solution in A .

Table 4.1 and Table 4.2, show the minimum, mean and standard deviation in the values of the hypervolume (HV) metric, IGD index $D(A, \mathcal{P}^*)$, mean nearest neighbour distance ratio $S_R(A, \mathcal{P}^*)$ and C-metric $C(\mathcal{P}^*, A)$, estimated for the non-dominated solutions obtained in 25 runs for the EHM system architecture design problem using the optimizers: NSGA-II, Omni-optimizer, MOGA and Random search method. The variation in the four performance metric values computed for the non-dominated solutions obtained by the optimizers in 25 runs, is shown

4. EHM System Architecture Design

Table 4.1: Hypervolume (HV) and $D(A, \mathcal{P}^*)$ values of the obtained solutions by the optimizers, where A is the obtained set in each run, \mathcal{P}^* is the considered PF.

Optimizer	Hypervolume (HV)			IGD $D(A, \mathcal{P}^*)$		
	min	mean	std	min	mean	std
NSGA-II	0.2666	0.4075	0.0754	0.1547	0.2649	0.0672
Omni-Optimizer	0.2586	0.4636	0.1078	0.0789	0.2066	0.0678
MOGA	0.7701	0.8142	0.0248	0.0043	0.0287	0.0120
Random	0.0216	0.1260	0.0676	0.8010	0.8451	0.0283

Table 4.2: $S_R(A, \mathcal{P}^*)$ and C -Metric values of the solutions obtained by the optimizers, where A is the obtained set in each run, \mathcal{P}^* is the considered PF.

Optimizer	ESSm $S_R(A, \mathcal{P}^*)$			$C(\mathcal{P}^*, A)$		
	min	mean	std	min	mean	std
NSGA-II	1.4141	1.5937	0.0668	0.2937	0.4431	0.0879
Omni-Optimizer	1.1837	1.4368	0.0987	0.3762	0.4829	0.0377
MOGA	0.8918	0.9316	0.0246	0.0292	0.0971	0.0339
Random	0.2330	0.3499	0.0653	0.9431	0.9731	0.0164

in Figure 4.15 and Figure 4.16 using box plots representation. In these box plots, the bottom and top lines of the box are represent first quartile (25th percentile) and third quartile (75th percentile), and the line inside the box represents the second quartile (the median) of the distribution. The whiskers extend to the most extreme min and max data points representing the range of the distribution, without considering the outlier data points. The outliers are plotted as individual points outside the whiskers.

- The hypervolume metric scores for the optimizers indicate that MOGA is able to enclose more hypervolume by converging towards the Pareto front consistently in all the 25 runs with less variation. However, NSGA-II and Omni-optimizer have moderate hypervolume scores implying that they are converging to sub-optimal solutions and the variation is higher compare to MOGA optimizer. Random search got a very low score for hypervolume, due to very few solutions are found to be non-dominated in the feasible

4. EHM System Architecture Design

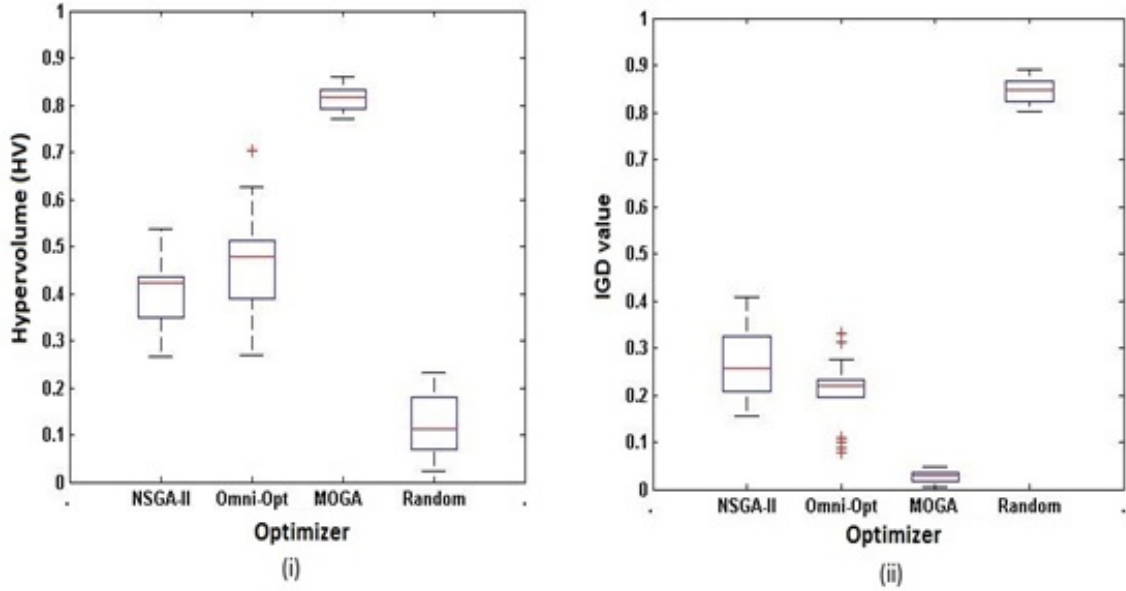


Figure 4.15: Box plots of the (i) HV and (ii) IGD metric values computed for the solutions obtained in 25 runs by the optimizers.

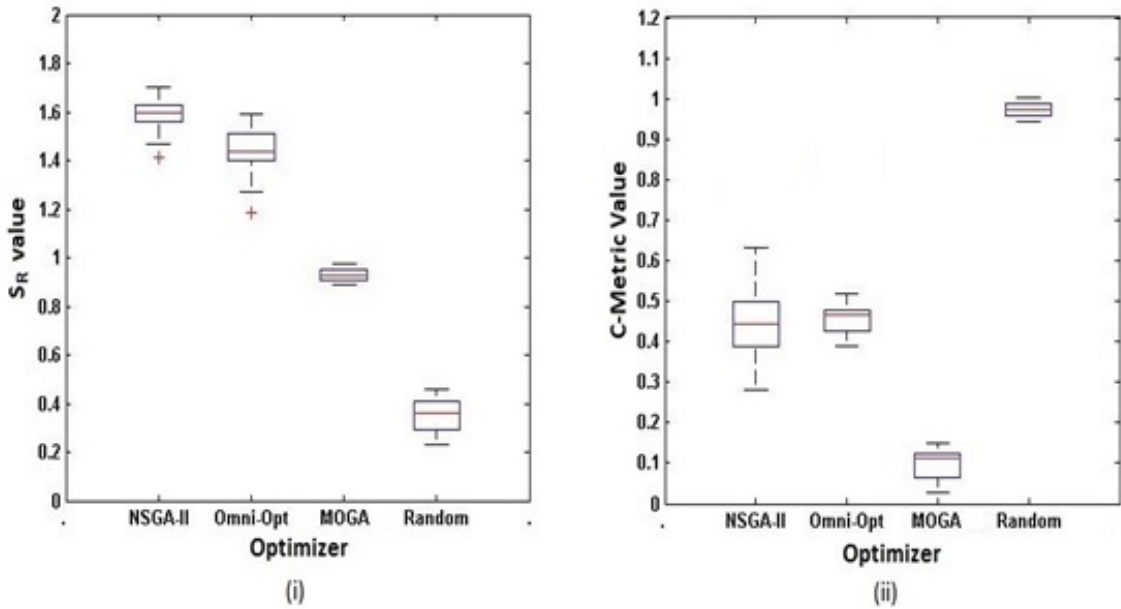


Figure 4.16: Box plots of the (i) S_R and (ii) C-metric values computed for the solutions obtained in 25 runs by the optimizers.

4. EHM System Architecture Design

space which are not converged to the Pareto front.

- The IGD index scores indicate the relative distance of obtained non-dominated solutions to the best Pareto front obtained overall. The IGD index scores for the NSGA-II and Omni-optimizer imply that the solutions obtained are not close the Pareto front, hence they are sub-optimal solutions. There are several outliers in the IGD index scores for the Omni-optimizer, implying that the optimizer is not consistent in converging to the Pareto front. The low scores of IGD index $D(A, \mathcal{P}^*)$ with less variation for the MOGA indicate that the obtained non-dominated sets in each run are very close to the best Pareto front obtained overall. Therefore, for this EHM system architecture design problem, the proposed methodology is consistent in producing solutions that are close to the PF in every run. Random search got high scores for IGD index with less variation, due to very few solutions in the feasible space.
- The $S_R(A, \mathcal{P}^*)$ ratio values indicate that the obtained solutions in each run from MOGA optimizer have a good distribution and diversity compare to solutions obtained from the NSGA-II and Omni-optimizer with respect to the Pareto front. The low variation in metric values for the three optimizers indicate that they are able to find solutions with good distribution and diversity in all the runs.
- The C-metric values for NSGA-II and Omni-optimizer indicate that 25 – 50% solutions are being dominated by Pareto front solutions. Low values of the C-metric for MOGA and their low variation imply that there are very few solutions that are being dominated by solutions in the Pareto front in all the 25 runs. The scores for the random search indicate that almost all the solutions are being dominated by solutions in the Pareto front.

The scores of the performance metrics and the box plots indicate that the proposed optimizer is able to find well distributed non-dominated solutions with good repeatability in the decision maker’s preferred region close to the true Pareto front.

4. EHM System Architecture Design

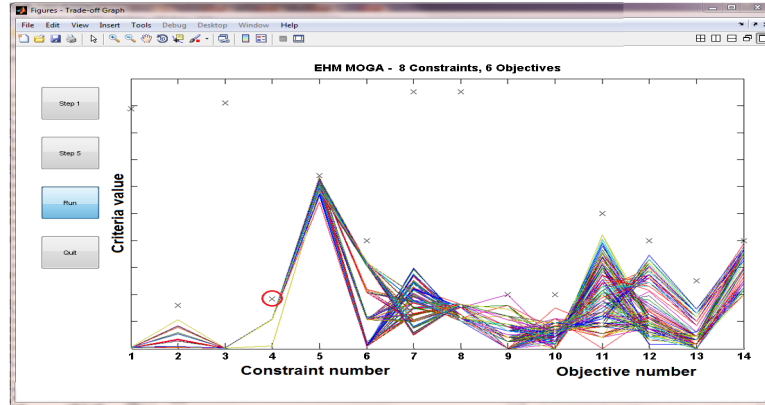


Figure 4.17: Trade-off solutions with increased dataflow rate to the on-ground station.

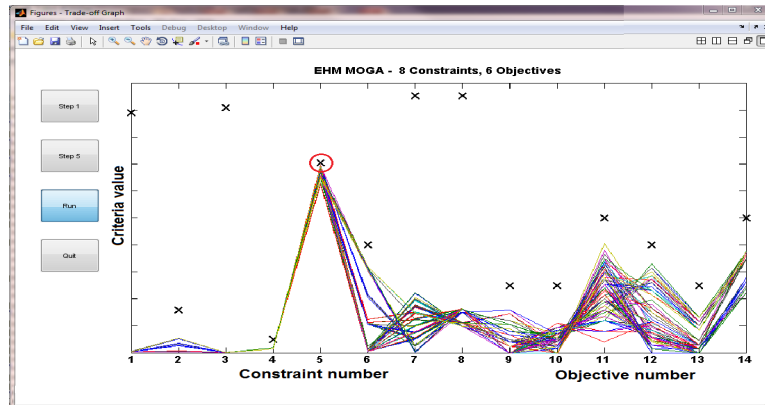


Figure 4.18: Trade-off solutions with increased processor limit on the EMU.

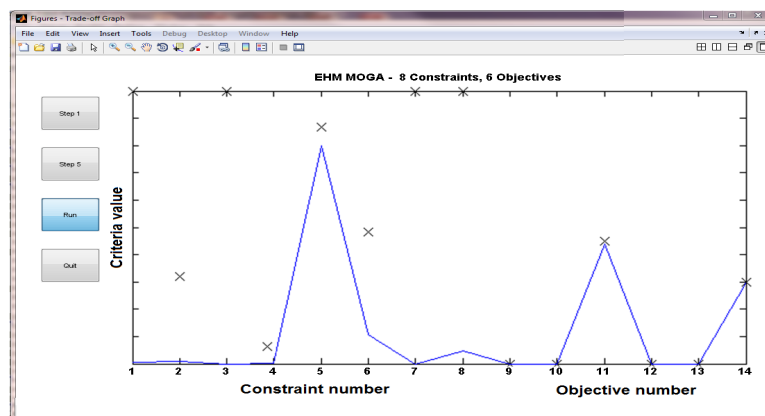


Figure 4.19: Optimal deployment solution with four objectives satisfied.

4.7 Exploring What-If Design Scenarios

There is additional value in the proposed approach. The Progressive preference articulation (PPA) technique in the MOGA optimizer enables the decision maker to explore different architecture design scenarios, such as improved processor technology on the EMU and improved wireless transmission rate between on-engine and on-ground systems. This is possible by interacting with constraints and specifying different goal/limitation settings. It can be observed from Figure 4.10, that the constraints on: data flow rate to on-ground (4), processing capacity on EMU (5) and processing capacity on EEC (6), are narrowly satisfied. By increasing the goal values for these constraints, the decision maker can explore future (“what-if”) architecture design scenarios and analyze prospective performance improvements. Here, in the first attempt, the constraint goal value for the ACARS (Aircraft Communications, Addressing and Reporting System) data flow rate to the on-ground system is increased (50%). The trade-off solutions obtained after running the MOGA for a further 50 generations are shown in Figure 4.17. It is observed that solutions are obtained with zero values (completely satisfied) for coupling objective (11) and with a slight improvement in flexibility objective (14).

In the second attempt, the constraint goal value for the processing capacity on the EMU is increased (20%), and the data flowrate for the on-ground system decreased to the actual constraint value. Figure 4.18 shows the trade-off solutions obtained after running the MOGA for a further 50 generations. It can be seen that solutions are obtained with a zero value for the security objective (12) and no significant improvement in the coupling objective (14). Out of these solutions one interesting solution is obtained with zero values for four objectives: criticality (9), immediacy (10), security (12) and IP sensitivity (14), as shown in Figure 4.19. This means that in this deployment solution all OP requirements in criticality, immediacy, security and IP sensitivity attributes are completely satisfied, when the EMU processing capacity is increased.

4.8 Summary

In this chapter, system architecture design process using the proposed multi-criteria optimization framework is described for an aero engine health management (EHM) system. The system architecture design process involves various steps, which include, stakeholder identification, requirement capture, functional definition and decomposition, physical architecture decomposition into subsystems, function deployment and optimization. The EHM system architecture design problem is formulated as a multi-criteria optimization problem involving eight constraints in terms of resource limitations and six objectives in terms of operational attribute excess requirements and 74 discrete decision variables representing the operations deployment locations. The proposed multi-criteria optimization framework is employed to solve the EHM system architecture design optimization problem. The optimizer generated a well distributed family of Pareto optimal trade-off solutions for the EHM system architecture design. Cluster analysis identified four groups of Pareto optimal solutions which provide valuable insight into EHM system architecture design trade-offs. For each cluster, several additional Pareto optimal solutions are estimated using the Pareto estimation method with less computation expense. Through this design process, it is revealed that it is not possible to fully satisfy all attributes for the EHM system, while observing the given constraints, thereby highlighting the value of a multi-criteria approach. Various architecture design scenarios, such as hardware upgrades to data input rates and processor capacities, are explored by changing the goal values of constraints. It is shown that improved system performance achieved for the new specification. Performance of the proposed method is assessed by evaluating test metrics and comparing the results with other popular optimizers. The results of test metrics demonstrated the superior performance of the proposed optimization framework.

The next chapter presents an uncertainty analysis performed for the EHM system architecture design for estimating the sensitivities of objectives considering variations in design parameters caused by underlying assumptions .

Chapter 5

Uncertainty Analysis and Robust Optimization of EHM System Architecture Design

5.1 Introduction

The proposed multi-criteria optimization approach for system architecture design does not consider the variations in the design parameters and uncertainties in the design assumptions made by experts. The obtained Pareto optimal architecture solutions are sensitive to these variations in the design parameters.

In this Chapter, an uncertainty analysis using an efficient polynomial chaos expansions (PCE) based approach is presented for the system architecture design optimization. The PCE approach requires a smaller number of samples to estimate the uncertainty metrics such as mean, variance and standard deviation values for the objectives with more accuracy when compared with the standard Monte Carlo sampling approach. This helps in speeding up the uncertainty analysis and results in a robust optimization process with reduced computational expense. A robustness metric is introduced for representing the sensitivity of the objectives in the optimization formulation. In the optimization process for each candidate solution the criteria evaluations over the number of samples from the

design of experiments, are executed using the parallel processing facility available in the MATLAB software. Using the PPA technique in the MOGA optimizer, preferences are expressed and robust Pareto optimal architecture solutions for the EHM system architecture design are obtained. The system architecture design process is validated using a baseline EHM system architecture implemented on a working aero gas turbine engine.

5.2 Uncertainties in EHM System Architecture Design

In the multi-objective optimization studies the main goal is to find the Pareto optimal solutions. However, in real-world design optimization, a decision maker may not be interested in the Pareto optimal solutions if these solutions are very sensitive to the variations in the design parameters. In such cases, the decision maker is interested in finding the solutions which are less sensitive to the variations in the design parameters and uncertainties in the design assumptions made by experts. These solutions are called “robust solutions”.

In the current EHM system architecture design optimization, the operational attribute requirements for the 74 EHM functional operations are specified by design experts in terms of qualitative levels of ‘*low*’, ‘*medium*’ and ‘*high*’. In order to integrate the EHM system architecture models into an optimization platform and to evaluate the six objective functions, the functional OP attribute requirements in terms of ‘*low*’, ‘*medium*’ and ‘*high*’ qualitative levels need to be transformed to suitable numeric values. In this work, we have selected numerical values of 1, 4, and 9 for transforming the qualitative requirement levels; ‘*low*’, ‘*medium*’ and ‘*high*’ respectively.

The obtained Pareto optimal architecture solutions from the optimization process in the Section 4.4, are specific to the given model specifications. They are sensitive to the design assumptions made by experts and to the considered numerical conversion values for the EHM operational attribute requirements in

the design. In order to select an architecture solution for the EHM system, one can perform uncertainty analysis on the solutions and find out solutions having least sensitivity to the variations in the design parameters caused by the underlying assumptions.

5.3 Robustness Metric

In this work, in order to find solutions with low sensitivity to the uncertainties, a robustness metric R_F (similar to Erfani & Utyuzhnikov, 2012) is introduced as an additional objective function in the system architecture design. The robustness metric is described as:

$$R_F = \frac{1}{ku} \sum_{i=1}^k \sum_{j=1}^u \frac{\sigma f_i}{\sigma p_j} \quad (5.1)$$

Here, the robustness metric is computed as the sum of the ratios of the standard deviation in objectives, f , with respect to the standard deviation in uncertain design parameters, p , normalized by total number of objectives, k and the total number of uncertain parameters, u .

The robustness metric value for a candidate architecture solution gives an indication of the level of sensitivity to variations in the uncertain parameters. A high value for the robustness metric indicates that the solution is very sensitive to the parameter variations and a low value indicates that the solution is less sensitive to the parameter variations.

In the optimization process, finding the Pareto optimal solutions having low values for the robustness metric objective function, leads to robust architecture solutions which are less sensitive to the variations present in the model, arising from the underlying assumptions.

5.4 Polynomial Chaos Expansion Approach

In order to calculate the robustness metric, the uncertain quantities of the objective functions such as mean, variance and standard deviation are estimated. The most widely used method for uncertainty analysis is Monte Carlo (MC) sampling method. There are a number of Monte Carlo sampling methods described in the literature (Helton & Davis, 2000; Kalos & Whitlock, 2008; Rubinstein & Kroese, 2011). The main advantage of the MC methods is that they are simple to implement and they treat the model as a black-box. In the MC sampling method, the random samples of uncertain variables are taken from their probability distribution and used for computing the model outputs. Then the statistical parameters for the output are estimated. The main drawback of Monte Carlo methods is that they require a large number of samples in order to estimate accurate mean and standard deviation values for the output.

Another sampling technique with faster convergence than the MC method is Latin Hypercube Sampling (LHS) (McKay *et al.*, 2000). In this method, the samples of uncertain parameters are spread across the entire range of the parameter into intervals with equal probability, such that there is only one sample in each interval. Since LHS covers the whole range of the uncertain parameters, the accuracy of output statistics is better than that of the Monte Carlo method.

There exists another methodology called Polynomial Chaos Expansion (PCE) (Wiener, 1938) which is more efficient than both the Monte Carlo and Latin hypercube sampling methods, for estimating uncertain quantities of model outputs. In this methodology, the uncertain variables are expanded using a specific orthogonal polynomial series based on the probability distribution of the uncertain variables, such as Hermite polynomials for a normal distribution and Legendre polynomials for a uniform distribution. Then the uncertain quantities are estimated using a truncated expansion of these polynomials. The method is popularly used in computationally expensive aerospace engineering applications (Najm, 2009; Perez, 2008; Poles & Lovison, 2009; Sudret, 2008).

5. Uncertainty Analysis and Robust Optimization

In the polynomial chaos expansion method, a function $f(x)$ is expanded in an infinite series using orthogonal polynomials $p_i(x)$. Then the coefficients of the polynomial series α_i are determined in order to estimate the means, variance and standard deviations, thus:

$$f(x) = \sum_{i=0}^{\infty} \alpha_i p_i(x), \quad (5.2)$$

$$\mu = \alpha_0, \quad (5.3)$$

$$\sigma^2 = \sum_{i=1}^{\infty} \alpha_i^2 \|p_i(x)\|^2. \quad (5.4)$$

In the general case, the expansion is truncated at a certain order, h , and coefficients, α_i , are determined using either Galerkin projections (Fletcher, 1984) or chaos collocation method (Villadsen & Michelsen, 1978) by exploiting the orthogonality of the polynomials.

The equations associated with the truncated polynomial expansion are:

$$f(x) \cong \sum_{i=0}^h \alpha_i p_i(x), \quad (5.5)$$

$$\mu \cong \alpha_0, \quad (5.6)$$

$$\sigma^2 \cong \sum_{i=1}^h \alpha_i^2 \|p_i(x)\|^2. \quad (5.7)$$

The chaos collocation method described in Perez (2008); Poles & Lovison (2009), is employed in this work. The polynomial coefficients, α_i , are estimated from the evaluations of the function $f(x)$ at selected orthogonal collocation points (Villadsen & Michelsen, 1978) and minimizing the sum of the squares of the differences between $f(x)$ and polynomial chaos expansion of order h .

$$\alpha_i : \min_{\alpha_i} \sum_{s=1}^N \left[f(\bar{x}_s) - \sum_{i=1}^h \alpha_i p_i(\bar{x}_s) \right]^2, \quad (5.8)$$

$$\{\bar{x}_1, \dots, \bar{x}_N\},$$

The number of samples N , considered for estimating the polynomial coefficients is

$$N = 2 \times \frac{(h + u)!}{h!u!} \quad (5.9)$$

where h is order of the polynomials and u is number of uncertain variables.

5.5 Uncertainty Analysis and Robust Optimization of EHM System Architecture Design

In order to perform uncertainty analysis for the EHM system architecture design, the uncertain design parameters and their probability distributions need to be identified. In the EHM system architecture design problem, the OP attribute requirements for the 74 EHM functional operations are specified by the industrial system design experts in terms of ‘*low*’, ‘*medium*’ and ‘*high*’ qualitative levels. These OP attribute requirements are transformed to suitable numerical values of 1, 4 and 9 respectively, for easy integration into the optimization process and for evaluation of the objective functions.

The three levels of attribute requirements are considered as three uncertain parameters in the current uncertainty analysis for the EHM system architecture design. The variation in the choice of the values for the attribute requirement levels by the design experts is considered to follow a Gaussian probability distribution. For the three uncertain parameters, mean values are taken as assumed numerical transformation values 1, 4 and 9 and variance is considered to be 10% of the mean value, which is proportionate to the level of attribute requirement. Here, the variation is greater for the high level of attribute requirement and the

5. Uncertainty Analysis and Robust Optimization

variation is lower for the low level attribute requirement.

In the polynomial chaos expansions approach the order of the polynomials is selected to be 5 and the number of uncertain parameters is set equal to 3. From these values the required number of samples for uncertainty analysis is computed (equation 5.9) and found to be 112 samples. The number of samples needed in the PCE method is much less when compared with the Monte Carlo sampling methods.

In the EHM system architecture design optimization process, for each candidate architecture solution the six objective function values along with the additional robustness metric objective value need to be estimated. For this purpose, the PCE method is integrated with the proposed optimization framework. Here, the PCE method will generate a design of experiments by selecting 112 sample values over the range of Gaussian distributions for the three uncertain parameters. For each candidate architecture solution, the criteria values are required to be evaluated for the 112 samples with different parameter values. Here, we have utilized the parallel processing facility available in MATLAB (MathWorks, 2010) for evaluating the criteria values for the 112 samples on 8 parallel processing MATLAB pool sessions, in order to reduce the total elapsed computation time. Then from the criteria values evaluated for the 112 samples using the PCE approach, the uncertain quantities of EHM system objective functions, such as mean, variance and standard deviation, are estimated. The robustness metric objective for the candidate architecture solutions is then computed as the sum of ratios of the standard deviations of six objective functions with respect to standard deviations of uncertain parameters (equation 5.1). The process is repeated for all of the candidate architecture solutions in the robust optimization process and the non-dominated Pareto optimal solutions are obtained.

For the EHM system architecture design optimization, the non-dominated Pareto optimal solutions obtained after the uncertainty analysis are shown in Figure 5.1. Here the trade-off of the Pareto optimal architecture solutions is shown with respect to the six objectives of EHM system OP attributes and the

5. Uncertainty Analysis and Robust Optimization

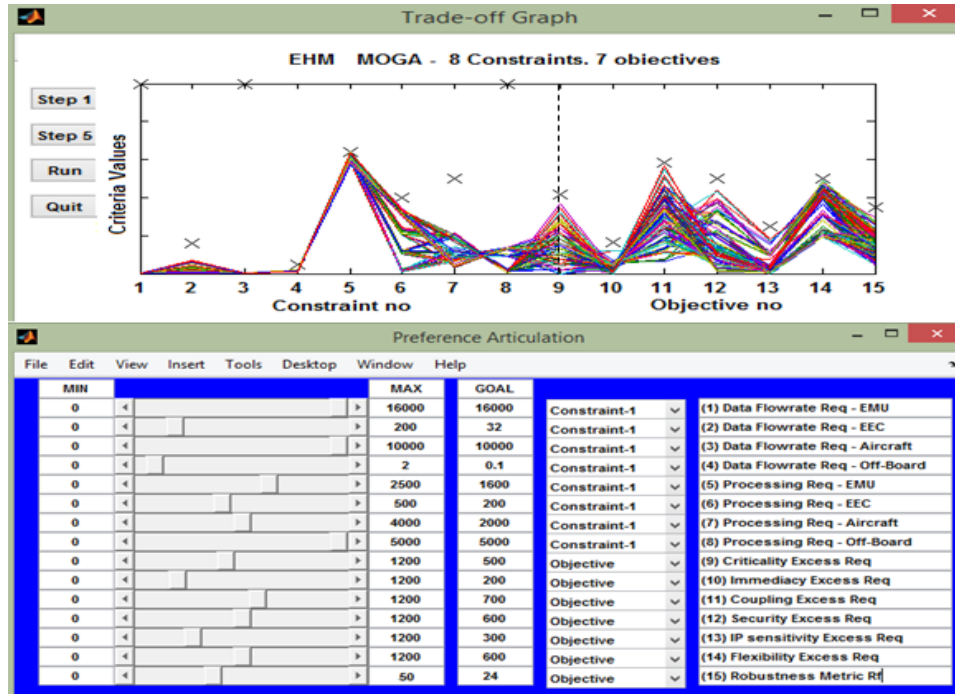


Figure 5.1: EHM system architecture Pareto optimal solutions with robustness metric values.

robustness metric values of the solutions. Here, the values of the robustness metric objective indicate the level of sensitivity of the Pareto optimal architecture solutions with respect to the variations in the uncertain parameters. The Pareto optimal solutions having higher robustness metric values are more sensitive to the variations, and the solutions having low robustness metric values are less sensitive to the uncertain parameter variations.

In order to select a robust Pareto optimal architecture solution for the EHM system architecture design, the decision maker can specify preferences for the robustness metric value using the PPA technique, indicating the level of sensitivity acceptable for the architecture design. In Figure 5.2, initially a preference of zero goal value for the ‘Criticality excess requirement’ objective is expressed using the slider in the preference articulation window. Then a second preference on the robustness metric objective is expressed until the lowest robustness metric value of the remaining solutions is isolated. This process eliminates most of the Pareto

5. Uncertainty Analysis and Robust Optimization

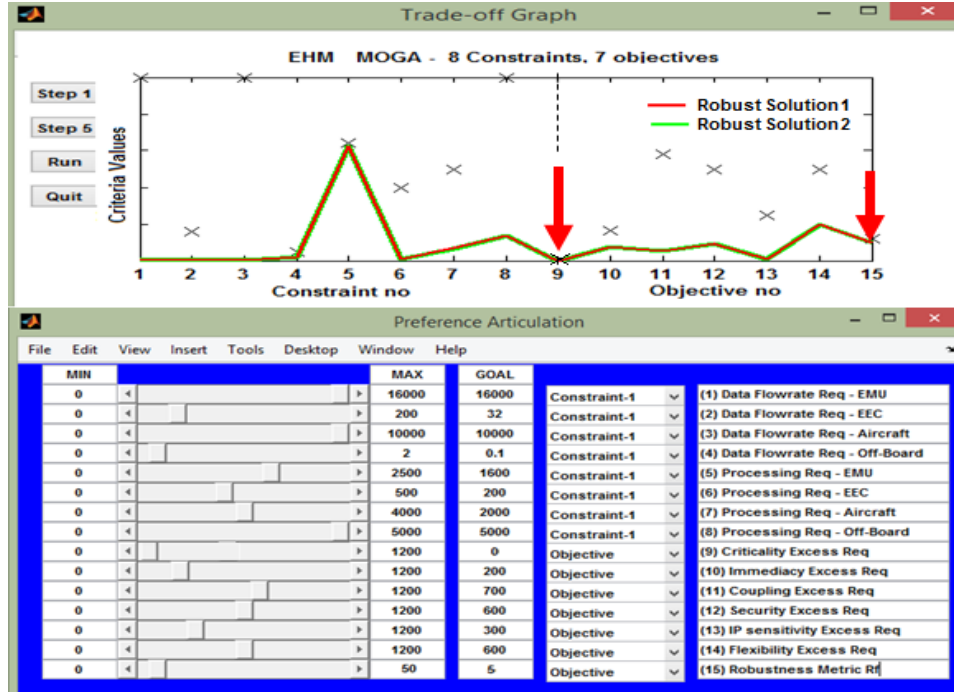


Figure 5.2: Robust EHM system architecture Pareto optimal solutions obtained by expressing preferences on criticality and robustness metric objectives.

optimal solutions and two robust Pareto optimal solutions remain that satisfy the decision maker's preferences. Criteria scores for the robust multi-modal Pareto optimal architecture solutions 1 and 2 are given in Figure 5.3. The two robust Pareto optimal solutions have equivalent objective function values (from objective 9 to 15), indicating that they are multi-modal Pareto optimal solutions. (c.f. Section 3.6)

Deployment of the 74 EHM functional operations in the obtained robust Pareto optimal EHM system architecture solution 1 is presented in Figure 5.4. In this architecture solution, out of 74 functional operations, 38 operations are deployed on the EMU, 1 operation is deployed on the EEC, 11 operations are deployed on the aircraft subsystem and 24 operations are deployed on the ground subsystem. The list of EHM system functions and associated 74 operations presented in Figure 4.4 illustrates the relationship of each OP to its corresponding EHM system function. In this architecture solution, the functional operations

5. Uncertainty Analysis and Robust Optimization

Constraints and Objective functions	Values
(1) Data flow rate requirement - EMU (Kbits/sec)	64.00
(2) Data flow rate requirement - EEC (Kbits/sec)	0.50
(3) Data flow rate requirement - Aircraft (Kbits/sec)	6.00
(4) Data flow rate requirement - on-Ground (Kbits/sec)	0.04
(5) Processing requirement - EMU (MFLOPS)	1569.00
(6) Processing requirement - EEC (MFLOPS)	3.00
(7) Processing requirement - Aircraft (MFLOPS)	265.00
(8) Processing requirement - on-Ground (MFLOPS)	683.00
(9) Criticality excess requirement	0.00
(10) Immediacy excess requirement	90.48
(11) Coupling excess requirement	65.09
(12) Security excess requirement	112.42
(13) IP-sensitivity excess requirement	9.49
(14) Flexibility excess requirement	236.41
(15) Robustness metric	4.93

Robust Solution 1

Constraints and Objective functions	Values
(1) Data flow rate requirement - EMU (Kbits/sec)	52.50
(2) Data flow rate requirement - EEC (Kbits/sec)	0.50
(3) Data flow rate requirement - Aircraft (Kbits/sec)	7.00
(4) Data flow rate requirement - on-Ground (Kbits/sec)	0.04
(5) Processing requirement - EMU (MFLOPS)	1566.00
(6) Processing requirement - EEC (MFLOPS)	3.00
(7) Processing requirement - Aircraft (MFLOPS)	268.00
(8) Processing requirement - on-Ground (MFLOPS)	683.00
(9) Criticality excess requirement	0.00
(10) Immediacy excess requirement	90.48
(11) Coupling excess requirement	65.09
(12) Security excess requirement	112.42
(13) IP-sensitivity excess requirement	9.49
(14) Flexibility excess requirement	236.41
(15) Robustness metric	4.93

Robust Solution 2

Figure 5.3: Criteria scores for robust EHM system architecture Pareto optimal solutions.

related to engine vibration monitoring, fan balance, ETRECS report generation and compressor damage monitoring functions are deployed on the EMU, which has close proximity to the numerous sensors mounted on the engine. The high criticality operation for capturing life cycle DAC (Direct Accumulation Counting) system data (OP34) is deployed on the EEC. The functional operations related to the output of the ETRECS event report, mode report, summary report and performance reports, through wireless transfer and bulk data storage are deployed on the aircraft system. Operations for ETRECS report analysis, advanced vibration analysis, life-cycle data analysis, bulk data analysis and LRU data analysis are deployed on the ground subsystem.

Deployment of the 74 EHM functional operations in the robust architecture solution 2, is presented in Figure 5.5. In this architecture solution, out of 74 functional operations, 36 operations are deployed on the EMU, 1 operation is deployed on the EEC, 13 operations are deployed on the aircraft subsystem and 24 operations are deployed on the ground subsystem. When compared with the robust architecture solution 1, the operational deployment is the same except four operations. In this architecture solution, three operations; OP9-Input ODMS data,

OP45-Acquire oil quantity data and OP73-capture control system data for LRU (Line Replaceable Units) anomaly analysis, are deployed on the Aircraft. The operation OP42-Input compressor pressure P26 spectrum for bulk storage is deployed on the EMU. Even though the deployment of these operations is different, their attribute requirements are satisfied in both EMU and Aircraft subsystems. Hence, the objective functions values of the two robust Pareto optimal solutions are the same. The constraints of the data flow rate and processing resource requirement, however, are different. These two solutions represent two multi-modal Pareto solutions for the EHM system architecture design.

5.6 Validation of the EHM System Architecture Design Optimization

In this section, the EHM system architecture design process using the proposed multi-criteria optimization methodology is validated using a baseline aero engine EHM system architecture provided by Rolls-Royce system engineers (Tanner, 2010), which is currently implemented on a working gas turbine engine. The validation process consists of evaluating criteria scores for the baseline EHM system architecture using the proposed multi-criteria formulation in Section 4.3. The baseline EHM system architecture and deployment of the 74 EHM functional operations, is shown in Figure 5.6. In the baseline architecture, out of 74 functional operations, 36 operations are deployed on the EMU, 2 operations are deployed on the EEC, 12 operations are deployed on the aircraft subsystem and 24 operations are deployed on the ground subsystem.

In order to evaluate the multiple criteria scores for the baseline EHM system architecture, the values of 74 decision variables representing the deployment location of EHM functional operations, are captured from the Figure 5.6. The criteria scores are computed, for the eight constraints representing the total data flow rate requirements and processing resource requirements and six objectives of excess requirements in terms of the six operational attributes: ‘Criticality’, ‘Im-

5. Uncertainty Analysis and Robust Optimization

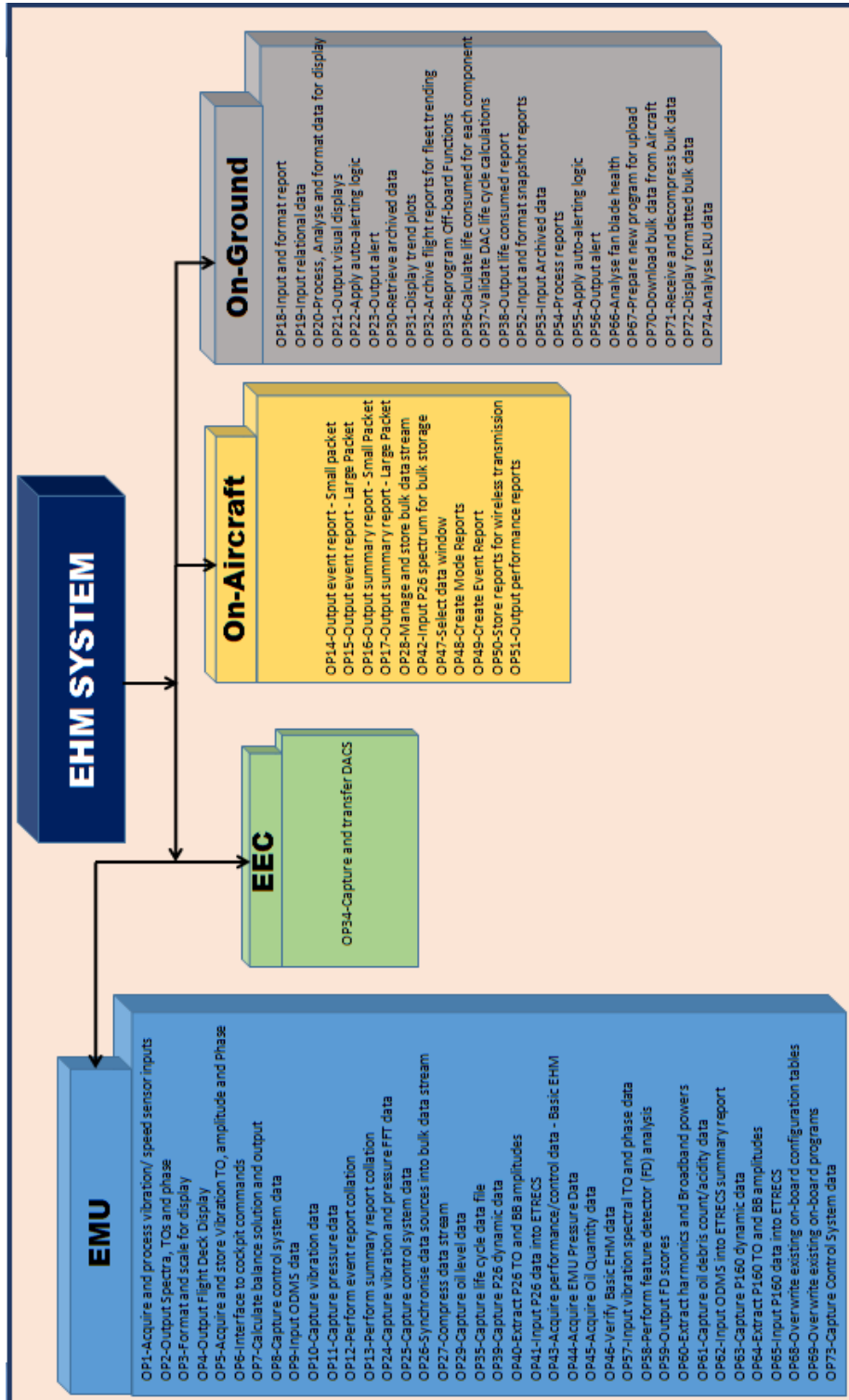


Figure 5.4: Robust solution 1 EHM system architecture with functional deployment.

5. Uncertainty Analysis and Robust Optimization

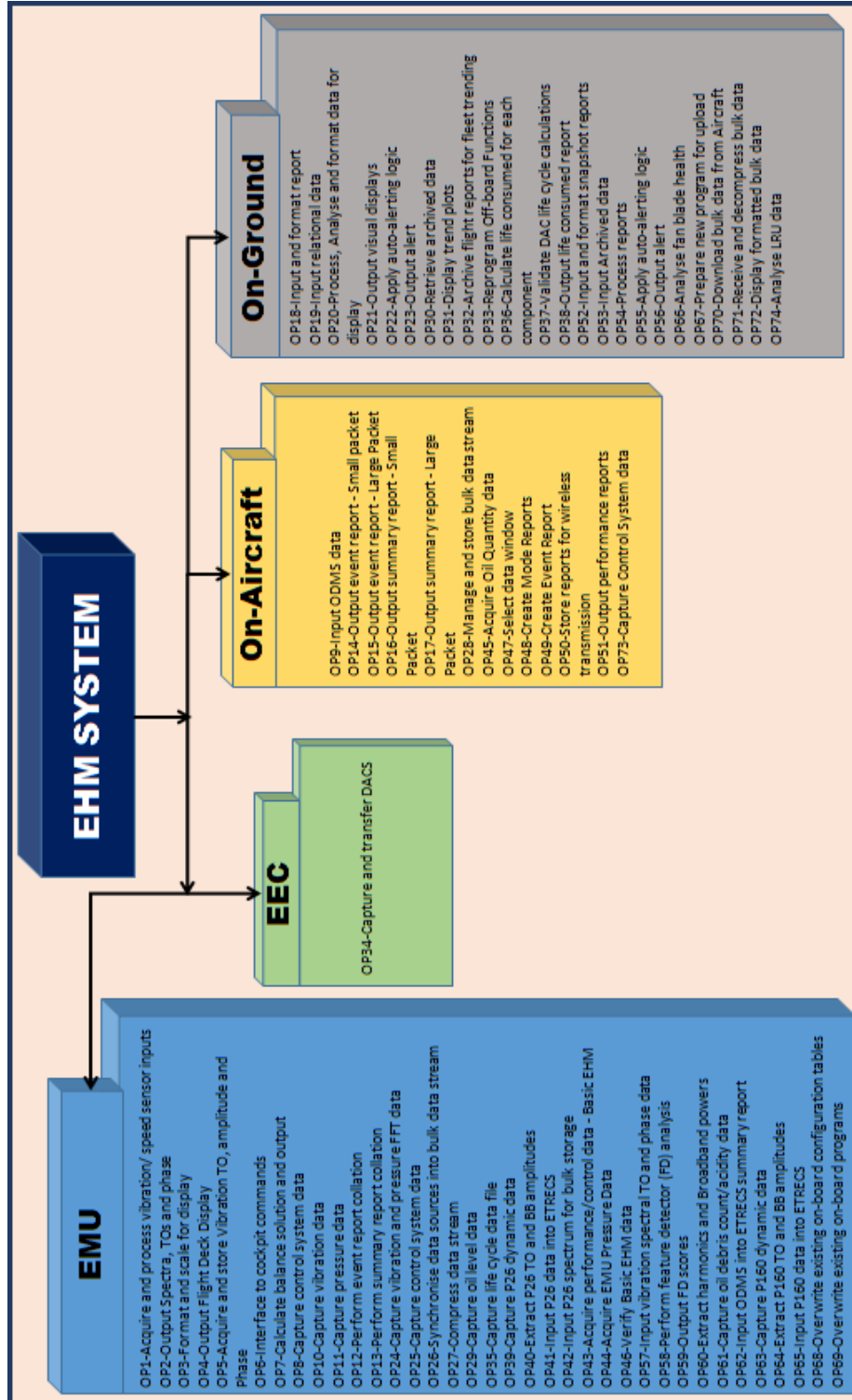


Figure 5.5: Robust solution 2 EHM system architecture with functional deployment.

5. Uncertainty Analysis and Robust Optimization

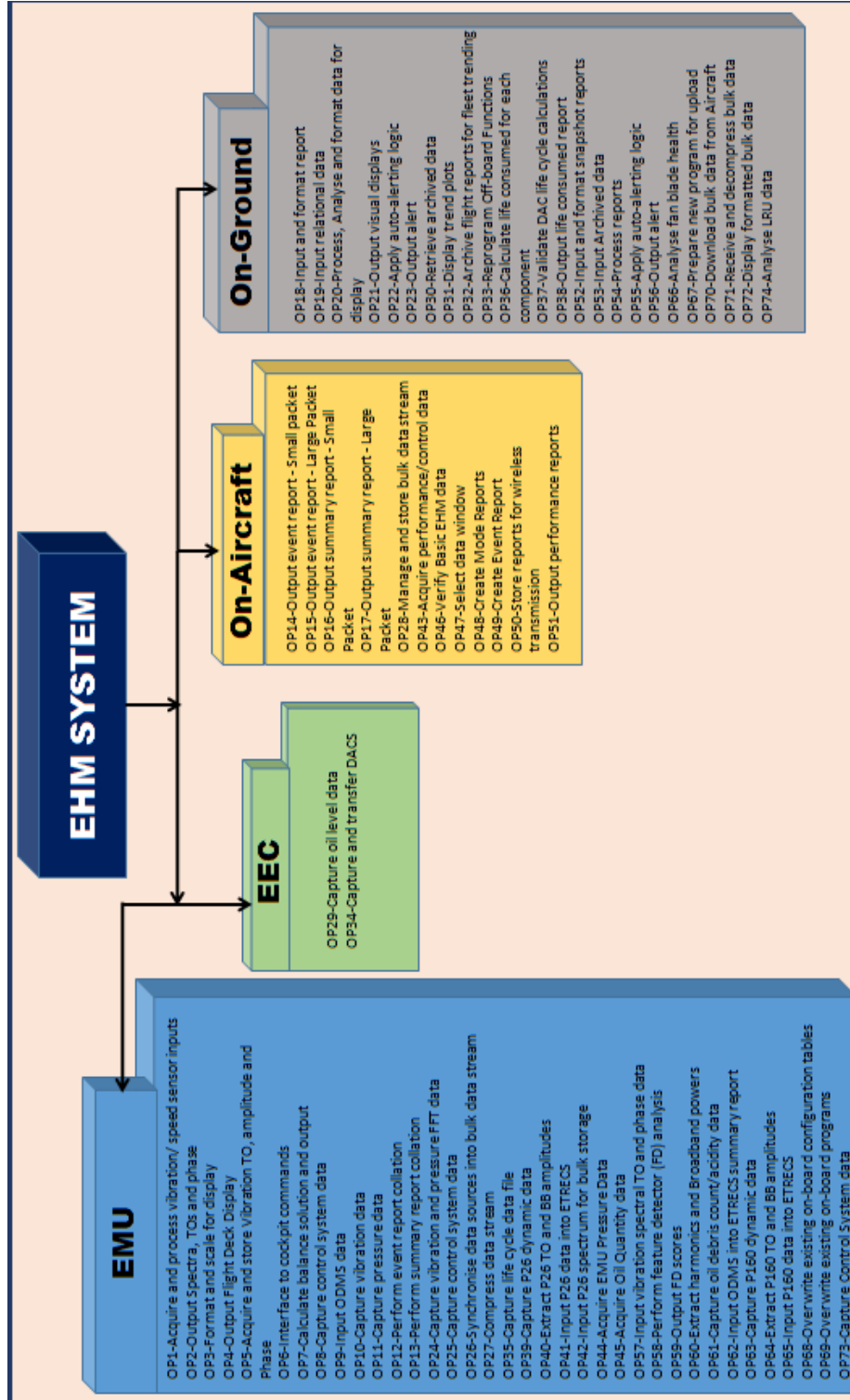


Figure 5.6: Baseline EHM system architecture with functional deployment.

5. Uncertainty Analysis and Robust Optimization

Constraints and Objective functions	Values
(1) Data flow rate requirement - EMU (Kbits/sec)	43.00
(2) Data flow rate requirement - EEC (Kbits/sec)	1.00
(3) Data flow rate requirement - Aircraft (Kbits/sec)	15.00
(4) Data flow rate requirement - on-Ground (Kbits/sec)	0.04
(5) Processing requirement - EMU (MFLOPS)	1566.00
(6) Processing requirement - EEC (MFLOPS)	6.00
(7) Processing requirement - Aircraft (MFLOPS)	265.00
(8) Processing requirement - on-Ground (MFLOPS)	683.00
(9) Criticality excess requirement	64.99
(10) Immediacy excess requirement	109.46
(11) Coupling excess requirement	64.88
(12) Security excess requirement	131.39
(13) IP-sensitivity excess requirement	9.49
(14) Flexibility excess requirement	236.41
(15) Robustness metric	5.87

Figure 5.7: Criteria scores for the baseline EHM system architecture.

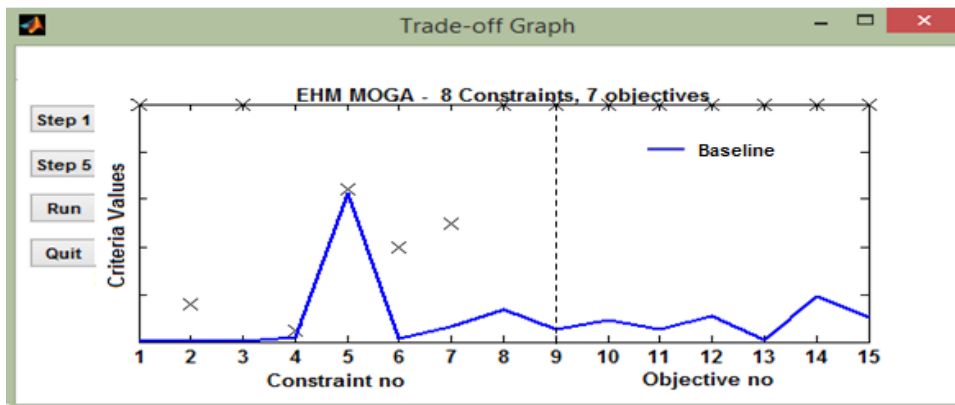


Figure 5.8: Baseline EHM system architecture criteria scores on MOGA parallel co-ordinates graph.

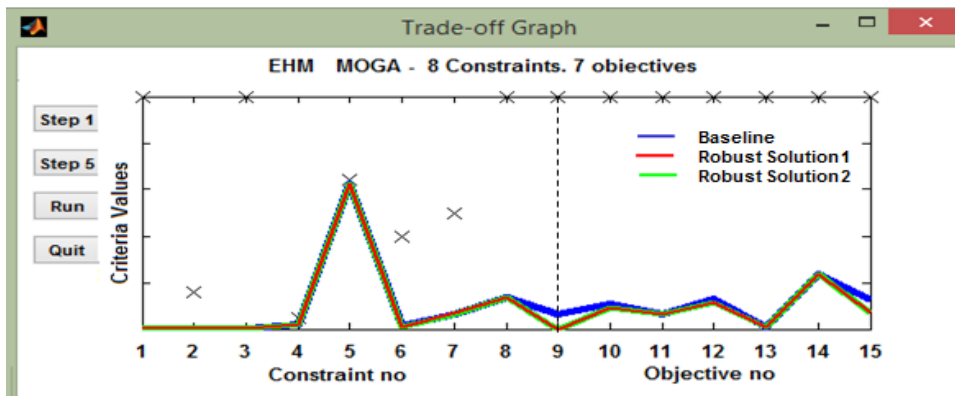


Figure 5.9: Comparison of Robust EHM system architecture solutions obtained from the optimization process with the baseline EHM system architecture.

5. Uncertainty Analysis and Robust Optimization

mediacy’, ‘Coupling’, ‘Security’, ‘IP sensitivity’, ‘Flexibility’ and the additional ‘Robustness metric’ objective, as per the procedure described in the Section 4.3.5 and Section 5.5.

The baseline EHM system architecture criteria scores are given in Figure 5.7 and they are plotted on the MOGA parallel co-ordinates graph as shown in Figure 5.8. It can be observed that, for the baseline EHM system architecture data flow and processing resource requirements are within the resource constraints limitations of the physical architecture subsystems. The criteria scores for the six objectives, from objective number 9 to 14 attribute excess requirement, indicate that with this deployment in the baseline EHM system architecture the six operational attribute requirements are not completely satisfied. The robustness metric (objective 15) value indicates that the baseline architecture is less sensitive to uncertain variations in the design parameters.

The robust Pareto optimal architecture solutions obtained from the proposed methodology are compared with the baseline EHM system architecture in Figure 5.9. The following observations are made from the comparison:

- The criteria scores for the baseline EHM system architecture are very close to the robust architecture solutions obtained from the optimization process. However, the Criticality operational attribute requirement (objective 9) is not completely satisfied in the baseline architecture, whereas, it is completely satisfied (criticality excess requirement is zero) in both robust Pareto optimal architecture solutions.
- The immediacy and security excess requirements (objectives 10 and 12) in the robust Pareto optimal architectures are slightly lower than the baseline architecture.
- The three remaining operational attributes, coupling, IP-sensitivity and flexibility (objective 11, 13 and 14) are equally satisfied in baseline and optimal EHM system architecture solutions.
- The robustness metric (objective 15) values for the Pareto optimal solutions

5. Uncertainty Analysis and Robust Optimization

obtained from the proposed methodology are better than the baseline EHM system architecture.

From the comparison of the baseline and the robust Pareto optimal EHM system architecture deployments shown in Figures 5.4, 5.5 and 5.6, it is observed that the operation for capturing oil level data (OP29) is deployed on the EMU in optimal solutions, where its criticality requirement is satisfied. Whereas, the OP29 is deployed on the EEC in the Baseline architecture and this is responsible for the criticality excess requirement value.

Compared with the baseline architecture, in robust solution 1, the operation 'OP42-input compressor pressure P26 spectrum for bulk storage' is deployed on the aircraft subsystem and the operations 'OP43-acquire performance and pressure data' and 'OP46 - verify basic EHM data' are deployed on the EMU. Whereas in robust solution 2, the operations 'OP9-Input ODMS data', 'OP45-acquire oil quantity data' and 'OP73-capture control system data for LRU anomaly analysis' are deployed on the aircraft subsystem and operations OP43 and OP46 are deployed on EMU. The resulting changes are a reduction in immediacy and security excess requirements for the robust Pareto optimal solutions. Deployment of the all the remaining EHM functional operations is identical for all three architectures.

From the above observations, out of the two robust Pareto optimal EHM system architecture solutions, the decision maker preferred to select the robust solution 1. This solution has fewer changes in EHM functional operations deployment, better criteria scores and least sensitivity to variations, compared with the current baseline EHM system architecture on the working gas turbine engine.

This demonstrates that the proposed multi-criteria optimization approach for system architecture design is successful in obtaining the best Pareto optimal architecture solutions for the EHM system architecture. In addition, the multi-criteria optimization approach provides a systematic means of evaluating system architecture alternatives, there by generating confidence that the entire search

space has been covered, yielding the most suitable solution(s), and taking account of uncertainties inherent in the evaluation process.

5.7 Summary

In this chapter, uncertainty analysis is presented for the EHM system architecture design and the polynomial chaos expansion method is used for estimating the sensitivities of the objectives. The obtained Pareto optimal solutions are sensitive with respect to the underlying assumptions. A robustness metric is introduced as an additional objective function for the EHM system architecture design. The robustness metric is computed using the variations of the criteria values computed using an efficient polynomial chaos expansion technique with the design of experiments with respect to the parameter uncertainties. A parallel processing facility, available in MATLAB, is utilized for evaluating the robustness metric for each candidate architecture solution over number of samples from the design of experiments. Preferences are expressed relating to the robustness metric objective and a set of robust non-dominated solutions are obtained using the proposed approach for the EHM system architecture design. The EHM system architecture design, using the proposed multi-criteria optimization methodology, is validated by comparing obtained architecture solutions with a baseline aero engine EHM system architecture. The validation process demonstrates the success of the proposed system architecture design approach using multi-criteria optimization and its great potential for application to other real world complex system design problems.

Chapter 6

Conclusions and Future Work

This concluding chapter summarizes the research work described in this thesis and presents the research contributions together with possible extensions to the work in future.

6.1 Research Objectives

The main objective of the current research work is to develop a methodology for system architecture design optimization using multi-criteria optimization techniques and to apply it to real world system architecture design applications.

System architecture is defined as the description of a complex system in terms of its functional requirements, physical elements and their interrelationships. Designing a complex system architecture can be a difficult task involving multifaceted trade-off decisions. The system architecture design is generally described by large multidisciplinary design phases. In order to find a best possible system architecture, the architecture design process can be formulated as an optimization problem. However, system architecture design problems often have many project-specific goals involving a mix of quantitative and qualitative criteria, which need to be optimized over a large decision space. The design process needs to consider data, models and experience from many design disciplines. For these multi-objective optimization problems, there is no one single optimal solution. Instead, there is a family of solutions called Pareto optimal solutions,

where each solution represents a compromise, or trade-off, between the competing objectives. In the last few decades, several tools and methods have been developed to support the system architecture design process. Prior work in this field has mainly focused on two or three objectives, not taking “many-objective” optimization issues into account. Many conventional techniques face difficulties in solving complex system architecture design problems having many objectives (i.e. in excess of 3). In the case of system architecture design problems, there may exist multiple decision vectors which map to identical objective vectors on the Pareto front; these are multi-modal Pareto optimal solutions. The main goal of a multi-objective evolutionary algorithm (MOEA) is to find a representative set of the Pareto-optimal solutions as close as possible to the true Pareto front (*convergence*), which are well distributed (*diversity*) within a decision maker’s region of interest (*pertinency*) on the Pareto front. However, with the increased number of objectives, the number of non-dominant solutions increases exponentially, decreasing the selection pressure and convergence towards the true Pareto optimal surface. Many multi-objective evolutionary algorithms fail to find and preserve all of the multi-modal solutions available in the decision space. Visualization of the Pareto optimal front also becomes difficult for many-objective optimization problems. The research work presented in this thesis successfully overcomes these difficulties by developing an interactive multi-criteria design optimization framework.

6.2 Research Contributions

The main contributions from this research work are briefly summarized and critiqued:

- **Development of an interactive multi-criteria design optimization framework.**

An interactive design optimization framework is proposed for solving many-objective system architecture design optimization problems. The framework uses an evolutionary multi-objective genetic algorithm (MOGA) with a progressive preference articulation technique. Using this technique the op-

timization search is focused onto a region of interest in the objective space, thus increasing selective pressure and convergence towards the Pareto optimal front. In many-objective optimization it is not feasible to cover all the regions of the objective space with a small size of population. The optimization algorithm is further enhanced with a crowding distance operator in order to maintain the diversity of the solutions in both objective and decision spaces. This will enable the optimization algorithm to find and preserve multi-modal Pareto optimal solutions present in the decision space for the given optimization problem. However, a large population size is required in order to achieve a good distribution of solutions in all of the multi-modal Pareto sets. A parallel coordinates graph in the MOGA software suite facilitates visualization of the interplay between the different objectives. A limitation of the parallel coordinates representation is that only adjacent objectives can be easily compared. However, in MOGA, there is a facility to interactively switch the order of representation of the objectives. The optimization framework supports the decision maker to express goals and preferences on the objectives in a progressive manner during the optimization, empowering a closely coupled user and optimization process interaction. The proposed method is tested on several benchmark test problems, with different case studies. In all test cases, the proposed optimization algorithm has successfully found multi-modal Pareto solutions with a good convergence towards the true Pareto front and well distributed in objective space and decision space. Finding more of the available multi-modal solutions would give the decision maker a greater selection when choosing between solutions. The proposed multi-criteria optimization framework is a valuable tool for system architecture design optimization with many conflicting objective functions, due to its ability to converge to the true Pareto front and find diversified Pareto optimal solutions in the decision maker's pertinent region of interest.

- **Development of a novel methodology for increasing the density of available Pareto solutions.**

A novel Pareto estimation (PE) method is introduced and an extension is

proposed for applying it to multi-objective multi-modal optimization problems. The main motivation for the Pareto estimation method is the ability to estimate additional Pareto optimal solutions in the decision maker's region of interest with reduced computational expense, without re-running of entire optimization process with a larger population size. The proposed PE method uses an efficient iVAT clustering technique, which does not require an input of 'number of clusters' to be specified for the clustering process, which is, in general, unknown for complex design data. In the iVAT clustering method, the pairwise dissimilarity data is re-ordered and displayed as a gray-scale image, thus highlighting the clusters present in the data as dark blocks along the diagonal of the image. The computational effort required by the iVAT clustering algorithm increases exponentially with the size of the data. In the proposed framework, the iVAT clustering method is utilized to identify and partition the different clusters available in the decision variable space which correspond to the multi-modal Pareto optimal solutions. These individual clusters are then used to train the radial basis function based neural network (RBFNN) models to find the relationship between Pareto objective solutions and the decision space. These are then employed to estimate additional Pareto solutions for the optimization problem. The PE method has been employed on several benchmark test problems, with different case studies in order to estimate a large number of Pareto optimal solutions, thereby increasing the density of available non-dominated solutions in the multi-objective problems. In all cases, the Pareto estimation method has successfully found many non-dominated solutions corresponding to different multi-modal solutions. The success of the proposed method using RBFNN models is dependent on the number of solutions available in an individual cluster for training and estimating the one-to-one mapping between objective space and decision vector space. For the test cases, around 10 – 20% of the estimated solutions were found to be dominated, due to the approximation errors in training and estimation of solution using RBFNN models.

- **Application of multi-criteria design optimization approach to a real-world system architecture design problem.**

System architecture design using the proposed multi-criteria optimization framework is demonstrated with a real-world application of an aero gas turbine engine health management (EHM) system. An EHM system can help to reduce costs due to unanticipated disruptions to service. All the functional requirements of the EHM system are captured and they are further decomposed into 74 EHM functional operations. The EHM system architecture design problem is formulated as a multi-criteria optimization problem having many objectives and constraints of resource limitations. EHM system architecture models are developed in SysML and successfully integrated into the optimization platform. Faithful capturing of expert elicitation while developing the system architecture models can improve the accuracy of the performance evaluations of alternatives. A strategy for optimal deployment of the functional operations on physical architecture locations has been successfully developed. The EHM system architecture design problem is successfully optimized using the proposed optimization methodology. The optimizer supports the decision maker by providing a facility for progressive preference articulation. The optimization algorithm successfully generated a well diversified family of Pareto optimal solutions through expression of decision maker preferences progressively. Cluster analysis using the iVAT clustering technique revealed four groups of Pareto optimal solutions; this provides valuable insight into EHM system architecture design trade-offs. Using the Pareto estimation method, several additional Pareto optimal solutions are successfully generated with reduced computational expense. However, around 35% of the estimated solutions using PE method were found to be dominated or infeasible, due to the very low data-flow rate constraint on the ground system. Through this design process, it is revealed that it is not possible to fully satisfy all attributes for the EHM system, while observing the given constraints, thereby highlighting the value of the multi-criteria approach. Moreover, using this approach it was possible to identify the most significant design constraints (“hot spots”) and the opportunities afforded by either the relaxation or the tightening of these constraints

along with their performance implications. Performance of the optimization framework is assessed by evaluating test metrics and comparing with popular multi-objective optimization algorithms using established statistical methods. The obtained results demonstrated the superior performance of the proposed methodology with better convergence and diversity of the solutions with good repeatability.

- **Uncertainty analysis for finding robust system architecture design solutions.**

Uncertainty analysis is presented for the EHM system architecture design using the polynomial chaos expansion method for estimating the sensitivities of the objectives. In the EHM system architecture design optimization, in order to integrate the EHM architecture models into the optimization platform and to evaluate the various objective functions, the qualitative functional attribute requirements are transformed to suitable numeric values. The obtained Pareto optimal solutions are sensitive with respect to the chosen numerical parameter values. In order to find architecture solutions with low sensitivity to the parameter uncertainties, a robustness metric is introduced as an additional objective function for the EHM system architecture design. The robustness metric is computed using the variations of the criteria values evaluated with respect to variations in the parameter values. An efficient uncertainty quantification technique using polynomial chaos expansion is integrated within the optimization process. The approach generates a design of experiments with respect to the parameter uncertainties. In the PCE method, the number of samples required for estimating the polynomial coefficients increases proportional to the factorial of the number of uncertain parameters, thereby increasing computational effort. A parallel processing facility available in MATLAB is utilized for evaluating the robustness metric for each candidate architecture solution over the number of samples from the design of experiments with reduced computational time. In the optimization framework, using progressive preference articulation, the decision maker can specify the level of sensitivity that can be allowed in the design. Finally, a set of robust non-dominated

solutions are obtained using the proposed approach for the EHM system architecture design; these are further validated by comparing with a baseline aero engine EHM system architecture.

The proposed methodology has great potential to be implemented on real-world complex system architecture design applications. It has already been applied to a novel oil system architecture design and a novel cooled cooling air system architecture design of aero gas turbine engine. The methodology is being adopted by the co-funder of the research, Rolls-Royce plc, for general application to their system architecture design process.

6.3 Future Work

Although, the system architecture design optimization process has been successfully demonstrated with the proposed interactive multi-criteria optimization framework, a number of extensions to the research work are possible as presented below.

- During the research work presented in this thesis several tools, techniques and approaches have been developed for multi-objective optimization, integration of architecture models, cluster analysis and Pareto estimation methods for the system architecture design. In future these should be integrated into a generic optimization framework for multi-disciplinary design optimization, where different models created using different software and different tools and techniques for simulation and analysis, can be easily integrated and information can be easily exchanged among all the techniques and models. This will enable design practitioners to engage their complex design problems effortlessly, without the need of code modifications, and apply the required tool-set to achieve design goals.
- In the Pareto estimation method, the quality of the estimated Pareto solutions highly depend on the radial basis function neural network models and the solutions used for training. In future, the Pareto estimation method can

be extended to use other popular meta-modeling approaches such as Kriging and Bayesian networks in order to estimate a better mapping between the Pareto objective vectors and decision vectors. This will improve the efficiency and accuracy approach in generating additional Pareto optimal solutions.

- For large data sets the VAT clustering algorithms becomes computationally expensive. In future, different clustering techniques can be integrated into the optimization framework and a suitable clustering method can be selected for analysis depending on the nature of the data.
- In the optimization framework, several other visualization methods can be incorporated such as hyper radial visualization for viewing the Pareto optimal solutions in many-objective optimization, to facilitate cognitive understanding of the optimization results.

References

- Adra, S.F., Griffin, I.A. & Fleming, P.J. (2007). A comparative study of progressive preference articulation techniques for multiobjective optimisation. In *Evolutionary Multi-Criterion Optimization*, 908–921, Springer.
- Andrews, J.D. & Moss, T.R. (2002). *Reliability and risk assessment*. Wiley-Blackwell.
- Armstrong, M., de Tenorio, C., Garcia, E. & Mavris, D. (2008). Function based architecture design space definition and exploration. *26th International Congress of Aeronautical Sciences*, 14–19.
- Athans, M. & Falb, P. (1966). *Optimal control*. McGraw-Hill, New York.
- Bader, J. & Zitzler, E. (2011). Hype: An algorithm for fast hypervolume-based many-objective optimization. *Evolutionary Computation*, 19, 45–76.
- Baxter, B. (1992). *The interpolation theory of radial basis functions*. Ph.D. thesis, University of Cambridge.
- Ben-Tal, A. & Nemirovski, A. (2002). Robust optimization—methodology and applications. *Mathematical Programming*, 92, 453–480.
- Bentley, P.J. & Wakefield, J.P. (1998). Finding acceptable solutions in the Pareto-optimal range using multiobjective genetic algorithms. In *Soft Computing in Engineering Design and Manufacturing*, 231–240, Springer.
- Beyer, H.G. & Sendhoff, B. (2007). Robust optimization - A comprehensive survey. *Computer Methods in Applied Mechanics and Engineering*, 196, 3190–3218.

REFERENCES

- Bezdek, J.C. & Hathaway, R.J. (2002). VAT: a tool for visual assessment of (cluster) tendency. In *Proceedings of the 2002 International Joint Conference on Neural Networks IJCNN'02.*, vol. 3, 2225–2230, IEEE.
- Bianchi, L., Dorigo, M., Gambardella, L.M. & Gutjahr, W.J. (2009). A survey on metaheuristics for stochastic combinatorial optimization. *Natural Computing: An International Journal*, 8, 239–287.
- Bishop, C.M. (1995). *Neural Networks for Pattern Recognition*. Oxford University Press.
- Blanquart, J.P., Astruc, J.M., Baufreton, P., Boulanger, J.L., Delseny, H., Gassino, J., Ladier, G., Lediot, E., Leeman, M., Machrouh, J., Philippe, Q. & Bertrand, R. (2012). Criticality categories across safety standards in different domains. *ERTS-2012, Toulouse*, 1–3.
- Blickle, T., Teich, J. & Thiele, L. (1998). System-level synthesis using evolutionary algorithms. *Design Automation for Embedded Systems*, 3, 23–58.
- Bors, A.G. & Pitas, I. (1996). Median radial basis function neural network. *IEEE Transactions on Neural Networks*, 7, 1351–1364.
- Bourne, D., Dixon, R. & Horne, A. (2011). Architectural design of distributed control systems for aero gas turbine engines using genetic algorithms. In *21st International Conference on Systems Engineering (ICSEng)*, 3–7, IEEE.
- Branke, J. & Deb, K. (2004). Integrating user preferences into evolutionary multi-objective optimization. *Studies in Fuzziness and Soft Computing*, 167, 461–478.
- Branke, J., Kauler, T. & Schmeck, H. (2001). Guidance in evolutionary multi-objective optimization. *Advances in Engineering Software*, 32, 499 – 507.
- Chan, K.P. & Ray, T. (2005). An evolutionary algorithm to maintain diversity in the parametric and the objective space. In *International Conference on Computational Robotics and Autonomous Systems (CIRAS)*, Centre for Intelligent Control, National University of Singapore.

REFERENCES

- Cheung, J., Scanlan, J., Wong, J., Forrester, J., Eres, H., Collopy, P., Hollingsworth, P., Wiseall, S. & Briceno, S. (2012). Application of value-driven design to commercial aeroengine systems. *Journal of Aircraft*, 49, 688–702.
- Cleveland, W.S. (1993). *Visualizing data*. Hobart Press.
- Coello-Coello, C.A., Lamont, G.B. & Van Veldhuisen, D.A. (2007). *Evolutionary algorithms for solving multi-objective problems*. Springer.
- Collopy, P.D. & Hollingsworth, P.M. (2011). Value-driven design. *Journal of Aircraft*, 48, 749–759.
- Corne, D.W. & Knowles, J.D. (2007). Techniques for highly multi-objective optimisation: some non-dominated points are better than others. In *Proceedings of the 9th annual conference on genetic and evolutionary computation*, 773–780, ACM.
- Crawley, E.F., de Weck, O., Eppinger, S.D., Magee, C., Moses, J., Seering, W., Schindall, J., Wallace, D. & Whitney, D.E. (2004). The influence of architecture in engineering systems. *Engineering Systems Monograph*.
- Curry, H.B. (1944). The Method of Steepest Descent for Nonlinear Minimization Problems. *Quart. Appl. Math*, 2, 250–261.
- Cvetković, D. & Coello-Coello, C.A. (2005). Human preferences and their applications in evolutionary multi-objective optimization. In *Knowledge Incorporation in Evolutionary Computation*, 479–502, Springer.
- Darwin, C. (1858). *The origin of species*. 811, Hayes Barton Press.
- Deb, K. (2001). *Multi-objective optimization using evolutionary algorithms*, vol. 16. John Wiley & Sons Hoboken, NJ.
- Deb, K. & Gupta, H. (2006). Introducing robustness in multi-objective optimization. *Evolutionary Computation*, 14, 463–494.
- Deb, K., Pratap, A., Agarwal, S. & Meyarivan, T. (2002a). A Fast and Elitist Multiobjective Genetic Algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*, 6, 182–197.

REFERENCES

- Deb, K., Thiele, L., Laumanns, M. & Zitzler, E. (2002b). Scalable Multi-Objective Optimization Test Problems. In *Congress on Evolutionary Computation*, vol. 1, 825–830.
- Deb, K. & Tiwari, S. (2008). Omni-optimizer: A generic evolutionary algorithm for single and multi-objective optimization. *European Journal of Operational Research*, 185, 1062–1087.
- Dick, R.P. & Jha, N.K. (1997). MOGAC: a multiobjective genetic algorithm for the co-synthesis of hardware-software embedded systems. In *Proceedings of the 1997 IEEE/ACM international conference on Computer-aided design*, 522–529, IEEE Computer Society.
- Dorigo, M. (1992). *Optimization, learning and natural algorithms*. Ph.D. thesis, Politecnico di Milano, Italy.
- Dorigo, M., Maniezzo, V. & Colorni, A. (1991). The ant system: An autocatalytic optimizing process. *TR91-016, Politecnico di Milano*.
- Drechsler, N., Drechsler, R. & Becker, B. (2001). Multi-objective optimisation based on relation favour. In *Evolutionary multi-criterion optimization*, 154–166, Springer.
- Edgeworth, F. (1881). *Mathematical Psychics: An Essay on the Application of Mathematics to the Moral Sciences*. 10, CK Paul.
- Erfani, T. & Utyuzhnikov, S. (2012). Control of robust design in multiobjective optimization under uncertainties. *Structural and Multidisciplinary Optimization*, 45, 247–256.
- Farina, M. & Amato, P. (2002). On the optimal solution definition for many-criteria optimization problems. In *Proceedings of the NAFIPS-FLINT international conference, Fuzzy Information Processing Society*, 233–238, IEEE.
- Fisher, M.L. (2004). The lagrangian relaxation method for solving integer programming problems. *Management science*, 50, 1861–1871.

REFERENCES

- Fisher, R.A. (1930). *The Genetical Theory of Natural Selection: A Complete Variorum Edition*. Oxford University Press.
- Fleming, P.J., Purshouse, R.C. & Lygoe, R.J. (2005). Many-objective optimization: An engineering design perspective. In *Evolutionary Multi-Criterion Optimization*, 14–32, Springer.
- Fletcher, C.A. (1984). *Computational galerkin methods*. Springer.
- Fletcher, R. (1980). *Practical Methods of Optimization*. John Wiley & Sons, Inc., Chichester.
- Fonseca, C.M. & Fleming, P.J. (1993). Genetic Algorithms for Multiobjective Optimization: Formulation, Discussion and Generalization. In *Proceedings of the fifth international conference on genetic algorithms*, vol. 1, 416–423, San Mateo, California.
- Fonseca, C.M. & Fleming, P.J. (1998a). Multiobjective Optimization and Multiple Constraint Handling with Evolutionary Algorithms. I. A Unified Formulation. *IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans*, 28, 26–37.
- Fonseca, C.M. & Fleming, P.J. (1998b). Multiobjective Optimization and Multiple Constraint Handling with Evolutionary Algorithms. II. Application Example. *IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans*, 28, 38–47.
- Fonseca, C.M., Paquete, L. & Lopez-Ibanez, M. (2006). An Improved Dimension-Sweep Algorithm for the Hypervolume Indicator. In *IEEE Congress on Evolutionary Computation*, 1157 – 1163.
- Friedenthal, S., Moore, A. & Steiner, R. (2006). OMG Systems Modeling Language (OMG SysML) Tutorial. *INCOSE International Symposium*, 19, 1840–1972.
- Gembicki, F. (1974). *Vector optimization for control with performance and parameter sensitivity indices*. Ph.D. thesis, Case Western Reserve University, Cleveland, Ohio.

REFERENCES

- Giagkiozis, I. & Fleming, P.J. (2012). Increasing the Density of Available Pareto Optimal Solutions. Research Report No. 1028, Department of Automatic Control and Systems Engineering, The University of Sheffield.
- Giagkiozis, I., Purshouse, R.C. & Fleming, P.J. (2013). An overview of population-based algorithms for multi-objective optimisation. *International Journal of Systems Science*, 1–28.
- Goldberg, D.E. (1989). *Genetic Algorithms in Search, Optimization, and Machine Learning*. Addison-Wesley Professional.
- Goldberg, D.E. & Richardson, J. (1987). Genetic algorithms with sharing for multimodal function optimization. In *Genetic algorithms and their applications: Proceedings of the Second International Conference on Genetic Algorithms*, 41–49, Hillsdale, NJ: Lawrence Erlbaum.
- Greenwood, G.W., Hu, X. & D’Ambrosio, J.G. (1996). Fitness functions for multiple objective optimization problems: Combining preferences with Pareto rankings. In *FOGA*, vol. 96, 437–455.
- Gries, M. (2004). Methods for evaluating and covering the design space during early design development. *Integration, the VLSI Journal*, 38, 131–183.
- Harik, G.R. (1995). Finding multimodal solutions using restricted tournament selection. In *ICGA*, 24–31.
- Hause, M. (2006). The sysml modelling language. In *Fifteenth European Systems Engineering Conference*, vol. 9, Citeseer.
- Heady, E.O. & Candler, W. (1963). *Linear programming methods*. Iowa State University Press.
- Helton, J.C. & Davis, F.J. (2000). *Sampling-based methods for uncertainty and sensitivity analysis*. Sandia National Laboratories Albuquerque.
- Henley, E.J. & Kumamoto, H. (1981). *Reliability engineering and risk assessment*, vol. 193. Prentice-Hall Englewood Cliffs (NJ).

REFERENCES

- Holland, J.H. (1975). *Adaptation in natural and artificial systems*. The University of Michigan Press, Ann Arbor.
- Hughes, E.J. (2003). Multiple single objective Pareto sampling. In *The 2003 Congress on Evolutionary Computation*, vol. 4, 2678–2684, IEEE.
- Hwang, C.L. & Masud, A.S.M. (1979). *Multiple Objective Decision Making - Methods and Applications*, vol. 164 of *Lecture Notes in Economics and Mathematical Systems*. Springer-Verlag Berlin Heidelberg.
- Ignizio, J.P. (1976). *Goal programming and extensions*. Lexington Books.
- Ikedo, K., Kita, H. & Kobayashi, S. (2001). Failure of Pareto-based MOEAs: does non-dominated really mean near to optimal? In *Proceedings of the 2001 Congress on Evolutionary Computation*, vol. 2, 957–962, IEEE.
- Inselberg, A. (1985). The plane with parallel coordinates. *The Visual Computer*, 1, 69–91.
- Ishibuchi, H., Tsukamoto, N. & Nojima, Y. (2007). Iterative approach to indicator-based multiobjective optimization. In *IEEE Congress on Evolutionary Computation CEC 2007*, 3967–3974, IEEE.
- Ishibuchi, H., Tsukamoto, N. & Nojima, Y. (2008). Evolutionary Many-Objective Optimization: A Short Review. In *IEEE Congress on Evolutionary Computation*, 2419–2426, IEEE.
- Jain, A.K., Murty, M.N. & Flynn, P.J. (1999). Data clustering: a review. *ACM computing surveys (CSUR)*, 31, 264–323.
- Jones, D.R., Schonlau, M. & Welch, W.J. (1998). Efficient Global Optimization of Expensive Black-Box Functions. *Journal of Global Optimization*, 13, 455–492.
- Kalos, M.H. & Whitlock, P.A. (2008). *Monte carlo methods*. John Wiley & Sons.
- Kelley, J.E., Jr (1960). The cutting-plane method for solving convex programs. *Journal of the Society for Industrial & Applied Mathematics*, 8, 703–712.

REFERENCES

- Kennedy, J. & Eberhart, R. (1995). Particle swarm optimization. In *Proceedings of IEEE International Conference on Neural Networks*, vol. 4, 1942–1948, IEEE.
- Khare, V., Yao, X. & Deb, K. (2003). Performance scaling of multi-objective evolutionary algorithms. In *Evolutionary Multi-Criterion Optimization*, 376–390, Springer.
- Kroese, D.P., Taimre, T. & Botev, Z.I. (2011). *Handbook of Monte Carlo Methods*, vol. 706. Wiley.
- Kudikala, R., Giagkiozis, I. & Fleming, P.J. (2013a). Estimation of Pareto Optimal Solutions for Multi-Objective Multi-Modal Problems. In *19th International Conference on Soft Computing - MENDEL2013*, vol. 1, Brno University of Technology, Czech Republic.
- Kudikala, R., Giagkiozis, I. & Fleming, P.J. (2013b). Increasing the Density of Multi-Objective Multi-Modal Solutions using Clustering and Pareto Estimation Techniques. In *Proceedings of The 2013 International Conference on Genetic and Evolutionary Methods (GEM'13)*, vol. 1, 8–14, WORLDCOMP 2013, World Academy of Science, LasVegas, Nevada, USA.
- Kukkonen, S. & Lampinen, J. (2007). Ranking-dominance and many-objective optimization. In *IEEE Congress on Evolutionary Computation CEC 2007*, 3983–3990, IEEE.
- Künzli, S. (2006). *Efficient design space exploration for embedded systems*. Ph.D. thesis, Swiss Federal Institute of Technology Zurich, Switzerland.
- Kursawe, F. (1991). A Variant of Evolution Strategies for Vector Optimization. *Parallel Problem Solving from Nature*, 193–197.
- Li, H. & Zhang, Q. (2009). Multiobjective Optimization Problems with Complicated Pareto Sets, MOEA/D and NSGA-II. *IEEE Transactions on Evolutionary Computation*, 13, 284–302.
- MacQueen, J. *et al.* (1967). Some methods for classification and analysis of multivariate observations. In *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, vol. 1, 281–297, California, USA.

REFERENCES

- Mahfoud, S.W. (1992). Crowding and preselection revisited. *Urbana*, 51, 61801.
- Martens, A., Koziolok, H., Becker, S. & Reussner, R. (2010). Automatically improve software architecture models for performance, reliability, and cost using evolutionary algorithms. In *Proceedings of the first joint WOSP/SIPEW international conference on Performance engineering*, 105–116, ACM.
- MathWorks (2010). *MATLAB version 7.10.0 (R2010a)*. The MathWorks Inc., Natick, Massachusetts.
- McKay, M., Beckman, R. & Conover, W. (2000). A comparison of three methods for selecting values of input variables in the analysis of output from a computer code. *Technometrics*, 42, 55–61.
- Miettinen, K. (1999). *Nonlinear multiobjective optimization*, vol. 12. Springer.
- Mizuno, S., Akao, Y. & Ishihara, K. (1994). *QFD, the customer-driven approach to quality planning and deployment*. Asian Productivity Organization.
- Molinari, M. (2004). *XML Toolbox for MATLAB version 3.2.1*. Geodise Lab, The Geodise Project, University of Southampton, Highfield, Southampton.
- Murty, K.G. (1983). *Linear programming*, vol. 57. Wiley New York.
- Najm, H.N. (2009). Uncertainty quantification and polynomial chaos techniques in computational fluid dynamics. *Annual Review of Fluid Mechanics*, 41, 35–52.
- Nemhauser, G.L. & Wolsey, L.A. (1988). *Integer and combinatorial optimization*, vol. 18. Wiley New York.
- Pareto, V. (1896). *Cours D'Économie Politique*. Droz, Geneva.
- Perez, R.A. (2008). *Uncertainty analysis of computational fluid dynamics via polynomial chaos*. Ph.D. thesis, Virginia Polytechnic Institute and State University.
- Poles, S. & Lovison, A. (2009). A polynomial chaos approach to robust multi-objective optimization. In *Dagstuhl Seminar Proceedings of Hybrid and Robust*

REFERENCES

- Approaches to Multiobjective Optimization*, 09041, Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, Dagstuhl, Germany.
- Preuss, M., Naujoks, B. & Rudolph, G. (2006). *Pareto set and EMOA behavior for simple multimodal multiobjective functions*, vol. 4193 of *Lecture Notes in Computer Science*. Springer.
- Purshouse, R.C. (2003). *On the Evolutionary Optimisation of Many Objectives*. Ph.D. thesis, The University of Sheffield.
- Purshouse, R.C. & Fleming, P.J. (2003). Evolutionary Many-Objective Optimisation: An Exploratory Analysis. In *IEEE Congress on Evolutionary Computation*, vol. 3, 2066–2073, IEEE.
- Purshouse, R.C. & Fleming, P.J. (2007). On the Evolutionary Optimization of Many Conflicting Objectives. *IEEE Transactions on Evolutionary Computation*, 11, 770–784.
- Purshouse, R.C., Jalbă, C. & Fleming, P.J. (2011). Preference-Driven Co-Evolutionary Algorithms Show Promise for Many-Objective Optimisation. In *Evolutionary Multi-Criterion Optimization*, 136–150, Springer.
- Rachmawati, L. & Srinivasan, D. (2006). Preference incorporation in multi-objective evolutionary algorithms: A survey. In *IEEE Congress on Evolutionary Computation CEC 2006*, 962–968, IEEE.
- Rao, S.S. (2009). *Engineering Optimization: Theory and Practice*. John Wiley & Sons.
- Ravindran, A., Reklaitis, G.V. & Ragsdell, K.M. (2006). *Engineering Optimization: Methods and Applications*. John Wiley & Sons.
- Rechenberg, I. (1973). *Evolutionstrategie: Optimierung technischer systeme nach prinzipien der biologischen evolution*. Frommann-Holzboog, Stuttgart.
- Reussner, R., Becker, S., Happe, J., Koziol, H., Krogmann, K. & Kuperberg, M. (2007). The palladio component model. *Interner Bericht*, 21.

REFERENCES

- Robert, C.P. & Casella, G. (1999). *Monte Carlo statistical methods*. Springer.
- Rockafellar, R.T. (1987). Linear-quadratic programming and optimal control. *SIAM Journal on Control and Optimization*, 25, 781–814.
- Rubinstein, R.Y. & Kroese, D.P. (2011). *Simulation and the Monte Carlo method*, vol. 707. John Wiley & Sons.
- Rudolph, G., Naujoks, B. & Preuss, M. (2007). Capabilities of EMOA to detect and preserve equivalent Pareto subsets. In *Evolutionary Multi-Criterion Optimization*, 36–50, Springer.
- Sato, H., Aguirre, H.E. & Tanaka, K. (2007). Controlling dominance area of solutions and its impact on the performance of MOEAs. In *Evolutionary Multi-Criterion Optimization*, 5–20, Springer.
- Schaffer, J.D. (1985). Multiple Objective Optimization with Vector Evaluated Genetic Algorithms. In *Conference on Genetic Algorithms*, 93–100, L. Erlbaum Associates Inc.
- Selva, D. & Crawley, E.F. (2010). Integrated assessment of packaging architectures in earth observing programs. In *IEEE Aerospace Conference*, 1–17, IEEE.
- Shir, O.M., Preuss, M., Naujoks, B. & Emmerich, M. (2009). Enhancing decision space diversity in evolutionary multiobjective algorithms. In *Evolutionary Multi-Criterion Optimization*, 95–109, Springer.
- Snyman, J. (2005). *Practical Mathematical Optimization: An Introduction to Basic Optimization Theory and Classical and New Gradient-Based Algorithms*, vol. 97. Springer Science & Business Media.
- Sparx Systems (2007). *Enterprise Architect software, version 7.0.818*. Sparx Systems Pty Ltd., Victoria, Australia.
- Srinivas, N. & Deb, K. (1994). Multiobjective Optimization Using Nondominated Sorting in Genetic Algorithms. *Evolutionary Computation*, 2, 221–248.

REFERENCES

- Sudret, B. (2008). Global sensitivity analysis using polynomial chaos expansions. *Reliability Engineering & System Safety*, 93, 964–979.
- Sülflow, A., Drechsler, N. & Drechsler, R. (2007). Robust multi-objective optimization in high dimensional spaces. In *Evolutionary multi-criterion optimization*, 715–726, Springer.
- Tanner, G.F. (2010). An aero engine health management system sysml model. Technical Report DNS157549, Rolls-Royce plc., Derby, UK.
- Tanner, G.F. & Crawford, J.A. (2003). An integrated engine health monitoring system for gas turbine aero-engines. *IEE Seminar on Aircraft Airborne Condition Monitoring*, 2003, 5–5.
- Tarafder, A., Rangaiah, G. & Ray, A.K. (2007). A study of finding many desirable solutions in multiobjective optimization of chemical processes. *Computers & chemical engineering*, 31, 1257–1271.
- Thiele, L., Chakraborty, S., Gries, M. & Künzli, S. (2002). Design space exploration of network processor architectures. *Network Processor Design: Issues and Practices*, 1, 55–89.
- Thompson, H.A., Chipperfield, A.J., Fleming, P.J. & Legge, C. (1999). Distributed aero-engine control systems architecture selection using multi-objective optimisation. *Control Engineering Practice*, 7, 655–664.
- Toffolo, A. & Benini, E. (2003). Genetic diversity as an objective in multi-objective evolutionary algorithms. *Evolutionary Computation*, 11, 151–167.
- Van Veldhuizen, D.A. (1999). *Multiobjective Evolutionary Algorithms: Classifications, Analyses, and New Innovations*. Ph.D. thesis, Air Force Institute of Technology.
- Villadsen, J. & Michelsen, M.L. (1978). *Solution of differential equation models by polynomial approximation*, vol. 7. Prentice-Hall Englewood Cliffs, NJ.

REFERENCES

- Wagner, T., Beume, N. & Naujoks, B. (2007). Pareto-, aggregation-, and indicator-based methods in many-objective optimization. In *Evolutionary multi-criterion optimization*, 742–756, Springer.
- Wang, L., Nguyen, U., Bezdek, J., Leckie, C. & Ramamohanarao, K. (2010). iVAT and aVAT: enhanced visual analysis for cluster tendency assessment. *Advances in Knowledge Discovery and Data Mining*, 16–27.
- Wang, R., Purshouse, R.C. & Fleming, P.J. (2012). Preference-inspired co-evolutionary algorithms for many-objective optimisation. *IEEE Transactions on Evolutionary Computation*, 17, 474–494.
- Weilkiens, T. (2011). *Systems engineering with SysML/UML: modeling, analysis, design*. Morgan Kaufmann.
- Wiener, N. (1938). The homogeneous chaos. *American Journal of Mathematics*, 60, 897–936.
- Yin, X. & Gerny, N. (1993). A fast genetic algorithm with sharing scheme using cluster analysis methods in multimodal function optimization. In *Artificial neural nets and genetic algorithms*, 450–457, Springer.
- Zhang, Q. & Li, H. (2007). MOEA/D: A Multiobjective Evolutionary Algorithm Based on Decomposition. *IEEE Transactions on Evolutionary Computation*, 11, 712–731.
- Zhang, Q., Liu, W. & Li, H. (2009). The Performance of a New Version of MOEA/D on CEC09 Unconstrained MOP Test Instances. In *IEEE Congress on Evolutionary Computation*, 203–208, IEEE.
- Zitzler, E. & Künzli, S. (2004). Indicator-Based Selection in Multiobjective Search. In *Parallel Problem Solving from Nature*, 832–842, Springer.
- Zitzler, E., Laumanns, M. & Thiele, L. (2001). SPEA2: Improving the Strength Pareto Evolutionary Algorithm. Tech. Rep. 103, Computer Engineering and Networks Laboratory (TIK), ETH Zurich, Zurich, Switzerland.

REFERENCES

- Zitzler, E. & Thiele, L. (1998). An evolutionary algorithm for multiobjective optimization: The strength Pareto approach. *Swiss Federal Institute of Technology, TIK-Report*, 43.
- Zitzler, E. & Thiele, L. (1999). Multiobjective Evolutionary Algorithms: A Comparative Case Study and the Strength Pareto Approach. *IEEE Transactions on Evolutionary Computation*, 3, 257–271.
- Zou, X., Chen, Y., Liu, M. & Kang, L. (2008). A new evolutionary algorithm for solving many-objective optimization problems. *IEEE Transactions on Systems, Man, and Cybernetics*, 38, 1402–1412.
- Zou, X., Liu, M., Kang, L. & He, J. (2004). A high performance multi-objective evolutionary algorithm based on the principles of thermodynamics. In *Parallel Problem Solving from Nature-PPSN VIII*, 922–931, Springer.