

UC Merced

UC Merced Electronic Theses and Dissertations

Title

ALGORITHMS FOR MULTI-OBJECTIVE OPTIMIZATION OF DYNAMICAL SYSTEMS

Permalink

<https://escholarship.org/uc/item/1wx964cx>

Author

Naranjani, Yousef

Publication Date

2016

Peer reviewed|Thesis/dissertation

UNIVERSITY OF CALIFORNIA, MERCED

**ALGORITHMS FOR MULTI-OBJECTIVE
OPTIMIZATION OF DYNAMICAL SYSTEMS**

by

Yousef Naranjani

A dissertation submitted in partial satisfaction of the
requirements for the degree of
Doctor of Philosophy

in

Mechanical Engineering

Committee in charge:
Professor Jian-Qiao Sun, Chair
Professor YangQuan Chen
Professor Roummel Marcia

©2016 Yousef Naranjani
All rights are reserved.

The thesis of Yousef Naranjani is approved:

Jian-Qiao Sun, Chair

Date

YangQuan Chen

Date

Roummel Marcia

Date

University of California, Merced

©2016 Yousef Naranjani

To my dear wife Negin, and my parents.

ACKNOWLEDGMENTS

Firstly, I would like to express my utmost gratitude to my advisor Prof. Jian-Qiao Sun for the continuous support, patience, motivation, and immense knowledge. In addition to being a wonderful advisor, he has been my role model. I learned much about the philosophy of life and science from his lifestyle and manner.

Besides my advisor, I would like to thank the rest of my thesis committee: Prof. Yang-Quan Chen and Prof. Roummel Marcia for their insightful comments and encouragement, and proofreading this dissertation.

My sincere thanks also goes to my collaborators Dr. Yousef Sardahi, Mr. Carlos Hernández, Dr. Fu-Rui Xiong, Professor Oliver Schütze and Mr. Wei Liang, whose efforts resulted in many publications together.

Last but not the least, I would like to thank my dear wife, Negin. Through all these years, we laughed and cried and learned together and from each other. She has been my best friend and supporter and I would definitely not be able to make it without her.

CURRICULUM VITAE

Education

B.S. in Mechanical Engineering, Ferdowsi University of Mashhad (Mashhad, IRAN), 2008.

M.S. in Mechanical Engineering, Ferdowsi University of Mashhad (Mashhad, IRAN), 2011.

Honors

Graduate Dean's Dissertation Fellowship Award (2016), University of California Merced.

SoE Dean's Travel Award (2015, 2016), University of California Merced.

Bobcat Fellowship Award (2014, 2015, 2016), University of California Merced.

SoE Travel Award (2014), University of California Merced.

MEAM graduate fellowship award (2013), University of California Merced.

Graduate Tuition Scholarship (2012 - 2016), University of California Merced.

Tuition Scholarship (2008 - 2011), Ferdowsi University of Mashhad.

Tuition Scholarship (2003 - 2008), Ferdowsi University of Mashhad.

Publications

Yousef Naranjani, Jian-Qiao (2016) "Multi-Objective Optimal Airfoil Design for Small Single Engine Aircraft ". *Submitted to AIAA Journal*.

Yousef Naranjani, Jian-Qiao (2016) "Multi-Objective Optimal Airfoil Design for Cargo Aircrafts ". *Accepted to ASME 2016 International Mechanical Engineering Congress and Exposition*.

Y. Naranjani, C. Hernández, F.R. Xiong, O. Schütze and J.-Q. Sun (2016) "A hybrid method of evolutionary algorithm and simple cell mapping for multi-objective optimization problems". *International Journal of Dynamics and Control*, 1-13.

Yousef Naranjani, Jian-Qiao (2015) "A Multi-Objective Path Planning Algorithm for Mobile Robots Based on Cellular Automata ". *Proceedings of ASME AM3D/IDETC/CIE 2015*.

Yousef Naranjani, Yousef Sardahi, YangQuan Chen, Jian-Qiao Sun (2015) "Multi-Objective Optimization of Distributed-Order Fractional Damping". *Communications in Nonlinear Science and Numerical Simulation*. 24(1:3), 159-168.

Naranjani, Y., et al. (2014) “A genetic algorithm and cell mapping hybrid method for multi-objective optimization problems”. *11th International Conference on Electrical Engineering, Computing Science and Automatic Control (CCE)*.

Y. Naranjani, Y. Sardahi, Jesús Fernández, O. Schütze and J.-Q. Sun (2014) “A Simple Cell Mapping and Genetic Algorithm Hybrid Method for Multi-Objective Optimization Problems”. *EVOLVE-A Bridge between Probability, Set Oriented Numerics, and Evolutionary Computation V*.

Yousef Sardahi, Yousef Narajani, Wei Liang, Fu-Rui Xiong, Zhi-Chang Qin, Yang Xue, Carlos Hernández, Oliver Schütze, Jian-Qiao Sun (2013) “Multi-objective Optimal Design of Feedback Controls for Dynamical Systems Time Delay”. *Proceedings of CSTAM CONFERENCE-2013*.

Y. Naranjani, Y. Sardahi, J.-Q. Sun, C. Hernández and O. Schütze (2013) “Fine Structure of Pareto Front of Multi-objective Optimal Feedback Control Design”. *Proceedings of the 2013 ASME Dynamic Systems and Control Conference - Control, Monitoring, and Energy Harvesting of Vibratory Systems. Stanford, California. DSCC2013-3944*.

Y. Sardahi, Y. Naranjani, W. Liang, J. Q. Sun, C. Hernández and O. Schütze (2013) “Multi-objective Optimal Control Design with the Simple Cell Mapping Method”. *Proceedings of ASME 2013 International Mechanical Engineering Congress & Exposition. San Diego, California. IMECE2013-65506*.

Xiong, F., Qin, Z., Hernández, C., Sardahi, Y., Narajani, Y., Liang, W., Xue, Y., Schütze, O., Sun, J., (2013) “A multi-objective optimal PID control for a nonlinear system with time delay”. *Theoretical and Applied Mechanics Letters*. 3(6), 9-063006 .

Naranjani, Y., Hernández, C., Xiong, F. R., Schütze, O., & Sun, J. Q. (2013) “A Hybrid Algorithm for the Simple Cell Mapping Method in Multi-objective Optimization”. In *EVOLVE-A Bridge between Probability, Set Oriented Numerics, and Evolutionary Computation IV*. 207-223.

C. Hernández, Y. Naranjani, Y. Sardahi, W. Liang, O. Schütze, J.Q. Sun (2013) “Simple Cell Mapping Method for Multi-objective Optimal Feedback Control Design”. *International Journal of Dynamics and Control*. 1(3), 231-238.

TABLE OF CONTENTS

ACKNOWLEDGMENTS	ii
CURRICULUM VITAE	iii
LIST OF FIGURES	viii
LIST OF TABLES	xiii
ABSTRACT	xv

Chapter

1 INTRODUCTION	1
1.1 Motivation	1
1.2 Objectives and Organization of the Thesis	2
2 MULTI-OBJECTIVE OPTIMIZATION	3
2.1 MOP Definition	3
2.2 Solution Methods	4
2.2.1 Scalarization methods	4
2.2.2 Descent direction methods	6
2.2.3 Stochastic methods	9
3 A HYBRID METHOD OF EA AND SCM FOR MOPS	10
3.1 Introduction	10
3.2 The EA+SCM Hybrid Method	11
3.3 Case Studies	16
3.3.1 Kursawe’s Problem	17
3.3.2 Deb’s Problem	19
3.3.3 Viennet’s Second Problem	21

3.3.4	PID Control for Duffing-Van der Pol Nonlinear Oscillator with Time Delay	25
3.4	Notes on Parallel Computing	30
3.5	Summary	36
4	SCM METHOD AND FEEDBACK CONTROL DESIGN	41
4.1	Introduction	41
4.2	Linear Control System	43
4.3	Time-Delayed Linear Control System	44
4.4	Simple Cell Mapping Method	44
4.5	Multi-objective Optimal Design of Feedback Controls	46
4.6	Numerical Examples	46
4.6.1	First Order System plus Time Delay	46
4.6.2	Second Order Linear Oscillator	48
4.7	Summary	51
5	MOP OF DISTRIBUTED-ORDER FRACTIONAL DAMPING	58
5.1	Introduction	58
5.2	Distributed-Order Fractional Damper	59
5.3	Objective Functions	60
5.4	Distribution Function $c(\alpha)$	61
5.5	Results and Discussions	62
5.6	Summary	69
6	MULTI-OBJECTIVE OPTIMAL PATH PLANNING	74
6.1	Introduction	74
6.2	Method	75
6.2.1	Framework	75
6.2.2	Cellular Automata	76
6.3	Case Studies	76
6.3.1	Obstacle/Radar Avoidance	76
6.3.2	Climb Slope Constraint	77

6.3.3	Randomized Terrain	77
6.4	Summary	77
7	AIRFOIL OPTIMIZATION	84
7.1	Introduction	84
7.2	Analysis Method	85
7.2.1	PARSEC parametrization	85
7.2.2	Solver	86
7.2.3	MOPSO	86
7.2.4	Problem Setup	86
7.2.5	Post-Processing	87
7.3	Results	87
7.4	Summary	90
8	CONCLUSION AND FUTURE WORK	100
	BIBLIOGRAPHY	102

LIST OF FIGURES

3.1	The Pareto front of Kursawe’s problem.	18
3.2	The Pareto set of Kursawe’s problem. The legend is the same as Figure 3.1.	18
3.3	Convergence of Pareto front in Kursawe’s problem.	19
3.4	Convergence of Pareto set in Kursawe’s problem. The legend is the same as Figure 3.3.	20
3.5	The Pareto front of Deb’s problem. The legend is the same as Figure 3.1.	21
3.6	The Pareto set of Deb’s problem. The legend is the same as Figure 3.1.	22
3.7	Convergence of Pareto front in Deb’s problem. Legend is the same as Figure 3.3.	23
3.8	Convergence of Pareto set in Deb’s problem. The legend is the same as Figure 3.3.	24
3.9	The Pareto front of Viennet’s second problem. The legend is the same as Figure 3.1.	25
3.10	The Pareto set of Viennet’s second problem. The legend is the same as Figure 3.1.	26
3.11	Convergence of Pareto front of Viennet’s problem. Legend is the same as Figure 3.3.	27
3.12	Convergence of Pareto set of Viennet’s problem. Legend is the same as Figure 3.3.	28

3.13	The Pareto front of the time delayed Duffing-Van der Pol system.	31
3.14	The Pareto set of the PID control gains for the Duffing-Van der Pol system.	32
3.15	The closed-loop step responses (top) and control (bottom) of the Duffing-Van der Pol system.	33
3.16	Convergence of Pareto front in PID control problem. Legend is the same as Figure 3.3.	34
3.17	Convergence of Pareto set in PID control problem. Legend is the same as Figure 3.3.	35
4.1	The Pareto set of $[Q_1, Q_2, R]$ for the multi-objective LQR optimal control of the first order system with time delay. The color code indicates the level of the other design variable. Red denotes the highest value, and dark blue denotes the smallest value.	49
4.2	The Pareto front of $[t_p, M_p, e_{IAE}]$ for the multi-objective LQR optimal control of the first order system with time delay corresponding to the Pareto set in Figure 4.1. The color code indicates the level of the other objective function. Red denotes the highest value, and dark blue denotes the smallest value.	50
4.3	The first two dominant closed-loop poles for the multi-objective LQR optimal control of the first order system with time delay corresponding to the Pareto set in Figure 4.1. The color code indicates the level of the first objective function t_p . Red denotes the highest value, and dark blue denotes the smallest value.	51
4.4	An example of the closed-loop step response and control of the first order system with time delay. The PID gains are designed with the multi-objective LQR optimal control formulation. The optimal design parameters are $[Q_1, Q_2, R] = [3.10, 495.0, 0.012]$. The optimal gains for the original state \mathbf{x} are $\mathbf{k}_x = [53.62, 110.93]$. The performance indices are $[t_p, M_p, e_{IAE}] = [1.05, 0.1839, 0.4407]$	52

4.5	The Pareto set of $[k_p, k_i, k_d]$ for the multi-objective optimal control of the second order system. The color code indicates the level of the other design variable. Red denotes the highest value, and dark blue denotes the smallest value.	53
4.6	The Pareto front of $[t_p, M_p, e_{IAE}]$ for the multi-objective optimal control of the second order system corresponding to the Pareto set in Figure 4.5. The color code indicates the level of the other objective function. Red denotes the highest value, and dark blue denotes the smallest value.	54
4.7	The closed-loop poles for the multi-objective optimal control of the second order system corresponding to the Pareto set in Figure 4.5. The color code indicates the level of the second objective function M_p . Red denotes the highest value, and dark blue denotes the smallest value.	55
4.8	An example of the closed-loop step response and control of the second order system with $[k_p, k_i, k_d] = [40.0, 2.8796, 1.9792]$. The PID gains are designed with the multi-objective optimal control. $[t_p, M_p, e_{IAE}] = [0.1555, 0.0, 0.2774]$	56
5.1	The Pareto front of the system in Case I. The color code is based on the objective which is not present in the plot. In all figures, the red color represents higher value and the blue color represents lower value.	63
5.2	The Pareto set of the system in Case I. The color code is based on the value of the variable which is not present in the plot.	64
5.3	The step responses of the system in Case I with different design priorities for the objectives t_p , M_p and J_{ISE} . The last case shows the best integer order damper.	65
5.4	The integral of the squared difference between the step response of the integer order system and the fractional order MOP design as a function of the integer order damping coefficient. The best integer order damping coefficient to match the MOP design is $c = 1.208$ and $\Delta = 0.0231$	66

5.5	The Pareto front of the system in Case II. The color code is based on the value of the objective which is not present in the plot.	67
5.6	The Pareto set of the system in Case II.	68
5.7	The step responses of the system in Case II with different design priorities for the objectives t_p , M_p and J_{ISE}	68
5.8	The Pareto front for Case III. The color code is based on the value of the objective which is not present in the plot.	70
5.9	The Pareto set for Case III. The color code of the left plot is based on the value of β and the right plot is based on a	71
5.10	The step responses of the system in case III with different design priorities for the objectives t_p , M_p and J_{ISE}	72
6.1	Moore neighbors of the blue cell are shown in red. They are the only cells considered in the CA rule for the center cell.	80
6.2	Case study 1 results. The blue circle is the start point and the green triangle shows the goal. The contours show the radars active range and its color corresponds to the radar penalty factor value. The black boxes represent the obstacles. Red lines show the feasible paths suggested by the algorithm.	80
6.3	Case study 2 results. The terrain color is based on the elevation. Because choosing the shortest possible path would violate the slope constraint, the algorithm found a path through the series of ramps that is the shortest feasible path. The other found paths are longer but provide less elevation changes. Blue circle and green triangle are the start point and end point respectively.	82
6.4	Case study 3 results. The terrain color is based on the elevation. The contours show the range of the radar.	83
7.1	Design parameters for PARSEC shape functions	85
7.2	Hyper-volume convergence analysis of the solution. The case with selected initial points outperform random initial points as expected.	88

7.3	Pareto front of the airfoil MOP. the grey shadows show the 2D projection of the solution.	89
7.4	Variation of the design parameters between the branches B1-B4. The mean of the branch parameters are connected with a line while the error bar shows the confidence interval for the variations of the parameter. These results are quantitatively studies in Table 7.3 . . .	91
7.5	Four airfoils from the solution set. The first three correspond to the extreme objectives and the last case is the knee point selected by the post-processing algorithm. The objective and design parameter values can be found at Table 7.2.	92
7.6	Four airfoils each representing the knee point for every branch B1-B4. The objective and design parameter values can be found at Table 7.2.	93
7.7	The aerodynamic properties (polars) of the the first four selected airfoils from Table 7.2. The marked points on each curve shows the optimal operational point. This point corresponds to the attack angle where $\max(C_L/C_D)$ is observed.	94
7.8	The aerodynamic properties (polars) of the the second four selected airfoils from Table 7.2. The marked points on each curve shows the optimal operational point. This point corresponds to the attack angle where $\max(C_L/C_D)$ is observed.	95
7.9	The pressure distribution of the the first four selected airfoils from Table 7.2. The curves are calculated at the operational angle of attack for each airfoil as shown in Figure 7.7.	96
7.10	The pressure distribution of the the second four selected airfoils from Table 7.2. The curves are calculated at the operational angle of attack for each airfoil as shown in Figure 7.8.	97

LIST OF TABLES

3.1	Comparison of the EA-SCM hybrid method with GA and SCM when applied to Kursawe’s problem.	37
3.2	Comparison of the EA-SCM hybrid method with GA and SCM when applied to Deb’s problem.	38
3.3	Comparison of the EA-SCM hybrid method with GA and SCM when applied to Viennet’s second problem.	39
3.4	Comparison of the EA-SCM hybrid method with GA when applied to time delayed Duffing-Van der Pol system.	40
5.1	Different objective function values of the designs shown in Figure 5.3 for Case I.	63
5.2	Different objective function values of the designs shown in Figure 5.7 for Case II.	69
5.3	Different objective function values of the designs shown in Figure 5.10 for Case III.	70
6.1	The path finding algorithm. This post processing step finds the optimal path between the start point and the goal while it also calculates the path length L_p , total height change H_p and total REP Γ_p along the path knowing the starting cell <i>initCell</i> , the CA state values S , the height map Z and the maximum allowable slope m_{all}	78
6.2	Details of the optimal paths marked in Figure 6.2.	79
6.3	Details of the optimal paths marked in Figure 6.3.	79
6.4	Details of the optimal paths marked in Figure 6.4.	81

7.1	Physical parameters used in the PARSEC parametrization and their bounds.	98
7.2	Selected airfoils from the solution set with corresponding objective and parameter values. The first three entries are extreme cases and the last one is selected by the postprocessing algorithm.	98
7.3	The results for the statistical comparison between the branches B1-B4. Each of the 12 parameters has been compared between the branches using the independent sample ttest. P values under 0.05 show a significant difference between the groups.	99

ABSTRACT

Multi-Objective Optimization Problems (MOPs) deal with optimizing several objectives simultaneously and have diverse applications in engineering, economics, logistics, etc. The methods for solving MOPs can generally be classified into stochastic and deterministic approaches. Deterministic approaches are capable of finding the global solution even though they are computationally burdensome. Stochastic methods, on the other hand, can save on computations significantly, although they do not guarantee to find the global solution.

In engineering applications, MOPs can become nonlinear, multi-modal, high dimensional, and have complex structured solutions that makes them more challenging.

This theses follows two major goals. Firstly, it presents new methods and algorithms for solving engineering MOPs by hybridizing the existing methods and comparing their effectiveness by using benchmark problems. The hybrid method combines an evolutionary algorithm with a cell mapping method in order to reduce the computational time while maintaining the quality of the solution. Implementation details on parallel CPU/GPU programming of such methods are discussed as well. The second goal of this thesis is to introduce new applications for MOPs in different areas of engineering such as control design, path planning, fractional systems and airfoil design.

Chapter 1

INTRODUCTION

1.1 Motivation

Multi-objective optimization problems (MOPs) concern with simultaneously minimizing a set of usually conflicting objective functions. These problems have numerous applications in optimal control design, economics, finance *etc.* Instead of a point, the solution of a MOP consists of a set of points called the *Pareto set* and the corresponding set of objective function values are called the *Pareto front*. To solve these problems, both deterministic and stochastic approaches have been thoroughly investigated.

Evolutionary algorithms (EAs) are mainstream stochastic methods for MOPs [1]. In the family of EAs, genetic algorithms (GAs) are popular choices for engineering applications [2]. In this thesis, NSGA-II, a variant of GAs is used for comparison purposes. It should be noted that other EAs such as particle swarm optimization (PSO) and strength Pareto evolutionary algorithm (SPEA) can all be considered. Although GAs might be a convenient choice for solving MOPs, the accuracy in finding the global solution and speed of convergence are two of their drawbacks [3]. For instance, Shuai and Zhou [4] reported that in some cases, GAs may get stuck at a local optimum. Furthermore, GAs have a variety of operators, the best choice of which can be problem dependent. Elsayed *et al.* [5] showed that there is no single GA operator that can work well with all problems. To improve its performance, the notion of merging the GA with other methods has received much attention specially in the area of single objective optimization [3,5,6]. The trend of combining GAs with other techniques has also been found in the area of multi-objective optimization. For example, Zeidi *et al.* [7] presented a hybrid multi-objective approach based on the genetic algorithm and artificial neural network. In another study, Wang *et al.* [8] integrated the worst-case-scenario technique of anti-optimization with a constrained multi-objective genetic algorithm.

Hernández *et al.* [9] have found deterministic global search algorithms such as the simple cell mapping (SCM) to be effective in solving MOPs. The cell mapping methods describe the system dynamics with cell-to-cell mappings by discretizing the phase space. The dynamics of a cell is represented by that of its center [10]. The SCM method is effective to investigate the global solution of nonlinear dynamical systems. The SCM method can be gradient based or gradient free [11]. Although

the SCM method can find the global and fine structure of optimal solutions, its computational time increases dramatically as the dimension of the design space goes up [12]. The computational time can be reduced if we apply the SCM method only in the vicinity of the solution instead of sweeping the entire design space. This approach is particularly beneficial to MOPs of high dimensions, and is also the key feature for the set oriented numerics (SON) [13–15]. Subdivisions can also be applied to improve the accuracy and resolution of the solution.

The advantages and disadvantages of the methods mentioned, motivated us to create a hybridized method that utilizes the power points of both methods and can outperform them.

1.2 Objectives and Organization of the Thesis

The main objectives of this thesis is listed as follows,

- Implementing Simple Cell Mapping method for MOPs and evaluate its performance.
- Hybridizing GA and SCM to create a more efficient algorithm specially for engineering MOPs.
- Evaluating the performance of the hybrid algorithm by comparing it with other methods using benchmark problems.
- Study applications of MOPs in various areas of engineering including
 1. Control engineering
 2. Robot path planning
 3. Fractional systems
 4. Airfoil design

The organization of the thesis is as follows. Chapter 2 defines the MOP problem and briefly reviews the methods of solution. The hybrid EA+SCM method is presented and evaluated in Chapter 3. The SCM method is discussed in Chapter 4 followed by some control MOPs solved with this method. The rest of the thesis is followed by additional new applications of MOPs in engineering. Chapter 5 looks into a system with fractional order damping. A multi-objective robot path planning scheme using cellular automaton is presented in Chapter 6. Chapter 7 focuses on multi-objective optimal airfoil design. Finally the conclusion is made in Chapter 8.

Chapter 2

MULTI-OBJECTIVE OPTIMIZATION

2.1 MOP Definition

A multi-objective optimization problem can be stated as follows,

$$\min_{\mathbf{x} \in Q} \{\mathbf{F}(\mathbf{x})\}, \quad (2.1)$$

where $\mathbf{x} \in Q$ is q -dimensional vector of design parameters. \mathbf{F} is the map that consists of the objective functions $f_i : Q \rightarrow \mathbf{R}^1$.

$$\mathbf{F} : Q \rightarrow \mathbf{R}^k, \mathbf{F}(\mathbf{x}) = [f_1(\mathbf{x}), \dots, f_k(\mathbf{x})]. \quad (2.2)$$

The design parameter space $Q \subset \mathbf{R}^q$ can in general be expressed in terms of inequality and equality constraints,

$$Q = \{\mathbf{x} \in \mathbf{R}^q \mid h_j(\mathbf{x}) \leq 0, j = 1, \dots, l, \text{ and } h_j(\mathbf{x}) = 0, j = l + 1, \dots, m\}. \quad (2.3)$$

The solutions of the MOP are defined by the concept of *dominance* [16], which is described by the following definitions.

(a) Let $\mathbf{v}, \mathbf{w} \in \mathbf{R}^k$. The vector \mathbf{v} is said to be *less than* \mathbf{w} (in short: $\mathbf{v} <_p \mathbf{w}$), if $v_i < w_i$ for all $i \in \{1, \dots, k\}$. The relation \leq_p is defined analogously.

(b) A vector $\mathbf{v} \in Q$ is called *dominated* by a vector $\mathbf{w} \in Q$ ($\mathbf{w} \prec \mathbf{v}$) with respect to the MOP (2.1) if $\mathbf{F}(\mathbf{w}) \leq_p \mathbf{F}(\mathbf{v})$ and $\mathbf{F}(\mathbf{w}) \neq \mathbf{F}(\mathbf{v})$, else \mathbf{v} is called non-dominated by \mathbf{w} .

If a vector \mathbf{w} dominates a vector \mathbf{v} , then \mathbf{w} can be considered to be a ‘better’ solution of the MOP. The definition of optimality or the ‘best’ solution of the MOP is now straightforward.

(c) A point $\mathbf{w} \in Q$ is called *Pareto optimal* or a *Pareto point* of the MOP (2.1) if there is no $\mathbf{v} \in Q$ which dominates \mathbf{w} .

(d) The set of all Pareto optimal solutions is called the *Pareto set* denoted as

$$\mathcal{P} := \{\mathbf{w} \in Q : \mathbf{w} \text{ is a Pareto point of MOP (2.1)}\} \quad (2.4)$$

The image $\mathbf{F}(\mathcal{P})$ of \mathcal{P} is called the *Pareto front*.

2.2 Solution Methods

Since the behaviour of the objectives in a MOP is naturally conflicting, the solution would be a set instead of a point. The set of solution points is called the Pareto set and the corresponding set of objective function values are called the Pareto front. The solution methods provide us with a finite approximation of the Pareto front and follow two major goals. The first is to converge to the real Pareto front *i.e.* that the finite approximation is distributed along the Pareto set/front. The second one is to have a nicely spread solution *i.e.* that the distance between the approximated points of the Pareto front is ideally the same.

The solution approximation methods can be categorized from different perspectives. In this thesis, we divide them into three categories, scalarization methods, descent direction methods and stochastic methods which will be discussed below.

2.2.1 Scalarization methods

One of the classical ideas to solve a MOP is to transform the problem into an auxiliary Single-objective Optimization Problem (SOP). Using this method, we will reduce the number of objectives to one. Once the SOP is created, we are able to use numerous available methods to solve the proposed SOP. We should note that the solution of a SOP is consist of a single point while the solution of MOP is a set. As a result, the Pareto set can be approximated (but not guaranteed to be entirely found) by smartly solving a sequence of SOPs [17]. We will now briefly overview the most popular schalarization techniques. For a more detailed discussion please refer to [17].

Weighted sum method

It is probably the oldest scalarization method. The idea is to assign each objective a certain weight $\alpha_i \geq 0$, and to minimize the resulting weighted sum [17]. Thus the problem (2.1) can be restated as follows,

$$\begin{aligned} \min_{\mathbf{x} \in Q} f_{\alpha}(\mathbf{x}) &:= \sum_{i=1}^k \alpha_i f_i(\mathbf{x}) & (2.5) \\ \text{s.t. } \mathbf{x} &\in Q \\ \alpha_i &\geq 0, \quad i = 1, \dots, k, \\ \sum_{i=1}^k \alpha_i &= 1 \end{aligned}$$

The main advantage of the weighted sum method is that the solution of Equation (2.5) for any set of $\alpha_i \geq 0$ would result in a Pareto optimal solution. On the other hand, the proper choice of α can be a delicate problem. If the Pareto front happens to be concave, some parts of it may not be located with the global solution to Equation (2.5). That means, not all the points of the Pareto front can be reached, when using the weighted sum approach, which could be a major disadvantage.

ϵ - constrained method

The idea of this method is to select one objective, say $f_i, i \in \{1, \dots, k\}$, and to treat all other objectives as constraints by imposing upper bounds on the function values [18]. As a result, the optimization problem translates to,

$$\begin{aligned} & \min_{\mathbf{x} \in Q} f_i(\mathbf{x}) & (2.6) \\ \text{s.t.} & f_j(\mathbf{x}) \leq \epsilon_j \quad \forall j \in \{1, \dots, k\} \setminus \{i\}. \end{aligned}$$

Regardless of the shape of the Pareto front, the ϵ -constrained method can find every Pareto optimal solution. However, the finding the right choice of ϵ could be a challenging problem itself.

Weighted Tchebycheff method

The goal of the weighted Tchebycheff method is to find a point whose image is closest to a given reference point $Z \in \mathbf{R}^k$ [17]. For the distance assignment, the weighted Tchebycheff metric is used: Let $\alpha \in \mathbf{R}^k$ with $\alpha_i \geq 0, i = 1, \dots, k$, and $\sum_{i=1}^k \alpha_i = 1$, and let $Z = (z_1, \dots, z_k)$, then the weighted Tchebycheff method reads as follows,

$$\min_{\mathbf{x} \in Q} \max_{i=1, \dots, k} \alpha_i |f_i(x) - z_i|. \quad (2.7)$$

Please note that the solution to Equation (2.7) depends on Z as well as α . The main advantage of this method is that by proper choice of these two vectors, every point on the Pareto front can be found.

Normal boundary intersection

The Normal Boundary Intersection (NBI) method computes the finite size approximation of the Pareto front in two steps [19]:

1. Computing the Convex Hull of Individual Minima (CHIM), which is a $(k - 1)$ -simplex connecting the objective values of the minimum of each objective $f_i, i = 1, \dots, k$ (*i.e.* the utopian).
2. Selecting points y_i from CHIM and computing the point $\mathbf{x}_i^* \in Q$ such that the image $F(\mathbf{x}_i^*)$ has the maximal distance from y_i in the direction that is normal to the CHIM and points toward the origin.

The second step is called the NBI-subproblem and can be stated in mathematical terms as follows: Given and initial value \mathbf{x}_0 and a direction $\alpha \in \mathbf{R}^k$, solve

$$\begin{aligned} & \max_{\mathbf{x}, l} f_i(\mathbf{x}) & (2.8) \\ \text{s.t.} & F(\mathbf{x}_0) + l\alpha = F(\mathbf{x}) \\ & \mathbf{x} \in Q. \end{aligned}$$

This formulation could be helpful since there are some cases in which our goal is to steer the search in a certain direction in the objective space. On the other hand, the solutions of Equation (2.8) do not have to be Pareto optimal [19].

2.2.2 Descent direction methods

Given a descent direction ν at a point \mathbf{x} , a further candidate solution \mathbf{x}_{new} can easily be found by a line search, *i.e.* by setting

$$\mathbf{x}_{new} = \mathbf{x} + t\nu,$$

where $t \in \mathbf{R}_+$ is the step size. The solution to this problem would result in a curve of dominant points, *i.e.* the new points will dominate the starting points.

In the following, we review some methods that use this idea to find the descent direction ν .

Simplest bi-objective descent direction

This method finds the descent direction by combining two gradients by a vector sum. This method states that if $f_1, f_2 : \mathbf{R}^n \rightarrow \mathbf{R}$ define a two objective MOP and if $\nabla f_i \neq 0$, for $i = 1, 2$, then

$$\nu(\mathbf{x}_0) = - \left(\frac{\nabla f_1(\mathbf{x})}{\|\nabla f_1(\mathbf{x})\|} + \frac{\nabla f_2(\mathbf{x})}{\|\nabla f_2(\mathbf{x})\|} \right) \quad (2.9)$$

is a descent direction at \mathbf{x}_0 for the MOP.

The drawback for this method is that it cannot be generalized for more than two objectives.

Directed search

The directed search method enables us to steer the search direction from a given point $\mathbf{x} \in Q$ into a desired direction $\mathbf{d} \in \mathbf{R}^k$. A direction vector $\mathbf{v} \in \mathbf{R}^n$ can be calculated such that

$$\lim_{t \rightarrow 0} \frac{f_i(\mathbf{x}_0 + t\mathbf{v}) - f_i(\mathbf{x}_0)}{t} = d_i, \quad i = 1, \dots, k. \quad (2.10)$$

The next two methods presented, use this idea. The first one is a descent method that steers toward a given direction d . The second method is a continuation method with the particular advantage that unlike other methods, this method does not need any second gradient information [20, 21].

Descent method

Assume a point $\mathbf{x}_0 \in Q$ is given as well as the vector $\mathbf{d} \in \mathbf{R}^k$ which represents the desired direction in objective space. It can be expressed as follows

$$J(\mathbf{x})\mathbf{v} = \mathbf{d}, \quad (2.11)$$

where $\mathbf{v} \in \mathbf{R}^n$ is a search direction in parameter space and $J(\mathbf{x})$ is the Jacobian matrix defined by

$$J(\mathbf{x}) = \begin{bmatrix} \frac{\partial f_1}{\partial x_1}(x) & \cdots & \frac{\partial f_1}{\partial x_n}(x) \\ \vdots & \ddots & \vdots \\ \frac{\partial f_k}{\partial x_1}(x) & \cdots & \frac{\partial f_k}{\partial x_n}(x) \end{bmatrix}. \quad (2.12)$$

As a result, \mathbf{v} can be determined by solving a system of linear equations. Since usually the number of parameters is higher than the number of objectives, the system of equations will be undetermined, causing the solution to be not unique. To handle this issue, the problem can be formulated as

$$\mathbf{v} = J(\mathbf{x}_0)^+ \mathbf{d} \quad (2.13)$$

where $J(\mathbf{x}_0)^+$ denoted the pseudo inverse of the Jacobian $J(\mathbf{x}_0) \in \mathbf{R}^{k \times n}$. Further, we can solve the following initial value problem (IVP):

$$\begin{aligned} \mathbf{x}(0) &= \mathbf{x}_0 \in \mathbf{R}^n \\ \dot{\mathbf{x}}(m) &= \mathbf{v}_\alpha(\mathbf{x}(m)), t > 0. \end{aligned} \quad (2.14)$$

Continuation method

As soon as one optimal point is reached with the method above, this method starts a movement along the Pareto set of a given MOP.

Assume a Pareto point \mathbf{x} and a convex weight α is given such that

$$\sum_{i=1}^k \alpha_i \nabla f_i(\mathbf{x}) = 0, \quad (2.15)$$

and further we assume that

$$\text{rank}(J(\mathbf{x})) = k - 1. \quad (2.16)$$

It is known that in this case α is orthogonal to the Pareto front [20]. As a result, a search orthogonal to α in objective space could be promising to find new predictor points. A QR -factorization of α can be computed to use this method, *i.e.*,

$$\alpha = QR, \quad (2.17)$$

where Q is an orthogonal matrix consist of the column vectore $\mathbf{q}_1, \dots, \mathbf{q}_k$. Since by Equation (2.17) $\alpha = r_{11}\mathbf{q}_1$ and Q is orthogonal, it follows that the column vectors $\mathbf{q}_2, \dots, \mathbf{q}_k$ build and orthonormal basis of hyperplane which is orthogonal to α . Thus, a promising set of search directions \mathbf{v}_i maybe the ones which satisfy

$$J(\mathbf{x})\mathbf{v}_i = \mathbf{q}_i, i = 2, \dots, k. \quad (2.18)$$

Since α is not in the image of $J(\mathbf{x})$ (otherwise \mathbf{x} would not be a Pareto point), it follows that the vectors $\mathbf{q}_2, \dots, \mathbf{q}_k$ are in the image of $J(\mathbf{x})$, *i.e.* Equation (2.18)

can be solved for each $i \in \{2, \dots, k\}$. Then, the set of predictor directions can be chosen as:

$$\mathbf{p}_i = \mathbf{x}_0 + t\mathbf{v}_i. \quad (2.19)$$

Please note that this method of selecting predictor direction does not require the second derivative of the objectives.

At this point, a corrector step can be used. Given a predictor $\mathbf{p}_i \in \mathbf{p}$, we can use \mathbf{p}_i as initial value to Equation (2.14) and choosing α_0 *i.e.*, the weight from the previous solution \mathbf{x}_0 leads to a new solution \mathbf{x}_1 .

Method of Schäffler Schültz and Weinzierl

The following function is defined [22]:

$$\mathbf{q}(\mathbf{x}) = \sum_{i=1}^k \hat{\alpha}_i \nabla f_i(\mathbf{x}), \quad (2.20)$$

where $\mathbf{q} : \mathbf{R}^n \rightarrow \mathbf{R}^n$ and $\hat{\alpha}$ is a solution of

$$\min_{\alpha \in \mathbf{R}^k} \left\{ \left\| \sum_{i=1}^k \alpha_i \nabla f_i(\mathbf{x}) \right\|_2^2, \alpha_i \geq 0, i = 1, \dots, k, \sum_{i=1}^k \alpha_i = 1 \right\}, \quad (2.21)$$

where ∇f_i is the gradient of i th objective function f_i .

From Equation (2.20) if we have $\mathbf{q}(\mathbf{x}) \neq \mathbf{0}$ then $-\mathbf{q}(\mathbf{x})$ is a descent direction for all the objective functions. As a result, each \mathbf{x} with $\mathbf{q}(\mathbf{x}) = \mathbf{0}$ fulfills the first-order necessary condition for Pareto optimality.

Method of Fliege and Svaiter

According to the authors [23], the following function is defined:

$$f_{\mathbf{x}}(\mathbf{v}) = \max(A\mathbf{v})_i, \quad i = 1, \dots, k \quad (2.22)$$

where $f_{\mathbf{x}} : \mathbf{R}^n \rightarrow \mathbf{R}$. We can see that $f_{\mathbf{x}}$ is convex and homogeneous. Using this function the following problem is proposed:

$$\begin{aligned} & \min f_{\mathbf{x}}(\mathbf{v}) + \frac{1}{2} \|\mathbf{v}\|^2 \\ & \text{subject to } \mathbf{v} \in \mathbf{R}^n \end{aligned} \quad (2.23)$$

From this we have that, if \mathbf{x} is Pareto optimal, then $\mathbf{v}(\mathbf{x}) = \mathbf{0}$. If that is not the case, then $\mathbf{v}(\mathbf{x})$ is a descent direction.

Simple Cell Mapping

Simple cell mapping (SCM) method was originally created by Hsu [24, 25] for global analysis of dynamical system. Later, it was found out that they can be beneficial in solving MOPs. A more detailed description of this method is presented in Section 4.4.

2.2.3 Stochastic methods

An alternative to the classical methods that became popular recently are stochastic methods which are also known as Multi-Objective Evolutionary Algorithms (MOEAs). Their main idea is to mimic the evolution process by evolving a set of individuals (population) over a number of generations. A wide variety of methods are developed in this area. One advantage of MOEAs is that they do not require gradient information of the problem to be available, instead they rely on stochastic search procedures. Another advantage is that they give an approximation of the solution in one execution. Examples of these methods can be found in [1,26]. The major drawback of these methods is that they do not guarantee convergence toward the Pareto front. Some of the most popular methods in this category include nondominated sorting genetic algorithm (NSGA-II), multi-objective particle swarm optimization (MOPSO), strength Pareto evolutionary algorithm 2 (SPEA-2) and simulated annealing.

Chapter 3

A HYBRID METHOD OF EA AND SCM FOR MOPS

3.1 Introduction

Evolutionary algorithms (EAs) are mainstream stochastic methods for MOPs [1]. In the family of EAs, genetic algorithms (GAs) are popular choices for engineering applications [2]. In this study, we apply the NSGA-II, a variant of GAs. It should be noted that other EAs such as MOEA/D [27] and SMS-EMOA [28] can all be considered. Reasons for the high interest in EAs include that algorithms of this kind are very robust, do not require hard assumptions on the model, and allow to compute a finite size representation of the solution set, the Pareto set, in one single run [26–28]. Further, it is known that under certain (mild) assumptions on the generation process the algorithms converge in the limit and in the probabilistic sense toward finite size approximations of the solution sets with given approximation qualities [29, 30]. On the other hand, it is widely accepted that MOEAs need quite a few function evaluations in order to evolve to a suitable approximation of the set of interest. As one way out, researchers have proposed *hybrid* or *memetic* algorithms that hybridize local search techniques with EAs in order to obtain fast and reliable global search procedures. See, for example, [31–34].

In [9], the authors have found deterministic global search algorithms such as the simple cell mapping (SCM) to be effective in solving MOPs. The SCM method can be gradient based or gradient free [11]. Although the SCM method can find the global and fine structure of optimal solutions, its computational time increases dramatically as the dimension of the design space goes up [12]. The computational time can be reduced if we apply the SCM method only in the vicinity of the solution instead of sweeping the entire design space. This approach is particularly beneficial to MOPs of high dimensions, and is also the key feature for the set oriented numerics (SON) [13–15]. Subdivisions can also be applied to improve the accuracy and resolution of the solution.

Since the Pareto set forms under some mild regularity conditions locally a $(k - 1)$ -manifold, specialized continuation methods which perform a search *along* the Pareto set are very efficient if one (or more) solution is at hand. One of the first methods of that kind is proposed in [20] which has been coupled with set oriented methods leading to the *recovering algorithm* in [35, 36]. This method is by

construction of local nature and needs multiple starting points in case the Pareto set/front is disconnected.

Although many of the named methods can be successful in the mathematical sense, engineering problems might be more challenging. Certain requirements and delicacy of these problems has led scientists to utilize specific approaches for tackling them [37–39]. In fact many hybrid algorithms have been developed to solve these problems efficiently. Examples exist in a wide range of applications such as wind farm design [40], control design [41], product design [42] and aircraft analysis [43].

The main idea of the hybrid method is to use EAs to find a random collection of points close to the true Pareto set. The SCM method is then applied to the covering region of these random points. The benefit of using EAs is that it reduces the computational burden of searching the whole design space, because the Pareto set occupies only a small fraction of the design space. Furthermore, we use a small population size in EAs. Hence, the results of EAs do not cover the whole Pareto set. The SCM method is modified such that it checks the neighbor cells for possible optimal solutions. As a result, the SCM method can find not only a more accurate Pareto set, but also recover the part missed by EAs. Finally, subdivisions of the cells representing the approximate solution of MOPs are applied to enhance the accuracy of the Pareto set with fine structures. The EA+SCM hybrid method is compared to the EA and SCM methods when they are applied separately on a selection of test problems. In order to be fair in the comparison, the total number of function evaluations is kept to be nearly the same for all the methods.

This chapter is structured as follows. Section 3.2 explains the EA+SCM hybrid algorithm. Case studies are reported in Section 7.3.

3.2 The EA+SCM Hybrid Method

Simple Cell Mapping

The cell mapping methods describe the system dynamics with cell-to-cell mappings by discretizing the phase space. The dynamics of a cell is represented by that of its center [10]. The SCM method is effective to investigate the global solution of nonlinear dynamical systems. In this paper, the SCM method is applied to only a subset of the discretized domain of the design space, and will expand the set if needed. The simple cell mappings for MOPs are constructed with the help of a local search algorithm for optimal solutions [9, 11].

Genetic Algorithm

The GA is a heuristic search method belonging to the family of EAs. The process starts with a randomly selected population of candidate solutions in the design space. In each iteration or generation, the fitness or objective function of the population is evaluated. The superior ones are selected to be stochastically mutated or modified to create a new population for the next generation. With this repetitive procedure, the population eventually converges to the solution. According

to [44], the convergence rate of the evolutionary algorithms are high in the first few generations.

The Hybrid Method

As mentioned earlier, the hybrid method takes advantages of the GA and SCM. The hybrid method starts with the GA which returns a rough Pareto solution. The set of cells that contain all the solutions obtained with the GA is identified. The SCM method starts with this covering set of cells.

Algorithm 1 shows the steps of the hybrid method. The problem inputs include the number N of initial cell partitions, cell subdivision number sub , iterations of subdivisions $iter$, and objective functions F . The output is the set \mathcal{B}_l of cells, an approximation of the Pareto set \mathcal{P} .

Algorithm 1: The programming logic of the EA+SCM hybrid method

- Require:** EA configuration; Coarse cell space partition N ; Cell subdivision sub ;
Number of iteration $iter$; MOP definition F
- Ensure:** Final solution set \mathcal{B}_l
- 1: $\mathcal{B}_0 \leftarrow$ Run EA to find rough solution
 - 2: **for** $l = 1$ **to** $iter$ **do**
 - 3: $\tilde{\mathcal{B}}_l \leftarrow$ Explore(N, \mathcal{B}_{l-1}, F)
 - 4: $N \leftarrow sub \times N$
 - 5: $\mathcal{B}_l \leftarrow$ Mapping $\tilde{\mathcal{B}}_l$ to refined grid
-

NSGA-II, a multi-purpose variant of the genetic algorithm, has been used in this study. Because there are extensive studies and analyses available on this method [26, 45], it is not discussed further in this paper. The cell mapping method is implemented in the program named *Explore*. It takes a set of cells, and explores these cells and their neighbors to look for the possible solution. In the process, the parts of the Pareto set missed by the GA are recovered.

The details of the explore algorithm are presented in Algorithm 2. The set \mathcal{B} is a dynamically increasing array which contains all the cells we are interested. The objective function values for each cell cs in \mathcal{B} are compared to all its orthogonal neighbors $\mathcal{N}(cs)$ (to be more precise, the center points $center(s_i)$ of cells s_i and s_j are compared). If one or more dominant cells is found, the one with the steepest decent is chosen to be the destination cell and is added to \mathcal{B} . If no dominant neighbor is found, the current cell cs is a part of the solution and is added to the solution set \mathcal{B}_l . All its non-dominant neighbors are added to \mathcal{B} for further investigation later. The set \mathcal{B} grows initially, and stops growing when most cells in the Pareto set are discovered. When the solution set is found, the dominance of all the cells in the set is checked. The recovery procedure of the exploration is similar to the recovery process used in the continuation method for multi-objective optimization in [46],

which requires the knowledge of at least one exact solution point on the Pareto set. Furthermore, we should emphasize that the dominance relationship among the neighboring cells is applied to construct the simple cell mappings instead of using the Jacobian gradient matrix.

Algorithm 2: Explore algorithm for searching the Pareto optimal solution with recovery technique

Require: Intermediate solution set of cells \mathcal{B}_l ; MOP definition F ; Cell space partition N ;
Orthogonal neighbors function \mathcal{N} ; cell center function $center$

Ensure: Non-dominant solutions at current partition level \mathcal{B}_l

- 1: $S \leftarrow \mathcal{B}_l, \mathcal{B}_l \leftarrow \emptyset, i \leftarrow 0$
- 2: **while** $i < |S|$ **do**
- 3: $cs \leftarrow S_i, dest_cell \leftarrow cs, dist \leftarrow 0$
- 4: **for all** $s \in \mathcal{N}(cs)$ **do**
- 5: **if** $center(s) \prec center(cs)$ **and** $\|F(center(s)) - F(center(cs))\|_2 > dist$ **then**
- 6: $dest_cell \leftarrow s$
- 7: $dist \leftarrow \|F(center(s)) - F(center(cs))\|_2$
- 8: **if** $dest_cell \notin S$ **then**
- 9: $S \leftarrow S \cup dest_cell$
- 10: **if** $dest_cell = cs$ **then**
- 11: **if** $dest_cell \notin \mathcal{B}_l$ **then**
- 12: $\mathcal{B}_l \leftarrow \mathcal{B}_l \cup dest_cell$
- 13: **for all** $s \in \mathcal{N}(cs)$ **do**
- 14: **if** $center(cs) \not\prec center(s)$ **and** $\nexists b \in \mathcal{B}_l : b \prec s$ **then**
- 15: $S \leftarrow S \cup s$
- 16: $i \leftarrow i + 1$
- 17: $\mathcal{B}_l \leftarrow dominance_check(\mathcal{B}_l)$

Here we make a first attempt to investigate the algorithm EA+SCM theoretically. As the algorithm is defined on the discretized search space, error bounds (e.g., via Lipschitz estimations) for the analysis are hard to obtain in practice and may lead to unrealistic results as they may be too pessimistic. Instead, we follow the suggestion made in [13, 35] and investigate the underlying abstract algorithm where we consider *all* points in all cells of the collections. Note that GA+SCM performs a recovering and a subdivision step applied in a loop. For the abstract algorithm, however, only *one* iteration has to be performed as the following convergence analysis shows as both recovering and subdivision of the abstract algorithm are without error. That is, the initial recover step is performed starting from an

initial cell collection \mathcal{B} until no more non-dominated neighboring cells can be added to the current collection. In a second step, subdivision is started on the resulting cell collection \mathcal{B}_0 . Algorithm 3 shows the pseudo code of the abstract algorithm.

Algorithm 3: Abstract Algorithm of GA-SCM

Require: Cell collection $\mathcal{B} \subset P(\hat{Q}, d_0)$
Ensure: Sequence \mathcal{B}_l of cell collections

- 1: create a queue Q using \mathcal{B}
- 2: set $V := \mathcal{B} \{(\text{Start Recover})\}$
- 3: **while** Q is not empty **do**
- 4: $T \leftarrow Q.\text{dequeue}()$, where T is the current cell
- 5: **for all** $C \in \Delta(T)$ s.t. $\nexists b \in V : b \preceq C$ **do**
- 6: **if** $C \notin V$ **then**
- 7: add C to V
- 8: enqueue C onto Q
- 9: $\mathcal{B}_0 := \{v \in V : \nexists \tilde{v} \in V, \tilde{v} \preceq v\} \subset P(\hat{Q}, d_0) \{(\text{Start Subdivision})\}$
- 10: **for** $l = 1, 2, \dots$ **do**
- 11: Subdivision: construct $\hat{\mathcal{B}}_l \subset P(\hat{Q}, d_0 + l)$ from \mathcal{B}_{l-1} such that

$$\bigcup_{B \in \hat{\mathcal{B}}_l} B = \bigcup_{B \in \mathcal{B}_{l-1}} B$$

12: Selection: define the new collection $\mathcal{B}_l \subset P(\hat{Q}, d_0 + l)$ by

$$\mathcal{B}_l = \{b \in \hat{\mathcal{B}}_l : \nexists \tilde{b} \in \hat{\mathcal{B}}_l : \tilde{b} \preceq b\}$$

Hereby, we assume that the domain is contained inside

$$Q \subset \hat{Q} = [a_1, b_1] \times \dots \times [a_n, b_n] \subset \mathbb{R}^n, \quad (3.1)$$

where the bounds $a_i \leq b_i$, $i = 1, \dots, n$, are chosen accordingly. To realize the subdivision in lines 13-15 we consider multi-level partitions of \hat{Q} as described in [13]:

A n -dimensional cell B (or box) can be expressed as

$$B = B(c, r) = \{x \in \mathbb{R} : c_i - r_i \leq x_i \leq c_i + r_i, \\ i = 1, \dots, n\}, \quad (3.2)$$

where $c \in \mathbb{R}^n$ denotes the center and $r \in \mathbb{R}^n$ the box size, respectively. Every cell B can be subdivided with respect to the j^{th} coordinate. This division leads to two cells $B_-(c^-, \hat{r})$ and

$$\hat{r}_i = \begin{cases} r_i & \text{for } i \neq j \\ r_i/2 & \text{for } i = j \end{cases}, \quad (3.3)$$

$$\hat{c}_i = \begin{cases} c_i & \text{for } i \neq j \\ c_i/2 & \text{for } i = j \end{cases}. \quad (3.4)$$

Let $P(\hat{Q}, 0) := \hat{Q}$, that is, $P(\hat{Q}, 0) = B(c^0, r^0)$, where

$$c_i^0 = \frac{a_i + b_i}{2}, \quad r_i^0 = \frac{b_i - a_i}{2}, \quad i = 1, \dots, n. \quad (3.5)$$

Denote by $P(\hat{Q}, d)$, $d \in \mathbb{N}$, the set of cells obtained after $d \in \mathbb{N}$ subdivision steps (d is also called the insertion depth) starting with $B(c^0, r^0)$, where in each step $i = 1, \dots, d$ the cells are subdivided with respect to the j_i^{th} coordinate, where j_i is varied cyclically. That is, $j_i = ((i - 1) \bmod n) + 1$. Note that for every point $y \in Q$ and every subdivision step d there exists exactly one cell $B = b(y, d) \in P(Q, d)$ with center c and radius r such that $c_i - r_i \leq y_i < c_i + r_i$, $\forall i = 1, \dots, n$. Thus, every set of solutions S_B leads to a (unique) set of cell collections $\mathcal{B}_d(S_B) := \{b(y, d) \in P(\hat{Q}, d) : y \in S_B\}$.

Further, $\Delta(T) \subset P(\hat{Q}, d)$ denotes the set of all neighboring cells of a given cell $T \in P(\hat{Q}, d)$. The dominance relation is considered cell-wise as follows: a cell $b_j \in \mathcal{B}_l$ is dominated by a vector $b_i \in \mathcal{B}_l$ (in short $b_i \prec b_j$) if $\forall y \in b_j$ there exists a $x \in b_i$ such that $x \prec y$. Else b_j is non-dominated by b_i ($b_i \not\prec b_j$). Finally, points $x \in \hat{Q} \subset Q$ are discarded from the algorithm as they are not feasible.

Convergence toward the Pareto set can be guaranteed under certain assumptions on the given MOP as the following result shows.

Theorem 1 *Let an MOP of the form (2.1) be given and assume that there exists no weak Pareto point in $Q \setminus \mathcal{P}$. Further, assume that \mathcal{P} is connected and compact and that the initial cell collection \mathcal{B} contains a part of the Pareto set, i.e.,*

$$\mathcal{B} \cap \mathcal{P} \neq \emptyset. \quad (3.6)$$

Then, an application of Algorithm 3 leads to a sequence of cell collections \mathcal{B}_l such that

$$d_H(\mathcal{B}_l, \mathcal{P}) \rightarrow 0 \text{ for } l \rightarrow \infty. \quad (3.7)$$

Proof. Let d be the insertion depth of \mathcal{B} . First, we show that $\mathcal{P} \subset \mathcal{B}_0$. For this, let $\rho \in \mathcal{P}$. By assumption (3.6) there exist a point $x \in \mathcal{B} \cap \mathcal{P}$ and by connectivity of \mathcal{P} there exists a path from x to ρ , i.e., a curve $c : [0, 1] \rightarrow \mathcal{P}$ with $c(0) = x$ and $c(1) = \rho$. By construction of the recover algorithm (lines 3 to 11) every cell $B(c(t), d)$, $t \in [0, 1]$, will be added to the collection \mathcal{B}_0 . Thus, in particular ρ will be added to \mathcal{B}_0 by which the claim follows. Further, the recover algorithm stops with finitely many cells as \mathcal{P} is connected. The convergence toward \mathcal{P} is then guaranteed by the following subdivision algorithm: by construction of the algorithm

it follows directly that every cell $b(x, l)$ for every point $x \in \mathcal{P}$ and every step $l \in \mathbb{N}$ will be kept in the cell collection \mathcal{B}_l . Finally, we have to show that for every $x \notin \mathbb{P}$ there exists a depth $l_0 \in \mathbb{N}$ such that the cell $b(x, l_0)$ is not contained in \mathcal{B}_{l_0} . Let $x \in \hat{Q} \subset \mathcal{P}$. Since x is not weakly Pareto optimal there exists a point $\rho \in \mathcal{P}$ such that $F(\rho) <_p F(x)$. Further, by continuity of F there exists a neighborhood $\Delta(\rho)$ of ρ with

$$F(y) <_p F(x) \quad \forall y \in \Delta(\rho). \quad (3.8)$$

Finally there exists a $l_0 \in \mathbb{N}$ with $b(\rho, l_0) \subset \Delta(\rho)$. Thus, $b(x, l_0)$ will be discarded from the cell collection (if contained before), and the claim follows. ■

We stress that the above proof is done for connected Pareto sets, but that these sets do not have to be connected in general. The result, however, can easily be extended to all connected components of \mathcal{P} that have a non-zero intersection with \mathcal{B} . Thus, the entire Pareto set can be retrieved in case \mathcal{B} contains elements from all connected components of \mathcal{P} . As the EAs tend to quickly identify promising regions and as we start with relatively large initial cells (respectively low insertion depths d) the chance is quite high that the EA has detected all connected components of the given MOP, and that the set oriented methods in EA+SCM can reliably compute suitable finite size approximations of the entire set of interest.

3.3 Case Studies

Four problems are presented here in order to gain a better understanding of how the EA+SCM hybrid method performs in comparison to the GA and SCM. The first three are mathematical benchmark problems from the book [26]. The complexity level of the problems increases in the presented order. The first three problems are low dimensional problems, for which some EAs have trouble finding all the branches of the solution. The last problem deals with multi-objective optimal design of controls for nonlinear dynamical systems, and intends to demonstrate the effectiveness of the proposed method in engineering applications. The Pareto set and Pareto front of engineering problems can have far more complex structures than the artificially designed benchmark problems [12].

For the evaluation of the performance of the different methods, we keep the total number of function evaluations approximately the same for all the methods. This provides a fair basis for comparison. For the GA, the total number of function calls is equal to the product of the number of generations and population size. It should be pointed out that finding the best combination of the number of generations versus population size is an optimization problem itself and is not the purpose of the current study because of the random nature of the GA, we repeat the GA computing 30 times to obtain the statistical average of the performance metric. Further, we used the Wilcoxon ranking sum to test the null hypothesis with a 95% significance level. Whenever there was a failure to reject the null hypothesis, it will be stated

explicitly. In addition, we present the averaged convergence curves for studying the performance of each method during the solution process by calculating the Hausdorff distance between the current solution and the reference solution. These curves can qualitatively express the effectiveness and reliability of each method.

3.3.1 Kursawe's Problem

The problem statement is as follows:

$$\min(f_1(x), f_2(x)), \quad (3.9)$$

where

$$\begin{aligned} f_1(x) &= \sum_{i=1}^{n-1} (-10e^{-0.2\sqrt{x_i^2+x_{i+1}^2}}), \\ f_2(x) &= \sum_{i=1}^n (|x_i|^a + 5 \sin(x_i)^b), \end{aligned} \quad (3.10)$$

$$\begin{aligned} -5 &\leq x_i \leq 5, \quad i = 1, 2, 3 \\ a &= 0.8; \quad b = 3. \end{aligned}$$

This test problem was built upon Ackley's single objective optimization problem [47] and a modification provided by [48]. The problem is scalable as the number n of variables can be changed. The solution of this problem has disconnected branches and is asymmetric in the design and objective spaces.

Figures 3.1 and 3.2 show the Pareto set and Pareto front of Kursawe's problem. The EA+SCM hybrid method shows a better performance in both parameter and objective space in terms on the Hausdorff distance. Although in objective space there was a failure to reject the null hypothesis at the 5% significance level according to the Wilcoxon ranking sum test. A summary of the solution is presented in Table 3.1. The table shows that because two branches of the Pareto set in Figure 3.2 (the one at $x_3 \simeq 0$ plane and $x_1 \simeq 0$ plane) are mapped to the left section of the Pareto front in 3.1, the performance metric in terms of the Pareto front is not impacted significantly by missing a branch of the Pareto set.

The convergence of the methods are compared in Figures 3.3 and 3.4. The shaded area shows the range for which the solution is varying when the method is being repeated and the lines show the average. It should be noticed that the increase in Hausdorff distance for the hybrid method at around 1000 function evaluations is because at that point the initial evolutionary algorithm switches to the cell mapping method and the solution points are replaced with the center of the closest cell in discretized space.

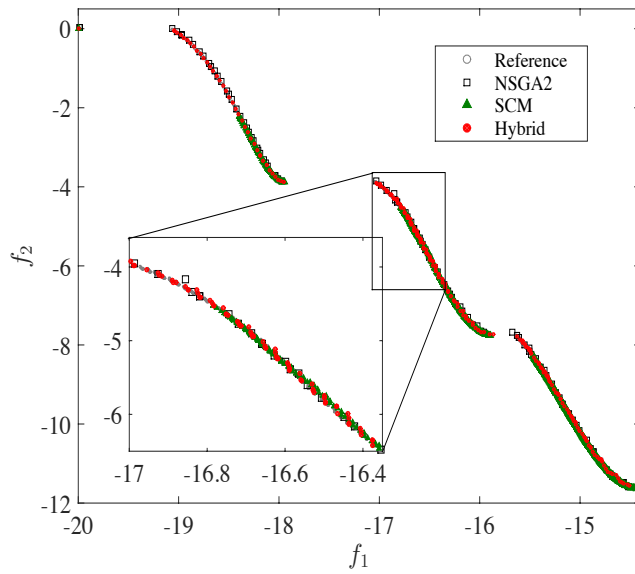


Figure 3.1: The Pareto front of Kursawe's problem.

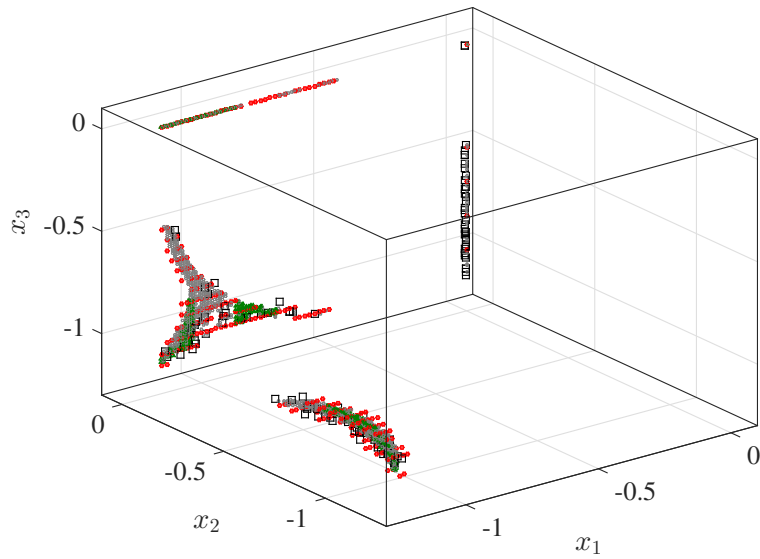


Figure 3.2: The Pareto set of Kursawe's problem. The legend is the same as Figure 3.1.

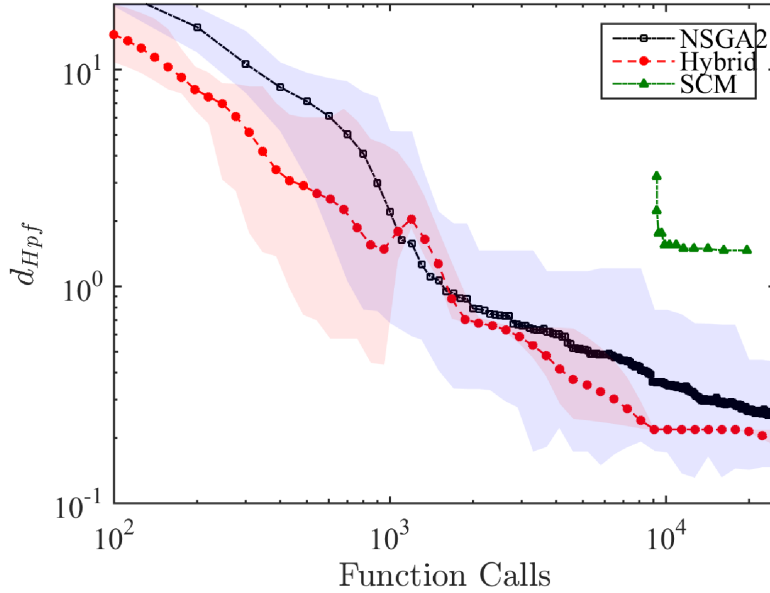


Figure 3.3: Convergence of Pareto front in Kursawe's problem.

3.3.2 Deb's Problem

The problem statement is as follows:

$$\min(f_1(x), f_2(x)), \quad (3.11)$$

where

$$\begin{aligned} f_1(x) &= x_1, \\ f_2(x) &= (1 + 10x_2) \left[1 - \left(\frac{x_1}{1+10x_2} \right)^\alpha - \frac{x_1}{1+10x_2} \sin(2\pi qx_1) \right], \end{aligned} \quad (3.12)$$

$$x = [x_1, x_2]^T, \quad 0 \leq x_1, x_2 \leq 1, \quad q = 6, \quad \alpha = 2.$$

This problem is constructed to test Deb's methodology. The true Pareto set and Pareto front consist of disconnected branches. The number of branches can be controlled by the parameter q . The Pareto set and Pareto front for this problem are presented in Figures 3.5 and 3.6. Table 3.2 shows the solution report for this problem. The zoom-in view in Figure 3.6 shows the distribution of the solution points obtained by different methods. The segments of the true solution lie on the line $x_2 = 0$ (the gray line). The values of d_{HPs} and d_{HPf} from Table 3.2 correlates well with the observation from Figure 3.6. The convergence of the methods are compared

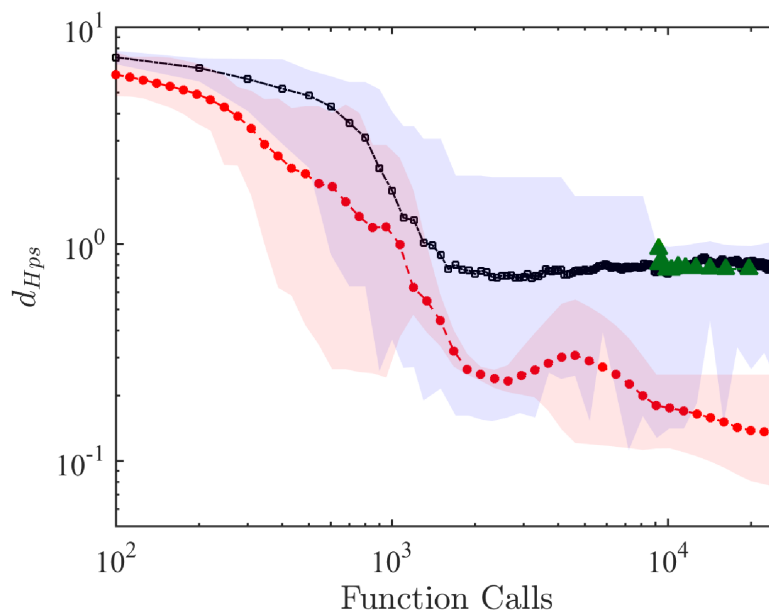


Figure 3.4: Convergence of Pareto set in Kursawe's problem. The legend is the same as Figure 3.3.

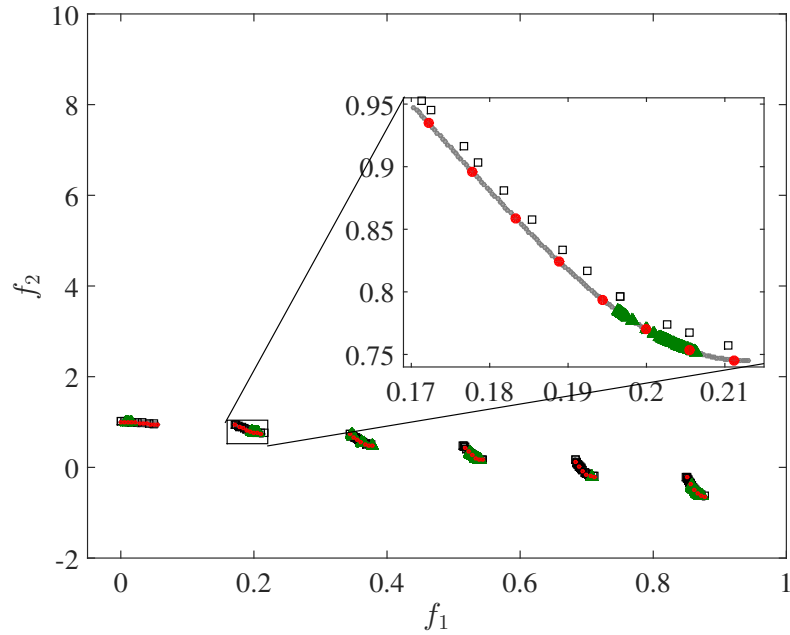


Figure 3.5: The Pareto front of Deb's problem. The legend is the same as Figure 3.1.

in Figures 3.7 and 3.8. Notice that the solution quality of the SCM method does not change significantly with number of function evaluations. The reason is that a big portion of the solution is missed from the course initial cell size and they cannot be recovered.

As a programming note, regardless of the method being used for solving the MOP, we need to be able to capture the solution on the boundaries. Unlike most of the engineering applications, the artificially designed MOPs often have solutions located on the boundaries of the design space, which presents challenges to the numerical solutions.

3.3.3 Viennet's Second Problem

The problem statement is as follows:

$$\min(f_1(x), f_2(x), f_3(x)), \quad (3.13)$$

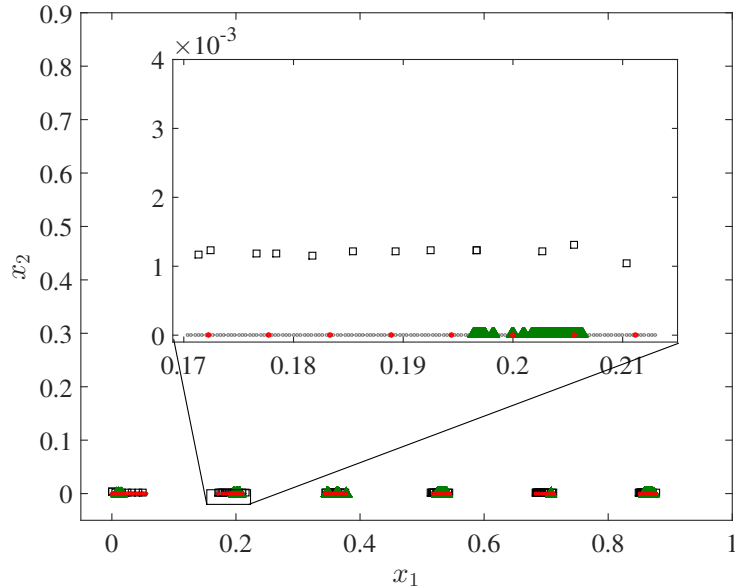


Figure 3.6: The Pareto set of Deb's problem. The legend is the same as Figure 3.1.

where

$$\begin{aligned}
 f_1(x) &= \frac{(x_1 - 2)^2}{2} + \frac{(x_2 + 1)^2}{13} + 3, \\
 f_2(x) &= \frac{(x_1 + x_2 - 3)^2}{36} + \frac{(-x_1 + x_2 + 2)^2}{8} - 17, \\
 f_3(x) &= \frac{(x_1 + 2x_2 - 1)^2}{175} + \frac{(2x_2 - x_1)^2}{17} - 13,
 \end{aligned} \tag{3.14}$$

$$x = [x_1, x_2]^T, \quad -400 \leq x_1, x_2 \leq 400.$$

The solution of this problem consists of a connected asymmetric two dimensional surface in the design space and a three dimensional thin manifold in the objective space, as shown in Figures 3.9 and 3.10. The grey area in Figure 3.10 shows the reference solution. The region in the design space that is searched by the SCM recovery algorithm is considerably bigger than the area where the true solution lies. As a result, the SCM method is less efficient than the GA. The EA+SCM can gain computational efficiency when the recovery search does not go far behind the vicinity of the solution. For this problem, the SCM method alone found only a piece of the solution after refinement, while GA returned points that are more condensed on one edge of the solution. However, the EA+SCM hybrid method successfully

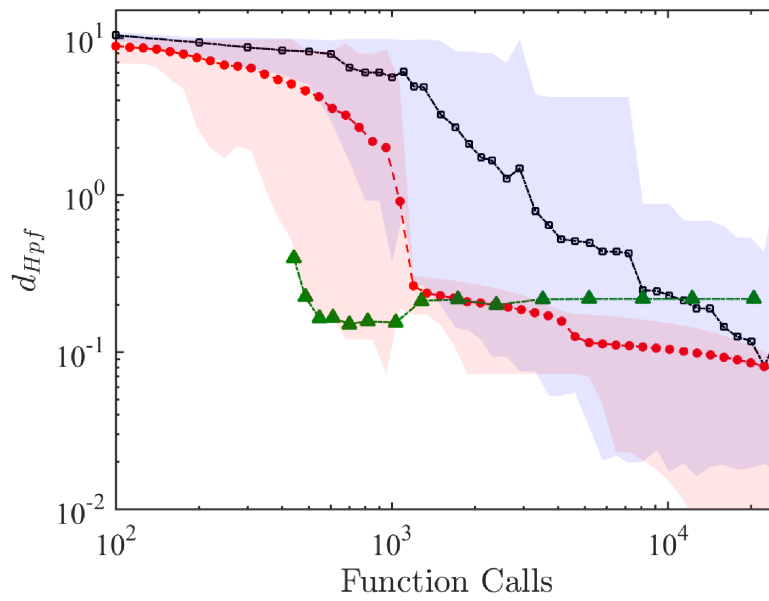


Figure 3.7: Convergence of Pareto front in Deb's problem. Legend is the same as Figure 3.3.

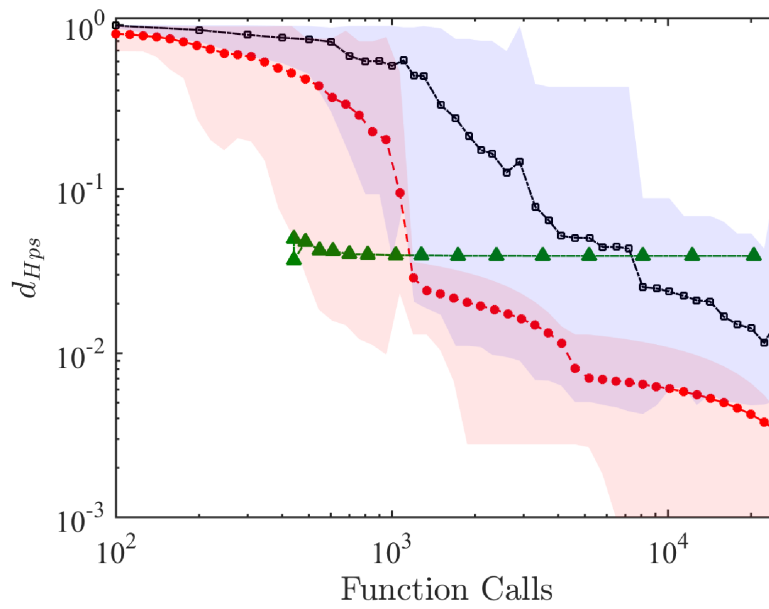


Figure 3.8: Convergence of Pareto set in Deb's problem. The legend is the same as Figure 3.3.

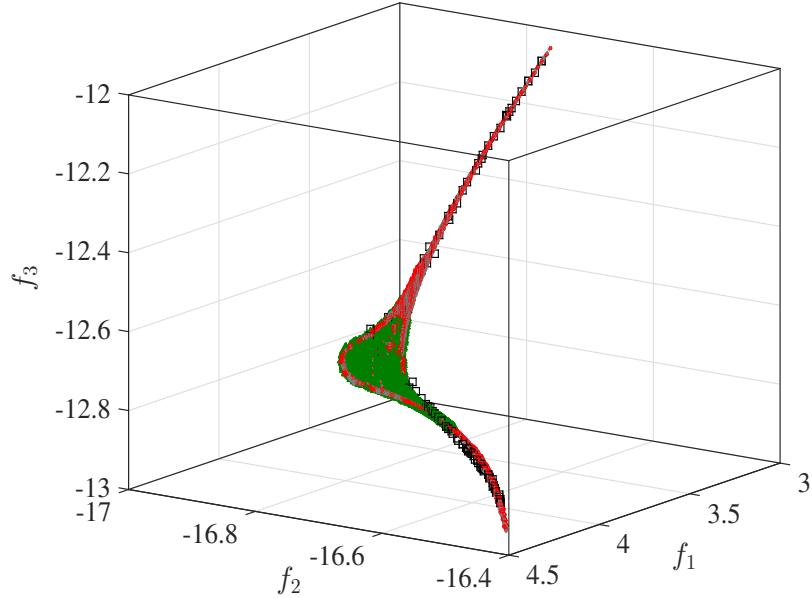


Figure 3.9: The Pareto front of Viennet’s second problem. The legend is the same as Figure 3.1.

found an evenly distributed solution matching well with the reference. The values of d_{Hps} and d_{Hpf} listed in Table 3.3 also confirm this observation.

Figures 3.11 and 3.12 represent the convergence of the methods in Viennet’s problem. It should be pointed out that NSGA2 almost stops improving after a certain point (around 2000 function calls in this case) while the hybrid method can be continuously improving if the initial points happen to be close to each branch of the solution.

3.3.4 PID Control for Duffing-Van der Pol Nonlinear Oscillator with Time Delay

Next, we consider an engineering problem of optimal proportional-integral-derivative (PID) control design with time delay. Time delay is a common phenomenon in control systems due to signal transmission delay. The time delay may deteriorate the control performance and stability. For linear time-delayed systems, the transfer function in the frequency domain leads to a transcendental equation that helps to determine the stability region for control gains. Unfortunately, the frequency domain analysis cannot be applied to nonlinear systems. In this example, we present an approach to design multi-objective optimal PID controls for nonlinear systems with time delay.

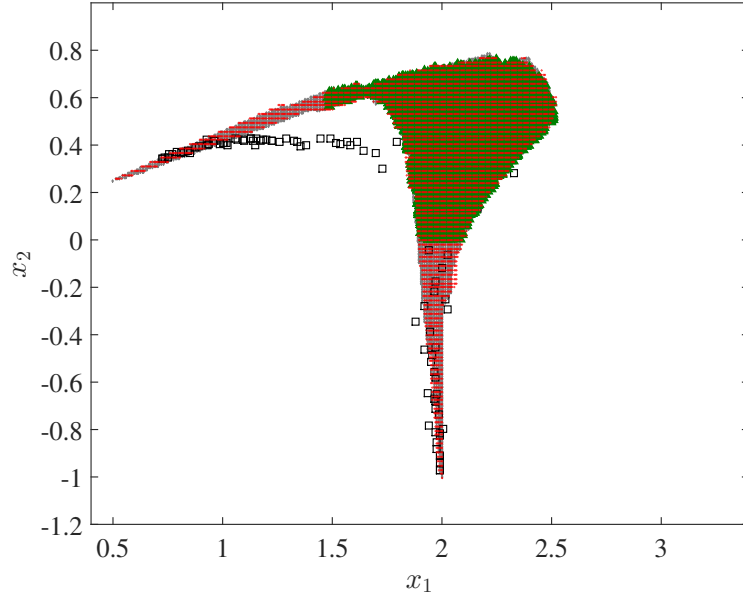


Figure 3.10: The Pareto set of Viennet's second problem. The legend is the same as Figure 3.1.

Consider a second order nonlinear dynamical system with a delayed feedback control. The primary control objective is to direct the system output to track a given signal while optimizing several conflicting performance indices.

$$\begin{aligned}\dot{x}_1 &= x_2, \\ \dot{x}_2 &= f(x_1, x_2) + u(t - \tau_c),\end{aligned}\tag{3.15}$$

where f is a nonlinear function of its arguments, u is the control and τ_c is the control delay. In this study, f is chosen to be a Duffing-Van der Pol oscillator such that

$$f(x_1, x_2) = -ax_1 - bx_1^3 + x_2(\zeta - \gamma x_1^2),\tag{3.16}$$

where $a = -1$, $b = 0.25$, $\gamma = 4$, $\zeta = 4$ and the control delay is chosen to be $\tau_c = 0.05$ seconds. The equilibria of the uncontrolled system are at $x_1 = 0, \pm 2$ and $x_2 = 0$. The Van der Pol term $x_2(\zeta - \gamma x_1^2)$ contributes to a limit cycle, which indicates the oscillatory nature of the uncontrolled system response.

Assume that u is a PID feedback control,

$$u(t) = k_p [r(t) - x_1(t)] + k_i \int_0^t [r(\hat{t}) - x_1(\hat{t})] d\hat{t} - k_d x_2(t),\tag{3.17}$$

where $r(t)$ is a reference input, k_p , k_i and k_d are the PID control gains to be optimized. We introduce a third state variable x_3 such that $\dot{x}_3(t) = r(t) - x_1(t)$.

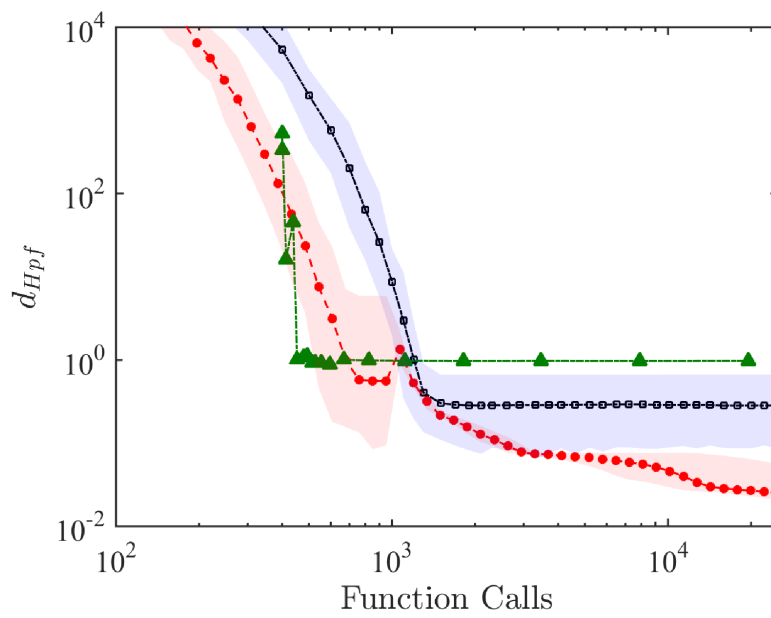


Figure 3.11: Convergence of Pareto front of Viennet's problem. Legend is the same as Figure 3.3.

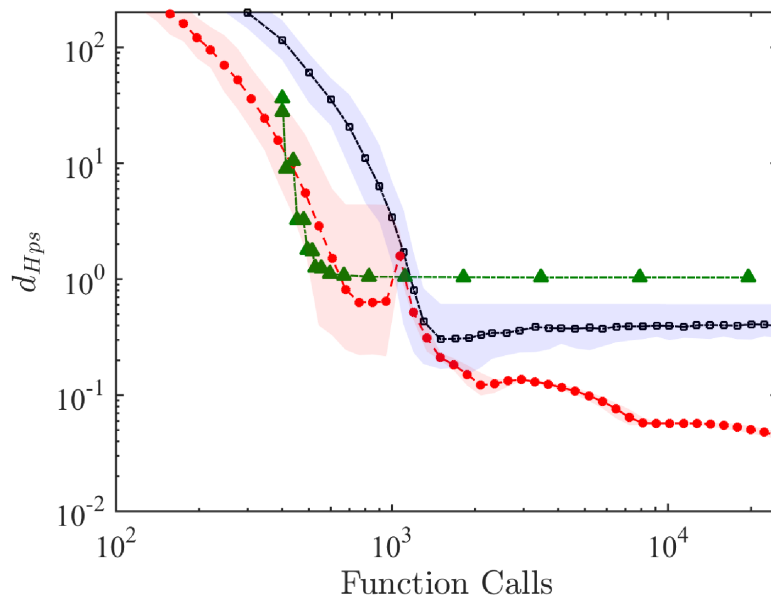


Figure 3.12: Convergence of Pareto set of Viennet's problem. Legend is the same as Figure 3.3.

Equation (3.17) is extended into a three dimensional state space. The extended system now reads,

$$\begin{aligned}\dot{x}_1 &= x_2, \\ \dot{x}_2 &= f(x_1, x_2) + u(t - \tau_c), \\ \dot{x}_3 &= r(t) - x_1(t),\end{aligned}\tag{3.18}$$

where

$$u(t) = k_p[r(t) - x_1(t)] + k_i x_3 - k_d x_2(t).\tag{3.19}$$

For tracking control design, we usually set $r(t)$ as a step function. To perform closed-loop stability analysis, we first find the equilibrium. The closed-loop system has only one equilibrium or the steady state response,

$$x_1^* = 1, \quad x_2^* = 0, \quad x_3^* = -\frac{1}{k_i} f(1, 0).\tag{3.20}$$

Recall that the uncontrolled nonlinear system has multiple equilibrium solutions. The stability analysis of the closed-loop system can be performed near the steady state. Let $z = [z_1, z_2, z_3]^T$ be the perturbation of the system away from the steady state $x^* = [x_1^*, x_2^*, x_3^*]^T$. We then have,

$$\dot{z}(t) = Az(t) + A_c z(t - \tau_c),\tag{3.21}$$

where A and A_c are matrices of the linearized system and are the functions of the control gains. The stability of the linearized system can be studied by using the continuous time approximation (CTA) method [49, 50]. The CTA method is an extension of the method of semi-discretization for stability analysis of time-delayed linear systems by [51]. The CTA method makes use of the finite difference schemes to approximate the effect of time delay and convert the time delayed system to a high dimensional delay-free system. In this study, we apply the CTA method to examine the stability of various multi-objective optimal control designs.

The design space of the control gains x is chosen as follows,

$$Q = \{x \in [60, 100] \times [10, 40] \times [10, 30] \subset \mathbb{R}^3\}.\tag{3.22}$$

The objectives of multi-objective optimal control design include the response peak time t_p , overshoot M_p , integral of the absolute error e_{IAE} , and λ_{ss} defined as,

$$\lambda_{ss} = \max[\text{Re}(\lambda_i)],\tag{3.23}$$

where λ_i are the eigenvalues of Equation (3.21). We further impose the constraints on the objective functions as,

$$[t_p, M_p, e_{IAE}, \lambda_{ss}] \leq [2.5, 6\%, 0.75, -0.25].\tag{3.24}$$

Note that the constraint $\lambda_{ss} \leq -0.25$ provides a necessary stability robustness of the closed-loop system.

We apply the EA+SCM hybrid method and initially partition the design space with $21 \times 21 \times 21$ cells. A subdivision of $3 \times 3 \times 3$ cells is applied to the set of cells representing the Pareto set. A total of 1095 cells are found to represent the Pareto set. An average of 11745 function calls are made in the computation. Table 3.4 shows the detail comparison of GA and the EA+SCM hybrid. In this case, the reference solution was created by running the two methods independently with 8×10^4 function calls and then combining the two sets and performing a Pareto dominance check on it.

Figure 3.13 shows two projections of the Pareto front where the color code is based on the values of e_{IAE} and M_p in the upper plot and lower plot, respectively. It can be observed that the Pareto front consists of three disjoint branches. The corresponding Pareto set is presented in Figure 3.14. In the 2D projections, the color code is based on the absent gain value. In the 3D plot, the color code is based on the value of t_p . The upper left figure shows that there is a correlation between the proportional and derivative gains of the control in the optimal solution. The step response and control of the system in time domain for minimum values of each objective are plotted in Figure 3.15. Note that the plot presents only 4 examples out of 1095 Pareto optimal designs. The designer has freedom to choose the combination of design parameters which best suit the application.

The convergence of the methods are compared in Figures 3.16 and 3.17. Although both methods were performing similarly in the beginning of the solution process, the hybrid method clearly outperforms NSGA2 ultimately.

3.4 Notes on Parallel Computing

Nowadays, parallel computing and High Performance Computing (HPC) has become a requirement for the areas of reaseach that require extensive amount of computation. On the other hand, the availability of parallel computing resources encourages us to take advantage of them. Since optimization techniques aim to find the most suitable and optimal answers for the such problems, we are naturally motivated to use efficient and fast algorithms to find the solutions. This need is more noticable in engineering MOPs since they usually involve a simulation for each function evaluation. Parallel computing enables us to run multiple processes simultaneously. The amount of time that could be saved by parallel computing depends on the structure of the algorithm as well as its method of implementation and memory management. The function evaluation is usually the most time consuming part of an engineering MOP and thus the parallelism of that part could result in a significant reduction of runtime.

One of the advantages of using evolutionary algorithms is their potential to be parallel. Each evolution cycle is consist of evaluating the objectives for each

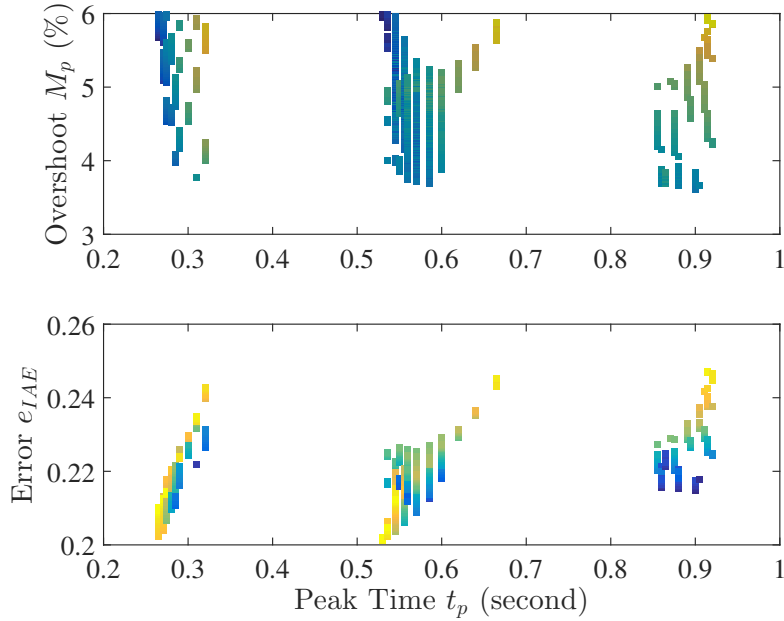


Figure 3.13: The Pareto front of the time delayed Duffing-Van der Pol system.

individual in the population. Considering that function evaluation for each individual is independent of the rest of the population, each evaluation can run on a separate processor. For the SCM method, however, the situation is different. For the classical SCM that requires us to sweep the design space and evaluate all cells, parallelism is as simple as EAs. Nonetheless, the modified SCM method used in the hybrid algorithm is more challenging since the list of the cells to be evaluated is dynamically updated in each subdivision. The method that was implemented in hybrid SCM was to keep a list of cells to be evaluated and using parallel evaluation multiple times in each subdivision until there was no new cell added to the list which means the end of subdivision is reached.

Parallel processing can be implemented using CPU, GPU or a combination of both (there exist other methods such as FPGAs and massively parallel processor which are out of the scope of this thesis). There are many details to be considered for choosing the suitable hardware and there exist many studies trying to compare them from various aspects [52–54]. Some of the CPU/GPU differences that was found influential in our programming is listed below

- GPUs provide more cores
- CPU cores provide higher clock speeds

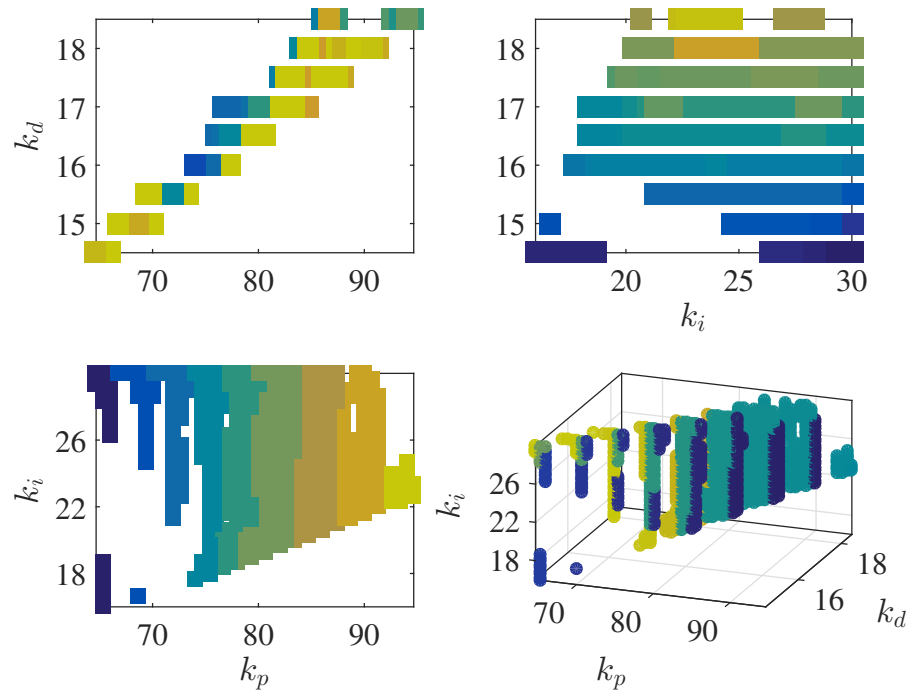


Figure 3.14: The Pareto set of the PID control gains for the Duffing-Van der Pol system.

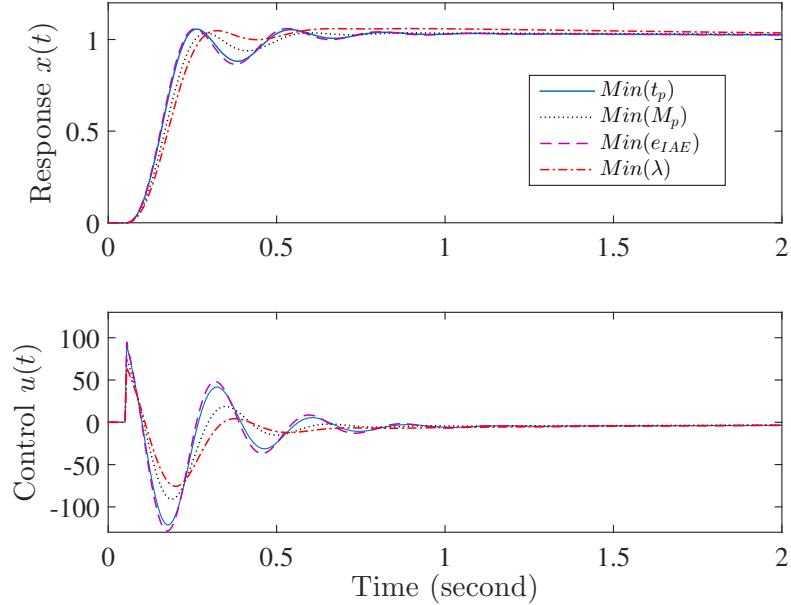


Figure 3.15: The closed-loop step responses (top) and control (bottom) of the Duffing-Van der Pol system.

- memory is expensive and limited in GPUs
- loading/unloading data from CPU to GPU could be time consuming

The underlying memory architecture of parallel computing can be classified into shared memory, distributed memory or shared distributed memory. In shared memory, the processors communicate through manipulating shared memory variables. In distributed memory architecture, however, the units communicate through message passing. In CPUs, OpenMP is probably the most used shared memory Application Program Interface (API) while the Message Passing Interface (MPI) is the most popular message passing API. In this thesis OpenMP is the only CPU parallelism method used.

For GPUs, the most popular programming languages and frameworks are OpenCL, NVIDIA Cuda, OpenACC, OpenHMPP. Cuda extends the C/C++ programming language to run parallel on Geforce 8 series GPUs. It is also the only GPU parallel processing framework utilized in this thesis where applicable.

Whatever methods of parallelism one might choose, it is important to have the performance of the program tuned to make sure the hardware resources are being used effectively. Performance tuning needs experience and tools. Profiling the program execution is a great way to monitor the performance and find out the possible

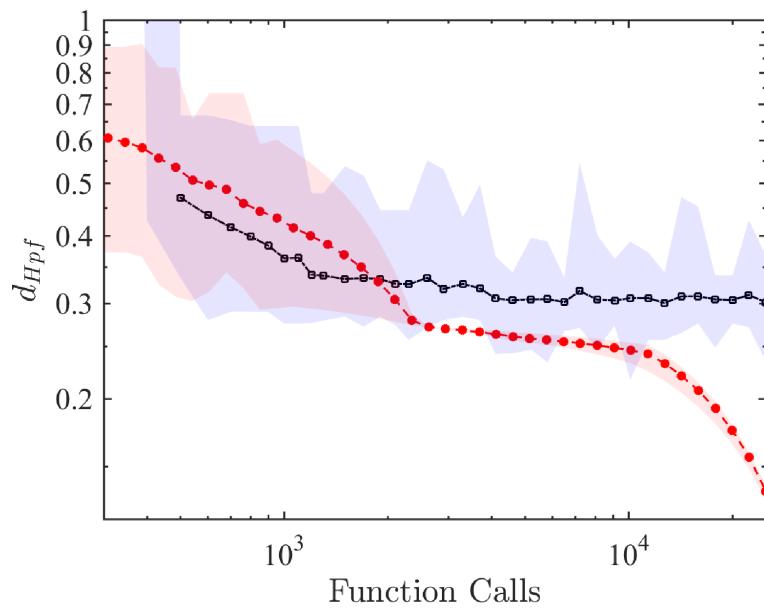


Figure 3.16: Convergence of Pareto front in PID control problem. Legend is the same as Figure 3.3.

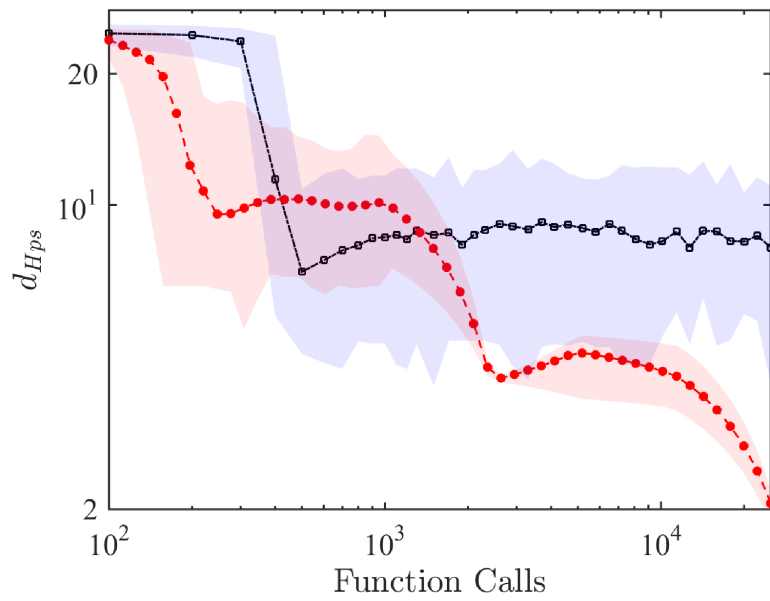


Figure 3.17: Convergence of Pareto set in PID control problem. Legend is the same as Figure 3.3.

bottle necks. Some of the tools used for monitoring the performance of the programs and their main features are,

- MUST - MPI usage correctness checking
- PAPI - Interfacing to hardware performance counters
- Periscope - Automatic analysis via an on-line distributed search
- Scalasca -Large-scale parallel performance analysis
- TAU - Integrated parallel performance system
- Vampir - Interactive graphical trace visualization & analysis
- Score-P - Community instrumentation & measurement infrastructure
- DDT/MAP/PR: Parallel debugging & profiling
- Kcachegrind - Callgraph-based cache analysis [x86 only]

As a final note, regardless of the hardware and framework used for the purpose of parallel computing, it is important to make sure that the hardware resources are utilized effectively. Improper implementation of parallel codes could even result in a performance poorer than the serial run of the program.

3.5 Summary

The EA+SCM hybrid method for MOPs has been presented. The results of several benchmark problems and a nonlinear control design obtained by the GA, SCM and EA+SCM hybrid method have been discussed in detail. Whenever applicable, three methods of GA, SCM and EA+SCM are compared quantitatively. The studies show that the EA+SCM hybrid method is not only more accurate, but also more reliable in terms of its ability to solve for complex MOPs with either continuous or disjointed Pareto sets than the GA or SCM applied separately. Although the EA+SCM hybrid method does not guarantee the discovery of the global solution, the recovery technique together with relatively large population size of the EA solution makes it far more likely to find the global solution when compared to other stochastic approaches. Because the EA+SCM hybrid method is so promising, further studies of its convergence and stability are warranted in the future.

As the EAs tend to quickly identify promising regions and as we start with relatively large initial cells (respectively low insertion depths d) the chance is quite high that the EA has detected all connected components of the given MOP, and that the set oriented methods in EA+SCM can reliably compute suitable finite size approximations of the entire set of interest.

Table 3.1: Comparison of the EA-SCM hybrid method with GA and SCM when applied to Kursawe’s problem.

		GA	Hybrid	SCM
Iterations		30	30	1
Initial division		N/A	$21 \times 21 \times 21$	$21 \times 21 \times 21$
Final resolution		N/A	$181 \times 181 \times 181$	$1701 \times 1701 \times 567$
GA generation		250	20	N/A
GA population		100	50	N/A
CPU time (second)	min	227	42.61	
	max	241	61.91	18.98
	ave	229	45.29	
	std	3.88	4.75	
Function evaluation	min		18663	
	max	25000	19063	19656
	ave		18826	
	std	0	104.8	
d_{Hps}	min	0.2277	0.0821	
	max	1.0336	0.2500	0.7666
	ave	0.8147	0.1388	
	std	0.2055	0.0757	
d_{Hpf}	min	0.1466	0.2181	
	max	0.4501	0.2181	1.4640
	ave	0.2504	0.2181	
	std	0.0738	$1.8e - 13$	

Table 3.2: Comparison of the EA-SCM hybrid method with GA and SCM when applied to Deb's problem.

		GA	Hybrid	SCM
Iterations		30	30	1
Initial division		N/A	21×21	21×21
Final resolution		N/A	181×181	137781×45927
GA generation		250	20	N/A
GA population		100	50	N/A
CPU time (second)	min	225	18.94	
	max	363	48.30	28.1
	ave	300	22.42	
	std	49.3	6.06	
Function evaluation	min		1509	
	max	25000	12638	20499
	ave		4417	
	std	0	1904	
d_{Hps}	min	$5.188e - 3$	$3.086e - 4$	
	max	0.133	0.013	0.0391
	ave	0.0161	$7.128e - 3$	
	std	0.0242	$5.244e - 3$	
d_{Hpf}	min	0.0171	$7.927e - 3$	
	max	1.3303	0.1740	0.2184
	ave	0.1256	0.1145	
	std	0.2553	0.0542	

Table 3.3: Comparison of the EA-SCM hybrid method with GA and SCM when applied to Viennet’s second problem.

		GA	Hybrid	SCM
Iterations		30	30	1
Initial division		N/A	21×21	21×21
Final resolution		N/A	131221×43741	413343×137781
GA generation		250	20	N/A
GA population		100	50	N/A
CPU time (second)	min	228	63.3	
	max	306	70.6	453
	ave	250	66.6	
	std	20.3	2.25	
Function evaluation	min		10875	
	max	25000	16865	19552
	ave		13660	
	std	0	1303	
d_{Hps}	min	0.3131	0.0573	
	max	0.6092	0.0589	1.0359
	ave	0.3965	0.0574	
	std	0.0680	$2.913e - 4$	
d_{Hpf}	min	0.0868	0.0274	
	max	0.6666	0.0758	0.9751
	ave	0.2818	0.0290	
	std	0.1618	$8.829e - 3$	

Table 3.4: Comparison of the EA-SCM hybrid method with GA when applied to time delayed Duffing-Van der Pol system.

		GA	Hybrid
Iterations		30	30
Initial division		N/A	$21 \times 21 \times 21$
Final resolution		N/A	$61 \times 61 \times 21$
GA generation		250	20
GA population		100	50
CPU time (second)	min	1429	1471
	max	1647	1874
	ave	1461	1587
	std	41.9	107
Function evaluation	min		11075
	max	25000	12495
	ave		11745
	std	0	557.7
d_{Hps}	min	3.866	3.399
	max	11.08	4.740
	ave	7.981	3.997
	std	1.526	0.621
d_{Hpf}	min	0.235	0.223
	max	0.367	0.251
	ave	0.301	0.241
	std	0.026	0.012

Chapter 4

SCM METHOD AND FEEDBACK CONTROL DESIGN

4.1 Introduction

Feedback controls are most popular in industry [55]. A great deal of effort has been made in designing optimal feedback control gains for various applications [56]. In the time domain, for example, the overshoot, rise time or peak time, settling time and the tracking error are often used to characterize the performance of the closed loop system. It is well-known that the overshoot and peak time are conflicting objectives, meaning that when the overshoot goes down, the peak time goes up, and vice versa. It is thus quite natural to consider the multi-objective feedback control design to minimize the overshoot, peak time, and tracking error at the same time. The multi-objective optimization algorithm can find a set of optimal solutions that represent the best compromises among these conflicting goals.

There have been many multi-objective optimal control studies in the literature [57]. Rani *et al.* [58] developed the global ranking genetic algorithm to design a proportional-integral-derivative (PID) controller for a rotary inverted pendulum. Three conflicting functions are considered in the design including the settling time, overshoot and mean square steady-state error. Another multi-objective genetic algorithm called the genetic artificial immune system algorithm was investigated by Khoei *et al.* [59]. The four fitness functions are considered: the rise time, overshoot, settling time, and integral square error. A non-dominated sorting genetic algorithm was used by Kumar and Nair [60] to tune the control gains to minimize the rise time, overshoot and settling time. Similar multi-objective optimal control designs are studied with different algorithms for multi-objective optimization problems (MOPs) including the ant colony algorithm by Ibtissem *et al.* [61], an immune algorithm by Kim [56] and the imperialist competitive algorithm by Esmail and Caro [62]. Other designs are in the frequency domain where the gain and phase margins are used to characterize the system stability and robustness, and the crossover frequency is used to assess the system response speed [63]. Liu and Daley [64] have proposed an optimal PID control for a rotary hydraulic system. The control parameters are tuned such that the crossover frequency, gain margin, phase margin, and steady state error are within the targeted range.

The solution of MOPs does not consist of a single point in the design space, but rather forms a set, called *Pareto set* named after an Italian economist Vilfredo

Pareto (1848-1923) [65]. The corresponding objective function values are called *Pareto front*. Many numerical methods for MOPs have been studied. There exist, for instance, scalarization methods that transform the MOP into a scalar optimization problem (SOP). By choosing a clever sequence of SOPs, a finite size approximation of the entire Pareto set can be obtained in certain cases [19, 23, 66, 67]. See the book by Eichfelder [68] for an overview. These methods are advantageous in particular for uni-modal (e.g., convex) objectives due to their fast convergence when classical mathematical programming techniques are used to solve the SOPs. However, they may run into trouble for multi-modal objectives due to the locality of the approach. It is possible that these methods get stuck in local optimal solutions that are not globally optimal. Since the Pareto set forms under some mild regularity conditions locally a manifold, the continuation methods which perform searches along the Pareto set are very efficient if one solution is at hand and if the Pareto set is connected [20, 21, 36, 69].

Evolutionary algorithms are most widely used for MOPs [1]. The underlying idea in evolutionary computation is to steer (or evolve) an entire set of solutions (population) toward the set of interest during the search. Evolutionary algorithms specialized for multi-objective problems have been shown their strength in many applications. Due to the global and stochastic nature of evolutionary algorithms, the Pareto set can be approximated quite well in most cases, although there is always an uncertainty left regarding whether the global Pareto set has indeed been found. There also exist attempts to compute nearly optimal solutions with evolutionary algorithms [70]. However, this approach suffers the drawback that only a subset of the nearly optimal solutions can be stored. In particular, a significant fraction of the set of interest may be ignored by the archiving strategy in certain cases [70].

Another approach to approximate the Pareto set is to use the set oriented methods with subdivision techniques [35, 71, 72]. The advantage of the set oriented methods is that they generate an approximation of the global Pareto set in one single run of the algorithm. Further, they are applicable to a wide range of optimization problems and are characterized by a great robustness. Hence, these methods are interesting alternatives against ‘classical’ mathematical programming techniques in particular for the thorough investigation of low or moderate dimensional MOPs. The cell mapping method in this study is the predecessor of the set oriented methods, and was proposed by [25] for global analysis of nonlinear dynamical systems. In the cell mapping method for MOPs, the dynamical systems are derived from multi-objective optimization search algorithms. The cell mapping method can obtain a finite size approximation \mathcal{A} of the Pareto set \mathcal{P} such that the distance between \mathcal{P} and \mathcal{A} is less or equal to a given threshold value in the Hausdorff sense. That is, the approximation quality of \mathcal{A} can in principle be determined. A first implementation of the multi-objective cell mapping technique is used that already yields promising results.

Two cell mapping methods have been extensively studied, namely, the simple cell mapping (SCM) and the generalized cell mapping (GCM) to study the global dynamics of nonlinear systems [24,25]. The cell mapping methods have been applied to optimal control problems of deterministic and stochastic dynamic systems [10, 73,74]. Other interesting applications of the cell mapping methods include optimal space craft momentum unloading by [75], single and multiple manipulators of robots by [76], optimum trajectory planning in robotic systems by Wang [77], and tracking control of the read-write head of computer hard disks by Yen [78]. Sun and his group studied the fixed final state optimal control problems with the simple cell mapping method [79, 80], and applied the cell mapping methods to the optimal control of deterministic systems described by Bellman’s principle of optimality [81]. Crespo and Sun further applied the generalized cell mapping based on the short-time Gaussian approximation to stochastic optimal control problems [74,82]. They also studied semi-active optimal control of populations of competing species in a closed environment with the cell mapping method [83].

In this chapter, we consider two time domain design problems of feedback controls for linear systems with or without time delay to demonstrate the SCM method for MOPs. In Section 4.2, we present the linear control systems. In Section 4.4, we describe the SCM method applied to the MOP. Then, the MOP of the feedback control design is presented in Section 4.5 with the goal to simultaneously minimize the peak time, overshoot and integrated absolute tracking error. In Section 4.6, we present numerical examples of the two linear control systems, one of which has a control time delay, and consider the design in the feedback gain space and with the help of \mathbf{Q} and \mathbf{R} matrices in the LQR optimal control formulation.

4.2 Linear Control System

Consider a general linear control system

$$\begin{aligned}\dot{\mathbf{x}}(t) &= \mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{u}(t), \\ \mathbf{y}(t) &= \mathbf{C}\mathbf{x}(t)\end{aligned}\tag{4.1}$$

where $\mathbf{x} \in \mathbf{R}^n$, $\mathbf{u} \in \mathbf{R}^m$ and $\mathbf{y} \in \mathbf{R}^p$. Consider a full state feedback control $\mathbf{u} = -\mathbf{K}\mathbf{x}(t)$. The control gain \mathbf{K} can be designed in a number of ways. When (\mathbf{A}, \mathbf{B}) is a controllable pair, we can design the control by the method of pole placement. For single-input-single-output (SISO) systems, the control can also be designed in the time domain by considering the time domain specifications or in the frequency domain by considering the stability margins [84]. One of the popular optimal control design methods is the linear quadratic regulator (LQR) for both SISO and multi-input-multi-output (MIMO) systems. The control is found to minimize a quadratic cost function,

$$J = \int_0^\infty (\mathbf{x}^T \mathbf{Q} \mathbf{x} + \mathbf{u}^T \mathbf{R} \mathbf{u}) dt\tag{4.2}$$

subject to the constraint of the state equation as well as initial and terminal conditions. The matrices \mathbf{Q} and \mathbf{R} determine the relative weights of the response and the control effort, and ultimately determine the control gain. The sub-optimal control gain is given by $\mathbf{K} = \mathbf{R}^{-1}\mathbf{B}^T\mathbf{P}$ where \mathbf{P} satisfies the algebraic Riccati equation [85]. Note that for tracking problems, the LQR control can be formulated in terms of the tracking error.

4.3 Time-Delayed Linear Control System

Consider now the system with a control time delay τ .

$$\begin{aligned}\dot{\mathbf{x}}(t) &= \mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{u}(t - \tau), \\ \mathbf{y}(t) &= \mathbf{C}\mathbf{x}(t).\end{aligned}\tag{4.3}$$

We define an extended state vector as

$$\mathbf{z}(t) = [\mathbf{x}(t), \mathbf{u}(t - \tau_N), \mathbf{u}(t - \tau_{N-1}), \dots, \mathbf{u}(t - \tau_1)]^T,\tag{4.4}$$

where $\mathbf{z} \in \mathbf{R}^{n+mN}$, $0 = \tau_0 < \tau_1 < \dots < \tau_{N-1} < \tau_N = \tau$, and N is the number of mesh grids to discretize the interval $[0, \tau]$. After introducing an interpolation scheme of $\mathbf{u}(t - \tau_i)$ over the mesh grid τ_i ($i \in [0, N]$), we can obtain an equation for $\mathbf{z}(t)$ without an explicit time delay of the control to replace Equation (4.3) as

$$\begin{aligned}\dot{\mathbf{z}}(t) &= \bar{\mathbf{A}}\mathbf{z}(t) + \bar{\mathbf{B}}\mathbf{u}(t), \\ \mathbf{y}(t) &= \bar{\mathbf{C}}\mathbf{z}(t).\end{aligned}\tag{4.5}$$

We should point out that Equation (4.5) is now in the same form as Equation (4.1). The control design can now be done in the extended state space \mathbf{R}^{n+mN} using the same methods as discussed above.

Note that when the system contains a retarded element as a function of the delayed response $\mathbf{x}(t - \tau_s)$, the extended state vector can be defined in the same way by including the delayed response $\mathbf{x}(t - \tau_j)$ where τ_j is the mesh grid on $[0, \tau_s]$, $j \in [0, N_s]$, and N_s is the number of mesh grids to discretize the interval $[0, \tau_s]$. This method is known as continuous time approximation. It can handle multiple independent time delays of the system and the control, and accurately predict a wide range of infinite poles of the time-delayed system. More details of the method can be found in [49, 86].

4.4 Simple Cell Mapping Method

The cell mapping methods describe system dynamics with cell-to-cell mappings by discretizing both the state space and time. It starts with a point-to-point mapping as a finite difference approximation of the governing differential equation of the system as

$$\mathbf{x}(k) = \mathbf{G}(\mathbf{x}(k - 1)),\tag{4.6}$$

where $\mathbf{x}(k) \in \mathbf{R}^n$ is the state vector at the k^{th} mapping step. Then, the SCM proposes to discretize the state space into a set of small cells, and represents the dynamics of an entire cell denoted as Z by the dynamics of its center. The center of Z is mapped according to the point-to-point mapping. The cell that contains the image point is called the image cell of Z . The cell-to-cell mapping is denoted by C ,

$$Z(k) = C(Z(k-1)). \quad (4.7)$$

We should note that the exact image of the center of Z is approximated by the center of its image cell according to the simple cell mapping. This approximation can cause significant errors in the long term solution of dynamical systems [10, 73, 77]. Nevertheless, the SCM offers an effective approach to investigate global response properties of the system.

The cell mapping with a finite number of cells in the computational domain will eventually lead to closed groups of cells of the period same as the number of cells in the group. The periodic cells represent invariant sets, which can be periodic motion and stable attractors of the system. The rest of the cells form the domains of attraction of the invariant sets. For more discussions on the cell mapping methods, their properties and computational algorithms, the reader is referred to the book by [25].

To apply cell mapping techniques to compute MOP solutions, we need to define dynamical systems derived from the search algorithms. As will be shown below, the invariant set of the dynamical system represents the Pareto set of the MOP.

Assume that the design parameter \mathbf{k} is updated by a map Φ that generates descent directions of all the objective functions f_i at a given point $\mathbf{k} \in Q$ [22, 23, 87]. Let $\nu \in \mathbf{R}^q$ be a descent direction, along which a dominating solution can be found. That is, for \mathbf{k} and ν there exists a $t_0 \in \mathbf{R}_+$ such that $\mathbf{k} + t_0\nu \prec \mathbf{k}$. Assume that the map is defined by the following equation,

$$\begin{aligned} \mathbf{k}(n) &= \mathbf{k}(n-1) + \Phi(\mathbf{k}(n-1)), \\ \mathbf{k}(0) &= \mathbf{k}_0 \in Q. \end{aligned} \quad (4.8)$$

The descent map Φ with a suitable step size can be used to search for the invariant set in Q . In the following, we present an example of the descent map by [23]. Define an auxiliary function,

$$\begin{aligned} \mathbf{g} &: \mathbf{R}^q \rightarrow \mathbf{R}^k \\ \mathbf{g}(\nu) &= \max_{i=1, \dots, k} [\mathbf{J}(\mathbf{k})\nu]_i, \end{aligned} \quad (4.9)$$

where $\mathbf{J}(\mathbf{k})$ denotes the Jacobian of \mathbf{F} at \mathbf{k} . \mathbf{g} is convex and positive homogeneous. With the help of \mathbf{g} , the following convex optimization problem defines a descent map,

$$\Phi_F(\mathbf{k}) = \min_{\nu \in \mathbf{R}^k} \left[\mathbf{g}(\nu) + \frac{1}{2} \|\nu\|_2^2 \right]. \quad (4.10)$$

Note that the Jacobian is calculated over the cell partition in the framework of cell mapping. Therefore, it is a finite difference approximation of the Jacobian. We should also point out that other search algorithms will lead to different maps. For a comprehensive survey of various search algorithms, see this book [88].

4.5 Multi-objective Optimal Design of Feedback Controls

Now, consider the multi-objective optimization approach to design feedback control gains of linear dynamical systems with or without time delay. In general, different objective (fitness) functions can be considered for optimization in time domain or frequency domain. For instance, peak time and overshoot are considered in time domain, while phase and gain margins are considered in frequency domain [55, 64, 89]. Here, we consider the multi-objective optimization design in time domain.

In particular, we consider the following multi-objective optimization problem to design the control gain \mathbf{k} or the weighting matrices \mathbf{Q} and \mathbf{R} in the performance index of the LQR control (4.2),

$$\min_{\mathbf{k} \in Q} \{t_p, M_p, e_{IAE}\}, \quad (4.11)$$

where Q represents the set of admissible control gains \mathbf{k} or the set of the weighting matrices \mathbf{Q} and \mathbf{R} , M_p stands for the overshoot of the response to a step reference input, t_p is the corresponding peak time and e_{IAE} is the integrated absolute tracking error

$$e_{IAE} = \int_0^{T_{ss}} |r(\hat{t}) - x(\hat{t})| d\hat{t}. \quad (4.12)$$

where $r(t)$ is a reference input and T_{ss} is the time when the response is close to be in the steady state. When the control magnitude is finite, the two factors, i.e. peak time and overshoot, are conflicting, while the integrated tracking error plays a compromising role.

4.6 Numerical Examples

4.6.1 First Order System plus Time Delay

Consider a first order plus time delay (FOPTD) system,

$$X(s) = \frac{K}{T_s + 1} e^{-\tau s} U(s), \quad (4.13)$$

where K and T are constants and τ is the time delay. Define a tracking error as $e = x - r$ where r is a reference input. We augment the plant with an additional state x_2 such that $\dot{x}_2 = x - r$, i.e. $x_2 = \int_0^t e(t)dt$. We set $x_1 = x$ and $\mathbf{x} = [x_1, x_2]^T$. The augmented state equations read

$$\begin{Bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{Bmatrix} = \begin{bmatrix} -\frac{1}{T} & 0 \\ 1 & 0 \end{bmatrix} \begin{Bmatrix} x_1 \\ x_2 \end{Bmatrix} + \begin{bmatrix} \frac{K}{T} \\ 0 \end{bmatrix} u(t - \tau) + \begin{bmatrix} 0 \\ -1 \end{bmatrix} r(t). \quad (4.14)$$

Note that a comparison study between the classical PI controller and the fractional order PI for this system is done by [90]. Here, we first convert the system (4.14) to the extended state Equation (4.5) without time delay, and then digitize Equation (4.5). We then apply the discrete time LQR formation of the system (4.5) to design the full extended state feedback control as

$$u(k) = -\mathbf{K}\mathbf{z}(k) = -\mathbf{k}_x\mathbf{x}(k) - k_1u(k - N) - k_2u(k - N + 1), \dots, k_Nu(k - 1), \quad (4.15)$$

where

$$\begin{aligned} \mathbf{z}(k) &= [\mathbf{x}(k), u(k - N), u(k - N + 1), \dots, u(k - 1)]^T, \\ \mathbf{x}(k) &= \mathbf{x}(k\Delta t), \\ u(k - i) &= u(k\Delta t - \tau_i), \quad i \in [0, N], \\ \mathbf{K} &= [\mathbf{k}_x, k_1, \dots, k_N], \end{aligned} \quad (4.16)$$

\mathbf{k}_x is 1×2 and represents the PI control gain for the original time-delayed system. Δt denotes the sample time. The performance index of the LQR optimal control reads

$$J = \frac{1}{2} \sum_{k=0}^{\infty} \mathbf{z}^T(k) \mathbf{Q} \mathbf{z}(k) + u(k) R u(k). \quad (4.17)$$

We have chosen the matrix $\mathbf{Q} = \text{diag}(Q_1, Q_2, 0, \dots, 0)$ in this example. In general, the weighting on the delayed control can be non-zero.

In the numerical example reported next, we have chosen $\tau = 0.5s$, $K = 1$ and $T = 1$. The sample time of the digital control is $\Delta t = 0.05s$. The discretization number of the time delay is $N = 10$. We take $\mathbf{k} = [Q_1, Q_2, R]$ as the design parameters of the MOP. The design space for the parameters is chosen as follows,

$$Q = \{[Q_1, Q_2, R] \in [1, 3] \times [200, 500] \times [0.011, 0.0134] \subset \mathbf{R}^3\}. \quad (4.18)$$

In the SCM method, we select the number of partitions of the design space Q as $\mathbf{N} = [20, 30, 10]$ without further subdivisions. The total CPU time for computing all the solutions is 27 minutes on a laptop PC.

Figure 4.1 shows the Pareto set of the MOP solution consisting of 1198 cells and Figure 4.2 shows the corresponding Pareto front. In the LQR control formulation, different parameters $[Q_1, Q_2, R]$ with the same proportionality will lead to the same optimal feedback gain \mathbf{K} and therefore the same Pareto front points. This explains why the Pareto front in Figure 4.2 contains so much overlap as compared to the Pareto front of the next example. The peak time of the Pareto front in Figure 4.2 is uniformly spaced at the interval equal to the sample time of the digital control. To yield a more dense distribution of the peak time, one can adopt continuous time control design instead of digital control. It is interesting to examine the closed-loop poles of the system on the Pareto set. Since the LQR design is done in the extended state space, N gains $[k_1, \dots, k_N]$ form a filter to regulate the history of the control. The gain vector \mathbf{k}_x describes the original system dynamics. We plot the closed-loop poles of the original system (4.14) on the Pareto set in Figure 4.3. The dot on the right represents the integrator pole and the line segment on the left contains the poles of the first order system. Figure 4.4 shows the closed-loop step response and the corresponding control with the gains that lead to the smallest e_{IAE} in the Pareto set. The control performance is quite satisfactory.

4.6.2 Second Order Linear Oscillator

Consider a second order oscillator subject to a PID control.

$$\ddot{x} + 2\zeta\omega_n\dot{x} + \omega_n^2x = \omega_n^2u(t), \quad (4.19)$$

where $\omega_n = 5$, $\zeta = 0.01$,

$$u(t) = k_p[r(t) - x(t)] + k_i \int_0^t [r(\hat{t}) - x(\hat{t})] d\hat{t} - k_d\dot{x}(t), \quad (4.20)$$

$r(t)$ is a step input, k_p , k_i and k_d are the PID control gains. We consider the MOP in Section ?? with the control gains $\mathbf{k} = [k_p, k_i, k_d]^T$ as design parameters. The design space for the parameters is chosen as follows,

$$Q = \{\mathbf{k} \in [10, 50] \times [1, 30] \times [1, 2] \subset \mathbf{R}^3\}. \quad (4.21)$$

Initially, we select the number of divisions in the three control gain intervals as $\mathbf{N} = [30, 18, 8]$. The integrated absolute tracking error e_{IAE} is calculated over time with $T_{ss} = 20$ seconds. After the first run of the SCM program, we refine the Pareto set with $7 \times 7 \times 7$ subdivisions. The total CPU time for this example is 20 minutes. The closed-loop response of the system for each design trial is computed with the help of closed form solutions.

Figure 4.5 shows the Pareto set of the MOP solution and Figure 4.6 shows the corresponding Pareto front. 5961 cells are in the Pareto set. In this case, the

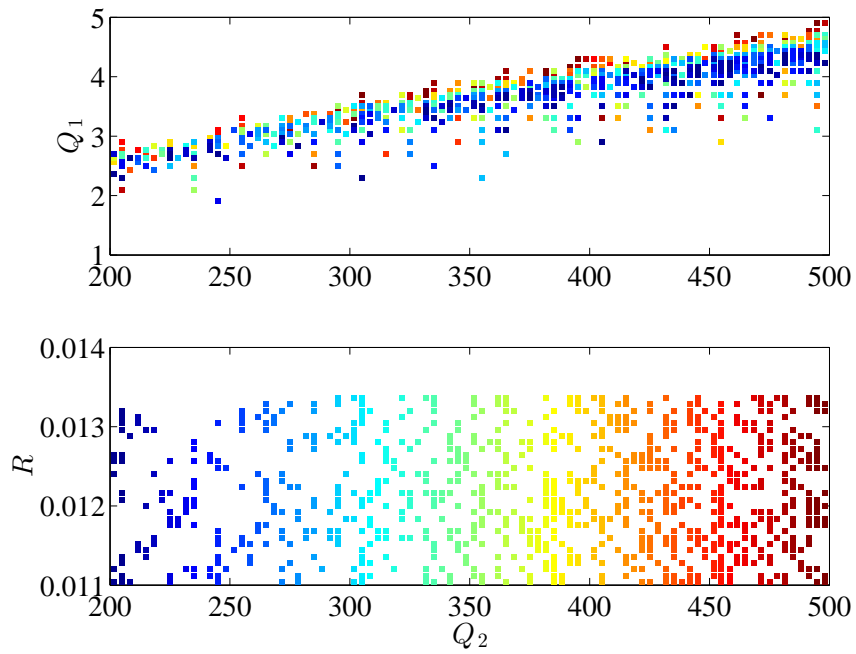


Figure 4.1: The Pareto set of $[Q_1, Q_2, R]$ for the multi-objective LQR optimal control of the first order system with time delay. The color code indicates the level of the other design variable. Red denotes the highest value, and dark blue denotes the smallest value.

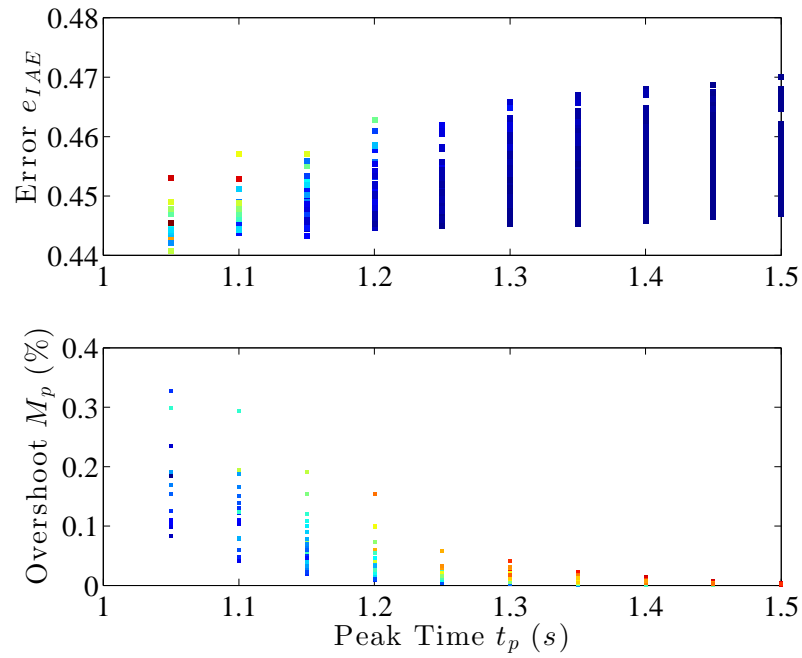


Figure 4.2: The Pareto front of $[t_p, M_p, e_{IAE}]$ for the multi-objective LQR optimal control of the first order system with time delay corresponding to the Pareto set in Figure 4.1. The color code indicates the level of the other objective function. Red denotes the highest value, and dark blue denotes the smallest value.

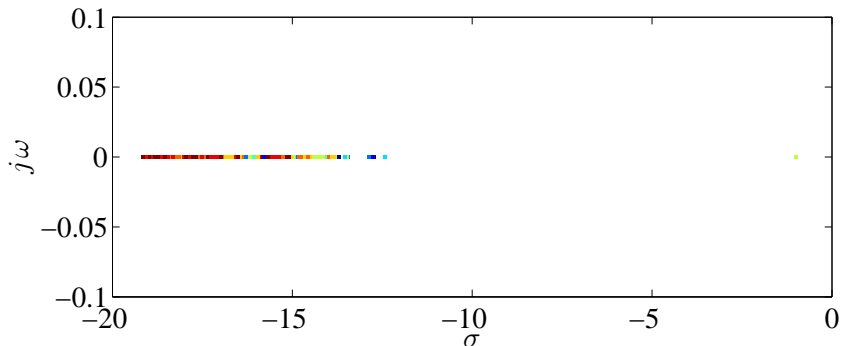


Figure 4.3: The first two dominant closed-loop poles for the multi-objective LQR optimal control of the first order system with time delay corresponding to the Pareto set in Figure 4.1. The color code indicates the level of the first objective function t_p . Red denotes the highest value, and dark blue denotes the smallest value.

Pareto front exhibits a fine structure, which has not been seen before, and cannot be readily obtained with random search algorithms such as the genetic algorithm. Figure 4.7 shows the closed-loop poles of the system on the Pareto set. The general location of the cluster of the poles is along the $\pm 45^\circ$ lines to the left of the imaginary axis, which is consistent with the well-known feedback control intuitions [84]. Figure 4.8 shows the closed-loop step response and the control with a gain that leads to the smallest e_{IAE} in the Pareto set. The performance functions are indicated in the figure caption.

4.7 Summary

We have reviewed the MOP formulation and the simple cell mapping (SCM) method applied to the MOP. We have then studied the multi-objective optimal time domain design of feedback controls for linear systems with or without time delay with the help of the SCM method. It should be pointed out that we have only considered simple feedback control examples for the purpose of demonstrating the proposed SCM-MOP design method. There are different structures of controls that render perfect tracking. We have considered two different sets of design parameters for the feedback control: a LQR based approach with the weighting matrices as design parameters, and a direct optimization with feedback gains as design parameters. Both approaches prove to be quite effective. The Pareto set and Pareto front consisting of the peak time, overshoot and integrated absolute tracking error are obtained for examples of two linear control systems, one of which has a control time delay. It is interesting to note that for the second order linear system, we have found

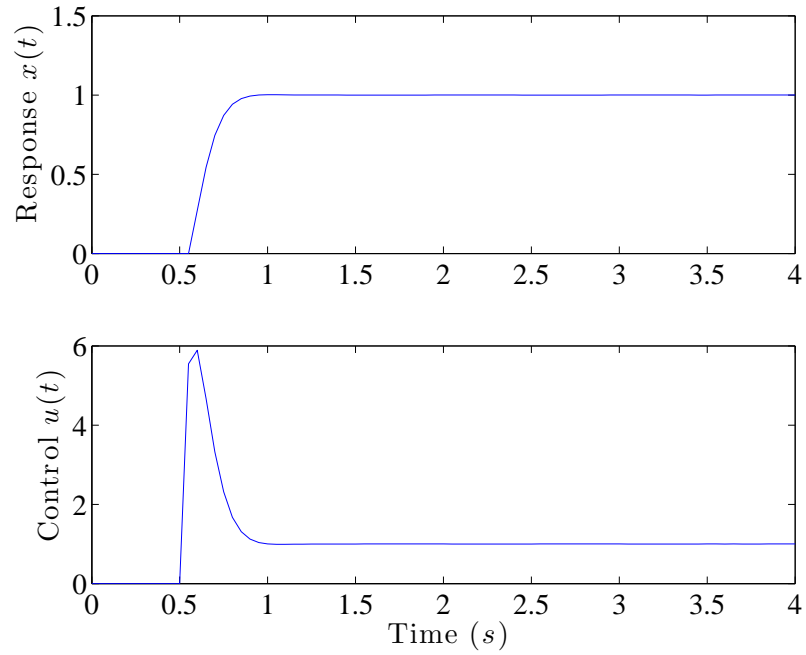


Figure 4.4: An example of the closed-loop step response and control of the first order system with time delay. The PID gains are designed with the multi-objective LQR optimal control formulation. The optimal design parameters are $[Q_1, Q_2, R] = [3.10, 495.0, 0.012]$. The optimal gains for the original state \mathbf{x} are $\mathbf{k}_x = [53.62, 110.93]$. The performance indices are $[t_p, M_p, e_{IAE}] = [1.05, 0.1839, 0.4407]$.

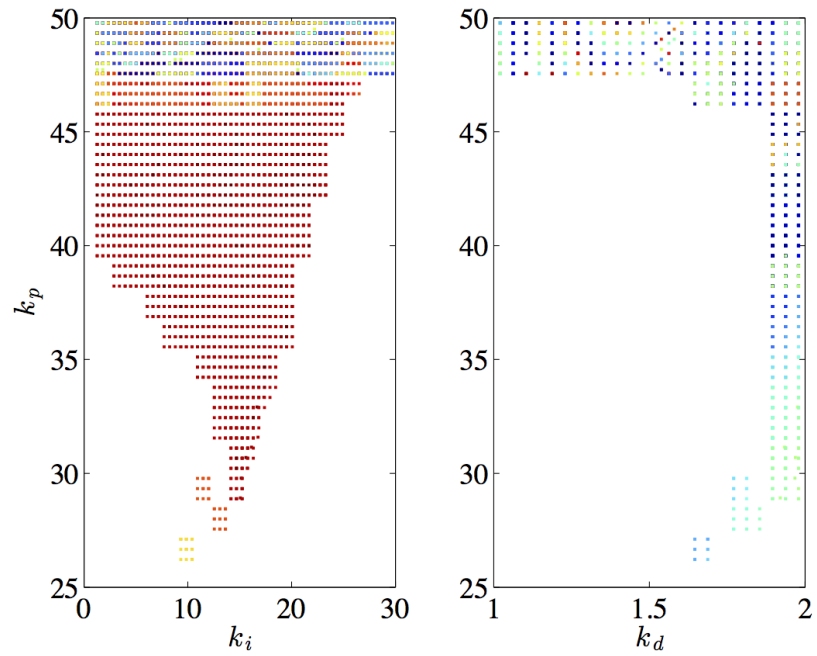


Figure 4.5: The Pareto set of $[k_p, k_i, k_d]$ for the multi-objective optimal control of the second order system. The color code indicates the level of the other design variable. Red denotes the highest value, and dark blue denotes the smallest value.

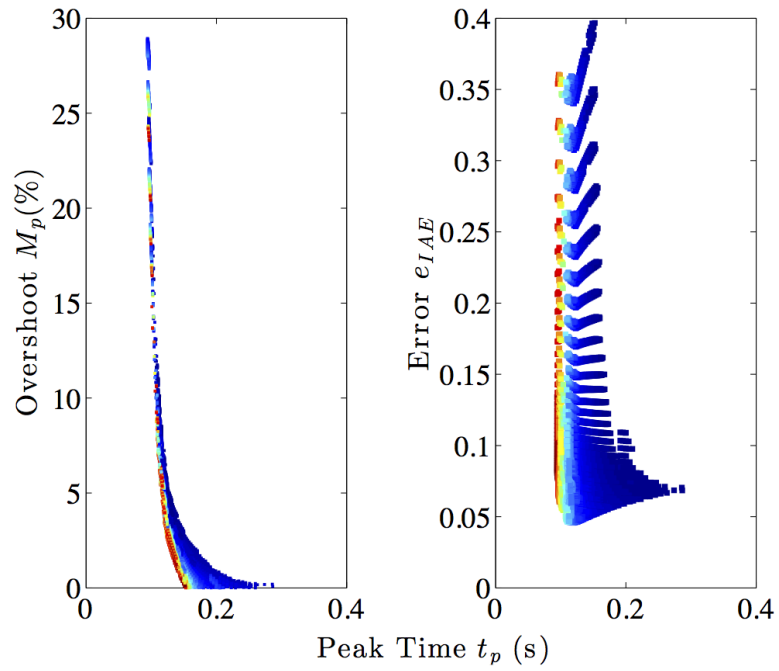


Figure 4.6: The Pareto front of $[t_p, M_p, e_{IAE}]$ for the multi-objective optimal control of the second order system corresponding to the Pareto set in Figure 4.5. The color code indicates the level of the other objective function. Red denotes the highest value, and dark blue denotes the smallest value.

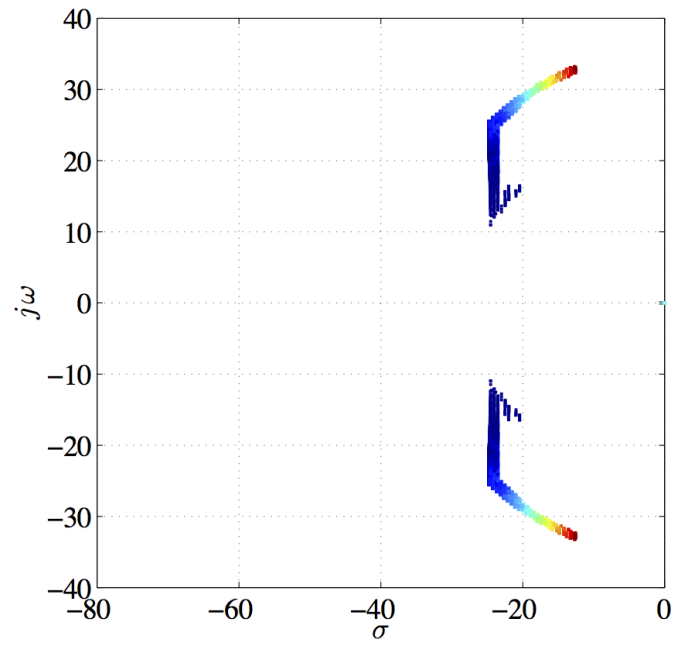


Figure 4.7: The closed-loop poles for the multi-objective optimal control of the second order system corresponding to the Pareto set in Figure 4.5. The color code indicates the level of the second objective function M_p . Red denotes the highest value, and dark blue denotes the smallest value.

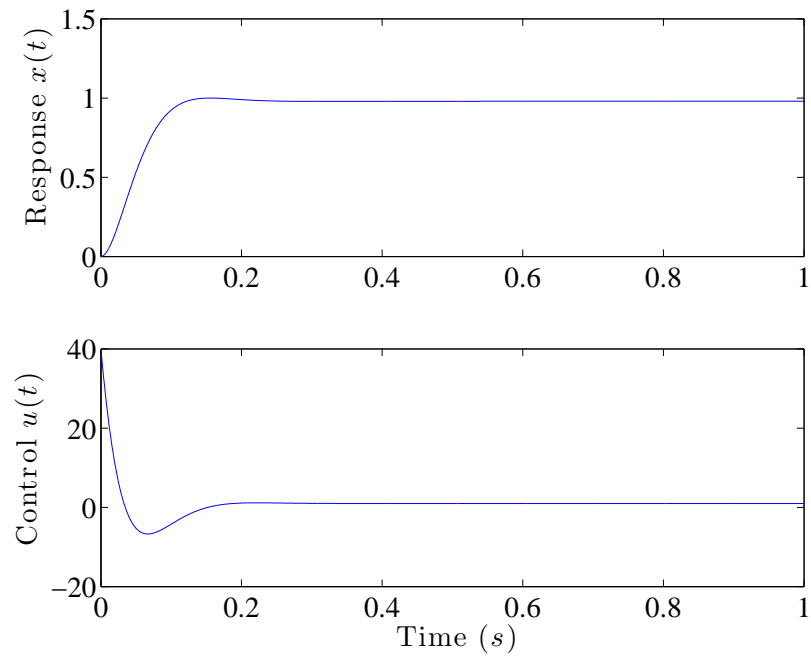


Figure 4.8: An example of the closed-loop step response and control of the second order system with $[k_p, k_i, k_d] = [40.0, 2.8796, 1.9792]$. The PID gains are designed with the multi-objective optimal control. $[t_p, M_p, e_{IAE}] = [0.1555, 0.0, 0.2774]$.

a fine structure of the Pareto front, which has been very difficult to obtain using stochastic search algorithms such as the genetic algorithm. This study suggests that the SCM method is a powerful method that can provide global and fine-structured solutions of MOPs for complex dynamical systems.

Chapter 5

MOP OF DISTRIBUTED-ORDER FRACTIONAL DAMPING

5.1 Introduction

The available tools and foundations of fractional calculus triggered numerous studies in science and engineering [91]. Fractional models have applications in various mechanical, physical, biological, electronic and chemical systems [91–93]. Viscous damper is one of the examples in mechanical systems. The damper reduces oscillations and stabilizes systems. Conventional viscous damper force is proportional to the velocity, an integer order derivative of the displacement. It has been found that the fractional order derivative can model the viscous damping material much better [94,95]. Recent advances in fractional order calculus even suggest more advantages of distributed order derivative in modeling viscous damping. However, such a model almost immediately precludes the efficient design of optimal damping for various applications in dynamical and control systems.

Many studies have focused on modelling the damping elements with fractional order differential operators [94–97]. For instance, De Espíndola *et al.* [98] designed an optimum viscoelastic vibration absorber for a given material using the fractional calculus model. Padovan *et al.* [99] found the response of viscoelastic systems for different excitations. Shokooh *et al.* [100] compared numerical methods applied to fractional models of damping materials. Koeller [101] showed that a fractional viscoelastic damping model is superior to the integer order model. Lorenzo and Hartley [102] have successfully utilized distributed order operators to model a viscoelastic material with spatially varying properties. Jiao *et al.* [103] introduced many applications of distributed order operators in dynamical systems including distributed order filtering and optimal damping.

Optimal design of distributed order fractional damping systems may lead to significant improvement in performance of such systems. Sheng and Chen [104] studied the application and design of an optimal distributed order fractional damping with the integrated squared error J_{ISE} as the objective to be minimized. In general, the goal of damping design is to effectively remove oscillatory responses of the system. The damping can also be designed for better step response characteristics. In this case, the damping is optimized to reduce the overshoot as well as the oscillatory

response in the settling phase of the step response. In the regard, the optimal design of the distributed order fractional damping based on the step response is consistent with the conventional damping design [105]. Since the step response is characterized by the conflicting criteria including the rise time, overshoot, settling time and integrated tracking error, the design of a distributed order fractional damper naturally leads to a multi-objective optimization problem (MOP).

The solution of MOPs is not a point in the design space, but rather forms a set, called the *Pareto set*. The corresponding objective functions are called the *Pareto front*. There have been various studies on multi-objective optimal design of dynamical and control systems with the help of stochastic methods [58–61] as well as deterministic methods [106]. Evolutionary algorithms are widely used for MOPs [1]. Although evolutionary algorithms have shown strength in many MOP applications, there is always an uncertainty left regarding whether the global Pareto set has indeed be found. Another approach for solving MOPs is the cell mapping method proposed by Hsu [25]. Two forms of the method are the simple cell mapping (SCM) and the generalized cell mapping (GCM) [24, 25]. The advantage of these methods is that they generate global solutions of the Pareto set. In addition, they are applicable to a wide range of problems with significant robustness.

This chapter applies the hybrid algorithm presented in Chapter 3 to multi-objective optimal design of distributed order fractional damping. The hybrid method starts by looking for rough solutions of the MOP with an evolutionary algorithm such as NSGA II [26]. The rough solutions are the starting point of a gradient free simple cell mapping method, which is to discover the solution of the Pareto set without sweeping the entire design space. A subdivision algorithm can be applied to further refine the Pareto set represented by a collection of cells [106]. We should note that this work is a theoretical research. It is difficult to validate the optimal designs of the distributed and fractional order damping experimentally using mechanical hardware. However, there have been studies to realize such optimal damping in digital controls [107, 108].

In Section 5.2, the distributed-order fractional damper is introduced. The choice of the objective functions is presented in Section 5.3. Section 5.4 explains the choices of the distribution function. The numerical results and discussions are presented in Section 5.5 and the conclusions are summarized in Section 5.6.

5.2 Distributed-Order Fractional Damper

A fractional order damper force is defined as

$$f_d(t) = -c {}_0D_t^\alpha x(t), \quad (5.1)$$

where ${}_0D_t^\alpha x(t)$ is the Reimann-Liouville fractional order derivative of the displacement $x(t)$. The Reimann-Liouville derivative of a function of time $f(t)$ is defined as [93]

$${}_0D_t^\alpha f(t) = \frac{1}{\Gamma(n - \alpha)} \left(\frac{d}{dt} \right)^n \int_a^t (t - \tau)^{n-\alpha-1} f(\tau) d\tau, \quad (5.2)$$

where $n = [\alpha] + 1$, $[\alpha]$ denotes the integer part of α , and $\Gamma(\cdot)$ is the gamma function. Compared to the integer order damping, the fractional order damper model has more flexibility and captures the history dependent property of many viscoelastic damping materials. Further flexibility and freedom can be obtained by considering the damper force as a function of a weighted sum of discrete fractional order terms

$$f_d(t) = - \sum_{i=1}^m c_i {}_0D_t^{\alpha_i} x(t), \quad (5.3)$$

or a function of continuously distributed order derivative,

$$f_d(t) = - \int_a^b c(\alpha) {}_0D_t^\alpha x(t) d\alpha, \quad (5.4)$$

where $c(\alpha)$ is an order dependent damping coefficient.

In this study, we consider a mass-spring-damper system and compare the step response when it has an integer order damper and a distributed order damping. The transfer functions for the system in these cases are given by

$$G(s) = \frac{1}{ms^2 + cs + k}, \quad (5.5)$$

and

$$G(s) = \frac{1}{ms^2 + \int_a^b c(\alpha) \cdot s^\alpha d\alpha + k}, \quad (5.6)$$

where m is the mass, c is the integer order damping coefficient, k is the spring coefficient, and $c(\alpha)$ is defined above.

5.3 Objective Functions

The time domain specifications of the step response include the rise time t_r , overshoot M_p , peak time t_p , settling time t_s , integrals of tracking errors such as the integrated squared error J_{ISE} , integrated time-weighted absolute error J_{ITAE} , integrated time-weighted squared error J_{ITSE} , and integrated absolute error J_{IAE} [106]. It is important to note that any two objectives of MOPs need to be conflicting in a domain of the design space, meaning that increasing one of them results in decreasing the other, and vice versa. If two objectives have the same trend over the domain, only one of them will be sufficient in the optimization problem.

In the current study, three objectives are considered including the peak time t_p , over-shoot M_p and integrated squared error J_{ISE} defined as

$$J_{ISE} = \int_0^{\infty} e^2(t)dt, \quad (5.7)$$

where $e(t)$ is the tracking error defined by the difference between the reference input and system output. Hence, the MOP for the damping design reads,

$$\min_{\mathbf{k} \in Q} \{\mathbf{F}(\mathbf{k})\} = \min_{\mathbf{k} \in Q} \{t_p, M_p, J_{ISE}\}, \quad (5.8)$$

where the design space Q and the design parameters \mathbf{k} are to be determined for each example reported in the next section.

The solution technique used for solving the MOP is the hybrid method which takes advantages of both GA and SCM methods. The hybrid method starts with the GA. An initial population is evolved over a small number of iterations or generations leading to a collection of random points near or on the Pareto set. This rough solution is then fed to the SCM. The set of cells is identified that contains the random points from the GA. A neighborhood search procedure and a recovery strategy are implemented over the cellular space to discover the Pareto set. The cells can be further subdivided to obtain much refined results [109].

5.4 Distribution Function $c(\alpha)$

Three different functions of $c(\alpha)$ are considered next. These cases have been considered in a single-objective optimization study [104]. The output of the system $Y(s)$ to a reference input $R(s)$ is given by $Y(s) = G(s)R(s)$. We shall consider the step input $R(s) = 1/s$. The step response of the system is computed with the numerical inverse Laplace transform (NILT) algorithm [110].

Case I: $c(\alpha) = \gamma$ where γ is a constant.

The step response reads

$$Y(s) = \frac{1}{s} \frac{1}{ms^2 + \gamma(s^b - s^a)(\ln s)^{-1} + k}. \quad (5.9)$$

The design parameters are a , b and γ .

Case II: $c(\alpha) = \gamma e^{\beta\alpha}$, $a = 0$ and $b = 1$.

The step response is

$$Y(s) = \frac{1}{s} \frac{1}{ms^2 + \gamma(e^{\beta s} - 1)(\beta + \ln s)^{-1} + k}. \quad (5.10)$$

The design variables are γ and β .

Case III: $c(\alpha) = \beta\alpha + \gamma$.

The step response is

$$Y(s) = \frac{1}{s} \frac{1}{ms^2 + [(\beta b \ln s - \beta + \gamma \ln s)s^b - (\beta a \ln s - \beta + \gamma \ln s)s^a](\ln s)^{-2} + k}. \quad (5.11)$$

The design variables are a , b , γ and β .

5.5 Results and Discussions

Case I: $c(\alpha) = \gamma$.

The design space Q of the parameters $\mathbf{k} = \{a, b, \gamma\}$ is defined as,

$$Q = \{\mathbf{k} \in \mathbf{R}^3 \mid 0.5 \leq a, b \leq 1.2, 5 \leq \gamma \leq 15\}. \quad (5.12)$$

The number of divisions N for each design parameter is 10. The number of subdivisions Div for refinement of the solution is 11 [106]. The Pareto front and Pareto set are shown in Figures 5.1 and 5.2. As can be seen from the figures, M_p and t_p are in conflict relationship all over the design space. There is a cluster of points in the vicinity of $t_p = 3.1s$. This cluster is related to the change of J_{ISE} as t_p increases. When $t_p < 3.1s$, all the three objectives are conflicting to each other. When $t_p > 3.1s$, J_{ISE} and t_p are no longer conflicting. It appears that this transition causes the occurrence of the cluster.

The step responses of five different designs are depicted in Figure 5.3. The results in the figure indicate that the design to optimize only one of the three objectives is not the best choice. For example, the design for the smallest peak time is associated with a larger overshoot and oscillations. Similarly, the system designed for the smallest J_{ISE} suffers from a relatively high overshoot and vibrations. In contrast, a design to achieve the lowest M_p leads to very slow response with no oscillations. The best design should compromise all three objectives, as shown in the figure with the dotted red line. The objective function values are listed in Table 5.1.

Figure 5.3 also compares the step response of the system with an integer order damper with that of the fractional order MOP design. The integral of the squared difference between the step response of the integer order system and the fractional order MOP design, denoted as Δ , is computed as a function of the integer order damping coefficient. The result is plotted in Figure 5.4. The integer order damping coefficient c that best matches the fractional order MOP design such that Δ reaches minimum is found from the figure as 1.208. This is the coefficient used to compute the step response of the system with the integer order damping. It appears that the fractional order MOP design is superior to the integer order damping as

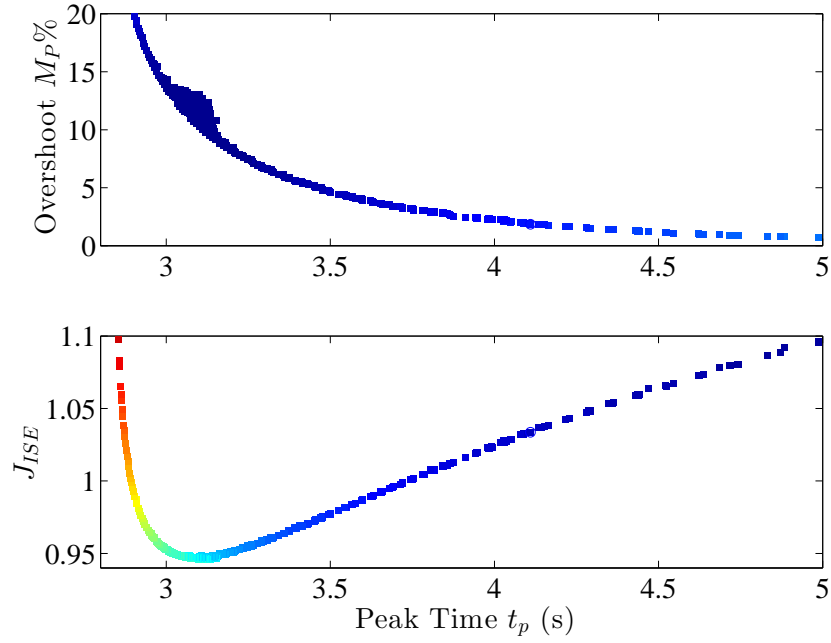


Figure 5.1: The Pareto front of the system in Case I. The color code is based on the objective which is not present in the plot. In all figures, the red color represents higher value and the blue color represents lower value.

Table 5.1: Different objective function values of the designs shown in Figure 5.3 for Case I.

Design	t_p	M_p	J_{ISE}
$\min(t_p)$	2.851	31.20	1.145
$\min(M_p)$	18.168	0.0101	1.517
$\min(J_{ISE})$	3.128	12.35	0.9459
MOP	3.411	5.519	0.9685

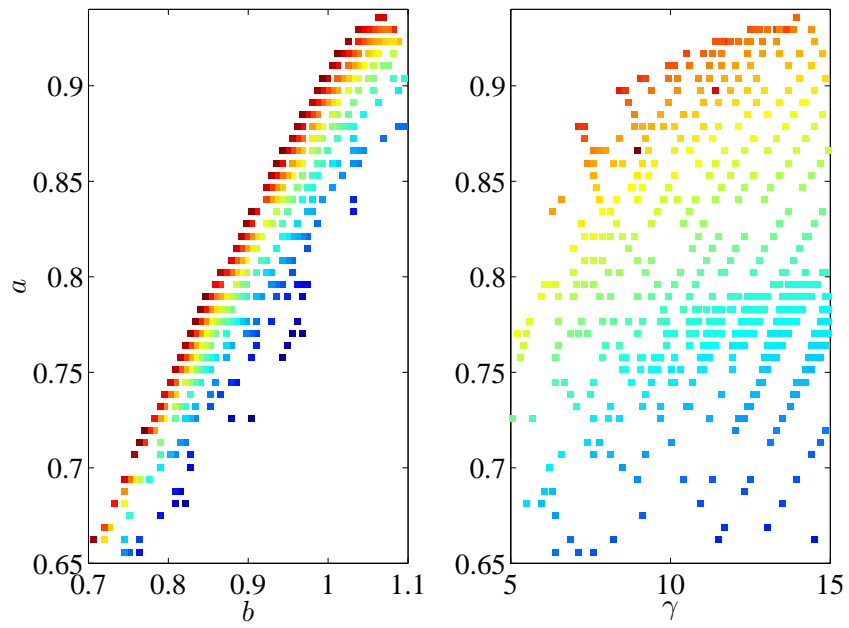


Figure 5.2: The Pareto set of the system in Case I. The color code is based on the value of the variable which is not present in the plot.

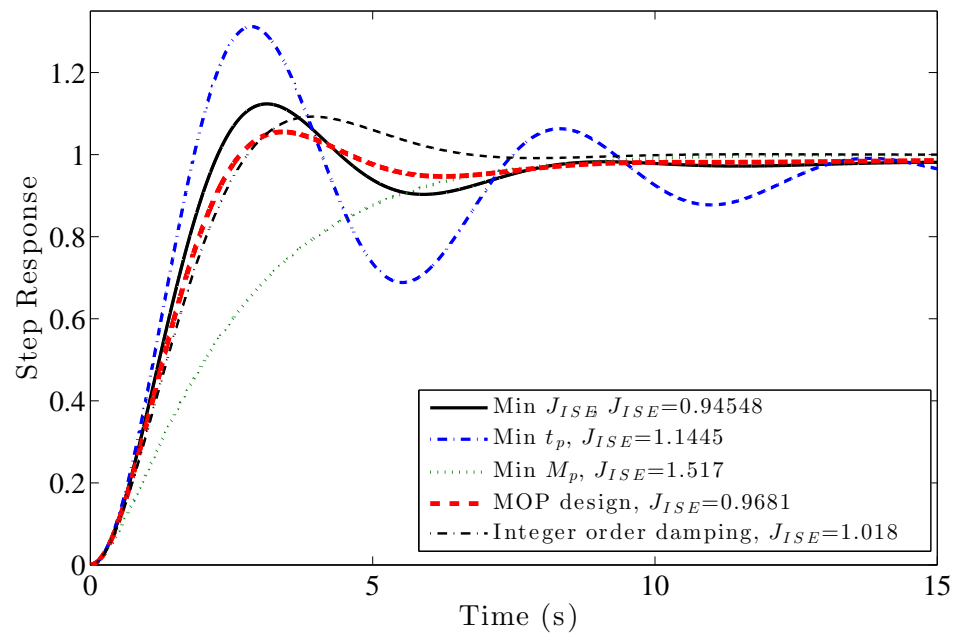


Figure 5.3: The step responses of the system in Case I with different design priorities for the objectives t_p , M_p and J_{ISE} . The last case shows the best integer order damper.

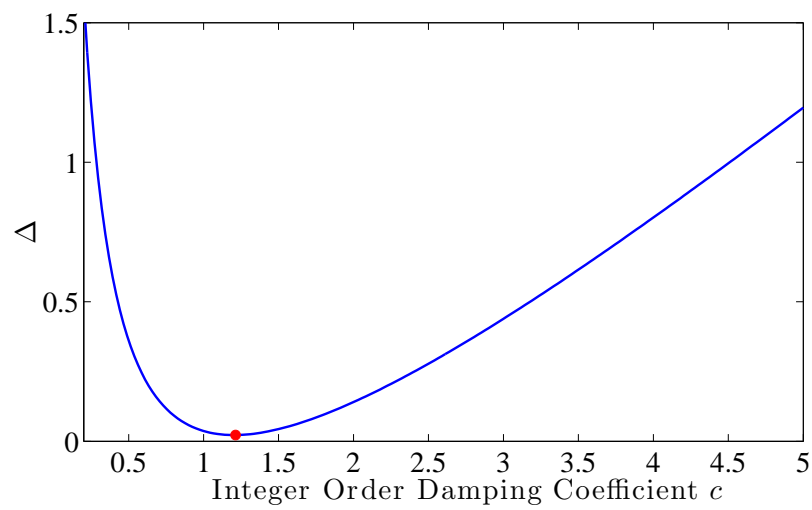


Figure 5.4: The integral of the squared difference between the step response of the integer order system and the fractional order MOP design as a function of the integer order damping coefficient. The best integer order damping coefficient to match the MOP design is $c = 1.208$ and $\Delta = 0.0231$.

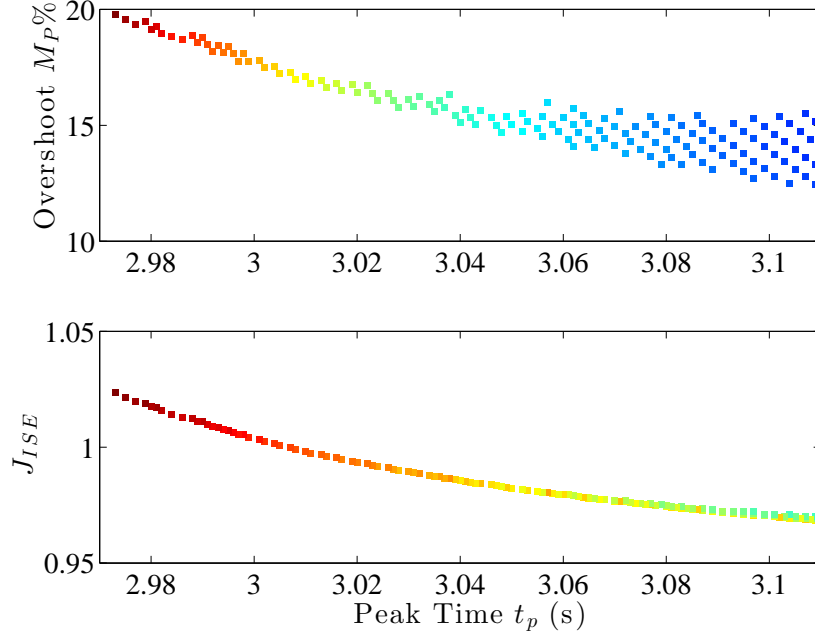


Figure 5.5: The Pareto front of the system in Case II. The color code is based on the value of the objective which is not present in the plot.

is shown in Figure 5.3. In the following cases, this observation still holds true. No further comparison of integer and fractional order damping is presented.

Case II: $c(\alpha) = \gamma e^{\beta\alpha}$.

The design space Q of the parameters $\mathbf{k} = \{\beta, \gamma\}$ is defined as,

$$Q = \{\mathbf{k} \in \mathbf{R}^2 \mid 1 \leq \beta \leq 20, 0 \leq \gamma \leq 3\}. \quad (5.13)$$

The number of divisions N for each design parameter is 205. The number of subdivisions Div for refinement of the solution is 11.

Figures 5.5 and 5.6 show the Pareto front and Pareto set, respectively. In this case, the peak time t_p is clearly in conflict with the overshoot M_p and the tracking error J_{ISE} . As is in Case I, one peak time in a certain range may be associated with several overshoot values. This is different from the integer second order systems where M_p and t_p have a one-to-one relationship.

The Pareto set in the design space of γ and β is shown in Figure 5.6. The step responses with four different designs are shown in Figure 5.7. As in Case I, a MOP design with a compromise among the objectives delivers the best response.

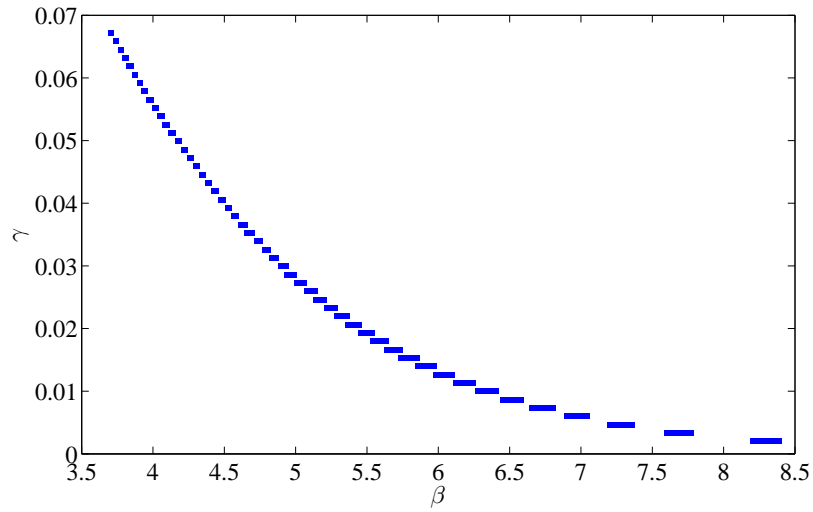


Figure 5.6: The Pareto set of the system in Case II.

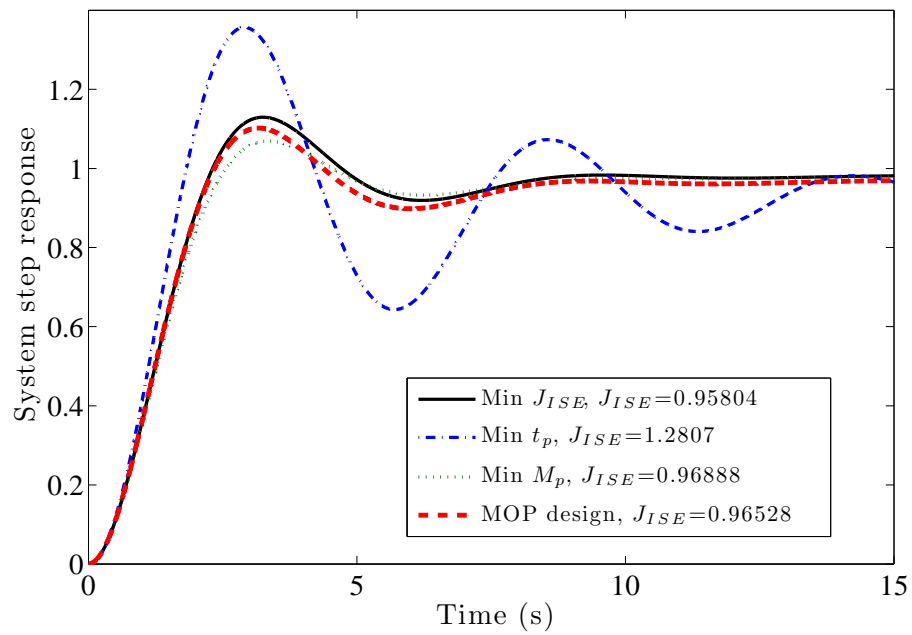


Figure 5.7: The step responses of the system in Case II with different design priorities for the objectives t_p , M_p and J_{ISE} .

Table 5.2: Different objective function values of the designs shown in Figure 5.7 for Case II.

Design	t_p	M_p	J_{ISE}
$\min(t_p)$	2.903	35.67	1.281
$\min(M_p)$	3.347	6.925	0.9693
$\min(J_{ISE})$	3.251	12.95	0.9585
MOP	3.172	10.21	0.9657

The objective function values of the four designs shown in Figure 5.7 are listed in Table 5.2.

Case III: $c(\alpha) = \beta\alpha + \gamma$.

The design space Q of the parameters $\mathbf{k} = \{a, b, \beta, \gamma\}$ is defined as,

$$Q = \{\mathbf{k} \in \mathbf{R}^4 \mid 0.5 \leq a, b \leq 1.2, 5 \leq \beta \leq 20, 10 \leq \gamma \leq 30\}. \quad (5.14)$$

The number of divisions N for each design parameter is 40. No sub-division is applied. The Pareto front is shown in Figure 5.8, which is very similar to that in Figure 5.1. The Pareto set is shown in Figure 5.9. The large gaps in the Pareto front and Pareto set are due to the relatively large cells without refinement. Figure 5.10 presents the step responses of the system with four different choices of damping design. It should be noted that the current damping coefficient $c(\alpha)$ leads to a smaller tracking error J_{ISE} than the one reported in [96], and hence is a good choice when minimizing J_{ISE} has a high priority. By comparing the objective function values in Tables 5.1 to 5.3, it appears that the more degrees of freedom in the multi-objective optimal design of the damper offer an overall improvement in the step response of the system.

5.6 Summary

A hybrid multi-objective optimization method has been successfully applied to design a distributed order fractional damper for a simple mechanical oscillator. Three different cases of distributed damping coefficient $c(\alpha)$ have been studied. The objective functions include the overshoot M_p , the peak time t_p and the integrated tracking error J_{ISE} . It has been found that the multi-objective optimal designs of the damper can deliver much improved step responses in terms of the three objective functions, compared with the single-objective optimal design by minimizing J_{ISE} .

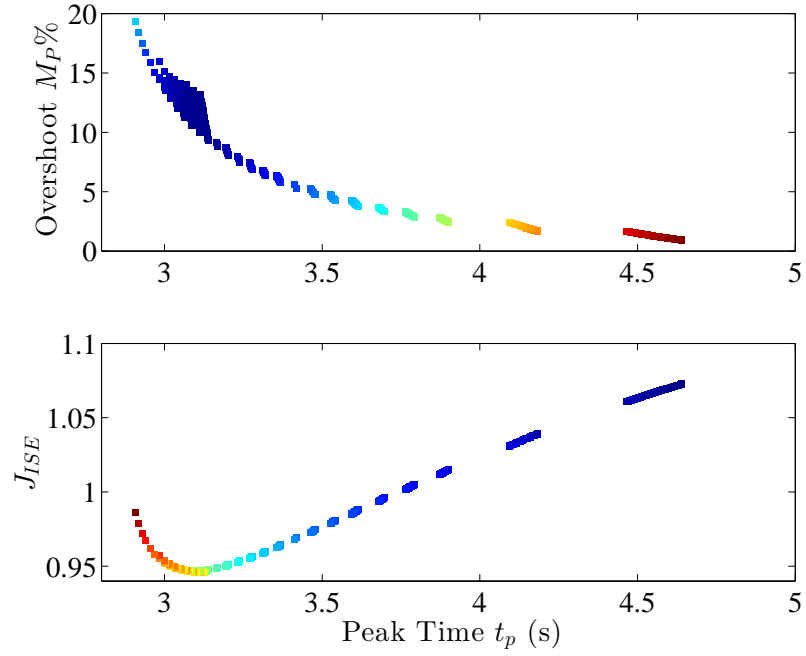


Figure 5.8: The Pareto front for Case III. The color code is based on the value of the objective which is not present in the plot.

Table 5.3: Different objective function values of the designs shown in Figure 5.10 for Case III.

Design	t_p	M_p	J_{ISE}
$\min(t_p)$	2.850	32.54	1.171
$\min(M_p)$	5.636	0.2846	1.126
$\min(J_{ISE})$	3.119	12.27	0.9457
MOP	3.688	3.4883	0.9948

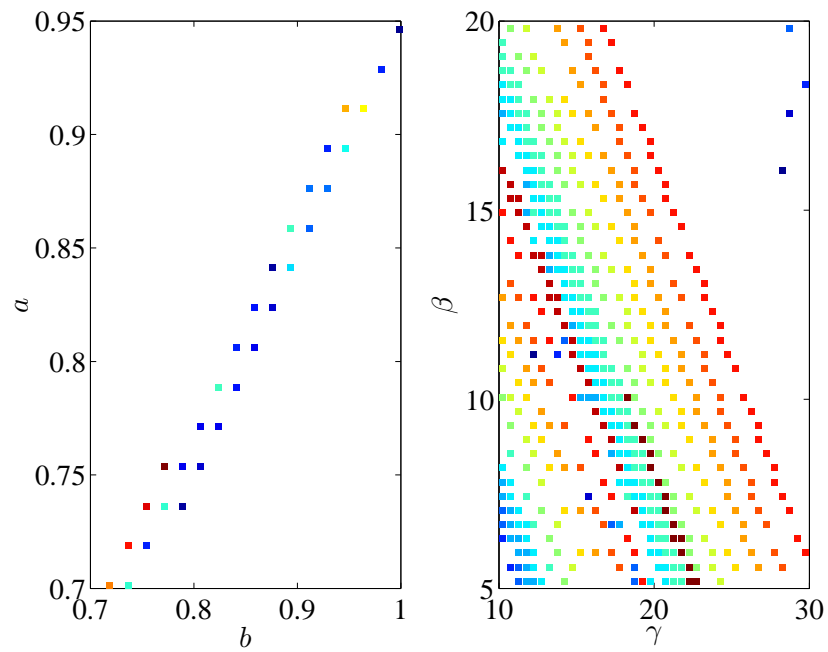


Figure 5.9: The Pareto set for Case III. The color code of the left plot is based on the value of β and the right plot is based on a .

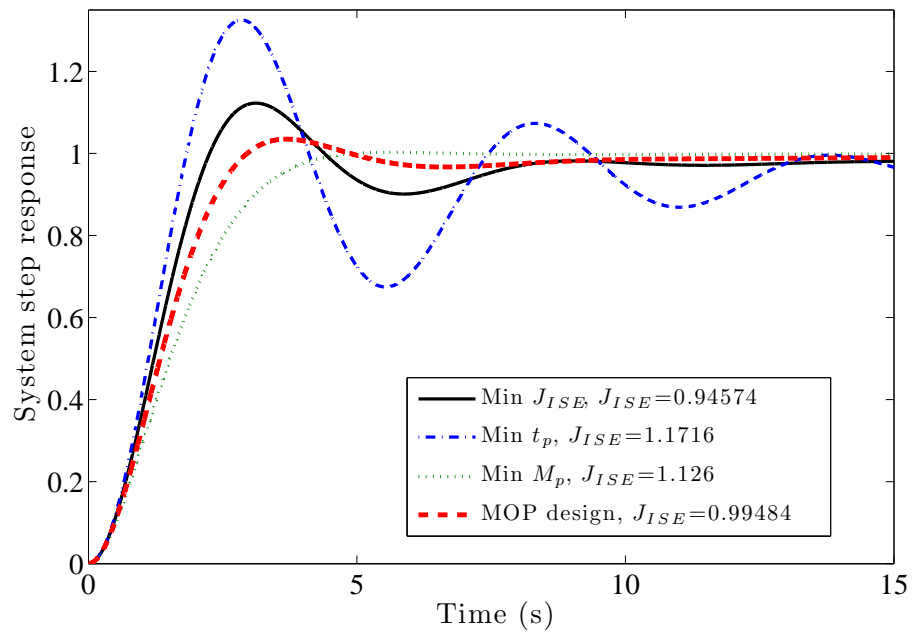


Figure 5.10: The step responses of the system in case III with different design priorities for the objectives t_p , M_p and J_{ISE} .

alone. Furthermore, while the three cases of $c(\alpha)$ lead to similar optimal designs in terms of the Pareto front, better performance can be achieved with larger number of design parameters.

Chapter 6

MULTI-OBJECTIVE OPTIMAL PATH PLANNING

6.1 Introduction

Many real-world applications of robot path planning involves not only finding the shortest path, but also achieving some other objectives such as minimizing fuel consumption or avoiding danger areas. This chapter introduces a 2D path planning scheme that solves a multi-objective path planning problem on a 3D terrain. This allows the controller to pick the most suitable path among a set of optimal paths. The algorithm generates a cellular automaton for the terrain based on the objectives by applying various weighting factors via an evolutionary algorithm and finds the optimal path between the start point and the goal for each set of parameters considering static obstacles and maximum slope constraints. All the final trajectories share the same characteristic that they are non-dominated with respect to the rest of the set in the Multi-Objective Optimization Problems (MOP) context. The objectives considered in this study includes the path length, the elevation changes and avoiding the radars. Testing the algorithm on several problems showed that the method is very promising for mobile robot path planning applications.

Path planning is an important part of the robotic system design, and has been extensively studied in the past [111, 112]. Among the vast literature on path planning problems, there are only a few that consider several objectives simultaneously. Mittal did a multi-objective UAV path planning with B-spline control points as variables [113], and was able to find effective paths. Ahmed used a modified non-dominated sorting genetic algorithm NSGAI to optimize multiple objectives on a plane [114]. Davoodi used a modified genetic algorithm to optimize the path length and clearance for 2D planning problems [115]. Zhang applied the multi-objective particle swarm optimization algorithm to do robot path planning in uncertain environments [116].

Cellular automata (CA) was invented as a discrete mathematical modeling tool in 1940s. Behring was one of the first who successfully did path planning with CA [117]. An optimized CA path planning algorithm was developed by Tavakoli *et al.* for a multi-agent system that was superior to the so-called A* algorithm [118]. Cellular automata has become more popular for path planning algorithms recently [119–122]. One of the advantages of the CA method is that it offers flexibilities to

incorporate different rules for various applications. This paper presents a version of cellular automata that contains rules for multi-objective optimal path planning.

In this work, we focus on multi-objective path planning for a mobile robot which travels between two points on a non-flat surface subjected to static constraints and ascend/descend slope constraints. We apply CA to find the most feasible path based on several objectives including the travel distance, the elevation changes and the distance from radars on the terrain. The algorithm presented in this study discretizes the domain into a collection of finite dimensional cells and scores each cell with CA rule consisting of the weighted objectives. The weighting factors are then taken as the design parameters for the multi-objective optimization problem. An evolutionary algorithm is applied to find the Pareto optimal set of parameters.

This chapter is structured as follows. Section 6.2 presents the preliminaries for this study. Section 6.2.1 states the problem, the method and the assumptions. The cellular automata rule and solution method are discussed in section 6.2.2. The case studies and results are presented in 6.3. The chapter concludes in Section. 6.4

6.2 Method

6.2.1 Framework

We simplify the continuum environment to an $m \times n$ grid of cells. A cell may either be free, obstacle, the starting point or the goal. A value of zero, one or three is initially assigned to the cells representing the free space, obstacle, or goal, respectively. We assume that for each cell the topography of the terrain is known and the radar exposure penalty (REP) is specified. The REP describes the probability that the robot could be identified by radar and is defined below.

Consider k radars located at $(X_j, Y_j), j = 1 \dots k$ each having an effective range R_j . The REP for cell i -denoted by Γ_i - is calculated as

$$\Gamma_i = \sum_{j=1}^k \left(1 - \frac{\delta(i, j)}{R_j}\right)^2 H\left(1 - \frac{\delta(i, j)}{R_j}\right), \quad (6.1)$$

where R_j is the effective range of radar j , $\delta(i, j)$ is the planar distance between cell i and radar j , $\delta(i, j) = \sqrt{(X_i - X_j)^2 + (Y_i - Y_j)^2}$, and $H(x)$ is the Heaviside step function. In Equation (6.1). The radar signal strength is inversely proportional to the squared distance between the radar and the cell.

We consider the Moore neighbors (Figure 6.1) as the possible movement from a cell. $\eta(i)$ denotes the set of admissible Moore neighbors of cell i excluding obstacle and out of range cells. Let $\Delta(i, x)$ denote the Cartesian distance between cell i and its neighbor x .

$$\Delta(i, x) = \sqrt{(X_i - X_x)^2 + (Y_i - Y_x)^2 + (Z_i - Z_x)^2}, \quad (6.2)$$

where (X_i, Y_i, Z_i) are the coordinates of cell i . Let $\Delta h(i, x) = |Z_i - Z_x|$ be the elevation difference between cell i and its neighbor x .

6.2.2 Cellular Automata

The cellular automaton rule proposed in this study is as follows:

$$S_i^{t+1} = \begin{cases} 1 & \text{if } S_i^t = 1 \\ S_i^t & \text{if } \nexists x \in \eta(i) | S_x^t > 2 \\ S_{comp} & \text{if } \exists x \in \eta(i) | S_x^t > 2 \text{ AND } \frac{Z_i - Z_x}{\delta(i, x)} < m_{all} \end{cases}, \quad (6.3)$$

$$S_{comp} = \min \{ S_x^t + \Delta(i, x) + \alpha \Delta h(i, x) + \beta \Gamma_i \},$$

where S_i^t is the state of cell i at time t that is determined by the CA rule. α and β are the weighting factors for the height difference penalty and radar exposure penalty, respectively. m_{all} is the maximum allowable slope for the robot to climb on the terrain. Equation (6.3) governs the evolution of CA starting from the initial conditions in Section 6.2.1. The state value of all the cells within the domain of interest is updated in each iteration. The evolution stops when no further change is made in the state of the cells.

Once the automaton stops, the path finding algorithm will start from the starting cell and follow the minimum state value in the feasible neighbors to reach the goal. The algorithm will also record the length L_p , total elevation change H_p and total REP Γ_p along a path. The details of the algorithm is presented in Table 6.1.

In this study, the design parameters are the weighting factors α and β in the CA rule. The objectives are path length L_p , total elevation change H_p and total REP Γ_p along the path.

6.3 Case Studies

Three cases are studied to test the algorithm. Each case is done on a 100x100 grid of cells. We first started with a flat surface only containing radars and static obstacles. Then the complexity is increased by choosing a terrain with elevations that dissatisfy the slope constraint. Finally we consider a randomly generated map with mountains and valleys to simulate the real-world applications.

6.3.1 Obstacle/Radar Avoidance

The first terrain is a flat surface to verify obstacle and radar avoidance of the planner. Hence, β is the only design parameter for optimization. The results are shown in Figure 6.2. The contours show the radar range and its intensity on the terrain. The squares represent the obstacle cells. The blue circle and green triangle represent the start point and the goal, respectively. The found paths are shown in

red. The design parameter and the corresponding objective values of four examples of the optimal paths are presented in Table 6.2. It should be noted that the path marked by (a) in the figure has the least path length (L) and highest REP (Γ). This correlates with the minimum REP weighting factor β . On the other hand, path (d) has the longest path length and lowest REP because the REP weighting factor β is maximum in this case.

6.3.2 Climb Slope Constraint

The second case is designed specifically to test the climbing slope constraint. The terrain is made of steep areas as well as a set of ramps that can be followed to reach the goal. Figure 6.3 shows the terrain with colors representing the height. The maximum allowable slope m_{all} is chosen to be 0.2. Figure 6.3 shows the terrain and the optimal paths. The details for the two paths are presented in Table 6.3. It should be noted that the path marked by (a) has the shorter length and higher height change when the height difference penalty factor is less.

6.3.3 Randomized Terrain

The terrain in this case tries to mimic a mountainous environment. This map is created with a random algorithm named hill algorithm [?]. Several optimal paths are found and are shown in Figure 6.4. Table 6.4 shows the details of the optimal paths. The algorithm successfully returned six non-dominated possible paths.

6.4 Summary

A new multi-objective path planning method for 2D terrain considering the elevation was presented. As a result, the controller can choose the most suitable path among a set of paths for the application. A cellular automata rule was defined for the cells on the grid to find the most feasible path between the start point and goal. The method considers static obstacles and maximum climb slope constraints. Design parameters include the elevation change and radar avoidance weighting factors. The objectives of the optimization include the path length, total elevation change and the total radar exposure penalty. The algorithm was tested on terrains with different conditions and it was found that the method can successfully find the set of optimal paths without violating the constraints.

Table 6.2: Details of the optimal paths marked in Figure 6.2.

	L	Γ	β
a	12.37	11.33	0.035
b	13.09	8.158	0.304
c	15.58	3.172	0.449
d	16.10	0.181	4.975

Table 6.3: Details of the optimal paths marked in Figure 6.3.

	L	H	α
a	16.88	0.020	1.516
b	23.11	0.002	6.188

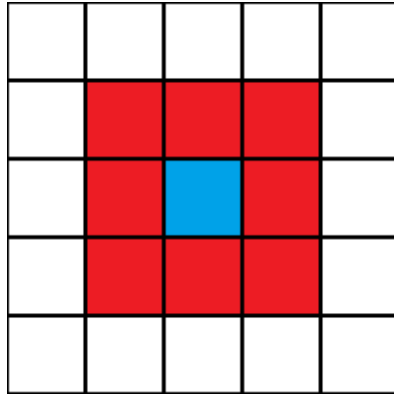


Figure 6.1: Moore neighbors of the blue cell are shown in red. They are the only cells considered in the CA rule for the center cell.

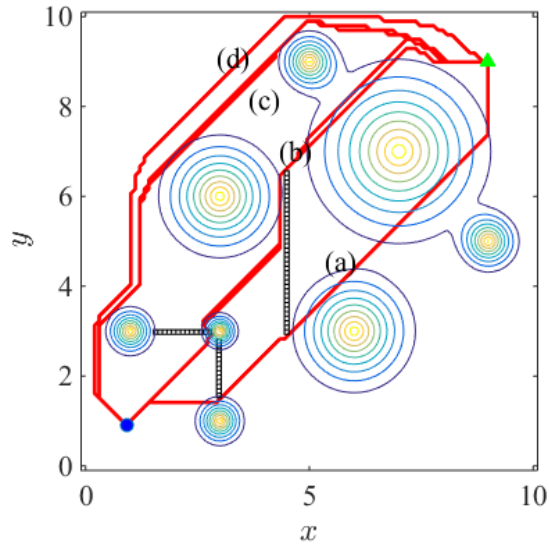


Figure 6.2: Case study 1 results. The blue circle is the start point and the green triangle shows the goal. The contours show the radars active range and its color corresponds to the radar penalty factor value. The black boxes represent the obstacles. Red lines show the feasible paths suggested by the algorithm.

Table 6.4: Details of the optimal paths marked in Figure 6.4.

	L	H	Γ	α	β
a	10.49	0.145	1.746	0.165	6.734
b	10.11	0.130	5.534	1.665	0.063
c	9.957	0.157	6.871	0.665	0.063
d	9.984	0.264	6.554	0	0
e	10.77	0.066	10.15	4.607	0.045
f	16.55	0.135	0.124	1.575	6.738

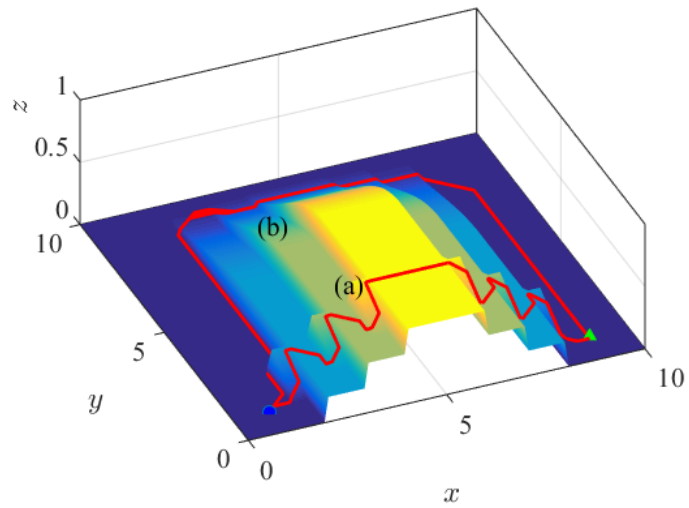


Figure 6.3: Case study 2 results. The terrain color is based on the elevation. Because choosing the shortest possible path would violate the slope constraint, the algorithm found a path through the series of ramps that is the shortest feasible path. The other found paths are longer but provide less elevation changes. Blue circle and green triangle are the start point and end point respectively.

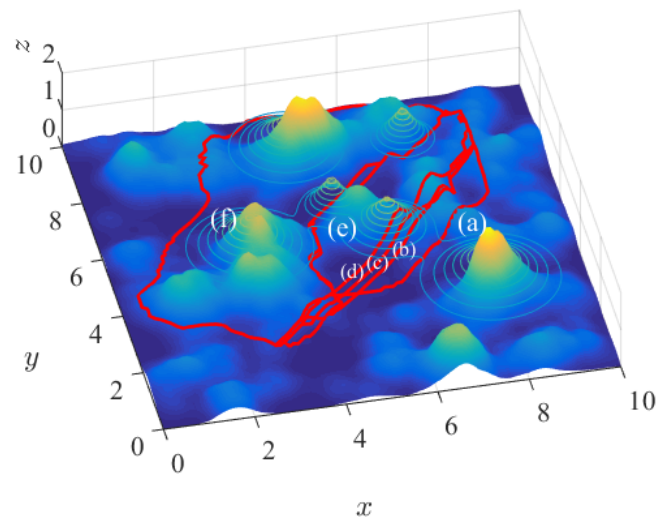


Figure 6.4: Case study 3 results. The terrain color is based on the elevation. The contours show the range of the radar.

Chapter 7

AIRFOIL OPTIMIZATION

7.1 Introduction

The airfoil/wing design is probably the most important part of an aircraft design. A practical aerodynamic design of airfoil requires optimal performance on a wide range of operating conditions. These requirements are often found to be conflicting and demand designer expertise for satisfactory results, not to mention the computational burden of the simulations. Although there exists many studies on direct and inverse design of airfoils, less attention has been paid to simultaneous consideration of multiple objectives.

Recent advances in computational power, optimization algorithms and reliable models has increased the simulation based design and optimization significantly, specially in the aerospace industry [123]. Despite the fact that these exists numerous studies on airfoil optimization, less attention has been paid for multi-objective airfoil design. It should be noted that the origination of evolutionary algorithms (EAs) played a key role in advancement of airfoil optimization and specially multi-objective airfoil design. The advantages of using EA includes robustness, handling nonlinearities and uncertainties, and most importantly being gradient free. In addition, EAs can successfully handle problems with high dimensional design space. On the other hand, they do not guarantee to find the global optima. All these features resulted in improvement of EAs and a significant increase in their applications specially in engineering [124].

Even though the improvement of the algorithms assisted with MOPs, computation burden of fluid simulation is still a big concern. MOPs usually require a significant number of objective evaluations. There exist numerous examples in the literature where many simplifications and assumptions has been made in order to effectively speedup the solution process [125–127]. In one of the early works [128], authors have demonstrated an inverse and direct airfoil design using multi-objective genetic algorithm. In [129], weighted sum method is used for a multi-objective design of single and multi-element airfoils. They both used b-Splines to express the geometry. Another work by Ray [130] utilizes a swarm algorithm for MOP while using PARSEC geometry representation and Euler code for flow simulation. The airfoil design can also be expanded into multi-objective design of 3D wings [131,132].

The goal of this chapter is to present a new direct approach in airfoil design considering multiple objectives and constraints simultaneously. The selection of objectives and constraints depend on the application considered such as a small single engine aircraft. The rest of this chapter is organized as follows. Section 7.2 discusses the analysis techniques used which include the parametrization method, MOP solver, problem setup and the post processing algorithm. The results are presented in Section 7.3 and the conclusion is stated in Section 4.7.

7.2 Analysis Method

This section will discuss the different steps taken in the optimization process.

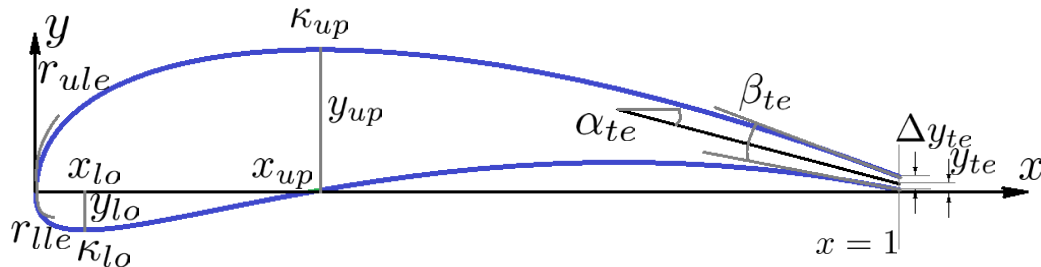
7.2.1 PARSEC parametrization

In order to reduce the degrees of freedom for a 2D cross section of airfoil, a parametrization scheme has to be deployed. The parametrization methods express the airfoil shape using some base functions [133,134]. In this work, PARSEC method with 12 parameters has been used [133,135]. The upper and lower surface of the airfoil can be obtained by

$$\begin{aligned} y_U &= \sum_{i=1}^6 a_{iU} x^{i-\frac{1}{2}} \\ y_L &= \sum_{i=1}^6 a_{iL} x^{i-\frac{1}{2}} \end{aligned} \quad (7.1)$$

where the coefficients $a_{1U}, \dots, a_{6U}, a_{1L}, \dots, a_{6L}$ are achieved by solving a linear system using the 12 physical parameters shown in Figure 7.1. Those physical parameters are considered as the variables for the optimizer. The description and bounds of the parameters are presented in Table 7.1.

Figure 7.1: Design parameters for PARSEC shape functions



7.2.2 Solver

After the parametrized airfoil has been regenerated, its aerodynamic performance needs to be evaluated. The objective values are then derived from the results. In this study, the aerodynamic performance of the airfoils has been calculated using Mark Drela's XFOIL code [136] which uses a linear-vorticity panel method for inviscid analysis coupled with an integral boundary-layer method for viscous analysis of subsonic airfoils [137]. XFOIL is accurate and efficient for specific ranges of applications and has been evaluated for over two decades [137]. For this study, the output of XFOIL includes the values of C_M , C_L and C_D for a range of attack angles (from zero with increments of 0.25 deg). The α for which C_L/C_D is maximum will be chosen as the operating angle of attack for that geometry and will be denoted as α_{opt} . The objectives will be reported at $\alpha = \alpha_{opt}$.

7.2.3 MOPSO

The solution methods for the MOPs can be classified into stochastic and deterministic approaches. Particle swarm optimization (PSO) is an EA which is inspired by a flock of birds flying and searching for food. There exist many successful attempts to have PSO solve MOPs [138, 139]. In this work we use a MOPSO based on [139] which is implemented in Fortran 90 [140].

For the current simulation the population is consist of 500 individuals and they have been evolved for 400 generations. The hypercube divisions on each objective is 50 and the archive can fit a maximum of 2000 individuals. To study the convergence progress and hyperspace analysis, the archive content has been recorded every 5 generations. The mutation rate is set to be 0.1.

7.2.4 Problem Setup

The airfoil MOP is designed as follows

$$\min\{f_1(\mathbf{x}), f_2(\mathbf{x}), f_3(\mathbf{x})\}, \quad (7.2)$$

where \mathbf{x} is the array of design variables which are the 12 PARSEC parameters expressed in Table 7.1. The objectives are

$$\begin{aligned} f_1(\mathbf{x}) &= -C_L \\ f_2(\mathbf{x}) &= C_D \\ f_3(\mathbf{x}) &= -\frac{dC_L}{d\alpha} \end{aligned} \quad (7.3)$$

subject to

$$\begin{aligned}
t_{\max} &< 25\% \\
C_M^2 &< 0.025 \\
C_L &> 0.5 \\
C_D &< 0.1 \\
\frac{dC_L}{d\alpha} &> 0.0
\end{aligned} \tag{7.4}$$

where a negative sign is used to transform the maximization objectives to a minimization. The objective f_1 correlates with maximizing the lift coefficient. The second objective is to minimize the drag coefficient. f_3 corresponds to maximization of the rate of change of C_L with respect to the attack angle. This objective delivers high lift at landing/take off because aircraft operates at higher angle of attack in those situations.

For evaluating the objectives for each airfoil, the attack angle α has been swept from zero with increments of 0.25 degrees to generate the highest $\frac{C_L}{C_D}$. The corresponding α can be considered as the optimal or operational condition of the airfoil. This angle will be denoted as α_{opt} in this paper.

This simulation tries to mimic the flight condition for small single engine aircrafts (Aircraft Approach Category 'A' [141]). For the simulations, Reynolds number is assumed to be $Re = 0.5 \times 10^6$, $Mach = 0.2$ and the turbulence level $N_{crit} = 9$.

7.2.5 Post-Processing

As we know, the solution of MOP is a set of individuals. All the points on the Pareto front are equally important. Even though the decision-maker has the authority to pick up any solution from this set, the selection process is not easy. To facilitate this operation, we introduce an algorithm that operates on the Pareto front. The algorithm starts by finding the ideal point made of the minimum values of all the objectives (in a minimization problem) of the Pareto front. Then, the L^2 -norm distance d between the ideal point and points on the Pareto front is calculated. The maximum distance d_{\max} and the minimum distance d_{\min} are found. The difference $d_{\max} - d_{\min}$ is divided by a pre-specified number $nDiv$ such that $h = (d_{\max} - d_{\min})/nDiv$. Circles centered at the ideal point with radii $r_i = d_{\min} + i * h$ for $i = 0, 2, \dots, nDiv$ are used to section the Pareto front and rank the control design by the radius r_i . This tool helps to extract a single point or a subset of the solution set. The point closest to the ideal point is called the "Knee point" of the solution.

7.3 Results

Two simulations were performed in this study. In the first run, the initial population is generated by the random sampling from the design space. The second

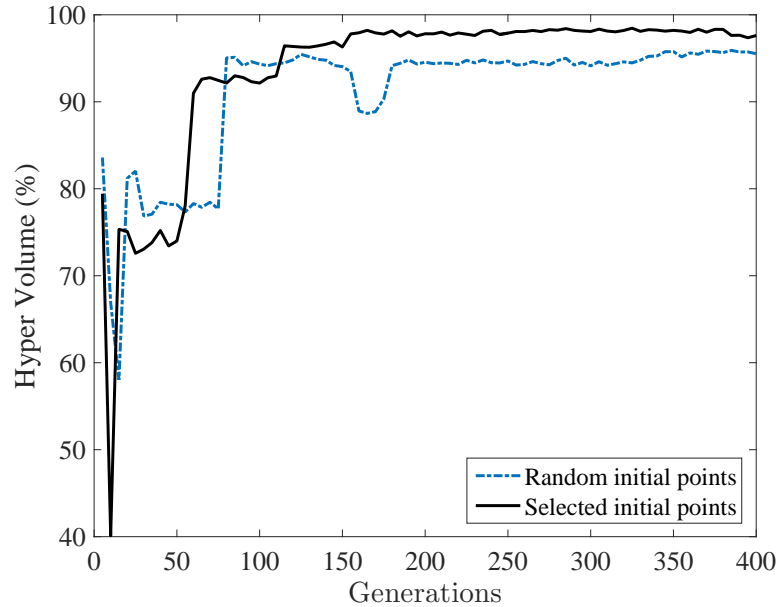


Figure 7.2: Hyper-volume convergence analysis of the solution. The case with selected initial points outperform random initial points as expected.

run however, used 36 exported airfoils from an already existing airfoil database. The first simulation took 6 hours and returned 140 points while the second simulation took 3.5 hours and returned 227 points. All programs ran on a single core of Intel Xeon E5-2680v3 CPU with clock speed of 2.5 GHz. The hyper-volume analysis for the convergence of the methods is presented in Figure 7.2. The same reference point is used for calculating the hyper-volume for both cases. As expected, the case with selected initial points outperform the case with random initial points. It is observed that no significant improvement is obtained after 200 generations. Using a database for airfoil optimization can be beneficial as behalf of the quality of the of initial population can be enhanced while keeping the diversity among them. From this point on, the results for the second run would be in the interest of this study since it expresses superior quality with the same effort.

Figure 7.3 shows the Pareto front of the solution set. The grey points show the 2D projection of the solution. Each point represent a specific geometry of an airfoil. Unlike the classical MOPs that usually have a $k - 1$ dimensional manifold solution, this problem returned 4 distinct lightly curved branches. The branches are named B1-B4 from the highest $\frac{dC_L}{d\alpha}$ to the lowest respectively for further analysis.

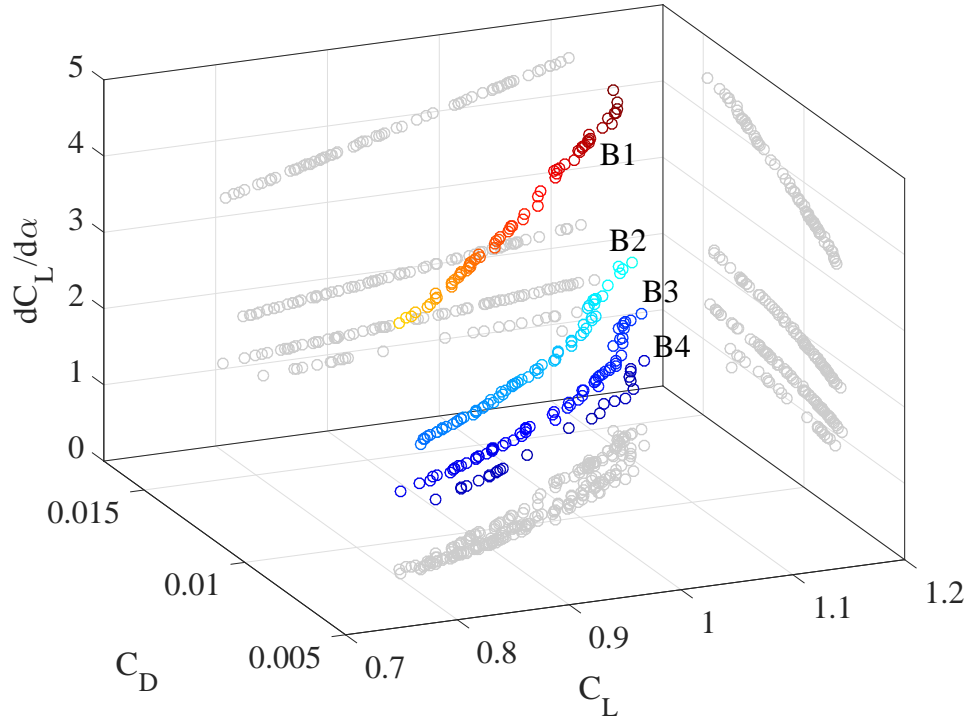


Figure 7.3: Pareto front of the airfoil MOP. the grey shadows show the 2D projection of the solution.

Although the individuals in the solution set are all non-dominated, the designer needs choose the most appropriate design. Eight cases have been chosen for further analysis from the 227 available points in the solution set. Three of them are the extreme cases for the objectives, corresponding to $\max(C_L)$, $\min(C_D)$ and $\max(\frac{dC_L}{d\alpha})$. These cases are theoretically equivalent to results from single objective optimization. The rest are chosen by the post processing algorithm using the whole solution set (one case) and each branch separately (4 cases). The objectives and corresponding PARSEC parameters for these four solution points are presented in Table 7.2.

For studying the characteristics of the branches B1-B4, their parameter variations are plotted along with their error bars in Figure 7.4. The error bars show the confidence interval for each of the branches. In order to quantitatively evaluate the differences between the branches, an independent sample t-test was performed among the branches. The results are presented in Table 7.3. The boxed entries show a significant difference between the two branches. It is observed that there

are distinct variations in r_{ule} , r_{lle} , x_{up} and y_{te} . Having higher r_{ule} , r_{lle} and y_{te} , and a lower x_{up} would result in having a relatively larger $\frac{dC_L}{d\alpha}$. As seen in Figure 7.3, choosing an airfoil with higher $\frac{dC_L}{d\alpha}$ would result in slightly poorer quality C_L and C_D objectives. However, the decision maker might be interested to choose an airfoil from branch B1 since the gain in $\frac{dC_L}{d\alpha}$ is significant compared to the deterioration of the other two objectives.

Geometry of the selected cases are presented in Figures 7.5 and 7.6. The airfoil shapes in 7.5 shows more diversity since they tend to extremize each objective independently. It is observed that the thinnest airfoil correspond to $\min(C_D)$ case as expected. The polar plots of these airfoils can be found in Figure 7.7 and 7.8. The optimal operational point α_{opt} is shown with a marker on each curve. In Figure 7.7, the α_{opt} ranges between 2 and 4 degrees while in Figure 7.8, they vary in a wider range of angles.

Although the polar plots provide valuable information about the properties of the airfoils, the designer might be interested in the pressure distribution on the candidate airfoils at their operational angle of attack. Figures 7.9 and 7.10 show the C_P curves for the selected geometries. As expected, the area under the curve in Figure 7.9 is maximum and minimum for the $\max(C_L)$ and $\min(C_D)$ cases respectively. The separation of the boundary layer is observed to happen in relative length 0.4 to 0.6 of the chord for all cases.

7.4 Summary

Design for application has always been a challenge for engineers since it deals with multiple design parameter and objectives to be met. A multi-objective airfoil design procedure was presented in this chapter. PARSEC parametrization method was used to express the geometry in terms of 12 variables. The aerodynamic properties of the airfoil was extracted by panel method using XFOIL code. The three objectives considered were maximization of lift coefficient ($\max(C_l)$), minimization of drag coefficient ($\min(C_d)$), and maximization of rate of change of lift to attack angle ($dC_l/d\alpha$). Multi-objective particle swarm optimization algorithm was utilized for solving the MOP. Using a database of already existing airfoils helped to enhance the quality of the solution.

The method successfully converged and found 227 feasible airfoil configurations in a reasonable time (3.5 hrs). A post processing algorithm was utilized to extract and rank the solutions in order to help designer with the decision making process. The tools used in this chapter can be easily adopted to solve other multi-disciplinary airfoil design problems by simply updating the objectives and constraints.

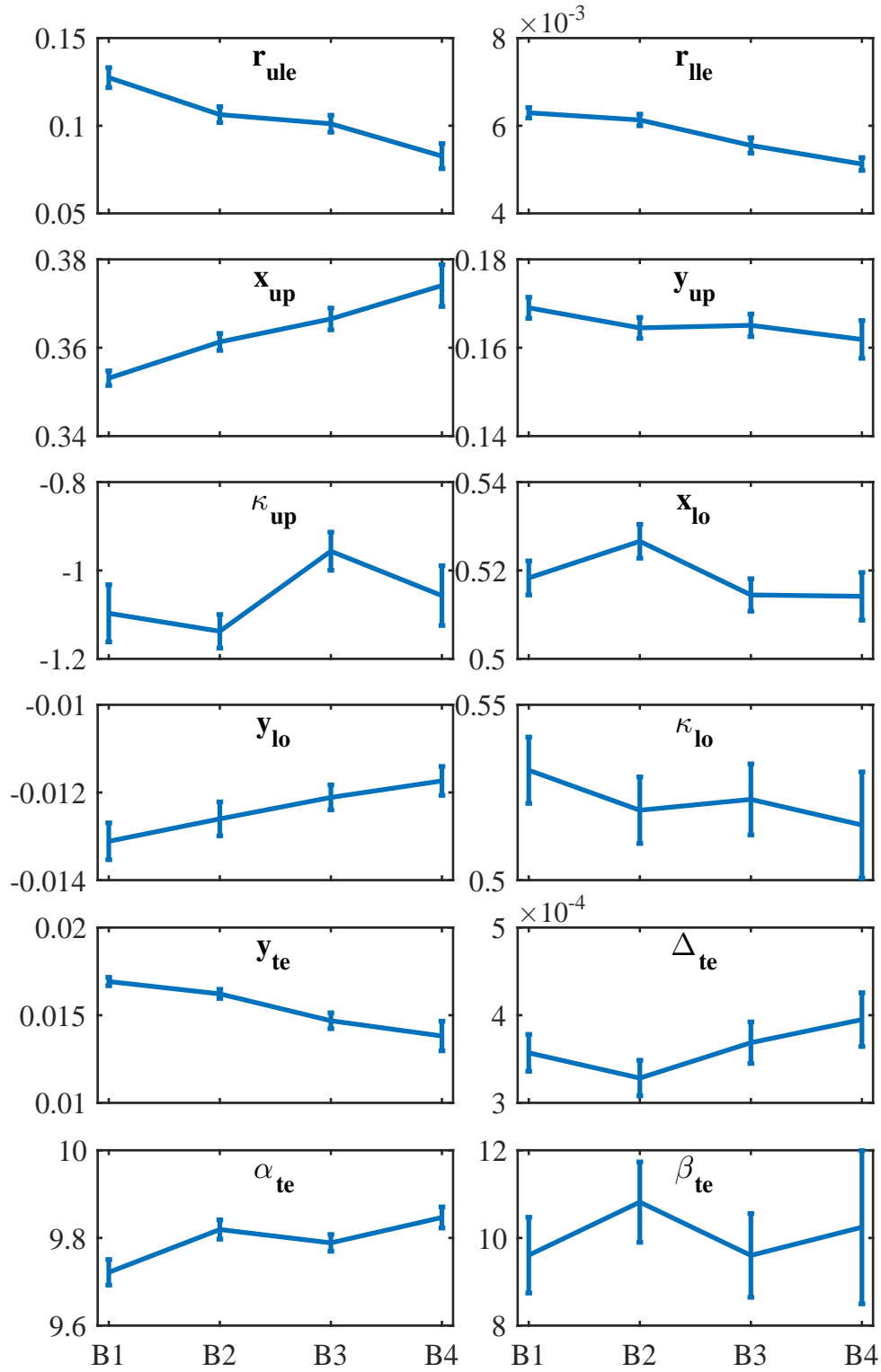


Figure 7.4: Variation of the design parameters between the branches B1-B4. The mean of the branch parameters are connected with a line while the error bar shows the confidence interval for the variations of the parameter. These results are quantitatively studies in Table 7.3

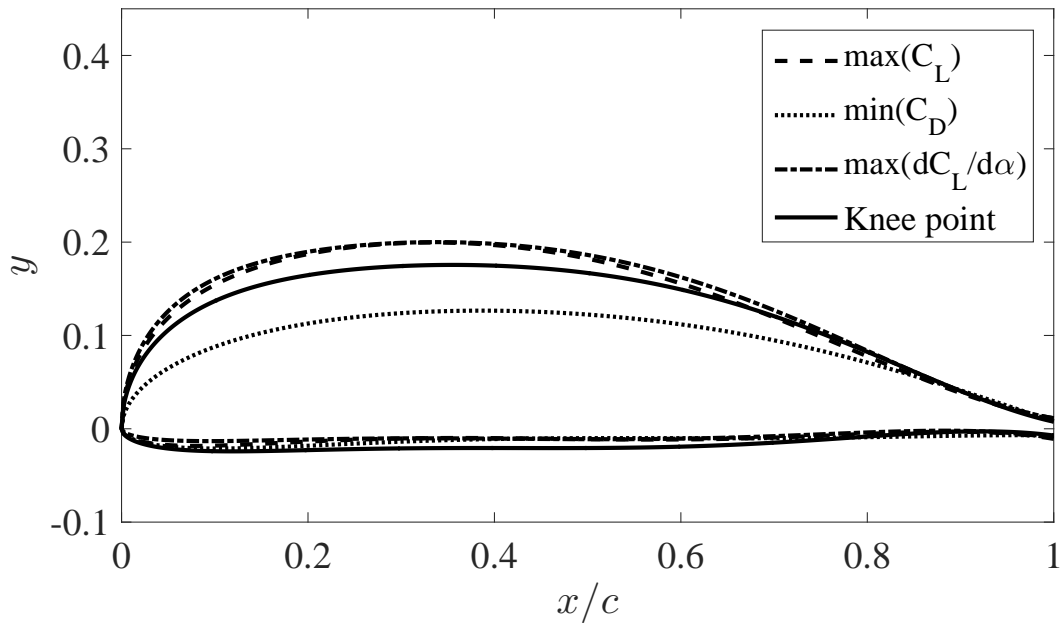


Figure 7.5: Four airfoils from the solution set. The first three correspond to the extreme objectives and the last case is the knee point selected by the post-processing algorithm. The objective and design parameter values can be found at Table 7.2.

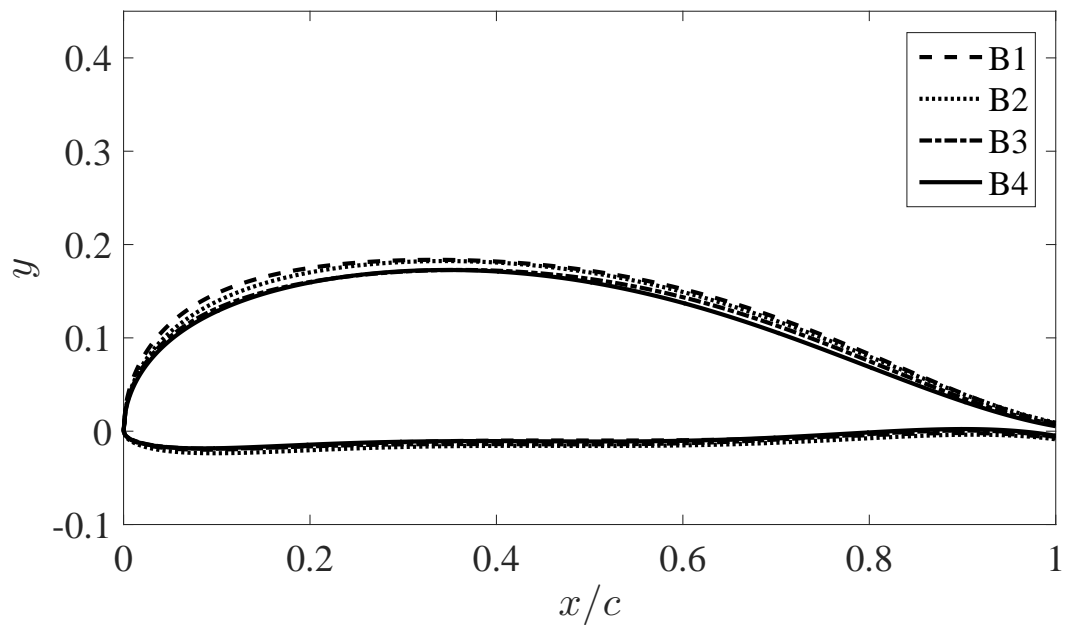


Figure 7.6: Four airfoils each representing the knee point for every brach B1-B4. The objective and design parameter values can be found at Table 7.2.

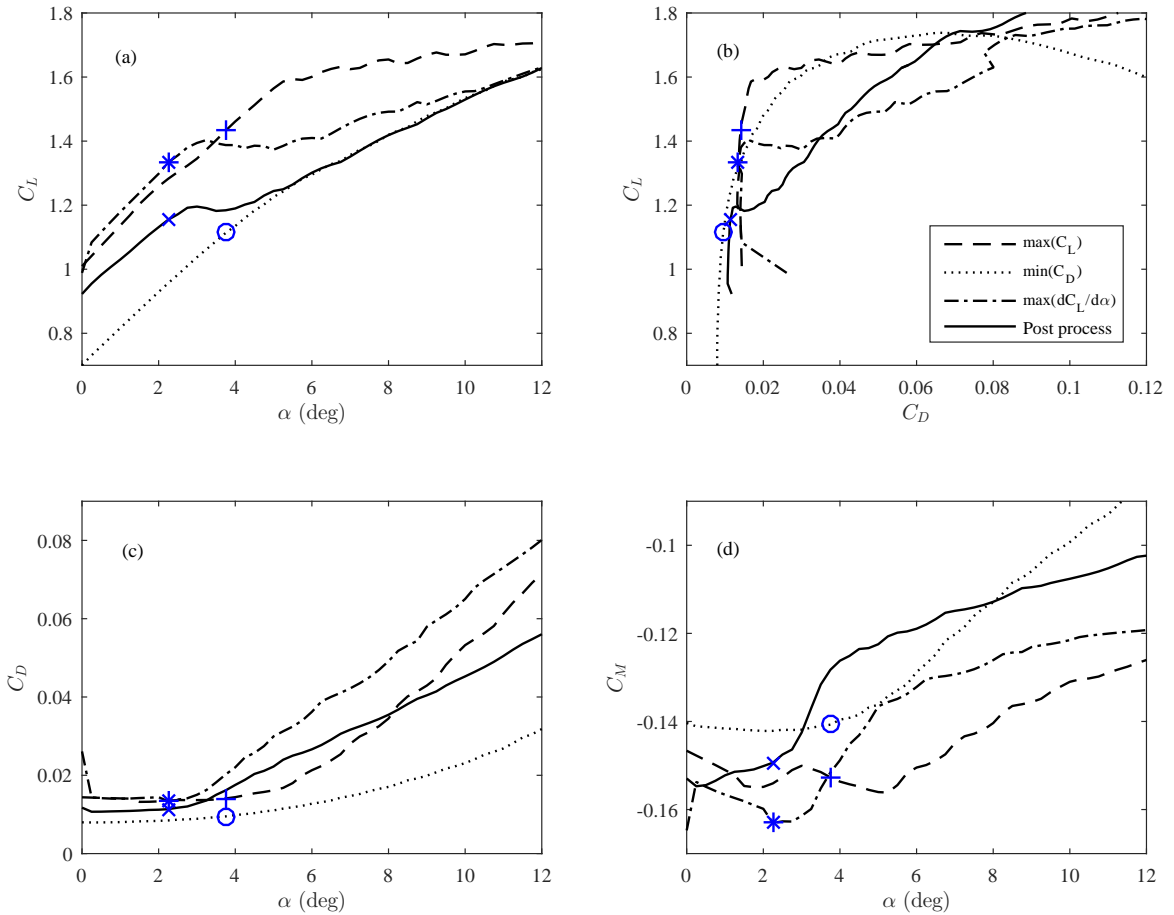


Figure 7.7: The aerodynamic properties (polars) of the the first four selected airfoils from Table 7.2. The marked points on each curve shows the optimal operational point. This point corresponds to the attack angle where $\max(C_L/C_D)$ is observed.

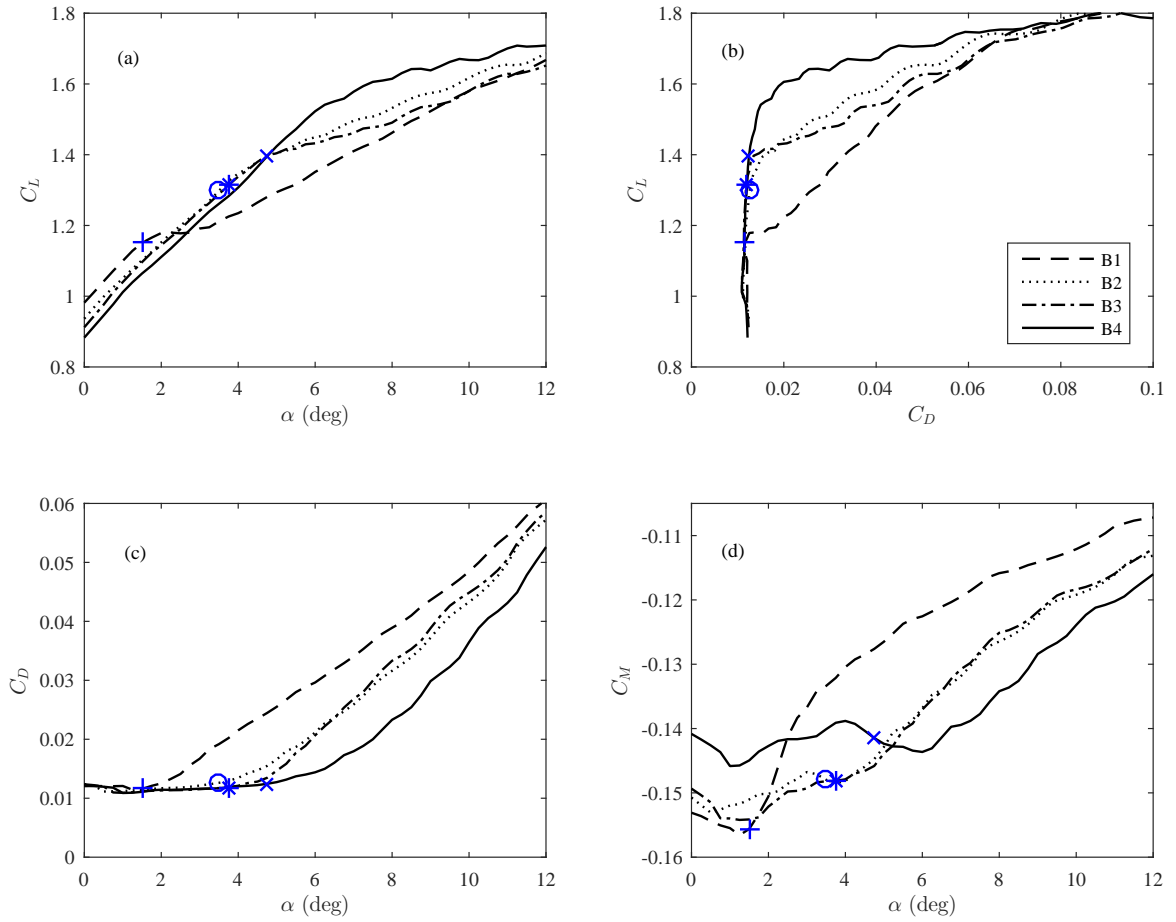


Figure 7.8: The aerodynamic properties (polars) of the the second four selected airfoils from Table 7.2. The marked points on each curve shows the optimal operational point. This point corresponds to the attack angle where $\max(C_L/C_D)$ is observed.

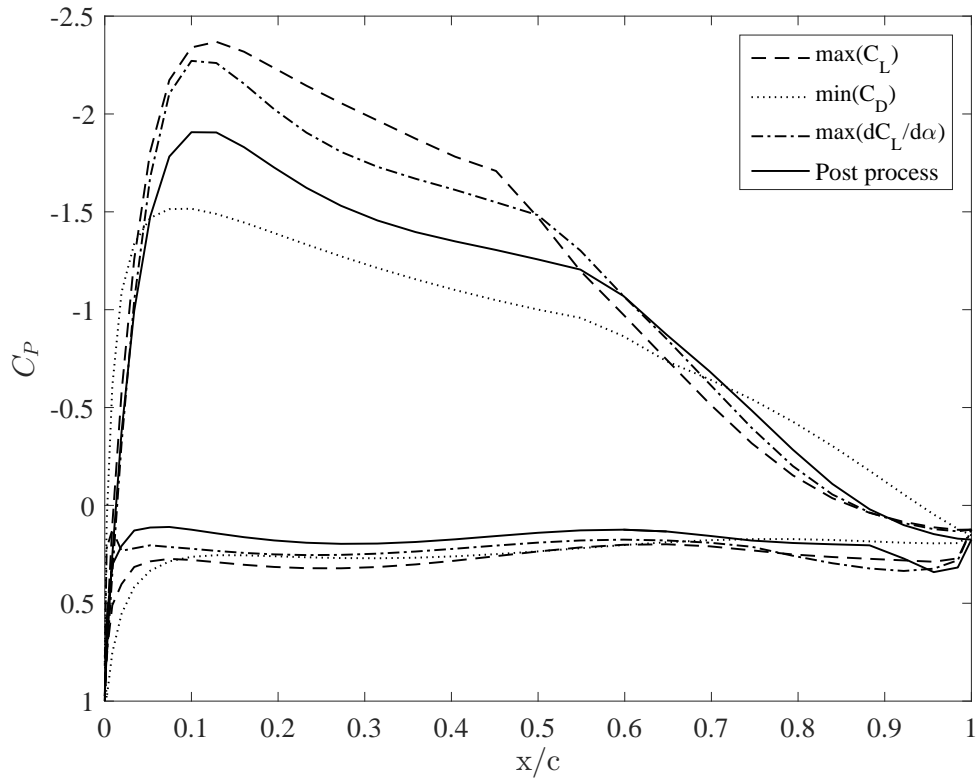


Figure 7.9: The pressur distribution of the the first four selected airfoils from Table 7.2. The curves are calculated at the operational angel of attack for each airfoil as shown in Figure 7.7.

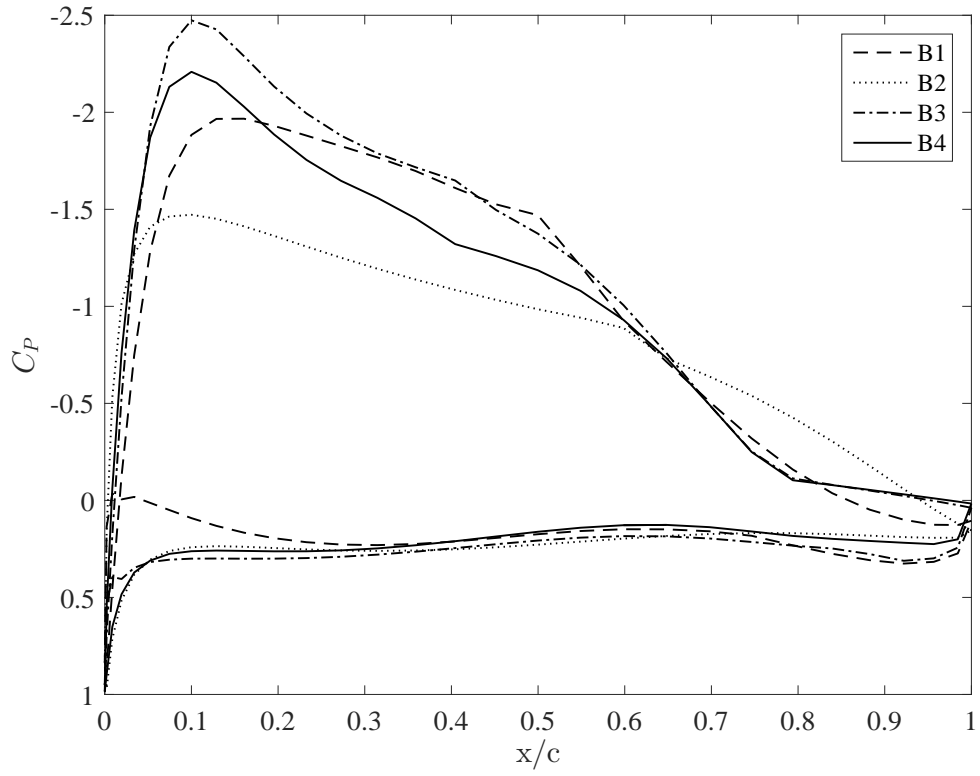


Figure 7.10: The pressur distribution of the the second four selected airfoils from Table 7.2. The curves are calculated at the operational angel of attack for each airfoil as shown in Figure 7.8.

Table 7.1: Physical parameters used in the PARSEC parametrization and their bounds.

Parameter	Description	min	max
r_{ule}	upper leading edge radius	0.0001	0.2
r_{lle}	lower leading edge radius	0.0001	0.2
x_{up}	upper crest x coord.	0.1	0.6
y_{up}	upper crest y coord.	0.01	0.2
κ_{up}	upper crest curvature	-2.0	-0.001
x_{lo}	lower crest x coord.	0.1	0.6
y_{lo}	lower crest y coord.	-2.0	-0.01
κ_{lo}	lower crest curvature	0.001	2.0
α_{te}	trailing edge direction	0.0°	10.0°
β_{te}	trailing edge wedge angle	0.0°	20.0°
y_{te}	trailing edge y coord.	0.0	0.1
Δy_{te}	trailing edge thickness	0.0	0.1

Table 7.2: Selected airfoils from the solution set with corresponding objective and parameter values. The first three entries are extreme cases and the last one is selected by the postprocessing algorithm.

Case	$\max(C_L)$	$\min(C_D)$	$\max(\frac{dC_L}{d\alpha})$	Knee point	B1	B2	B3	B4
C_L	1.1385E+00	8.0598E-01	1.1156E+00	9.8040E-01	1.0385E+00	1.0271E+00	1.0400E+00	1.0400E+00
C_D	1.4640E-02	8.1323E-03	1.4800E-02	1.1039E-02	1.2434E-02	1.1461E-02	1.1311E-02	1.1272E-02
$\frac{dC_L}{d\alpha}$	1.5198E+00	1.0746E+00	4.4625E+00	3.9216E+00	4.1539E+00	2.0542E+00	1.3867E+00	1.0400E+00
r_{ule}	1.7099E-01	4.5428E-02	2.0000E-01	1.3442E-01	1.6353E-01	1.3234E-01	1.2339E-01	1.1041E-01
r_{lle}	5.5677E-03	5.7194E-03	2.5511E-03	7.0143E-03	5.8642E-03	7.7143E-03	5.0299E-03	5.3635E-03
x_{up}	3.3394E-01	3.8661E-01	3.3866E-01	3.5544E-01	3.3379E-01	3.4418E-01	3.6044E-01	3.4709E-01
y_{up}	1.9957E-01	1.2656E-01	2.0000E-01	1.7557E-01	1.8362E-01	1.8231E-01	1.7275E-01	1.7272E-01
κ_{up}	-3.8567E-01	-1.1579E+00	-2.4640E-01	-1.2130E+00	-8.8642E-01	-1.1283E+00	-7.4765E-01	-8.0087E-01
x_{lo}	5.6996E-01	5.3417E-01	5.1880E-01	4.6341E-01	5.2154E-01	5.0872E-01	4.9008E-01	5.1205E-01
y_{lo}	-1.1548E-02	-1.0000E-02	-1.1173E-02	-2.0691E-02	-1.0000E-02	-1.5799E-02	-1.4098E-02	-1.1003E-02
κ_{lo}	5.4729E-01	2.9032E-01	5.0013E-01	4.7175E-01	5.0638E-01	4.8766E-01	5.0034E-01	5.4477E-01
Δy_{te}	2.2129E-02	1.4961E-02	2.1290E-02	1.4809E-02	1.5903E-02	1.7843E-02	1.3520E-02	1.0249E-02
y_{te}	2.5880E-04	3.7191E-04	6.3805E-04	1.6348E-04	5.2449E-04	2.1251E-04	3.6694E-04	2.2839E-04
α_{te}	9.5283E+00	9.7724E+00	9.2389E+00	9.9435E+00	9.5185E+00	9.6986E+00	9.7650E+00	9.8014E+00
β_{te}	0.0000E+00	1.7740E+01	0.0000E+00	5.0011E+00	4.4756E+00	4.5612E+00	3.1408E-01	2.6895E-01

Table 7.3: The results for the statistical comparison between the branches B1-B4. Each of the 12 parameters has been compared between the branches using the independent sample ttest. P values under 0.05 show a significant difference between the groups.

Parameter	B1,B2	B1,B3	B1,B4	B2,B3	B2,B4	B3,B4
r_{ule}	$4.35e-3$	$5.97e-4$	$4.60e-5$	$4.31e-1$	$8.00e-3$	$4.65e-2$
r_{lle}	$3.60e-1$	$6.73e-4$	$5.62e-7$	$9.33-3$	$6.17e-5$	$1.73e-1$
x_{up}	$1.58e-3$	$1.45e-5$	$8.69e-7$	$9.71e-2$	$3.68e-3$	$1.33e-1$
y_{up}	$1.78e-1$	$2.59e-1$	$1.33e-1$	$8.67e-1$	$5.81e-1$	$5.22e-1$
κ_{up}	$5.86e-1$	$7.22e-2$	$7.29e-1$	$1.98e-3$	$2.88e-1$	$2.27e-1$
x_{lo}	$1.29e-1$	$4.71e-1$	$5.64e-1$	$2.39e-2$	$8.73e-2$	$9.64e-1$
y_{lo}	$3.71e-1$	$5.02e-2$	$6.10e-2$	$3.12e-1$	$2.07e-1$	$4.67e-1$
κ_{lo}	$3.96e-1$	$5.50e-1$	$3.92e-1$	$8.24e-1$	$8.19e-1$	$7.04e-1$
α_{te}	$8.14e-3$	$5.50e-2$	$1.52e-2$	$2.98e-1$	$4.91e-1$	$1.01e-1$
β_{te}	$3.39e-1$	$9.96e-1$	$7.21e-1$	$3.60e-1$	$7.58e-1$	$7.37e-1$
y_{te}	$4.75e-2$	$2.64e-5$	$5.37e-6$	$3.76e-3$	$4.68e-4$	$3.40e-1$
Δy_{te}	$3.26e-1$	$7.16e-1$	$3.40e-1$	$1.97e-1$	$8.82e-2$	$5.48e-1$

Chapter 8

CONCLUSION AND FUTURE WORK

The main purpose of this thesis has been to propose a new method for solving MOPs. The focus of this work is to develop methods that are most suitable for engineering applications. The hybrid algorithm combines evolutionary algorithm with a modified simple cell mapping technique with subdivision to solve MOPs. The hybrid method was investigated and compared for a set of benchmark problems. This thesis also aims to study and introduce new applications of MOPs in different areas of engineering.

The simple cell mapping method was used as a tool to solve MOPs. The method has been applied to control design MOPs for a first order system with time delay and a second order linear oscillator. Subdivision is used to enhance the quality of the solution and to see the fine structures in the Pareto set/front. This method could be fast and efficient for low dimensional problems and most importantly, it finds the global solution in the domain of interest. However, this method slows down exponentially for higher dimensional problems.

Further, by studying the properties of EAs and SCM method for solving MOPs, their strength and weakness were characterized. The EA starts converging fast and slows down as it gets to the vicinity of the solution. For that reason, the hybrid methods starts with EA to reach the neighborhood of the solution. From that point, the modified SCM method will take over the process. The drawback of the classical SCM method is that it requires to sweep the parameter space which could be a computationally expensive process. In the modified SCM, the explore algorithm starts with the covering set of the cells and evaluate them by comparing their objectives with neighbors and recover the missed cells. Applying subdivision at the end of the process and exploring again result in a higher quality solution with minimal computations. This method is compared with the classical SCM method and NSGA-II over a few mathematical benchmarking problems and a nonlinear control system with time delay. The results prove that the hybrid method outperforms the other two methods in most of the cases.

In addition to MOP control design for linear and nonlinear systems with or without time delay, this thesis looks into other engineering applications. The concept of multi-objective optimization is expanded into fractional order systems by designing a distributed order fractional damping for a second order system.

We further looked into robot path planning application. A cellular automaton platform was used to design robot paths while avoiding radars, saving energy, and looking for the shortest path considering certain constraints for the robot.

The last application provided is the airfoil design. Three main performance indices are defined as objectives and optimized while the geometry is expressed with 12 parameters. Panel method was used to evaluate airfoil properties.

MOPs have important applications in engineering and design. Lack of suitable tools cause scientists and engineers to avoid dealing with such problems. By having a proper understanding of MOPs and availability of efficient and accurate algorithms, we can conveniently solve the MOP of interest. As a result, we gain the ability to choose from a range of optimal designs instead of a single one.

There are many ways for this research to be continued. A parallel version for the hybrid algorithm is developed which is suitable for handling high dimensional MOPs. The next step would be validating and comparing the results for the new code.

Another important step would be preparing a MOP toolbox or library for engineering applications. Several solution methods can be implemented in the program. Having parallel computing options will help the users to accelerate the solution of time consuming or high dimensional MOPs.

There are also potentials for more engineering applications. For instance, in the airfoil design problem, we can consider the 3D wing by looking at multiple 2D sections through the wing. This will result in a higher dimension in parameter space. In the robot path planning case, one can include the robot dynamics in the simulations and add the travel time in the objectives. For the control problems, one can consider adding nonlinearity/noise/uncertainties (to mimic real systems) to the models and try the MOP design of some state of the art controllers such as sliding mode control or adaptive control.

BIBLIOGRAPHY

- [1] C. A. Coello Coello, G. B. Lamont, D. A. Veldhuizen, *Evolutionary Algorithms for Solving Multi-Objective Problems*, Springer, New York, 2007.
- [2] K. F. Man, K. S. Tang, S. Kwong, Genetic algorithms: concepts and applications, *IEEE Transactions on Industrial Electronics* 43 (5) (1996) 519–534.
- [3] M. S. Arumugam, M. V. C. Rao, R. Palaniappan, New hybrid genetic operators for real coded genetic algorithm to compute optimal control of a class of hybrid systems, *Applied Soft Computing* 6 (1) (2005) 38–52.
- [4] X. Shuai, X. Zhou, A genetic algorithm based on combination operators, *Procedia Environmental Sciences* 11, Part A (2011) 346–350.
- [5] S. M. Elsayed, R. A. Sarker, D. L. Essam, Multi-operator based evolutionary algorithms for solving constrained optimization problems, *Computers & Operations Research* 38 (12) (2011) 1877–1896.
- [6] P. Larranaga, C. M. H. Kuijpers, R. H. Murga, I. Inza, S. Dizdarevic, Genetic algorithms for the travelling salesman problem: A review of representations and operators, *Artificial Intelligence Review* 13 (2) (1999) 129–170.
- [7] J. R. Zeidi, N. Javadian, R. Tavakkoli-Moghaddam, F. Jolai, A hybrid multi-objective approach based on the genetic algorithm and neural network to design an incremental cellular manufacturing system, *Computers & Industrial Engineering* 66 (4) (2013) 1004–1014.
- [8] N. F. Wang, X. M. Zhang, Y. W. Yang, A hybrid genetic algorithm for constrained multi-objective optimization under uncertainty and target matching problems, *Applied Soft Computing* 13 (8) (2013) 3636–3645.
- [9] C. Hernández, Y. Naranjani, Y. Sardahi, W. Liang, O. Schütze, J.-Q. Sun, Simple cell mapping method for multiobjective optimal PID control design, *International Journal of Dynamics and Control* 1 (3) (2013) DOI 10.1007/s40435-013-0021-1.

- [10] C. S. Hsu, A discrete method of optimal control based upon the cell state space concept, *Journal of Optimization Theory and Applications* 46 (4) (1985) 547–569.
- [11] Y. Naranjani, C. Hernández, F.-R. Xiong, O. Schütze, J.-Q. Sun, A hybrid algorithm for the simple cell mapping method in multi-objective optimization, in: M. Emmerich, A. Deutz, O. Schütze, T. Bck, E. Tantar, A.-A. Tantar, P. D. Moral, P. Legrand, P. Bouvry, C. A. C. Coello (Eds.), *EVOLVE - A Bridge between Probability, Set Oriented Numerics, and Evolutionary Computation IV*, Vol. 227 of *Advances in Intelligent Systems and Computing*, Springer International Publishing, 2013, pp. 207–223.
- [12] Y. Naranjani, Y. Sardahi, J.-Q. Sun, C. Hernández, O. Schütze, Fine structure of pareto front of multi-objective optimal feedback control design, in: *Proceedings of ASME 2013 Dynamic Systems and Control Conference*, 2013, pp. V001T15A009–V001T15A009.
- [13] M. Dellnitz, A. Hohmann, A subdivision algorithm for the computation of unstable manifolds and global attractors, *Numerische Mathematik* 75 (3) (1997) 293–317.
- [14] M. Dellnitz, O. Junge, An adaptive subdivision technique for the approximation of attractors and invariant measures, *Computing and Visualization in Science* 1 (2) (1998) 63–68.
- [15] M. Dellnitz, O. Junge, Set oriented numerical methods for dynamical systems, *Handbook of Dynamical Systems* 2 (2002) 221–264.
- [16] V. Pareto, *Manual of Political Economy*, The MacMillan Press, London, 1971 (original edition in French in 1927).
- [17] K. M. Miettinen, *Nonlinear Multiobjective Optimization*, Kluwer Academic Publishers, 1999.
- [18] Y. Y. Haimes, L. S. Lasdon, D. A. Wismer, On a bicriterion formulation of the problems of integrated system identification and system optimization, *IEEE Transactions on Systems, Man, and Cybernetics SMC-1* (3) (1971) 296–297.
- [19] I. Das, J. Dennis, Normal-boundary intersection: A new method for generating the Pareto surface in nonlinear multicriteria optimization problems, *SIAM Journal of Optimization* 8 (1998) 631–657.
- [20] C. Hillermeier, *Nonlinear Multiobjective Optimization - A Generalized Homotopy Approach*, Birkhäuser, Berlin, 2001.

- [21] E. L. Allgower, K. Georg, *Numerical Continuation Methods*, Springer, New York, 1990.
- [22] S. Schäffler, R. Schultz, K. Weinzierl, Stochastic method for the solution of unconstrained vector optimization problems, *Journal of Optimization Theory and Applications* 114 (1) (2002) 209–222.
- [23] J. Fliege, B. F. Svaiter, Steepest descent methods for multicriteria optimization, *Mathematical Methods of Operations Research* 51 (3) (2000) 479–494.
- [24] C. S. Hsu, A theory of cell-to-cell mapping dynamical systems, *Journal of Applied Mechanics* 47 (1980) 931–939.
- [25] C. S. Hsu, *Cell-to-Cell Mapping, A Method of Global Analysis for Nonlinear Systems*, Springer-Verlag, New York, 1987.
- [26] K. Deb, *Multi-Objective Optimization Using Evolutionary Algorithms*, John Wiley & Sons, Inc., 2001.
- [27] Q. Zhang, H. Li, MOEA/D: A multi-objective evolutionary algorithm based on decomposition, *IEEE Transactions on Evolutionary Computation* 11 (6) (2007) 712–731.
- [28] N. Beume, B. Naujoks, M. Emmerich, SMS-EMOA: Multiobjective selection based on dominated hypervolume, *European Journal of Operational Research* 181 (3) (2007) 1653–1669.
- [29] O. Schütze, M. Laumanns, C. A. C. Coello, M. Dellnitz, E.-G. Talbi, Convergence of stochastic search algorithms to finite size Pareto set approximations, *Journal of Global Optimization* 41 (4) (2008) 559–577.
- [30] O. Schütze, M. Laumanns, E. Tantar, C. A. C. Coello, E.-G. Talbi, Computing gap free Pareto front approximations with stochastic search algorithms, in: *Proceedings of Proceedings of the 2007 Conference on Evolutionary Computation*, 2007.
- [31] J. D. Knowles, D. W. Corne, M-PAES: a memetic algorithm for multiobjective optimization, in: *Proceedings of Proceedings of the IEEE Congress on Evolutionary Computation*. Piscataway, New Jersey, 2000, pp. 325–332.
- [32] H. Ishibuchi, T. Yoshida, T. Murata, Balance between genetic search and local search in memetic algorithms for multiobjective permutation flowshop scheduling, *IEEE Transactions on Evolutionary Computation* 7 (2) (2003) 204–223.

- [33] A. Lara, G. Sanchez, C. A. Coello Coello, O. Schütze, HCS: A new local search strategy for memetic multiobjective evolutionary algorithms, *IEEE Transactions on Evolutionary Computation* 14 (1) (2010) 112–132.
- [34] M. Vasile, F. Zuiani, Multi-agent collaborative search: an agent-based memetic multi-objective optimization algorithm applied to space trajectory design, *Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering* 225 (11) (2011) 1211–1227.
- [35] M. Dellnitz, O. Schütze, T. Hestermeyer, Covering Pareto sets by multi-level subdivision techniques, *Journal of Optimization Theory and Applications* 124 (1) (2005) 113–136.
- [36] O. Schütze, A. Dell’Aere, M. Dellnitz, On continuation methods for the numerical treatment of multi-objective optimization problems, in: *Proceedings of Practical Approaches to Multi-Objective Optimization*, Dagstuhl Seminar, 2005.
- [37] K. Ganapathy, J. Jerome, Control of dead-time systems using derivative free local search guided population based incremental learning algorithms, *Optimization and Engineering* 15 (2) (2012) 331–354.
- [38] M. Gobbi, A k, k - ϵ optimality selection based multi objective genetic algorithm with applications to vehicle engineering, *Optimization and Engineering* 14 (2) (2012) 345–360.
- [39] A. P. C. Cuco, F. L. Sousa, V. V. Vlassov, A. J. e. Silva Neto, Multi-objective design optimization of a new space radiator, *Optimization and Engineering* 12 (3) (2011) 393–406.
- [40] B. DuPont, J. Cagan, A hybrid extended pattern search/genetic algorithm for multi-stage wind farm optimization, *Optimization and Engineering* (201) 1–27.
- [41] M. G. Villarreal-Cervantes, C. A. Cruz-Villar, J. Alvarez-Gallegos, Synergetic structure–control design via a hybrid gradient-evolutionary algorithm, *Optimization and Engineering* 16 (3) (2014) 511–539.
- [42] F. Li, T. Wu, M. Hu, Design of a decentralized framework for collaborative product design using memetic algorithms, *Optimization and Engineering* 15 (3) (2013) 657–676.
- [43] G. Lombardi, G. Mengali, F. Beux, A hybrid genetic based optimization procedure for aircraft conceptual analysis, *Optimization and Engineering* 7 (2) (2006) 151–171.

- [44] D. Greenhalgh, S. Marshall, Convergence criteria for genetic algorithms, *SIAM Journal on Computing* 30 (1) (2000) 269–282.
- [45] K. Deb, A. Pratap, S. Agarwal, T. Meyarivan, A fast and elitist multiobjective genetic algorithm: Nsga-ii, *Evolutionary Computation, IEEE Transactions on* 6 (2) (2002) 182–197.
- [46] O. Schütze, S. Mostaghim, M. Dellnitz, J. Teich, Covering pareto sets by multilevel evolutionary subdivision techniques, in: *Proceedings of Evolutionary Multi-Criterion Optimization. Second International Conference, EMO 2003*, Berlin, Germany, 2003, pp. 118–32.
- [47] T. Bäck, *Evolutionary Algorithms in Theory and Practice*, Oxford University Press, New York, 1996.
- [48] H.-P. P. Schwefel, *Evolution and Optimum Seeking: The Sixth Generation*, John Wiley & Sons, Inc., 1993.
- [49] J. Q. Sun, A method of continuous time approximation of delayed dynamical systems, *Communications in Nonlinear Science and Numerical Simulation* 14 (4) (2008) 998–1007.
- [50] B. Song, J. Q. Sun, Lowpass filter-based continuous-time approximation of delayed dynamical systems, *Journal of Vibration and Control* 17 (8) (2011) 1173–1183.
- [51] T. Insperger, G. Stepan, *Semi-discretization for time-delay systems*, Springer, New York, 2011.
- [52] D. B. Thomas, L. Howes, W. Luk, A comparison of CPUs, GPUs, FPGAs, and massively parallel processor arrays for random number generation, *Proceedings of the ACM/SIGDA international symposium on Field programmable gate arrays* (2009) 63–72.
- [53] J. Jaros, P. Pospichal, A fair comparison of modern CPUs and GPUs running the genetic algorithm under the knapsack benchmark, in: *Proceedings of European Conference on the Applications of Evolutionary Computation, 2012*, pp. 426–435.
- [54] S. Han, On fair comparison between CPU and GPU (2013) <http://people.eecs.berkeley.edu/~sangjin/2013/02/12/CPU--GPU--comparison.html>.
- [55] G. P. Liu, S. Daley, Optimal-tuning nonlinear PID control of hydraulic systems, *Control Engineering Practice* 8 (9) (2000) 1045–1053.

- [56] D. H. Kim, Tuning of PID controller using gain/phase margin and immune algorithm, in: Proceedings of the 2005 IEEE Mid-Summer Workshop on Soft Computing in Industrial Applications, 2005, pp. 69–74.
- [57] A. Gambier, E. Badreddin, Multi-objective optimal control: An overview, in: Proceedings of the 16th IEEE International Conference on Control Applications, Singapore, 2007, pp. 170–175.
- [58] M. Rani, H. Selamat, H. Zamzuri, Z. Ibrahim, Multi-objective optimization for PID controller tuning using the global ranking genetic algorithm, *International Journal of Innovative Computing, Information and Control* 8 (1) (2012) 269–284.
- [59] M. Khoie, A. Sedigh, K. Salahshoor, PID controller tuning using multi-objective optimization based on fused genetic-immune algorithm and immune feedback mechanism, *International Conference on Mechatronics and Automation* (2011) 2459–2464.
- [60] C. A. Kumar, N. K. Nair, NSGA-II based multiobjective PID controller tuning for speed control of DC motor drives, *International Journal of Soft Computing* 5 (3) (2010) 83–87.
- [61] C. Ibtissem, L. Noureddine, B. Pierre, Tuning PID controller using multiobjective ant colony optimization, *Applied Computational Intelligence and Soft Computing* 2012 (2012) <http://dx.doi.org/10.1155/2012/536326>.
- [62] A.-G. Esmaeil, L. Caro, Designing an optimal PID controller using imperialist competitive algorithm, *Intelligent Systems Scientific Society of Iran* 1 (2007) 1–6.
- [63] G. P. Liu, J. F. Whidborne, G. R. Duan, Multiobjective design using various control techniques, in: Proceedings of IEEE International Symposium on Computer Aided Control System Design, 2002, pp. 1–6.
- [64] G. P. Liu, S. Daley, Optimal-tuning PID controller design in the frequency domain with application to a rotary hydraulic system, *Control Engineering Practice* 7 (7) (1999) 821–830.
- [65] V. Pareto, *Manuale di Economia Politica* (English Translation by A.S. Schwier: *Manual of Political Economy*, Macmillan, New York, 1971), Societa Editrice Libreria, Milano, Italy, 1906.
- [66] K. Klamroth, J. Tind, M. Wiecek, Unbiased approximation in multicriteria optimization, *Mathematical Methods of Operations Research* 56 (2002) 413–437.

- [67] J. Fliege, Gap-free computation of Pareto-points by quadratic scalarizations, *Mathematical Methods of Operations Research* 59 (2004) 69–89.
- [68] G. Eichfelder, *Adaptive Scalarization Methods in Multiobjective Optimization*, Springer, Berlin Heidelberg, 2008.
- [69] V. Pereyra, Fast computation of equispaced Pareto manifolds and Pareto fronts for multiobjective optimization problems, *Math. Comput. Simul.* 79 (6) (2009) 1935–1947.
- [70] O. Schütze, M. Vasile, C. A. Coello Coello, Computing the set of epsilon-efficient solutions in multiobjective space mission design, *Journal of Aerospace Computing, Information, and Communication* 8 (3) (2011) 53–70.
- [71] J. Jahn, Multiobjective search algorithm with subdivision technique, *Computational Optimization and Applications* 35 (2) (2006) 161–175.
- [72] O. Schütze, M. Vasile, O. Junge, M. Dellnitz, D. Izzo, Designing optimal low thrust gravity assist trajectories using space pruning and a multi-objective approach, *Engineering Optimization* 41 (2) (2009) 155–181.
- [73] F. H. Bursal, C. S. Hsu, Application of a cell-mapping method to optimal control problems, *International Journal of Control* 49 (5) (1989) 1505–1522.
- [74] L. G. Crespo, J. Q. Sun, Stochastic optimal control of nonlinear dynamic systems via bellman’s principle and cell mapping, *Automatica* 39 (12) (2003) 2109–2114.
- [75] H. Flashner, T. F. Burns, Spacecraft momentum unloading: the cell mapping approach, *Journal of Guidance, Control and Dynamics* 13 (1990) 89–98.
- [76] W. H. Zhu, M. C. Leu, Planning optimal robot trajectories by cell mapping, in: *Proceedings of Conference on Robotics and Automation*, 1990, pp. 1730–1735.
- [77] F. Y. Wang, P. J. A. Lever, A cell mapping method for general optimum trajectory planning of multiple robotic arms, *Robotics and Autonomous Systems* 12 (1994) 15–27.
- [78] J. Y. Yen, Computer disk file track accessing controller design based upon cell to cell mapping, in: *Proceedings of the American Control Conference*, 1992.
- [79] L. G. Crespo, J. Q. Sun, Solution of fixed final state optimal control problems via simple cell mapping, *Nonlinear Dynamics* 23 (2000) 391–403.
- [80] L. G. Crespo, J. Q. Sun, Optimal control of target tracking via simple cell mapping, *Journal of Guidance and Control* 24 (2000) 1029–1031.

- [81] L. G. Crespo, J. Q. Sun, Fixed final time optimal control via simple cell mapping, *Nonlinear Dynamics* 31 (2003) 119–131.
- [82] L. G. Crespo, J. Q. Sun, Stochastic optimal control of non-linear systems via short-time Gaussian approximation and cell mapping, *Nonlinear Dynamics* 28 (2002) 323–342.
- [83] L. G. Crespo, J. Q. Sun, Optimal control of populations of competing species, *Nonlinear Dynamics* 27 (2002) 197–210.
- [84] G. F. Franklin, J. D. Powell, A. Emami-Naeini, *Feedback Control of Dynamic Systems*, Addison-Wesley, Reading, Massachusetts, 1986.
- [85] F. L. Lewis, V. L. Syrmos, *Optimal Control*, John Wiley and Sons, New York, 1995.
- [86] J. Q. Sun, B. Song, Some control studies of dynamical systems with time delay, in: A. C. J. Luo (Ed.), *Dynamics and Vibrations of Discontinuous, Stochastic and Time Delay Systems*, Springer, New York, 2010, pp. 135–157.
- [87] O. Schütze, A. Lara, C. A. Coello Coello, The directed search method for unconstrained multi-objective optimization problems, Tech. Rep. COA-R1, CINVESTAV-IPN (2010).
- [88] G. P. Liu, J.-B. Yang, J. F. Whidborne, *Multiobjective Optimisation and Control*, Research Studies Press, Baldock, Hertfordshire, England, 2002.
- [89] S. Panda, Multi-objective PID controller tuning for a FACTS-based damping stabilizer using non-dominated sorting genetic algorithm-II, *International Journal of Electrical Power & Energy Systems* 33 (7) (2011) 1296–1308.
- [90] Y. Luoa, Y. Chen, Stabilizing and robust fractional order PI controller synthesis for first order plus time delay systems, *Automatica* 48 (2012) 2159–2167.
- [91] R. L. Magin, *Fractional Calculus in Bioengineering*, Begell House Redding, 2006.
- [92] Y. A. Rossikhin, M. V. Shitikova, Application of fractional calculus for dynamic problems of solid mechanics: Novel trends and recent results, *Applied Mechanics Reviews* 63 (1) (2009) 010801–010801.
- [93] I. Podlubny, *Fractional Differential Equations: An Introduction to Fractional Derivatives, Fractional Differential Equations, to Methods of Their Solution and Some of Their Applications*, Academic Press, 1998.

- [94] F. Rüdinger, Tuned mass damper with fractional derivative damping, *Engineering Structures* 28 (13) (2006) 1774–1779.
- [95] M. Dalir, M. Bashour, Applications of fractional calculus, *Applied Mathematical Sciences* 4 (21) (2010) 1021–1032.
- [96] S. Hu, Y. Chen, T. Qiu, *Fractional processes and fractional-order signal processing*, Springer, London, 2012.
- [97] Y. A. Rossikhin, M. V. Shitikova, Application of fractional derivatives to the analysis of damped vibrations of viscoelastic single mass systems, *Acta Mechanica* 120 (1-4) (1997) 109–125.
- [98] J. J. De Espíndola, C. A. Bavastri, E. M. De Oliveira Lopes, Design of optimum systems of viscoelastic vibration absorbers for a given material based on the fractional calculus model, *Journal of Vibration and Control* 14 (9-10) (2008) 1607–1630.
- [99] J. Padovan, Y. Guo, General response of viscoelastic systems modelled by fractional operators, *Journal of the Franklin Institute* 325 (2) (1988) 247–275.
- [100] A. Shokooh, L. Suárez, A comparison of numerical methods applied to a fractional model of damping materials, *Journal of Vibration and Control* 5 (3) (1999) 331–354.
- [101] R. C. Koeller, Applications of fractional calculus to the theory of viscoelasticity, *Journal of Applied Mechanics* 51 (2) (1984) 299–307.
- [102] C. F. Lorenzo, T. Hartley, Variable order and distributed order fractional operators, *Nonlinear Dynamics* 29 (1-4) (2002) 57–98.
- [103] Z. Jiao, Y. Chen, I. Podlubny, *Distributed-order Dynamic Systems*, Springer, New York, 2012.
- [104] H. Sheng, Y. Chen, Optimal distributed-order fractional damping, in: *Proceedings of the 4th IFAC Workshop Fractional Differentiation and its Applications*, Badajoz, Spain, 2010.
- [105] R. C. Dorf, *Modern Control Systems*, Addison-Wesley, Boston, 1995.
- [106] C. Hernández, Y. Naranjani, Y. Sardahi, W. Liang, O. Schütze, J.-Q. Sun, Simple cell mapping method for multi-objective optimal feedback control design, *International Journal of Dynamics and Control* 1 (3) (2013) 231–238.
- [107] H. Butler, C. de Hoon, Fractional-order filters for active damping in a lithographic tool, *Control Engineering Practice* 21 (4) (2013) 413–419.

- [108] A. Tepljakov, E. Petlenkov, J. Belikov, J. Finajev, Fractional-order controller design and digital implementation using FOMCON toolbox for MATLAB, in: Proceedings of the IEEE International Symposium on Computer-Aided Control System Design, 2013, pp. 340–345.
- [109] Y. Naranjani, Y. Sardahi, J. Fernández, O. Schütze, J.-Q. Sun, A simple cell mapping and genetic algorithm hybrid method for multi-objective optimization problems, in: Proceedings of EVOLVE 2014 - A Bridge between Probability, Set Oriented Numerics, and Evolutionary Computation, Beijing, China, 2014.
- [110] H. Sheng, Y. Li, Y. Chen, Application of numerical inverse laplace transform algorithms in fractional calculus, *Journal of the Franklin Institute* 348 (2) (2011) 315–330.
- [111] J. C. Latombe, *Robot Motion Planning*, Kluwer Academic Publishers, 1991.
- [112] S. M. LaValle, *Planning algorithms*, Cambridge University Press, 2006.
- [113] S. Mittal, K. Deb, Three-dimensional offline path planning for uavs using multiobjective evolutionary algorithms, in: Proceedings of IEEE Congress on Evolutionary Computation. CEC, 2007, pp. 3195–3202.
- [114] F. Ahmed, K. Deb, Multi-objective optimal path planning using elitist non-dominated sorting genetic algorithms, *Soft Computing* 17 (7) (2013) 1283–1299.
- [115] M. Davoodi, F. Panahi, A. Mohades, S. N. Hashemi, Multi-objective path planning in discrete space, *Applied Soft Computing* 13 (1) (2013) 709–720.
- [116] Y. Zhang, D.-W. Gong, J. h. Zhang, Robot path planning in uncertain environment using multi-objective particle swarm optimization, *Neurocomputing* 103 (2013) 172–185.
- [117] C. Behring, M. Bracho, M. Castro, J. A. Moreno, An algorithm for robot path planning with cellular automata, in: Proceedings of The Fourth International Conference on Cellular Automata for Research and Industry, 2000.
- [118] Y. Tavakoli, H. H. S. Javadi, S. Adabi, A cellular automata based algorithm for path planning in multi-agent systems with a common goal, *International Journal of Computer Science and Network Security* 8 (7) (2008) 119–123.
- [119] K. Ioannidis, G. C. Sirakoulis, I. Andreadis, A path planning method based on cellular automata for cooperative robots, *Applied Artificial Intelligence* 25 (8) (2011) 721–745.

- [120] S. U. Ahmed, A. Akhter, F. Kunwar, Cellular automata based real time path planning for mobile robots, in: Proceedings of 12th International Conference on Control Automation Robotics & Vision (ICARCV), 2012, pp. 142–147.
- [121] G. S. Ferreira, P. Vargas, G. B. Oliveira, An improved cellular automata-based model for robot path-planning, in: Advances in Autonomous Robotics Systems, Vol. 8717 of Lecture Notes in Computer Science, Springer International Publishing, 2014, pp. 25–36.
- [122] U. A. Syed, F. Kunwar, Cellular automata-based real-time path-planning for mobile robots, International Journal of Advanced Robotic Systems 11 (2014) 93.
- [123] M. Drela, Pros and cons of airfoil optimization, Frontiers of computational fluid dynamics 1998.
- [124] M. Gen, R. Cheng, Genetic algorithms and engineering optimization, John Wiley & Sons, 2000.
- [125] R. Ma, B. Zhong, P. Liu, D. Drikakis, Multi-objective optimization design of lowreynolds- number airfoils s1223, in: Proceedings of 27th Congress of the International Council of the Aeronautical Sciences 2010, ICAS 2010, Vol. 2, Nice, France, 2010, pp. 916–925.
- [126] M. Rong, B. Zhong, L. Peiqing, W. Wenhui, Multi-objective optimization design of low-reynolds-number airfoil in near space, in: Proceedings of Systems and Control in Aeronautics and Astronautics (ISSCAA), 2010 3rd International Symposium on, 2010, pp. 147–152.
- [127] A. Arias-Montano, C. A. Coello Coello, E. M. Montes, Multi-objective airfoil shape optimization using a multiple-surrogate approach, in: Proceedings of Evolutionary Computation (CEC), 2012 IEEE Congress on, 2012, pp. 1–8.
- [128] A. Vicini, D. Quagliarella, Inverse and direct airfoil design using a multiobjective genetic algorithm, AIAA journal 35 (9) (1997) 1499–1505.
- [129] M. Nemec, D. W. Zingg, T. H. Pulliam, Multipoint and multi-objective aerodynamic shape optimization, AIAA Journal 42 (6) (2004) 1057–1065.
- [130] T. Ray, H. M. Tsai, Swarm algorithm for single- and multiobjective airfoil design optimization, AIAA Journal 42 (2) (2004) 366–373.
- [131] K. Chiba, S. Obayashi, K. Nakahashi, H. Morino, High-fidelity multidisciplinary design optimization of wing shape for regional jet aircraft, in: Proceedings of Evolutionary Multi-Criterion Optimization. Third International Conference, EMO 2005. Proceedings, Berlin, Germany, 2005, pp. 621–35.

- [132] Z. Zhu, H. Fu, R. Yu, J. Liu, Multi-objective optimization design of airfoil and wing, *Science in China Series E: Technological Sciences* 47 (1) (2004) 15–25.
- [133] J. Hájek, Parameterization of airfoils and its application in aerodynamic optimization, in: *Proceedings of WDS*, Vol. 7, 2007, pp. 233–240.
- [134] N. P. Salunke, J. A. RA, S. Channiwala, Airfoil parameterization techniques: A review, *American Journal of Mechanical Engineering* 2 (4) (2014) 99–102.
- [135] H. Sobieczky, Parametric airfoils and wings, in: *Recent Development of Aerodynamic Design Methodologies*, Springer, 1999, pp. 71–87.
- [136] M. Drela, XFOIL, subsonic airfoil development system (2000) <http://web.mit.edu/drela/Public/web/xfoil/>.
- [137] M. Drela, XFOIL: An analysis and design system for low reynolds number airfoils, in: *Low Reynolds number aerodynamics*, Springer, 1989, pp. 1–12.
- [138] M. Reyes-Sierra, C. A. Coello Coello, Multi-objective particle swarm optimizers: A survey of the state-of-the-art, *International journal of computational intelligence research* 2 (3) (2006) 287–308.
- [139] C. A. Coello Coello, G. T. Pulido, M. S. Lechuga, Handling multiple objectives with particle swarm optimization, *Evolutionary Computation, IEEE Transactions on* 8 (3) (2004) 256–279.
- [140] Y. Naranjani, Multi-objective particle swarm optimization (MOPSO) code implemented in fortran 90 (2016) https://github.com/ynaranjani/MOPSO_FORTRAN.git.
- [141] FAA, United states standards for terminal instrument procedures, 3d ed, FAA Handbook 8260.3B .