Centro de Investigación y de Estudios
Avanzados
del Instituto Politécnico Nacional

Unidad Zacatenco

Departamento de Computación

Métodos Orientados a Conjuntos para Optimización Multi-objetivo

Tesis que presenta

Carlos Ignacio Hernández Castellanos

para obtener el Grado de

Doctor en Ciencias

en Computación

Director de la Tesis:

Dr. Oliver Steffen Schütze

Ciudad de México **Diciembre 2017**

Centro de Investigación y de Estudios
Avanzados
del Instituto Politécnico Nacional

Zacatenco Campus

Computer Science Department

Set Oriented Methods for Multi-objective Optimization

Thesis submitted by

Carlos Ignacio Hernández Castellanos

as fulfillment of the requirement for

Doctor in

Computer Science

Advisor:

Dr. Oliver Steffen Schütze

Mexico City **December 2017**

Resumen

En las últimas décadas se ha incrementado el interés en resolver problemas de optimización multi-objetivo. Este tipo de problemas aparecen en casi cada aspecto de la vida, dado que es típico que se tengan varios objetivos en conflicto. La principal tarea del área es el encontrar una o varias de las mejores soluciones compromiso (que forman el conjunto/frente de Pareto). La mayoría de los algoritmos del estado del arte tienen como objetivo obtener una aproximación de estos conjuntos. Sin embargo, en muchos de los casos lo hacen sin dar información adicional del problema.

En esta tesis, se presenta el diseño y estudio de métodos orientados a conjuntos que son capaces de explotar información tal como bases de atracción, óptimos locales e información del vecindario. Esta información es útil para computar distintos conjuntos de interés para el tomador de decisiones además del conjunto/frente global de Pareto. Estos conjuntos incluyen los conjuntos/frentes locales de Pareto y las soluciones aproximadas que pueden ser útiles como respaldo. Además, el conjunto de soluciones óptimas ligeramente robustas que son de particular importancia cuando los problemas están sujetos a incertidumbre.

Abstract

In the last decades there has been an increased interest to solve multi-objective optimization problems. This kind of problems appear in almost every aspect of life, since it is typical to have several objectives that are in conflict. The focus of the area is to find one or several best trade-off solutions (that form the so-called global Pareto set/front). Most state-of-the-art algorithms aim to find an approximation of these sets. However, in most of the cases they do not give further information about the problem.

In this thesis, we focus on the design and study of set-oriented methods that are capable to exploit information such as basins of attraction, local optimal solutions and neighborhood information. Such information is useful to compute different sets of interest for the decision maker besides the global Pareto set/front. These sets include the local Pareto set/front and the set of nearly optimal solutions which can be useful as backup solutions. Further, the set of lightly robust optimal solutions which is in particular important when the problems are subject to uncertainties.

Acknowledgment

Over these years I have crossed path with a lot of people who have helped me to develop this work in one way or another. Some, directly guide me, gave me research advises. Others, helped me through off-research discussions and even by distracting me into doing other needed activities. I think all of these components were important for this work. Thus, I would like to take this opportunity to thank and acknowledge the people that have help me to get to this point.

To my adviser Dr. Oliver Schütze for all his guidance and support over these years. Also, for teaching me a lot of what it means to be a researcher. It was my pleasure to learn and work with him.

To the professors of the Computer Science department and to my thesis committee Dr. Carlos Coello, Dr. Luis Gerardo de la Fraga, Dr. Adriana Lara and Dr. Amilcar Meneces for the all comments and encouragement they gave me.

To all the people I had the opportunity to collaborate with. In particular, to Dr. Jian-Qiao Sun and his research group Dr. Yousef Naranjani, Dr. Yousef Sardahi and Dr. Fu-Rui Xiong for the opportunity to work with them as well as the help I received during the visits to UC Merced and Tianjin University.

To the NEO Group in particular to Serio Alvarado, Eduardo Salinas, Adrián Sosa, Oliver Cuate and Jesús Fernández for all the research discussions and the spare time activities.

To Sofy, Felipa, Erika, Dr. Santiago and Arcadio who were always available and whose help allowed me to focus in my research without having to worry about all bureaucratic procedures.

To all my lifelong friends and those latecomers for all the things they have taught me and all the moments we have shared.

To my family for all their support and for believing in me at all times.

To CINVESTAV and CONACYT for the opportunity given as well as the scholarship that allowed me to stay in the PhD program.

Contents

List of Figures

List of Tables

List of Algorithms

Chapter 1

Introduction

1.1 The Problem

There is always the wish for getting things better, cheaper, quicker, etc. which is inherent in human nature. Optimization is the field that deals with this problem. Sometimes only one objective is selected to be optimized, this leads to what is known as a single-objective optimization problem (SOP). However, in many cases we have more than one objective to be optimized and we need to consider them at the same time. This leads to the so-called multi-objective optimization problems (MOPs).

In the first case, we can use our intuition to define what is best in terms of our objective. If we are in the context of minimization, we know that the lower the value of the objective function, the better it is for our problem and we also know, that we are looking for "the solution", i.e. we expect to find that one solution is better than all the others.

In the second case, the problem gets more complicated because the definition of what is "better" is not as easy to define as it was in the previous case. This leads to another problem, since now we do not have "the solution" i.e, a unique one, but rather a set of solutions that are incomparable to each other.

In this kind of studies, typically the main focus is to find the global optimum (in single-objective optimization) or the global Pareto front (in multi-objective optimization). However, in practice, the decision maker may not always be interested in the best solutions, in particular, if these solutions are sensitive to perturbations. Thus, there exists an additional challenge. One has to search not only for solutions with a good performance but also that the solutions are possible to implement.

Notice that the computation of optimal solutions represents in general already a challenge. Even more, if as mentioned before the decision maker is interested in other kinds of solutions. For instance, approximate solutions may allow the decision maker to find alternative or backup solutions to a given problem. However, the set of these approximate solutions even forms an n-dimensional set, where n is the number of decision variables involved in the model. In contrast, a Pareto set typically, it is $n >> k$ and forms a $k-1$-dimensional set, where k is the number of objectives.

Another kind of solutions that the decision maker may be interested in are the so-called robust solutions. Robustness is an important issue when one faces real world applications. It might be the case that solutions given by traditional methods cannot be actually implemented with arbitrary accuracy. This means that the implemented solution differs from the original solution along with its objective value(s).

Designs which are affected by perturbations of any kind might no longer be acceptable to the decision maker from the practical point of view. Ideally, there exist optimal solutions that are insensitive to perturbations, up to some point. However, for most part, robustness and objective values are goals in conflict and one has to be willing to make a trade-off between them, in order to achieve an acceptable robustness level.

In this project, we propose to use the cell mapping techniques, which propose a discretization of the space into hypercubes called cells. The evolution of a dynamical system is then reduced to a new function, which is not defined in $\mathbb{R}^n$, but on the cell space. In this case, we restrict ourselves to functions that are strictly deterministically defined. For this case, we have the so-called simple cell mapping method, which is effective to obtain the attractors and basins of attraction of a dynamical system.

The basins of attraction are useful for both the computation of approximate and robust solutions, since they might be used to to get further information of a certain region of the space without the use of additional sampling techniques.

Thus, we propose to consider the basins of attraction in the context of multi-objective optimization, first by means of simple cell mapping. Further, we propose to design archiving techniques that allow us for a representation of the sets of interest and finally to include this idea into evolutionary algorithms in order to solve this kind of problems for higher dimensions.

1.2 Objectives

In the following, we state the objectives of this thesis.

1.2.1 General Objective

To design set oriented methods for the numerical treatment of multi-objective optimization problems with a special attention to nearly optimal solutions and robustness.

1.2.2 Particular Objectives

- To adapt cell mapping techniques to multi-objective optimization
- To design an archiving technique to approximate the set of nearly optimal solutions
- To design an evolutionary algorithm to approximate the set of nearly optimal solutions

- To design set oriented methods for light robust multi-objective optimization
- To apply the proposed methods to real world applications in multi-objective optimal control

1.3 Contributions

In the following, we list the publications obtained from this work.

Monographs

1. Sun, J.-Q., Xiong, F.-R., Schütze, O., Hernández, C. (2017). "Cell Mapping Methods – An Algorithmic Approach" (accepted). Springer International Publishing.

JCR Journals

1. Hernández, C., Schütze, O., Sun, J.-Q. (2017). Numerical Computation of Lightly Multi-objective Robust Optimal Solutions by Means of Generalized Cell Mapping. Knowledge-Based Systems (submitted).

2. Schütze, O., Hernández, C., Talbi, E-G. , Sun, J.-Q., Naranjani, Y., Xiong, F.-R. (2017). Archivers for the Representation of the Set of Approximate Solutions for MOPs. Journal of Heuristics (submitted).

3. Sardahi, Y., Sun, J. Q., Hernández, C., Schütze, O. (2017). Many-Objective Optimal and Robust Design of Proportional-Integral-Derivative Controls With a State Observer. Journal of Dynamic Systems, Measurement, and Control, 139(2).

4. Qin, Z. C., Xiong, F. R., Ding, Q., Hernández, C., Fernandez, J., Schütze, O., Sun, J. Q. (2017). Multi-objective optimal design of sliding mode control with parallel simple cell mapping method. Journal of Vibration and Control, 23(1), 46-54.

5. Fernández, J., Schütze, O., Hernández, C., Sun, J. Q., Xiong, F. R. (2016). Parallel simple cell mapping for multi-objective optimization. Engineering Optimization, 48(11), 1845-1868.

6. Xiong, F. R., Qin, Z. C., Ding, Q., Hernández, C., Fernandez, J., Schütze, O., Sun, J. Q. (2015). Parallel cell mapping method for global analysis of high-dimensional nonlinear dynamical systems. Journal of Applied Mechanics, 82(11).

Other journals

1. Enríquez-Zárate, J., Hernández, C., Trujillo L., Toledo-Ramírez, G.K., Ramos-Cirilo, Á. J. (2017). Smart Materials in Buildings: Vibration Control. Computación y Sistemas (to appear).

2. Naranjani, Y., Hernández, C., Xiong, F. R., Schütze, O., Sun, J. Q. (2017). A hybrid method of evolutionary algorithm and simple cell mapping for multi-objective optimization problems. International Journal of Dynamics and Control, 5(3), 570-582.

3. Hernández, C., Naranjani, Y., Sardahi, Y., Liang, W., Schütze, O., Sun, J. Q. (2013). Simple cell mapping method for multi-objective optimal feedback control design. International Journal of Dynamics and Control, 1(3), 231-238.

4. Xiong, F., Qin, Z., Hernández, C., Sardahi, Y., Narajani, Y., Liang, W., 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.

Book chapters

1. Hernández, C., Schütze, O., Sun, J. Q. (2017). Global Multi-objective Optimization by Means of Cell Mapping Techniques. In EVOLVE–A Bridge between Probability, Set Oriented Numerics and Evolutionary Computation VII (pp. 25-56). Springer International Publishing.

In proceedings of international conferences

1. Menchaca-Mendez, A., Hernández, C., Coello, C. A. C. (2016). Δ_p-MOEA: A new multi-objective evolutionary algorithm based on the Δ_p indicator. In Evolutionary Computation (CEC), 2016 IEEE Congress on (pp. 3753-3760). IEEE.

2. Qin, Z. C., Xiong, F. R., Ding, Q., Hernández, C., Fernandez, J., Schütze, O., Sun, J. Q. (2015). Multi-Objective Optimal Design and Validation of Sliding Mode Control. In ASME 2015 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference (pp. V008T13A015-V008T13A015). American Society of Mechanical Engineers.

3. Kerschke, P., Preuss, M., Hernández, C., Schütze, O., Sun, J. Q., Grimme, C., Trautmann, H. (2014). Cell mapping techniques for exploratory landscape analysis. In EVOLVE-A Bridge between Probability, Set Oriented Numerics, and Evolutionary Computation V (pp. 115-131). Springer International Publishing.

4. Hernández, C., Schütze, O., Emmerich, M., Xiong, F.-R., Sun, J.-Q. (2014). Barrier Tree for Continuous Landscapes by Means of Generalized Cell Mapping.

Proceedings of EVOLVE 2014 - A Bridge between Probability, Set Oriented Numerics, and Evolutionary Computing. Beijing, P. R. China.

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

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

In memories of national conferences

1. Hernández, C., Schütze, O., Sun, J.-Q. (2016). Cómputo del conjunto soluciones aproximadas para problemas de optimización multi-objetivo. V Congreso Nacional de la Sociedad Mexicana de Investigación de Operaciones.

1.4 Outline

The document is organized as follows. In Chapter 2, we present the required background including the basic concepts of multi-objective optimization, nearly optimal solutions and robustness. It also presents some representative algorithms used to solve those problems. Next, in Chapter 3, we study the cell mapping techniques in the context of multi-objective optimization. We present several adaptions of the methods and applications to optimal control problems. Then, in Chapter 4, we study the problem of finding the set of nearly optimal solutions through stochastic algorithms. Further, in Chapter 5 we present algorithms that aim for an approximation of the set of lightly robust optimal solutions. Finally, Chapter 6 states our conclusions and possible paths for future work.

Chapter 2

Background and Related Work

In this chapter, we look into the basic concepts that are needed to understand this thesis work. First, we review multi-objective optimization (Section 2.1), next nearly optimal solutions (Section 2.2), then uncertainty in optimization (Section 2.3) and finally robust multi-objective optimization (Section 2.4).

2.1 Multi-objective Optimization

In this section, we introduce the formulation of the problem, the notion of optimality, as well as some most representative methods to solve a multi-objective optimization problem (MOP).

2.1.1 Formulation of the problem

The multi-objective optimization problem can be defined in its general form as

$$\begin{aligned} &\min_{x \in Q} \{F(x)\}, \\ &\text{s.t.} \\ &g_i(x) \leq 0,\ i = 1, \ldots, I, \\ &h_j(x) = 0,\ j = 1, \ldots, J, \end{aligned} \tag{2.1}$$

where $Q \subseteq \mathbb{R}^n$ is the feasible region defined by

$$Q = \{x \in \mathbb{R}^n : g_i(x) \leq 0 \text{ and } h_j(x) = 0\}, \tag{2.2}$$

and $F : Q \to \mathbb{R}^k$ is a vector consisting of the objective functions

$$f_i : Q \to \mathbb{R},\ i = 1, \ldots, k, \tag{2.3}$$

$x \in Q$ is known as a decision vector, $g_i : \mathbb{R}^n \to \mathbb{R},\ i = 1, \ldots, I$, are inequality constraints and $h_i : \mathbb{R}^n \to \mathbb{R},\ j = 1, \ldots, J$, are equality constraints.

In case there are no constraints MOP(2.1) is said to be unconstrained. We can also see that in case $k = 1$ the problem is a single-objective optimization problem (SOP). It is important to notice that we could also state the MOP as a maximization problem, however, any maximization problem can be stated as a minimization problem, by multiplying the objective function vector by -1. For the remainder of this document we will use the term MOP for problems where the feasible region is only defined by box constraints Q_B

$$Q_B = \{x : lb_i \leq x_i \leq ub_i\}, \tag{2.4}$$

where lb_i $i = 1, \ldots, n$, and ub_i $i = 1, \ldots, n$, are the lower and upper bounds respectively.

2.1.2 Pareto optimality

Here, we present the concept of Pareto dominance and the necessary optimality conditions for differentiable MOPs.

Pareto dominance

In order to compare two vectors of solutions of a given MOP, we introduce the concept of Pareto dominance (Pareto, 1927).

Definition 1. *Let $v, w \in \mathbb{R}^k$. Then the vector v is less than w (denoted by $v <_p w$), if $v_i < w_i$, for all $i \in 1, \ldots, k$. The relation $\leq_p$ is defined analogously.*

Definition 2 (Strong Pareto dominance)**.** *A vector $y \in Q$ is called strongly dominated by a vector $x \in Q$ (denoted by $x \prec\prec y$) with respect to MOP(2.1) if $F(x) <_p F(y)$.*

Definition 3 (Pareto dominance)**.** *A vector $y \in Q$ is called dominated by a vector $x \in Q$ (denoted by $x \preceq y$) with respect to MOP(2.1) if $F(x) \leq_p F(y)$ and $F(x) \neq F(y)$, else y is called non-dominated by x.*

Definition 4 (Weak Pareto dominance)**.** *A vector $y \in Q$ is called weakly dominated by a vector $x \in Q$ (denoted by $x \prec y$) with respect to MOP(2.1) if $F(x) \leq_p F(y)$.*

Figure 2.1 shows an example of Pareto dominance.

The Pareto dominance determines which of two solutions x and y is "better". This leads naturally yo the definition of optimality.

Definition 5 (Pareto optimal solution)**.** *A point $x \in Q$ is called Pareto point of MOP(2.1) if there is no $y \in Q$ that dominates x.*

Strong and weak optimality are defined analogously. Usually, several solutions will be non-dominated by any other solution in the feasible region. This set is the so-called Pareto set and its image is known as the Pareto front.

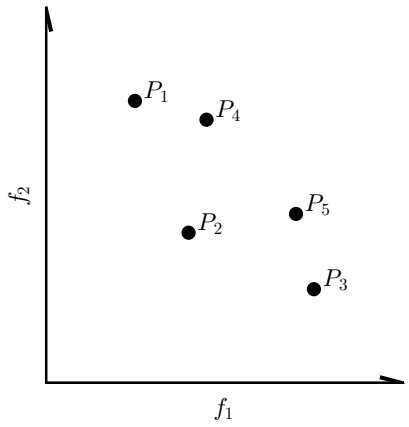


Figure 2.1: Example of Pareto dominance. In this case P_1, P_2 and P_3 are mutually non-dominated. P_2 dominates both P_4 and P_5.

Definition 6 (Pareto set and Pareto front)**.** *1. The set of all Pareto optimal solutions is called the Pareto set, i.e.,*

$$P = \{x \in Q : x \text{ is a Pareto optimal point of MOP(2.1)}\}. \tag{2.5}$$

2. The image $F(P)$ of P is called the Pareto front.

Typically, both the Pareto set and the Pareto front form $(k-1)$-dimensional objects under some mild assumptions on MOP(2.1) see Hillermeier (2001) for a more thorough discussion.

As an example, consider the following bi-objective optimization problem

$$\begin{aligned} F(x) &= (f_1(x), f_2(x)), \text{ where:} \\ f_1(x_1, x_2) &= (x_1 - 1)^2 + (x_2 - 1)^4, \\ f_2(x_1, x_2) &= (x_1 + 1)^2 + (x_2 + 1)^2, \end{aligned} \tag{2.6}$$

where $-3 \leq x_1, x_2 \leq 3$. Figure 2.2 shows the Pareto set/front of MOP 2.6.

Optimality conditions

If all the objectives in an MOP are differentiable, the Theorem of Kuhn and Tucker (Kuhn and Tucker, 1951) states a necessary condition for Pareto optimality.

Theorem 1 (Karush-Kuhn-Tucker condition)**.** *Let $x^* \in \mathbb{R}^n$ be a Pareto point of*

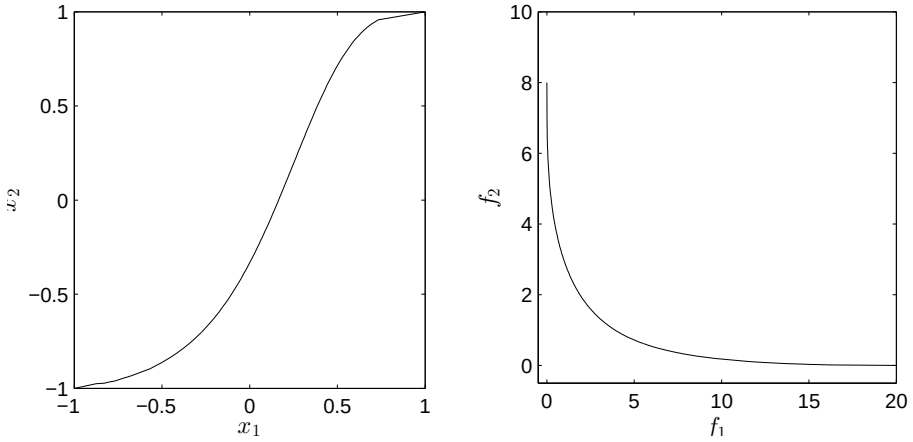


Figure 2.2: Example of Pareto set (left) and Pareto front (right) of MOP 2.6.

MOP(2.1), then there exists a vector $\alpha \in \mathbb{R}^k$ *such that*

$$\begin{aligned} &\sum_{i=1}^{k} \alpha_i \nabla f_i(x^*), + \sum_{j=1}^{I} \mu_i \nabla g_j(x^*) = 0 \\ &\alpha_i \geq 0, i = 1, \ldots, k, \\ &\sum_{i=1}^{k} \alpha_i = 1, \\ &\mu_j g_j(x^*) = 0, j = 1, \ldots, I, \end{aligned} \tag{2.7}$$

where $\nabla f_i(x^*)$ is the gradient of f_i at the point x^* is defined as follows

$$\nabla f_i(x^*) = \left(\frac{\partial f_1}{\partial x_1}(x^*), \ldots, \frac{\partial f_i}{\partial x_n}(x^*) \right), \ i = 1, \ldots, k. \tag{2.8}$$

This theorem claims for the unconstrained case that the zero vector can be expressed as a convex combination of the gradients of the objective functions at x^*,

$$\sum_{i=1}^{k} \alpha_i \nabla f_i(x^*). \tag{2.9}$$

2.1.3 Solving a multi-objective optimization problem

In the following, we review several methods that have been proposed in the literature to solve a given MOP.

Classification of techniques

Here, we present two classifications of methods to solve MOPs.

By the number of solutions generated. Algorithms of this class of methods is divided by identifying the number of solutions generated (Talbi, 2009).

- **Single solution:** in this case, one solution is generated at each step. Examples are scalarization methods such as weighted sum (Zadeh, 1963), weighted Tchebycheff (Bowman, 1976), normal boundary intersection (Das and Dennis, 1998), among others.

- **Population based:** generate an approximation of the front in one run of the algorithm. Examples are cell mapping techniques (Hernández et al., 2013; Hernández et al., 2013; Fernández Cruz et al., 2014; Naranjani et al., 2013a), subdivision techniques (Schütze et al., 2003; Dellnitz et al., 2005a,b; Jahn, 2006; Schütze et al., 2009) and evolutionary algorithms (Deb, 2001a; Coello Coello et al., 2007). Methods of this kind are also known as set-based methods.

By participation of the decision maker. This classification is based on when the decision maker participates to select some solutions (Hwang and Masud, 1979).

- **A priori:** the decision maker must define the preferences of the objective functions before starting the search.

- **A posteriori:** first, the approximated Pareto front is generated, and then, it is presented to the decision maker, who selects the most preferred one(s) according to her/his preferences.

- **Interactive:** both optimizer and decision maker work progressively. The optimizer produces solutions and the decision maker provides preference information.

Scalarization techniques

One of the ideas to solve an MOP is to transform the problem into an auxiliary SOP. With this approach, we simplify the problem by reducing the number of objectives to one. Once we do this, we are now able to use one of the numerous methods to solve SOPs that have been proposed. However, typically the solution of a SOP consists of only one point, while the solution of an MOP is a set. Thus, the Pareto set can be approximated (in some cases not entirely) by solving a clever sequence of SOPs (Eichfelder, 2009; Das and Dennis, 1998; Fliege, 2004).

In the following, we shortly review the most widely used scalarization techniques. For a more thorough discussion we refer the interested reader to Miettinen (1999).

- **The weighted sum method (Zadeh, 1963):** the weighted sum method Miettinen (1999) is probably the oldest scalarization method. The underlying idea is to assign to each objective a certain weight $\alpha_i \geq 0$, and to minimize the

resulting weighted sum. Given MOP (2.1), the weighted sum problem can be stated as follows:

$$\begin{aligned} \min \;\; & f_\alpha(x) := \sum_{i=1}^{k} \alpha_i f_i(x) \\ \text{s.t.} \;\; & x \in Q, \\ & \alpha_i \geq 0, i = 1, \ldots, k, \\ & \sum_{i=1}^{k} \alpha_i = 1. \end{aligned} \tag{2.10}$$

The main advantage of the weighted sum method is that one can expect to find Pareto optimal solutions, to be more precise:

Theorem 2. *Let $\alpha_i > 0$, $i = 1, \ldots, k$, then a solution of Problem (2.10) is Pareto optimal.*

On the other hand, the proper choice of α, though it appears to be intuitive at first sight, is, in certain cases, a delicate problem. Further, the images of (global) solutions of Problem (2.10) cannot be located in parts of the Pareto front, where it is concave. That is, not all points of the Pareto front can be reached, when using the weighted sum method, which represents a severe drawback.

- **Weighted Tchebycheff method (Bowman, 1976):** the aim of the weighted Tchebycheff method is to find a point whose image is as close as possible to a given reference point $Z \in \mathbb{R}^k$. For the distance assignment, the weighted Tchebycheff metric is used: Let $\alpha \in \mathbb{R}^k$ with $\alpha_i \geq 0$, $i = 1, \ldots, k$, and $\sum_{i=1}^{k} \alpha_i = 1$, and let $Z = (z_1, \ldots, z_k) \in \mathbb{R}^k$, then the weighted Tchebycheff method reads as follows:

$$\min_{x \in Q} \max_{i=1,\ldots,k} \alpha_i |f_i(x) - z_i|. \tag{2.11}$$

Note that the solution of Problem (2.11) depends on Z as well as on α. The main advantage of the weighted Tchebycheff method is that by a proper choice of these vectors, every point on the Pareto front can be reached.

Theorem 3. *Let $x^* \in Q$ be Pareto optimal. Then there exists $\alpha \in \mathbb{R}^k_+$ such that x^* is a solution of Problem (2.11), where Z is chosen as the utopian vector of the MOP.*

The **utopian vector** $F^* = (f_1^*, \ldots, f_k^*)$ of an MOP consists of the minimum objective values f_i^* of each function f_i.
On the other hand, the proper choice of Z and α might also get a delicate problem for particular cases.

- **Normal boundary intersection (Das and Dennis, 1998):** the Normal boundary intersection (NBI) method Das and Dennis (1998) computes finite size approximations of the Pareto front in the following two steps:

  1. The Convex Hull of Individual Minima (CHIM) is computed, which is the $(k-1)$-simplex connecting the objective values of the minimum of each objective f_i, $i = 1, \ldots, k$ (i.e., the utopian).
  2. The points y_i from the CHIM are selected and the point $x_i^* \in Q$ is computed such that the image $F(x_i^*)$ has the maximal distance from y_i in the direction that is normal to the CHIM and points toward the origin.

  The latter is called the NBI-subproblem and can, in mathematical terms, be stated as follows: Given an initial value x_0 and a direction $\alpha \in \mathbb{R}^k$, solve

$$\begin{aligned} \max_{x,l} \ & l \\ \text{s.t.} \ & F(x_0) + l\alpha = F(x) \\ & x \in Q. \end{aligned} \tag{2.12}$$

  Problem (2.12) can be helpful, since there are scenarios where the aim is to steer the search in a certain direction given in objective space. On the other hand, solutions of Problem (2.12) do not have to be Pareto optimal Das and Dennis (1998).

Descent direction methods

Definition 7 (Descent direction)**.** *A vector ν is called a descent direction for f at $x \in \mathbb{R}^n$ if $f(x + t\nu) < f(x)$ for all $t \in (0, t)$.*

If f is differentiable in x then

$$\langle \nabla f(x), \nu \rangle < 0. \tag{2.13}$$

If a descent direction ν is given at a point x, a further candidate solution x_{new} that dominates x can easily be found by line search, i.e., by setting

$$x_{\text{new}} = x + t\nu, \tag{2.14}$$

where $t \in \mathbb{R}_+$ is a step size.

The solution of this kind of problems would give as result a curve of dominated points, i.e., the new point dominates the previous one. Figure 2.3 shows an example of a descent direction.

In the following, we present several methods which use this idea to find a descent direction ν.

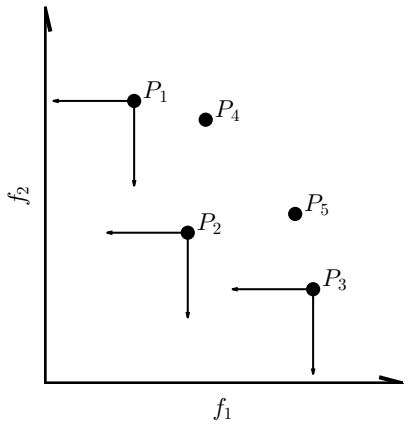


Figure 2.3: Example of descent directions.

- **Lara's bi-objective descent direction:** one way to combine two gradients to obtain a descent direction is by a vector sum.

  Theorem 4 (Lara's bi-objective descent direction Lara (2012))**.** *Let $x \in \mathbb{R}^n$, and $f_1, f_2 : \mathbb{R}^n \to \mathbb{R}$ define a two-objective MOP. If $\nabla f_i(x) \neq 0$, for $i = 1, 2$, then the direction*

  $$\nu(x_0) = -\left(\frac{\nabla f_1(x)}{||\nabla f_1(x)||} + \frac{\nabla f_2(x)}{||\nabla f_2(x)||}\right) \tag{2.15}$$

  is a descent direction of f_1 and f_2 at x_0 of MOP.

  However, this approach cannot be generalized for more than two objective functions and it is only for unconstrained problems.

- **Directed search:** the directed search method Schütze et al. (2016) allows to steer the search from a given point $x \in \mathbb{R}^n$ into a desired direction $d \in \mathbb{R}^k$. A direction vector $\nu \in \mathbb{R}^n$ can be computed such that

  $$\lim_{t \leftarrow 0} \frac{f_i(x_0 + tv) - f_i(x_0)}{t} = d_i, \ i = 1, \dots k. \tag{2.16}$$

  In the following, we describe two methods that use this idea. The first one is a descent method that steers in a given direction $d < 0$. The second one is a continuation method with the particular advantage that this method does not require any second gradient information in contrast to other methods (Allgower and Georg, 1990; Hillermeier, 2001).

  Descent method The following idea is proposed. Assume a point $x_0 \in Q$ is given as well as a vector $d \in \mathbb{R}^k$ representing a desired direction in objective space. This can be expressed as follows

  $$J(x)\nu = d, \tag{2.17}$$

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

$$J(x) = \begin{bmatrix} \frac{\partial f_1}{\partial x_1}(x) & \cdots & \frac{\partial f_1}{\partial x_n}(x) \\ \vdots & \ddots & \vdots \\ \frac{\partial f_k}{\partial x_1}(x) & \cdots & \frac{\partial f_k}{\partial x_n}(x) \end{bmatrix}. \tag{2.18}$$

With this the authors propose that v can be computed by solving a system of linear equations. Since typically the number of parameters is higher than the number of objectives, the system of equations is under-determined, which implies that its solution is not unique. To find the greedy direction one can chose the solution with the lowest norm, the problem can be formulated as

$$\nu = J(x_0)^+ d, \tag{2.19}$$

where $J(x_0)^+$ denotes the pseudo inverse of the Jacobian $J(x_0) \in \mathbb{R}^{k \times n}$. Further, we can solve the following initial value problem (IVP):

$$\begin{aligned} x(0) &= x_0 \in \mathbb{R}^n \\ \dot{x}(m) &= \nu_\alpha(x(m)),\ t > 0. \end{aligned} \tag{2.20}$$

Continuation method Once an optimal point has been found e.g. by the method above, this second method performs a movement along the Pareto set of a given MOP.

Assume we are given a (local) Pareto point x and the convex weight α such that

$$J(x)^T \alpha = 0 \tag{2.21}$$

and further we assume that

$$\operatorname{rank}(J(x)) = k - 1. \tag{2.22}$$

It is known (e.g., Hillermeier (2001)) that in this case α is orthogonal to the Pareto front, i.e.,

$$\alpha \perp T_y \delta F(\mathbb{R}^n), \tag{2.23}$$

where $y = F(x)$ and $\delta F(\mathbb{R}^n)$ denotes the border of the image $F(\mathbb{R}^n)$. Thus, a search orthogonal to α (in objective space) could be promising to obtain new predictor points. A QR-factorization of α can be computed to use the method above, i.e.,

$$\alpha = QR, \tag{2.24}$$

where $Q = (q_1, \ldots, q_k) \in \mathbb{R}^{k \times k}$ is an orthogonal matrix and $q_i = 1, \ldots, k$ its column vectors, and $R = (r_{11}, 0, \ldots, 0)^T \in \mathbb{R}^{k \times 1}$ with $r_{11} \in \mathbb{R} \backslash \{0\}$. Since by Equation (2.24) $\alpha = r_{11} q_1$ and Q orthogonal, it follows that the column vectors

$q_2, \ldots, q_k$ build an orthonormal basis of the hyperplane which is orthogonal to α. Thus, a promising set of search directions ν_i may be the ones which satisfy

$$J(x)\nu_i = q_i,\ i = 2, \ldots, k. \tag{2.25}$$

Since α is not in the image of $J(x)$ (else x would not be a Pareto point), it follows that the vectors $q_2, \ldots, q_k$ are in the image of $J(x)$, i.e., Problem (2.25) can be solved for each $i \in \{2, \ldots, k\}$. Then, the following can be chosen as the set of predictor direction:

$$p_i = x_0 + t\nu_i. \tag{2.26}$$

Note that by this choice of predictor direction, no second derivative of the objectives are required.

Now, a corrector step can be used. Given a predictor $p_i \in p$, we can use p_i as initial value for IVP (2.20) and choosing α_0, i.e., the weight from the previous solution x_0 leads to a new solution x_1.

- **Method of Schäffler, Schultz and Weinzierl:**

  Theorem 5 ((Schaeffler et al., 2002)). *Let (MOP) be given and $q : \mathbb{R}^n \to \mathbb{R}^n$ be defined by*

  $$q(x) = \sum_{i=1}^{k} \hat{\alpha}_i \nabla f_i(x), \tag{2.27}$$

  where $\hat{\alpha}$ is a solution of

  $$\min_{\alpha \in \mathbb{R}^k} \left\{ \left\| \sum_{i=1}^{k} \alpha_i \nabla f_i(x) \right\|_2^2 ; \alpha_i \geq 0, i = 1, \ldots, k, \sum_{i=1}^{k} \alpha_i = 1 \right\}. \tag{2.28}$$

  Then either $q(x) = 0$ or $-q(x)$ is a descent direction for all objective functions $f_1, \ldots, f_k$ in x.

- **Method of Fliege and Svaiter:** the following function is defined Fliege and Fux Svaiter (2000):

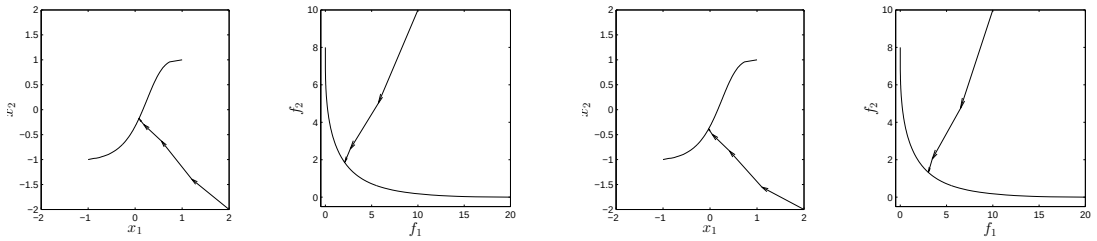
  $$f_x(\nu) = max(Av)_i, i = 1, \ldots, k, \tag{2.29}$$

  where $f_x : \mathbb{R}^n \to \mathbb{R}$. We can see that f_x is convex and positive homogeneous. Using this function the authors propose the following problem:

  $$\begin{aligned} &\min f_x(\nu) + \frac{1}{2}||v||^2 \\ &\text{subject to } \nu \in \mathbb{R}^n. \end{aligned} \tag{2.30}$$

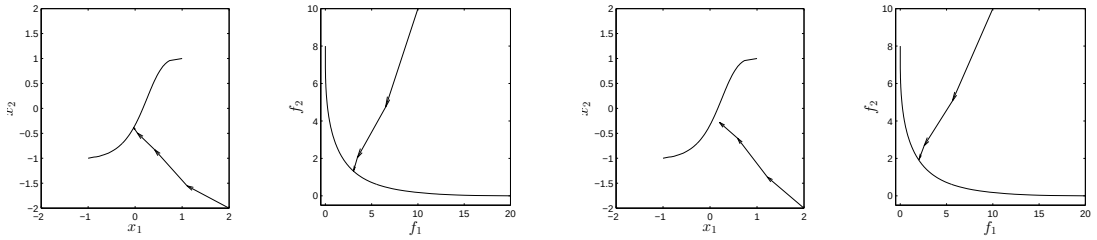
  From this we have that, if x is Pareto optimal, then $\nu(x) = 0$. If it is not the case, then $\nu(x)$ is a descent direction.

Figure 2.4 shows an example of the previous descent direction methods on MOP 2.6 with $x_0 = [2, -2]^T$.



(a) Lara's descent direction

(b) Directed search with $d = [-0.5; -0.5]^T$



(c) Fliege's descent direction

(d) SSW's descent direction

Figure 2.4: Result of the descent direction methods on the MOP 2.6 with $x_0 = [2, -2]^T$.

Multi-objective evolutionary methods

An alternative to the classical methods is given by an area known as Evolutionary Multi-Objective Optimization. This area has developed a wide variety of methods. These methods are known as Multi-Objective Evolutionary Algorithms (MOEAs). Some of the advantages are that they do not require gradient information about the problem, instead they rely on stochastic search procedures. Another advantage is that they give an approximation of the entire Pareto set and Pareto front in one execution. Examples of these methods can be found in Deb (2001a); Coello Coello et al. (2007).

Algorithm 1 shows a generic framework of a stochastic optimization algorithm Laumanns et al. (2002), which we will consider in this work. Hereby, j denotes the iteration step, P_j the candidate set (or population) and A_j the archive.

Algorithm 1 Generic Stochastic Search Algorithm

1: $P_0 \subset Q$ drawn at random
2: $A_0 = ArchiveUpdate(P_0, \emptyset)$
3: **for** $j = 0, 1, 2, \ldots$ **do**
4: $\quad P_{j+1} = Generate(P_j)$
5: $\quad A_{j+1} = ArchiveUpdate(P_{j+1}, A_j)$
6: **end for**

In the following, we review the most representative methodologies to design a

MOEA and its most important exponent.

- **Dominance based methods:** methods of this kind rely on the dominance relationship directly. In general, these methods have two main components: a dominance ranking and a diversity estimator. The dominance ranking sorts the solutions according to the partial order induced by Pareto dominance. This mechanism puts a pressure towards the Pareto front. The diversity estimator is introduced to have a total order of the solutions and thus, it allows a comparison between solutions even when they are mutually non-dominated. The density estimator is based on the idea that the decision maker would like to have a good distribution of the solution on the front (diversity).

  One of the most popular methods in this class is the non-dominated sorting genetic algorithm II (NSGA-II) that was proposed in Deb et al. (2002). The NSGA-II has two main mechanisms. The first one is the non-dominated sorting, where the idea is to rank the current solutions using the Pareto dominance relationship, i.e., the ranking says by how many solutions the current solution is dominated. This helps to identify the best values and to use them to generate new solutions.

  The second mechanism is called crowding distance. The idea is to measure the distance from a given point to its neighbors. This helps to identify the solutions with less distance as it means that there are more solutions in that region.

  These components aim to accomplish the goals of convergence and spread respectively. Due to these mechanisms, NSGA-II has a good overall performance and it has become a prominent method when two or three objectives are considered. Other methods in this class can be found in Zitzler and Thiele (1999); Zitzler et al. (2002).

- **Decomposition based methods:** methods of this kind rely on scalarization techniques to solve an MOP. The idea behind these methods is that a set of well distributed scalarization problems will generate also a good distribution of the Pareto front. They use the evaluation of the scalar problem instead of the evaluation of the MOP. These methods do not need a density estimator since it is coded in the selection of the scalar problems.

  The most prominent algorithm in this class is the MOEA based on decomposition (MOEA/D) which was proposed in Zhang and Li (2007). The main idea of this method is to make a decomposition of the original MOP into N different SOPs also called subproblems. Then the algorithm solves these subproblems using the information from its neighbor subproblems. The decomposition is made by using one of the following methods: weighted sum, weighted Tchebycheff or PBI. Other methods in this class can be found in Zuiani and Vasile (2013); Moubayed et al. (2014).

- **Indicator based methods:** these methods rely on performance indicators to assign the contribution of an individual to the solution found by the algorithm.

By using a performance indicator, these methods reduce an MOP to an SOP. In most cases, these methods do not need a density estimator, since it is coded in the indicator.

One of the methods that has received the most attention is the SMS-EMOA (Beume et al., 2007). This method is a $\mu + 1$ evolutionary strategy that uses hypervolume indicator to guide the search. The method computes the contribution to the hypervolume of each solution. Next, the solution with the minimum contribution is removed from the population. In order to improve the running time of the algorithm, SMS-EMOA is combined with the dominance based methods. In this case, when the whole population is non-dominated the indicator is used to rank the solutions. Other methods in this class are presented in Zitzler et al. (2008); Wagner and Trautmann (2010) for the hypervolume indicator and Rudolph et al. (2016); Schütze et al. (2016) for the Δ_p indicator.

Diversity based multi-objective evolutionary algorithms

The evolutionary methods described before focus on finding a well distributed approximation of the Pareto front. However, in some cases it is necessary to provide extra indications in the characteristics of the solution. For instance, one would like to reward solutions that are well distributed in design space. This is of special interest in the presence of many-to-one mappings . In such cases, the methods described before could neglect some interesting regions of the space for the decision maker. In the following, we review methods that consider diversity both in design and objective space. Either to improve their exploratory capacities or to find equivalent Pareto sets.

Diversity in both design and objective space. In the following, we review some works that maintain diversity in both design and objective space. These methods are based on the idea that diversity is the key to find better solutions.

- Toffolo and Benini (2003) used diversity as an aditional objective. An individual that is distant from all the others has more chances, when mating, to produce offspring in regions of the search space not covered by the current population. The solutions are first ranked according to their fitness and then sorted according their diversity measured in euclidean or hamming distance.

- Shir et al. (2009) proposed a CMA-ES niching framework. In this case, the method ranks the individuals based upon non-dominated sorting. Then, the distance between niches is calculated in the aggregated space and the estimation of the niche radius is adjusted. In order to test their approach, the authors used the test problems omni-test (Deb and Tiwari, 2008), EBN (Emmerich et al., 2005), two-on-one (Preuss et al., 2006) and Lamé superspheres (Emmerich, 2007).

- Zhou et al. (2009) proposed an estimation of distribution algorithm (EDA) called MMEA. In this algorithm, the population is clustered into a number of subpopulations based on their distribution in the objective space, the principal component analysis is used to estimate the dimensionality of the Pareto set in each subpopulation, and then a probabilistic model is built to model the distribution of the Pareto optimal solutions in decision space. Such modeling procedure could promote the population diversity in both decision and objective space.s. It is interesting to mention that the diversity in design space is promoted by an operator instead via selection.

- Ulrich et al. (2010) present a modified version of the hypervolume to measure diversity in design space. For design space diversity, the algorithm computes the size of the intersections between points to see which individuals are better. This approach only considers bi-objective problems.

- Zechman et al. (2011) propose an algorithm that generates subpopulations and that evolves them independently. Then it applies k-means to all individuals (in objective space) and form niches with individuals from the same subpopulation and clusters them. From the subpopulation 1, it finds the set of non-dominated solutions and relaxes it. In order to do this, all solutions from other subpopulations are feasible if they are below a target (using $-\epsilon$-dominance). Then, the spread of the solutions is used in a similar way to crowding distance. The approach was tested on the Lamé superspheres (Emmerich, 2007).

- Xia et al. (2014a) provide a review of methods to maintain diversity in both spaces. Further, they propose a method that aggregates the non-dominated rank, and the crowding distance in both spaces. The method uses a local searcher, the differential evolution operators, as well as, an external archiver. The method was tested on the UF (Zhang et al., 2008) and the problems used in Zhou et al. (2009).

Preserving equivalent Pareto sets. Rudolph et al. (2007) identified several classes of problems according to the number of Pareto sets/fronts that a given MOP has.

- Type I, One Pareto set and one Pareto front,

- Type II, one Pareto set and multiple Pareto front parts,

- Type III, multiple Pareto subsets and one Pareto front, and

- Type IV, multiple Pareto subsets and Pareto front parts.

Further, Preuss et al. (2006); Rudolph et al. (2007) presented test problems that make use of the previous classification. In the following, we will introduce methods that fall into Type IV.

- Rudolph et al. (2007) propose a multi-start approach. The method uses an SOO algorithm to find the extreme points of an MOP with a multi-start approach. Then, the method clusters the solutions to identify all the Pareto subsets. Further, the method uses those solutions to find the utopian vector and finally reconstructs the Pareto subsets using weighted Tchebycheff on a $1+1$ evolutionary strategy. In order to test the approach the authors proposed the sym-part problems. These problems have 9 different connected components that map the Pareto front.

- Deb and Tiwari (2008) proposed a global optimizer in order to solve both single and multi-objective optimization problems. The method uses a restricted selection where first an individual is selected at random and then selects the individual closest to it. In order to rank the solutions a slightly modified ϵ-dominance is used in the same for as NSGA (Srinivas and Deb, 1994).

- Rudolph and Preuss (2009) aimed to find solutions with similar characteristics to one that was previously selected by the decision maker. The method uses 2 objectives, first the distance between the solutions in the objective space and second the distance in design space. The algorithm works as follows: first, it uses a MOEA on the bi-objective problem. Then, a local search is performed on the solutions found. The approach was tested on the two-on-one (Preuss et al., 2006) and sym-part problems (Rudolph et al., 2007).

- Kramer and Danielsiek (2010) proposed a MOEA with rake selection (Kramer and Koch, 2009) for objective space and DBScan (Ester et al., 1996) to cluster in decision space. After the initialization, the method executes DBScan for decision space. After the initialization, the initialization, the method executes DBScan to find the clusters and then evolves them independently. Next, for each solution the method computes the recluster indicator with its neighbor and if it is bigger than a threshold it applies DBScan again. Further, Kramer and Danielsiek (2011) present an extension that used SMS-EMOA and kernel density clustering.

2.1.4 Performance measures

In the following, we review the performance measures that will be used in this work in order to compare algorithms.

Hausdorff distance

Now, we present the Hausdorff distance d_H (e.g. Heinonen (2001)), which is used to measure the distance between sets. Then, we introduce the indicator Δ_p, which is used to make out comparisons in this work.

Definition 8 (Distance between sets)**.** *Let $u, v \in \mathbb{R}^n$ and $A, B \subset \mathbb{R}^n$. The maximum norm distance d_∞, the semi-distance $dist(\cdot,\cdot)$ and the Hausdorff distance $d_H(\cdot,\cdot)$ are defined as follows:*

1. $d_\infty = \max_{i=1,\ldots,n} |u_i - v_i|$
2. $dist(u, A) = \inf_{v \in A} d_\infty(u, v)$
3. $dist(B, A) = \sup_{u \in B} d_\infty(u, A)$
4. $d_H(A, B) = \max\{dist(A, B), dist(B, A)\}$

It will be assumed that the infinity norm is used unless it is specified otherwise.

Generational distance

The generational distance (GD) is used to measure the distance from the candidate set $A = \{a_1, \ldots, a_m\}$ to an approximation of the Pareto front B Van Veldhuizen (1999). The lower the value of GD the better is the candidate set A. A value of zero means, $A \in B$.

$$\mathrm{GD}(A, B) := \frac{1}{|A|} \left(\sum_{i=1}^{|A|} dist_2(a_i, B)^p \right)^{\frac{1}{p}}. \tag{2.31}$$

However, if an element of the set is duplicated, then the indicator gives a value closer to zero. To avoid this potential drawback a slight modification was proposed in Schütze et al. (2012).

$$\mathrm{GD}(A, B)_p := \left(\frac{1}{|A|} \sum_{i=1}^{|A|} dist_2(a_i, B)^p \right)^{\frac{1}{p}}. \tag{2.32}$$

Inverted generational distance

The inverted generational distance (IGD) measures the distance from an approximation of the Pareto front B to the candidate set A (Coello Coello and Cruz Cortés, 2005). However, this indicator is sensitive to the discretization of the Pareto front. If a better discretization of the Pareto front is used, then it will output a value closer to zero.

$$\mathrm{IGD}(A, B) := \frac{1}{|B|} \left(\sum_{i=1}^{|B|} dist_2(b_i, A)^p \right)^{\frac{1}{p}}. \tag{2.33}$$

To avoid this potential drawback, a slight modification was proposed in Schütze et al. (2012):

$$\mathrm{IGD}(A,B)_p := \left(\frac{1}{|B|}\sum_{i=1}^{|B|} dist_2(b_i,A)^p\right)^{\frac{1}{p}}. \tag{2.34}$$

Averaged Hausdorff distance (Schütze et al., 2012)

The Hausdorff distance defines a metric in certain cases. However, since it penalizes outliers heavily, it is not widely used in evolutionary computation to compare different algorithms. To partially overcome this issue (Schütze et al., 2012) proposed an indicator that can be viewed as an averaged Hausdorff distance.

Let $A = a_1, \ldots, a_n, B = b_1, \ldots, b_m \subset \mathbb{R}^k$ be finite and non-empty sets. Then, we define $\Delta_p(A,B)$ as

$$\Delta_p(A,B) := max(GD_p(A,B), IGD_p(A,B)). \tag{2.35}$$

Hypervolume (Zitzler and Thiele, 1998)

One of the issues with the indicators previously presented is that they are not Pareto compliant. In the following, we define an indicator that is Pareto compliant. That is, given two sets of solutions A and B whenever $A \preceq B$ and $B \not\preceq A$ the indicator of A should not be worse than the indicator of B. Thus, the Hypervolume indicator has received a lot of attention from the community of evolutionary computation.

Given a set of non-dominated solutions $P \subseteq \mathbb{R}^k$ and a reference point $R \in \mathbb{R}^k$, where Λ denotes the Lebesgue measure, the hypervolume is defined as follows:

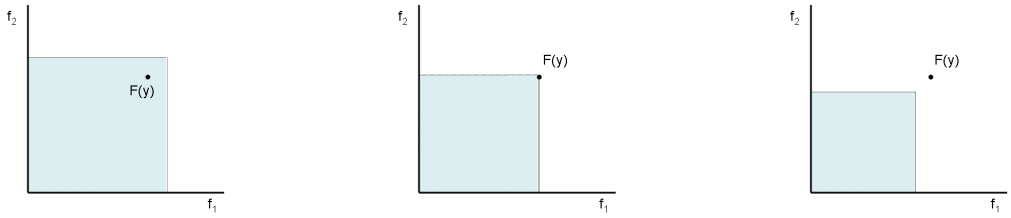
$$S(P,R) = \Lambda\left(\bigcup_{y\in P} y' | y \prec y' \prec R\right). \tag{2.36}$$

2.2 Nearly Optimal Solutions in Multi-objective Optimization

In certain situations it may be beneficial for the decision maker (DM) to consider in addition to the optimal solutions also nearly optimal ones, which are alternatively called approximate solutions or ϵ-efficient solutions. The reason for this is that by this the number of valid options for the current setting may increase. Note that if two points x and y are near in objective space, i.e., if $F(x) \approx F(y)$, this does not have to hold in decision space.

2.2.1 Formulation of the problem

We now define another notion of dominance which we will use to define nearly optimal solutions (Loridan, 1984). Here we define the set of interest, $P_{Q,\epsilon}$, and investigate



(a) ϵ-dominance (b) classical dominance (c) $-\epsilon$-dominance

Figure 2.5: Regions of dominance for the different dominance concepts. A point $x \in \mathbb{R}^n$ dominates the point y if the image $F(x)$ is contained in the shaded region in Figure (b). Figures (a) and (c) are analog for ϵ-dominance and $-\epsilon$-dominance.

its topology. First, we introduce two calculation rules that will be helpful for the following considerations.

Definition 9. *$-\epsilon$-dominance Let $\epsilon = (\epsilon_1, \ldots, \epsilon_k) \in R_+^k$ and $x, y \in Q$. x is said to $-\epsilon$-dominate y ($x \prec_{-\epsilon} y$) with respect to MOP(2.1) if*

$$F(x) - \epsilon \leq_p F(y) and F(x) + \epsilon \neq F(y). \tag{2.37}$$

Figure 2.5 shows the regions of dominance for the different dominance concepts. Now we are in the position to define $P_{Q,\epsilon}$.

Definition 10. *Denote by $P_{Q,\epsilon}$ the set of points in $Q \subset \mathbb{R}^n$ that are not $-\epsilon$-dominated by any other point in Q, i.e.,*

$$P_{Q,\epsilon} = \{x \in Q | \nexists y \in Q : y \prec_{-\epsilon x}\}. \tag{2.38}$$

Note that one would like to compute

$$P_{Q,\epsilon}^c := \{x \in Q | \exists p \in P_Q : x \prec_\epsilon p\} \tag{2.39}$$

as every point $x \in P_{Q,\epsilon}^c$ is 'close enough' to at least one efficient solution. Note, however, that the definition of this set contains the optimal set P_Q and is thus not easy to compute. Therefore, we consider $P_{Q,\epsilon}$ instead. Figure 2.6 shows an example of nearly optimal solutions.

Finally, some notations from topology. Let $d : \mathbb{R}^n \times \mathbb{R}^n \to [0, \infty)$ be a distance in $\mathbb{R}^n$ (within this study we will use the distance coming from the 2-norm, i.e., $d(x, y) := \|x - y\|_2$). Given a point $x_0 \in \mathbb{R}^n$ and a value $\epsilon > 0$, the set

$$B_\epsilon(x_0) := \{y \in \mathbb{R}^n \ : \ d(x, y) < \epsilon\} \tag{2.40}$$

denotes the ϵ-ball centered at x_0. A point $x_0 \in A \subset \mathbb{R}^n$ is called an *interior point* in A if there exists an $\epsilon > 0$ such that $B_\epsilon(x_0)$ lies entirely in A. A point $x_0 \in A$ is called a *boundary point* of A if any ϵ-ball $B_\epsilon(x_0)$ has non-empty intersections with both A and its complement $\mathbb{R}^n \backslash A$. The set of interior points in A constitutes its *interior*,

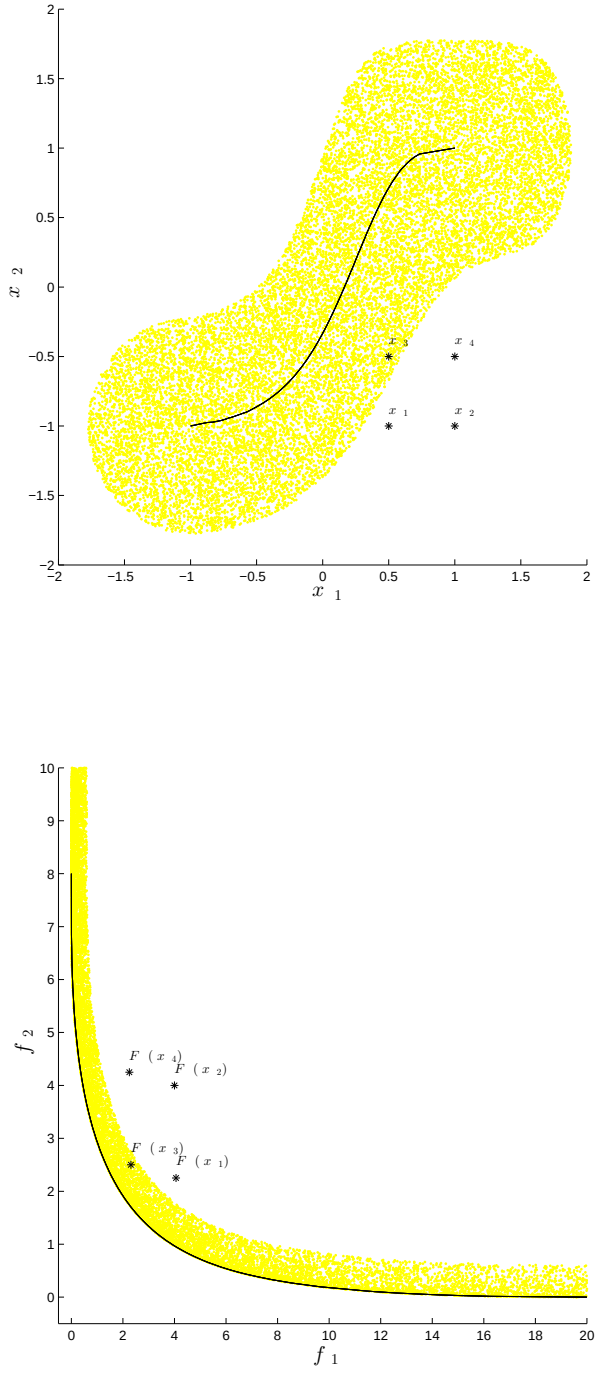


Figure 2.6: Example of nearly optimal solutions on MOP 2.6 with $\epsilon = [0.6, 0.6]^T$. In yellow, it is shown the set of nearly optimal solutions $P_{Q,\epsilon}$. The black lines represent the Pareto set (left) and front (right). In this case, only $x_3 \in P_{Q,\epsilon}$.

denoted by $\overset{\circ}{A}$, and the set of boundary points its *boundary*, denoted by ∂A. A is said to be *open* if any point in A is an interior point and it is *closed* if its boundary ∂A is contained in A. The *closure* of A, denoted by $\bar{A}$, is the union of A and its boundary, i.e., $\bar{A} = A \cup \partial A$. A set $A \subset \mathbb{R}^n$ is called *robust* if

$$\overline{\overset{\circ}{A}} = \bar{A}, \tag{2.41}$$

which means that every boundary point of A can be approached from the interior. A set $S \subset \mathbb{R}^n$ is *disconnected* if there exist open sets O_1, O_2 such that $S \subset O_1 \cup O_2$, $S \cap O_1 \neq \emptyset$, $S \cap O_2 \neq \emptyset$, and $S \cap O_1 \cap O_2 = \emptyset$. Otherwise, S is *connected*.

2.2.2 The Set of Interest and its Topology

The following discussion shows the main challenge to maintain a finite size approximation of $P_{Q,\epsilon}$, namely that it forms an n-dimensional set. For this, let $x_0 \in P_Q$ be given (such a point exists as Q is compact). That is, there exists no $y \in Q$ such that $y \prec x_0$. Since F is continuous and $\epsilon \in \mathbb{R}^k_+$ there exists a neighborhood U of x_0 such that

$$\nexists y \in Q : \; y \prec_{-\epsilon} u \quad \forall u \in U \cap Q, \tag{2.42}$$

and thus, $U \cap Q \subset P_{Q,\epsilon}$, and we are done since U is n-dimensional.

Before we can investigate the relation of $P^c_{Q,\epsilon}$ and $P_{Q,\epsilon}$ we have to state some results on the latter set.

Theorem 6. *Let $Q \subset \mathbb{R}^n$ be compact and*

(A1) let there be no weak Pareto point in $Q \backslash P_Q$,

(A2) let there be no $-\epsilon$ weak Pareto point in $Q \backslash \overline{P_{Q,\epsilon}}$, and

(A3) let $\mathcal{B} \subset \overset{\circ}{Q}$ and $q(x) \neq 0$ for all $x \in \mathcal{B}$, where q is as defined in Theorem 5 and $\mathcal{B} := \{x \in Q | \exists y \in P_Q : F(y) + \epsilon = F(x)\}$.

Then it holds:

$$\begin{aligned}
\overline{P_{Q,\epsilon}} &= \{x \in Q | \; \nexists y \in Q : F(y) + \epsilon <_p F(x)\}, \\
\overset{\circ}{P}_{Q,\epsilon} &= \{x \in Q | \; \nexists y \in Q : F(y) + \epsilon \leq_p F(x)\}, \\
\partial P_{Q,\epsilon} &= \{x \in Q | \exists y_1 \in P_Q : F(y_1) + \epsilon \leq_p F(x) \wedge \nexists y_2 \in Q : F(y_2) + \epsilon <_p F(x)\}.
\end{aligned} \tag{2.43}$$

The result shows that $P_{Q,\epsilon}$ is in general neither open nor closed, and that the closure gets 'completed' by $-\epsilon$ weak Pareto points. Further, Equation (2.43) can be used to characterize the boundary: as there exists a $j \in \{1, \ldots, k\}$ such that $f_j(y_1) + \epsilon_j = f_j(x)$, boundary points are $-\epsilon$ weak Pareto points that are bounded from $F(P_Q)$ by ϵ.

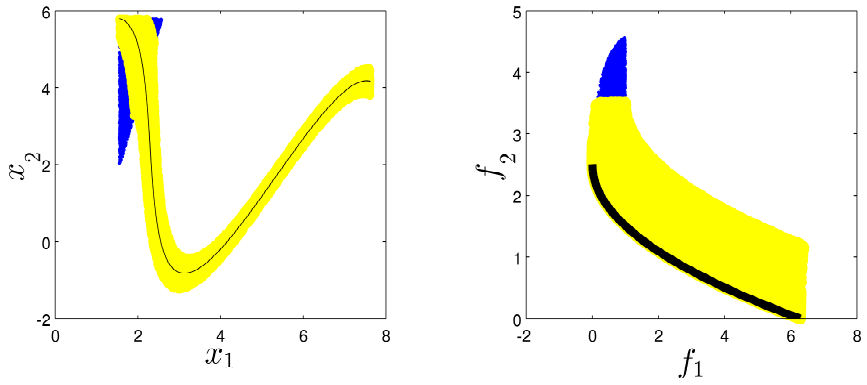


Figure 2.7: Left: P_Q (black),$P^c_{Q,\epsilon}$ (black and yellow), and $P_{Q,\epsilon}$ (black, yellow and blue) for OKA1. Right: the respective sets in objective space.

Now we are able to examine the relation between $P^c_{Q,\epsilon}$ and $P_{Q,\epsilon}$. It is in general $P_{Q,\epsilon} \not\subset P^c_{Q,\epsilon}$ (such an example can be found in White (1986) which holds also for $-\epsilon$-dominance). Further, $P_{Q,\epsilon}$ is in general a proper superset of $P^c_{Q,\epsilon}$.

Lemma 7. $P^c_{Q,\epsilon} \subset \overset{\circ}{P}_{Q,\epsilon}$

Example

(a) Figure 2.7 shows the different sets for the two-dimensional bi-objective problem OKA1 (Okabe et al. (2004)): left, black marks the Pareto set, yellow marks the points that are nearly optimal according to $P^c_{Q,\epsilon}$ (i.e., $P^c_{Q,\epsilon}$ is the union of the black and the yellow region), and blue marks the points that are included in $P_{Q,\epsilon}$ but not in $P^c_{Q,\epsilon}$. Right, the respective sets are shown in image space using the same colors.

(b) Let $F : \mathbb{R} \to \mathbb{R}^2, \quad F(x) = ((x-1)^2, (x+1)^2)^T$, $\epsilon = (1,1)^T$ and $Q = [-3,3]$. Then it is $P_Q = [-1,1]$ and $P_{Q,\epsilon} = (-2,2)$. The boundary $\partial P_{Q,\epsilon}$ is given by $\{x_1 := -2, x_2 := 2\}$. Since $F(x_1) = (9,1)$, $F(x_2) = (1,9)$, $F(-1) = (4,0)$ and $F(1) = (0,4)$ both points x_1 and x_2 are $-\epsilon$-dominated and thus not included in $P_{Q,\epsilon}$.

(c) Figure 2.8 shows an example for $k = 1$ where the set of interest is given by $P_{Q,\epsilon} = \{x^*\} \cup [c,d]$, i.e., the set is (i) disconnected and (ii) contains the isolated point x^*.

Example (c) shows that $P_{Q,\epsilon}$ can contain isolated points which can practically not be found or approximated via stochastic search. By Lemma 7 it follows that this is—despite for theoretical investigations—not problematic since its interior, which can be approximated in any case, already contains all the interesting parts. However,

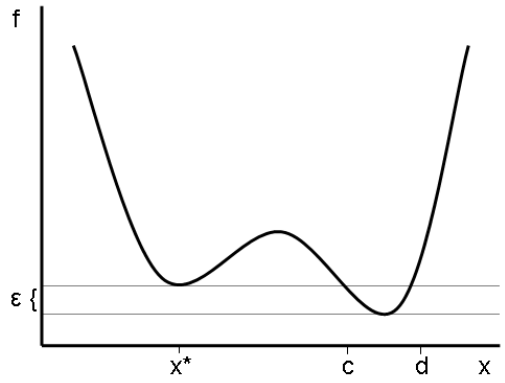


Figure 2.8: Example of a set $P_{Q,\epsilon}$ where the closure of its interior is not equal to its closure.

since we are interested in convergence results measured by the Hausdorff distance we have to make further assumptions on the model which will do in the following.

Lemma 8. *Let in addition to the assumptions made in Theorem 6 be* $q(x) \neq 0 \; \forall x \in \partial P_{Q,\epsilon}$*, then*

$$\overline{\overset{\circ}{P}_{Q,\epsilon}} = \overline{P_{Q,\epsilon}}. \tag{2.44}$$

Roughly speaking, $P_{Q,\epsilon}$ is under the assumptions made in Lemma 8 not 'flat' anywhere. As such an example consider the filled letter 'six' as A. The interior of A is then an open circle, and its closure the filled circle, i.e., the upper arc of A is missing.

The last topological property of $P_{Q,\epsilon}$ we investigate here is its connectedness. The connectedness of a set is an important property, in particular when tackling the problem with local search strategies: in that case, the entire set can be detected when starting with one single solution. Example 2.2.2 (c) shows that the connectedness of $P_{Q,\epsilon}$ cannot be expected in general. However, in the convex case the following holds.

Theorem 9. *Let* $\epsilon \in \mathbb{R}^k_+$*. If* Q *is convex and all* f_i*,* $i = 1, \ldots, k$*, are convex, then* $P_{Q,\epsilon}$ *and* $F(P_{Q,\epsilon})$ *are connected.*

2.2.3 Computing the Entire Set of Nearly Optimal Solutions

To the best of our knowledge all the works dealing with approximate solutions in multi-objective optimization are based on the concept of ϵ-dominance (Loridan (1984); White (1986)). ϵ-dominance or ϵ-efficiency has been studied and used by many researchers, e.g. to allow (or tolerate) nearly optimal solutions (Loridan (1984), White (1986)), to approximate the set of optimal solutions P_Q (Ruhe and Fruhwirt (1990)), or in order to discretize this set (Laumanns et al. (2002); Schütze et al. (2008, 2010)). ϵ-efficient solutions have also been used to tackle a variety of real-world problems

including portfolio selection problems (White (1998)), a location problem (Blanquero and Carrizosa (2002)), or a minimal cost flow problem (Ruhe and Fruhwirt (1990)). The numerical approximation of approximate solutions has been addressed in several studies. In most of them, scalarization methods have been employed (e.g., White (1986); Blanquero and Carrizosa (2002); Engau and Wiecek (2007)). By their nature, such algorithms can deliver only single solutions by one single execution.

There already exist quite a few archiving strategies for the use within multi-objective stochastic search algorithms such as the ones presented and analyzed in Hanne (1999); Eiben and Rudolph (1999); Rudolph and Agapie (2000); Knowles and Corne (2000); Laumanns et al. (2002); Knowles and Corne (2004); Schütze et al. (2007b); Schütze et al. (2008); Schütze et al. (2008); Horoba and Neumann (2008); Yuen et al. (2012); Moubayed et al. (2014). However, all of them store sets of mutually non-dominating points, and can hence not be utilized for our purpose.

First we aim for the computation of the entire set $P_{Q,\epsilon}$. This is not only of theoretical interest but also relevant in case the function evaluation is expensive and/or time restrictions just allow to evaluate few candidate solutions. In that case it makes sense to store all promising solutions to obtain the best overview over the gathered data.

Algorithm 2 $A := ArchiveUpdateP_{Q,\epsilon}\,(P, A_0, \epsilon)$

Require: population P, archive A_0
Ensure: updated archive A
1: $A := A_0$
2: **for all** $p \in P$ **do**
3: **if** $\nexists a \in A : \; a \prec_{-\epsilon} p$ **then**
4: $A := A \cup \{p\}$
5: **end if**
6: **for all** $a \in A$ **do**
7: **if** $p \prec_{-\epsilon} a$ **then**
8: $A := A \backslash \{a\}$
9: **end if**
10: **end for**
11: **end for**

Algorithm 2 shows the pseudo code of $ArchiveUpdateP_{Q,\epsilon}$. This archiver keeps all points that are not $-\epsilon$-dominated by any other considered candidate. First, we investigate the behavior of the sequence of archives for finitely many iterations. One possible problem archiving strategies have to cope with is cycling or deterioration in their entries (e.g., Hanne (1999)). This is of course an unwanted effect since this implies that information can get 'lost'. The following result, however, shows that this is not the case for Algorithm 2.

Theorem 10. *Let $l \in \mathbb{N}$, $\epsilon \in \mathbb{R}^k_+$, $P_1, \ldots, P_l \subset \mathbb{R}^n$ be finite sets, and $A_i, i = 1, \ldots, l$,*

be obtained by ArchiveUpdate$P_{Q,\epsilon}$ as in Algorithm 1, and $C_l = \bigcup_{i=1}^{l} P_i$. Then

$$A_l = P_{C_l,\epsilon} = \{x \in C_l \ : \nexists y \in C_l : y \prec_{-\epsilon} x\}. \tag{2.45}$$

Next, we investigate the limit behavior of the sequence A_i of archives. To guarantee convergence, we have to make several assumptions on the MOP and the generation process. The requirements on the MOP are given by the results in Sec. 2 while for the generator we have to assume the following (see also Schütze et al. (2008, 2010)):

$$\forall x \in Q \text{ and } \forall \delta > 0 : \quad P\left(\exists l \in \mathbb{N} \ : \ P_l \cap B_\delta(x) \cap Q \neq \emptyset\right) = 1, \tag{2.46}$$

where $P(A)$ denotes the probability for event A and $B_\delta(x)$ the n-dimensional sphere with center x and radius δ. Assumption (2.46) says that every neighborhood $U \cap Q$ of every point gets 'visited' by *Generate*() after finitely many steps with probability one. The following consideration shows that we cannot assume less: if (2.46) does not hold, there exists with probability one a point $x \in Q$ and a neighborhood $\tilde{U} = U \cap Q$ of x such that no candidate solution $p \in P_l$ lies in $\tilde{U}$ for all $l \in \mathbb{N}$. Thus, no convergence can be guaranteed since a part of the Pareto set can be contained in $\tilde{U}$ which is never 'visited'. In the case of evolutionary algorithms (2.46) is ensured by the use of mutation operators such as Polynomial Mutation Deb (2001b).

The following theorem shows that the sequence of archives converges under these conditions with probability one to $P_{Q,\epsilon}$ in the Hausdorff sense.

Theorem 11. *Let (2.1) be given, where F is continuous and is $Q \subset \mathbb{R}^n$ compact, and $\epsilon \in \mathbb{R}^k_+$. Further, let (2.46) be fulfilled. Then, under the assumptions made in Lemma 8, an application of Algorithm 1, where ArchiveUpdate$P_{Q,\epsilon}$ (Algorithm 2) is used to update the archive, leads to a sequence of archives $A_l, l \in \mathbb{N}$, with*

$$\lim_{l\to\infty} d_H(P_{Q,\epsilon}, A_l) = 0, \quad \textit{with probability one.} \tag{2.47}$$

Remark 1. *(a) In order to obtain convergence toward $P_{Q,\epsilon}$ we had to postulate some (mild) assumptions to guarantee (2.44). In the general case, however, we can still expect that the interior of $P_{Q,\epsilon}$ will be approximated in the limit, which contains already all the 'interesting' part due to Lemma 7. More precisely, even if assumptions (A1)–(A3) are not fulfilled, we still have*

$$\lim_{l\to\infty} dist(\overline{\overset{\circ}{P}_{Q,\epsilon}}, A_l) = 0, \quad \textit{with probability one.} \tag{2.48}$$

(b) Note the analogy of the approach proposed above to the straightforward way to approximate the Pareto front: in case all the non-dominated solutions which are found so far are kept in the archive, i.e., if

$$ArchiveUpdateND(A, P) := \{x \in A \cup P : y \not\prec x \ \forall y \in A \cup P\} \tag{2.49}$$

is used, one can show (Schütze et al. (2008)) that under similar assumptions as in Theorem 11 an application of Algorithm 1 leads to a sequence of archives $\{A_i\}_{i\in\mathbb{N}}$, such that

$$\lim_{l\to\infty} d_H(F(P_Q), F(A_l)) = 0 \quad \textit{with probability one.} \tag{2.50}$$

Due to the above discussion we can expect that the sequence of archives generated by $ArchiveUpdateP_{Q,\epsilon}$ converges toward the set of interest for a given MOP. However, as this is typically not finite (except for discrete problems), we have to expect that the magnitudes of the archive entries go beyond every threshold during the run of the algorithm. Clearly, this is not admissible for an effective and practical use of the archiver within a stochastic search algorithm, in particular when numerous candidate solutions are being considered during the process (which is normally the case). Hence, suitable discretization strategies are desired which we address in the following sections.

2.2.4 Discretizing the Set either in Decision or Objective Space

Algorithms 3 and 4 show the pseudocodes of the archivers $ArchiveUpdateP_{Q,\epsilon}D_x$ and $ArchiveUpdateP_{Q,\epsilon}D_y$ which are modifications of $ArchiveUpdateP_{Q,\epsilon}$ that aim for discretizations of $P_{Q,\epsilon}$ in decision and objective space, respectively. Hereby, $1\Delta := (\Delta, \ldots, \Delta) \in \mathbb{R}_+^k$, where $\Delta \in \mathbb{R}_+$ can be viewed as the discretization parameter. Since $P_{Q,\epsilon}$ is n-dimensional, a discretization in decision space could lead to tremendous magnitudes of the archives except for small values of n, say, $n \leq 5$ (see also the discussion of the archive bounds). Hence, we concentrate in the following on the analysis of Algorithm 4. As one result, we will show that the upper bound of the archive sizes obtained by this algorithm is basically the same as when aiming for gap free discretizations of the Pareto front.

Algorithm 3 $A := ArchiveUpdateP_{Q,\epsilon}D_x\ (P, A_0, \Delta)$

Require: population P, archive A_0, $\Delta \in \mathbb{R}_+$, $\Delta^* \in (0, \Delta)$
Ensure: updated archive A

```
1:  A := A_0
2:  for all p ∈ P do
3:      if ∄a_1 ∈ A : a ≺_{-ε} p and ∄a_2 ∈ A : d_∞(a_2, p) ≤ Δ* then
4:          A := A ∪ {p}
5:          for all a ∈ A do
6:              if p ≺_{-ε} a then
7:                  A := A\{a}
8:              end if
9:          end for
10:     end if
11: end for
```

First, we investigate the monotonicity behavior of Algorithm 4. An analog statement to Theorem 10 can of course not be expected due to the discretization mechanism in line 3. Example 1 shows that by using $ArchiveUpdateP_{Q,\epsilon}D_y$ one cannot prevent to maintain points $x \in P_{Q,\epsilon+2\Delta}\backslash P_{Q,\epsilon}$ in the (limit) archive, and by Theorem 12 it follows that the error of 2Δ (in image space) is already the upper bound.

Algorithm 4 $A := ArchiveUpdateP_{Q,\epsilon}D_y\ (P, A_0, \Delta)$

Require: population P, archive A_0, $\Delta \in \mathbb{R}_+$, $\Delta^* \in (0, \Delta)$
Ensure: updated archive A

```
A := A_0
for all p ∈ P do
    if ∄a_1 ∈ A : a ≺_{-ε} p and ∄a_2 ∈ A : d_∞(F(a_2), F(p)) ≤ Δ* then
        A := A ∪ {p}
        for all a ∈ A do
            if p ≺_{-(ε+1Δ)} a then
                A := A\{a}
            end if
        end for
    end if
end for
```

Remark 1. *Consider the problem $F : \mathbb{R} \to \mathbb{R}$, $F(x) = x$, and let $Q = [0, 5]$, $\epsilon = 1$, and set $\Delta^* = \Delta = 0.1$. Then, it is $P_{Q,\epsilon} = [0, 1]$. Assume that $A = \{a_1 := 1.2\}$. If further $a_2 = 0.1$ is considered, it will be inserted into the archive as $d_\infty(F(a_1), F(a_2)) > \Delta$ and since $a_2 \in P_{Q,\epsilon}$ is not $-\epsilon$-dominated by a_1 nor by any other point $x \in Q$, and will remain in the archive further on. Since a_2 is not $-(\epsilon + \Delta)$-dominating a_1 we have for the updated archive $A = \{a_1, a_2\}$. Hence, no element $a \in [0, \Delta]$ will be taken to the archive since for these points it holds $d_\infty(F(a), F(a_2)) \leq \Delta^*$, and thus, $a_2 \in P_{Q,\epsilon+2\Delta} \backslash P_{Q,\epsilon}$ will not be discarded from the archive during the run of the algorithm. When on the other side $a_1 = 0$ is taken to the archive, no element $x \in Q \backslash P_{Q,\epsilon}$ will ever be accepted in the sequel by Algorithm 4.*

Theorem 12. *Let $l \in \mathbb{N}$, $\epsilon \in \mathbb{R}_+^k$, $P_1, \ldots, P_l \subset \mathbb{R}^n$ be finite sets, and $A_i, i = 1, \ldots, l$, be obtained by $ArchiveUpdateP_{Q,\epsilon}D_y$ as in Algorithm 1. Then, for $C_l = \bigcup_{i=1}^{l} P_i$ it holds:*

$$\begin{aligned} &(i) \quad dist(F(P_{C_l,\epsilon}), F(A_l)) \leq \Delta \\ &(ii) \quad dist(F(A_l), F(P_{C_l,\epsilon})) \leq dist(F(P_{C_l,\epsilon+2\Delta}), F(P_{C_l,\epsilon})) \\ &(iii) \quad d_H(F(P_{C_l,\epsilon}), F(A_l)) \leq \max(\Delta, dist(F(P_{C_l,\epsilon+2\Delta}), F(P_{C_l,\epsilon}))) \end{aligned} \tag{2.51}$$

Hence, the approximation error depends next to the discretization parameter Δ on the distance of $F(P_{Q,\epsilon+2\Delta})$ to $F(P_{Q,\epsilon})$. While the former value can be adjusted or is given by the application, the latter is inherent in the problem and can get large in some (pathological) cases (Schütze et al. (2008)).

Next, we investigate the limit behavior of $ArchiveUpdateP_{Q,\epsilon}D_y$.

Theorem 13. *Let (2.1) be given, where F is continuous and Q is compact, and $\epsilon \in \mathbb{R}_+^k$, $\Delta, \Delta^* \in \mathbb{R}_+$ with $\Delta^* < \Delta$. For the generation process we assume (2.46) and for the MOP the assumptions made in Lemma 8. Then, an application of Algorithm 1, where $AchiveUpdateP_{Q,\epsilon}(P, A, \Delta)$ is used to update the archive, leads to a sequence $A_l, l \in \mathbb{N}$, with*

(a) For all $l \in \mathbb{N}$ *it holds* $\|F(a_1) - F(a_2)\|_\infty \geq \Delta^*$.

(b) There exists with probability one $l_0 \in \mathbb{N}$ *such that for all* $l \geq l_0$*:*

(b1) $dist(F(P_{Q,\epsilon}), F(A_l)) < \Delta$

(b2) $dist(F(A_l), F(P_{Q,\epsilon})) \leq dist(F(P_{Q,\epsilon+2\Delta}), F(P_{Q,\epsilon}))$

(b3) $d_H(F(P_{Q,\epsilon}), F(A_l)) \leq D$*, where* $D = max(\Delta, dist(F(P_{Q,\epsilon+2\Delta}), F(P_{Q,\epsilon})))$

Remark 2. *(a) For* $\Delta = \Delta^* = 0$ *the archiver coincides with the one presented in Algorithm 2.*

(b) Here we have used a scalar $\Delta_0 \in \mathbb{R}_+$ *to discretize the* ϵ*-efficient front. Analog results can be obtained by using a vector* $\Delta \in \mathbb{R}_+^k$ *leading to different discretizations in each objective.*

(c) $0 < \Delta^* < \Delta$ *is used for theoretical purposes. In practice,* $\Delta^* = \Delta$ *can be chosen.*

Further on we analyze the bounds of the archive sizes obtained by the two archivers. The lower bound for both methods is apparently given by one (e.g., for a sufficiently large value of Δ), more interesting are the upper bounds which represent a potential obstacle for the efficiency of the related stochastic search algorithms.

Theorem 14. *Let* $\epsilon \in \mathbb{R}_+^k$, $\Delta^*, \Delta \in \mathbb{R}_+$ *with* $\Delta^* < \Delta$ *be given. Further, let* $m_i = \min_{x \in Q} f_i(x)$, $M_i = \max_{x \in Q} f_i(x)$, $1 \leq i \leq k$, *and* l_0 *as in Theorem 11. Then, when using* $ArchiveUpdateP_{Q,\epsilon}$*, the archive size maintained in Algorithm 1 for all* $l \geq l_0$ *is bounded by*

$$|A_l| \leq \left(\frac{1}{\Delta^*}\right)^k \sum_{i=1}^{k} (\epsilon_i + 2\Delta + \Delta^*) \prod_{\substack{j=1 \\ j \neq i}}^{k} (M_j - m_j + \Delta^*). \tag{2.52}$$

One interesting aspect is the growth of the limit archives w.r.t. the discretization parameter Δ. Assuming that Δ is smaller than any entry of ϵ (i.e., $\epsilon_i = c_i \Delta$ for $c_i > 1$) and setting $\Delta = \Delta^*$ we obtain by (2.52)

$$|A_l(D_y)| \leq \left(\frac{1}{\Delta}\right)^{k-1} \sum_{i=1}^{k} (c_i + 3) \prod_{\substack{j=1 \\ j \neq i}}^{k} (M_j - m_j + \Delta^*), \tag{2.53}$$

where $A_l(D_y)$ denotes here for sake of a better distinction the archive obtained via $ArchiveUpdateP_{Q,\epsilon}D_y$. Thus, the magnitudes are or the order

$$|A_l(D_y)| \in \mathcal{O}\left(\left(\frac{1}{\Delta}\right)^{k-1}\right) \quad \text{for} \quad \Delta \to 0 \text{ and } l \to \infty. \tag{2.54}$$

Considering that the Pareto set which is included in $P_{Q,\epsilon}$ typically forms a $(k-1)$-dimensional set, we can say that the order of $|A_l(D_y)|$ is already optimal.

An analog result for a discretization in decision space, however, can not hold since $P_{Q,\epsilon}$ is n-dimensional. To be more precise, the upper bound for the magnitude of the archive when using $ArchiveUpdateP_{Q,\epsilon}D_x$ is given by

$$|A_l(D_x)| \leq \left(\frac{1}{\Delta^*} + 1\right)^n \prod_{j=1}^{n}(b_i - a_i), \tag{2.55}$$

where $Q \subset [a_1, b_1] \times \ldots \times [a_n, b_n]$ ($P_{Q,\epsilon+2\Delta}$ is certainly included in $[a_1, b_1] \times \ldots \times [a_n, b_n]$, and maximal $1/\Delta^* + 1$ elements can be placed in each coordinate direction).

2.2.5 Computing approximations of the set of nearly optimal solutions

In the following, we briefly describe some the proposed methods to find approximations of $P_{Q,\epsilon}$.

- Schütze et al. (2007a) proposed to couple a generic evolutionary algorithm (as the one shown in Algorithm 1) with the $ArchiverUpdateP_{Q,\epsilon}$ (Algorithm 2). The method was tested on Tanaka and sym-part (Rudolph et al., 2007) problems to show the capabilities of the method.

- Schütze et al. (2008) presented a modified version of NSGA-II, named $P_{Q,\epsilon}$-NSGA-II. The method uses an external archiver to maintain a representation of $P_{Q,\epsilon}$ and tested the method on a space machine design application. Later, in Schütze et al. (2009) the method was applied to solve a two impulse transfer to asteroid Apophis and a sequence Earth-Venus-Mercury (EVMe) problem.

- Schütze et al. (2011) proposed the $P_{Q,\epsilon}$-MOEA which is a steady-state archive-based MOEA that uses $ArchiverUpdateP_{Q,\epsilon}$ for the population. The algorithm draws at random two individuals from the archiver and generates two new solutions that are introduced to the archiver.

- Hernández et al. (2013) presented an adaption of the simple cell mapping method to find the set of approximate solutions using $ArchiverUpdateP_{Q,\epsilon}D_y$.

- Xia et al. (2014b) designed an algorithm for fault tolerance applications. The method uses two different kinds of problem dependent mutation and one for diversity. In this case, a child will replace a random individual if it is better or with probability one half if they are non-dominated. The method uses a bounded version of $ArchiverUpdateP_{Q,\epsilon}$ at the end of each generation.

2.3 Uncertainty in Optimization

In the following, we present some ideas that were introduced for uncertainty in optimization.

2.3.1 Sources of uncertainty

First, we review the sources of uncertainty according to engineering, optimization and evolutionary algorithms.

Uncertainties in engineering modeling

In engineering modeling there exist a classification according to the source of uncertainty Kiureghian and Ditlevsen (2007); Beyer and Sendhoff (2007). This is taken from an epistemological perspective differentiating the uncertainties into objective and subjective ones.

- Aleatory: is one that is presumed to be the intrinsic randomness of a phenomenon. Aleatory uncertainty is of intrinsically irreducible stochastic nature. The designer has to optimize his design according to this reality. Due to the probabilistic nature, probability distributions are the adequate means for the mathematical description of these uncertainties.

- Epistemic: is one that is presumed as being caused by lack of knowledge (or data) a designer has about the problem of interest (e.g. discretization error, approximation error, convergence problems).

The reason that it is convenient to have this distinction within an engineering analysis model is that the lack-of-knowledge-part of the uncertainty can be represented in the model by introducing auxiliary non-physical variables. These variables capture information obtained through the gathering of more data or use of more advanced scientific principles. An uttermost important point is that these auxiliary variables define statistical dependencies (correlations) in a clear and transparent way.

Uncertainty in Optimization

In the following we present the classification of uncertainty as presented in Beyer and Sendhoff (2007).

A Changing environmental and operating conditions. Examples are the angle of attack in airfoil design, operating temperature, pressure, humidity, changing material properties, among others.

B Production tolerances and actuator imprecision. The design parameters of a product can be realized only to a certain degree of accuracy. High precision machinery is expensive, therefore, a design less sensitive to manufacturing tolerances reduces costs.

C Uncertainties in the system output. These uncertainties are due to imprecision in the evaluation of the system output and the system performance. This kind of uncertainty includes measuring errors and all kinds of approximation errors due to the use of models instead of the real physical objects (model errors).

D Feasibility uncertainties. Uncertainties concerning the fulfillment of constraints the design variables must obey. This kind of uncertainty is different to A - C in that it does not consider the uncertainty effects on f but on the design space. In real-world applications it often appears together with the uncertainty types A and B.

Uncertainty in evolutionary algorithms

According to Jin and Branke (2005) uncertainty in evolutionary algorithms can be categorized in four classes.

1. Noise: The fitness evaluation is subject to noise. Noise in fitness evaluations may come from different sources such as sensory measurement errors or randomized simulations.

2. Robustness: The design variables are subject to perturbations or changes after the optimal solution has been determined. Therefore, a common requirement is that a solution should still work satisfactorily when the design variables change slightly, e.g., due to manufacturing tolerances. Such solutions are termed robust solutions.

3. Fitness approximation: When the fitness function is very expensive to evaluate, or an analytical fitness function is not available, fitness functions are often approximated based on data generated from experiments or simulations. The approximated fitness function is often known as meta-model.

4. Time-varying fitness functions: The fitness function is deterministic at any point in time, but is dependent on time t. As a consequence, also the optimum changes over time. Thus, the evolutionary algorithm should be able to continuously track the changing optimum rather than requiring a repeated restart of the optimization process.

2.3.2 Modeling uncertainty

According to Beyer and Sendhoff (2007) there are three different possibilities to quantify the uncertainties mathematically. The authors state that uncertainties can be modeled deterministically, probabilistically, or possibilistically.:

1. The deterministic type defines parameter domains in which the uncertainties α, δ, among others can vary,

2. the probabilistic type defines probability measures describing the likelihood by which a certain event occurs, and

3. the possibilistic type defines fuzzy measures describing the possibility or membership grade by which a certain event can be plausible or believable.

2.4 Robust Multi-objective Optimization

Now, we introduce the definition of a robust multi-objective optimization problem according to several authors.

2.4.1 Approaches to include robustness to an MOP

Here, we present some the ideas used to introduce robustness into an MOP.

Substitute the objective

One of the first approaches to handle robustness is to replace the nominal MOP with a measure that reflects uncertainty. Typical approaches that follows this approach are to use: mean, standard deviation or the worst case of the nominal function.

Adding an extra objective

Another option is to use one (or several) objective(s) that reflects the robustness of a solution. For instance, one could use a combination of the previous approaches with the nominal function. One drawback of the approach is that the number of objectives grow quickly (k times the number of additional objectives).

Adding a constraint

A third possibility is to restrict the search to solutions with a minimum accepted value of robustness. However, it is not always clear what should this value be and thus it needs further information on the given MOP.

2.4.2 Worst case multi-objective optimization

First, we define an decision uncertain multi-objective optimization problem Eichfelder et al. (2017). Here, we assume that uncertainties in the problem formulation are given as scenarios from a known uncertainty set $U \subseteq \mathbb{R}^n$. It is also assumed that $F : Q \times U \to \mathbb{R}^k$.

Definition 11. *An uncertain multi-objective optimization problem* $P(U) = (P(\delta), \delta \in U)$ *is defined as the family of parametrized problems*

$$P(\delta) := \min_{x \in Q} F(x + \delta), \tag{2.56}$$

where $F : Q \times U \to \mathbb{R}^k$ *and* $Q \subseteq \mathbb{R}^n$.

Note that it is not clear what a solution to such family of problems is. Figure 2.9 shows an example of decision uncertainty on MOP 2.6. In the following, we introduce several concepts of robustness.

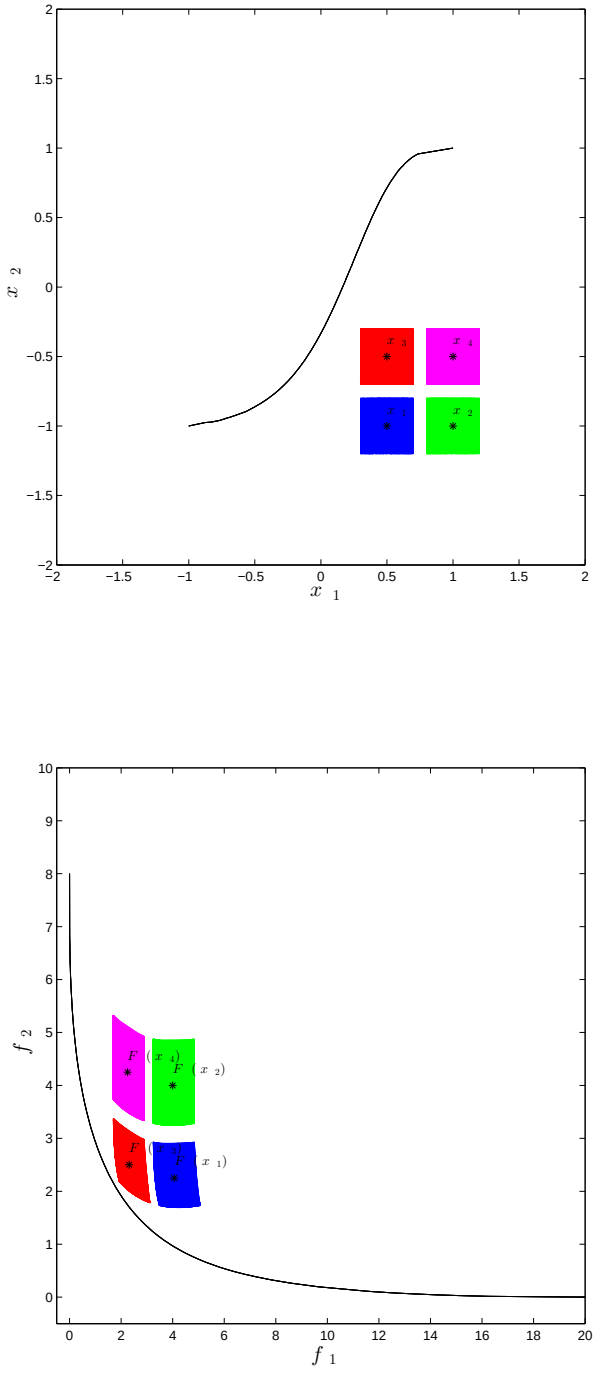


Figure 2.9: Example of decision uncertainty on MOP 2.6. Left shows 4 solutions in decision space and each colored region represents the region of uncertainty. Right shows the image of the 4 solutions. In black, the nominal Pareto set/front is shown .

Point-based MinMax robust efficiency

Kuroiwa and Lee (2012) replaced the objective functions by their respective worst cases over all scenarios and hence obtain a deterministic multi-objective optimization problem whose efficient solutions are called robust. This approach is closely connected to the concept of minmax robustness for single-objective problems.

Definition 12 (Kuroiwa and Lee (2012))**.** *Given an uncertain multi-objective optimization problem $P(U)$, a feasible solution $x \in Q$ is called point-based minmax robust efficient if it is efficient for*

$$\min_{x \in Q} F_U^{\max}. \tag{2.57}$$

Figure 2.10 shows an example of point-based efficiency.

Set-based MinMax robust efficiency

Ehrgott et al. (2014) Ehrgott et al. (2014) proposed an extension of the concept of point-based minmax robust efficiency. Here, for a given feasible solution x, the worst case of the objective vector is interpreted as a set, namely the set of efficient solutions to the multi-objective problem of maximizing the objective function over the uncertainty set.

Definition 13 (Ehrgott et al. (2014))**.** *Given an uncertain multi-objective optimization problem $P(U)$, a feasible solution $\bar{x} \in Q$ is called set-based minmax robust efficient if there is no $x' \in Q \backslash \{\bar{x}\}$ such that*

$$F_U(x') \subseteq F_U(\bar{x}) - \mathbb{R}^k_{\leq}. \tag{2.58}$$

Note that $\mathbb{R}^k_{\leq}$ represents the dominance cone. The robust counterpart of an uncertain multi-objective optimization problem is the problem of identifying all $x \in Q$ which are re. Thus, the robust counterpart problem can be defined as

$$\min_{x \in Q} \sup_{\delta \in U} F(x + \delta), \tag{2.59}$$

where $\sup_{\delta \in U}$ is defined as the set of efficient solutions of the following multi-objective optimization problem

$$\max_{\delta \in U} F(x + \delta). \tag{2.60}$$

Figure 2.10 shows an example of set-based efficiency.

Lightly robust efficiency

One of the main criticism to previous definitions is that it could be over-conservative since it considers the worst case. Thus, solutions with a poor performance in terms of their objective functions could be selected. As a possible remedy, in Ide and Schöbel (2016) the authors extended the notion of lightly robust solutions to the

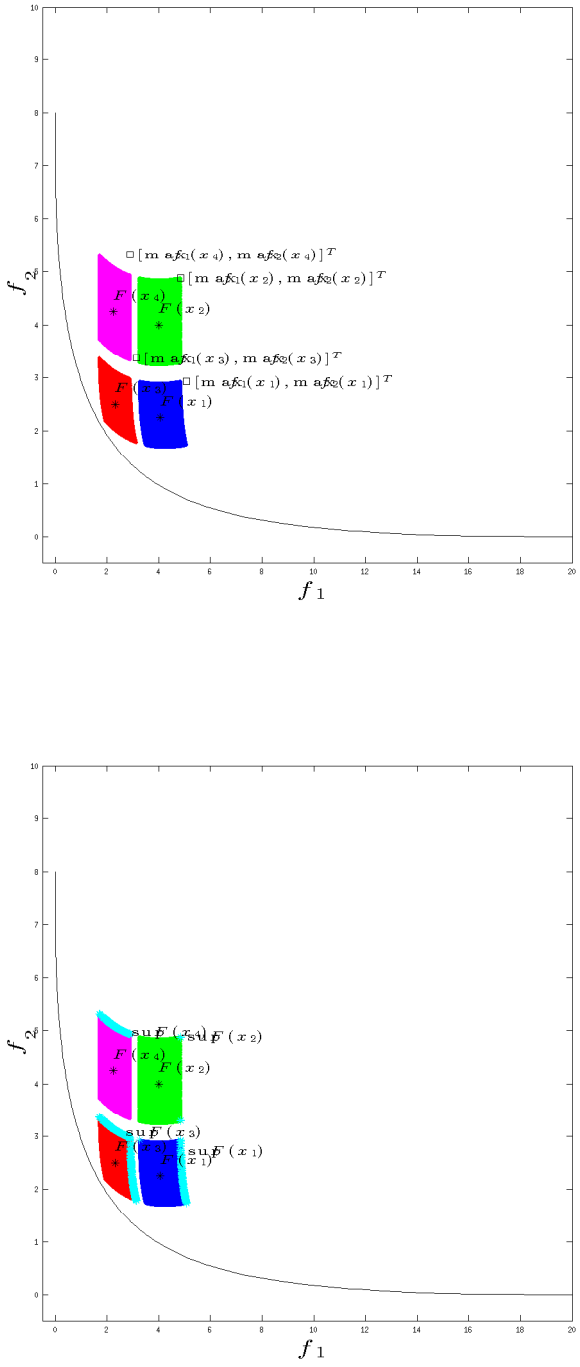


Figure 2.10: Example of point-based minmax (left)/right-based minmax (left)/ robust efficiency on MOP 2.6. In black, the nominal Pareto front is shown . The different colors represent the region of uncertainty in objective space. The squares (left)/light blue represents the worst case. In both cases, x_1, x_3, x_4 are mutually non-dominated.

multi-objective context. Given a nominal scenario $\hat{\delta} \in U$, let $Q_\delta(\hat{\delta})$ be the set of efficient solutions $P(\hat{\delta})$. For each efficient solution $\hat{x} \in Q_\delta(\hat{\delta})$ to $P(\delta)$ and some given $0 \leq \epsilon \in \mathbb{R}^k$ we define the uncertain multi-objective optimization problem $LR(\hat{x}, \epsilon, U) := LR(\hat{x}, \epsilon, \delta), \delta \in U$, as the family of parametrized, deterministic multi-objective optimization problems.

$$\begin{aligned} LR(\hat{x}, \epsilon, \delta) := &\min F(x + \delta) \\ &s.t. F(\hat{x}, \hat{\delta}) \in P^c_{Q,\epsilon}. \end{aligned} \tag{2.61}$$

Definition 14. *Given an uncertain multi-objective optimization problem $P(U)$ with nominal scenario $\delta \in U$ and some $\epsilon \in \mathbb{R}^k_{\geq}$. Then a solution $\bar{x} \in Q$ is called lightly robust efficient for $P(U)$ w.r.t. ϵ if it is set-based minmax robust efficient for $LR(\hat{x}, \epsilon, U)$ for some $\hat{x} \in Q_\delta(\hat{\delta})$.*

Thus, the robust counterpart of this uncertain multi-objective optimization problem is the problem of identifying all $x \in Q$ which are lre. Thus, the robust counterpart problem can be defined as

$$\min_{x \in P^c_{Q,\epsilon}} \sup_{\delta \in U} F(x + \delta). \tag{2.62}$$

Figure 2.11 shows an example of lightly robust efficiency.

2.4.3 Other definitions and related concepts

In this section, we discuss several alternative definitions for robustness as well as other related concepts.

Alternative definitions

Here, we present other definitions that have been used in the context of robust multi-objective optimization.

Deb and Gupta (2006) extended the idea of Branke (1998) to the context of multi-objective optimization. The idea is to replace the objective functions by their respective mean over a given neighborhood or to add a constraint to the MOP. Then, the authors use NSGA-II Deb et al. (2002) to solve the new problem. The main drawback is the high number of function evaluations (30 times higher that a regular MOEA). In the following, we present the two types of robust solutions proposed by the authors.

Definition 15 (Multi-objective Robust Solution of Type I Deb et al. (2005))**.** *A solution x^* is called a multi-objective robust solution of Type I, if it is the global feasible Pareto optimal solution to the following multi-objective minimization problem. Which is defined with respect to a δ-neighborhood ($B_\delta(x)$ of a solution x):*

$$\min_{x \in Q} (f_1^{eff}(x), f_2^{eff}(x), \ldots, f_k^{eff}(x)), \tag{2.63}$$

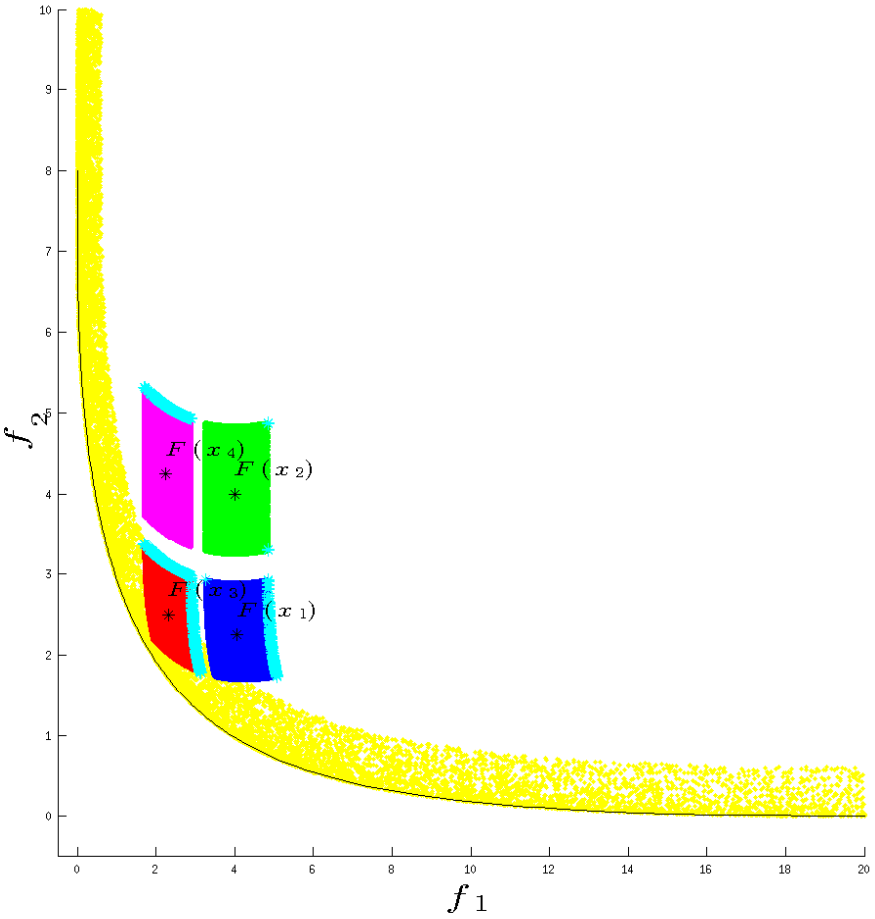


Figure 2.11: Example of lightly robust efficiency on MOP 2.6 with $\epsilon = [0.6, 0.6]^T$. In black, it is shown the set of nearly optimal solutions $P_{Q,\epsilon}$. The different colors represent the region of uncertainty in objective space. The light blue represents the worst case. In this case, only $x_3 \in P_{Q,\epsilon}$ and thus is the lightly robust efficient solution.

where $f_j^{eff}(x)$ *is defined as follows:*

$$f_j^{eff}(x) = \frac{1}{|B_\delta(x)|} \int_{y \in B_\delta(x)} f_j(y) dy. \tag{2.64}$$

Definition 16 (Multi-objective Robust Solution of Type II Deb et al. (2005))**.** *A solution* x^* *is called a multi-objective robust solution of Type II, if it is the global feasible Pareto optimal solution to the following multi-objective minimization problem:*

$$\begin{aligned} &\min_{x \in Q} \ (f_1(x), f_2(x), \ldots, f_k(x)), \\ &s.t. \ \frac{||F^p(x) - F(x)||}{||F(x)||} \leq \eta. \end{aligned} \tag{2.65}$$

Although these definitions do not follow the classical concepts of robustness for single-objective optimization (Ide and Schöbel, 2016), these definitions have attracted special interest in the field of evolutionary computation. In the following, we review some of this works.

- Bader and Zitzler (2010) studied how to incorporate robustness when using the hypervolume indicator within an evolutionary algorithm. They also proposed an extension of the hypervolume indicator that enables more general trade-offs between objective values and robustness of a solution.
- Saha et al. (2011) proposed several techniques in order to lower the number of function evaluations. They proposed to use an external archive to minimize the need for new function evaluations. Further, the method only evaluate the neighborhood of promising solutions and also to use a different sampling method than Latin hypercube sampling.
- Mendes et al. (2013) proposed to use a surrogate model to make an estimation of the neighborhood of a solution to lower the number of function evaluations needed.

For a further review on definitions of robust multi-objective optimization the interested reader is referred to Ide and Schöbel (2016).

Sensitivity

Gunawan and Azarm (2005) presented a method to measure the multi-objective sensitivity of a design alternative, and an approach to use such a measure to obtain multi-objective robust Pareto optimum solutions. The sensitivity measure does not require a presumed probability distribution of uncontrollable parameters and does not utilize gradient information.

Barrico and Antunes (2006) presented the concept of degree of robustness and use it in an evolutionary algorithm. In the method, non-dominated solutions are classified according to their degree of robustness. This information is provided to support the decision maker in the selection a robust compromise solution.

2.5 Cell Mapping Techniques

The cell mapping method was originally proposed by Hsu Hsu (1980, 1987) for the global analysis of non-linear dynamical systems in the decision space. The cell mapping methods have been extensively studied, namely, the simple cell mapping, the generalized cell mapping Hsu (1987), the interpolated cell mapping Tongue (1988), the adjoining cell mapping Zufiria and Guttalu (1993); Guttalu and Zufiria (1993), the hybrid cell mapping Martínez-Marín and Zufiria (1999), among others. The cell mapping methods have been applied to optimal control problems of deterministic and stochastic dynamic systems Hsu (1985); Bursal and Hsu (1989); Zufiria and Martínez-Marín (2003); Gomez et al. (2008). In Zufiria and Guttalu (1993), the cell mapping techniques where combined with dynamical systems theory in order to find all solutions to a system of non-linear algebraic equations.

The cell mapping methods transform the point-to-point dynamics into a cell-to-cell mapping by discretizing the space. The simple cell mapping (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 equal to the number of cells in the group. The periodic cells represent approximate 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 Hsu Hsu (1987).

2.5.1 Dynamical systems

Definition 17 (Dynamical system)**.** *A continuous dynamical system Hsu (1987) can be considered to be a model describing the temporal evolution of a system and it is defined as follows:*

$$\dot{x} = G(x),$$

where x is a N-dimensional vector and $G : \mathbb{R}^N \to \mathbb{R}^N$ is, in general, a non-linear vector function. The evolution of such a dynamical system can be described by a function of the form:

$$x_{m+1} = G(x(m), \mu), \tag{2.66}$$

where x is a N-dimensional vector, m denotes the mapping step, μ is a parameter vector, and G is a general non-linear vector function.

In this case ordinary differential equations can be used to describe the dynamical systems. These are defined as follows:

$$\dot{x} = F(x, t, \mu);\ x \in \mathbb{R}^n,\ t \in R,\ \mu \in \mathbb{R}^l,$$

where x is a N-dimensional state vector, t is the time variable, μ is a l-dimensional parameter vector, and F is a vector-valued function of x, t and μ.

Definition 18 (Fixed point)**.** *When the evolution of a dynamical system is made one may find a point that satisfies the following:*

$$x^* = G(x^*, \mu).$$

In this case, x^* is called a fixed point of Equation (2.66).

Definition 19 (Periodic group)**.** *A periodic solution of Equation (2.66) of period l is a sequence of l distinct points $x^*(j),\ j = 1, 2, \ldots, l$ such that*

$$\begin{aligned} x^*(o+1) &= G^o(x^*(1), \mu),\ o = 1, 2, \ldots, l-1, \\ x^*(1) &= G^l(x^*(1), \mu). \end{aligned} \tag{2.67}$$

We say that there exists a periodic solution of period l. Any of the points $x^*(j),\ j = 1, 2, \ldots, l$, is called a periodic point of period l. One can see that a fixed point is a periodic solution with $l = 1$.

Definition 20 (Domain of attraction)**.** *We say $x^*(j)$ is an attractor if there exists a neighborhood U of $x^*(j)$ such that for every open set $V \supset x^*(j)$ there is a $N \in \mathbb{N}$ such that $f^j(U) \subset V$ for all $j \geq N$. Hence, we can restrict ourselves to the closed invariant set $x^*(j)$, and in this case we obtain*

$$x^*(j) = \bigcap_{j \in N} G^j(U).$$

Thus, we can say that all the points in U are attracted by $x^*(j)$ (under iteration of G), and U is called basin of attraction of $x^*(j)$. If $U = \mathbb{R}^n$, then $x^*(j)$ is called the global attractor.

Several kinds of attractors exists, however, only the ones formed by the set of periodic solutions will be considered in this work.

2.5.2 Simple Cell Mapping

Definition 21 (Cell state space)**.** *An N-dimensional cell space S (Hsu, 1987) is a space whose elements are N-tuples of integers, and each element is called a cell vector or simply a cell, and is denoted by z.*

The simplest way to obtain a cell structure over a given Euclidean decision space is to construct a cell structure consisting of rectangular parallelepipeds of uniform size.

Definition 22 (Cell functions)**.** *Let S be the cell decision space for a dynamical system and let the discrete time evolution process of the system be such that each cell in a region of interest $S_0 \subset S$ has a single image cell after one mapping step. Such an evolution process is called Simple Cell Mapping (SCM)*

$$z(n+1) = C(z(n), \mu), z \in \mathbb{Z}^N, \mu \in \mathbb{R}^l, \tag{2.68}$$

where $C : \mathbb{Z}^N \times \mathbb{R}^l \to \mathbb{Z}^N$, and μ is a l-dimensional parameter.

Definition 23 (Periodic group)**.** *A cell z^* which satisfies $z^* = C(z*)$ is said to be an equilibrium cell of the system. Let C^m denote the cell mapping C applied m times with C^0 understood to be the identity mapping. A sequence of l distinct cells $z^*(j), j \in l$, which satisfies*

$$z^*(m+1) = C^m(z^*(1)), m \in l-1, z^*(1) = C^l(z^*(1)), \tag{2.69}$$

is said to constitute a periodic group or P-Group of period l and each of its elements $z^*(j)$ a periodic cell of period l. One can see that an equilibrium cell is a periodic group with $l = 1$.

Definition 24 (Domains of attraction)**.** *A cell z is said to be r steps away from a periodic group if r is the minimum positive integer such that $C^r(z) = z^*(j)$, where $z^*(j)$ is one of the cells of that periodic group.*

The set of all cells, which are r steps or less removed from a periodic group is called the r-step domain of attraction for that periodic group. The total domain of attraction of a periodic group is its r-step domain of attraction with $r \to \infty$.

The unraveling algorithm developed by Hsu (1987) extracts the cyclic groups of cells in the SCM that represent approximate solutions of the steady-state responses of the system including periodic and chaotic motions.

To describe the global properties of the steady-state responses of non-linear dynamical systems the SCM uses the following sets:

- Group motion number (Gr): the group number uniquely identifies a periodic motion; it is assigned to every periodic cell of that periodic motion and also to every cell in the domain of attraction. The group numbers, which are positive integers, can be assigned sequentially.
- Period (Pe): defines the period of each periodic motion.
- Number of steps to a P-group (St): used to indicate how many steps it takes to map a particular cell into a periodic cell.

After the algorithm is executed, each cell is assigned a group number, step number and periodicity number.

According to the previous discussion, the algorithm works as follows: until all cells are processed, the value of the group motion indicates the state of the current cell and it also points out the corresponding actions to the cell.

- A value of $Gr(cell) = 0$ means that the cell has not been processed. Hence, the state of the cell changes to "under process" and then, we follow the dynamical system to the next cell.
- A value of $Gr(cell) = -1$ means that the cell is under process, which means we have found a periodic group, and we can compute the global properties of the current periodic motion.

- A value $Gr(cell) > 0$ means that the cell has already been processed. Hence, we found a previous periodic motion along with its global properties which can be used to complete the information of the cells under process.

Chapter 3

Cell Mapping Techniques for Multi-objective Optimization

In this chapter, we argue that cell mapping techniques are in particular advantageous for the thorough investigation of low dimensional problems. Such problems occur, for instance, in optimal control (Zufiria and Martínez-Marín, 2003; Gomez et al., 2008; Hernández et al., 2013; Xiong et al., 2014). Cell mapping techniques were first introduced in Hsu (1980) for the global analysis of non-linear dynamical systems. They transform classical point-to-point dynamics into a cell-to-cell mapping by discretizing both phase space and the integration time. In particular the phase space discretization bounds the method to a small number of variables that can be considered (say, $n < 10$), but this global analysis offers in turn much more information than other methods. In the context of multi-objective optimization this is in particular the extended set of options that can be offered to the DM after analyzing the model. There are first of all the Pareto set and the set of approximate solutions as motivated above. In particular if there exist several possibilities to obtain the same optimal or nearly optimal performance, other methods have problems to detect them all since the notion of dominance is defined in objective space (and thus, typically only one of these solutions is detected). Further, the entire set of local optima can be identified that also serve as potential backup solutions (Schütze et al., 2011) and that are interesting for landscape analysis (Mersmann et al., 2011). It is important to note that the relevant information about all these sets of interest is available after one single run of the algorithm (together with an ex post analysis of the obtained data).

In this chapter, we will investigate adaptations of the cell mapping techniques to the context of multi-objective optimization where we will concentrate on the computation of optimal and nearly optimal solutions of a given MOP.

3.1 Simple Cell Mapping

In the following, we will present an adaptation to the SCM in order to handle multi-objective optimization problems.

3.1.1 The algorithm for multi-objective optimization problems

First, we need an appropriate dynamical system to run SCM. For this, we propose to utilize *descent directions*. For the bi-objective problems presented in this section, we have used the Lara's bi-objective descent direction 2.15.

Using such system

$$\dot{x}(t) = v(x(t)), \tag{3.1}$$

can now be used since it defines a pressure toward the Pareto set/front of the MOP at hand: $v(x) = 0$ for every (locally) optimal point, and for all other points improvements can be found by integrating (3.1). Thus, the set of locally optimal Pareto points is contained in the global attractor of (3.1).

It remains to discretize the time in (3.1), i.e., to define a 'suitable' step size t for the related discrete dynamical system

$$x_{i+1} = x_i + t\nu(x_i). \tag{3.2}$$

This is in general not an easy task as we have two conflicting aims. On the one hand, we would like to choose a step size t that lowers all objective functions as much as possible for a given descent direction ν. On the other hand, it is desired to make this decision as cheap as possible in terms of computing time and number of function evaluations. One option is to use an inexact step size control as the one proposed in Fliege and Fux Svaiter (2000).

Here, we can take advantage of the particular setting of the multi-objective SCM. Most importantly, we have the size $h_i = (ub_i - lb_i)/N_i$ for $i = 1, \ldots, n$ of the cells and know that the initial point is the center of a cell. Using this, we already have a value for sufficient decrease. If there exists a $t\nu_i \geq \frac{h_i}{2},\ i = 1, \ldots, n$, then we ensure that we leave the current cell, which is required for the SCM in case the cell does not contain a part of the Pareto set. Now, to decide if the step size t is accepted, we can use a dominance test. We are left with the choice of the initial step size t_0. In the current context, it is promising to compute the distance to the nearest neighbor cell given the descent direction ν from the current cell center. Thus, we suggest taking (compare to Figure 3.1):

$$t = \max\left(\frac{h_i}{\nu_i}\right) + \epsilon,\ \forall i | \nu_i \neq 0. \tag{3.3}$$

We used this approach in the present work, with good computational results. Alternatively, one could use a more sophisticated method such as the one presented in Zufiria and Guttalu (1993). The authors of this work propose an adaptive integration scheme which either finds a neighboring cell or stays in the same cell.

In both cases, although a bigger value of t_0 may lead to a bigger decrease in the objective function, this value of t_0 is enough to leave the current cell, and we have several advantages. We would lose less information since we would be moving to a

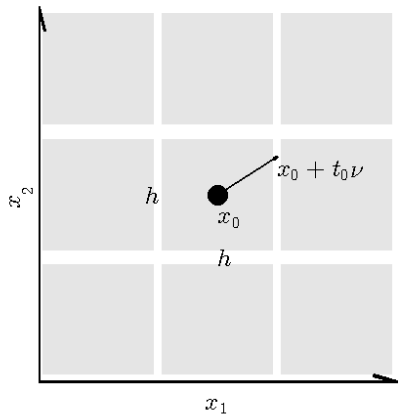


Figure 3.1: Illustration of the setting of the step size control problem for the SCM method.

neighbor cell. Also, with this step size control we would be in the frontier between the current cell and its neighbor, thus if the step size t_0 is not accepted there is no need to use backtracking. Given that we would not be able to leave the current cell and also, since all cells are visited in the SCM method the advantages that bigger step sizes would have by going to an optimal solution with less function evaluations would be lost.

Inequality constraints are handled in the following (straightforward) way: if the center point x_i of cell i is violating any constraint, it will be discarded (i.e., mapped to the sink cell), else, the point will be mapped as described above. The inclusion of more sophisticated constraint handling techniques including the adequate treatment of equality constraints is the subject of ongoing study.

Algorithm 5 shows the pseudo code of the cell mapping technique for the treatment of MOPs that contains the above discussion. Figure 3.2 gives some insight into the behavior of the SCM using the MOP CONV2 (see Table A.1) on a 10×10 grid. The figure shows the result of the SCM after $1, 3, 10$ and 50 iterations in cell space. First, we look at the cell located in $(1, 1)$, which has been taken as the starting cell. Next, we can follow the mapping from this cell by following its arrow. These arrows are formed as follows: We take the center point of the current cell, then we apply the dynamical system (e.g., the descent direction method that we have chosen) on the center point and finally, with the new solution found, we compute to which cell it belongs. In our example, the path is formed by the cells $(1, 1), (2, 1), (3, 2), (4, 3)$, and $(5, 4)$. Cell $(5, 4)$ is an end cell in this case, since there is not an arrow from this cell to another cell, which means we have a periodic group of 1. All the cells processed belong to the same domain of attraction and, therefore, they should have the same group number. Since this is the first group motion discovered, we assign to it the group number 2 (the group number 1 is reserved for those periodic motions that go to the sink cell). Once we have the global properties of those cells, we have to choose

a new starting cell. Since the cell $(2,1)$ has already been processed, we skip it and continue with the cell $(3,1)$. The mapping of this cell also finishes in the cell $(5,4)$. Thus, this cell together with the new path should have the same group number as before (group number 2).

Then, we choose a new starting cell and continue until we finish processing all the cells. As we process the cells, we gather more information of the problem. For this example we have 8 periodic motions with the same number of optimal solutions.

After one run of the SCM the information of the sets of interest can be extracted.

3.1.2 Computing the Pareto set

Since the Pareto set of a MOP is contained in the global attractor of the dynamical system that is derived from a descent direction, all cells with periodic groups are at first point interesting. That is, such cells can potentially contain a part of the Pareto set. It is important to note that due to the properties of the dynamical system periodic groups with size larger than 1 should not appear, however, due to the discretization both in space and time exactly this happens (i.e., oscillations around Pareto optimal solutions can be observed leading to such periodic groups). Hence, we also consider those cells as candidates. The collection of those cells form the candidate set.

This collection can then be further investigated (e.g., via a more fine grain cell mapping or via subdivision techniques), or an approximation of the Pareto set can be directly determined via the center points of the boxes (e.g., via a non-dominance test). Technically speaking, we introduce a set called cPs (see Algorithm 5). Candidate optimal cells are thus those cells with $St = 0$ and $Gr \neq 1$. $St = 0$ means they are part of a periodic group and $Gr \neq 1$ ensures we do not add cells that map to the sink cell.

3.1.3 Error estimates

Since SCM evaluates the entire discretized search space in one run of the algorithm, we are able to provide estimations on the maximal error that can occur in the approximation of the set of interest. Since we are particularly interested in errors of the Pareto front (i.e., errors in objective space) the following estimates are based on Lipschitz continuity.

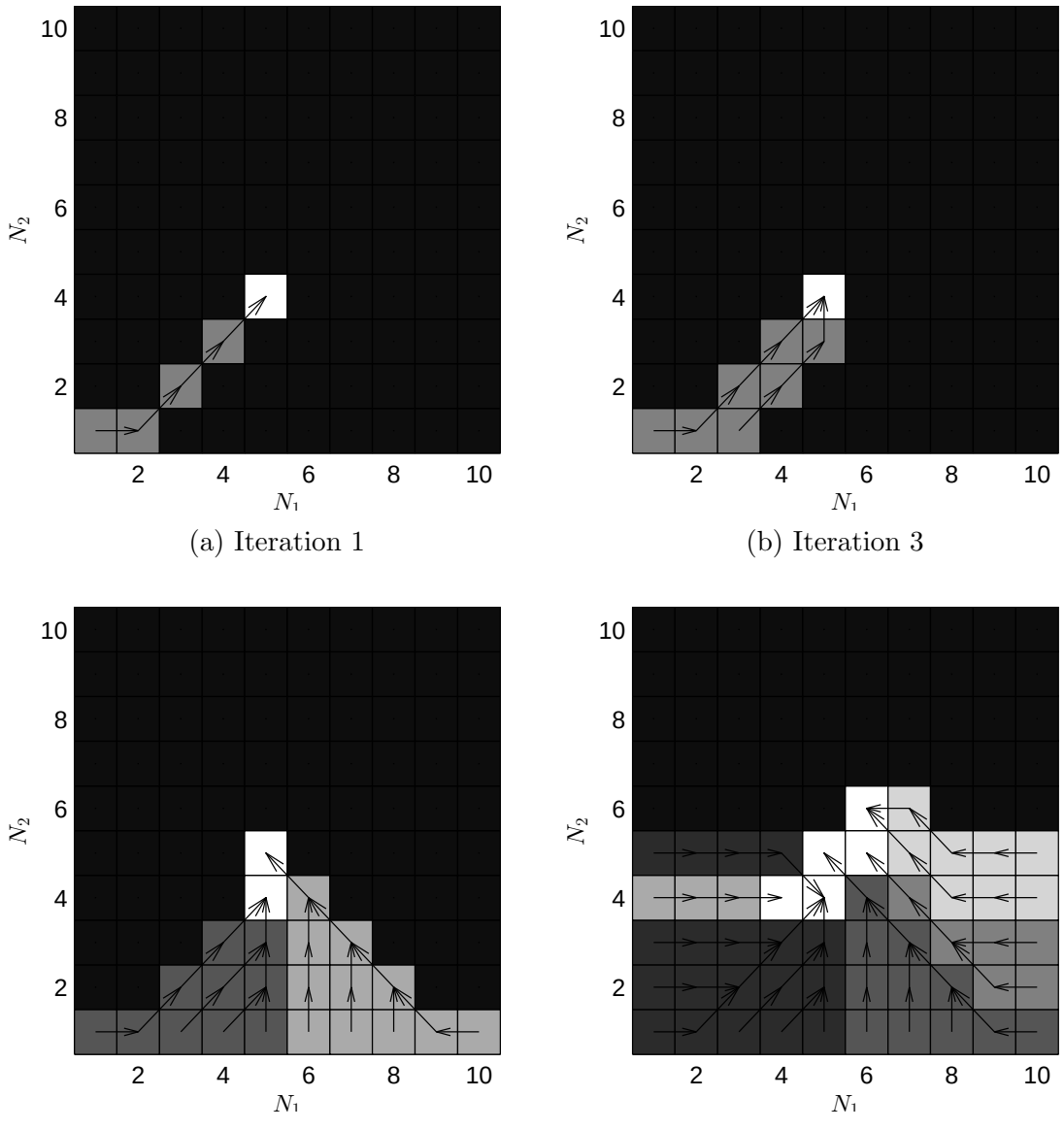
Assume in the following that the objective function F is Lipschitz continuous on each cell, i.e.,

$$\|F(x) - F(y)\| \leq L_{B(c,r)}\|x - y\|, \quad \forall x, y \in B(c,r), \tag{3.4}$$

where

$$B(c,r) := \{y \in \mathbb{R}^n \ : \ |c_i - y_i| \leq r_i, \ i = 1, \ldots, n\} \tag{3.5}$$

is a cell (or generalized box) with center c and radius r, and $L_{B(c,r)}$ is the Lipschitz constant of F within $B(c,r)$. Since SCM evaluates cells at the center c and since the



(a) Iteration 1

(b) Iteration 3

(c) Iteration 10

(d) Iteration 50

Figure 3.2: Iteration of the SCM. The white cells represent the optimal solutions found so far. The arrows show the path from the starting cell to an optimal solution of the MOP. The darker cells represent unexplored regions.

maximal distance on the right hand side of (3.4) is given for vertices of the cell, e.g., $y = c + r$, we can estimate (3.4) at least for unconstrained problems by

$$\|F(c) - F(y)\| \leq L_{B(c,r)}\|r\|, \quad \forall y \in B(c,r). \tag{3.6}$$

The above formula might already be used to measure errors in image space. In the context of multi-objective optimization, however, a potential trouble is that some objectives may be in completely different ranges (e.g., the model SSW that will be considered in the next section). We suggest hence to consider the error bounds for each objective, that is

$$\|F_i(c) - F_i(y)\| \leq L^{(i)}_{B(c,r)}\|r\|, \quad \forall y \in B(c,r),\ i = 1, \ldots, k, \tag{3.7}$$

where $L^{(i)}_{B(c,r)}$ is the Lipschitz constant for objective f_i. If the boxes are small enough, one may approximate this value by the absolute value of the gradient at the center point leading to the estimate

$$E(B(c,r), f_i) := |\nabla f_i(c)|\|r\|, \quad \forall y \in B(c,r),\ i = 1, \ldots, k, \tag{3.8}$$

which we use in this study. As errors for the entire approximation we thus define

$$E_i := \max_{E(c,r) \in Q} E(B(c,r), f_i), \quad i = 1, \ldots, k. \tag{3.9}$$

Complexity of SCM

Note that each cell is visited only twice during the execution of the algorithm. For each cell the following operations are computed: find the center point, compute the descent direction and compute the next cell. All of these operation take constant time with respect the number of cells N_c. Once a cell that was visited before, the method does a backtracking to assign the group number, step number, and period number. Thus, the complexity is given by $O(N_c)$. Further, the number of cells depends on the number of variables n of the problem

$$N_c = \prod_{i=1}^{n} N_i. \tag{3.10}$$

Thus, the complexity with respect to the number of variables n is $O(N^n_{max})$, where $N_{max} = \max(N_i)$ for $i = 1, \ldots, n$.

3.1.4 Numerical results

First we consider the capability of the SCM to compute approximations of the global Pareto set. Figure 3.3 shows some results obtained by SCM on four MOPs (see Appendix A for the description of the problems). CONV2 is a convex bi-objective problem. The Pareto set is a curve connecting the points $(-1, -1)^T$ and $(1, 1)^T$. SSW

is a bi-objective problem whose Pareto set falls into four connected components. Due to symmetries of the model two of these components (the two outer curves on the boundary of the domain) map to the same region in the Pareto front. Finally, CONV3 is a convex tri-objective problem. We have used a 1,000 $\times$ 1,000 grid to perform the cell mapping for the problems with $n = 2$ and a 100 $\times$ 100 $\times$ 100 grid for $n = 3$. In all cases SCM is able to obtain a fine-grain approximation of both Pareto set and front.

Table 3.1 shows the error estimate discussed in the previous section on MOPs considered with different grid sizes together with their IGD values. The match of both values (for each objective value) are (much) better than the E_i values since these describe the worst case scenario.

3.2 Generalized Cell Mapping

While the SCM offers an effective approach to investigate the global properties of a dynamical system for problems with complicated characteristics, we need a more sophisticated algorithm. This is due to the fact that SCM allows only one mapping for each cell, however it could be the case that a cell would map to different cells depending on the point considered. One way is to incorporate more information on dynamics of the system into the cell mapping – which is done in the GCM method. In GCM, a cell z is allowed to have several image cells, being the successors of z. Each of the image cells is assigned a fraction of the total transition probability, which is called the transition probability with respect to z.

The transition probabilities can be grouped into a transition probability matrix P of order $N_c \times N_c$, where N_c is the total number of cells. Then the evolution of the system is completely described by

$$p(n+1) = P \cdot p(n), \tag{3.11}$$

where p is a probability vector of dimension N_c that represents the probability function of the state. This generalized cell mapping formulation leads to absorbing Markov chains (Kemeny and Snell, 1976).

In the following, we introduce some concepts that are useful to our work.

Absorbing Markov chain A Markov chain is absorbing if it has at least one absorbing state, and it is possible to go to an absorbing state from every state (not necessarily in one step).

Classification of cells Two types of cells can be distinguished:

A *periodic cell* i is a cell that is visited infinitely often once it has been visited. In our work, we focus on periodic cells of period 1, i.e., $P_{ii} = 1$. This kind of cells correspond to the local optima candidates.

Table 3.1: Error of a given cell measured on maximum error (k_i err), error on the optimal points (k_i err opt) and IGD_i.

Problem	Size of grid	k1 err	k2 err	k3 err	k1 err opt	k2 err opt	k3 err opt	IGD^1	IGD^2	IGD^3
CONV2 (Schütze, 2004)	100×100	10.624	0.4764		1.348	0.2424		0.8288	0.1683	
	200×200	5.3724	0.2391		0.6892	0.124		0.4075	0.0485	
	500×500	2.1635	0.0959		0.2745	0.0492		0.1081	0.0725	
	$1,000 \times 1,000$	1.0842	0.048		0.1366	0.0241				
CONV3 (Schütze, 2004)	$20 \times 20 \times 20$	105.4206	105.4206	148.9384	32.6077	14.6249	14.4099	2.1294	3.548	1.981
	$30 \times 30 \times 30$	76.124	76.124	107.5589	24.4209	8.3636	14.8069	2.0873	3.3505	2.0557
	$60 \times 60 \times 60$	41.1418	41.1418	58.1362	10.8723	4.873	6.4499	1.5386	1.5386	1.3699
	$100 \times 100 \times 100$	25.4513	25.4513	35.9657	6.5238	3.4837	3.4519			
RUDOLPH (Rudolph et al., 2007)	100×100	1.8627	1.8627		0.2843	0.2843		0.1264	0.1264	
	200×200	0.9413	0.9413		0.1345	0.1345		0.0494	0.0494	
	500×500	0.3789	0.3789		0.0566	0.0566		0.0216	0.0216	
	$1,000 \times 1,000$	0.1899	0.1899		0.028	0.028				
SSW (Schäffler et al., 2002)	$20 \times 20 \times 20$	6	0.0031		6	0.0031		2.4	0.0005	
	$30 \times 30 \times 30$	4	0.0021		4	0.002		1.4	0.0004	
	$60 \times 60 \times 60$	2	0.0011		2	0.001		0.4	0.0002	
	$100 \times 100 \times 100$	1.2	0.0007		1.2	0.0006				

A *transient cell* is by definition a cell that is not periodic. For absorbing Markov chain, the system will leave the transient cells with probability one and will settle on an absorbing (periodic) cell.

Canonical form (cf) Consider an arbitrary absorbing Markov chain. Renumber the states so that the transient states come first. If there are r absorbing states and ts transient states ($N_c = r + ts$), the transition matrix has the following canonical form:

$$P = \begin{pmatrix} I & 0 \\ R & Q \end{pmatrix},$$

where Q is a ts by ts matrix, R is a non-zero t by r matrix, 0 is an r by t zero matrix, and I is the r by r identity matrix. Matrix Q gathers the probabilities of transitioning from some transient state to another whereas matrix R describes the probability of transitioning from some transient state to some absorbing state.

Fundamental matrix (fm) For an absorbing Markov chain the matrix $I - Q$ has an inverse $N = (I-Q)^{-1}$. The (i, j)-entry n_{ij} of the matrix N is the expected number of times the chain is in state s_j, given that it starts in state s_i. The initial state is counted if $i = j$. The matrix

$$fm = I + \sum_{k=1}^{\infty} Q^k \tag{3.12}$$

is called the *fundamental matrix (fm)* of the Markov chain which satisfies the following equation

$$fm = N. \tag{3.13}$$

Absorbing probability This is defined as the probability of being absorbed in the absorbing state j when starting from transient state i, which is the (i, j)-entry of the matrix $B = NR$. In terms of cell mapping, the set of all $B_{i,j} \neq 0$ for $i \in [1, \ldots, ts]$ is called the basin of attraction of state j, and an absorbing cell within that basin is called attractor.

3.2.1 Dynamical system

In the following, we present the dynamical systems used for both single and multi objective optimization.

Single-objective optimization In this case, a given cell $s_i \in S$ will map to all neighbors $N_e(s_i)$ that have a lower function evaluation. Then, the probability to map to each neighbor is proportional to the improvement in terms of function the evaluation. If there is no neighbor with lower function evaluation, then it belongs to

a periodic group (candidate to be a local optimum). In the following, we present the dynamical system used.

$$bc_i = \{s_j | f(s_j) < f(s_i) \text{ for all } s_j \in N_e(s_i)\} \tag{3.14}$$

$$pg_i = \{s_j | f(s_j) = f(s_i) \text{ for all } s_j \in N_e(s_i)\} \tag{3.15}$$

$$p_{ij} = \begin{cases} (f(s_i) - f(s_j)) \cdot \left(\sum_{k=1}^{|bc_i|} f(s_i) - f(s_k) \right)^{-1} & \text{, if } s_j \in bc_i \\ |pg_i|^{-1} & \text{, if } bc_i = \emptyset \text{ and } s_j \in pg_i \\ 0 & \text{, otherwise} \end{cases} \tag{3.16}$$

Multi-objective optimization Here, we extend the dynamical system used in the single-optimization case. A cell s_i will map to those neighbors that dominate s_i.

$$bc_i = \{s_j | F(s_j) \preceq F(s_i) \text{ for all } s_j \in N_e(s_i)\} \tag{3.17}$$

$$pg_i = \{s_j | F(s_j) = F(s_i) \text{ for all } s_j \in N_e(s_i)\} \tag{3.18}$$

$$p_{ij} = \begin{cases} \|F(s_i) - F(s_j)\| \cdot \left(\sum_{k=1}^{|bc_i|} \|F(s_i) - F(s_k)\| \right)^{-1} & \text{, if } s_j \in bc_i \\ |pg_i|^{-1} & \text{, if } bc_i = \emptyset \text{ and } s_j \in pg_i \\ 0 & \text{, otherwise} \end{cases} \tag{3.19}$$

3.2.2 The algorithm

Algorithm 6 shows the key elements to compute the characteristics needed to determine the features described within this section. For each cell z, we compare $f(z)$ to the objective values of its neighbors $N_e(z)$. Next, we assign a probability, proportional to their function values, to pass into those cells. If there is no better neighbor cell, equal transition probabilities are assigned to the neighbor cells with equal function values. Worse neighbor cells always get transition probability 0.

Complexity of GCM

In the case of GCM, all cells are visited once to compute the probability matrix. For each cell the following operations are computed: find the center point, compute the descent direction (which can be done comparing with the $2n$ or $3^n - 1$ neighbors based on the type of neighborhood used) and compute the transition probabilities.

Afterwards, the computation of the canonical matrix is linear with respect to N_c and the computation of the fundamental matrix takes cubic time since it implies solving a linear system of equations. Thus, the complexity is given by $O(N_c^3)$. Note that if only the canonical matrix is needed, then the complexity is $O(N_c)$. Further, the number of cells depends on the number of variables n of the problem $N_c = \prod_{i=1}^{n} N(i)$. Thus, the complexity with respect to the number of variables n is $O(N_{max}^n)$.

3.2.3 Numerical results

Similar to the numerical results of SCM, we use GCM to compute approximations of the global Pareto set. Figure 3.4 shows the results obtained by GCM on four MOPs (see Appendix A for the description of the problems). We have used a 1,000 × 1,000 grid to perform the cell mapping for the problems with $n = 2$ and a 100 × 100 × 100 grid for $n = 3$. In all cases GCM is able to obtain an approximation of both Pareto set and front.

3.3 Hybrid Cell Mapping

Although the SCM method can find the global and fine structure of optimal solutions, its computational time increases dramatically as the dimension of the decision space goes up (Naranjani et al., 2013b). The computational time can be reduced if we apply the SCM method only in the vicinity of the solution instead of sweeping the entire decision space. This approach is particularly beneficial to MOPs of high dimensions, and is also the key feature for the set oriented numerics (Dellnitz and Hohmann, 1997; Dellnitz and Junge, 1998, 2002). 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 Hillermeier (2001) which has been coupled with set oriented methods leading to the *recovering algorithm* in Dellnitz et al. (2005b); Schütze et al. (2005). This method is by construction of local nature and needs multiple starting points in case the Pareto set/front is disconnected.

The main idea 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 entire decision space, because the Pareto set occupies only a small fraction of the decision 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.

3.3.1 The algorithm

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 7 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 $\mathcal{P}$.

NSGA-II has been used in this study. Because there are extensive studies and analyses available on this method (Deb et al., 2002; Deb, 2001c), it is not discussed further in this paper. The cell mapping method is implemented in the algorithm 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 8. 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 has non-empty intersection with the solution set and is added to the set $\mathcal{B}_l$ of candidate boxes. 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 Schütze et al. (2003), 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.

3.3.2 Convergence of the method

Here, we 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 Dellnitz and Hohmann

(1997); Dellnitz et al. (2005b) 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 9 shows the pseudo code of the abstract algorithm.

Hereby, we assume that the domain is contained inside

$$Q \subset \hat{Q} = [a_1, b_1] \times \dots [a_n, b_n] \subset \mathbb{R}^n, \tag{3.20}$$

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 Dellnitz and Hohmann (1997):

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

$$\begin{aligned} B = B(c, r) = \{x \in \mathbb{R} \ : \ & c_i - r_i \leq x_i \leq c_i + r_i, \\ & i = 1, \dots, n\}, \end{aligned} \tag{3.21}$$

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

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

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

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

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

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, $j_i = ((i-1) \, mod \, n) + 1$. 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 15. *Let an MOP of the form (2.1) be given and assume that there exists no weak Pareto point in $Q \backslash \mathcal{P}$. 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.,*

$$\mathcal{B} \cap \mathcal{P} \neq \emptyset. \tag{3.22}$$

Then, an application of Algorithm 9 leads to a sequence of cell collections $\mathcal{B}_l$ such that

$$d_H(\mathcal{B}_l, \mathcal{P}) \to 0 \quad \textit{for } l \to \infty. \tag{3.23}$$

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 (3.22) 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 $c : [0,1] \to \mathcal{P}$ 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

$$F(y) <_p F(x) \quad \forall y \in \Delta(\rho). \tag{3.24}$$

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.

3.3.3 Numerical results

Here, we apply the hybrid cell mapping to compute approximations of the global Pareto set. Figure 3.5 shows the results obtained by GCM on four MOPs (see Appendix A for the description of the problems). In all cases, a population of size 50 and 50 generations were used. The results show that HCM is able to obtain an approximation of both Pareto set and front comparable to the one of GCM although using less function evaluations.

Algorithm 5 Simple cell mapping for MOPs.

Require: MOP F, dynamical system DS, upper bound ub, lower bound lb, divisions per dimension N, cell size $h = (ub_i - lb_i)/N_i$ for $i = 1, \ldots, n$, Total number of cells $N_c = N_1 \times N_2 \times \ldots N_i$ for $i = 1, \ldots, n$

Ensure: Set of cells z, image of cells C, group number Gr, period number Pe, step number St, candidate Pareto set cPs

$current\_group \leftarrow 1$
$cPs = \{\}$
$Gr(i) \leftarrow 0, \forall i \in N_c$
for all $pcell \in N_c$ **do**
  $cell \leftarrow pcell$
  $i \leftarrow 0$
  while $newcell = true$ **do**
    $x_i \leftarrow$ center point of $cell$
    if $Gr(cell) = 0$ **then**
      $\nu \leftarrow$ compute as in Equation (3.1)
      $t \leftarrow$ compute as in Equation (3.3)
      $p_{i+1} \leftarrow x_i + \nu t$
      $ncell \leftarrow$ cell where p_{i+1} is located
      $C(cell) \leftarrow ncell$
      $cell \leftarrow ncell$
      $i \leftarrow i + 1$
    end if
    if $Gr(cell) > 0$ **then**
      $Gr(C^j(pcell)) \leftarrow Gr(cell), j \leftarrow 0, \cdots, i$
      $Pe(C^j(pcell)) \leftarrow Pe(cell), j \leftarrow 0, \cdots, i$
      $St(C^j(pcell)) \leftarrow St(cell) + i - j, j \leftarrow 0, \cdots, i$
      $cell \leftarrow C(cell)$
      $newcell \leftarrow false$
    end if
    if $Gr(cell) = -1$ **then**
      $current\_group \leftarrow current\_group + 1$
      $Gr(C^k(pcell)) \leftarrow current\_group, k \leftarrow 0, \cdots, i$
      $j \leftarrow i^{th}$ value when period appears
      $Pe(C^k(pcell)) \leftarrow i - j,\ k \leftarrow 0, \cdots, i$
      $St(C^k(pcell)) \leftarrow j - k,\ k \leftarrow 0, \cdots, j - 1$
      $St(C^k(pcell)) \leftarrow 0,\ k \leftarrow j, \cdots, i$
      $cPs \leftarrow cPs \cup cell_k,\ k \leftarrow j, \cdots, i$
      $cell \leftarrow C(cell)$
      $newcell \leftarrow false$
    end if
  end while
end for

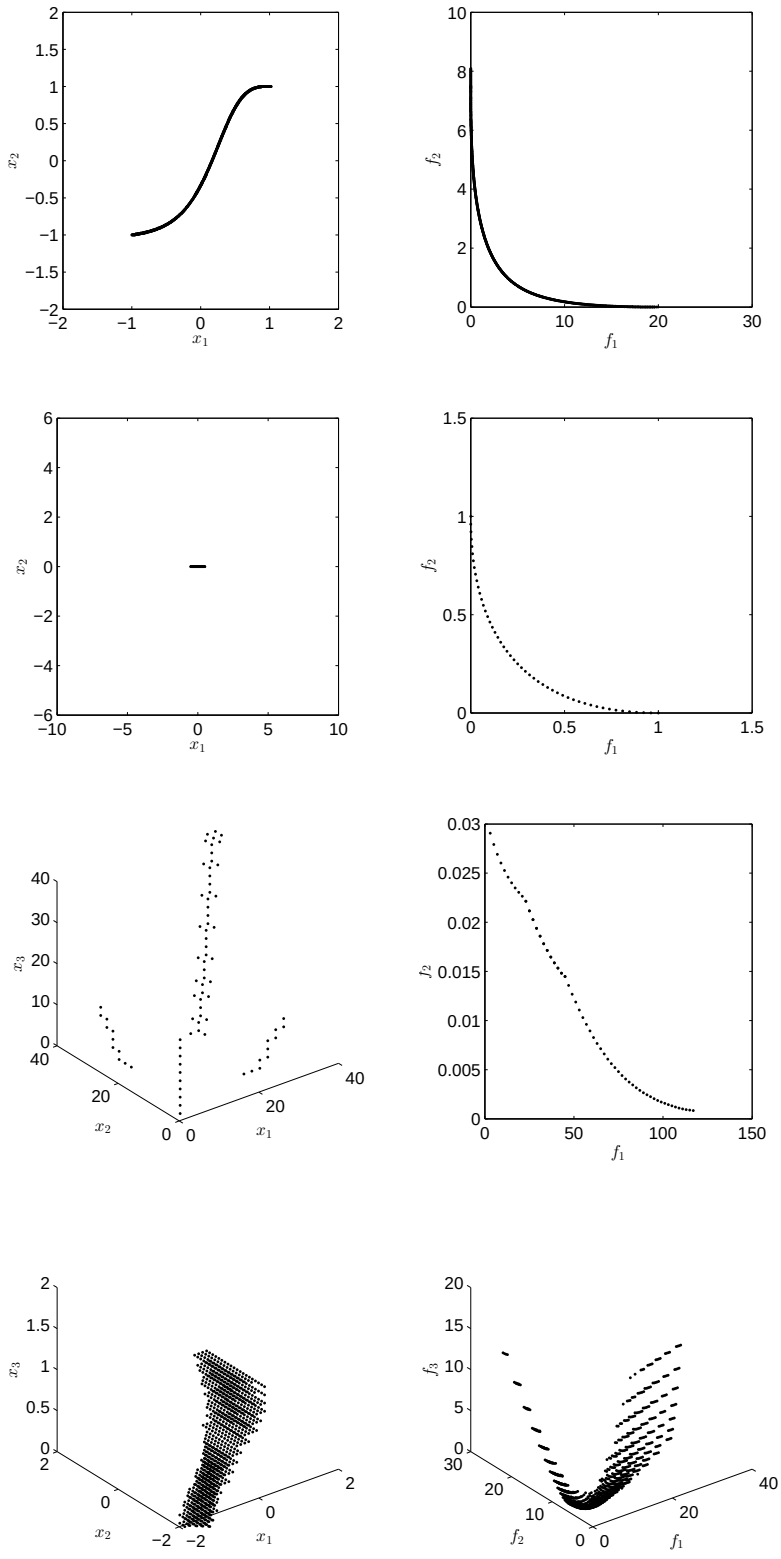


Figure 3.3: Pareto sets (left) and fronts (right) on the problems CONV2, RUDOLPH, SSW and CONV3.

Algorithm 6 Generalized Cell Mapping for Optimization

Require: f: Objective function, s: Set of cells
Ensure: cf, fm
Compute $f(s_i)$ for all $s_i \in N_e(s_i)$
Compute the set of better cells bc Equation 3.14 or 3.17
Compute the set of equal cells pg Equation 3.15 or 3.18
Compute the probability p Equation 3.16 or 3.19
Compute canonical form of p, $cf = \begin{bmatrix} I & 0 \\ R & Q \end{bmatrix}$
Compute fundamental matrix of cf, $fm = N = (I - Q)^{-1}$

Algorithm 7 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** $l = 1, \ldots, iter$ **do**
3: $\quad \tilde{\mathcal{B}}_l \leftarrow \text{Explore}(N, \mathcal{B}_{l-1}, F)$
4: $\quad N \leftarrow sub \times N$
5: $\quad B_l \leftarrow$ Mapping $\tilde{\mathcal{B}}_l$ to refined grid
6: **end for**

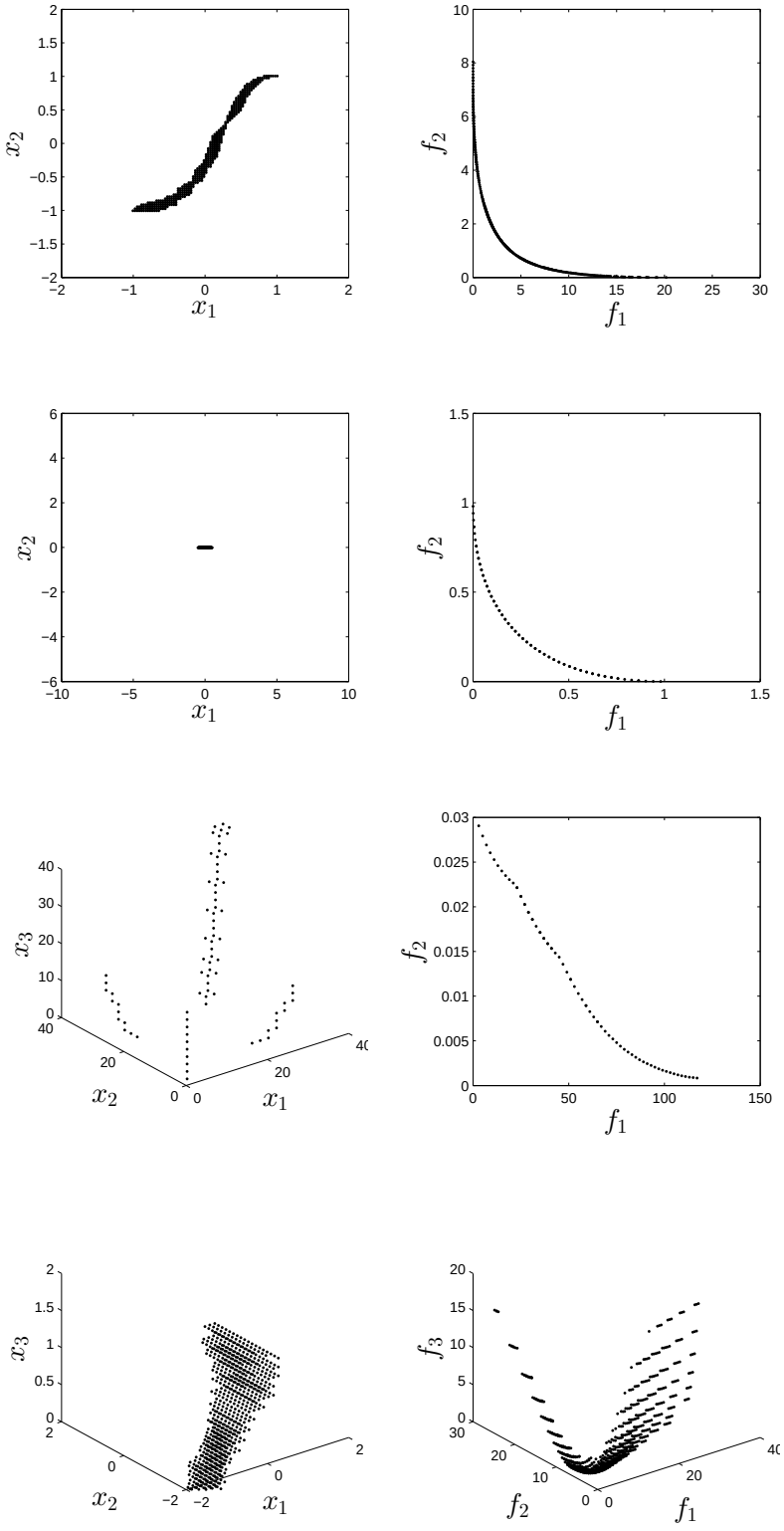


Figure 3.4: Pareto sets (left) and fronts (right) obtained on the problems CONV2, RUDOLPH, SSW and CONV3.

Algorithm 8 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 $\mathcal{B}_l$

$S \leftarrow \mathcal{B}_l$, $\mathcal{B}_l \leftarrow \emptyset$, $i \leftarrow 0$
while $i < |S|$ **do**
  $cs \leftarrow S_i$, $dest\_cell \leftarrow cs$, $dist \leftarrow 0$
  for all $s \in \mathcal{N}(cs)$ **do**
    if $center(s) \prec center(cs)$ & $\|F(center(s)) - F(center(cs))\|_2 > dist$ **then**
      $dest\_cell \leftarrow s$
      $dist \leftarrow \|F(center(s)) - F(center(cs))\|_2$
    end if
  end for
  if $dest\_cell \notin S$ **then**
    $S \leftarrow S \cup dest\_cell$
  end if
  if $dest\_cell = cs$ **then**
    if $dest\_cell \notin \mathcal{B}_l$ **then**
      $\mathcal{B}_l \leftarrow \mathcal{B}_l \cup dest\_cell$
    end if
    for all $s \in \mathcal{N}(cs)$ **do**
      if $center(cs) \not\prec center(s)$ & $\nexists b \in \mathcal{B}_l : b \prec s$ **then**
        $S \leftarrow S \cup s$
      end if
    end for
  end if
  $i \leftarrow i + 1$
end while
$\mathcal{B}_l \leftarrow dominance\_check(\mathcal{B}_l)$

Algorithm 9 Abstract Algorithm of GA-SCM

Require: Cell collection $\mathcal{B} \subset P(\hat{Q}, d_0)$
Ensure: Sequence $\mathcal{B}_l$ of cell collections
create a queue Q using $\mathcal{B}$
set $V := \mathcal{B}$ ▷ (Start Recover)
while Q is not empty **do**
  $T \leftarrow$ Q.dequeue(), where T is the current cell
  for all $C \in \Delta(T)$ s.t. $\nexists b \in V : b \preceq C$ **do**
    if $C \notin V$ **then**
      add C to V
      enqueue C onto Q
    end if
  end for
end while
$\mathcal{B}_0 := \{v \in V : \nexists \tilde{v} \in V, \tilde{v} \preceq v\} \subset P(\hat{Q}, d_0)$ ▷ (Start Subdivision)
for $l = 1, 2, \ldots$ **do**
  Subdivision: construct $\hat{\mathcal{B}}_l \subset P(\hat{Q}, d_0 + l)$ from B_{l-1} such that

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

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

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

end for

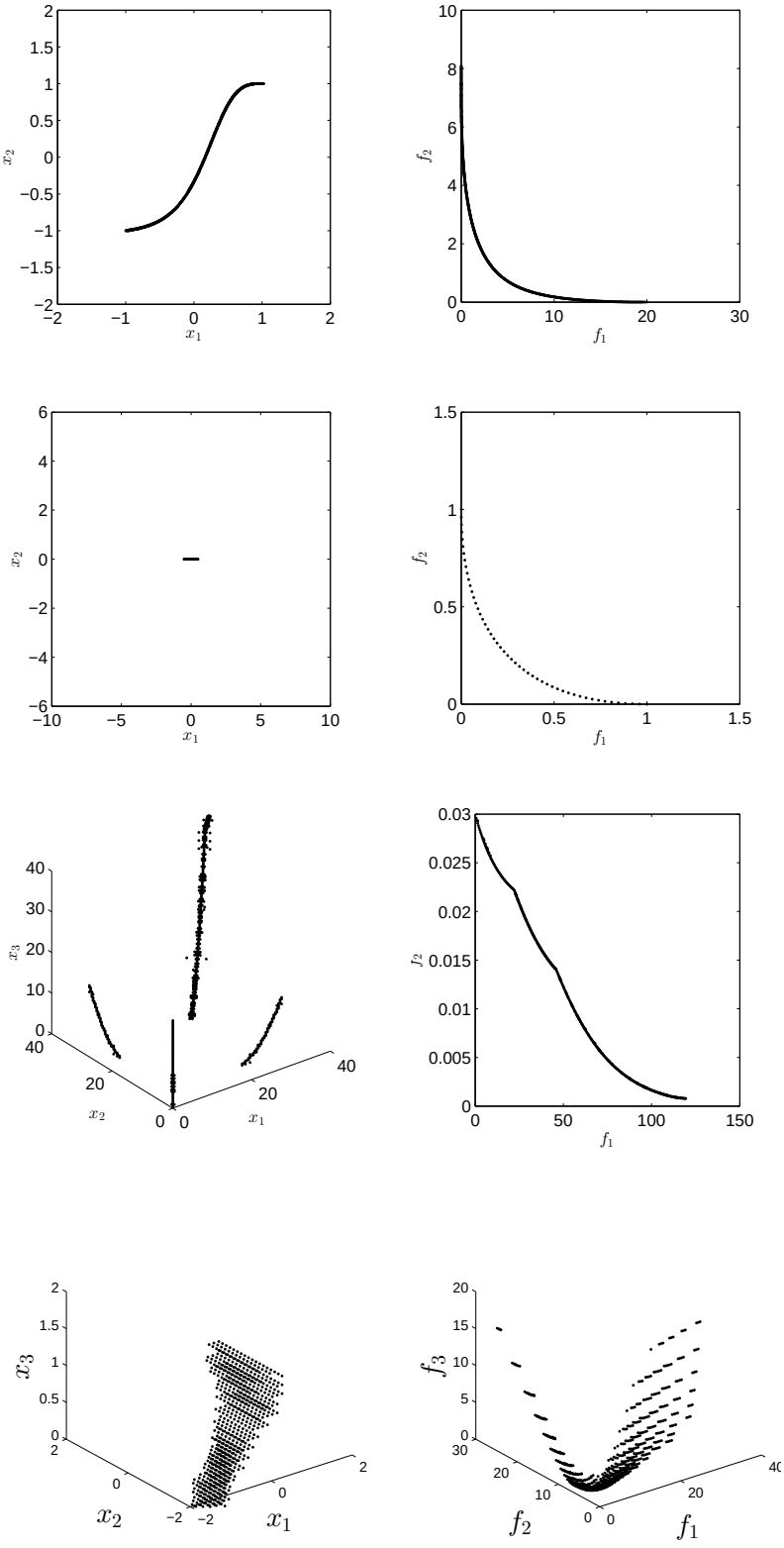


Figure 3.5: Pareto sets (left) and fronts (right) obtained CONV2, RUDOLPH, SSW and CONV3.

3.4 Comparison of the Cell Mapping Methods

In the following, we compare the results obtained by SCM, GCM and HCM in terms of $Delta_2$ in both decision and objective space. Table 3.2 shows the results for decision space; Table 3.3 shows the results for decision space. Further, we compare the number of function evaluations used by the methods in Table 3.4. From the results, it is possible to observe that GCM and HCM have similar performance on the test problems. This is expected since they use the same dynamical system. Also, in 3 out of the 4 problems SCM obtains a better performance.

Table 3.2: Δ_2 values in decision space.

Method	SCM	GCM	HCM
CONV2 $k = 2$	**0.013881**	0.023877	0.023877(7.1192e-18)
Rudolph	0.02434	**0.016094**	0.0283
SSW	**0.61219**	0.81151	1.0899(1.4596)
CONV3	**0.051548**	0.061572	0.061572(2.1357e-17)

Table 3.3: Δ_2 values in objective space.

Method	SCM	GCM	HCM
CONV2 $k = 2$	**0.013717**	0.038304	0.038304
Rudolph	0.022249	0.016115	**0.00054543(2.2247e-19)**
SSW	0.19784	**0.19629**	0.25255
CONV3	**0.33386**	0.47757	0.47757(5.6953e-17)

Table 3.4: Function evaluations used by the methods for each problem.

Method	SCM	GCM	HCM
CONV2	1e+06	1e+06	1.9063e+04(2.1255e+03)
Rudolph	1e+06	1e+06	8.2289e+03(620.0414)
SSW	1.25e+05	1.25e+05	4.6214e+04(7.8306e+03)
CONV3	1.25e+05	1.25e+05	6.0591e+04(5.1436e+03)

3.5 Selected Applications

In the following, we present several applications where the cell mapping techniques are well-suited to tackle them. These applications include from optimal control problems and landscape analysis.

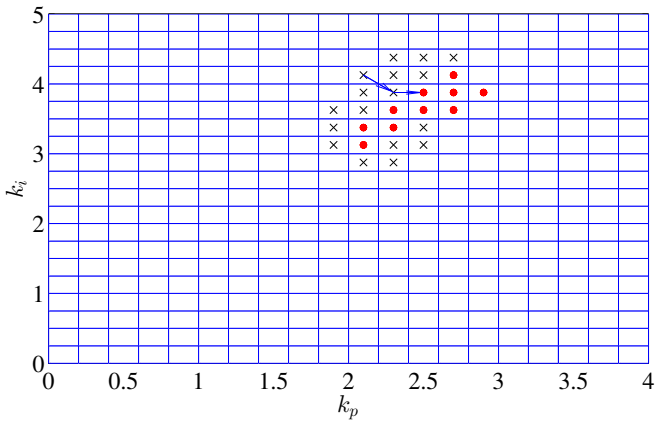


Figure 3.6: Demonstration of the cell evolution on a 20×20 grid of the FOPTD system. Black crosses are the transient cells while red dots are the invariant set cells represent coarse cell solutions for further refinement. Blue arrows represent a path from a transient cell evolutes to steady state.

3.5.1 First order system plus time delay

Here, we use the problem stated in Appendix B.1.3. For the SCM method, we select the number of initial partitions of the decision space Q as $\mathbf{N} = [50, 50]$. Subdivisions for finer structure of Pareto set are taken as 3 on each dimension. The total CPU time for computing all the solutions is 123.94 seconds on a laptop PC with an Intel core 2 duo and 2 GB of RAM. Figure 3.6 and 3.7 demonstrate the solving process of SCM on coarse and refined cellular space with a 20×20 grid. Figure 3.8 and 3.10 show the coarse Pareto set and front. 43 coarse cells are found as the covering set for further refinement. We take the subdivision as 3×3 on each searched cell at first stage. 130 cells are found in refined cellular space and the Pareto set and front are shown in Figure 3.9 and 3.11. The time domain response of a selected control is plotted in Figure 3.12 with the PI design parameters as $[k_p, k_i] = [2.4133, 3.5500]$ and corresponding objectives of this control design $t_p = 1.1500s$, $M_p = 1.0987\%$, $e_{IAE} = 0.2986$, $\lambda = -1.9163$. Good tracking performance can be reached.

3.5.2 Second order linear oscillator

Now, we use the problem stated in Appendix B.1.5. Initially, we select the number of divisions in the three control gain intervals $\mathbf{N} = [20, 20, 30]$. 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 $3 \times 3 \times 3$ subdivisions. The total CPU time for this example is 3893.4 seconds.

Figure 3.13 and 3.14 show the Pareto set of coarse and refined cellular space. 376 and 886 cells are found at each stage respectively. Figure 3.15 and 3.16 are the Pareto front of coarse and refined cellular space. The fine structure of Pareto front

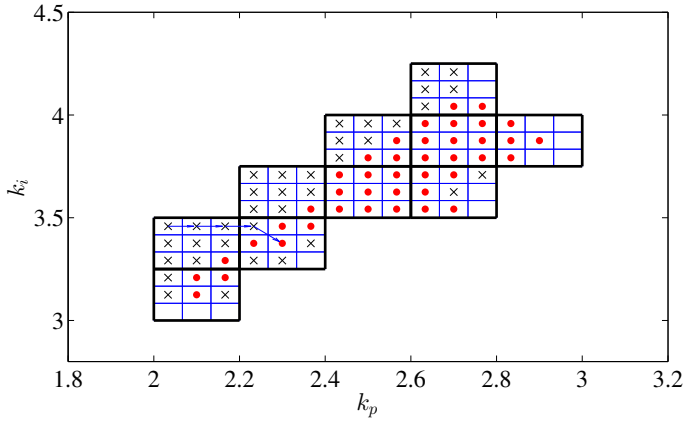


Figure 3.7: Demonstration of the cell evolution on a 60×60 grid of the FOPTD system at the refined cellular space. Note this process is based upon the acquired coarse cell set shown in Figure 3.6. The 3×3 subdivsion is taken on each coarse cell and SCM is constructed on smaller cells to acquire finer solutions. Black crosses are the transient cells while red dots are the invariant set cells represent coarse cell solutions for further refinement. Blue arrows represent a path from a a transient cell evolutes to steady state.

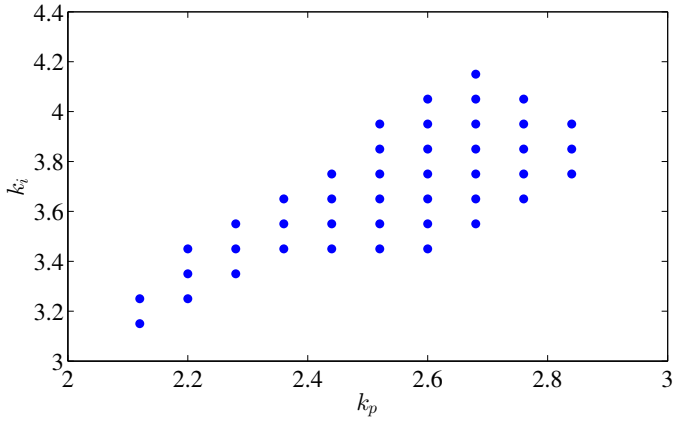


Figure 3.8: Coarse Pareto set of the FOPTD system with cell space partition 50×50. 43 cells are found as solution cells.

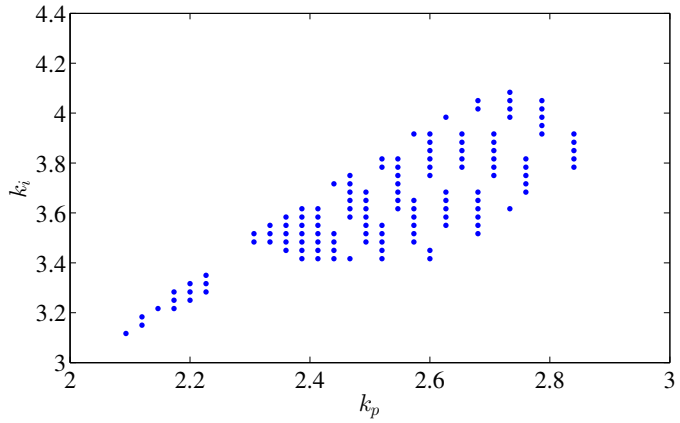


Figure 3.9: Refined Pareto set of the FOPTD system with initial cell space partition 50×50 and 3×3 as subdivision. 130 cells are found as solution cells.

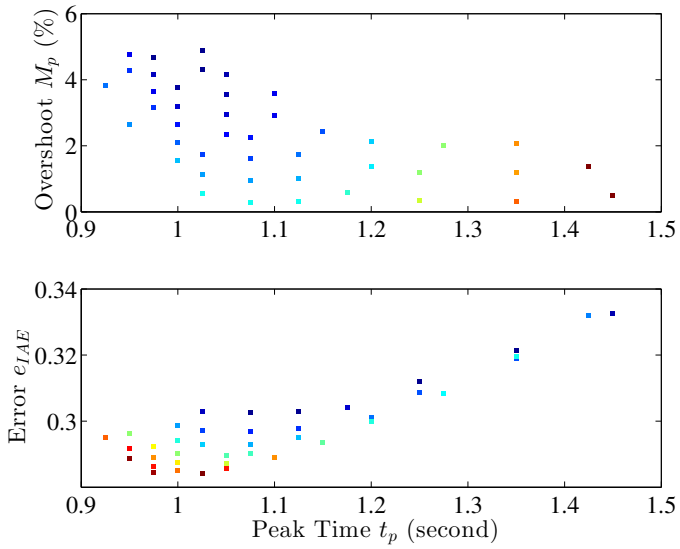


Figure 3.10: Coarse Pareto front of FOPTD system. Color code indicates the level of the other objective not shown in each subplot. Red is the highest level while blue is the lowest.

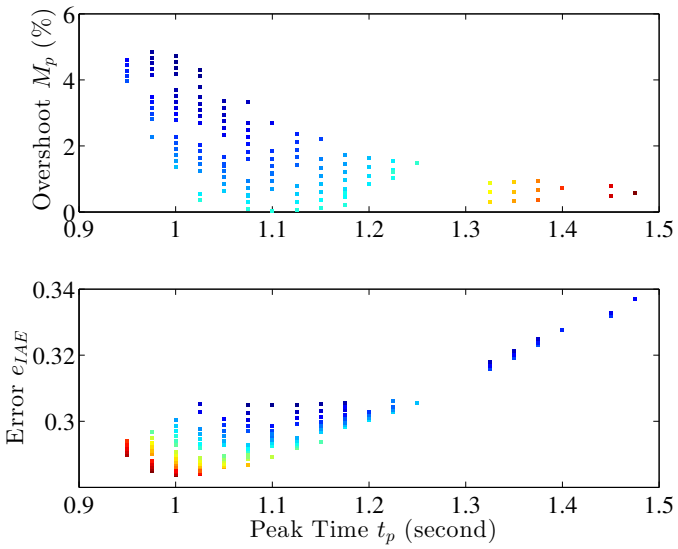


Figure 3.11: Refined Pareto front of FOPTD system. Color code indicates the level of the other objective not shown in each subplot. Red is the highest level while blue is the lowest.

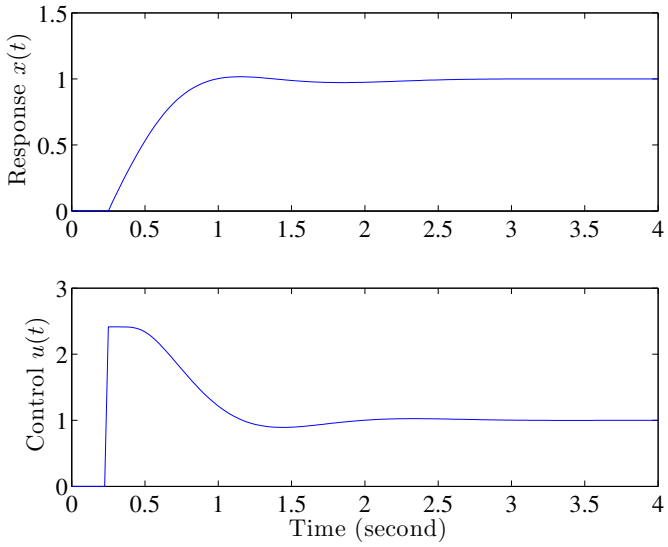


Figure 3.12: Time domain control effect of a selected control design from Pareto set. The selected PI design parameters are $[2.4133, 3.5500]$. Corresponding objectives of this control design are $t_p = 1.1500s$, $M_p = 1.0987\%$, $e_{IAE} = 0.2986$, $\lambda = -1.9163$. Good tracking performance can be reached.

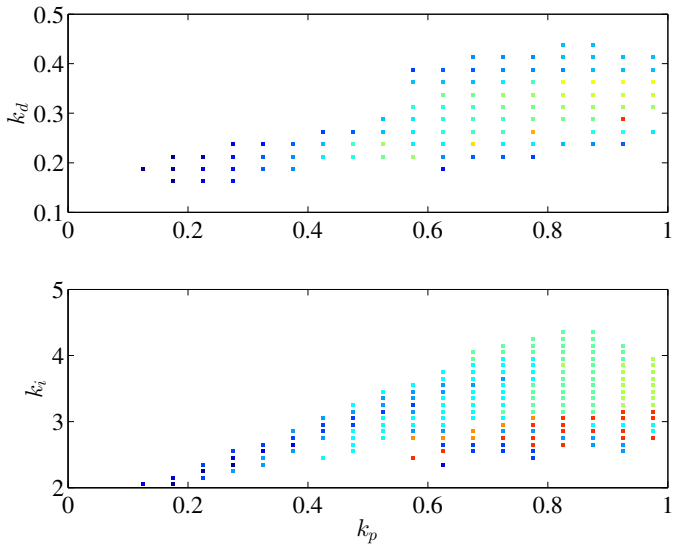


Figure 3.13: Coarse Pareto set of the second order LTI system. 376 cells are found as covering set. Color code indicates the other parameter level not shown in each subplot. Red indicates the highest level while blue the lowest.

can be clearly observed from 3.16 with the conflicting nature among design objectives. Temporal response of a selected control design is shown in Figure 3.17 with the PID design parameter as $[k_p, k_i, k_d] = [0.6750, 0.2708, 3.8500]$ and corresponding objectives $t_p = 0.7400s$, $M_p = 3.1689\%$, $e_{IAE} = 0.2917$, $\lambda = -2.3015$.

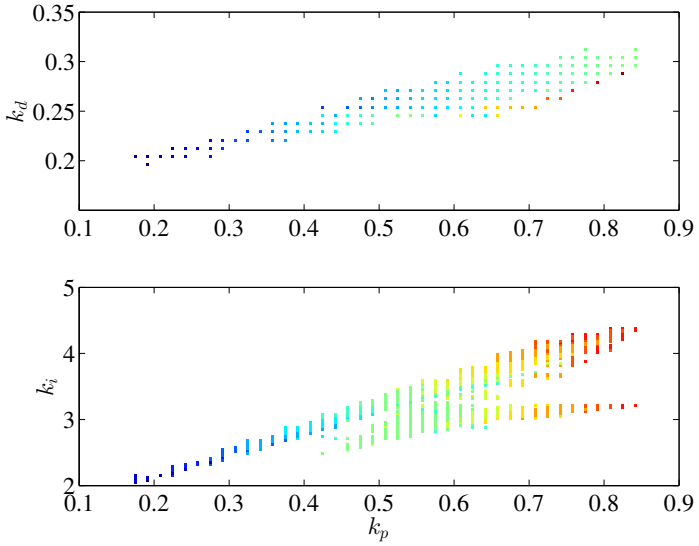


Figure 3.14: Refined Pareto set of the second order LTI system. 886 cells are found as covering set. Color code indicates the other parameter level not shown in each subplot. Red indicates the highest level while blue the lowest.

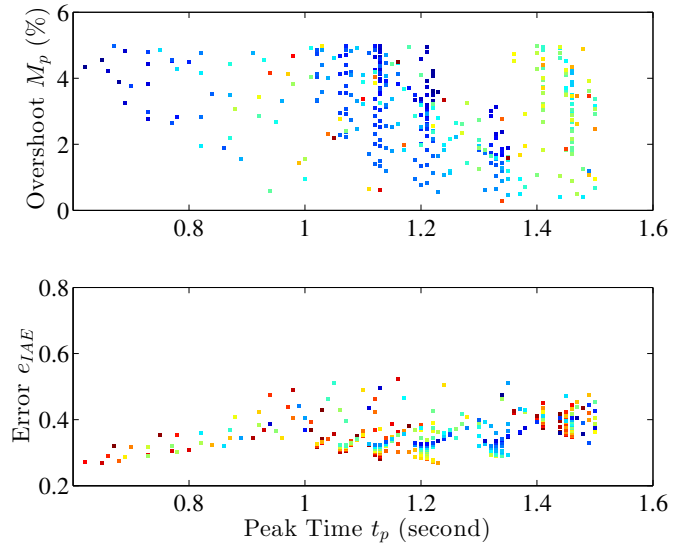


Figure 3.15: Coarse Pareto front of the second order LTI system.

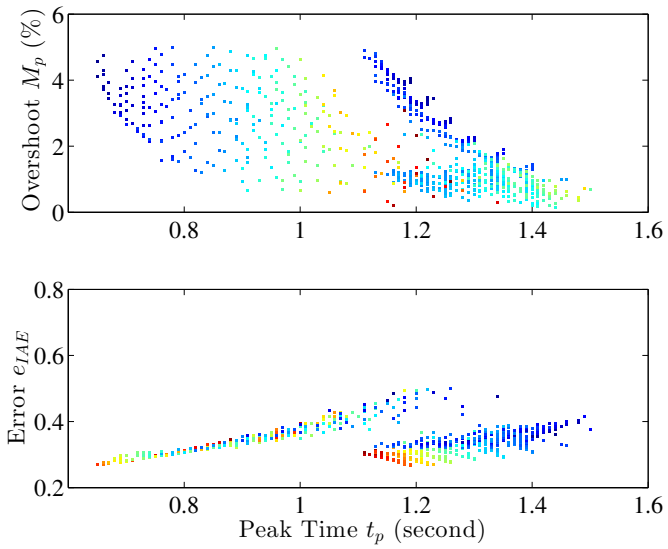


Figure 3.16: Refined Pareto front of the second order LTI system. Fine structure can be observed from the refined solutions, which indicates the intrinsic conflicting nature among the objectives.

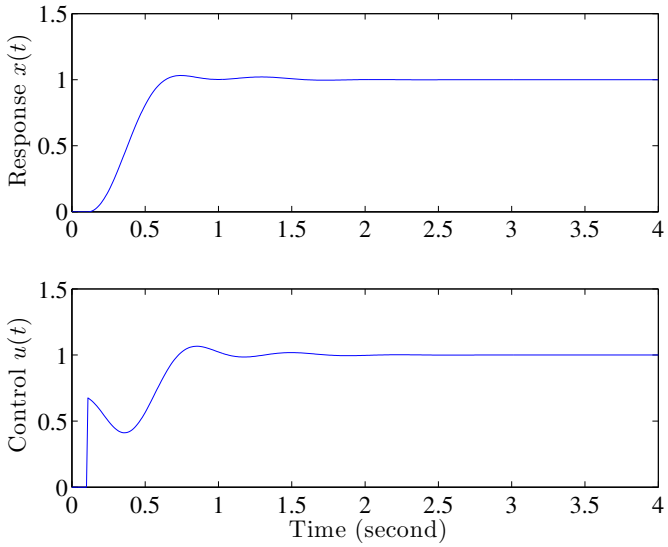


Figure 3.17: A selected control design from Pareto set as $[0.6750, 0.2708, 3.8500]$ with objectives as $t_p = 0.7400s$, $M_p = 3.1689\%$, $e_{IAE} = 0.2917$, $\lambda = -2.3015$.

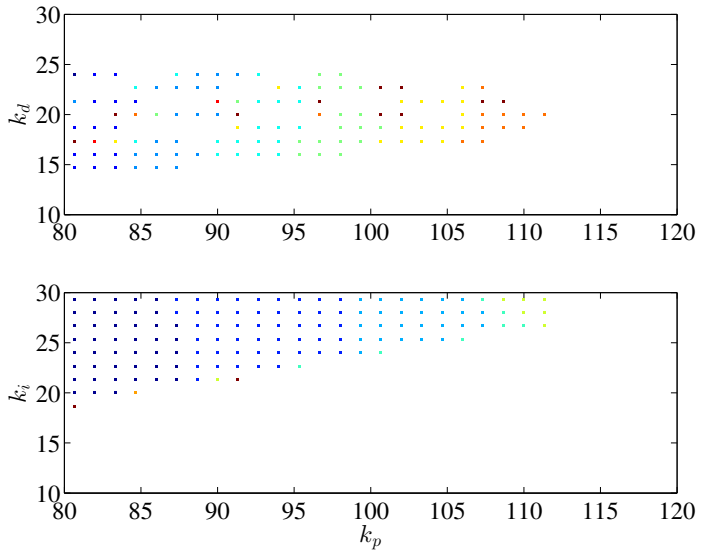


Figure 3.18: The Pareto set obtained on the rough grid by the SCM method for the Duffing system with delayed control.

3.5.3 Non-linear Duffing system

Now, we use the problem stated in Appendix B.1.6. Initially, we select the number of divisions in the three control gain intervals as $\mathbf{N} = [30, 15, 15]$. The cells of the rough Pareto set is sub-divided into 27 cells ($3 \times 3 \times 3$). The first run of the SCM method on the rough grid finds 460 cells representing the Pareto set shown in Figure 3.18. The corresponding Pareto front is shown in Figure 3.19. The CPU time of the first run is $1,382.4$ seconds. The second run on the sub-divided cells finds the Pareto set with 2386 cells shown in Figure 3.20. The refined Pareto front is shown in Figure 3.21. The CPU time of the second run is $5,695.5$ seconds.

We should point out that the Pareto fronts obtained by the SCM method have fine global structures. Such fine structures of Pareto fronts are not often found in the literature before. Finally, we present an example of step response under the delayed control with the gain $[k_p, k_i, k_d] = [82.4444, 21.7778, 14.2222]$. The result is shown in Figure 3.22. The step response shows excellent time-domain performance with $[t_p, M_p, e_{IAE}, \lambda] = [0.3300, 4.8829\%, 0.2155, -0.2781]$.

3.5.4 An inverted pendulum

Now, we use the problem stated in Appendix B.2.1. The SCM hybrid algorithm is carried out in the gain space with lower and upper bounds set as $[-1, 0, -1, 0]^T$ and $[0, 5, 0, 1]^T$. Hence,

$$Q = \{\mathbf{k} \in \mathbf{R}^4 \mid [-1, 0, -1, 0] \leq_p \mathbf{k} \leq_p [0, 5, 0, 1]\}. \tag{3.25}$$

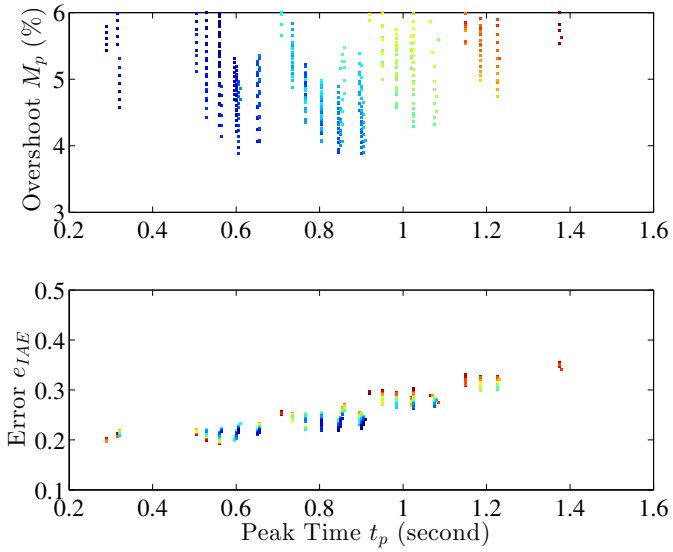


Figure 3.19: The Pareto front of the Duffing system corresponding to the Pareto set in Figure 3.18.

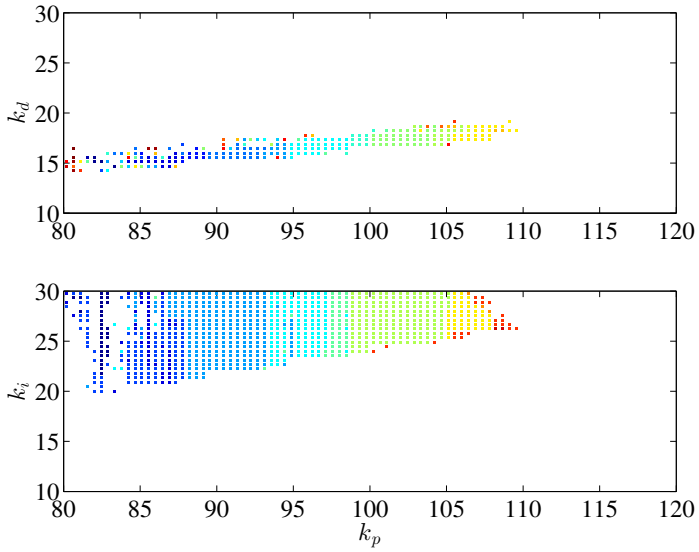


Figure 3.20: The refined Pareto set shown in Figure 3.18 of the Duffing system with delayed control by the SCM method.

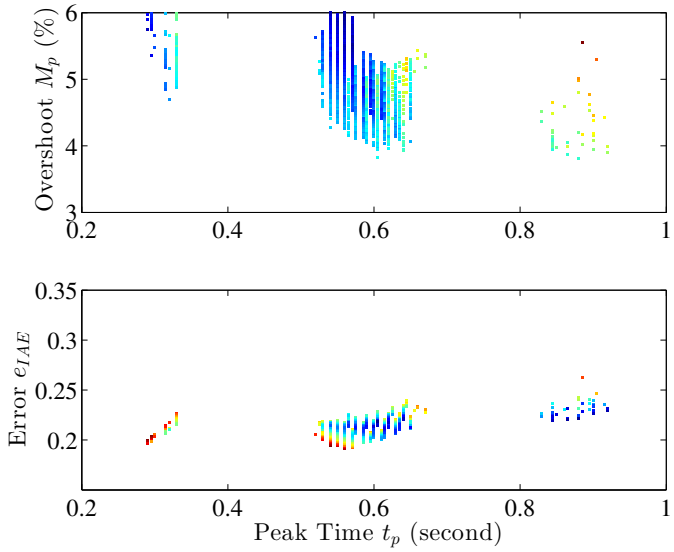


Figure 3.21: The refined Pareto front of the Duffing system corresponding to the Pareto set in Figure 3.20.

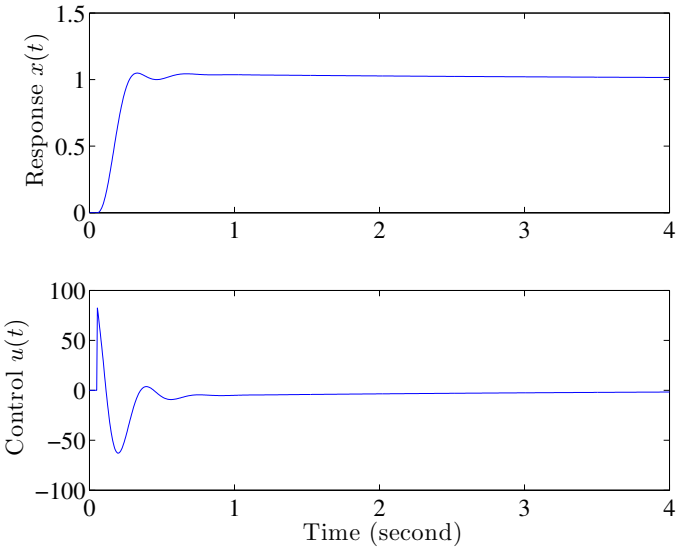


Figure 3.22: An example of the step response of the Duffing system under the delayed PID control with $[k_p, k_i, k_d] = [82.4444, 21.7778, 14.2222]$.

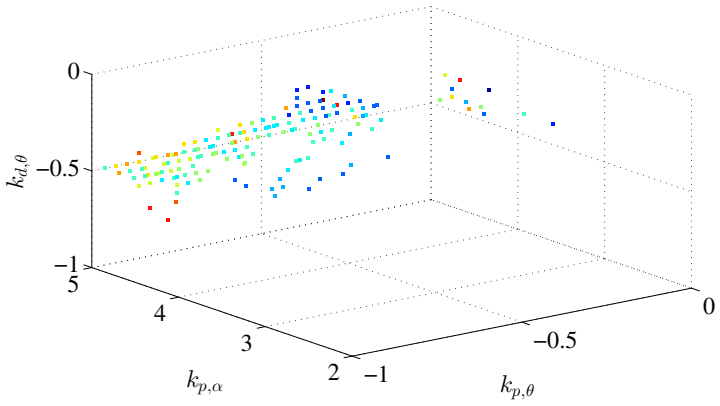


Figure 3.23: The 3D projection of Pareto set for the full state feedback control gains for the inverted pendulum. Color code indicates the level of $k_{d,\alpha}$. Bright color indicates the highest level while dark color indicates the lowest level.

The cell space partition for the coarse computation is by $10 \times 10 \times 8 \times 8$ and the refinement of the cell is by $3 \times 3 \times 3 \times 3$. The CPU time of the computation is 341 seconds in the first run, and 343 seconds in the refinement. The number of cells in the Pareto set in the first run is 41, and the number of cells in the refinement is 134.

Figures 3.23 and 3.24 show the Pareto set and Pareto front of this example. A fine structure of the Pareto front can be observed from Figure 3.24, which presents various compromises of the objectives. Figure 3.25 shows the time-domain responses of the system under four control gains of extremal cases (solid lines) when each objective function takes its minimum on the Pareto front as well as the responses with all other gains in the Pareto set. All the simulations start from an initial condition $\mathbf{x}(0) = [0.3, 0, 0, 0]^T$. As an example, the control gains corresponding to the smallest $\lambda_{\max}$ among all the Pareto optimal solutions are $\mathbf{k} = [-0.8176, 3.0833, -0.3125, 0.4375]$. The values of this control are $[t_{p,\alpha}, \max|\alpha|, t_s, \lambda_{\max}] = [0.1079, 5.5675, 1.1112, -5.6207]$. The four responses shown in the figure with the extremal gains in the Pareto set approximately define the range such that the time-domain responses of the system under all Pareto optimal controls are within or slightly outside the range.

3.5.5 A flexible rotary arm

Now, we consider the full state feedback tracking control of a flexible rotary arm (see Appendix B.2.1). The decision space is initially divided into $10 \times 10 \times 8 \times 8 \times 5$ cells. Each cell is sub-divided into $3 \times 3 \times 3 \times 3 \times 3$ smaller cells in the refinement step. The CPU time of the computation is $1,256$ seconds for the first run, and 38768 seconds for the refinement. The number of cells in the Pareto set in the first run is 704, and the number of cells in the refinement is $2,140$.

The Pareto set and Pareto front of the system are shown in Applications 3.26 and 3.27. Applications 3.28 and 3.29 show the time-domain simulations of the system

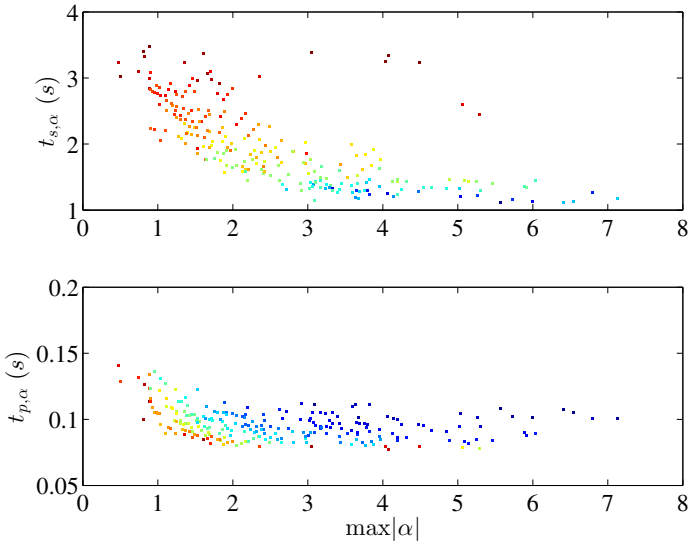


Figure 3.24: The Pareto fronts of the full state feedback control of the inverted pendulum. Color code of the upper figure indicates the level of $\lambda_{\max}$. For the lower figure, the color code indicates the level of t_s. In both sub-figures, red indicates the highest level while blue indicates the lowest level. Conflicting nature of the objectives can be clearly seen.

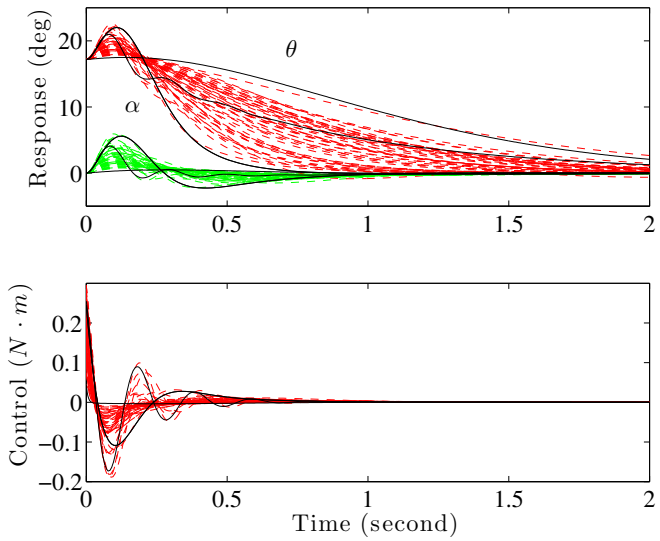


Figure 3.25: Numerical simulations of time domain responses of θ and α of the inverted pendulum and the control signal subject to an initial condition $[0.3, 0, 0, 0]^T$. Solid black lines: With gains of four extremal cases from the Pareto set when each objective function is minimal. Dashed color lines: With all other gains in the Pareto set.

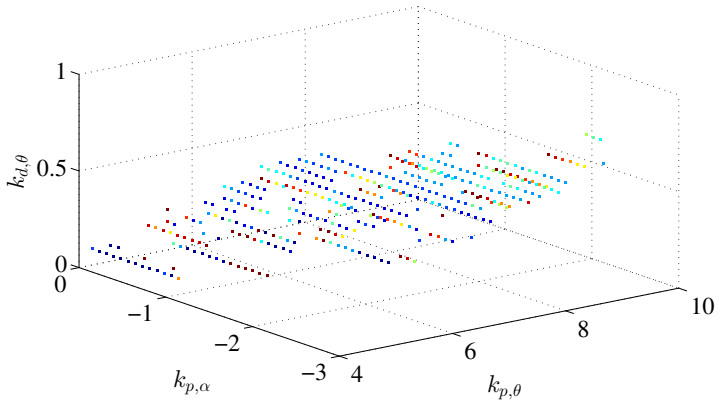


Figure 3.26: The Pareto set of full state feedback control gains of the flexible rotary arm. The $5D$ Pareto set is projected to a $3D$ sub-space $(k_{p,\theta}, k_{p,\alpha}, k_{d,\theta})$ with the color code indicating the level of $k_{d,\alpha}$. Red in the color code indicates the highest level while dark blue indicates the lowest level.

under six control gains of extremal cases (solid lines) when each objective function takes its minimum on the Pareto front as well as the responses with every 15 of all other gains in the Pareto set. When the settling time of θ is minimal, the control gains are $\mathbf{k}_e = [7.5000, -1.4583, 0.4062, 0.1562, 0.1833]$ and the corresponding objective function values are

$$[t_{s,\theta}, M_{p,\theta}, e_{IAE}, \max|\alpha|, t_{s,\alpha}, \lambda_{\max}] = [0.3s, 1.484\%, 0.0546, 4.5569^\circ, 0.738s, -0.0245].$$

As is the case in the previous example, the six responses shown in the figure with the extremal gains in the Pareto set approximately define the range such that the time-domain responses of the system under all Pareto optimal controls are within or slightly outside the range.

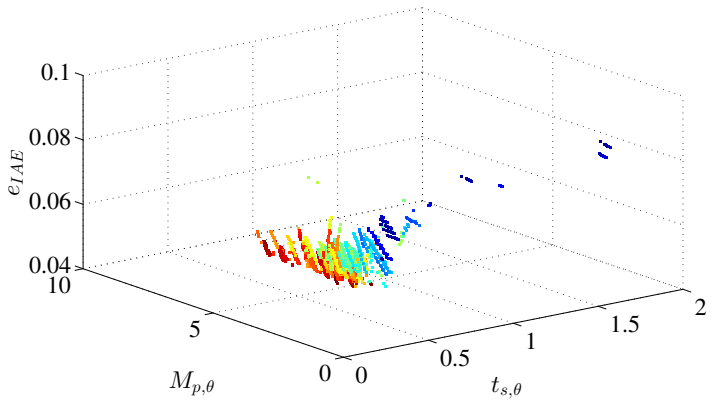


Figure 3.27: The Pareto front of full state feedback control of the flexible rotary arm. The Pareto front is projected to a $3D$ sub-space $(t_{s,\theta}, M_{p,\theta}, e_{IAE})$ with the color code indicating the level of $\max|\alpha|$. Red in the color code indicates the highest level while dark blue indicates the lowest level.

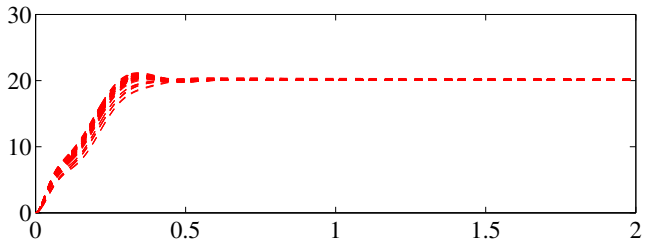


Figure 3.28: Numerical simulations of α and θ responses under Pareto optimal controls. Solid black lines: With gains of six extremal cases from the Pareto set when each objective function is minimal. Dashed color lines: With every 15 of all other gains in the Pareto set.

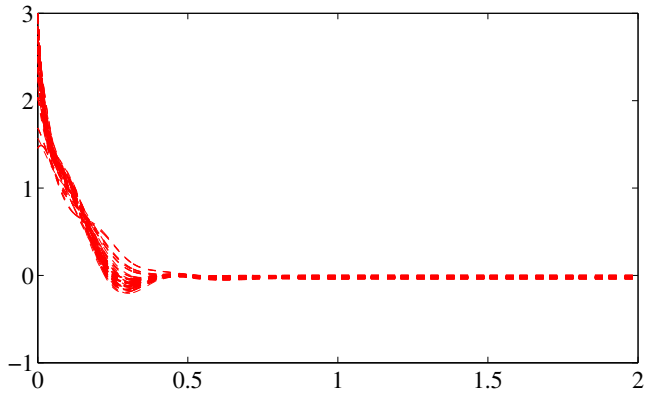


Figure 3.29: Pareto optimal controls of the flexible rotary arm. Solid black lines: With gains of six extremal cases from the Pareto set when each objective function is minimal. Dashed color lines: Every 15 of all other gains in the Pareto set.

Chapter 4

Computing the Set of Nearly Optimal Solutions

To tap the full potential of the additional consideration of approximate solutions against the 'sole' consideration of the optimal solutions it is required to maintain a representation of the *entire* set of approximate solutions $P_{Q,\epsilon}$. The challenge in this case is that for the consideration of a problem with n parameters and k objectives the set $P_{Q,\epsilon}$ forms an n-dimensional object while P_Q is under some mild regularity assumptions on F 'only' $(k-1)$-dimensional. Hence, for the effective extension from P_Q to $P_{Q,\epsilon}$ as the basis for the decision making process a suitable discretization of the latter set is essential.

In this chapter we address the problem of computing $P_{Q,\epsilon}$ via stochastic search algorithms. To be more precise, we design and investigate an archiving strategy that aims for the approximation of this set with a good distribution in both decision and objective space. Next, we make a comparative study on some test problems in order to visualize the effect of all strategies. Finally, we couple the proposed archiver with an evolutionary algorithm and compare with algorithms of the state of the art.

The remainder of this chapter is organized as follows: in Section 4.1 we present an archiver that aims for a good discretization according to decision and objective space of a given MOP. Then, in Section 4.2 we present a MOEA for the computation of the set of approximate solutions.

4.1 Discretizing the Set both in Decision and Objective Space

Here, we present an archiving strategy that aims for a finite representation of $P_{Q,\epsilon}$ in decision and objective space of an MOP. Further, we analyze its limit behavior as well as its bounds. We finish the section with numerical results on representative academic examples.

4.1.1 The algorithm

Algorithm 10 shows the pseudo code of $ArchiveUpdateP_{Q,\epsilon}D_{xy}$. The discretization allows to keep solutions that have similar objective values (measured by Δ_y) but are significantly different in decision space (measured by Δ_x).

Algorithm 10 $A := ArchiveUpdateP_{Q,\epsilon}D_{xy}\ (P, A_0, \epsilon, \Delta_x, \Delta_y)$

Require: population P, archive A_0, $\epsilon \in \mathbb{R}^n_+$, $\Delta_x \in \mathbb{R}_+$, $\Delta_y \in \mathbb{R}_+$, $\Delta^*_x \in (0, \Delta_x)$, $\Delta^*_y \in (0, \Delta_y)$

Ensure: updated archive A

```
A := A_0
for all p ∈ P do
    if ∄a_1 ∈ A : a_1 ≺_{-ε} p and ∄a_2 ∈ A : (d_∞(F(a_2), F(p)) ≤ Δ*_y and d_∞(a_2, p) ≤ Δ*_x) then
        A ← A ∪ {p}
        Â = {a_1 ∈ A | ∄a_2 ∈ A : a_2 ≺_{-(ε+Δ_y)} a_1}
        for all a ∈ A\Â do
            if p ≺_{-(ε+Δ_y)} a and dist(a, Â) ≥ 2Δ_x then
                A ← A\{a}
            end if
        end for
    end if
end for
```

Complexity of $ArchiveUpdateP_{Q,\epsilon}D_{xy}$

The algorithm goes through all the set of candidate solutions $p \in P$ and in the worst case, the algorithm compares them with all the solutions $a \in A$. Note that the $-\epsilon$-dominance operation takes constant time with respect of the sizes of P and A.Thus, the complexity of the archiver is $O(|P||A|)$ which in the worst case is quadratic $(O(|P|^2))$.

4.1.2 Limit behavior

The limit behavior of $ArchiveUpdateP_{Q,\epsilon}D_{xy}$ is investigated next.

Theorem 16. *Let (2.1) be given, where F is continuous and Q is compact, $\epsilon \in \mathbb{R}^k_+$, $\Delta_x, \Delta^*_x, \Delta_y, \Delta^*_y \in \mathbb{R}_+$ with $\Delta^*_y < \Delta_y$ and $\Delta^*_x < \Delta_x$. For the generation process we assume (2.46) and for the MOP the assumptions made in Lemma 8. Then, an application of Algorithm 1, where $AchiveUpdateP_{Q,\epsilon}D_{xy}$ is used to update the archive, leads to a sequence of archives $A_l, l \in \mathbb{N}$, such that there exists an $l_0 \in \mathbb{N}$ such that for all $l \geq l_0$:*

(i) $dist(P_{Q,\epsilon}, A_l) < \Delta_x$ *and* $dist(F(P_{Q,\epsilon}), F(A_l)) < \Delta_y$

(ii) $dist(A_l, P_{Q,\epsilon}) \leq dist(B_{2\Delta_x}(P_{Q,\epsilon+2\Delta_y}), P_{Q,\epsilon})$ *and*
$dist(F(A_l), F(P_{Q,\epsilon})) \leq dist(F(B_{2\Delta_x}(P_{Q,\epsilon+2\Delta_y})), F(P_{Q,\epsilon}))$

(iii) $d_H(P_{Q,\epsilon}, A_l) \leq \max(\Delta_x, dist(B_{2\Delta_x}(P_{Q,\epsilon+2\Delta_y}), P_{Q,\epsilon})$ *and*
$d_H(F(P_{Q,\epsilon}), F(A_l)) \leq \max(\Delta_y, dist(F(B_{2\Delta_x}(P_{Q,\epsilon+2\Delta_y})), F(P_{Q,\epsilon}))$

Proof. As the results in objective space are analog to the results in Theorem 10, we only prove here the results in decision space.

Note that a point p is discarded from an existing archive A in two cases (line 2):

$$\begin{array}{ll} (D1) & \exists a_1 \in A : \ a_1 \prec_{-\epsilon} p, \quad \text{or} \\ (D2) & \exists a_2 \in A : \ \|F(a_2) - F(p)\|_\infty \leq \Delta_y^* \text{ and } \|a_2 - p\|_\infty \leq \Delta_x^*. \end{array} \tag{4.1}$$

(i) since $\overline{P_{Q,\epsilon}}$ is compact, $dist(P_{Q,\epsilon}, A_l) = dist(\overline{P_{Q,\epsilon}}, A_l)$ and $A_l, l \in \mathbb{N}$, is finite it follows that

$$dist(P_{Q,\epsilon}, A_l) = \max_{p \in \overline{P_{Q,\epsilon}}} \min_{a \in A_l} \|p - a\|_\infty. \tag{4.2}$$

The claim is right for an archive $A_l, l \in \mathbb{N}$, if for every $p \in P_{Q,\epsilon}$ there exists an element $a \in A_l$ such that $\|p - a\|_\infty < \Delta_x$. Thus, we need to prove that $P_{Q,\epsilon}$ is contained in

$$C_{A,\Delta_x} := \bigcup_{a \in A} B^\infty_{\Delta_x}(a), \tag{4.3}$$

where $B^\infty_\delta(x) := \{y \in \mathbb{R}^n \ : \ \|x - y\|_\infty < \delta\}$.

First we show that if there exists an $l_0 \in \mathbb{N}$ with $dist(P_{Q,\epsilon}, A_l) < \Delta_x$ then this inequality holds for all $l \geq l_0$. Assume that such an l_0 is given. Let

$$\tilde{A} := \{a \in A_{l_0} | \exists p_1 \in P_{Q,\epsilon} : \|p_1 - a\|_\infty < \Delta_x \text{ or } \exists p_2 \in P_{Q,\epsilon} : \|F(p_2) - F(a)\|_\infty < \Delta_y\}. \tag{4.4}$$

The following discussion shows that no element $a \in \tilde{A}$ will be discarded further on due to the construction of the archiver. It holds

$$p \in P_{Q,\epsilon} \text{ and } a \in Q : \|p - a\| \leq \Delta_x \ \Rightarrow a \in B_{\Delta_x}(P_{Q,\epsilon}), \tag{4.5}$$

Assume an element $a \in \tilde{A}$ is discarded. For this to happen the following has to hold: (i) $a \notin P_{Q,\epsilon+\Delta_y}$, (ii) an element $a_2 \in A_l$ is added to the archiver such that $\nexists a_3 \in \tilde{A} : dist(F(a_2), F(a_3)) < \Delta_y$ and $a_2 \prec_{-(\epsilon+\Delta_y)} a$, and (iii) $\nexists b \in \tilde{A} : dist(a, b) \leq 2\Delta_x$, and hence, $dist(a, \hat{A}) > 2\Delta_x$.

For all $p \in P_{Q,\epsilon}$ there exists an element $a \in \tilde{A}$ such that $\|p - a\|_\infty < \Delta_x$.

Thus, there exists an element b such that $b \in \tilde{A} : dist(a, b) \leq 2\Delta_x$ and $b \in \hat{A}$. Otherwise, there would exist elements in $P_{Q,\epsilon}$ such that their distance to $\tilde{A}$ is bigger than Δ_x which is a contradiction to the assumption. Further, since by assumption $dist(P_{Q,\epsilon}, A_{l_0}) < \Delta_x$, there exists for all $p \in P_{Q,\epsilon}$ an element $a \in \tilde{A}$ with $\|p - a\|_\infty < \Delta_x$, and by the above consideration this holds for all archives A_l with $l \geq l_0$.

It remains to show the existence of such an integer l_0, which we will do by contradiction: first we show that by using $ArchiveUpdateP_{Q,\epsilon}D_{xy}$ and under the assumptions made above only finitely many replacements can be done during the run of the algorithm. Then, we construct a contradiction by showing that under the assumptions made above infinitely many replacements have to be done during the run of the algorithm with the given setting.

Let a finite archive A_0 be given. If a point $p \in \mathbb{R}^n$ replaces a point $a \in A_0$ (lines 5 to 7) it follows by construction of $ArchiveUpdateP_{Q,\epsilon}D_{xy}$ that

$$F(p) <_p F(a) - \Delta_y \text{ and } \|p - a\| \geq \Delta_y. \tag{4.6}$$

Since the relation '$\prec$' is transitive, there exists for every $a \in A$ a 'history' of replaced points $a_i \in A_{l_i}$ where Equation (4.6) holds for a_i and a_{i-1} which has to be finite as $F(Q)$ is bounded.

Assume that such an integer l_0 as claimed above does not exist, that is, that $P_{Q,\epsilon} \not\subset C_{A_l,\Delta}$ for all $l \in \mathbb{N}$. Hence, there exists a sequence of points

$$p_i \in P_{Q,\epsilon} : \quad p_i \in P_{Q,\epsilon} \backslash C_{A_i,\Delta} \quad \forall i \in \mathbb{N}. \tag{4.7}$$

Since $P_{Q,\epsilon} \subset Q$ and Q is compact there exists an accumulation point $p^* \in \overline{P_{Q,\epsilon}}$, that is, there exists a subsequence $\{i_j\}_{j\in\mathbb{N}}$ with

$$p_{i_j} \to p^* \in \overline{P_{Q,\epsilon}} \text{ for } j \to \infty. \tag{4.8}$$

By (8) it follows that the set

$$\tilde{U}_1 := B^{\infty}_{(\Delta_x - \Delta^*_x)/2}(p^*) \cap \overset{\circ}{P}_{Q,\epsilon}, \tag{4.9}$$

is not empty. By (2.46) it follows that there exists with probability one an $l_1 \in \mathbb{N}$ and an element $\tilde{x}_1 \in P_{l_0+l_1}$ generated by *Generate()* with $\tilde{x}_1 \in \tilde{U}_1$. There are two cases for the archive $A_{l_0+l_1}$: (a) x_1 can be discarded from the archive, or (b) x_1 is added to it.

Assume first that x_1 is discarded. Since, $x_1 \in P_{Q,\epsilon}$ there exists no $\bar{x} \in Q$ such that $\bar{x}$ $-\epsilon$-dominates x_1. Hence, $(D1)$ can not occur (see (4.1)), and thus, there must exist an $a_2 \in A_{l_0+l_1}$ such that $\|F(a_2) - F(x_1)\|_\infty \leq \Delta^*_y$ and $\|a_2 - x_1\| \leq \Delta^*_x$ (see $(D2)$). Thus, whether x_1 is added to the archive or not there exists an $\tilde{a}_1 \in A_{l_0+l_1}$ such that $\|\tilde{a}_1 - y^*\|_\infty \leq \Delta_x$ (since in case x_1 is added to the archive $\tilde{a}_1 = x_1$ can be chosen), and we obtain

$$\|\tilde{a}_1 - \tilde{y}\|_\infty \leq \|\tilde{a}_1 - x_1\|_\infty + \|x_1 - \tilde{y}\|_\infty < \Delta_x \quad \forall \tilde{y} \in \tilde{U}_1 \tag{4.10}$$

By (4.7) and (4.8) there exist integers $j_1, \tilde{l}_1 \in \mathbb{N}$ with

$$y_{i_{j_1}} \in \tilde{U}_1 \backslash C_{l_0+l_1+\tilde{l}_1,\Delta_x}. \tag{4.11}$$

Since by (4.10) it holds that $\|y_{i_{j_1}} - a_1\|_\infty < \Delta_x$ it follows that $a_1 \notin A_{l_0+l_1+\tilde{l}_1}$, which is only possible via a replacement in Algorithm 10 (lines 4 to 7).

In an analogous way a sequence $\{a_i\}_{i\in\mathbb{N}}$ of elements can be constructed which have to be replaced by other elements. Since, this leads to a sequence of infinitely many replacements. This is a contradiction to the assumption, and the proof is complete.

(ii) let $\tilde{A}$ and l_0 be as above, and let $l \geq l_0$. Further, let $x \in Q\backslash B_{2\Delta_x}(P_{Q,\epsilon+2\Delta_y})$, that is, there exists a $p \in P_{Q,\epsilon}$ such that $p \prec_{-(\epsilon+2\Delta_y)} x$. Since, $l \geq l_0$ there exists an $a \in \tilde{A} \subset A_l$ such that $\|F(p) - F(a)\|_\infty < \Delta_y$. Combining both facts we see that $a \prec_{-(\epsilon+\Delta_y)} x$ and $dist(a, x) \geq \Delta_x$ (line 6). Thus, no element $x \in Q\backslash B_{2\Delta_x}(P_{Q,\epsilon+2\Delta_y})$ is contained in $A_l, l \geq l_0$, or will ever be added to the archive further on. The claim follows since the archive can only contain elements in $B_{2\Delta_x}(P_{Q,\epsilon+2\Delta_y})$.

(iii) follows immediately by (i) and (ii).

□

4.1.3 Bounds on the archiver sizes

Finally, we analyze the bounds of the archive sizes obtained by Algorithm 10. Note that for the special case $\Delta_x = \infty$ Algorithm 10 is equivalent to $ArchiveUpdateP_{Q,\epsilon}D_y$ those upper bounds are discussed above. For $\Delta_x < \infty$ we have

$$|A_l^x| \leq \left(\frac{1}{\Delta^*} + 1\right)^n \prod_{j=1}^{n} (b_i - a_i), \tag{4.12}$$

where $Q \subset [a_1, b_1] \times \ldots \times [a_n, b_n]$ ($P_{Q,\epsilon+2\Delta}$ is included in $[a_1, b_1] \times \ldots \times [a_n, b_n]$, and maximal $1/\Delta_x^* + 1$ elements can be placed in each coordinate direction). To see that this bound is tight we consider the example

$$F : [0, 1]^n \to \mathbb{R}^k, \quad F(x) \equiv c_0 \in \mathbb{R}^k, \tag{4.13}$$

and let $\Delta_x = 1/s,\ s \in \mathbb{N}$. Define $x_{i_1,\ldots,i_n} = (i_1\Delta_x, \ldots, i_n\Delta_x)$ and

$$\mathcal{D} := \{x_{i_1,\ldots,i_n} | 0 \leq i_1, \ldots, i_n \leq s\}. \tag{4.14}$$

Since $\mathcal{D} \subset [0, 1]^n$ and $d_\infty(z_1, z_2) \geq \Delta_x > \Delta_x^*$ for all $z_1, z_2 \in \mathcal{D}$, $z_1 \neq z_2$, all points in $\mathcal{D}$ will be accepted by the archiver (assuming that only points $z \in \mathcal{D}$ are inserted) leading to an archive A with $|A| = |\mathcal{D}| = (s + 1)^n$.

Since $P_{Q,\epsilon}$ is n-dimensional, the growth of the magnitudes of the archives is also beyond this constructed example of order

$$\mathcal{O}\left(\left(\frac{1}{\Delta_x}\right)^n\right) \quad \text{for} \quad \Delta_x \to 0. \tag{4.15}$$

4.1.4 Numerical Results

Now, we present the numerical results of the novel archiver. First, we show an experiment with uniform sampling. Next, we use a non-uniform sampling which simulates the behavior of an evolutionary algorithm. Finally, we show the capabilities of the archiver on a non-uniform beam problem.

Uniform Sampling

First, we investigate the effect of all the archivers where we use uniform sampling of the candidate solutions within the domain Q. As first example, we consider the problem OKA1 to demonstrate the effect of the different strategies. We generated 20 files with $100,000$ candidate solutions that were uniformly sampled from Q. Then, we applied the archivers on each file in order to assure that each archiver is fed with the same data.

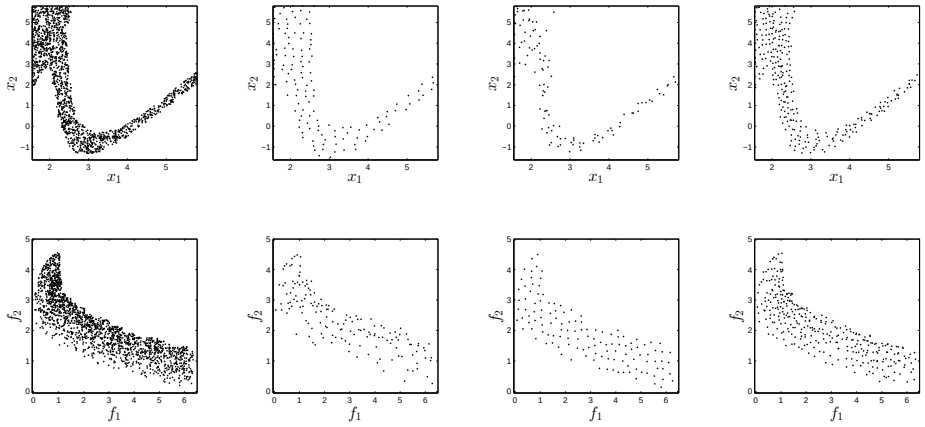
We used $\epsilon = [1, 1], \Delta_x = [0.2, 0.2]$, and $\Delta_y = [0.2, 0.2]$. In order to compare the approaches, we measured the Δ_2 indicator value (Schütze et al. (2012), using $p = \infty$), which measures the Hausdorff distance between two sets, in decision as well as in objective space. To construct the reference set for the Δ_2 computation, we used uniform sampling within Q using $10,000,000$ candidates.

Figure 4.1 shows the median results of each archiver on the benchmark problems for both decision and objective space with respect to Δ_2. Figure 4.2 shows the box plots for the different indicator values. From the results we can see that $ArchiverP_{Q,\epsilon}$ obtains the best results for this problem, however, with a much higher computational cost. $ArchiverP_{Q,\epsilon}D_x$ obtains a good distribution in decision space while $ArchiverP_{Q,\epsilon}D_y$ obtains a good distribution in objective space. $ArchiverP_{Q,\epsilon}D_{xy}$ obtains similar Δ_2 values compared to those of $ArchiverP_{Q,\epsilon}D_x$ (decision space) and $ArchiverP_{Q,\epsilon}D_y$ (objective space). However, note that the approximation qualities of $ArchiverP_{Q,\epsilon}D_{xy}$ come with a (slightly) higher cost in both time and archive size.

Next, we study the MOPs Two-on one, Sym-part (Rudolph et al. (2007)), and SSW (Schaeffler et al. (2002)), using the same setting of $100,000$ uniformly sampled candidate solutions. Table 4.1 shows the parameters used for each problem. To construct the reference set for the Δ_2 computation, we used Simple Cell Mapping Hernández et al. (2013) with a grid of 100×100 test points for the two-dimensional problems and $20 \times 20 \times 20$ for the three-dimensional problem.

Figures 4.3, 4.4 and 4.5 show the median results of each archive on the benchmark problems for both decision and objective space with respect to Δ_2 in decision space. Tables 4.2, 4.3 and 4.4 show the mean values of the averaged values on the different indicators. Further, Figure 4.6 shows the box plots for the different indicator values.

For Two-on one, $ArchiveUpdateP_{Q,\epsilon}D_{xy}$ shows better results in decision space than $ArchiveUpdateP_{Q,\epsilon}D_y$ and it is not possible to conclude which one is better in objective space according to the Wilcoxon rank sum test. This is not surprising since there are disconnected components that map to the same region in objective space. Thus, the discretization in objective space Δ_y becomes entire responsible for



(a) $AP_{Q,\epsilon}$ (b) $AP_{Q,\epsilon}$ (c) $AP_{Q,\epsilon}D_y$ (d) $AP_{Q,\epsilon}D_{xy}$

Figure 4.1: Set of approximate solutions on OKA1 with $\epsilon = 0$, $\Delta_x = 1$ and $\Delta_y = 0.01$.

Table 4.1: Parameters used on benchmark problems.

Problem	ϵ	Δ_y	Δ_x
Two-on one	[0.15, 0.15]	0.1	0.02
Sym-part	[0.15, 0.15]	0.2	1
SSW n=3	[0.1, 0.001]	$\frac{1}{3}\epsilon$	1

the acceptance rule in the archiver. In Sym-part, $ArchiveUpdateP_{Q,\epsilon}D_{xy}$ can be seen as an intermediate result between the other archivers. It captures all 9 connected components which leads to similar results of the ones of $ArchiveUpdateP_{Q,\epsilon}$ on Δ_2 indicator. While, it represents the set with 4 times fewer solutions. For SSW, all archivers show the same behavior in objective space, but $ArchiveUpdateP_{Q,\epsilon}D_y$ has the worst Δ_2 value in decision space. However, in this case more than 20% of the search space is in the set $P_{Q,\epsilon}$. Thus, the size of the set increases a lot which shows the potential problem of $ArchiveUpdateP_{Q,\epsilon}D_{xy}$ if the discretization values are not chosen properly.

Table 4.2: Averaged Δ_2 values on benchmark problems (decision space).

Problem	$ArchiveP_{Q,\epsilon}$	$ArchiveP_{Q,\epsilon}D_y$	$ArchiveP_{Q,\epsilon}D_{xy}$
Two-on one	**0.0347**	0.1551	0.1067
Sym-part	**0.2374**	5.3890	0.3970
SSW n=3	**4.2106**	5.0036	4.2119

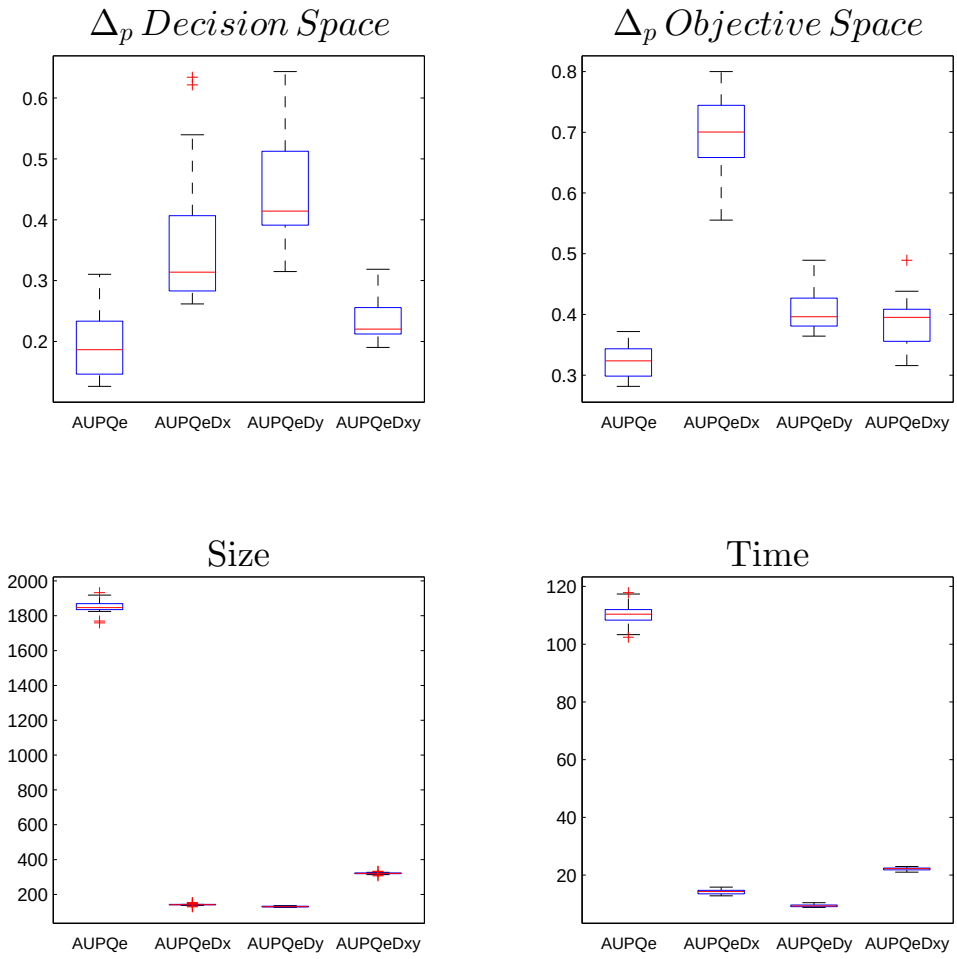


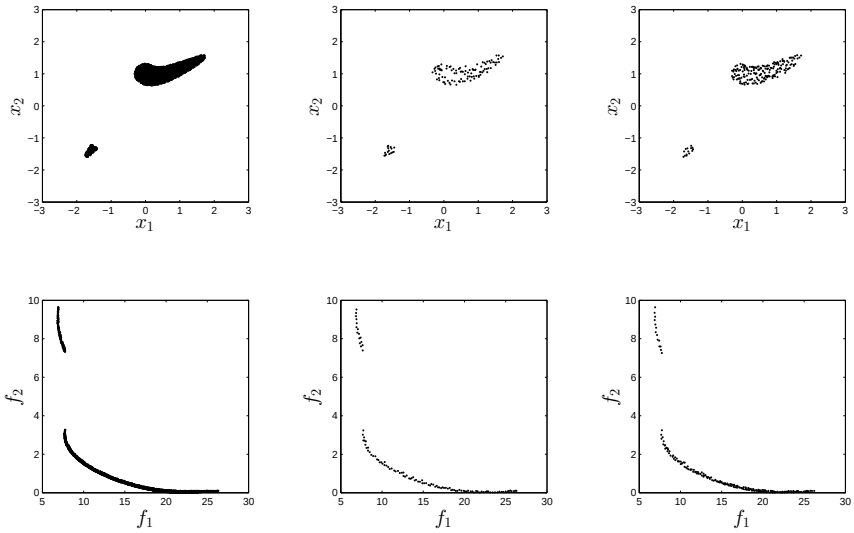
Figure 4.2: Box plots of Δ_2 in decision/objective space, archive size and computational time on OKA1.

Table 4.3: Averaged Δ_2 values on benchmark problems (objective space).

Problem	$ArchiveP_{Q,\epsilon}$	$ArchiveP_{Q,\epsilon}D_y$	$ArchiveP_{Q,\epsilon}D_{xy}$
Two-on one	**0.0880**	0.1957	0.1906
Sym-part	**0.1157**	0.2035	0.1569
SSW n=3	**1.9283**	**1.9283**	**1.9283**

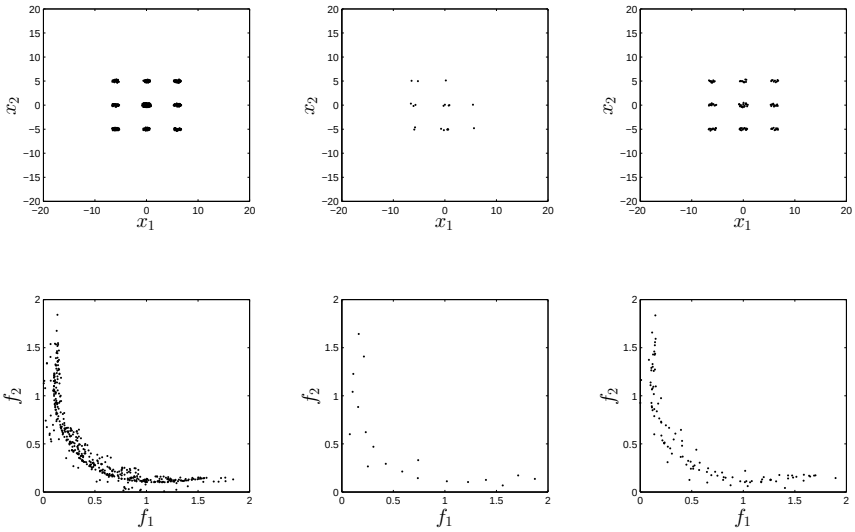
Table 4.4: Averaged archive sizes on benchmark problems.

Problem	$ArchiveP_{Q,\epsilon}$	$ArchiveP_{Q,\epsilon}D_y$	$ArchiveP_{Q,\epsilon}D_{xy}$
Two-on one	2778.6	**117.8**	117.8
Sym-part	442.6	**20.3**	104.6
SSW n=3	21360	**4194.3**	18708



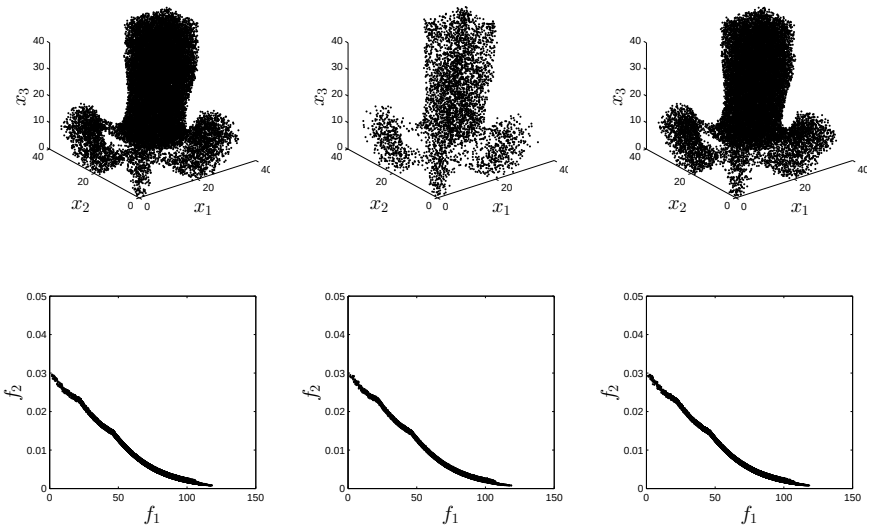
(a) $ArchiveP_{Q,\epsilon}$ (b) $ArchiveP_{Q,\epsilon}D_y$ (c) $ArchiveP_{Q,\epsilon}D_{xy}$

Figure 4.3: Set of approximate solutions of MOP (Two-on one) with $\epsilon = 0$, $\Delta_x = 1$ and $\Delta_y = 0.01$.



(a) $ArchiveP_{Q,\epsilon}$ (b) $ArchiveP_{Q,\epsilon}D_y$ (c) $ArchiveP_{Q,\epsilon}D_{xy}$

Figure 4.4: Set of approximate solutions of MOP (Sym-part) with $\epsilon = 0$, $\Delta_x = 1$ and $\Delta_y = 0.01$.



(a) $ArchiveP_{Q,\epsilon}$ (b) $ArchiveP_{Q,\epsilon}D_y$ (c) $ArchiveP_{Q,\epsilon}D_{xy}$

Figure 4.5: Set of approximate solutions of SSW with $\epsilon = [0.1, 0.001]$, $\Delta_x = 1$ and $\Delta_y = \frac{1}{3}\epsilon$.

Non-uniform Sampling

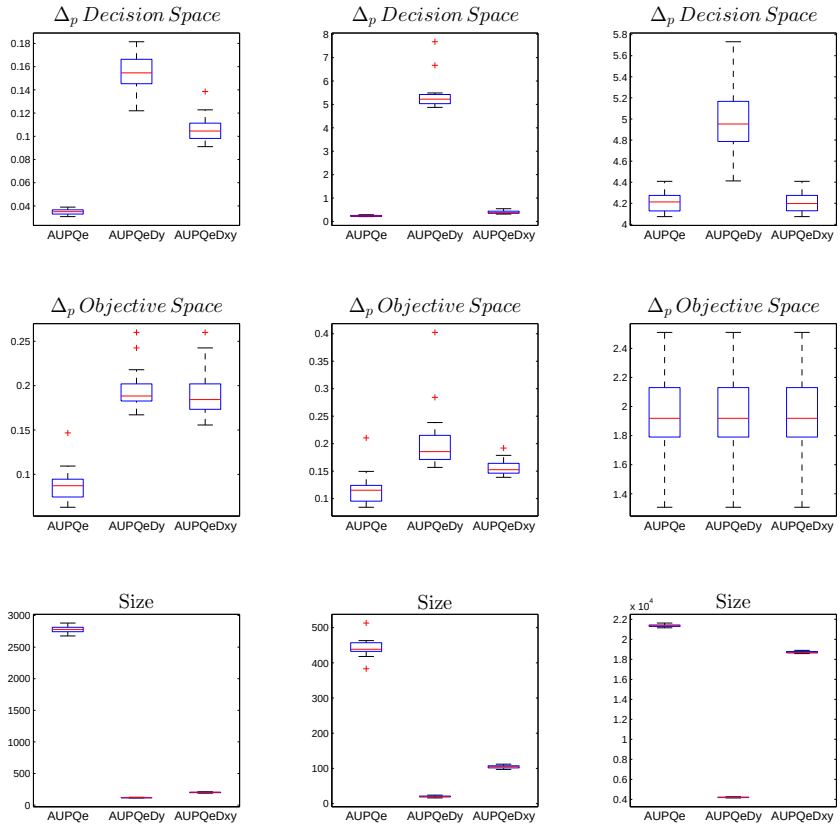
So far we used uniform sampling to see the effect of the different archivers. If, however, an evolutionary algorithm or another stochastic search algorithm is used, typically one region of the search space is exploited first before the rest is explored. Here, we try to simulate such a case for Sym-part. We generated 20 archives in the following manner. First, we generated $20,000$ uniform random points in the region $[-3, 3]^2$ and fed the archivers with this data. Next, we generated another $80,000$ uniform random points in the whole domain and updated the archives by this information.

Figure 4.7 shows of the median results of each archive on Sym-part problem for both decision and objective space with respect to Δ_2 in decision space. Next, Table 4.5 shows the mean values of the averaged values on the Δ_2 indicator. Further, Figure 4.8 shows the box plots for the different indicator values.

From the results, we can observe that $ArchiveP_{Q,\epsilon}D_{xy}$ captures all 9 connected components while $ArchiveP_{Q,\epsilon}D_y$ keeps only the component that was visited first and a few solutions in other connected components. Thus, $ArchiveP_{Q,\epsilon}D_{xy}$ leads here to better results in both decision and objective space which is also in general the case at least for complex MOPs by construction of the archiver.

Application to Non-uniform Beam

Finally, we consider a non-uniform beam problem taken from Sun (1995). Non-uniform beams present a commonly used structure in many applications such as ship hull, rocket surface and aircraft fuselage. In this application structural and acoustic

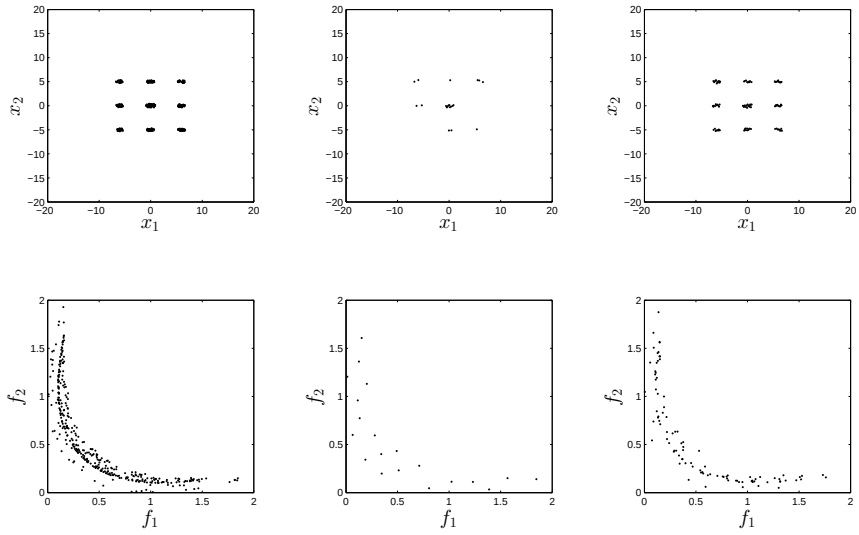


(a) Two-on one (b) Sym-part (c) SSW

Figure 4.6: Box plots of Δ_2 in decision/objective space and archive size on the benchmark problems.

Table 4.5: Averaged Δ_2 values on RUDOLPH.

Archiver	Decision space Δ_2	Objective space Δ_2
$ArchiveP_{Q,\epsilon}$	**0.2583**	**0.1228**
$ArchiveP_{Q,\epsilon}D_y$	4.8192	0.2146
$ArchiveP_{Q,\epsilon}D_{xy}$	0.4047	0.1714



(a) $ArchiveP_{Q,\epsilon}$ (b) $ArchiveP_{Q,\epsilon}D_y$ (c) $ArchiveP_{Q,\epsilon}D_{xy}$

Figure 4.7: Set of approximate solutions Sym-part with $\epsilon = 0$, $\Delta_x = 1$ and $\Delta_y = 0.01$.

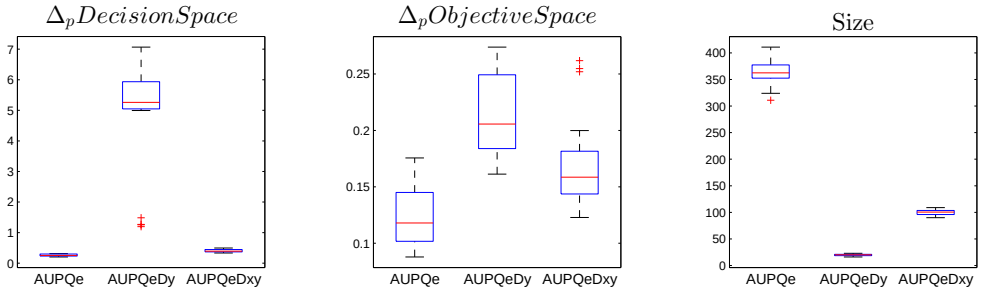


Figure 4.8: Box plots of Δ_2 in decision/objective space and archive size on on Sym-part.

properties are presented as constraints and performance indexes of candidate non-uniform beams.

The primary goal of structural-acoustic optimization is to design a light-weight and quiet structure. Thus, it is a bi-objective optimization problem where the objectives are weight reduction and sound isolation. The number of interpolation coordinates to describe the thickness profile of the beam is 10 (i.e., the decision space is 10-dimensional) and the number of beam discretizations for the response calculation with the transfer matrix method is 50. Here, we consider a simply supported beam design. Mass constraint is 10-30kg; minimum fundamental frequency is 8Hz and maximum successive segment height difference of the discretized beam is 1.75mm. Searching range of spline coordinates is 1-15mm. Beam length is 1.0m. For more details we refer to Sun (1995).

The parameters for the archiver were set to $\epsilon = [1, 0.00003]$, $\Delta_x = 2.5 \times 10^{-3}$ and $\Delta_y = [0.3, 0.000003]$. That is to say, we are willing to accept a deterioration of 1 Kg and 0.3×10^{-5} W/m/s. In this case, we took 10 million points uniformly sampled from the domain $Q = [1e-3, 1.5e-2]^{10}$.

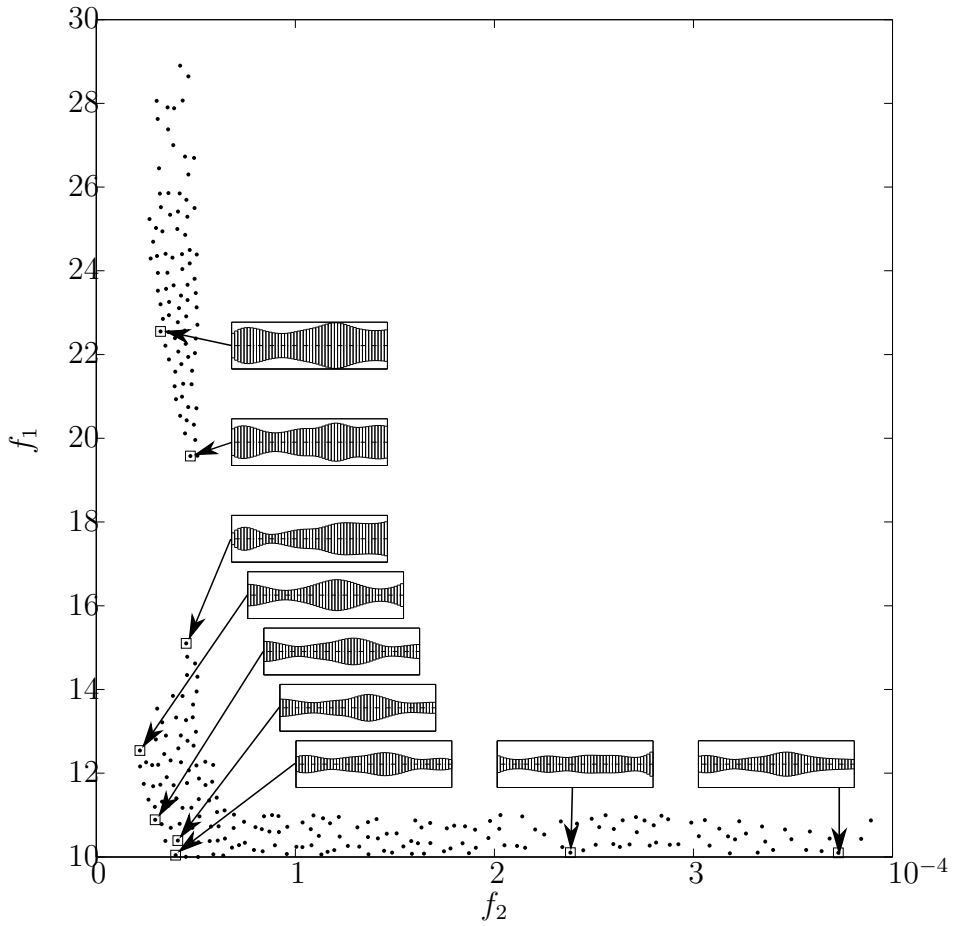


Figure 4.9: Set of approximate solutions for the non-uniform beam problem.

Figure 4.9 shows the approximated set $P_{Q,\epsilon}$ together with the cross sections of nine selected beams, where three of them are non-dominated and the other six ones are nearly optimal. From this we can observe that it is possible to design beams that have similar objective values while having a completely different shape. Thus, the additional consideration of nearly optimal solutions leads in this case to a wider range of geometrical choices for possible structural designs.

4.2 Multi-objective Evolutionary Algorithm for the Set of Approximate Solutions

In this section, we present an evolutionary multi-objective algorithm that aims for a finite representation of $P_{Q,\epsilon}$.

4.2.1 Ranking nearly optimal solutions

First, we present an algorithm in order to rank nearly optimal solutions. The algorithm is an adaption from NSGA-II. In this case, the best solutions are those that are non ϵ-dominated by any other solution in the population and are also well distributed in both decision and objective space. The second layer will be formed with those solutions the solutions that are non ϵ-dominated but were not well distributed. The next layers follow the same pattern.

Algorithm 11 Non ϵ-Dominated Sorting

Require: $P_0, \epsilon, \Delta_x, \Delta_y$
Ensure: F
1: **while** $p \neq \emptyset$ **do**
2: $A_{i+1} \leftarrow MArchive P_{Q,\epsilon} D_{xy}(P_i, [], \epsilon, \Delta_x, \Delta_y)$
3: $P_{i+i} \leftarrow P_i \cap A_{i+1}$
4: $F_i \leftarrow A_{i+1}$
5: **end while**

4.2.2 Avoiding weakly dominated solutions

The archivers presented before assume that no weak Pareto optimal solutions exits. However, this is not always the case in practice. Here, we modify the archiver to remove weakly dominated solutions. In order to achieve this, we propose to approximate the nadir point. Algorithm 12 shows the modified archiver. The difference with respect to the original archiver can be found in line 3. The archiver first checks if the candidate solution p dominates $nadir + 2\epsilon$ and then it continues as usual.

Algorithm 12 $A := MArchiveUpdateP_{Q,\epsilon}D_{xy}\ (P, A_0, \epsilon, \Delta_x, \Delta_y)$

Require: population P, archive A_0, $\epsilon \in \mathbb{R}^n_+$, $\Delta_x \in \mathbb{R}_+$, $\Delta_y \in \mathbb{R}_+$, $\Delta^*_x \in (0, \Delta_x)$, $\Delta^*_y \in (0, \Delta_y)$, $nadir \in \mathbb{R}$

Ensure: updated archive A

1: $A := A_0$
2: **for all** $p \in P$ **do**
3: **if** $p \preceq nadir + 2\epsilon$ and $\nexists a_1 \in A : a_1 \prec_{-\epsilon} p$ and $\nexists a_2 \in A : (d_\infty(F(a_2), F(p)) \leq \Delta^*_y$ and $d_\infty(a_2, p) \leq \Delta^*_x)$ **then**
4: $A \leftarrow A \cup \{p\}$
5: $\hat{A} = \{a_1 \in A | \nexists a_2 \in A : a_2 \prec_{-(\epsilon+\Delta_y)} a_1\}$
6: **for all** $a \in A \backslash \hat{A}$ **do**
7: **if** $p \prec_{-(\epsilon+\Delta_y)} a$ and $dist(a, \hat{A}) \geq 2\Delta_x$ **then**
8: $A \leftarrow A \backslash \{a\}$
9: **end if**
10: **end for**
11: **end if**
12: **end for**

4.2.3 The algorithm

Algorithm 13 presents the complete algorithm to approximate the set of nearly optimal solutions. First, the algorithm initializes a random population and generates the first archiver with the best solutions found. Next, in each iteration the genetic operators are applied and then the solutions are ranked using Algorithm 11 and the best solutions are kept. Finally, the archiver is applied in order to keep the non ϵ-dominated solutions. It is important to notice that the algorithm uses diversity in both decision and objective space. This is due to the fact that nearly optimal solutions can be similar in objective space but very different in decision space. In order to achieve it, half the population is selected according their NearestNeighborDistance in objective space and half using decision space.

Complexity of NϵSGA

In the following, we comment on the complexity of NϵSGA. Let G be the number of generations and $|P|$ the size of the population.

- Initialization: in this step all the individuals are generated at random and evaluated. This task takes $O(|P|)$
- External archiver: from the previous section, the complexity of the archiver is $O(|P||A|)$, where $|A|$ is the size of the archiver. Note that the maximum size of the archiver at the end of the execution is $G|P|$.
- Non ϵ-dominated sorting: until there are no solution in the population P, $MArchiveUpdateP_{Q,\epsilon}D_{xy}$ is executed. In the worst case each rank has only

one solution and the archiver is executed $|P|$ times. Thus, the complexity of this part of the algorithm is $O(|P|^3)$.

- Main loop: the main loop selected the parents and applies the genetic operators (each of this tasks take $O(|P|)$), then the algorithm uses the Non ϵ-dominated sorting ($O(|P|^3)$), next it finds those solutions that are well distributed ($O(|P|^2)$) and finally applies the external archiver $O(|P|^2)$. Note that all these operations are executed G times. Thus, the complexity of the algorithm is $O(max(G|P|^3, (G|P|)^2))$. Which is the maximum of the Non ϵ-dominated sorting or the external archiver.

Algorithm 13 Non ϵ-Dominated Sorting EMOA

Require: number of generations: $ngen$, population size: $npop$, $\epsilon \in \mathbb{R}^n_+$, $\Delta_x \in \mathbb{R}_+$, $\Delta_y \in \mathbb{R}_+, \Delta^*_x \in (0, \Delta_x), \Delta^*_y \in (0, \Delta_y)$
Ensure: updated archive A

Generate initial population P_1
$A \leftarrow MArchive P_{Q,\epsilon} D_{xy}(P_0, [], \epsilon, \Delta_x, \Delta_y, nadir)$
for $i = 1 \ldots ngen$ **do**
    Select $\lceil \frac{\lambda}{2} \rceil$ Parents Q_i with Tournament
    $\tilde{O}_i \leftarrow SBXCrossover(Q_i)$
    $O_i \leftarrow PolynomialMutation(\tilde{O}_i)$
    $\tilde{P}_i \leftarrow P_i \cup O_i$
    $F \leftarrow$ Non ϵ-DominatedSorting$(\tilde{P}_i)$
    $P_{i+1} = \emptyset$
    $j = 1$
    while $|P_{i+1}| + |F_j| \leq npop$ **do**
        $NearestNeighborDistance(F_j)$ ▷ in both parameter/objective space
        $P_{i+1} = P_{i+1} \cup F_j$
        $j \leftarrow j + 1$
    end while
    $Sort(F_i, \prec_n)$
    $P_{i+1} = P_{i+1} \cup F_j[1 : npop - |P_{i+1}|]$
    $A_{i+1} \leftarrow MArchive P_{Q,\epsilon} D_{xy}(O_i, A_i, \epsilon, \Delta_x, \Delta_y, nadir)$
end for

4.2.4 Using subpopulations

Here, we propose the use of two subpopulations in order to improve the pressure towards the set of interest of the novel algorithm. Each subpopulation has a different aim. The first one aims to approximate the global Pareto set/front. While the second one aims to approximate $P_{Q,\epsilon}$. A key aspect of the approach is how the information between the subpopulations is shared. Thus, in the following we describe two schemes to share the information via migration.

1. Every nf generations, nr individuals are exchange between the population. The individuals are chosen at random from the rank 1 individuals.

2. Every nf generations, the crossover is performed between the populations. One parent is chosen from each population at random.

As it can be observed. The use of subpopulations introduces three extra parameters to the algorithm, namely:

- nf: frequency of migration,

- nr: the number of individuals to be exchanged, and

- $ratio$: the ratio of individuals between subpopulation in the range $[0, 1]$.

4.2.5 Numerical Results

In this section, we present the numerical result corresponding the evolutionary algorithm. We performed several experiments in order to validate the novel algorithm. First, we enhance the state-of-the-art algorithms with the novel $ArchiverUpdateP_{Q,\epsilon}D_{xy}$. Then, we validate the use of an external archiver in the evolutionary algorithm. Next, we compare several some strategies of subpopulations. Finally, we compare the resulting algorithm with the enhanced state-of-the-art algorithms.

In all cases we used the following problems: Deb99, two-on-one, sym-part, ssw, omni test, Lamé superspheres. Next, 20 independent experiments were performed with the following parameters: 100 individuals, 100 generations, crossover rate = 0.9, mutation rate = $1/n$, $eta\_c = 20$ and $eta\_m = 20$. Then, Table 4.6 shows the ϵ, Δ_y and Δ_x values used for each problem. Further, for all cases were measured with the Δ_2 indicator and a Wilcoxon ranksum test was performed with a significance level $\alpha = 0.05$. Finally, all Δ_p tables, the bold font represents the best mean value and the arrows represent:

- $\uparrow$ ranksum rejects the null hypothesis of equal medians and the algorithm has a better median,

- $\downarrow$ ranksum rejects the null hypothesis of equal medians and the algorithm has a worse median and

- $\leftrightarrow$ ranksum cannot reject the null hypothesis of equal medians.

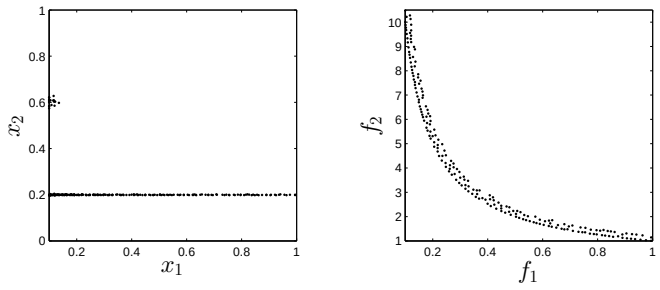
The comparison is always made with the algorithm in the first column.

Table 4.6: Parameters used for all experiments.

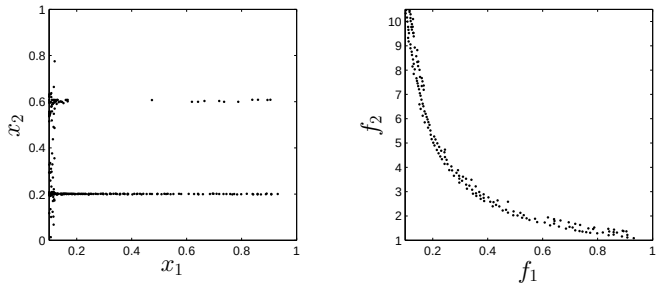
Problem	ϵ	Δ_y	Δ_x
Deb99	[0.011 0.011]	[0.2 0.2]	[0.2 0.2]
Two-on-one	[0.1, 0.1]	[0.1 0.1]	[0.02 0.2]
Sym-part	[0.15, 0.15]	[0.2 0.1]	[1 1]
SSW n=3	[0.011, 0.0001]	$\frac{1}{3}\epsilon$	[1 1 1]
Omni-test	[0.01 0.01]	[0.05 0.05]	[0.1 0.1 0.1 0.1]
LSS	[0.1 0.1]	[0.05 0.05]	[0.1 0.1 0.1 0.1 0.1]

Experiment 1: enhancing the state-of-the-art algorithms

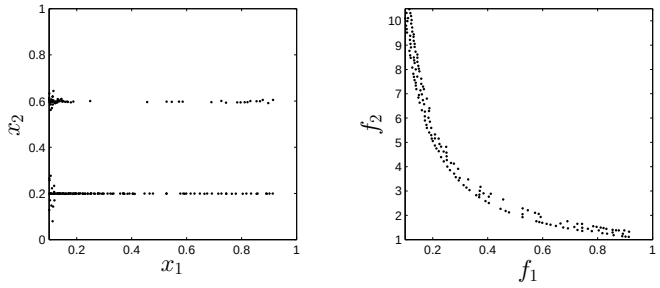
Before analyzing the proposed algorithm, we present the numerical results of adapted versions $P_{Q,\epsilon}$NSGA-II and $P_{Q,\epsilon}$MOEA using $ArchiverUpdateP_{Q,\epsilon}D_{xy}$ against the original versions that use $ArchiverUpdateP_{Q,\epsilon}D_y$. As it could be expected, the results show that allowing the algorithms to keep solutions that are well distributed in decision space improved their performance according to the Δ_2 indicator. Figures 4.10-4.15 show the median approximations to $P_{Q,\epsilon}$ and $F(P_{Q,epsilon})$ obtained by the algorithms. In all cases, it is possible to observe that the algorithms using the novel archiver have a better distribution in decision space, as expected, while maintaining a good distribution in objective space. Figure 4.16 and Figure 4.17 show the box plots in both decision and objective space. From the results, we can observe that in none of the cases the algorithms that use $ArchiverUpdateP_{Q,\epsilon}$ are superior to those using $ArchiverUpdateP_{Q,\epsilon}D_{xy}$ which indicates an advantage of using the novel archiver.



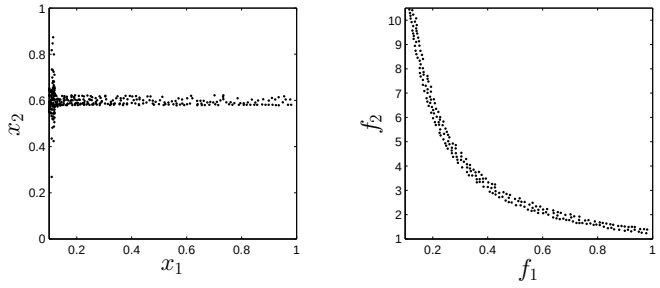
(a) $P_{Q,\epsilon}$NSGA-II



(b) $P_{Q,\epsilon}D_{xy}$NSGA-II

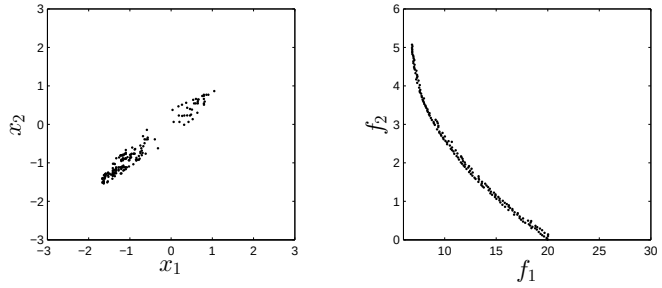


(c) $P_{Q,\epsilon}$MOEA

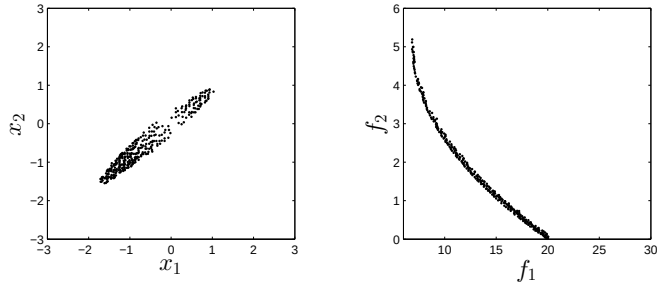


(d) $P_{Q,\epsilon}D_{xy}$MOEA

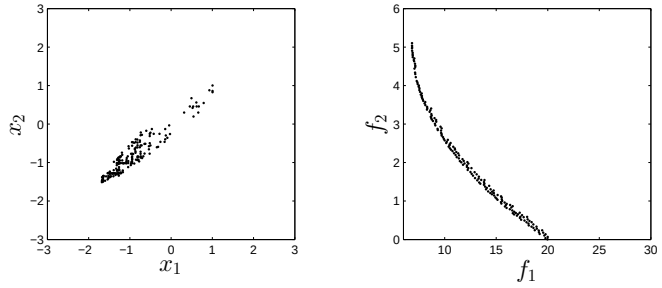
Figure 4.10: Numerical results on Deb99 for the state-of-the-art algorithms.



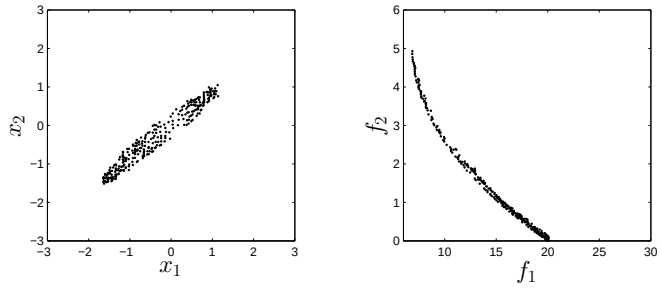
(a) $P_{Q,\epsilon}$NSGA-II



(b) $P_{Q,\epsilon}D_{xy}$NSGA-II

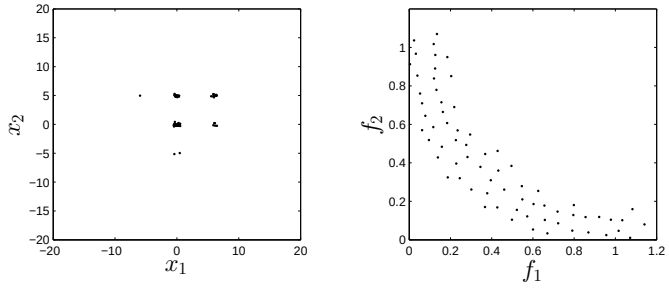


(c) $P_{Q,\epsilon}$MOEA

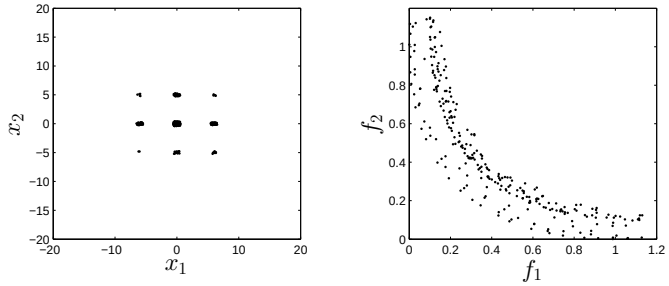


(d) $P_{Q,\epsilon}D_{xy}$MOEA

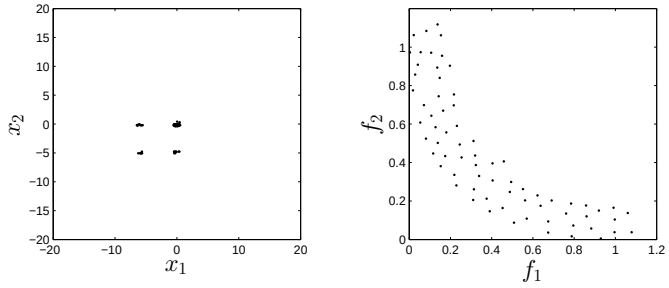
Figure 4.11: Numerical results on Two-on-one for the state-of-the-art algorithms.



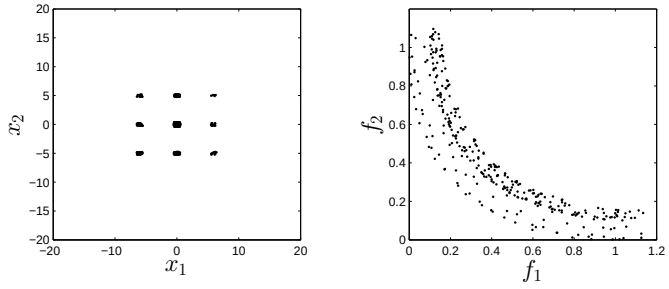
(a) $P_{Q,\epsilon}$NSGA-II



(b) $P_{Q,\epsilon}D_{xy}$NSGA-II

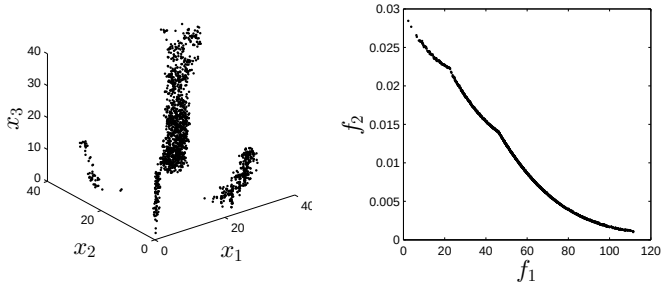


(c) $P_{Q,\epsilon}$MOEA

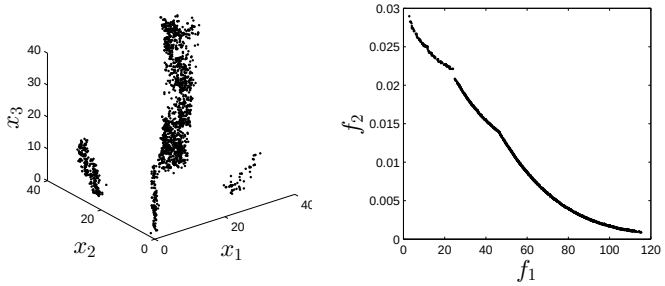


(d) $P_{Q,\epsilon}D_{xy}$MOEA

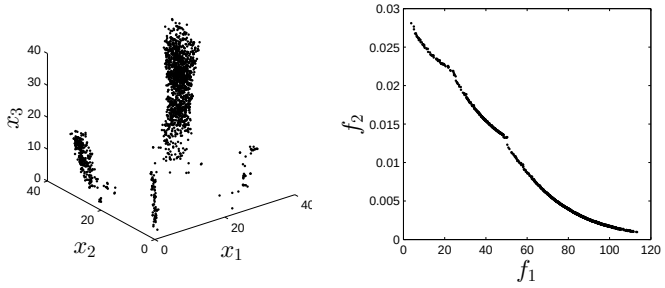
Figure 4.12: Numerical results on sym-part for the state-of-the-art algorithms.



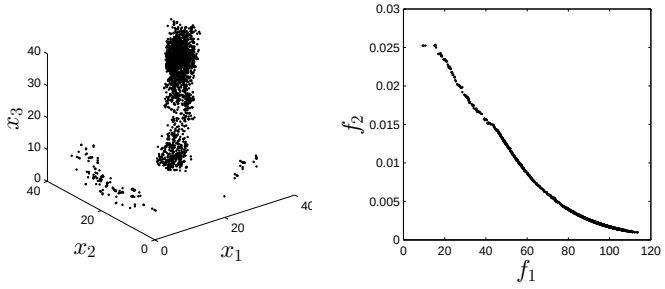
(a) $P_{Q,\epsilon}$NSGA-II



(b) $P_{Q,\epsilon}D_{xy}$NSGA-II

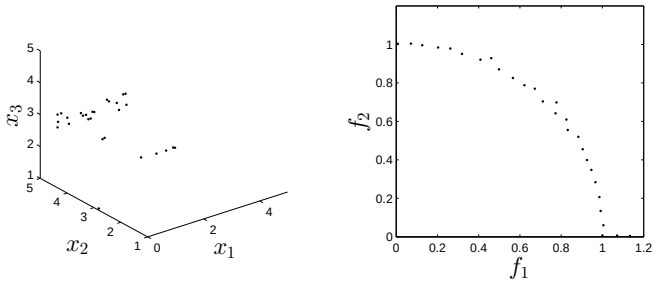


(c) $P_{Q,\epsilon}$MOEA

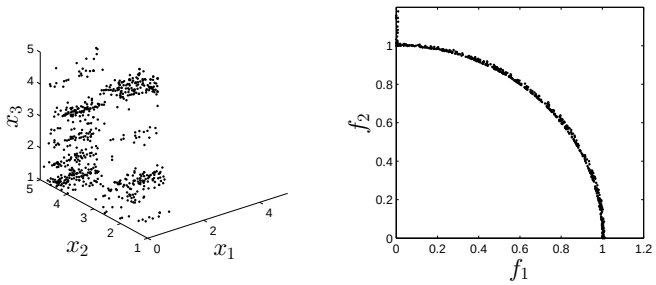


(d) $P_{Q,\epsilon}D_{xy}$MOEA

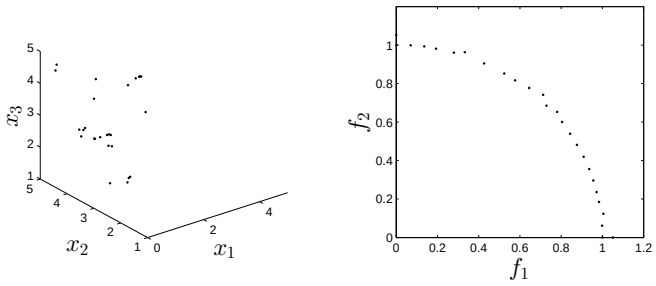
Figure 4.13: Numerical results on SSW for the state-of-the-art algorithms.



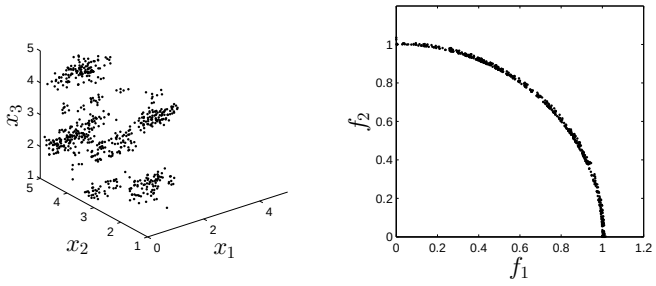
(a) $P_{Q,\epsilon}$NSGA-II



(b) $P_{Q,\epsilon}D_{xy}$NSGA-II

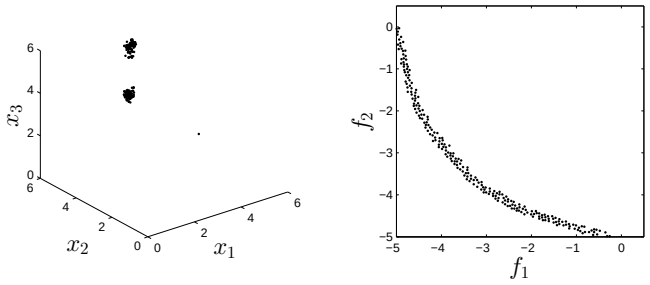


(c) $P_{Q,\epsilon}$MOEA

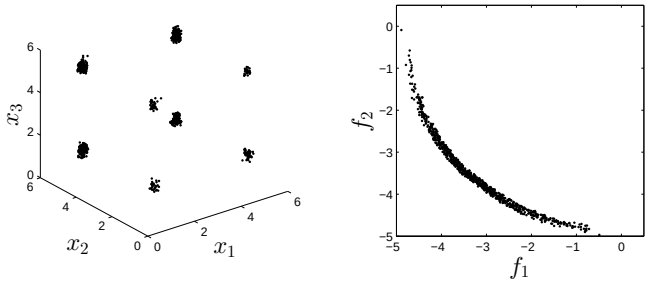


(d) $P_{Q,\epsilon}D_{xy}$MOEA

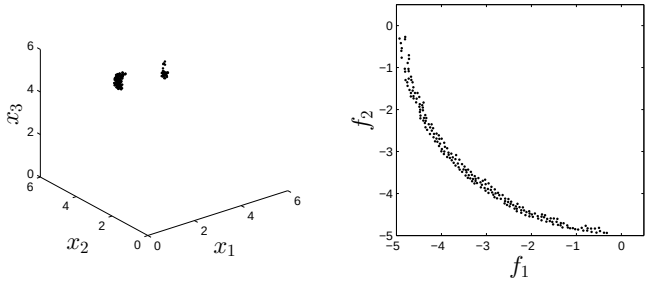
Figure 4.14: Numerical results on Omni test for the state-of-the-art algorithms.



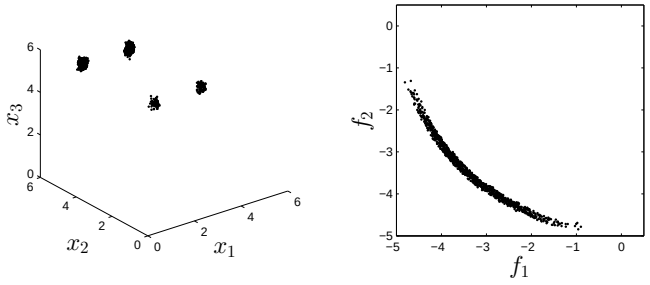
(a) $P_{Q,\epsilon}$NSGA-II



(b) $P_{Q,\epsilon}D_{xy}$NSGA-II

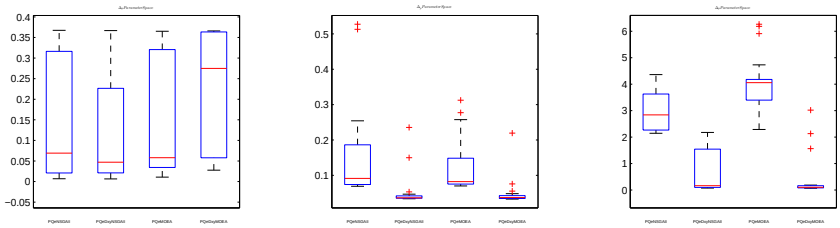


(c) $P_{Q,\epsilon}$MOEA

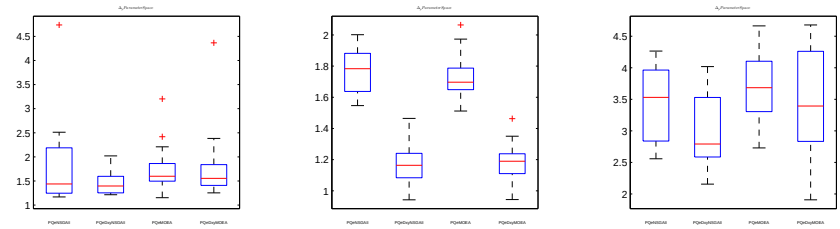


(d) $P_{Q,\epsilon}D_{xy}$MOEA

Figure 4.15: Numerical results on Lamé Superspheres for the state-of-the-art algorithms.

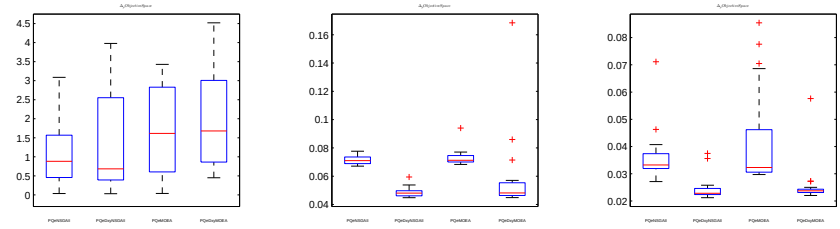


(a) Deb99 (b) Two-on-one (c) Sym-part

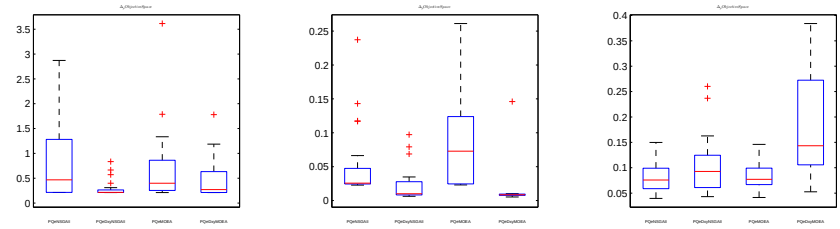


(d) SSW (e) Omni test (f) Lamé Superspheres

Figure 4.16: Box plots of Δ_2 in decision space for $P_{Q,\epsilon}$-NSGA2, $P_{Q,\epsilon}D_{xy}$-NSGA2, $P_{Q,\epsilon}$-MOEA and $P_{Q,\epsilon}D_{xy}$-MOEA (from left to right).



(a) Deb99 (b) Two-on-one (c) Sym-part

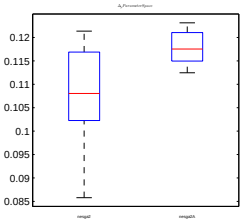


(d) SSW (e) Omni test (f) Lamé Superspheres

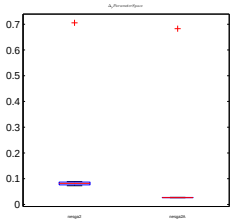
Figure 4.17: Box plots of Δ_2 in objective space for $P_{Q,\epsilon}$-NSGA2, $P_{Q,\epsilon}D_{xy}$-NSGA2, $P_{Q,\epsilon}$-MOEA and $P_{Q,\epsilon}D_{xy}$-MOEA (from left to right).

Experiment 2: validation of the use of an external archiver

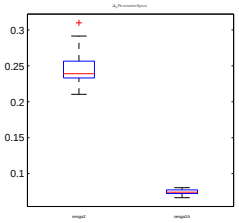
Next, we perform an experiment to validate the use of an external archiver in the novel algorithm. Figure 4.18 and Figure 4.19 show the box plots in both decision and objective space. Tables 4.7 and 4.8 show the mean and standard deviation of the $Delta_2$ values obtained by the algorithms. From the results, we can observe that in 5 of the 6 problems the use of the external archiver has a significant impact on the algorithm. This indicates an advantage of using the novel archiver. It is important to notice, that the use of the external archiver does not use any function evaluation. Thus, the comparison is fair in terms of the number of evaluations used.



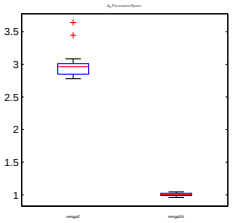
(a) Deb99



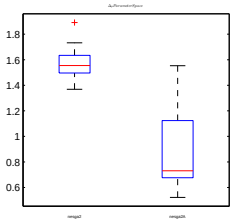
(b) Two-on-one



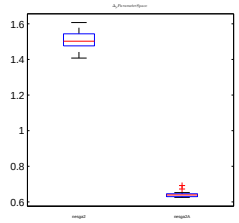
(c) Sym-part



(d) SSW

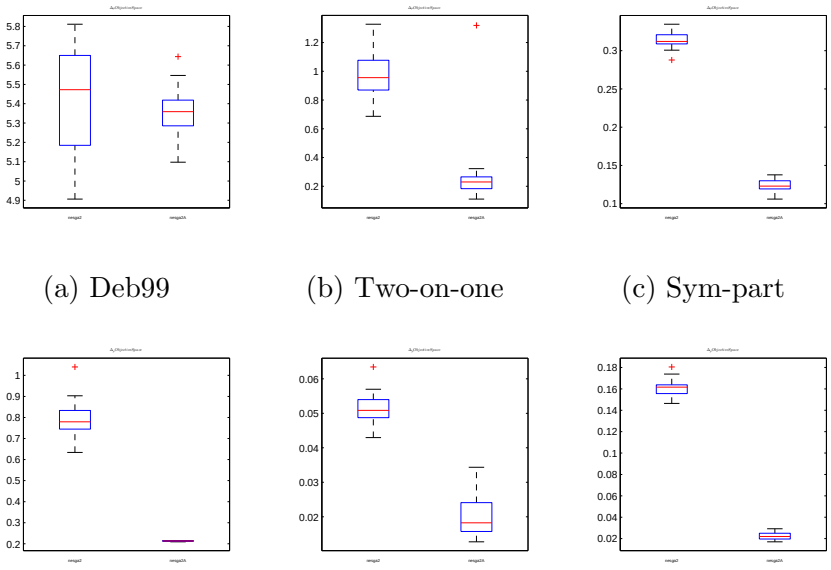


(e) Omni test



(f) Lamé Superspheres

Figure 4.18: Box plots of Δ_2 in decision space of NϵSGA without external archiver (left) and with external archiver(right).



(a) Deb99 (b) Two-on-one (c) Sym-part

(d) SSW (e) Omni test (f) Lamé Superspheres

Figure 4.19: Box plots of Δ_2 in objective space of NϵSGA without external archiver (left) and with external archiver(right).

Table 4.7: Averaged Δ_2 in decision space for the use of the external archiver and the base case.

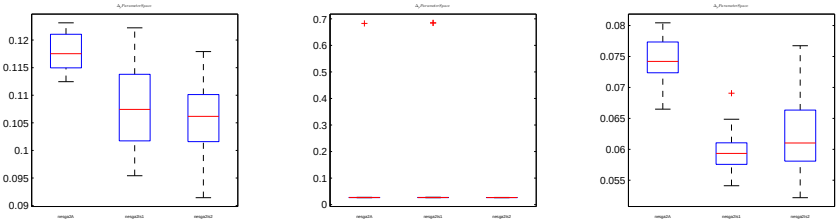
Problem	NϵSGA	NϵSGA-A
Deb99	**0.1077**(0.0100)	0.1177(0.0032)↓
Two-on-one	0.1125(0.1397)	**0.0595**(0.1467)↑
Sym-part	0.2466(0.0242)	**0.0741**(0.0039)↑
SSW	2.9840(0.2123)	**1.0050**(0.0226)↑
Omni test	1.5740(0.1163)	**0.8680**(0.2806)↑
Lamé Superspheres	1.5092(0.0532)	**0.6415**(0.0163)↑

Table 4.8: Averaged Δ_2 in objective space for the use of the external archiver and the base case.

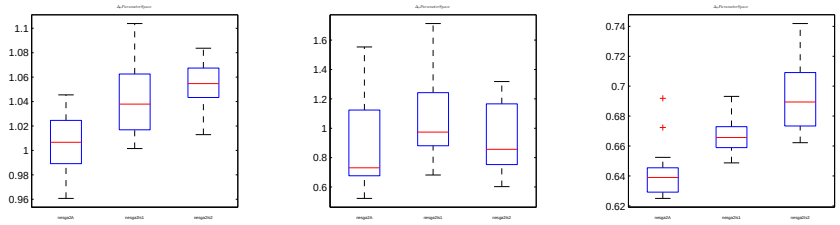
Problem	NϵSGA	NϵSGA-A
Deb99	5.4251(0.2816)	**5.3544**(0.1341)↔
Two-on-one	0.9771(0.1385)	**0.2726**(0.2522)↑
Sym-part	0.3147(0.0116)	**0.1232**(0.0087)↑
SSW	0.7916(0.0908)	**0.2128**(0.0016)↑
Omni test	0.0513(0.0047)	**0.0203**(0.0061)↑
Lamé Superspheres	0.1610(0.0079)	**0.0222**(0.0036)↑

Experiment 3: validation of the use of subpopulations

Further, we perform an experiment to validate the use of subpopulations in the novel algorithm. Figure 4.20 and Figure 4.21 show the box plots in both decision and objective space. Tables 4.9 and 4.10 show the mean and standard deviation of the $Delta_2$ values obtained by the algorithms. From the results, we can observe that the first subpopulation approach is better in 2 and 4 problems out of 6 according to decision/objective space respectively. While the second approach outperforms the base algorithm in 2 and 5 problems out of 6 according to decision/objective space. Thus, there is an advantage with the use of subpopulation in the test problems used.

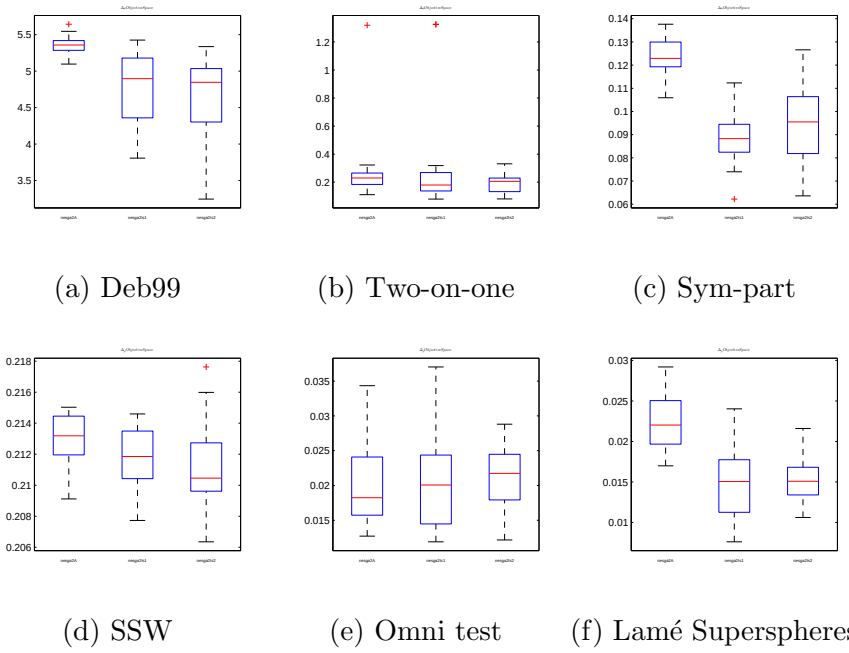


(a) Deb99 (b) Two-on-one (c) Sym-part



(d) SSW (e) Omni test (f) Lamé Superspheres

Figure 4.20: Box plots of Δ_2 in decision space of NϵSGA without external archiver (left) and with external archiver(right).



(a) Deb99 (b) Two-on-one (c) Sym-part

(d) SSW (e) Omni test (f) Lamé Superspheres

Figure 4.21: Box plots of Δ_2 in objective space of NϵSGA without external archiver (left) and with external archiver(right).

Table 4.9: Averaged Δ_2 for the algorithms that use subpopulations and the base case in decision space.

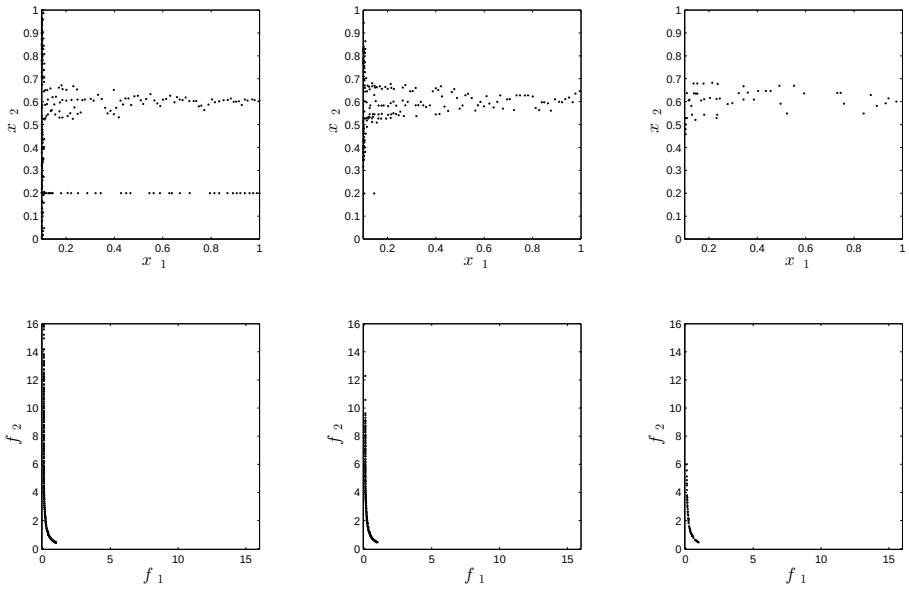
Problem	NϵSGA	NϵSGA-Is1	NϵSGA-Is2
Deb99	0.1177(0.0032)	0.1077(0.0078)↑	**0.1059**(0.0065)↑
Two-on-one	0.0595(0.1467)	0.0925(0.2025)↔	**0.0266**(0.0003)↔
Sym-part	0.0741(0.0039)	**0.0597**(0.0034)↑	0.0621(0.0061)↑
SSW	1.0050(0.0226)	**1.0402**(0.0279)↓	1.0554(0.0179)↓
Omni test	**0.8680**(0.2806)	1.0647(0.3026)↓	0.9321(0.2316)↔
Lamé Superspheres	**0.6415**(0.0163)	0.6657(0.0109)↓	0.6917(0.0219)↓

Table 4.10: Averaged Δ_2 for the algorithms that use subpopulations and the base case in objective space.

Problem	NϵSGA	NϵSGA-Is1	NϵSGA-Is2
Deb99	5.3544(0.1341)	4.7753(0.4766)↑	**4.6843**(0.5331)↑
Two-on-one	0.2726(0.2522)	0.3004(0.3565)↔	**0.1874**(0.0659)↑
Sym-part	0.1232(0.0087)	**0.0884**(0.0108)↑	0.0943(0.0170)↑
SSW	0.2128(0.0016)	0.2116(0.0020)↑	**0.2109**(0.0029)↑
Omni test	0.0203(0.0061)	**0.0198**(0.0063)↔	0.02092(0.0044)↔
Lamé Superspheres	0.0222(0.0036)	**0.0145**(0.0043)↑	0.01560(0.0029)↑

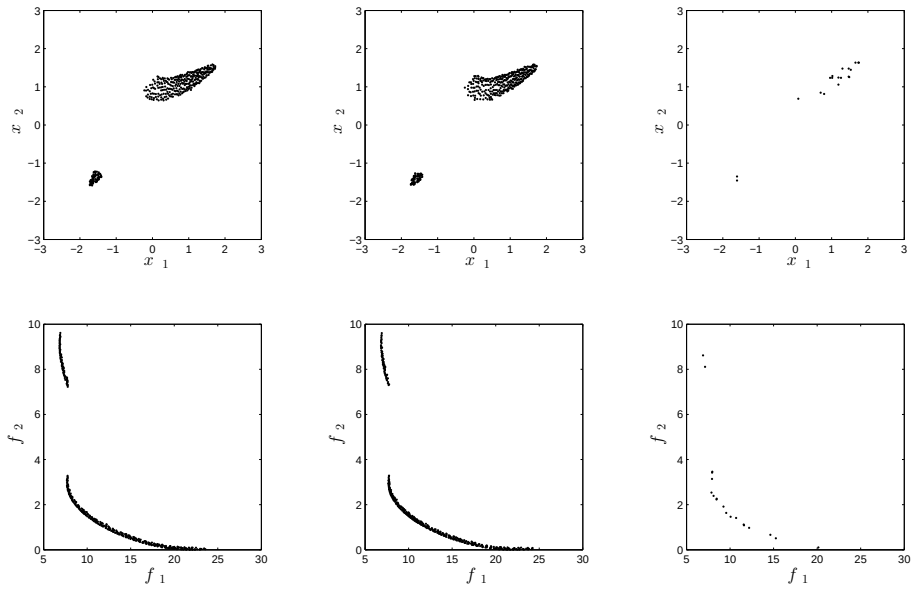
Experiment 4: comparison to state-of-the-art algorithms

Now, we compare the proposed algorithm with $P_{Q,\epsilon}D_{xy}$-NSGA2 and $P_{Q,\epsilon}D_{xy}$-MOEA. Figures 4.22-4.27 show the median approximations to $P_{Q,\epsilon}$ and $F(P_{Q,epsilon})$ obtained by the algorithms. Figure 4.28 and Figure 4.29 show the box plots in both decision and objective space. Tables 4.11 and 4.12 show the mean and standard deviation of the $Delta_2$ values obtained by the algorithms. From the results, we can observe that the novel algorithm outperforms $P_{Q,\epsilon}D_{xy}$-NSGA2 in 5 out of 6 problems according to decision space and 3 according to objective space. When compared with $P_{Q,\epsilon}D_{xy}$-MOEA, the novel algorithm outperforms in 5 problems in according both decision and objective space. This shows that an improvement in the search engine to maintain solutions well spread in decision and objective space is advantageous when approximating $P_{Q,\epsilon}$.



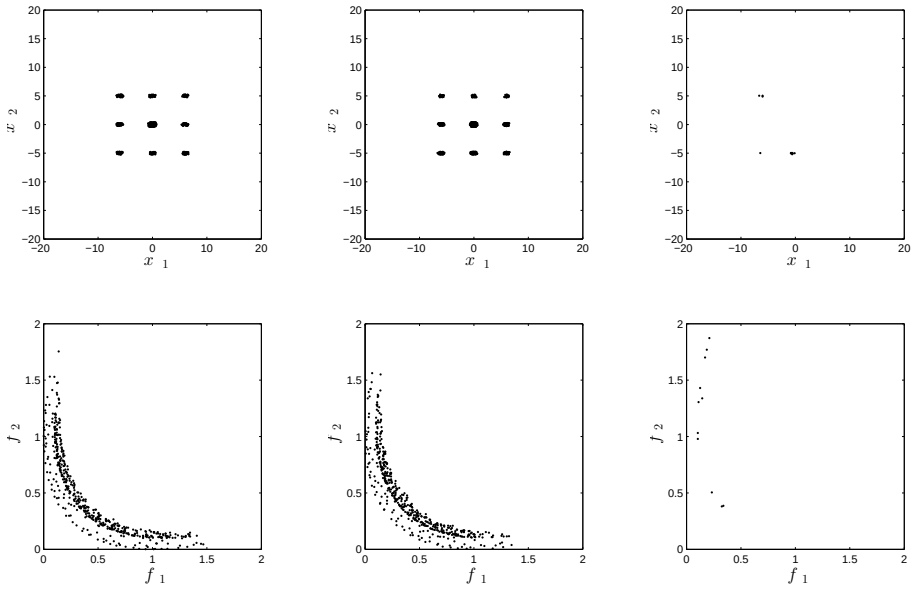
(a) $P_{Q,\epsilon}$NSGA-II (b) $P_{Q,\epsilon}D_{xy}$NSGA-II (c) $P_{Q,\epsilon}$MOEA

Figure 4.22: Numerical results on Deb99 for the comparison of NϵSGA and the state-of-the-art algorithms.



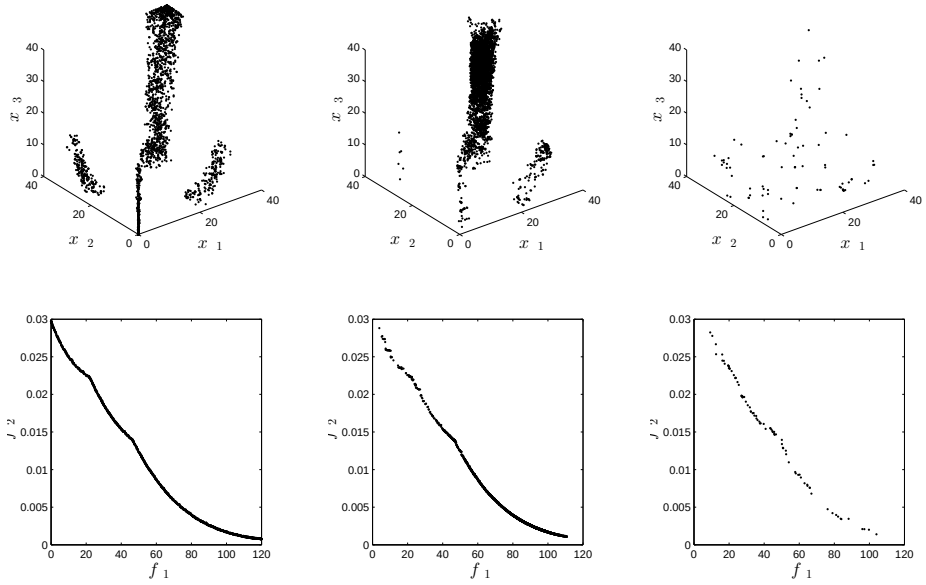
(a) $P_{Q,\epsilon}$NSGA-II (b) $P_{Q,\epsilon}D_{xy}$NSGA-II (c) $P_{Q,\epsilon}$MOEA

Figure 4.23: Numerical results on Two-on-one for the comparison of NϵSGA and the state-of-the-art algorithms.



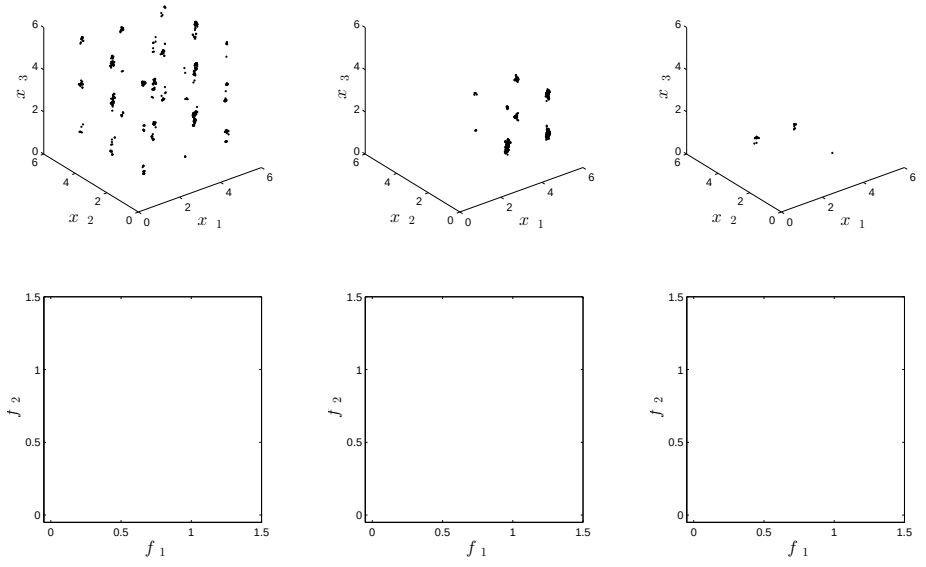
(a) $P_{Q,\epsilon}$NSGA-II (b) $P_{Q,\epsilon}D_{xy}$NSGA-II (c) $P_{Q,\epsilon}$MOEA

Figure 4.24: Numerical results on sym-part for the comparison of NϵSGA and the state-of-the-art algorithms.



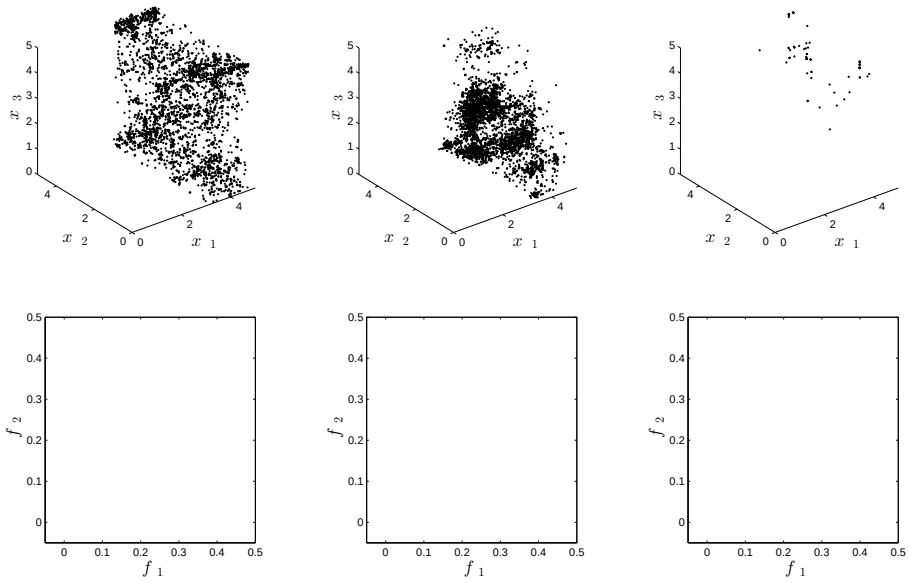
(a) $P_{Q,\epsilon}$NSGA-II (b) $P_{Q,\epsilon}D_{xy}$NSGA-II (c) $P_{Q,\epsilon}$MOEA

Figure 4.25: Numerical results on SSW for the comparison of NϵSGA and the state-of-the-art algorithms.



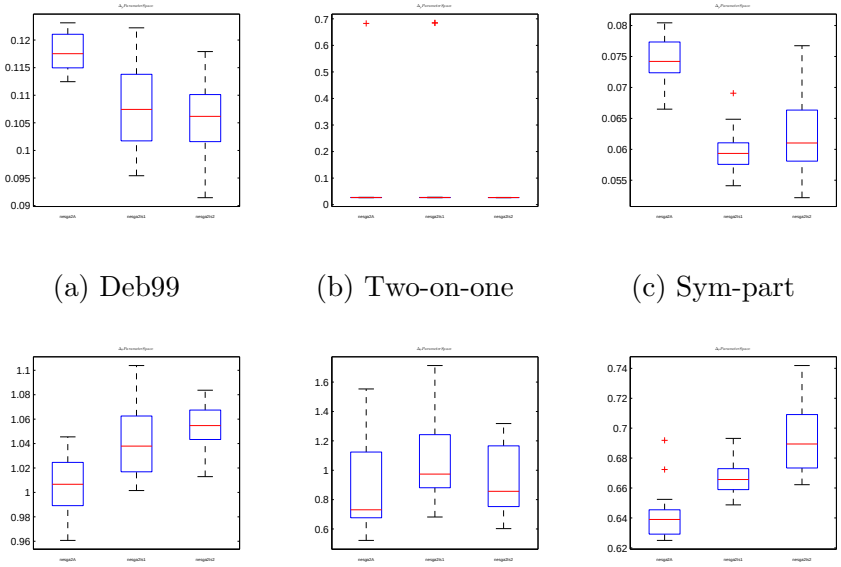
(a) $P_{Q,\epsilon}$NSGA-II (b) $P_{Q,\epsilon}D_{xy}$NSGA-II (c) $P_{Q,\epsilon}$MOEA

Figure 4.26: Numerical results on Omni test for the comparison of NϵSGA and the state-of-the-art algorithms.



(a) $P_{Q,\epsilon}$NSGA-II (b) $P_{Q,\epsilon}D_{xy}$NSGA-II (c) $P_{Q,\epsilon}$MOEA

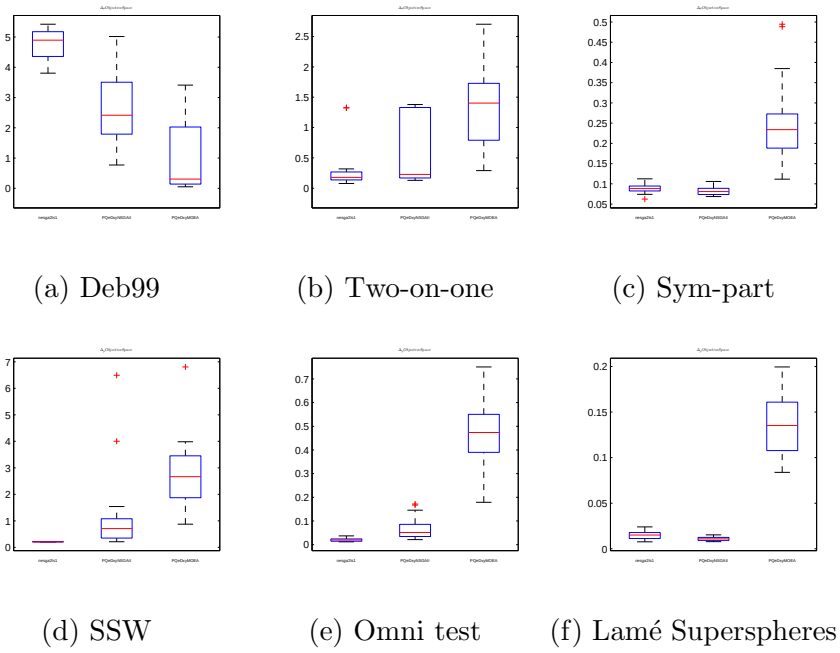
Figure 4.27: Numerical results on Lamé Superspheres for the comparison of NϵSGA and the state-of-the-art algorithms.



(a) Deb99 (b) Two-on-one (c) Sym-part

(d) SSW (e) Omni test (f) Lamé Superspheres

Figure 4.28: Box plots of Δ_2 in decision space of NϵSGA without external archiver (left) and with external archiver (right).



(a) Deb99 (b) Two-on-one (c) Sym-part

(d) SSW (e) Omni test (f) Lamé Superspheres

Figure 4.29: Box plots of Δ_2 in objective space of NϵSGA without external archiver (left) and with external archiver(right).

Table 4.11: Averaged Δ_2 in decision space for the comparison of the state-of-the-art algorithms.

Problem	NϵSGA	$P_{Q,\epsilon}D_{xy}$-NSGA-II	$P_{Q,\epsilon}D_{xy}$-MOEA/D
Deb99	0.1077(0.00787)	**0.0769**(0.01702)↑	0.0771(0.00700)↑
Two-on-one	**0.0925**(0.20251)	0.2585(0.32223)↓	0.4298(0.29135)↓
Sym-part	**0.0597**(0.00342)	0.2238(0.47463)↓	6.2978(2.07630)↓
SSW	**1.0402**(0.02791)	2.0611(1.26990)↓	4.6774(0.57909)↓
Omni test	**1.0647**(0.30264)	2.3839(0.39110)↓	3.4810(0.57159)↓
Lamé Superspheres	**0.6657**(0.01099)	0.9327(0.10483)↓	1.9653(0.17019)↓

Table 4.12: Averaged Δ_2 in objective space for the comparison of the state-of-the-art algorithms.

Problem	NϵSGA	$P_{Q,\epsilon}D_{xy}$-NSGA-II	$P_{Q,\epsilon}D_{xy}$-MOEA/D
Deb99	0.1177(0.0032)	0.1077(0.0078)↑	**0.1059**(0.0065)↑
Two-on-one	0.0595(0.1467)	0.0925(0.2025)↓	**0.0266**(0.0003)↓
Sym-part	0.0741(0.0039)	**0.0597**(0.0034)↑	0.0621(0.0061)↓
SSW	**1.0050**(0.0226)	1.0402(0.0279)↓	1.0554(0.0179)↓
Omni test	**0.8680**(0.2806)	1.0647(0.3026)↓	0.9321(0.2316)↓
Lamé Superspheres	**0.6415**(0.0163)	0.6657(0.0109)↑	0.6917(0.0219)↓

Application to Non-uniform beam

Finally, we present the results of the novel algorithm on the non-uniform beam presented before. The parameters were set as follows:

- Population size: 500
- Number of generations: 200
- The rest of the parameters were set as before

Figure 4.30 shows the approximated set $P_{Q,\epsilon}$. The results matches those solutions in Figure 4.9 while using significantly less function evaluations (10 million function evaluations for the archiver and 100,000 for the novel algorithm).

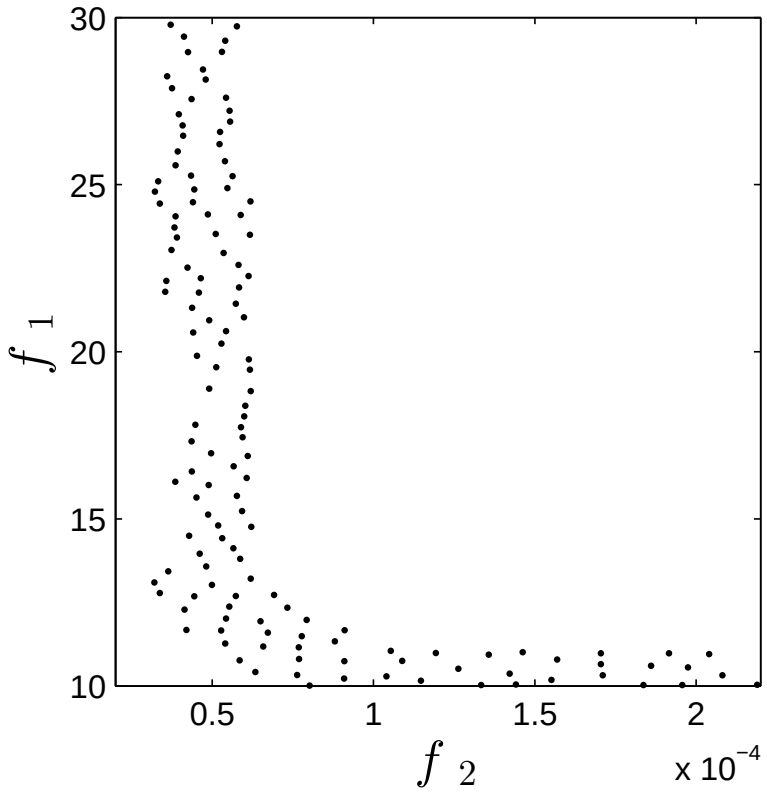


Figure 4.30: Approximation of $P_{Q,\epsilon}$ obtained with NϵSGA.

Chapter 5

Computing Lightly Robust Optimal Solutions of an MOP

The typical goal of an MOP is to identify the Pareto set/front. However, in practice, the decision maker may not always be interested in the best solutions, in particular, if these solutions are sensitive to perturbations (Beyer and Sendhoff, 2007; Jin and Branke, 2005; Deb and Gupta, 2006; Avigad and Branke, 2008). In such cases, there exists an additional challenge. One has to search not only for solutions with a good performance but also for solutions that can be implemented, leading to the so-called robust multi-objective optimization problem (RMOP) (Deb and Gupta, 2006; Ehrgott et al., 2014). In this context, the notion of robustness is not clear since it relies on the information at hand from the given problem as well as the preferences of the decision maker. Thus, there exist multiple definitions of robustness according to the different scenarios (Deb and Gupta, 2006; Kuroiwa and Lee, 2012; Doolittle et al., 2012; Fliege and Werner, 2014; Ehrgott et al., 2014).

Recently, the lightly robust multi-objective optimal solutions were proposed (Ide and Schöbel, 2016). These solutions are often good candidates for the decision maker, since solutions have to be reliable as well as to yield good performances. In this case, a solution is considered to be feasible if it is "close enough" to an optimal solution. Then, the "most reliable" solutions are chosen with respect to the set-based minmax robust efficiency Ehrgott et al. (2014). Thus, lightly robust optimal solutions yield similar performance to optimal ones while being more reliable.

In this chapter, we address the problem of computing $P_{Q,\epsilon}$ via generalized cell mapping technique and stochastic search algorithms. To be more precise, adapt the generalized cell mapping to the context of lightly robust optimal solutions. Next, we design and investigate an archiving strategy based on $ArchiveUpdateP_{Q,\epsilon}D_{xy}$ that maintains a representation of the set of worst cases for the elements in $P_{Q,\epsilon}$. Then, we integrate the archiver an the evolutionary algorithm NϵSGA and we make a comparative study on some test problems in order to visualize the effect of all strategies. Finally, we present an application to a non-uniform beam.

5.1 GCM for Lightly Robust Optimal Solutions

In this section, we present a novel algorithm for the computation of lightly robust optimal solutions for low dimensional multi-objective optimization problems. We adapt the generalized cell mapping, which was originally designed for the global analysis of dynamical systems. Further, we argue that cell mapping techniques (Hsu, 1987) are in particular advantageous for the computation of lightly robust multi-objective optimal solutions in optimal control problems. As these methods allow for the thorough investigation of small dimensional problems (Zufiria and Martínez-Marín, 2003; Gomez et al., 2008; Hernández et al., 2013; Xiong et al., 2014, 2016). The algorithm couples GCM with subdivision techniques to first compute the set of nearly optimal solutions. Then, it computes the worst case scenarios for each solution found exploiting the information about the basin of attraction already computed by the GCM. Finally, the algorithm keeps the most reliable solutions with respect to the set-based minmax robust efficient. The results show that the algorithm is able to compute a good approximation of the solution set on several low-dimensional academic test function measured with the Δ_2 indicator.

5.1.1 Proposed algorithm

In this section, we present the algorithm for the computation of lightly robust optimal solutions based on the generalized cell mapping.

General framework

In the following, we present the general procedure to compute lightly robust optimal solutions (Algorithm 14). First, the algorithm computes GCM to compute the canonical matrix (line 2 of Algorithm 14). The sub-matrix I contains the periodic cells (candidate optimal solutions of the MOP). Next, these solutions are the starting point to look for nearly optimal solutions with a backward search algorithm (line 3 of Algorithm 14). Then, the cells containing the set of nearly optimal solutions are subdivided and process is repeated for a number of iterations. After that, the algorithm computes the worst case for each cell found in the previous step by solving $\max_{\delta \in U} F(x+\delta)$ where $x \in P_{Q,\epsilon}$ (line 6 of Algorithm 14). Finally, the algorithm uses an archiver to filter the best sets worst cases (line 7 of Algorithm 14). The next sections give details on how to perform each of the steps.

Algorithm 14 GCM for Multi-objective Light Robust Optimal Solutions

Require: F: objective function, $\delta \in \mathbb{R}^n$: error, $lb \in \mathbb{R}^n$ and $ub \in \mathbb{R}^n$: lower and upper bounds respectively, $N^0 \in \mathbb{R}^n$: cells per dimension, s^0 set of cells, *iter* number of subdivision steps

Ensure: LR: Set of lightly robust solutions

1: **for** $l = 0, \ldots, iter$ **do**
2: $[P^l, \bar{s}^l] \leftarrow GCM(F, s^l, lb, ub, N^l)$
3: $P^l_{Q,\epsilon} \leftarrow BackwardSearch(P^l, \bar{s}^l, \epsilon)$
4: $[s^{l+1}, N^{l+1}] \leftarrow Subdivide(P^l_{Q,\epsilon}, l+1)$
5: **end for**
6: $WC \leftarrow ComputeWC(P^{iter}, s^{iter}, \delta)$
7: $LR \leftarrow ArchiveUpdateP_{re}(WC, [])$
8: return LR

After one run of the GCM algorithm, we have gathered the information on the global dynamics of the system and are hence able to approximate the set of interest in a post-processing step. For the problem at hand, the approximation of $P_{Q,\epsilon}$, we use the archiving technique $ArchiveUpdateP_{Q,\epsilon}$ (Schütze et al., 2008, 2010).

The integration of both algorithms is as follows: Algorithm 15 updates the archive first with the periodic cells discovered with GCM and continues with the rest of the periodic motion by inverting the cell mappings. First, a queue is generated with the periodic cells and until the queue is empty the algorithm searches for nearly optimal solutions. The algorithm takes advantage of the fact that GCM has already encoded the mappings in the canonical matrix. Thus, it is possible to exploit that information to perform a breadth-first search where new cells are enqueued if they are accepted by the archiver. Note that if a cell s is not accepted by the archiver neither would be the cells that map to s, since by construction these cells are dominated by s. Thus, Algorithm 15 computes the set of nearly optimal solutions without testing all cells in the search space.

Algorithm 15 Computation of $P_{Q,\epsilon}$ with backward search

Require: P: canonical form of probability matrix, s: set of cells

Ensure: $P_{Q,\epsilon}$ approximation

1: $A \leftarrow ArchiveUpdateP_{Q,\epsilon}(I, [], \epsilon)$
2: Create a queue Q using A
3: **while** $Q \neq \emptyset$ **do**
4: $cell \leftarrow Q.dequeue()$
5: $c \leftarrow P^T_{cell}$
6: $A \leftarrow ArchiveUpdateP_{Q,\epsilon}(c, A, \epsilon)$
7: $Q.enqueue(c \cap A)$
8: **end while**
9: return A

Compute worst case

Once we have a suitable representation of the set of nearly optimal solutions, we can search for the worst cases for each solution in $P_{Q,\epsilon}$. As before, the information provided from GCM allows us to compute the set of worst cases with a post-processing of the data. For a given cell s, the algorithm finds the $2\lceil\frac{\delta_i}{h_i}\rceil$ neighbors for $i = 1, \ldots, n$, where h is the size of the cell and δ is the uncertainty. Then, the algorithm computes the set of worst cases. Note that this can be done by transposing the matrix P and then looking for those cells that do not have any image in $\bar{Q} = \{\bar{x}|x_i - \delta_i \leq \bar{x} \leq x_i + \delta_i\}$. Algorithm 16 shows the procedure to compute the worst cases.

Algorithm 16 Computation of worst cases

Require: $P_{Q,\epsilon}$: $P_{Q,\epsilon}$ approximation, p: probability matrix
Ensure: set of worst cases
1: **for all** $cell \in P_{Q,\epsilon}$ **do**
2: Select neighbors of cell
3: Compute max of the selected elements
4: **end for**
5: return WC

Compute best worst cases

Finally, it is required to filter the solutions to keep the best worst cases. Algorithm 17 extends the archiver $ArchiveUpdateP_Q$ to handle families of solution sets. In this case, both P and A_0 are families of sets. Note that in line 3 of the algorithm uses set-based dominance instead of classical Pareto dominance.

Algorithm 17 $A := ArchiveUpdateP_{re}\ (P, A_0)$

Require: population P, archive A_0
Ensure: updated archive A
1: $A := A_0$
2: **for all** $p \in P$ **do**
3: **if** $\nexists a \in A : \ a \prec p$ **then**
4: $A := A \cup \{p\}$
5: **end if**
6: **for all** $a \in A$ **do**
7: **if** $p \prec_{-\epsilon} a$ **then**
8: $A := A \backslash \{a\}$
9: **end if**
10: **end for**
11: **end for**
12: return A

Computational complexity

In this section, we discuss the computational time complexity of each of the algorithms presented with respect to the number of cells to process.

- GCM: all cells are visited once $O(Nc)$ and for each cell the algorithm computes its neighbors. The neighbors depend on the type of vicinity that one uses. It could be n if one selects orthogonal neighbors or $3^n - 1$ with the full neighborhood. Note that the number of neighbors is in general much lower that the number of cells. Thus, the complexity of GCM is $O(Nc)$.

- BackwardSearch: in the worst case all the cells have to be visited (all cells are nearly optimal solutions). Since a breadth first search is used, the cells are visited only once. Next, the complexity of $ArchiveUpdateP_{Q,\epsilon}$ is $O(Nc)$ since in the worst case all candidate solutions are compared with the solutions in the archiver. Thus, the complexity of BackwardSearch is $O(Nc^2)$.

- Computation of worst cases: in this case, the algorithm has to analyze at most Nc cell to find their worst cases. The size of each grid is of size $2\lceil\frac{\delta_i}{h_i}\rceil) + 1$ since their size is given by the number of neighbors. Note the as in GCM it takes linear time to find the worst cases. Thus, the complexity of this algorithm is $O(max(\lceil\frac{\delta_i}{h_i}\rceil)Nc)$ for $i = 1, \ldots, n$.

- $ArchiveUpdateP_{re}$: in the worst case each candidate solution will be formed by $2\lceil\frac{\delta_i}{h_i}\rceil) + 1$ solutions. From this follows, that each dominance comparison have a complexity of $O(max(\lceil\frac{\delta_i}{h_i}\rceil)$. Thus, the complexity of the archiver is $O(max(\lceil\frac{\delta_i}{h_i}\rceil)Nc^2)$ for $i = 1, \ldots, n$.

From the above discussion it follows that the total time complexity of the algorithm to compute lr solutions is $O(max(\lceil\frac{\delta_i}{h_i}\rceil)Nc^2)$. It is also important to notice that the total number of cells is given by $\prod_{i=1}^{n} N$. If one would like to maintain the precision, the number of cells required will increase exponentially with the number of dimensions. Thus, the complexity with respect to the number of dimensions is $O(\max(N)^n)$. Which is the main reason GCM is restricted to low dimensional problems.

5.1.2 Numerical results

In the following, we present the experimental design used to validate the performance of the novel algorithm. The algorithms were implemented in Matlab®R2012a and C (connected through mex files). All the executions were performed in a desktop computer with Intel®Core^{TM} i7-2600K CPU 3.40GHz processor and 4 GB of RAM. Here, we use four academical test problems. These problems were proposed for multi-objective optimization without uncertainty. However, the problems present interesting features for light robust optimization (see Appendix A) for the description of the problems):

Figure 5.1 shows the results of the novel algorithm on Deb99 in each step of the algorithm. First, Figure 5.1a shows the GCM, there we can see that the Pareto set is located in $x_2 = 0.2$. Next, Figure 5.1b shows the results after the application of the BackwardSearch algorithm. In this case, there are two regions of interest located around $x_2 = 0.2$ and $x_2 = 0.6$. These regions corresponds to the local Pareto sets. Then, once the set $P_{Q,\epsilon}$ was computed, the next step is to compute the worst cases for each $x \in PQ, \epsilon$. Figure 5.1c shows the computation of worst cases. Here, we can observe a curve connecting $[1, 2]$ and $[0.110]$ that did not appear in the set of nearly optimal solutions. This curve corresponds to the worst case of the global Pareto front. Finally, Figure 5.1d shows the lightly robust optimal solutions after the application of the $Archiver P_{re}$. There we can see the global Pareto front is dominated (in the re sense) by the local front. Thus, the lightly robust solutions are those solutions in the local front located in $x_2 = 0.6$.

Now, we show the results for all the problems. Table 5.1 shows the parameters used to perform the experiments. In this case ϵ denotes the allowed deterioration from the Pareto optimal solutions, δ represents the uncertainty and N is the number of cells used per dimension. In our experimental study, we use the Δ_2 indicator to measure the distance of the best solutions found by the algorithm to the real solution in decision space.

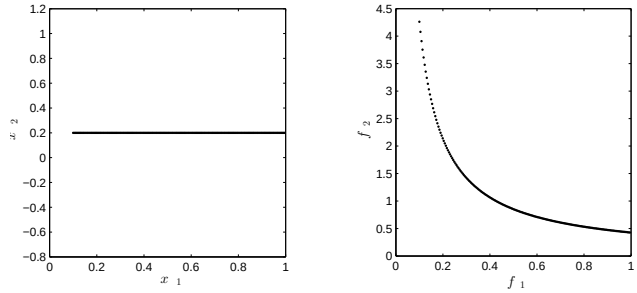
Table 5.1: Parameters used for each problem.

Problem	ϵ	δ	N
Deb99	$[0.0110\,0.0110]^T$	$[0.0068\,0.0075]^T$	$[200\,200]^T$
Two-on-one	$[0.1000\,0.1000]^T$	$[0.0450\,0.0450]^T$	$[200\,200]^T$
Sym-part	$[0.1500\,0.1500]^T$	$[0.3000\,0.3000]^T$	$[200\,200]^T$
SSW	$[0.0100\,0.0001]^T$	$[3.00\,3.00\,3.00]^T$	$[20\,20\,20]^T$

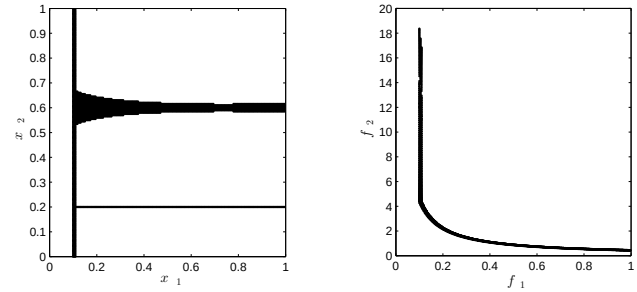
Figure 5.2 shows the GCM approximation of the set of lightly robust optimal solutions as well as their respective worst-case images. Table 5.2 shows the Δ_2 values in decision space between the real solution and the approximation set found. From the results we can observe that in Deb99 the solutions in the nominal global front are dominated by those in the local front in terms of lre. In the cases of two-to-one and sym-part the nominal global front is the lightly robust front. Finally, in SSW one of the connected components that was optimal for the nominal MOP is now dominated in terms of lre.

Table 5.2: Δ_2 values and running times (in seconds) of GCM for each problem.

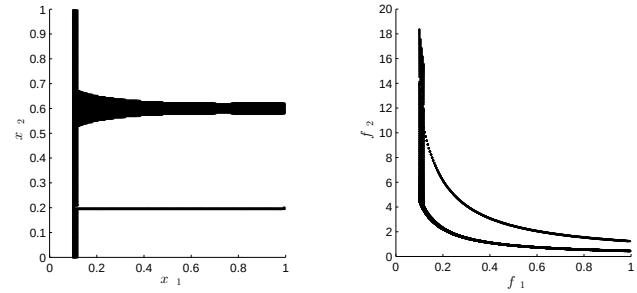
Problem	Δ_2	Time
Deb99	0.0015	131.73
Two-on-one	0.0124	127.73
Sym-part	0.0739	130.45
SSW	8.2199	287.32



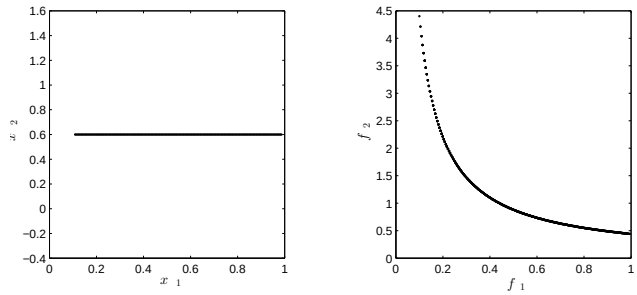
(a) Pareto optimal solutions of Deb99 found by GCM.



(b) Nearly optimal solutions of Deb99 after the backward search.



(c) Worst cases of $P_{Q,\epsilon}$ of Deb99.



(d) Lightly robust solutions of Deb99 computed by GCM.

Figure 5.1: Numerical results on Deb99.

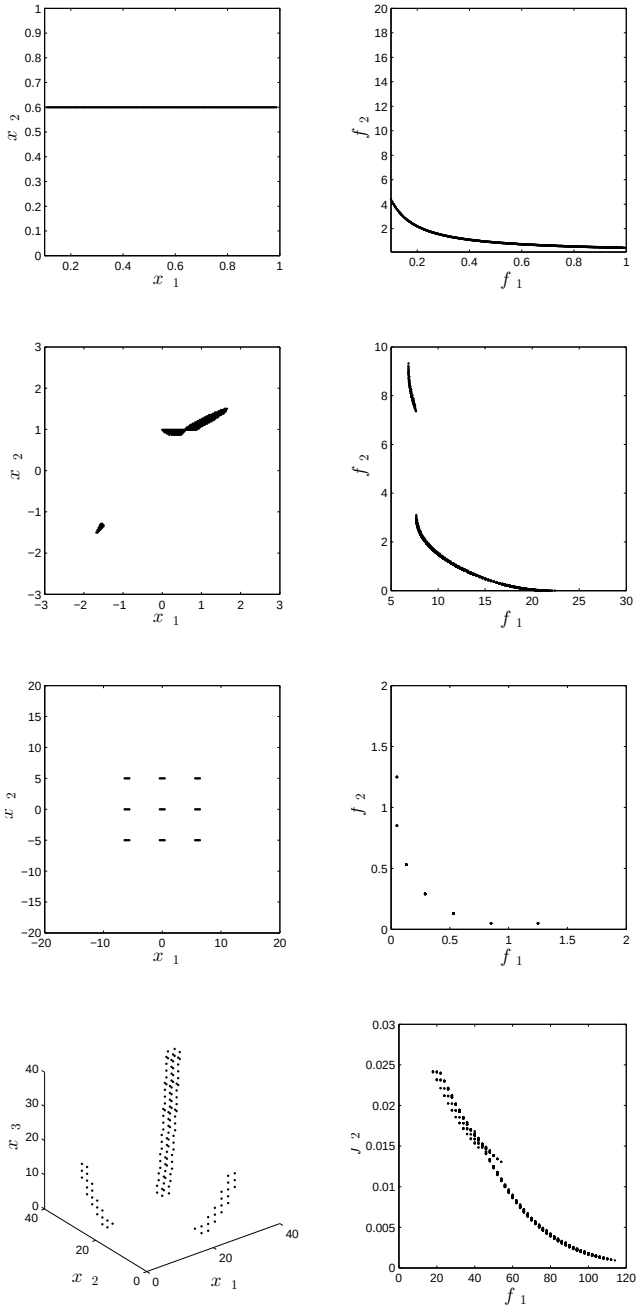


Figure 5.2: Numerical results of GCM on the academical problems. Decision space (left) and objective space (right) obtained on the problems Deb99, two-on-one, sympart and SSW from above to below.

5.1.3 Application to optimal control

Next, we consider a second order oscillator subject to a proportional-integral-derivative (PID) control B.1.4. Previously, this problem was only studied without uncertainty. In this case, the error was considered to be $\delta = [0.4, 0.29, 0.01]^T$ which corresponds to a 1% error and $\epsilon = [0.10.10.1]^T$.

GCM was executed with an initial grid of $N = [30, 18, 8]^T$ and 3 subdivision steps. Figure 5.3 shows the Pareto optimal solutions. Figure 5.4 shows the nearly optimal solutions. Figure 5.5 shows the approximation of the lightly robust optimal solutions and their worst-case image found by GCM as well as the Pareto optimal solutions of the nominal problem. Note that optimal solutions and lightly optimal solutions have the same structure. However, there is a $\Delta_2(P_Q, P_{LR}) = 1.0525$. The running time was 709.13s.

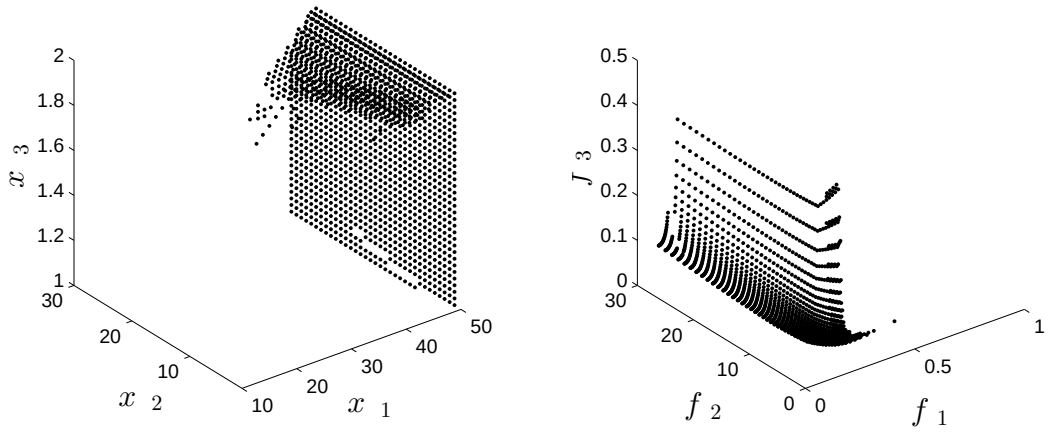


Figure 5.3: Approximation of the optimal solutions and their worst-case image found by GCM.

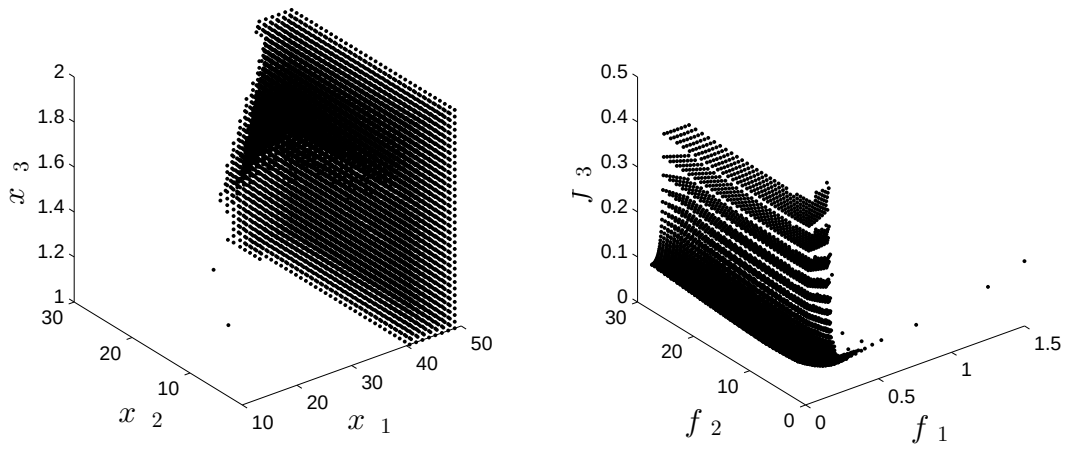


Figure 5.4: Approximation of the nearly optimal solutions and their worst-case image found by GCM.

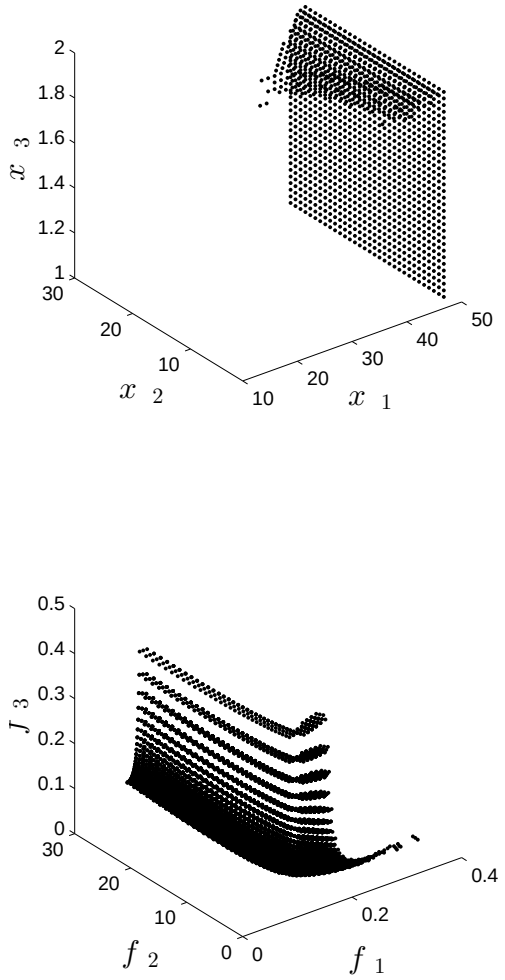


Figure 5.5: Approximation of the lightly robust optimal solutions and their worst-case image found by GCM.

5.2 A MOEA for Lightly Robust Optimal Solutions

In this section, we present an evolutionary algorithm that aims to approximate the set of lightly robust optimal solutions. An evolutionary algorithm is of high interest since, when dealing with problems with a higher number of dimensions ($n \geq 5$) the cell mapping algorithm are not able to work due to its computation complexity. First, we present an external archiver that aim to maintain a representation of the worst cases for the solutions in $P_{Q,\epsilon}$. Then, we present the integration of the archiver to a MOEA an the computation of the lightly robust optimal solutions as a post-processing step.

Next, we show the numerical results on several academical problems and compare the results with those of GCM.

5.2.1 Archiver for lightly robust optimal solutions

Now, we present the adaption of $ArchiverUpdateP_{Q,\epsilon}D_x$ to the computation of the worst cases for each solution in $P_{Q,\epsilon}$. As it was mentioned before, in the set-based case for each solution x the worst case $\max_{\delta \in U} F(x+\delta)$ is a set of points (see Figure 2.10). Thus, it is necessary for each solution in the archiver a to keep a representation of the set of worst cases. Algorithm 18 computes the set of nearly optimal solutions $P_{Q,\epsilon}$ with a good distribution in both decision an objective space (lines 2-11). When a solution a is added to the archiver, it also initializes the set of worst cases $a.max$ with $-F(a)$. Further, if a candidate solution p is close to a solution (line 13), then p will be used to update the set of worst cases of a ($a.max$). This process is done by using the archiver $ArchiveUpdateP_{Q,\epsilon}$ with $\epsilon = 0$. In this case, the parameter Δ_x is used as the uncertainty (2δ). Note that the archiver in this case is used for maximization since we are interested in the worst cases (e.g. the evaluation of p, $F(p)$ is multiplied by -1).

Algorithm 18 $A := ArchiveUpdateP_{Q,wc}\,(P, A_0, \epsilon, \Delta_x)$

Require: population P, archive A_0, $\epsilon \in \mathbb{R}^n_+$, $\Delta_x \in \mathbb{R}_+$, $\Delta_y \in \mathbb{R}_+$, $\Delta^*_x \in (0, \Delta_x)$
Ensure: updated archive A

1: $A := A_0$
2: **for all** $p \in P$ **do**
3: **if** $\nexists a_1 \in A : a_1 \prec_{-\epsilon} p$ and $\nexists a_2 : d_\infty(a_2, p) \leq \Delta^*_x)$ **then**
4: $a.max \leftarrow -F(a)$
5: $A \leftarrow A \cup \{p\}$
6: $\hat{A} = \{a_1 \in A | \nexists a_2 \in A : a_2 \prec_{-(\epsilon+\Delta_y)} a_1\}$
7: **for all** $a \in A \backslash \hat{A}$ **do**
8: **if** $p \prec_{-(\epsilon+\Delta_y)} a$ and $dist(a, \hat{A}) \geq 2\Delta_x$ **then**
9: $A \leftarrow A \backslash \{a\}$
10: **end if**
11: **end for**
12: **else if** $\exists a_1 \in A : d_\infty(a_1, p) \leq \Delta^*_x)$ **then**
13: $a.worst \leftarrow ArchiveUpdateP_{Q,\epsilon}(a.worst, p, 0)$
14: **end if**
15: **end for**

5.2.2 Proposed evolutionary algorithm for lightly robust optimal solutions

Next, we present the integration of the archiver with NϵSGA. We have chosen this algorithm since it has shown the best capabilities to maintain a representation of $P_{Q,\epsilon}$ which is the feasible set to search for lightly robust optimal solutions. The algorithm works as follows: first, NϵSGA computes an approximation of $P_{Q,\epsilon}$ and with the use of the archiver $ArchiveUpdateP_{Q,wc}$ the algorithm is also capable to compute the set of worst cases in one execution of the algorithm. Then, as a post-processing step the algorithm filters the worst case using Algorithm 17. Algorithm 19 shows main steps of the method.

Algorithm 19 LiRo-EMOA

Require: number of generations: G, population size: $npop$, $\epsilon \in \mathbb{R}^n_+$, $\Delta_x \in \mathbb{R}_+$, $\Delta_y \in \mathbb{R}_+, \Delta^*_x \in (0, \Delta_x)$
Ensure: updated archive LR with the set of lightly robust optimal solutions

```
Generate initial population P_1
A ← ArchiveUpdateP_{Q,wc}(P_0, [], ϵ, Δ_x)
for i = 1 ... ngen do
    Select ⌈λ/2⌉ Parents Q_i via Tournament
    Õ_i ← SBXCrossover(Q_i)
    O_i ← PolynomialMutation(Õ_i)
    P̃_i ← P_i ∪ O_i
    F ← Non ϵ-DominatedSorting(P̃_i)
    P_{i+1} = ∅
    j = 1
    while |P_{i+1}| + |F_j| ≤ npop do
        NearestNeighborDistance(F_j)        ▷ in both parameter/objective space
        P_{i+1} = P_{i+1} ∪ F_j
        j ← j + 1
    end while
    Sort(F_i, ≺_n)
    P_{i+1} = P_{i+1} ∪ F_j[1 : npop − |P_{i+1}|]
    A_{i+1} ← ArchiveUpdateP_{Q,wc}(O_i, A_i, ϵ, Δ_x)
end for
LR ← ArchiveUpdateP_re(A, [])
```

Computational complexity of LiRo-MOEA

In this section, we discuss the computational time complexity of each of the algorithms presented. Let G be the number of generation of the algorithm, $|P|$ the size of the population and $|A|$ the size of the archiver.

- $ArchiveUpdateP_{Q,wc}$: by construction a candidate solution p can only belong to the neighborhood of a solution $a \in A$. Thus, the complexity remains the same as $ArchiveUpdateP_{Q,\epsilon}D_x$, $O(|P||A|)$.

- MOEA: as it was stated before the complexity of NϵSGA is $O(max(G|P|^3, (G|P|)^2))$.

- $ArchiveUpdateP_{re}$: in the worst case all solutions generated during the search have to be compared $(G|P|)$. Thus, the complexity of the comparison is $O((G|P|)^2)$. Note that in general the complexity of the archiver is $O(|W||P||A|)$ where $W := \max_{a \in A} |a.max|$.

From the above discussion it follows that the total time complexity of LiRo-MOEA is $O(max(G|P|^3, (G|P|)^2))$.

5.2.3 Numerical results

In this section, we present the numerical result corresponding the evolutionary algorithm. We performed several experiments in order to validate the novel algorithm. First, we investigate the effect of the novel $ArchiverUpdateP_{Q,lre}D_{xy}$. Then, we validate the use of an external archiver in the evolutionary algorithm. Finally, we compare the resulting algorithm with the GCM.

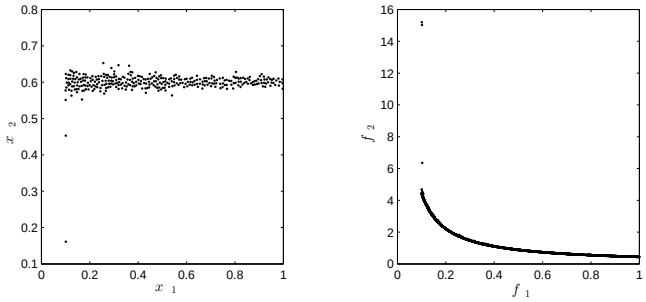
In all cases we used the following problems: Deb99, two-on-one, sym-part, ssw, omni test, Lamé superspheres. Table 5.3 shows the ϵ, Δ_y and Δ_x values used for each problem. Finally, for all cases were measured with the Δ_2 indicator.

Table 5.3: Parameters used for all experiments.

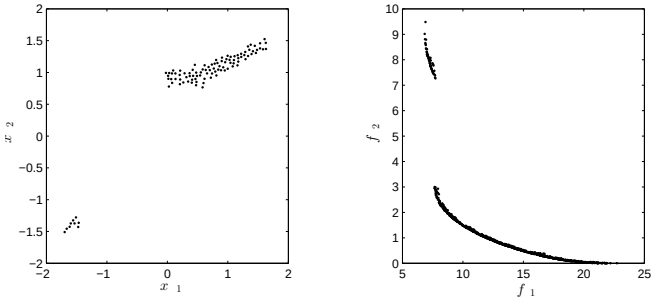
Problem	ϵ	Δ_y	Δ_x
Deb99	[0.011 0.011]	[0.2 0.2]	[0.2 0.2]
Two-on-one	[0.1, 0.1]	[0.1 0.1]	[0.02 0.2]
Sym-part	[0.15, 0.15]	[0.2 0.1]	[1 1]
SSW n=3	[0.011, 0.0001]	$\frac{1}{3}\epsilon$	[1 1 1]
Omni-test	[0.01 0.01]	[0.05 0.05]	[0.1 0.1 0.1 0.1]
LSS	[0.1 0.1]	[0.05 0.05]	[0.1 0.1 0.1 0.1 0.1]

Experiment 1: stand-alone archiver

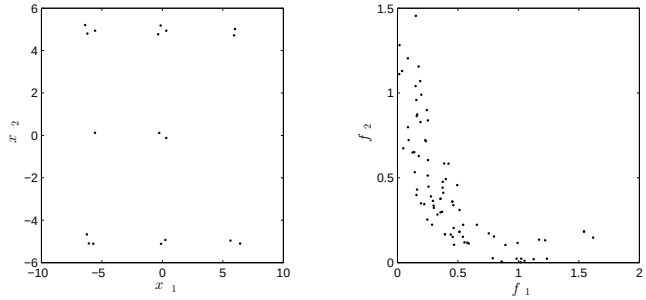
First, we investigate the effect of all the archivers where we use uniform sampling of the candidate solutions within the domain Q. Here, we study the MOPs Two-on one, Sym-part (Rudolph et al. (2007)), and SSW (Schaeffler et al. (2002)), generating 20 files with $100,000$ uniformly sampled candidate solutions. Figure 5.6 show the median results of each archive on the benchmark problems with respect to Δ_2 in decision space.



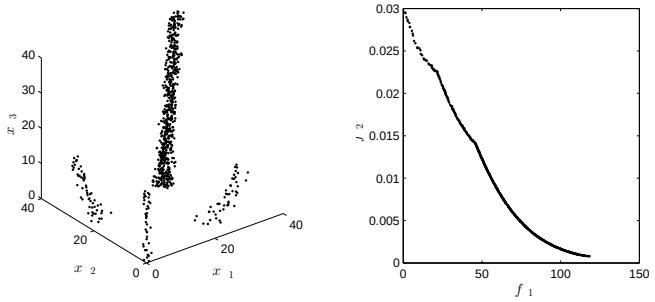
(a) Deb99



(b) Two-on-one



(c) Sym-part



(d) SSW

Figure 5.6: Set lightly robust optimal solutions found by the archiver.

Experiment 2: MOEA for lightly robust optimal solutions

Now, we compare the proposed algorithm with the GCM results obtained previously. To this end, 20 independent experiments were performed with the following parameters: 200 individuals, 200 generations, crossover rate = 0.9, mutation rate = $1/n$, $eta\_c = 20$ and $eta\_m = 20$. Figures 5.7 show the median approximations to the lightly robust optimal solutions obtained by LiRo-MOEA. Table 5.4 shows the mean and standard deviation of the Δ_2 values obtained by the algorithms in decision space. From the results, we can observe that GCM outperfoms the evolutionary algorithm in all cases. This is to be expected since GCM has a global view of the problem. However, the results of LiRo-MOEA are promising since the algorithm can be scaled to medium size problems ($10 \leq n \leq 20$).

Table 5.4: Δ_2 values for Liro-MOEA and GCM in decision space.

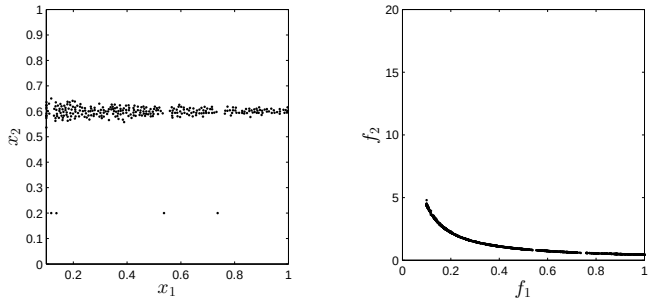
Problem	GCM	LiRo-MOEA
Deb99	**0.0015**	0.0558(0.0288)
Two-on-one	**0.0124**	0.0408(0.0038)
Sym-part	**0.0739**	1.4070(0.9794)
SSW	**8.2199**	12.470(2.1213)

Application to Non-uniform beam

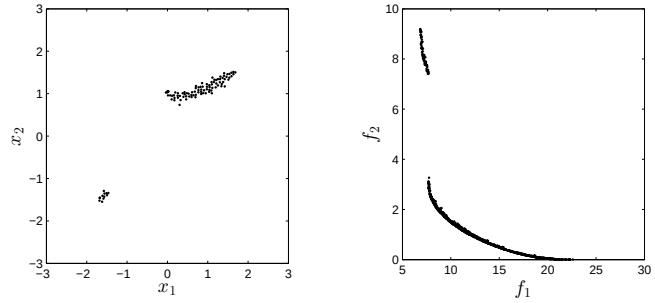
Now, we present the results of the novel algorithm on the non-uniform beam presented before. The parameters were set as follows:

- Population size: 500
- Number of generations: 200
- The rest of the parameters were set as before

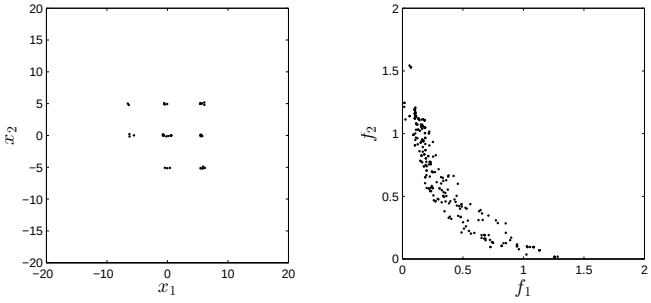
Figure 5.8 shows the approximated set $P_{Q,\epsilon}$. From the results, we can observe that the algorithm is capable to detect 52 solutions from the set of approximate solutions found. In this case, we can observe that some of these solutions belong to the nominal Pareto front, while others are dominated solutions but they are non-dominaeted in the re sense.



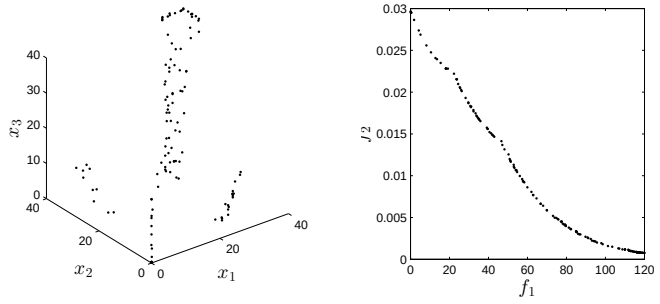
(a) Deb99



(b) Two-on-one



(c) Sym-part



(d) SSW

Figure 5.7: Set lightly robust optimal solutions found with LiRo-MOEA.

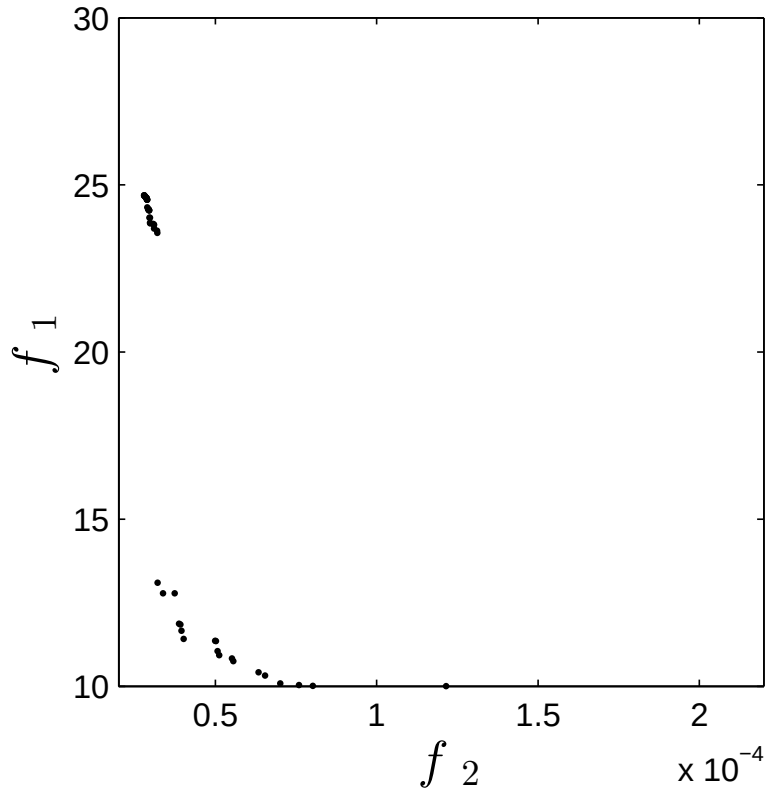


Figure 5.8: Approximation of the set of lightly optimal solutions obtained with LiRo-MOEA.

Chapter 6

Conclusions and Future Work

In this work, we have studied in depth different set oriented methods for multi-objective optimization problems. We have analyzed the advantages that knowledge acquired during the search can give to compute different sets of interest, namely the Pareto set/front, set of approximate solutions and the set of lightly robust optimal solutions.

This is an important issue since most of the current research in multi-objective optimization focuses on the computation of the Pareto set/front. However, the decision maker could be interested in different solutions. In particular this is the case, if the solutions are sensitive to perturbations. The computation of the set of approximate solutions offers the decision maker solutions that have a small deterioration with respect to the Pareto optimal ones. Thus, this kind of solutions can be of interest as backup solutions. Further, the set of lightly robust optimal solutions offer solutions that are also close to the optimal ones and that they are possible to implement.

The material presented in this thesis contributes to increase the knowledge about techniques and to motivate their use to both analyze and solve a given problem. Also, we have shown the potential of the use of knowledge acquired during the search to look for different sets of solutions. In the following, we state our conclusions as well as future paths for research.

6.1 Conclusions

First we have investigated cell mapping techniques for the numerical treatment of multi-objective optimization problems. Cell mapping techniques have been designed for the global analysis of dynamical systems that replace the common point-to-point by a cell-to-cell mapping via a discretization of both space and time. We have adapted the cell mapping techniques to the given context via considering dynamical systems derived from descent methods and have argued that the resulting algorithm is in particular beneficial for the thorough investigation of small problems. That is, the new algorithms are capable of detecting the global Pareto set in one run of the algorithm which is of course important for the related decision making process. For the latter,

however, also other points are of potential interest such as locally optimal solutions and approximate solutions which can serve as backup solutions for the DM in case he/she is willing to accept a certain deterioration measured in objective space. The cell mapping techniques are capable of delivering all the sets after the same run of the algorithm in the same approximation quality as the computed Pareto set. While satisfactory algorithms for the computation of the Pareto set exist, such as specialized evolutionary algorithms, this does not hold for the local and approximate solutions. The cell mapping techniques presented in this work offer hence a surplus in the design of small dimensional problems.

Next, we applied the novel and efficient cell mapping method for multi-objective optimal design of full state feedback and PID controls. The time-domain specifications of the step response are used as the objective function. A constraint on the closed-loop eigenvalue of the linearized system about the steady-state equilibrium solution is also imposed that provides the stability robustness of the optimized PID controls. In the first run of the hybrid algorithm, the directed search for the SCM method delivers a set of cells that covers the Pareto set. In the second run, a gradient search is applied to the covering set of cells and delivers much fine resolution of the global optimal solutions of Pareto set and Pareto front. We have found that the hybrid algorithm for the SCM method delivers substantial computational savings while obtaining comparably accurate solutions for the Pareto set and Pareto front. Simulations of the multi-objective optimal full state feedback controls for an inverted pendulum and a flexible rotary arm are presented to validate the SCM design method. The controls are optimally designed for 4 and 5 design parameters respectively, which represents a numerical challenge in MOP studies. The simulations show that the controls in the Pareto set are closely clustered indicating the tight range of performances in tuning the control gains when multi-objectives are considered in the design. What is more, the hybrid searching algorithm provides an effective tool for quantitative design of controls for non-linear dynamic systems.

Then, we have addressed the problem of computing suitable representations of the entire set of approximate solutions of a given multi-objective optimization problem with stochastic search algorithms. Since the computation of the entire set may be too costly in many applications, we have proposed an archiving strategy that aims to capture a good representation of $P_{Q,\epsilon}$ in both decision and objective space and we have studied its limit behavior as well as the size bounds. We have investigated the novel archiver empirically in order to illustrate the behavior and applicability of the archiver on some benchmark problems as well as for a non-uniform beam problem. The numerical results indicate that the archiver $ArchiveP_{Q,\epsilon}D_{xy}$ that aims for a discretization of both decision and objective space might be of most practical use for the use within stochastic search algorithms though the adjustment of the right discretization parameter value in decision space might be a delicate issue. Finally, we used the archivers within a stochastic search algorithms. In this case the interplay of generator and archiver was a delicate problem including a proper balance of exploration and exploration, a suitable density distribution for all generational operators.

The results show that the novel algorithm is competitive against other methods of the state of the art on academic problems.

After that, we have proposed a novel multi-objective evolutionary algorithm which aims to compute the set of approximate solutions. The algorithm uses the archiver $ArchiveP_{Q,\epsilon}D_{xy}$ in order to maintain a well-distributed representation in both decision and objective space. The results show that the novel algorithm is competitive against other methods of the state of the art on academic problems as well as the non-uniform beam problem.

Further, we have investigated cell mapping techniques for the numerical treatment of uncertain multi-objective optimization problems in terms of lre. The new algorithm is capable of detecting solutions that have almost the same performance than the optimal solutions but that are more reliable. This gives the decision maker solutions that are less susceptible to uncertainties and thus are good candidates to implement. The main advantage of the novel algorithm is that lr solutions can be computed with the same effort as computing optimal solutions in terms of function evaluations. It is important to notice that this algorithm is capable to deliver approximations to the Pareto set/front, set of approximate solutions and the set of lightly optimal solutions in one execution of the algorithm.

Finally, we have proposed a novel multi-objective evolutionary algorithm which aims to compute the set of lightly optimal solutions. In this case, we have found that the external archiver integrated with the evolutionary algorithm is capable to approximate the set of interest on several academical problems. This shows that it is possible to extract information from past solutions (individuals) in order to improve the quality of the result.

6.2 Future Work

For future work, there are many interesting open questions to address. Though the results presented in this work regarding the cell mapping techniques are very promising, there are some points that have to be addressed in order to make the algorithm applicable to a broader class of problems. The main drawback of the cell mapping techniques is that they are restricted to small dimensional problems. Thus, it is of high interest to parallelize the algorithms since the core of the algorithm is the mapping of each cell which can be realized with small effort. We expect thus that the use of massive parallelism realized e.g. via GPUs will lead to an applicability to higher dimensional problems.

Next, it is of high interest to design bounded versions of the archiver techniques. This can improve their applicability while making them easier to integrate with evolutionary algorithms. Since, it would not be necessary to give explicitly the values for Δ_x and Δ_y. Then, the knowledge acquired during the search could be further exploited by using it to build meta-models and performing a search in them. Further, from the evolutionary algorithms perspective, it is interesting to apply different kind of meta-heuristics on this kind of problems to exploit their advantages. For instance,

MOPSO, that have a high pressure towards the set of interest, are a good candidate as search engine of the first subpopulation of NϵGA.

Finally, there exists several definition and kinds of uncertainty. In this work, we have focused on lre and uncertainty in decision variables, such as the one found in manufacturing tolerance errors. Thus, it would be interesting to extend the algorithm to the treatment of other kinds and definitions of uncertainty. For instance, it would be interesting to compute the confidence radius of a given basin of attraction. It also opens the possibility to design an archiving technique to get an estimation of the domain of attraction of a given MOP and to integrate the techniques with a multi-objective evolutionary algorithm. Finally, it would be interesting to further test the novel methods on real world applications.

Appendix A

Test Functions

In this Appendix, we define the MOPs used in this thesis.

Table A.1: MOPs used in this work.

Problem	Description	Comments
Conv2	$F(x) = (f_1(x), f_2(x))$, where: $f_1(x_1, x_2) = (x_1 - 1)^2 + (x_2 - 1)^4$, $f_2(x_1, x_2) = (x_1 + 1)^2 + (x_2 + 1)^2$.	$-3 \leq x_1 \leq 3$ $-3 \leq x_2 \leq 3$
Conv3	$F(x) = (f_1(x), f_2(x), f_3(x))$, where: $f_1(x_1, x_2) = (x_1 - 1)^4 + (x_2 - 1)^2 + (x_3 - 1)^2$, $f_2(x_1, x_2) = (x_1 + 1)^2 + (x_2 + 1)^4 + (x_3 + 1)^2$, $f_2(x_1, x_2) = (x_1 + 1)^2 + (x_2 + 1)^4 + (x_3 + 1)^4$.	$-3 \leq x_1 \leq 3$ $-3 \leq x_2 \leq 3$ $-3 \leq x_3 \leq 3$
Deb99	$F(x) = (f_1(x), f_2(x))$, where: $f_1(x_1, x_2) = x_1$, $f_2(x_1, x_2) = \frac{g(x_2)}{x_1}$, where: $g(x) = 2 - \exp\left(-\left(\frac{x_2 - 0.2}{0.004}\right)^2\right) - 0.8\exp\left(-\left(\frac{x_2 - 0.6}{0.4}\right)^2\right)$	$0 < x_1 \leq 1$ $0 \leq x_2 \leq 1$
Two-on-one	$F(x) = (f_1(x), f_2(x))$, where: $f_1(x) = (x_1 - t_1(c + 2a) + a)^2 + (x_2 - t_2 b)^2 + \delta_t$ $f_2(x) = (x_1 - t_1(c + 2a) - a)^2 + (x_2 - t_2 b)^2 + \delta_t$ where $t_1 = \text{sgn}(x_1)\min\left(\left\lceil \frac{\lvert x_1\rvert - a - c/2}{2a + c} \right\rceil, 1\right)$, $t_2 = \text{sgn}(x_2)\min\left(\left\lceil \frac{\lvert x_2\rvert - b/2}{b} \right\rceil, 1\right)$, $\delta_t = \begin{cases} 0 & \text{for} \quad t_1 = 0 \text{ and } t_2 = 0 \\ 0.01 & \text{else} \end{cases}$	$a = 0.5$ $b = 5$ $c = 5$ $-10 \leq x_1 \leq 10$ $-10 \leq x_2 \leq 10$
Sym-part	$F(x) = (f_1(x), f_2(x))$, where: $f_1(x) = (x_1 - t_1(c + 2a) + a)^2 + (x_2 - t_2 b)^2 + \delta_t$ $f_2(x) = (x_1 - t_1(c + 2a) - a)^2 + (x_2 - t_2 b)^2 + \delta_t$ where $t_1 = \text{sgn}(x_1)\min\left(\left\lceil \frac{\lvert x_1\rvert - a - c/2}{2a + c} \right\rceil, 1\right)$, $t_2 = \text{sgn}(x_2)\min\left(\left\lceil \frac{\lvert x_2\rvert - b/2}{b} \right\rceil, 1\right)$, $\delta_t = \begin{cases} 0 & \text{for} \quad t_1 = 0 \text{ and } t_2 = 0 \\ 0.01 & \text{for} \quad t_1 = -1 \text{ and } t_2 = 0 \\ 0.02 & \text{for} \quad t_1 = 1 \text{ and } t_2 = 0 \\ 0.03 & \text{for} \quad t_1 = 0 \text{ and } t_2 = -1 \\ 0.04 & \text{for} \quad t_1 = -1 \text{ and } t_2 = -1 \\ 0.05 & \text{for} \quad t_1 = 1 \text{ and } t_2 = -1 \\ 0.06 & \text{for} \quad t_1 = 0 \text{ and } t_2 = 1 \\ 0.07 & \text{for} \quad t_1 = -1 \text{ and } t_2 = 1 \\ 0.08 & \text{for} \quad t_1 = 1 \text{ and } t_2 = 1 \end{cases}$	$a = 0.5$ $b = 5$ $c = 5$ $-20 \leq x_1 \leq 20$ $-20 \leq x_2 \leq 20$
SSW	$F(x) = (f_1(x), f_2(x))$, where: $f_1(x) = \sum_{j=1}^{n} x_j$, $f_2(x) = 1 - \prod_{j=1}^{n}(1 - w_j(x_j))$, $w_j(z) = \begin{cases} 0.01 \cdot \exp(-(\frac{z}{20})^{2.5}) & \text{for} \quad j = 1, 2 \\ 0.01 \cdot \exp(-\frac{z}{15}) & \text{for} \quad j > 2 \end{cases}$	$0 \leq x_1 \leq 40$ $0 \leq x_2 \leq 40$ $0 \leq x_3 \leq 40$
Omni-test	$F(x) = (f_1(x), f_2(x))$, where: $f_1(x) = \sum_{j=1}^{n} \sin(\pi x_i)$, $f_2(x) = \sum_{j=1}^{n} \cos(\pi x_i)$,	$0 \leq x_i \leq 6, \quad$ for: $i = 1, \ldots, 5$
Lamé super-spheres	$F(x) = (f_1(x), f_2(x))$, where: $f_1(x) = (1 + r)\cos(x_1)$, $f_2(x) = (1 + r)\sin(x_1)$, where $r = \sin^2(\pi\xi)$, $\xi = \frac{1}{n-1}\sum_{i=2}^{n} x_i$	$0 \leq x_1 \leq \frac{\pi}{2}$, $1 \leq x_i \leq 5, \quad$ for: $i = 2, \ldots, 5$

Appendix B

Real World Problems

B.1 Multi-objective Optimal PID Controller Design

B.1.1 Time-Delayed Control System

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 delayed system, the transfer function in the frequency domain leads to a transcendental character equation that helps determine the stability region for controller design. Unfortunately, the frequency domain analysis cannot be applied to nonlinear systems. In this section, we present an approach to design multi-objective optimal PID controls for linear and nonlinear systems with feedback time delay.

Consider a second order nonlinear dynamical system with time delays given by,

$$\begin{aligned} \dot{x}_1 &= x_2 \\ \dot{x}_2 &= f\left(x_1, x_2, x_1\left(t-\tau_s\right), x_2\left(t-\tau_s\right)\right) + u\left(t-\tau_c\right), \end{aligned} \tag{B.1}$$

where f is a nonlinear function of its arguments. τ_s is a system delay, and τ_c is a control delay. We consider a PID feedback control given by

$$u(t) = k_p\left[r(t) - x_1(t)\right] + k_i \int_0^t \left[r(\hat{t}) - x_1(\hat{t})\right] d\hat{t} - k_d x_2(t), \tag{B.2}$$

where $r(t)$ is a reference input, k_p, k_i and k_d are the PID control gains. We introduce a third state variable x_3 such that $\dot{x}_3(t) = r(t) - x_1(t)$. The extended system is governed by the following equations.

$$\begin{aligned} \dot{x}_1 &= x_2, \\ \dot{x}_2 &= f\left(x_1, x_2, x_1\left(t-\tau_s\right), x_2\left(t-\tau_s\right)\right) + u\left(t-\tau_c\right), \\ \dot{x}_3 &= r(t) - x_1(t), \end{aligned} \tag{B.3}$$

where

$$u(t) = k_p \left[r(t) - x_1(t) \right] + k_i x_3 - k_d x_2(t). \tag{B.4}$$

Assume that the closed-loop system is stable and $r(t)$ is a step function. In steady-state, we have a unique equilibrium solution,

$$\begin{aligned} & x_1^* = 1, \quad x_2^* = 0, \\ & x_3^* = -\frac{1}{k_i} f(1, 0, 1, 0). \end{aligned} \tag{B.5}$$

It should be pointed out that the uncontrolled nonlinear system may have multiple equilibrium solutions. The stability of the steady state response can be analyzed by linearizing the system. Let $\mathbf{z} = [z_1, z_2, z_3]^T$ be the perturbation of the system away from the steady state $\mathbf{x}^* = [x_1^*, x_2^*, x_3^*]^T$. We have

$$\dot{\mathbf{z}}(t) = \mathbf{A}\mathbf{z}(t) + \mathbf{A}_s \mathbf{z}\left(t - \tau_s\right) + \mathbf{A}_c \mathbf{z}\left(t - \tau_c\right), \tag{B.6}$$

where $\mathbf{A}$, $\mathbf{A}_s$ and $\mathbf{A}_c$ are matrices of the linearized system and are functions of the control gains. The stability of the linearized system can be analyzed by the method of continuous time approximation (CTA) Sun (2008); Song and Sun (2011).

B.1.2 Multi-objective Optimal Design Formulation

As an example of MOPs, we consider the multi-objective optimal control design with the gains $\mathbf{k} = [k_p, k_i, k_d]^T$ as design parameters for the system discussed the following examples. Peak time and overshoot are common objectives in time domain control design (Liu and Daley, 1999, 2000; Panda, 2011). We consider the multi-objective optimization problem (MOP) for the optimal control gain $\mathbf{k}$ to minimize the following three objectives

$$\min_{\mathbf{k} \in Q} \{t_p, M_p, e_{IAE}\} \text{ subject to the stability of the system (B.6)}, \tag{B.7}$$

where 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

$$e_{IAE} = \int_0^{T_{ss}} \left| r(\hat{t}) - x_1(\hat{t}) \right| d\hat{t}. \tag{B.8}$$

where $r(t)$ is a reference input and T_{ss} is the time when the response is close to be in the steady state. The closed-loop response of the system for each design trial can be computed with the help of numerical integration programs of delayed differential equations.

B.1.3 First Order Plus Time Delay System

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

$$X(s) = \frac{K}{Ts+1}e^{-\tau s}U(s), \tag{B.9}$$

where K and T are constants and τ is the time delay. We take $K = 1$, $T = 1$, $\tau = 0.25$ in this study. We augment the plant with an additional state x_2 such that $\dot{x}_2 = r(t) - x_1(t)$ where $r(t)$ is a reference input and $x_1 = x$. The augmented state equations and the control read

$$\dot{x}_1 = -\frac{1}{T}x_1(t) + \frac{K}{T}u(t-\tau), \tag{B.10}$$

$$\dot{x}_2 = r(t) - x_1(t), \tag{B.11}$$

$$u(t) = k_p[r(t) - x_1(t)] + k_i x_2(t). \tag{B.12}$$

We consider the MOP design of the feedback control with respect to the gains $\mathbf{k} = [k_p, k_i]^T$. The space of the parameters is chosen as follows,

$$Q = \{\mathbf{k} \in [0,4] \times [0,5] \subset \mathbf{R}^2\}. \tag{B.13}$$

Design objectives are selected as shown in the next equation

$$\min_{\mathbf{k}\in Q}\{t_p, M_p, e_{IAE}\} \text{ subject to the stability of the system (B.6)}, \tag{B.14}$$

The stability of closed loop system near setpoint is computed as

$$\lambda = \max\{\mathrm{Re}\,[eig(\mathbf{A})]\} \tag{B.15}$$

where $\mathbf{A}$ is the CTA discretized coefficient matrix. To further ensure control quality, we impose constrains over objectives

$$\{t_p, M_p, e_{IAE}, \lambda\} \leq \{1.5s, 5\%, 0.8, -1.5\} \tag{B.16}$$

B.1.4 Second Order Oscilator

Next, we consider a second order oscillator subject to a proportional-integral-derivative (PID) control (Hernández et al., 2013).

$$\ddot{x} + 2\zeta\omega_n\dot{x} + \omega_n^2 x = \omega_n^2 u(t), \tag{B.17}$$

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

$$u(t) = k_p\left[r(t) - x(t)\right] + k_i\int_0^t \left[r(\hat{t}) - x(\hat{t})\right]d\hat{t} - k_d\dot{x}(t), \tag{B.18}$$

$r(t)$ is a step input, k_p, k_i and k_d are the PID control gains. We consider the MOP 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,

$$Q = \{\mathbf{k} \in [10, 50] \times [1, 30] \times [1, 2] \subset \mathbb{R}^3\}. \tag{B.19}$$

Peak time and overshoot are common in time domain control design objectives Liu and Daley (1999, 2000); Panda (2011). We consider the multi-objective optimization problem to design the control gain $\mathbf{k}$,

$$\min_{\mathbf{k} \in Q} \{t_p, M_p, e_{IAE}\}, \tag{B.20}$$

where 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

$$e_{IAE} = \int_0^{T_{ss}} \left| r(\hat{t}) - x(\hat{t}) \right| d\hat{t}, \tag{B.21}$$

where $r(t)$ is a reference input and T_{ss} is the time when the response is close to be in the steady state. The closed-loop response of the system for each design trial is computed with the help of closed form solutions. The integrated absolute tracking error e_{IAE} is calculated over time with $T_{ss} = 20s$. In this case, the error was considered to be $\delta = [0.4, 0.29, 0.01]^T$ which corresponds to a 1% error and $\epsilon = [0.10.10.1]^T$.

B.1.5 Second Order Linear-Time Invariant System

Consider a second order oscillator subject to a delayed PID control.

$$\ddot{x} + 2\zeta\omega_n \dot{x} + \omega_n^2 x = \omega_n^2 u(t - \tau), \tag{B.22}$$

where $\omega_n = 5$, $\zeta = 0.01$, $\tau = 0.1$,

$$u(t) = k_p \left[r(t) - x(t) \right] + k_i \int_0^t \left[r(\hat{t}) - x(\hat{t}) \right] d\hat{t} - k_d \dot{x}(t), \tag{B.23}$$

$r(t)$ is a step input, k_p, k_i and k_d are the PID control gains. We consider the MOP in Section B.1.2 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,

$$Q = \{\mathbf{k} \in [0, 1] \times [0, 0.5] \times [2, 5] \subset \mathbf{R}^3\}. \tag{B.24}$$

Objectives of this system are the same as the previous one and the performance constrains are set as

$$\{t_p, M_p, e_{IAE}, \lambda\} \leq \{1.5s, 5\%, 0.8, -1\}. \tag{B.25}$$

B.1.6 Nonlinear Duffing System

Next, we study the Duffing system such that

$$f\left(x_1, x_2, x_1\left(t-\tau_s\right), x_2\left(t-\tau_s\right)\right) = -ax_1 - bx_1^3 - cx_2, \tag{B.26}$$

where f is the term of the general second order system defined in Equation (B.1). $a = -1$, $b = 0.25$ and $c = 0.01$. Note that the system at the origin of the state space is unstable. The control time delay is 0.05 seconds. The system is under the delayed PID control in Equation (B.4). We study the multi-objective optimization problem defined in Equation (B.7) to design the control gain $\mathbf{k}$. The time-domain response of the Duffing system for each selection of the control gain is generated with the delayed differential equation integration algorithm (dde23) in Matlab. The integrated absolute tracking error e_{IAE} is calculated over time with $T_{ss} = 20$ seconds. The design space for the parameters is chosen as follows,

$$Q = \{\mathbf{k} \in [80, 120] \times [10, 30] \times [10, 30] \subset \mathbf{R}^3\}. \tag{B.27}$$

We impose the constraints

$$[t_p, M_p, e_{IAE}, \lambda] \leq [2.5, 6\%, 0.75, -0.25], \tag{B.28}$$

The constraint on the eigenvalues is intended to provide the stability robustness of the optimized control system. Since the original system is nonlinear, the stability condition should be imposed on the steady-state equilibrium solutions.

B.2 Multi-objective Optimal Full State Feedback Control Design

B.2.1 Regulation Control: An Inverted Pendulum

An inverted pendulum model is used to study the full state feedback regulation control in this subsection. The model of the 2DOF inverted pendulum is given below.

$$\dot{\mathbf{x}}(t) = \mathbf{A}\mathbf{x}(t) + \mathbf{B}u(t), \tag{B.29}$$

where

$$\mathbf{A} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 53.1012 & -0.6586 & 0.6575 \\ 0 & 98.3814 & -0.6575 & 1.2182 \end{bmatrix}, \tag{B.30}$$

$$\mathbf{B} = [0, 0, 274.4012, 273.9627]^T. \tag{B.31}$$

The state space vector is given as $\mathbf{x} = [\theta, \alpha, \dot{\theta}, \dot{\alpha}]^T$ where θ denotes the angle of a rotating base and α denotes the angle of the pendulum consisting of a rigid rod.

Figure B.1: The rotary flexible joint experimental setup made by Quanser.

$\alpha = 0$ represents the upward vertical position of the pendulum. The model of the pendulum is provided by Quanser.

A full state feedback control $u = -\mathbf{kx}$ with $\mathbf{k} = [k_{p,\theta}, k_{p,\alpha}, k_{d,\theta}, k_{d,\alpha}]$ is applied to stabilize the system at the upward vertical position. The first priority of the control design is to guarantee the robust stability of the closed-loop system. Thus, we select the maximum real part of the closed loop eigenvalues denoted by $\lambda_{\max}$ as an objective. Since the goal of control is to make the pendulum stand up, we consider the maximum absolute oscillatory response $\max|\alpha|$ of α when α is in the vicinity of the upward vertical position. The peak time $t_{p,\alpha}$ when α reaches $\max|\alpha|$ is also considered as a measure of the regulation speed. The system settling time t_s is considered when both θ and α reach their steady-state positions. Therefore, the full state control MOP design of the inverted pendulum can be formulated as

$$\min_{\mathbf{k}\in Q}\{t_{p,\alpha}, \max|\alpha|, t_s, \lambda_{\max}\}. \tag{B.32}$$

B.2.2 Tracking Control: A Flexible Rotary Arm

Here, we present the full state feedback tracking control of a flexible rotary arm. The experimental apparatus by Quanser is shown in Figure B.1.

Feedback controls are often designed to achieve the best tracking response of the system according to the time domain specifications such as the peak time, settling time, overshoot, and steady-state tracking error. The state vector describing the system dynamics is given as $\mathbf{x} = [\theta, \alpha, \dot{\theta}, \dot{\alpha}]^T$ where θ is the angle of the base and α is the relative angle of the flexible arm with respect to the base. The control goal is to let θ follow a given command while keeping the vibration of α minimal.

The state equation for this system has the same form as Equation (B.29) with the

state matrices given by

$$\mathbf{A} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 628.5625 & -40.4033 & 0 \\ 0 & -1024.7473 & 40.4033 & 0 \end{bmatrix}, \tag{B.33}$$

$$\mathbf{B} = [0, 0, 61.7567, -61.7567]^T. \tag{B.34}$$

To achieve tracking control, we need to augment the original dynamic system by introducing an additional state variable $x_i = \int_0^t (\theta - \theta_d) ds$. The state equation for the augmented system can be written as

$$\dot{\mathbf{x}}_e(t) = \mathbf{A}_e \mathbf{x}_e(t) + \mathbf{B}_e u(t) + \mathbf{G}_e \theta_d(t), \tag{B.35}$$

where $\mathbf{x}_e(t) = [\mathbf{x}^T, x_i]^T$ is the augmented state vector. $\mathbf{A}_e, \mathbf{B}_e$ and $\mathbf{G}_e$ are given as

$$\mathbf{A}_e = \begin{bmatrix} \mathbf{A} & \mathbf{0} \\ \mathbf{e} & 0 \end{bmatrix}, \mathbf{B}_e = \begin{bmatrix} \mathbf{B} \\ 0 \end{bmatrix}, \mathbf{G}_e = \begin{bmatrix} \mathbf{0} \\ -1 \end{bmatrix}, \tag{B.36}$$

where $\mathbf{e} = [1, 0, 0, 0]^T$.

The full state feedback control for the augmented system is given as

$$u(t) = -\mathbf{k}_e \mathbf{x}_e(t) + k_{p,\theta} \theta_d(t), \tag{B.37}$$

where

$$\mathbf{k}_e = [k_{p,\theta}, k_{p,\alpha}, k_{d,\theta}, k_{d,\alpha}, k_{i,\theta}]. \tag{B.38}$$

The performance indices to be minimized are the overshoot $M_{p,\theta}$, settling time $t_{s,\theta}$ and absolute integrated error e_{IAE} of θ. The control objectives of θ are selected to optimize the tracking control. In order to suppress the unwanted vibration of α, the maximum absolute response of α, denoted as $\max|\alpha|$, and the settling time of $t_{s,\alpha}$ are also considered. All these objectives serve the control mission that θ follows the given command and α oscillates as little as possible. To ensure the robust stability of the closed loop system, we also minimize the maximum real part $\lambda_{\max}$ of the closed-loop eigenvalues, and even impose a constraint on it. The MOP formulation of this control design can be stated as

$$\min_{\mathbf{k}_e \in Q} \{t_{s,\theta}, M_{p,\theta}, e_{IAE}, \max|\alpha|, t_{s,\alpha}, \lambda_{\max}\}. \tag{B.39}$$

The lower and upper bounds of the five feedback gains are chosen to be $[0, -2.5, 0, 0, 0]$ and $[10, 0, 1.5, 1.5, 0.5]$ so that

$$Q = \{\mathbf{k}_e \in \mathbf{R}^5 \mid [0, -2.5, 0, 0, 0] \leq_p \mathbf{k} \leq_p [10, 0, 1.5, 1.5, 0.5]\}. \tag{B.40}$$

We further narrow down the choices by imposing a performance constraint such that

$$\{t_{s,\theta}, M_{p,\theta}, e_{IAE}, \max|\alpha|, t_{s,\alpha}, \lambda_{\max}\} \leq_p [2s, 8\%, 0.1, 5^\circ, 1s, -0.02]. \tag{B.41}$$

Bibliography

Allgower, E. and Georg, K. (1990). *Numerical Continuation Methods*. Springer.

Avigad, G. and Branke, J. (2008). Embedded evolutionary multi-objective optimization for worst case robustness. In *Proceedings of the 10th Annual Conference on Genetic and Evolutionary Computation*, GECCO '08, pages 617–624, New York, NY, USA. ACM.

Bader, J. and Zitzler, E. (2010). Robustness in Hypervolume-based Multiobjective Search. Technical Report 317, ETH.

Barrico, C. and Antunes, C. H. (2006). Robustness analysis in multi-objective optimization using a degree of robustness concept. In *2006 IEEE International Conference on Evolutionary Computation*, pages 1887–1892.

Beume, N., Naujoks, B., and Emmerich, M. (2007). SMS-EMOA: Multiobjective selection based on dominated hypervolume. *European Journal of Operational Research*, 181(3):1653–1669.

Beyer, H.-G. and Sendhoff, B. (2007). Robust optimization – a comprehensive survey. *Computer Methods in Applied Mechanics and Engineering*, 196(33–34):3190 – 3218.

Blanquero, R. and Carrizosa, E. (2002). A. d.c. biobjective location model. *Journal of Global Optimization*, 23(2):569–580.

Bowman, V. J. (1976). On the Relationship of the Tchebycheff Norm and the Efficient Frontier of Multiple-Criteria Objectives. In *Multiple Criteria Decision Making*, volume 130 of *Lecture Notes in Economics and Mathematical Systems*, pages 76–86. Springer Berlin Heidelberg.

Branke, J. (1998). *Creating robust solutions by means of evolutionary algorithms*, pages 119–128. Springer Berlin Heidelberg, Berlin, Heidelberg.

Bursal, F. H. and Hsu, C. S. (1989). Application of a cell-mapping method to optimal control problems. *International Journal of Control*, 49(5):1505–1522.

Coello Coello, C. A. and Cruz Cortés, N. (2005). Solving multiobjective optimization problems using an artificial immune system. *Genetic Programming and Evolvable Machines*, 6(2):163–190.

Coello Coello, C. A., Lamont, G. B., and Van Veldhuizen, D. A. (2007). *Evolutionary Algorithms for Solving Multi-Objective Problems*. Springer, New York, second edition. ISBN 978-0-387-33254-3.

Coello Coello, C. A., Lamont, G. B., and Veldhuizen, D. A. (2007). *Evolutionary Algorithms for Solving Multi-Objective Problems*. Springer, New York.

Das, I. and Dennis, J. (1998). Normal-boundary intersection: A new method for generating the Pareto surface in nonlinear multicriteria optimization problems. *SIAM Journal of Optimization*, 8:631–657.

Deb, K. (2001a). *Multi-Objective Optimization using Evolutionary Algorithms*. John Wiley & Sons, Chichester, UK. ISBN 0-471-87339-X.

Deb, K. (2001b). *Multi-Objective Optimization using Evolutionary Algorithms*. John Wiley & Sons, Chichester, UK. ISBN 0-471-87339-X.

Deb, K. (2001c). *Multi-Objective Optimization Using Evolutionary Algorithms*. John Wiley & Sons, Inc.

Deb, K. and Gupta, H. (2006). Introducing Robustness in Multi-objective Optimization. *Evolutionary Computation*, 14(4):463–494.

Deb, K., Mohan, M., and Mishra, S. (2005). Evaluating the epsilon-domination based multi-objective evolutionary algorithm for a quick computation of Pareto-optimal solutions. *Evolutionary Computation*, 13(4):501–525.

Deb, K., Pratap, A., Agarwal, S., and Meyarivan, T. (2002). A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*, 6(2):182–197.

Deb, K. and Tiwari, S. (2008). Omni-optimizer: A generic evolutionary algorithm for single and multi-objective optimization. *European Journal of Operational Research*, 185(3):1062–1087.

Dellnitz, M. and Hohmann, A. (1997). A subdivision algorithm for the computation of unstable manifolds and global attractors. *Numerische Mathematik*, 75(3):293–317.

Dellnitz, M. and Junge, O. (1998). An adaptive subdivision technique for the approximation of attractors and invariant measures. *Computing and Visualization in Science*, 1(2):63–68.

Dellnitz, M. and Junge, O. (2002). Set oriented numerical methods for dynamical systems. *Handbook of Dynamical Systems*, 2:221–264.

Dellnitz, M., Schütze, O., and Hestermeyer, T. (2005a). Covering Pareto sets by multilevel subdivision techniques. *Journal of Optimization Theory and Applications*, 124(1):113–136.

Dellnitz, M., Schütze, O., and Hestermeyer, T. (2005b). Covering Pareto sets by multilevel subdivision techniques. *Journal of Optimization Theory and Applications*, 124:113–155.

Doolittle, E. K., Kerivin, H. L., and Wiecek, M. M. (2012). A robust multiobjective optimization problem with application to internet routing. In *Tech. Rep. R2012-11-DKW, Clemson University.*

Ehrgott, M., Ide, J., and Schöbel, A. (2014). Minmax robustness for multi-objective optimization problems. *European Journal of Operational Research*, 239(1):17–31.

Eiben, A. E. and Rudolph, G. (1999). Theory of evolutionary algorithms: A bird's eye view. *Theoretical Computer Science*, 1:3–9.

Eichfelder, G. (2009). Scalarizations for adaptively solving multi-objective optimization problems. *Computational Optimization and Applications*, 44(2):249.

Eichfelder, G., Krüger, C., and Schöbel, A. (2017). Decision uncertainty in multiobjective optimization. *Journal of Global Optimization*, 69(2):485–510.

Emmerich, M., D. A. (2007). *Test problems based on lamé superspheres*, pages 922–936. Springer Berlin Heidelberg, Berlin, Heidelberg.

Emmerich, M., Beume, N., and Naujoks, B. (2005). An emo algorithm using the hypervolume measure as selection criterion. In *International Conference on Evolutionary Multi-Criterion Optimization*, pages 62–76. Springer.

Engau, A. and Wiecek, M. M. (2007). Generating epsilon-efficient solutions in multiobjective programming. *European Journal of Operational Research*, 177(3):1566–1579.

Ester, M., Kriegel, H.-P., Sander, J., and Xu, X. (1996). A density-based algorithm for discovering clusters a density-based algorithm for discovering clusters in large spatial databases with noise. In *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*, KDD'96, pages 226–231. AAAI Press.

Fernández Cruz, J., Sun, J.-Q., Schütze, O., and Xiong, F.-R. (2014). Parallel Cell Mapping for Unconstrained Multi-Objective Optimization Problems. In *EVOLVE – A Bridge between Probability, Set Oriented Numerics and Evolutionary Computation V*. Springer.

Fliege, J. (2004). Gap-free computation of Pareto-points by quadratic scalarizations. *Mathematical Methods of Operations Research*, 59:69–89.

Fliege, J. and Fux Svaiter, B. (2000). Steepest descent methods for multicriteria optimization. *Mathematical Methods of Oper Res*, 51(3):479–494.

Fliege, J. and Werner, R. (2014). Robust multiobjective optimization & applications in portfolio optimization. *European Journal of Operational Research*, 234(2):422–433.

Gomez, M., Martinez-Marie, T., Sanchez, S., and Meziat, D. (2008). Optimal control for wheeled mobile vehicles based on cell mapping techniques. In *Intelligent Vehicles Symposium, 2008 IEEE*, pages 1009–1014.

Gunawan, S. and Azarm, S. (2005). Multi-objective robust optimization using a sensitivity region concept. *Structural and Multidisciplinary Optimization*, 29(1):50–60.

Guttalu, R. S. and Zufiria, P. J. (1993). The adjoining cell mapping and its recursive unraveling, part ii: Application to selected problems. *Nonlinear Dynamics*, 4(4):309–336.

Hanne, T. (1999). On the convergence of multiobjective evolutionary algorithms. *European Journal Of Operational Research*, 117(3):553–564.

Heinonen, J. (2001). *Lectures on Analysis on Metric Spaces*. Springer Science & Business Media.

Hernández, C., Naranjani, Y., Sardahi, Y., Liang, W., Schütze, O., and Sun, J.-Q. (2013). Simple cell mapping method for multi-objective optimal feedback control design. *International Journal of Dynamics and Control*, 1(3):231–238.

Hernández, C., Sun, J.-Q., and Schütze, O. (2013). Computing the set of approximate solutions of a multi-objective optimization problem by means of cell mapping techniques. In *EVOLVE-A Bridge between Probability, Set Oriented Numerics, and Evolutionary Computation IV*, pages 171–188. Springer.

Hernández, C., Sun, J.-Q., and Schütze, O. (2013). Computing the set of approximate solutions of a multi-objective optimization problem by means of cell mapping techniques. In et al., M. E., editor, *EVOLVE – A Bridge between Probability, Set Oriented Numerics and Evolutionary Computation IV*, pages 171–188. Springer.

Hillermeier, C. (2001). *Nonlinear Multiobjective Optimization: A Generalized Homotopy Approach*, volume 135. Springer Science & Business Media.

Horoba, C. and Neumann, F. (2008). Benefits and drawbacks for the use of epsilon-dominance in evolutionary multi-objective optimization. In *Genetic and Evolutionary Computation Conference (GECCO-2008)*, pages 641–648.

Hsu, C. S. (1980). A theory of cell-to-cell mapping dynamical systems. *Journal of Applied Mechanics*, 47:931–939.

Hsu, C. S. (1985). A discrete method of optimal control based upon the cell state space concept. *Journal of Optimization Theory and Applications*, 46(4):547–569.

Hsu, C. S. (1987). *Cell-to-cell mapping: a method of global analysis for nonlinear systems*. Applied mathematical sciences. Springer-Verlag.

Hwang, C. and Masud, A. (1979). Multiple objective decision making-methods and applications: a state-of-the-art survey. *Lecture notes in economics and mathematical systems*, (164).

Ide, J. and Schöbel, A. (2016). Robustness for uncertain multi-objective optimization: A survey and analysis of different concepts. *OR Spectrum*, 38(1):235–271.

Jahn, J. (2006). Multiobjective search algorithm with subdivision technique. *Computational Optimization and Applications*, 35(2):161–175.

Jin, Y. and Branke, J. (2005). Evolutionary optimization in uncertain environments-a survey. *IEEE Transactions on Evolutionary Computation*, 9(3):303–317.

Kemeny, J. and Snell, J. (1976). *Finite Markov Chains: With a New Appendix "Generalization of a Fundamental Matrix"*. Undergraduate Texts in Mathematics. Springer.

Kiureghian, A. D. and Ditlevsen, O. (2007). Aleatory or epistemic? does it matter?

Knowles, J. D. and Corne, D. W. (2000). Approximating the non-dominated front using the Pareto Archived Evolution Strategy. *Evolutionary Computation*, 8(2):149–172.

Knowles, J. D. and Corne, D. W. (2004). *Metaheuristics for Multiobjective Optimisation*, volume 535 of *Lecture Notes in Economics and Mathematical Systems*, chapter Bounded Pareto Archiving: Theory and Practice, pages 39–64. Springer.

Kramer, O. and Danielsiek, H. (2010). Dbscan-based multi-objective niching to approximate equivalent pareto-subsets. In *Proceedings of the 12th Annual Conference on Genetic and Evolutionary Computation*, GECCO '10, pages 503–510, New York, NY, USA. ACM.

Kramer, O. and Danielsiek, H. (2011). A clustering-based niching framework for the approximation of equivalent pareto-subsets. *International Journal of Computational Intelligence and Applications*, 10(03):295–311.

Kramer, O. and Koch, P. (2009). Rake selection: A novel evolutionary multi-objective optimization algorithm. In *Proceedings of the 32Nd Annual German Conference on Advances in Artificial Intelligence*, KI'09, pages 177–184, Berlin, Heidelberg. Springer-Verlag.

Kuhn, H. and Tucker, A. (1951). Nonlinear programming. pages 481–492.

Kuroiwa, D. and Lee, G. M. (2012). On robust multiobjective optimization. *Vietnam J. Math*, 40(2-3):305–317.

Lara, A. (2012). *Using Gradient Based Information to build Hybrid Multi-objective Evolutionary Algorithms*. PhD thesis, CINVESTAV-IPN.

Laumanns, M., Thiele, L., Deb, K., and Zitzler, E. (2002). Combining convergence and diversity in evolutionary multiobjective optimization. *Evolutionary Computation*, 10(3):263–282.

Liu, G. P. and Daley, S. (1999). Optimal-tuning PID controller design in the frequency domain with application to a rotary hydraulic system. *Control Engineering Practice*, 7(7):821–830.

Liu, G. P. and Daley, S. (2000). Optimal-tuning nonlinear PID control of hydraulic systems. *Control Engineering Practice*, 8(9):1045–1053.

Loridan, P. (1984). ϵ-Solutions in Vector Minimization Problems. *Journal of Optimization, Theory and Application*, 42:265–276.

Martínez-Marín, T. and Zufiria, P. J. (1999). Optimal control of non-linear systems through hybrid cell-mapping/artificial-neural-networks techniques. *International Journal of Adaptive Control and Signal Processing*, 13(4):307–319.

Mendes, M., Soares, G., Coulomb, J., and Vasconcelos, J. (2013). A Surrogate Genetic Programming Based Model to Facilitate Robust Multi-Objective Optimization: A Case Study in Magnetostatics. *IEEE TRANSACTIONS ON MAGNETICS*, 32(5):585–618.

Mersmann, O., Bischl, B., Trautmann, H., Preuss, M., Weihs, C., and Rudolph, G. (2011). Exploratory landscape analysis. In *GECCO*, pages 829–836.

Miettinen, K. (1999). *Nonlinear Multiobjective Optimization*. Kluwer Academic Publishers, Boston, Massachusetts.

Moubayed, N. A., Petrovski, A., and McCall, J. (2014). (DMOPSO)-M-2: MOPSO Based on Decomposition and Dominance with Archiving Using Crowding Distance in Objective and Solution Spaces. *Evolutionary Computation*, 22(1).

Naranjani, Y., Hernández, C., Xiong, F.-R., Schütze, O., and Sun, J.-Q. (2013a). A Hybrid Algorithm for the Simple Cell Mapping Method in Multi-objective Optimization. In et al., M. E., editor, *EVOLVE – A Bridge between Probability, Set Oriented Numerics and Evolutionary Computation IV*, pages 207–223. Springer.

Naranjani, Y., Sardahi, Y., Sun, J.-Q., Hernández, C., and Schütze, O. (2013b). Fine structure of Pareto front of multi-objective optimal feedback control design. In *Proceedings of ASME 2013 Dynamic Systems and Control Conference*, page V001T15A009.

Okabe, T., Jin, Y., Olhofer, M., and Sendhoff, B. (2004). *Parallel Problem Solving from Nature - PPSN VIII: 8th International Conference, Birmingham, UK, September 18-22, 2004. Proceedings*, chapter On Test Functions for Evolutionary Multi-objective Optimization, pages 792–802.

Panda, S. (2011). 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):1296–1308.

Pareto, V. (1971 original edition in French in 1927). *Manual of Political Economy*. The MacMillan Press.

Preuss, M., Naujoks, B., and Rudolph, G. (2006). *Pareto Set and EMOA Behavior for Simple Multimodal Multiobjective Functions*, pages 513–522. Springer Berlin Heidelberg, Berlin, Heidelberg.

Rudolph, G. and Agapie, A. (2000). Convergence Properties of Some Multi-Objective Evolutionary Algorithms. In *Proceedings of the 2000 Conference on Evolutionary Computation*, volume 2, pages 1010–1016, Piscataway, New Jersey. IEEE Press.

Rudolph, G., Naujoks, B., and Preuss, M. (2007). *Capabilities of EMOA to Detect and Preserve Equivalent Pareto Subsets*, pages 36–50. Springer Berlin Heidelberg, Berlin, Heidelberg.

Rudolph, G. and Preuss, M. (2009). A multiobjective approach for finding equivalent inverse images of pareto-optimal objective vectors. In *2009 IEEE Symposium on Computational Intelligence in Multi-Criteria Decision-Making, MCDM 2009, Nashville, TN, USA, March 30 - April 2, 2009*, pages 74–79.

Rudolph, G., Schütze, O., Grimme, C., Domínguez-Medina, C., and Trautmann, H. (2016). Optimal averaged hausdorff archives for bi-objective problems: theoretical and numerical results. *Computational Optimization and Applications*, 64(2):589–618.

Ruhe, G. and Fruhwirt, B. (1990). ϵ-optimality for bicriteria programs and its application to minimum cost flows. *Computing*, 44:21–34.

Saha, A., Ray, T., and Smith, W. (2011). Towards practical evolutionary robust multi-objective optimization. In *Evolutionary Computation (CEC), 2011 IEEE Congress on*, pages 2123–2130.

Schaeffler, S., Schultz, R., and Weinzierl, K. (2002). Stochastic Method for the Solution of Unconstrained Vector Optimization Problems. *Journal of Optimization Theory and Applications*, 114(1):209–222.

Schäffler, S., Schultz, R., and Weinzierl, K. (2002). A stochastic method for the solution of unconstrained vector opimization problems. *Journal of Optimization, Theory and Application*, 114(1):209–222.

Schütze, O. (2004). *Set Oriented Methods for Global Optimization*. PhD thesis, University of Paderborn.

Schütze, O., Coello, C. A. C., and Talbi, E.-G. (2007a). Approximating the ε-efficient set of an mop with stochastic search algorithms. In *Mexican International Conference on Artificial Intelligence*, pages 128–138. Springer.

Schütze, O., Coello, C. A. C., Tantar, E., and Talbi, E.-G. (2008). Computing finite size representations of the set of approximate solutions of an MOP with stochastic search algorithms. In *GECCO 2008: Proceedings of the 10th annual conference on Genetic and evolutionary computation*, pages 713–720, New York, NY, USA. ACM.

Schütze, O., Coello Coello, C. A., Mostaghim, S., Talbi, E.-G., and Dellnitz, M. (2008). Hybridizing Evolutionary Strategies with Continuation Methods for Solving Multi-Objective Problems. *Engineering Optimization*, 40(5):383–402.

Schütze, O., Dell'Aere, A., and Dellnitz, M. (2005). On continuation methods for the numerical treatment of multi-objective optimization problems. In J. Branke et. al, editor, *Practical Approaches to Multi-Objective Optimization*, number 04461 in Dagstuhl Seminar Proceedings.

Schütze, O., Esquivel, X., Lara, A., and Coello Coello, C. A. (2012). Using the averaged Hausdorff distance as a performance measure in evolutionary multi-objective optimization. *IEEE Transactions on Evolutionary Computation*, 16(4):504–522.

Schütze, O., Lara, A., and Coello, C. A. C. (2011). The directed search method for unconstrained multi-objective optimization problems. In *Proceedings of the EVOLVE – A Bridge Between Probability, Set Oriented Numerics, and Evolutionary Computation*, pages 1–4.

Schütze, O., Laumanns, M., Coello, C. A. C., Dellnitz, M., and Talbi, E.-G. (2008). Convergence of stochastic search algorithms to finite size Pareto set approximations. *Journal of Global Optimization*, 41(4):559–577.

Schütze, O., Laumanns, M., Tantar, E., Coello, C. A. C., and Talbi, E.-G. (2007b). Convergence of stochastic search algorithms to gap-free Pareto front approximations. In *GECCO-2007*, pages 892–901.

Schütze, O., Laumanns, M., Tantar, E., Coello, C. A. C., and Talbi, E.-G. (2010). Computing gap free Pareto front approximations with stochastic search algorithms. *Evolutionary Computation*, 18(1):65–96.

Schütze, O., Martín, A., Lara, A., Alvarado, S., Salinas, E., and Coello, C. A. (2016). The directed search method for multi-objective memetic algorithms. *Comput. Optim. Appl.*, 63(2):305–332.

Schütze, O., Mostaghim, S., Dellnitz, M., and Teich, J. (2003). Covering Pareto sets by multilevel evolutionary subdivision techniques. In *Proceedings of Second International Conference on Evolutionary Multi-Criterion Optimization, EMO*, pages 118–32, Berlin, Germany.

Schütze, O., Vasile, M., and Coello Coello, C. (2008). On the benefit of ϵ-efficient solutions in multi objective space mission design.

Schütze, O., Vasile, M., and Coello Coello, C. A. (2011). Computing the set of epsilon-efficient solutions in multiobjective space mission design. *Journal of Aerospace Computing, Information, and Communication*, 8(3):53–70.

Schütze, O., Vasile, M., Junge, O., Dellnitz, M., and Izzo, D. (2009). Designing optimal low thrust gravity assist trajectories using space pruning and a multi-objective approach. *Engineering Optimization*, 41(2):155–181.

Schütze, O., Domínguez-Medina, C., Cruz-Cortés, N., de la Fraga, L. G., Sun, J.-Q., Toscano, G., and Landa, R. (2016). A scalar optimization approach for averaged hausdorff approximations of the pareto front. *Engineering Optimization*, 48(9):1593–1617.

Shir, O. M., Preuss, M., Naujoks, B., and Emmerich, M. (2009). *Enhancing Decision Space Diversity in Evolutionary Multiobjective Algorithms*, pages 95–109. Springer Berlin Heidelberg, Berlin, Heidelberg.

Song, B. and Sun, J. Q. (2011). Lowpass filter-based continuous-time approximation of delayed dynamical systems. *Journal of Vibration and Control*, 17(8):1173–1183.

Srinivas, N. and Deb, K. (1994). Muiltiobjective optimization using nondominated sorting in genetic algorithms. *Evolutionary computation*, 2(3):221–248.

Sun, J. Q. (1995). Vibration and sound radiation of non-uniform beams. *Journal of Sound and Vibration*, 185(5):827–843.

Sun, J. Q. (2008). A method of continuous time approximation of delayed dynamical systems. *Communications in Nonlinear Science and Numerical Simulation*, 14(4):998–1007.

Talbi, E.-G. (2009). Metaheuristics: From design to implementation.

Toffolo, A. and Benini, E. (2003). Genetic diversity as an objective in multi-objective evolutionary algorithms. *Evol. Comput.*, 11(2):151–167.

Tongue, B. H., G. K. (1988). Interpolated cell mapping of dynamical systems. *Journal of Applied Mechanics*, 55(2):461 – 466.

Ulrich, T., Bader, J., and Zitzler, E. (2010). Integrating decision space diversity into hypervolume-based multiobjective search. In *Proceedings of the 12th Annual Conference on Genetic and Evolutionary Computation*, GECCO '10, pages 455–462, New York, NY, USA. ACM.

Van Veldhuizen, D. A. (1999). *Multiobjective Evolutionary Algorithms: Classifications, Analyses, and New Innovations*. Wright-Patterson AFB, Ohio.

Wagner, T. and Trautmann, H. (2010). Integration of preferences in hypervolume-based multiobjective evolutionary algorithms by means of desirability functions. *IEEE Transactions on Evolutionary Computation*, 14(5):688–701.

White, D. J. (1986). Epsilon efficiency. *Journal of Optimization Theory and Applications*, 49(2):319–337.

White, D. J. (1998). Epsilon-dominating solutions in mean-variance portfolio analysis. *European Journal of Operational Research*, 105(3):457–466.

Xia, H., Zhuang, J., and Yu, D. (2014a). Combining crowding estimation in objective and decision space with multiple selection and search strategies for multi-objective evolutionary optimization. *IEEE transactions on cybernetics*, 44(3):378–393.

Xia, H., Zhuang, J., and Yu, D. (2014b). Multi-objective unsupervised feature selection algorithm utilizing redundancy measure and negative epsilon-dominance for fault diagnosis. *Neurocomputing*, 146:113–124.

Xiong, F. R., Qin, Z. C., Xue, Y., Schütze, O., and Q. Ding, J. Q. S. (2014). Multi-objective optimal design of feedback controls for dynamical systems with hybrid simple cell mapping algorithm. *Communications in Nonlinear Science and Numerical Simulation*, 19(5):1465–1473.

Xiong, F.-R., Schütze, O., Ding, Q., and Sun, J.-Q. (2016). Finding zeros of nonlinear functions using the hybrid parallel cell mapping method. *Communications in Nonlinear Science and Numerical Simulation*, 34(Supplement C):23 – 37.

Yuen, J., Gao, S., Wagner, M., and Neumann, F. (2012). An adaptive data structure for evolutionary multi-objective algorithms with unbounded archives. In *Evolutionary Computation (CEC), 2012 IEEE Congress on*, pages 1–8.

Zadeh, L. (1963). Optimality and Non-Scalar-Valued Performance Criteria. *IEEE Transactions on Automatic Control*, 8:59–60.

Zechman, E. M., Giacomoni, M. H., and Shafiee, M. E. (2011). A multi-objective niching co-evolutionary algorithm (mnca) for identifying diverse sets of non-dominated solutions. In *Proceedings of the 13th annual conference companion on Genetic and evolutionary computation*, pages 805–806. ACM.

Zhang, Q. and Li, H. (2007). MOEA/D: A multi-objective evolutionary algorithm based on decomposition. *IEEE Transactions on Evolutionary Computation*, 11(6):712–731.

Zhang, Q., Zhou, A., Zhao, S., Suganthan, P. N., Liu, W., and Tiwari, S. (2008). Multiobjective optimization test instances for the cec 2009 special session and competition.

Zhou, A., Zhang, Q., and Jin, Y. (2009). Approximating the set of pareto-optimal solutions in both the decision and objective spaces by an estimation of distribution algorithm. *IEEE Transactions on Evolutionary Computation*, 13(5):1167–1189.

Zitzler, E., Laumanns, M., and Thiele, L. (2002). SPEA2: Improving the Strength Pareto Evolutionary Algorithm for Multiobjective Optimization. In Giannakoglou, K. et al., editors, *Evolutionary Methods for Design, Optimisation and Control with Application to Industrial Problems (EUROGEN 2001)*, pages 95–100. International Center for Numerical Methods in Engineering (CIMNE).

Zitzler, E. and Thiele, L. (1998). *Multiobjective optimization using evolutionary algorithms — A comparative case study*, pages 292–301. Springer Berlin Heidelberg, Berlin, Heidelberg.

Zitzler, E. and Thiele, L. (1999). Multiobjective evolutionary algorithms: A comparative case study and the strength pareto evolutionary algorithm. *IEEE Transactions on Evolutionary Computation*, 3(4):257–271.

Zitzler, E., Thiele, L., and Bader, J. (2008). SPAM: Set preference algorithm for multiobjective optimization. In *Parallel Problem Solving From Nature – PPSN X*, pages 847–858.

Zufiria, P. J. and Guttalu, R. S. (1993). The adjoining cell mapping and its recursive unraveling, part I: Description of adaptive and recursive algorithms. *Nonlinear Dynamics*, 4(3):207–226.

Zufiria, P. J. and Martínez-Marín, T. (2003). Improved optimal control methods based upon the adjoining cell mapping technique. *Journal of Optimization Theory and Applications*, 118(3):657–680.

Zuiani, F. and Vasile, M. (2013). Multi agent collaborative search based on tchebycheff decomposition. *Computational Optimization and Applications*, 56(1):189–208.