



CENTRO DE INVESTIGACIÓN Y DE ESTUDIOS
AVANZADOS
DEL INSTITUTO POLITÉCNICO NACIONAL

Unidad Zacatenco

Departamento de Computación

**Mapeo de Celda a Celda para Optimización Global
Multi-objetivo**

Tesis que presenta

Carlos Ignacio Hernández Castellanos

obtener el Grado de

Maestro en Ciencias

en Computación

Director de la Tesis:

Dr. Oliver Schütze

México, D.F

Agosto 2013



CENTRO DE INVESTIGACIÓN Y DE ESTUDIOS
AVANZADOS
DEL INSTITUTO POLITÉCNICO NACIONAL

Zacatenco Campus

Computer Science Department

**Cell-to-Cell Mapping for Global Multi-Objective
Optimization**

Submitted by

Carlos Ignacio Hernández Castellanos

as fulfillment of the requirement for the degree of

Master in

Computer Science

Advisor:

Dr. Oliver Schütze

Mexico, D.F

August 2013

Resumen

De manera frecuente nos encontramos con el problema de optimizar varios objetivos de manera simultanea y típicamente estos objetivos están en conflicto entre ellos. A este tipo de problemas se les conoce con el nombre de problemas de optimización multi-objetivo (POM). En la mayoría de los casos la solución a estos problemas no es única, si no un compromiso entre los objetivos.

En esta tesis, presentamos métodos orientados a conjuntos para resolver estos problemas. En particular, nos enfocamos al problema de encontrar el conjunto de soluciones óptimas, así como el problema de encontrar el conjunto de soluciones aproximadas de un problema de optimización multi-objetivo. Este último conjunto es de interés para el tomador de decisiones, dado que le puede proporcionar soluciones adicionales a las óptimas para la implementación de su proyecto relacionado al POM. En este estudio, hacemos una primera adaptación de las bien conocidas técnicas de mapeo de celdas para el análisis global de sistemas dinámicos del problema en cuestión. Dado el carácter global del enfoque, estos métodos son adecuados para la investigación exhaustiva de problemas pequeños, incluyendo el cómputo del conjunto de soluciones aproximadas. También mostramos que nuestra propuesta es competitiva con los algoritmos evolutivos para problemas de un número bajo de dimensiones.

Abstract

One is frequently faced with the problem of optimizing several objectives simultaneously and typically these objectives are in conflict with each other. These kind of problems are known as multi-objective optimization problems (MOPs). Typically, the solution set of a given MOP does not consist of a single point as for single objective optimization problems but forms a $k - 1$ dimensional entity where k is the number of objectives involved in the MOP.

In this thesis, we present set oriented methods for the treatment of these problems. In particular, we address the problem of computing the set of optimal solutions as well as the set approximate solutions of a given MOP. The later set is of potential interest for the decision maker since it might give him/her additional solutions to the optimal ones for the realization of the project related to the MOP. In this study, we make a first attempt to adapt well-known cell mapping techniques for the global analysis of dynamical systems related to the problem at hand. Due to their global approach, these methods are well-suited for the thorough investigation of small problems, including the computation of the set of approximate solutions. We also show that the proposed approach is competitive to evolutionary strategies for low dimensional problems.

Acknowledgment

First, I would like to thank the CONACyT for the scholarship given by them for the pursuit of my master's studies.

Also, I want to thank the CINVESTAV and specially to the computer science department, to all my professors, to the secretaries and to my colleagues for all that they have thought me in the past two years.

Further, I would like to acknowledge my advisor Dr. Oliver Schütze for his support and guidance through this project.

Finally, I would like to give an special acknowledgement to my family that have all ways supported me.

Contents

List of Figures	xi
List of Tables	xiv
List of Algorithms	xv
1 Introduction	1
1.1 Motivation	1
1.2 The Problem	2
1.3 Objectives	2
1.4 Contributions	3
1.5 Organization of the thesis	4
2 Background and Related Work	5
2.1 Multi-objective Optimization	5
2.1.1 Formulation of the Problem	6
2.1.2 Pareto Optimality	6
2.1.3 Optimality conditions	9
2.1.4 Descent directions	10
2.2 Solving MOPs	10
2.2.1 Scalarization methods	10
2.2.2 Descent direction methods	13
2.2.3 Stochastic methods	16
2.2.4 Performance Measures	17
2.3 Dynamical Systems	19
2.3.1 Formulation of the problem	19
2.3.2 Fixed point	20
2.3.3 Periodic group	20
2.3.4 Domain of attraction	20
2.4 Global Analysis of Dynamical Systems	20
2.4.1 Discretization of the space	20
2.4.2 Simple cell mapping	21
2.4.3 Subdivision techniques	23

3	SCM for MOO	27
3.1	Simple Cell Mapping	27
3.1.1	Description of the SCM Method	27
3.1.2	Center point method	28
3.1.3	Analysis of SCM	32
3.2	Adapting the SCM Method for MOPs	32
3.2.1	Dynamical system	33
3.2.2	Step size control	33
3.2.3	Finding Pareto optimal solutions	34
3.3	Description the SCM method for MOPs	35
3.4	Numerical Results	38
3.4.1	Problem 1	38
3.4.2	Problem 2	39
3.4.3	Problem 3	39
3.4.4	Problem 4	39
4	SCM for Approximate Solutions	49
4.1	Archiving Techniques	50
4.2	Description of SCM for Approximate Solutions	51
4.3	Numerical Results	52
4.3.1	Problem 1	55
4.3.2	Problem 2	55
4.3.3	Problem 3	57
5	Analysis of Descent Direction Methods	61
5.1	The Problem	61
5.2	Empirical Comparison of DDMs	62
5.3	Comparison of DDMs by SCM	62
5.4	Numerical Results	64
5.4.1	Problem 1	64
5.4.2	Problem 2	64
5.4.3	Problem 3	64
6	Refine and Recover	71
6.1	Refine	71
6.1.1	Refine in Cell	71
6.1.2	SCM-Refine	72
6.1.3	SCM with subdivision techniques	73
6.1.4	Comparison of the methods	73
6.2	Recover	74
6.2.1	Moving Boxes	74
6.2.2	Continuation	76
6.2.3	Comparison of the methods	76

<i>CONTENTS</i>	xi
7 Conclusions and Future Work	79
7.1 Conclusions	79
7.2 Future Work	80
A Appendix A: Test Functions	81
Bibliography	89

List of Figures

2.1	Pareto Dominance	7
2.2	Pareto set and Pareto front	8
2.3	Approximate solutions	9
2.4	Curve of dominated points	13
2.5	Group Motion	22
2.6	Simple Cell Mapping	23
2.7	Subdivision Techniques	25
3.1	Neighborhood of a point	28
3.2	Periodic motion and global properties of a cell	29
3.3	Cell Mapping	31
3.4	Step size control for SCM	34
3.5	Iterations of SCM	36
3.6	SCM on MOP (A.1) on cell space	38
3.7	Effect of grid size on SCM	41
3.8	SCM for three Academic models on cell space	42
3.9	Numerical results for MOP (A.1 in page 81)	43
3.10	Numerical results for MOP (A.2 in page 82)	44
3.11	Numerical results for MOP (A.3 in page 83) with $n = 3$	45
3.12	Numerical results for MOP (A.3 in page 83) with $n = 5$	46
3.13	Numerical results for PID problem (A.6)	47
4.1	Iterations of SCM for $P_{Q\epsilon}$	53
4.2	Impact of <i>epsilon</i> values	54
4.3	SCM on MOP (A.1) for $P_{Q,\epsilon}$ on cell space	55
4.4	Numerical results for MOP (A.1 in page 81) for $P_{Q,\epsilon}$	56
4.5	Numerical results for MOP (A.2 in page 82) for $P_{Q,\epsilon}$	58
4.6	Numerical results for MOP (A.3 in page 83) for $P_{Q,\epsilon}$	59
4.7	Hypothetical decision making problem for MOP (A.2)	60
5.1	Empirical Comparison of descent direction methods	63
5.2	SCM on MOP (A.1) in cell space with different descent direction methods.	65
5.3	Comparison of Domain of Attraction for MOP (A.1)	66

5.4	SCM on MOP (A.4) in cell space with different descent direction methods.	67
5.5	Comparison of Domain of Attraction for MOP (A.4)	68
5.6	SCM on MOP (A.5) in cell space with different descent direction methods.	69
5.7	Comparison of Domain of Attraction for MOP (A.5)	70
6.1	Comparison of refinement methods	75
6.2	Comparison of recover methods	77
A.1	Pareto set and Pareto front of MOP (A.1).	81
A.2	Pareto set and Pareto front of MOP (A.2).	82
A.3	Pareto set and Pareto front of MOP (A.3).	83
A.4	Pareto set and Pareto front of MOP (A.4).	84
A.5	Pareto set and Pareto front of MOP (A.5).	85
A.6	Pareto set and Pareto front of MOP (A.6).	87

List of Tables

3.1	Δ_p values for the distances of the candidate solution set to $P_{Q,\epsilon}$	40
3.2	Δ_p values for the distances of the images of the candidate sets to $F(P_{Q,\epsilon})$. 40	
4.1	Δ_p values for the distances of the candidate solution set to $P_{Q,\epsilon}$	57
4.2	Δ_p values for the distances of the images of the candidate sets to $F(P_{Q,\epsilon})$. 57	
5.1	Δ_p values for the distances of the candidate solution set to $P_{Q,\epsilon}$ and $F(P_{Q,\epsilon})$ for MOP (A.1).	62
6.1	Δ_p values for the distances of the candidate solution set to $P_{Q,\epsilon}$ and $F(P_{Q,\epsilon})$ for refinement methods	74
6.2	Δ_p values for the distances of the candidate solution set to $P_{Q,\epsilon}$ and $F(P_{Q,\epsilon})$ for refinement methods	77

List of Algorithms

1	$P_{Q,\epsilon}$ -NSGA-II.	17
2	Simple Cell Mapping Algorithm.	30
3	Cell integer to cell coordinates	31
4	Cell coordinates to cell number.	31
5	Simple Cell Mapping for MOPs.	37
6	ArchiveUpdateTight2	51
7	Post-processing to get $P_{Q,\epsilon}$	52
8	Refine in Cell (rinc)	72
9	SCM-Refine	73
10	Moving Boxes	76
11	SCM Continuation	76

Chapter 1

Introduction

One is frequently faced with the problem of optimizing several objectives simultaneously and typically these objectives are in conflict with each other. Such objectives can be, for instance, the quality and the cost of a product. One would like to make a product, which has the highest quality, but at the same time the lowest cost. One can see that, since the objectives are in conflict in this example, there is not one single solution, but rather a set of them, which in this case represents the trade-off between quality and cost. Thus, in applications one is confronted with the problem of finding the best trade-off solutions for the given problems.

1.1 Motivation

So far, many numerical methods for the treatment of a given Multi-objective Optimization Problem (MOP) have been proposed. There exist, for instance, many scalarization methods that transform the MOP into a scalar optimization problem (SOP). By choosing a clever sequence of SOPs a finite size approximation of the entire Pareto set can be obtained in certain cases [1, 2, 3, 4]. Since the solution set (the so-called Pareto set) forms under some mild regularity conditions locally a $(k-1)$ -manifold, where k is the number of objectives involved in the MOP, specialized continuation methods, which perform a search along the Pareto set are very efficient if one solution is at hand [5, 6, 7, 8] and if the Pareto set is connected.

Another approach to approximate the Pareto set is to use set oriented methods such as subdivision techniques [9, 10, 11] or stochastic search methods [12, 13, 14]. The advantage of such set based methods is that they generate an approximation of the (global) Pareto set in one single run of the algorithm. Further, they are applicable to a large range of optimization problems and are characterized by a great amount of robustness. Hence, these methods are interesting alternatives against ‘classical’ mathematical programming techniques in particular for the thorough investigation of low or moderate dimensional MOPs. The method that is used in this study falls into the last category: Cell mapping techniques as proposed in [15] are coupled with dynamical systems derived from multi-objective descent directions that allow for the

computation of a suitable finite size approximation of the set of interest.

1.2 The Problem

The set of approximate solutions forms a n -dimensional object, where n is the number of parameters. For this reason it might be useful to have the global view of MOPs to see its behavior and then to compute the set of interest.

Some numerical schemes are proposed in [2, 16, 17, 18] to solve unconstrained MOPs. The general idea is starting from an initial solution to steer the search process in a desired direction given in objective space. It is important to mention that for these schemes the proposed process can be formulated as an initial value problem (IVP).

From that idea, if an IVP formulation is used to solve MOPs then we could make a match between Pareto optimal solutions and an attractor of this dynamical system. We would also be able to obtain the basin of attraction of every point of the Pareto set. However, to obtain the global view just by using this, we would have to follow the IVP for all points in parameter space, which is not possible in practice.

To solve this problem, let us point out that, since the representation of the numbers in a computer is finite, a number represents not only the number represented by its digits, but also an infinite neighborhood of numbers given by the precision of the machine. This does not allow to assume variables to be continuous, due to rounding errors and for this reason it is possible to consider the space as small hypercubes whose size is given by the machine precision.

The cell mapping approach [15] proposes to increase this discretization by dividing the state space in bigger hypercubes called cells. The evolution of the dynamical system is then reduced to a new function, which is defined not 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.

The simple cell mapping method gives us a useful tool to obtain the attractors and basins of attractions of a dynamical system. Thus, to extend this idea to the context of multi-objective optimization, in order to obtain the set of approximate solutions is an important contribution.

1.3 Objectives

General Objective

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

Particular Objectives

1. To develop a simple cell mapping method for the computation of optimal solutions of a given MOP.
2. To develop a simple cell mapping method for the computation of the set of approximate solutions of a given MOP.
3. To hybridize simple cell mapping with subdivision techniques.
4. To study the domains of attraction for each descent direction method.

1.4 Contributions

- Set oriented algorithms for global multi-objective optimization
 - Simple cell mapping for multi-objective optimization
 - Simple cell mapping for the set of approximate solutions
- Comparison between descent direction methods by means of simple cell mapping
- Collaboration with the University of California at Merced
- Contribution at EVOLVE 2013 international conference: C. Hernández, O. Schütze, J. Q. Sun. Computing the Set of Approximate Solutions of a Multi-Objective Optimization Problem by Means of Cell Mapping Techniques (published)
- Contribution at EVOLVE 2013 international conference: Y. Narajani, C. Hernández, F. R. Xiong, O. Schütze, J. Q. Sun. A Hybrid Algorithm for the simple cell mapping Method in Multi-objective Optimization (published)
- Contribution at ASME 2013 international conference: Y. Sardahi, Y. Narajani, W. Liang, J. Q. Sun, C. Hernández, O. Schütze. Multi-objective Optimal Control Design with the simple cell mapping Method (accepted)
- Contribution at CSTAM 2013 international conference: Y. Sardahi, Y. Narajani, W. Liang, F. R. Xiong, Z. C. Qin, Y. X., C. H., O. Schütze, J. Q. Sun. Multi-objective Optimal Design of Feedback Controls for Dynamical Systems Time Delay (submitted)
- Contribution at International Journal of Dynamics and Control: C. Hernández, Y. Narajani, Y. Sardahi, W. Liang, O. Schütze. simple cell mapping Method for Multiobjective Optimal PID Control Design (published)

1.5 Organization of the thesis

This thesis consists of seven chapters, including this introductory chapter. The remainder of this document is organized as follows:

Chapter 2 presents the basic concepts of multi-objective optimization and dynamical systems that are fundamental for understanding the current work. Further, we review some of the methods for solving a multi-objective optimization problem along with some methods for global analysis of dynamical systems.

Chapter 3 is devoted to present the simple cell mapping Method in the context of multi-objective optimization. In this chapter, we discuss the key elements to adapt this method and further on present numerical results on some academic models.

Chapter 4 describes the simple cell mapping for the set of approximate solutions. In this chapter, we present the elements that are incorporated to the method to compute this set and we also present numerical results on academic models.

Chapter 5 provides a comparison between some of the different descent directions in the literature. This is of particular interest to evolutionary algorithms, since a bias may exist on this methods.

Chapter 6 includes a hybridization of the simple cell mapping with other techniques to help to refine its results and to recover solutions that may be lost while performing the search.

Finally, chapter 7 contains the conclusions and some possible future ideas to be developed from this work.

Chapter 2

Background and Related Work

In this chapter, we look into the basic concepts that are needed for the understanding of this thesis work. We review the basic concepts of multi-objective optimization (Section 2.1) and dynamical systems (Section 2.3), which are the main topics related to this work. Further, we review some of the methods for solving a multi-objective optimization problem (Section 2.2) and also methods which provide a global analysis of dynamical systems (Section 2.4).

2.1 Multi-objective Optimization

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 better 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 section, we first define the multi-objective optimization problems (Section 2.1.1), then we define optimality for MOPs along with our sets of interest (Section 2.1.2). Next, we present the necessary optimality condition for MOPs (Section 2.1.3) and finally we introduce the concept of descent directions, which helps us to identify directions toward optimal solutions can be found (Section 2.1.4).

2.1.1 Formulation of the Problem

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

$$\begin{aligned} & \min_{x \in \mathbb{R}^n} \{F(x)\}, \\ & \text{s.t.} \\ & g_i(x) \leq 0, \quad i = 1, \dots, I, \\ & h_j(x) = 0, \quad j = 1, \dots, J, \end{aligned} \tag{2.1}$$

let $Q \subset \mathbb{R}^n$ be the feasible region defined by

$$Q = \{x \in \mathbb{R}^n | g(x) \leq 0 + h(x) = 0\}, \tag{2.2}$$

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

$$f_i : Q \rightarrow \mathbb{R}, \quad i = 1, \dots, k, \tag{2.3}$$

$x \in Q$ is known as a parameters vector, $g_i : \mathbb{R}^n \rightarrow \mathbb{R}$, $i = 1, \dots, I$ is an inequality constraint and $h_j : \mathbb{R}^n \rightarrow \mathbb{R}$, $j = 1, \dots, J$ is an equality constraint.

In case there are no constraints the MOP is known as 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 is the lower bound and ub_i is the upper bound for x_i , $i = 1, \dots, n$. We will also use the MOP as a minimization problem, which comes out without loss of generality.

2.1.2 Pareto Optimality

Now, we need to define what is an optimal solution in the context of multi-objective optimization. For the case $k = 1$ and given feasible points $a, b \in Q$ with their objective values $F(a), F(b) \in \mathbb{R}^k$, we would be able to say, that the one with the smallest objective value is better by doing a comparison of the values. However, for the case $k > 1$, it is not longer as simple as it was in the previous case, since doing the comparison might end up with a solution being better according to one objective but worse according to another one. To overcome this problem, we introduce the concept of Pareto dominance [19].

Definition 1 (Pareto Dominance) *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 $i \in 1, \dots, k$. The relation \leq_p is defined analogously.*

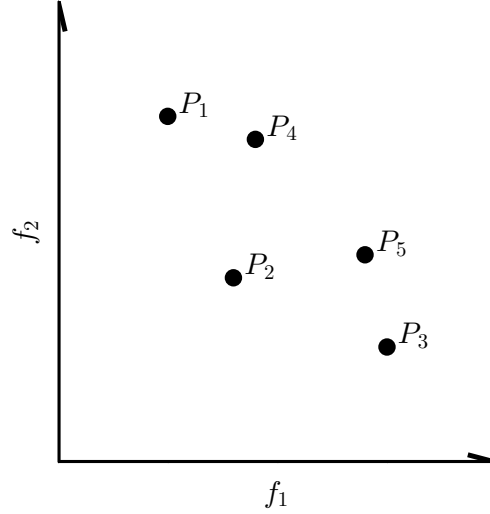


Figure 2.1: Point P_2 dominates the points P_4 and P_5 , as P_2 is better in both objectives. Points P_1 , P_2 and P_3 are mutually non-dominated as each of them is better in one objective but worse according to the another.

2. A vector $y \in Q$ is called dominated by a vector $x \in Q$ (denoted by $x \prec y$) with respect to Equation (2.1) if

$$F(x) \leq_p F(y) \text{ and } F(x) \neq F(y),$$

else y is called non-dominated by x .

Figure 2.1 shows an example of the Pareto dominance relationship.

The Pareto dominance defines a way to determine which of two solutions x and y is “better”. This can be used to define an optimal solution, also known as Pareto optimal solution.

Definition 2 (Pareto optimal solution) A point $x \in Q$ is called Pareto point of Equation (2.1) if there is no $y \in Q$ that dominates x .

Usually, we will find, that several solutions are non-dominated by any other solution in the feasible region. Thus, there exists an entire set of solutions. This set is the so-called Pareto set and its image it is known as the Pareto front.

Definition 3 (Pareto set and Pareto front) 1. The set of all Pareto optimal solutions is called the Pareto set, i.e.,

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

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

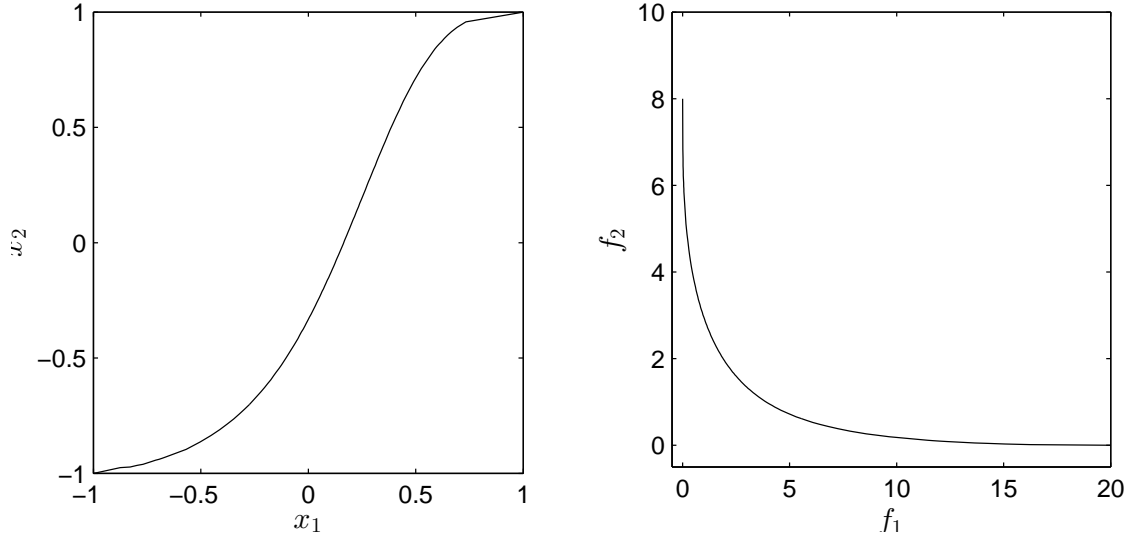


Figure 2.2: Pareto set (left) and Pareto front (right)

Figure 2.2 shows an example of a Pareto set and Pareto front for MOP (A.1) (compare also to Appendix A).

Typically, both Pareto set and Pareto front form $(k-1)$ -dimensional objects under some mild assumptions on Problem (2.1) [20].

We now define another notion of dominance which we use to define approximate solutions [21].

Definition 4 (ϵ -dominance) Let $\epsilon = (\epsilon_1, \dots, \epsilon_k) \in R_+^k$ and $x, y \in Q$.

1. x is said to ϵ -dominate y ($x \prec_\epsilon y$) with respect to Equation (2.1) if

$$F(x) - \epsilon \leq_p F(y) \text{ and } F(x) - \epsilon \neq F(y).$$

2. x is said to $-\epsilon$ -dominate y ($x \prec_{-\epsilon} y$) with respect to Equation (2.1) if

$$F(x) + \epsilon \leq_p F(y) \text{ and } F(x) + \epsilon \neq F(y).$$

Both notions of ϵ -dominance are equivalent. However, the definition of $-\epsilon$ -dominance help us to define our set of interest.

Definition 5 (Set of Approximate solutions $P_{Q,\epsilon}$) 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.,

$$P_{Q,\epsilon} = \{x \in Q \mid \nexists y \in Q : y \prec_{-\epsilon} x\}, \quad (2.6)$$

$F(P_{Q,\epsilon})$ is the image of $P_{Q,\epsilon}$.

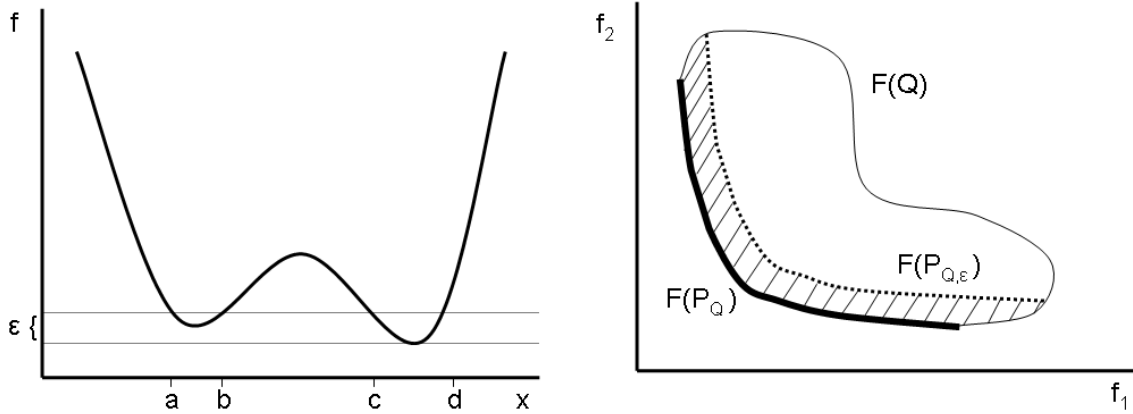
Figure 2.3: Two different examples for sets $P_{Q,\epsilon}$

Figure 2.3 shows two examples of the set of approximate solutions, given a value of ϵ . a) shows the case for $k = 1$ and in parameter space with $P_{Q,\epsilon} = [a, b] \cup [c, d]$. Note that the image solutions $f([a, b])$ are nearly optimal (measured in objective space), but that the entire interval $[a, b]$ is not ‘near’ to the optimal solution which is located within $[c, d]$. At the right, we show an example for $k = 2$ in image space, $F(P_{Q,\epsilon})$ is the approximate Pareto front (taken from [22]).

2.1.3 Optimality conditions

If all the objectives in an unconstrained MOP are differentiable, the Theorem of Kuhn and Tucker [23] states a necessary condition for Pareto optimality.

Theorem 1 (Karush-Kuhn-Tucker equation) *Let x^* be a Pareto point of Equation (2.1), then there exists a vector $\alpha \in \mathbb{R}^k$ such that*

$$\begin{aligned} \sum_{i=1}^k \alpha_i \nabla f_i(x^*) &= 0 \\ \text{s.t.} \\ \alpha_i &\geq 0, i = 1, \dots, k, \\ \sum_{i=1}^k \alpha_i &= 1, \end{aligned} \tag{2.7}$$

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

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

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

2.1.4 Descent directions

Now, we introduce the concept of descent directions, which is of high interest for this thesis.

A vector $\nu \in \mathbb{R}^n$ is called a descent direction if a search in this direction leads to an improvement of all objective values. To be more precise, ν is a descent direction of Equation (2.1) at a point $x \in \mathbb{R}^n$ if there exists a $\bar{t} \in \mathbb{R}_+$ such that

$$F(x + t\nu) <_p F(x), \quad \forall t \in (0, \bar{t}). \quad (2.9)$$

We highlight, that this descent direction it is not unique, since several vector ν may fulfill the requirement for a descent direction. The set of descent directions also known as descent cone, this set is defined for a point $x_0 \in Q$ as follows

$$D(x_0) = \{\nu \in \mathbb{R}^n \setminus \{0\} \mid \langle \nabla f_i(x_0), \nu \rangle \leq 0 \forall i = 1, \dots, k\}. \quad (2.10)$$

2.2 Solving MOPs

The concepts from the last section help us to understand the problem at hand. Having this in mind, we turn our attention to the approaches that have been developed to solve it. In general these methods provide a finite approximation of the Pareto front and they have two main goals. The first one is to converge to the real Pareto front i.e. that the finite approximation is distributed along the Pareto set/front. The second one is to have a good spread of the solutions i.e. that the distance between solutions of the approximation of the Pareto front is ideally the same.

In this section, we review some of these methods. We divide them into three categories, scalarization methods (Section 2.2.1), descent direction methods (Section 2.2.2) and stochastic methods (Section 2.2.3). Although the descent direction methods can be seen as scalarization methods, we highlight them here since they are crucial for this thesis.

2.2.1 Scalarization methods

One of the ideas to solve a 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 SOP that have been proposed. However, typically the solution of a SOP consists of only one point, while the solution of a MOP is a set. Thus, the Pareto set can be approximated (in some cases not entirely) by solving a clever sequence of SOPs [20].

In the following, we shortly review the most widely used scalarization techniques for a more thorough discussion we refer to [20].

Weighted sum method

The weighted sum method [20] 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 Equation (2.1), the weighted sum problem can be stated as follows:

$$\begin{aligned} \min \quad & f_\alpha(x) := \sum_{i=1}^k \alpha_i f_i(x) \\ \text{s.t.} \quad & x \in Q, \\ & \alpha_i \geq 0, i = 1, \dots, k, \\ & \sum_{i=1}^k \alpha_i = 1. \end{aligned} \tag{2.11}$$

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 $\alpha_i > 0$, $i = 1, \dots, k$, then a solution of Equation (2.11) is Pareto optimal.*

On the other hand, the proper choice of α , though it appears to be intuitive at first sight. Actually, it is in certain cases a delicate problem. Further, the images of (global) solutions of Equation (2.11) 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.

ϵ - constrained method

The idea of the ϵ -constraint method [24] is to select one objective f_i , $i \in \{1, \dots, k\}$, and to treat all the others as constraints by imposing upper bounds on the function values. This leads to the following optimization problem:

$$\begin{aligned} \min_{x \in Q} \quad & f_i(x) \\ \text{s.t.} \quad & f_j(x) \leq \epsilon_j \quad \forall j \in \{1, \dots, k\} \setminus \{i\}. \end{aligned} \tag{2.12}$$

Theorem 3 *A vector $x^* \in Q$ is Pareto optimal if and only if it is a solution of Equation (2.12) for every $i = 1, \dots, k$, where $\epsilon_j = f_j(x^*)$ for $j \in \{1, \dots, k\} \setminus \{i\}$.*

Hence, using the ϵ -constraint method, it is possible to find every Pareto optimal solution regardless of the form of the Pareto front. However, the choice of ϵ is not an easy task.

Weighted Tchebycheff method

The aim of the weighted Tchebycheff method [20] 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 $\alpha_i \geq 0$, $i = 1, \dots, k$, and $\sum_{i=1}^k \alpha_i = 1$, and let $Z = (z_1, \dots, z_k)$, then the weighted Tchebycheff method reads as follows:

$$\min_{x \in Q} \max_{i=1, \dots, k} \alpha_i |f_i(x) - z_i|, \quad (2.13)$$

note that the solution of Equation (2.13) 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 4 *Let $x^* \in Q$ be Pareto optimal. Then there exists $\alpha \in \mathbb{R}_+^k$ such that x^* is a solution of Equation (2.13), where Z is chosen as the utopian vector of the MOP.*

The **utopian vector** $F^* = (f_1^*, \dots, f_k^*)$ of a 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

The Normal boundary intersection (NBI) method [1] 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 , $i = 1, \dots, k$ (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

$$\begin{aligned} \max_{x, l} \quad & l \\ \text{s.t.} \quad & F(x_0) + l\alpha = F(x) \\ & x \in Q. \end{aligned} \quad (2.14)$$

Equation (2.14) 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 Equation (2.14) do not have to be Pareto optimal [1].

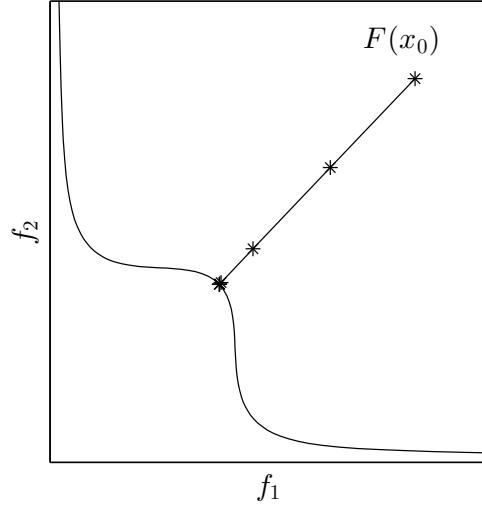


Figure 2.4: Curve of dominated points.

2.2.2 Descent direction methods

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 a line search, i.e., by setting

$$x_{\text{new}} = x + t\nu, \quad (2.15)$$

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 2.4 shows an example of a dominated curve.

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

Simplest bi-objective descent direction

One way to combine two gradients to obtain a descent direction is by a vector sum.

Theorem 5 (Simplest bi-objective descent direction [18]) *Let $x \in \mathbb{R}^n$, and $f_1, f_2 : \mathbb{R}^n \rightarrow \mathbb{R}$ define a two-objective MOP. if $\nabla f_i(x) \neq 0$, for $i = 1, 2$, then the direction*

$$\nu(x_0) = - \left(\frac{\nabla f_1(x)}{\|\nabla f_1(x)\|} + \frac{\nabla f_2(x)}{\|\nabla f_2(x)\|} \right), \quad (2.16)$$

is a descent direction at x_0 of MOP.

However, this approach cannot be generalized for more than two objective functions.

Directed search

The directed search method [17] allows to steer the search from a given point $x \in Q$ into a desired direction $d \in \mathbb{R}^k$. A direction vector $\nu \in \mathbb{R}^n$ can be computed such that

$$\lim_{t \rightarrow 0} \frac{f_i(x_0 + t\nu) - f_i(x_0)}{t} = d_i, \quad i = 1, \dots, k. \quad (2.17)$$

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 . 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 [5, 6].

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

$$J(x)\nu = d,$$

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

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

With this the authors propose that ν 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 underdetermined, which implies that its solution is not unique. To deal with this, the problem can be formulated as

$$\nu = J(x_0)^+ d,$$

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):

$$\begin{aligned} x(0) &= x_0 \in \mathbb{R}^n \\ \dot{x}(m) &= \nu_\alpha(x(m)), \quad t > 0. \end{aligned} \quad (2.19)$$

Continuation method Once an optimal point has been found 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

$$\sum_{i=1}^k \alpha_i \nabla f_i(x) = 0 \quad (2.20)$$

and further we assume that

$$\text{rank}(J(x)) = k - 1. \quad (2.21)$$

It is known (e.g., [6]) that in this case α is orthogonal to the Pareto front, i.e.,

$$\alpha \perp T_y \delta F(\mathbb{R}^n), \quad (2.22)$$

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.,

$$\alpha = QR, \quad (2.23)$$

where $Q = (q_1, \dots, q_k) \in \mathbb{R}^{k \times k}$ is an orthogonal matrix and $i = 1, \dots, k$ its column vectors, and $R = (r_{11}, 0, \dots, 0)^T \in \mathbb{R}^{k \times 1}$ with $r_{11} \in \mathbb{R} \setminus \{0\}$. Since by Equation (2.23) $\alpha = r_{11}q_1$ and Q orthogonal, it follows that the column vectors q_2, \dots, 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

$$J(x)\nu_i = q_i, \quad i = 2, \dots, k. \quad (2.24)$$

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, \dots, q_k are in the image of $J(x)$, i.e., Equation (2.24) can be solved for each $i \in \{2, \dots, k\}$. Then, the following can be chosen as the set of predictor direction:

$$p_i = x_0 + t\nu_i. \quad (2.25)$$

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 Equation (2.19) 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

The following function is defined [16]:

$$q(x) = \sum_{i=1}^k \hat{a} \nabla f_i(x),$$

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

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

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

From this we have that either $q(x) = 0$ or $-q(x)$ is a descent direction for all the objective functions; hence, each x with $q(x) = 0$ fulfills the first-order necessary conditions for Pareto optimality.

Method of Fliege and Svaiter

The following function is defined [2]:

$$f_x(\nu) = \max(A\nu)_i, i = 1, \dots, k,$$

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

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

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.

2.2.3 Stochastic 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 their 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 Pareto set and the Pareto front in one execution. Examples of these methods can be found in [12, 13]. One drawback is that MOEAs do not guarantee convergence towards the Pareto front. In the following we review two of the most important stochastic methods to solve a MOP and also an approach to find the set of approximate solutions $P_{Q,\epsilon}$.

Nondominated Sorting Genetic Algorithm II

The Nondominated Sorting Genetic Algorithm II (NSGA-II) was proposed in [25]. The NSGA-II has two main mechanisms. The first one is the non-dominated sort, 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 these mechanisms it has a good overall performance and it has become a prominent method to the point. It is one of the methods against others have to be compared, at least if two and three objectives are considered.

MOEA based on decomposition

Another prominent algorithm is called MOEA based on decomposition (MOEA/D) which was proposed in [26]. The main idea of this method is to make a decomposition of the original MOP into N 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 NBI.

As it is the case of the NSGA-II, this method also gives good results in solving MOPs and has become one of the most popular algorithms. Also this is one of the methods against others have to be compared.

Evolutionary computation methods for $P_{Q,\epsilon}$

The methods described above aim for the approximation of the Pareto set and Pareto front, but now the goal is to compute the set of approximate solutions.

The method $P_{Q,\epsilon}$ -NSGA-II, is a hybrid of NSGA-II and the archiver ArchiveUpdate $P_{Q,\epsilon}$ [27]. The idea is to use the archiver in each iteration to keep the approximate solutions found so far and to use them for the mechanisms of NSGA-II. Algorithm 1 describes the basic steps of this method.

Algorithm 1 $P_{Q,\epsilon}$ -NSGA-II.

Require: archive \mathbb{A}_l , $\Delta \in \mathbb{R}_+$, *population_size*, n_p , *maxiter*

for $i = 1$ **to** *maxiter* **do**

Ensure: updated archive \mathbb{A}_{l+1}

$C \leftarrow \text{Select}(\mathbb{A}_l, n_p/2)$

$C' \leftarrow \text{DominationSort}(C)$

$P \leftarrow \text{TournamentSelection}(C')$

$O \leftarrow \text{GeneticOperator}(P)$

$\mathbb{A}_{l+1} \leftarrow \text{ArchiveUpdatePQ}, \epsilon(\mathbb{A}_l, O, \Delta)$

end for

2.2.4 Performance Measures

As many methods have been proposed to solve MOPs, there is the need to compare the quality of their outcome. However, this is also not an easy task, since the comparison of sets in this context is not clearly defined [28]. Thus, different approaches can be used to compare the solutions.

In this section, we present some of performance measures that can be used to compare solutions from different methods.

In the following, we would like to measure the quality of a given candidate set $A = \{a_1, \dots, a_N\}$ in objective space to a discretized Pareto front $F(P_Q) = \{y_1, \dots, y_M\}$.

Generational Distance

The generational distance (GD) is used to measure the distance from the candidate set A to the Pareto front $F(P_Q)$ [29]. The lower value of GD the better is the candidate set A . A value of zero means, $A \in F(P_Q)$. However, if an element an element of the set is duplicated then the indicator gives a value closer to zero.

$$\text{GD}(A, F(P_Q)) = \frac{1}{|A|} \left(\sum_{i=1}^{|A|} \text{dist}_2(a_i, F(P_Q))^p \right)^{\frac{1}{p}}. \quad (2.26)$$

To avoid the potential drawback a slight modification was proposed [30].

$$\text{GD}(A, F(P_Q))_p = \left(\frac{1}{|A|} \sum_{i=1}^{|A|} \text{dist}_2(a_i, F(P_Q))^p \right)^{\frac{1}{p}}. \quad (2.27)$$

Inverted Generational Distance

The inverted generational distance (IGD) measures the distance from the Pareto front $F(P_Q)$ to the candidate set A [31]. 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.

$$\text{IGD}(A, F(P_Q)) = \frac{1}{|F(P_Q)|} \left(\sum_{i=1}^{|F(P_Q)|} \text{dist}_2(y_i, A)^p \right)^{\frac{1}{p}}. \quad (2.28)$$

To avoid the potential drawback a slight modification was proposed [30].

$$\text{IGD}(A, F(P_Q))_p = \left(\frac{1}{|F(P_Q)|} \sum_{i=1}^{|F(P_Q)|} \text{dist}_2(y_i, A)^p \right)^{\frac{1}{p}}. \quad (2.29)$$

Delta p

Now, we present the Hausdorff distance d_H , which is used to measure the distance between sets. Then, we introduce the indicator Δ_p , which is used to make our comparisons in this thesis work.

Definition 6 (Distance between sets [32]) *Let $u, v \in \mathbb{R}^n$ and $A, B \subset \mathbb{R}^n$. The maximum norm distance d_∞ , the semi-distance $\text{dist}(\cdot, \cdot)$ and the Hausdorff distance $d_H(\cdot, \cdot)$ are defined as follows:*

1. $d_\infty = \max_{i=1, \dots, n} |u_i - v_i|$

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.

Note that for $p = \infty$, we have $\Delta_\infty = d_H$, and for finite values of p the distances in Δ_p are averaged. In [30], Δ_p is discussed as a performance indicator in context of Pareto front approximations. In that case, the indicator can be viewed as a combination of the slight variations of the GD and the IGD.

2.3 Dynamical Systems

In this section, we first define a dynamical system (Section 2.3.1), and further on the solution of a dynamical system (Section 2.3.2 and Section 2.3.3). We also present the concept of the domain of attraction (Section 2.3.4) and finally we look into methods that were proposed to perform a global analysis of a given dynamical system (Section 2.4).

2.3.1 Formulation of the problem

A dynamical system [15] 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 $G : \mathbb{R}^N \rightarrow \mathbb{R}^N$ is, in general, a nonlinear vector function. The evolution of such a dynamical system can be described by a function of the form:

$$x_{m+1} = G(x(m), \mu), \tag{2.30}$$

where x is a N -dimensional vector, m denotes the mapping step, μ is a parameter vector, and G is a general nonlinear vector function. In this case ordinary differential equations can be used to describe the dynamical systems. These are defined as follows:

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

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 .

2.3.2 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 (2.30).

2.3.3 Periodic group

A periodic solution of Equation (2.30) of period l is a sequence of l distinct points $x^*(j)$, $j = 1, 2, \dots, l$ such that

$$\begin{aligned} x^*(o+1) &= G^o(x^*(1), \mu), \quad o = 1, 2, \dots, l-1, \\ x^*(1) &= G^l(x^*(1), \mu). \end{aligned} \tag{2.31}$$

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

2.3.4 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.

2.4 Global Analysis of Dynamical Systems

In this section, we review methods that are used to compute the global properties of a dynamical system. We present the simple cell mapping [15], which is useful to compute global attractors and domains of attraction of a given dynamical system. The other methods that we present are the subdivision techniques [9], which are useful to compute the global attractors of a given dynamical system.

2.4.1 Discretization of the space

Now, we do not consider the state space as a continuum but rather as a collection of state cells, with each cell being taken as a state entity. Because of this, now we need to introduce some basic concepts regarding the new model.

Cell state space

An N dimensional cell space S [15] 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 state space is to construct a cell structure consisting of rectangular parallelepipeds of uniform size.

Cell functions

Let S be the cell state 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)

$$z(n+1) = C(z(n), \mu), z \in \mathbb{Z}^N, \mu \in \mathbb{R}^l, \quad (2.32)$$

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

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 identity mapping. A sequence of l distinct cells $z^*(j), j \in l$, which satisfies

$$z^*(m+1) = C^m(z^*(1)), m \in l-1, z^*(1) = C^l(z^*(1)), \quad (2.33)$$

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 $l = 1$ periodic group.

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 $r \rightarrow \infty$.

2.4.2 Simple cell mapping

The main idea of this method is based on the fact that the representation of the numbers in a computer is finite. A number does not only represent the number represented by its digits, but also an infinite neighborhood of numbers given by the precision of the machine. This does not allow to assume variables to be continuous, due to rounding errors and for this reason it is possible to consider the space as small hypercubes whose size is given by the machine precision.

The cell mapping approach [15] proposes to increase this discretization by dividing the state space into bigger hypercubes. The evolution of the dynamical system is then reduced to a new function, which is not defined in \mathbb{R}^n , but rather 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.

Figure 2.5 shows the mapping for a group motion. The group motion has a 12 periodic group.

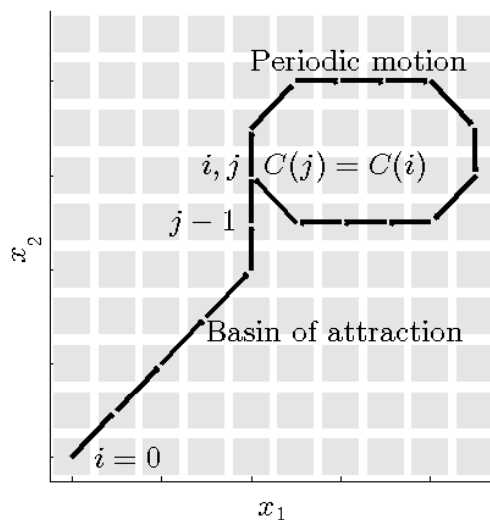


Figure 2.5: Group motion with a 12 periodic group. The arrows represent the mapping C from one cell to another. $C(i) = C(j)$ is the mapping where the periodic group is detected.

The cell mapping methods have been applied to optimal control problems of deterministic and stochastic dynamic systems [33, 34, 35]. Other interesting applications of the cell mapping methods include optimal space craft momentum unloading [36], single and multiple manipulators of robots [37], optimum trajectory planning in robotic systems by [38], and tracking control of the read-write head of computer hard disks [39].

Now, we present a small example of the SCM. We consider the following SOP:

$$\min_x f = 4x^3 - 2x, \quad (2.34)$$

where $f \in \mathbb{R}$ and $x \in \mathbb{R}$. For this problem, we have two optimal points at $\frac{\sqrt{2}}{2}$ and $-\frac{\sqrt{2}}{2}$. Figure 2.6 shows the result for different values of N and $Q = [-3, 3]$. The figure shows the mapping from one cell to another until it reaches a periodic group. Further, it shows two different group motions and for $N = 14$ and $N = 28$, we can

also see where the domain of attraction of $\frac{\sqrt{2}}{2}$ ends and the domain of attraction of $-\frac{\sqrt{2}}{2}$ begins.

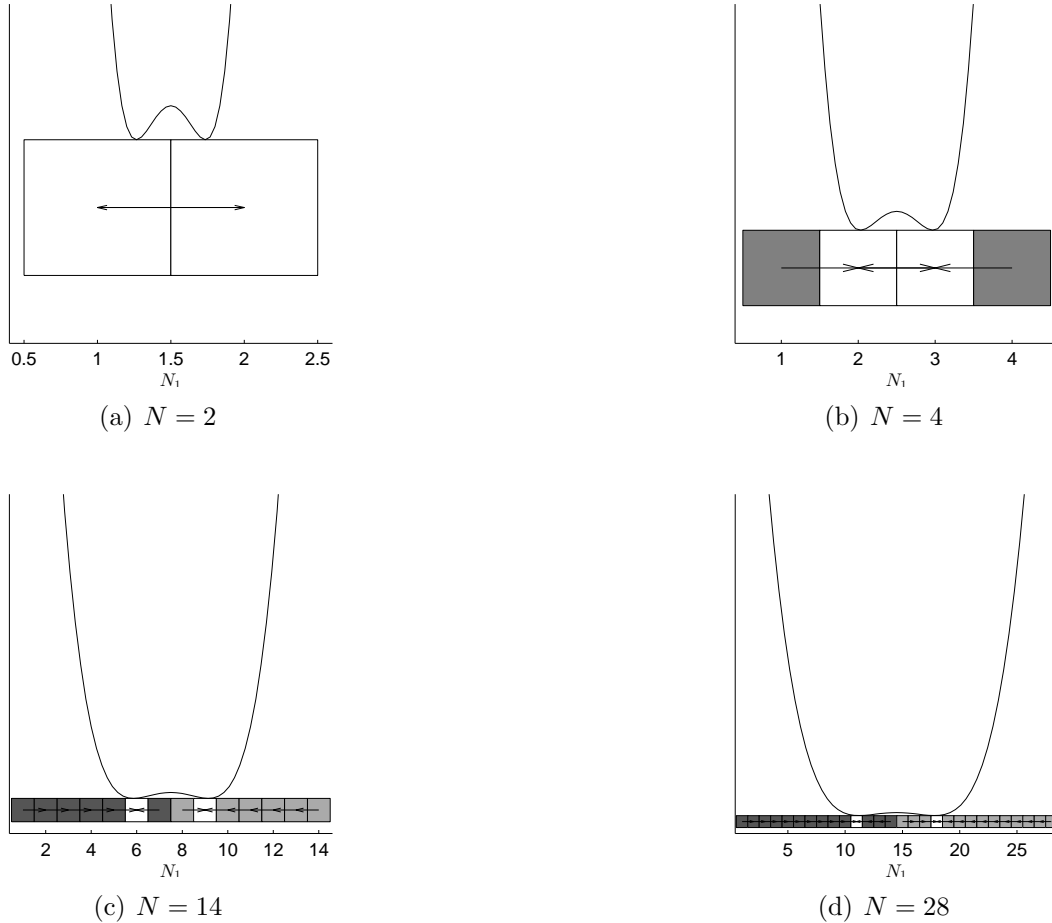


Figure 2.6: Numerical result of the SCM on Equation (2.34) with different grid size, (a) $N = 2$; (b) $N = 4$; (c) $N = 14$; (d) $N = 28$. The white cells represents the optimal solutions. The cells with the same color belong to the same domain of attraction (for those cells their mapping end in the same cell). The arrows represent the cell mapping. Finally, the black curve is the graphic representation of the problem.

2.4.3 Subdivision techniques

The subdivision techniques [9] are based on a generation of a collection of boxes in parameter space, which by the end of the execution covers the periodic groups of a given dynamical system. Let \mathbb{B}_0 be an initial collection of finitely many subsets of the compact set Q such that $\cup_{B \in \mathbb{B}_0} B = Q$. Then \mathbb{B}_k is obtained from \mathbb{B}_{k-1} in the following two steps:

1. Subdivision: Construct from \mathbb{B}_{k-1} a new system $\hat{\mathbb{B}}_k$ of subsets such that

$$\bigcup_{B \in \hat{\mathbb{B}}_k} B = \bigcup_{B \in \mathbb{B}_{k-1}} B$$

and

$$\text{diam}(\hat{\mathbb{B}}_k) = \theta_k \text{diam}(\mathbb{B}_{k-1}),$$

where $0 < \theta_{\min} \leq \theta_k \leq \theta_{\max} < 1$.

2. Selection: Define the new collection \mathbb{B}_k by

$$\mathbb{B}_k = \{B \in \hat{\mathbb{B}}_k : \text{there exists } \hat{B} \in \hat{\mathbb{B}}_k \text{ such that } C^{-1}(B) \cap \hat{B} \neq \emptyset\}.$$

These methods converge to the periodic group of one (or several) dynamical system(s) in the Hausdorff sense [40]. In the context of optimization this can be the set of global minimizers [40, 10, 9].

Now, we present a small example of the subdivision techniques on Equation (2.34). We have also chosen $Q = [-3, 3]$. Figure 2.7 shows the boxes that contain the periodic groups for different iterations.

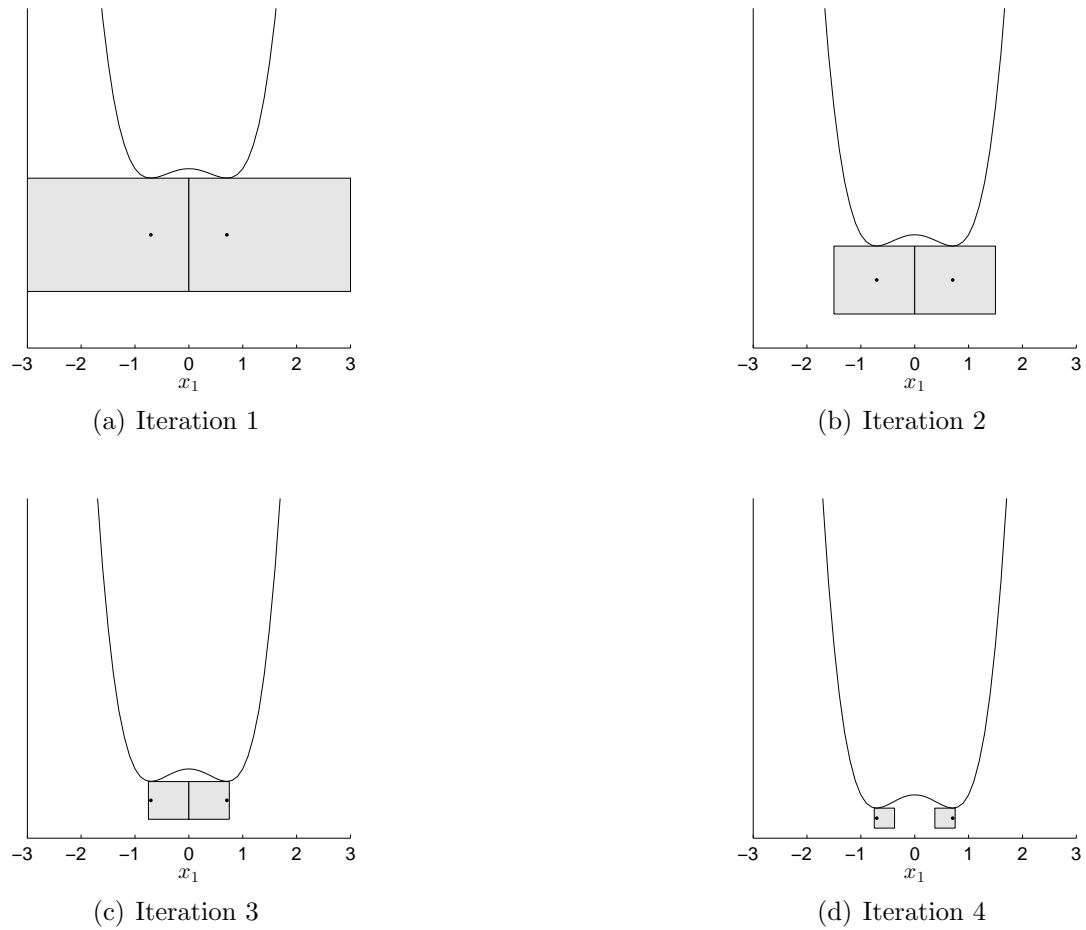


Figure 2.7: Numerical result of the subdivision algorithm on Equation (2.34), (a) Iteration 1; (b) Iteration 2; (c) Iteration 3; (d) Iteration 4. The black curve is the graphic representation the problem.

Chapter 3

Simple Cell Mapping for Multi-Objective Optimization

In this chapter, we first describe the simple cell mapping method for general dynamical systems (Section 3.1). We further adapt the simple cell mapping method to the context of multi-objective optimization (Section 3.2) and finally we present numerical results on academic models and a comparison with NSGA-II and MOEA/D (Section 3.4).

3.1 Simple Cell Mapping

In general, when we solve a problem, we do not obtain an exact result. This is either due to the limited machine precision (roundoff errors) or by a limited measurement accuracy while doing an experiment.

This is the main idea behind the SCM method [15]. Since these problems are unavoidable, the method proposes to increase this error by a given discretization and assumes every value as a discrete quantity.

The SCM method is an useful tool to do a global analysis of a given dynamical system. In this section we present the classical SCM method and describe some of the key elements used by the method to capture the global properties of a dynamical system. We also provide an analysis of the simple cell mapping method based on graph theory.

3.1.1 Description of the SCM Method

The SCM method uses some sets in order to capture the global properties of a cell, which we describe in the following:

- 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.

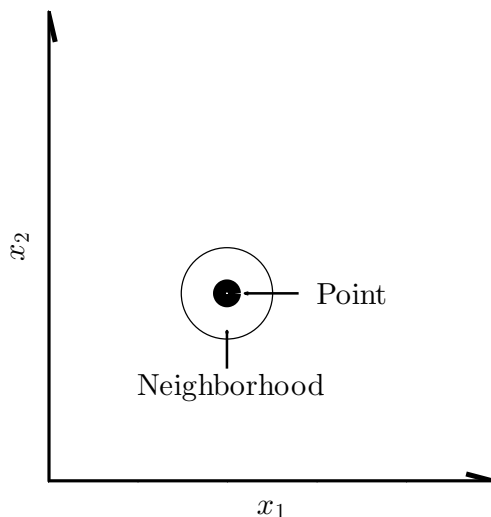


Figure 3.1: Shows a point x_0 and the neighborhood of numbers that are represented by x_0 .

- 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 this cell into a periodic cell.

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, 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, 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, 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.

Algorithm 2 shows the key elements of the SCM method.

3.1.2 Center point method

Now, we need to choose a point to represent the whole cell, for this we have picked the center point method because of its simplicity and its good results if we have a

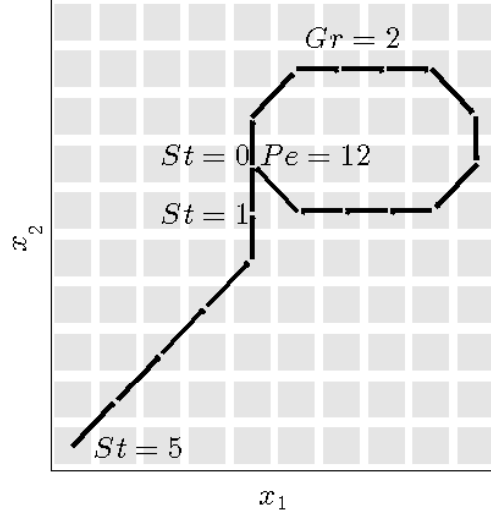


Figure 3.2: Shows a periodic motion along with its global properties.

fine grid. This method is as follows, given a cell $pcell \in \mathbb{Z}$, we compute its center $pcell_center \in \mathbb{R}^n$.

When a cell is picked by the SCM method to start the process, a cell number is given and we need to convert it into its cell center in order to start following the dynamical system.

The first step is to convert the cell number into its cell coordinates $\in \mathbb{Z}^n$ of the cell. If we have the same number of divisions for each dimension i.e. $N_i = b, \forall i \in n$, then this problem can be seen as a numerical base conversion (e.g. convert from decimal to binary), where the base is given by N . This problem can be solved by means of successive divisions.

Now, in order to generalize this process, we can use the same algorithm but in each step, we change the base, which in this case is given by N_i . Algorithm 3 shows the realization of this idea.

The next step is to take the cell coordinates and output the center of the cell. This is given by the following expression:

$$pcell_center_i = lb_i + h_i z_i - \frac{1}{2} h_i, \quad i = 1, \dots, n, \quad (3.1)$$

where lb is a vector with the lower bound and h is a vector with the size of the cell for each dimension.

Once we have followed from one cell to another one, we need to compute the cell number of the new cell, which represents the inverse process of the algorithm described above. Given a point $\in \mathbb{R}^n$ identify in which cell is located. Given the cell center $\in \mathbb{R}^n$, the cell coordinates $\in \mathbb{Z}^n$ are given by the following expression:

$$pcell_i = round((cell_center_i - lb_i)/h_i + 1/2), \quad i = 1 \dots, n. \quad (3.2)$$

Algorithm 2 Simple Cell Mapping Algorithm.

Require: $DynamicalSystem, F, ub, lb, N, h, N_c$

Ensure: z, C, Gr, Pe, St

```

1:  $current\_group \leftarrow 1$ 
2:  $Gr(i) \leftarrow 0, \forall i \in N_c$ 
3: for all  $p_{cell} \in N_c$  do
4:    $cell \leftarrow p_{cell}$ 
5:    $i \leftarrow 0$ 
6:   while  $newcell = \mathbf{true}$  do
7:      $x_i \leftarrow$  center point of  $cell$ 
8:     if  $Gr(cell) = 0$  then
9:        $p_{i+1} \leftarrow DynamicalSystem(x_i)$ 
10:       $ncell \leftarrow$  cell where  $p_{i+1}$  is located
11:       $C(cell) \leftarrow ncell$ 
12:       $cell \leftarrow ncell$ 
13:       $i \leftarrow i + 1$ 
14:    end if
15:    if  $Gr(cell) > 0$  then
16:       $Gr(C^j(p_{cell})) \leftarrow Gr(cell), j \leftarrow 0, \dots, i$ 
17:       $Pe(C^j(p_{cell})) \leftarrow Pe(cell), j \leftarrow 0, \dots, i$ 
18:       $St(C^j(p_{cell})) \leftarrow St(cell) + i - j, j \leftarrow 0, \dots, i$ 
19:       $cell \leftarrow C(cell)$ 
20:       $newcell \leftarrow \mathbf{false}$ 
21:    end if
22:    if  $Gr(cell) = -1$  then
23:       $current\_group \leftarrow current\_group + 1$ 
24:       $Gr(C^k(p_{cell})) \leftarrow current\_group, k \leftarrow 0, \dots, i$ 
25:       $j \leftarrow i^{th}$  value when period appears
26:       $Pe(C^k(p_{cell})) \leftarrow i - j, k \leftarrow 0, \dots, i$ 
27:       $St(C^k(p_{cell})) \leftarrow j - k, k \leftarrow 0, \dots, j - 1$ 
28:       $St(C^k(p_{cell})) \leftarrow 0, k \leftarrow j, \dots, i$ 
29:       $cell \leftarrow C(cell)$ 
30:       $newcell \leftarrow \mathbf{false}$ 
31:    end if
32:  end while
33: end for

```

Next, given the cell coordinates $\in \mathbb{Z}^n$, we compute the cell number $\in \mathbb{Z}$ based in the evaluation of a polynomial. Algorithm 4 shows such a realization. Although the center point method is rather simple, it is an effective way to represent a cell, however, if the cells are too big the center point might no longer be representative of the dynamics of the whole cell.

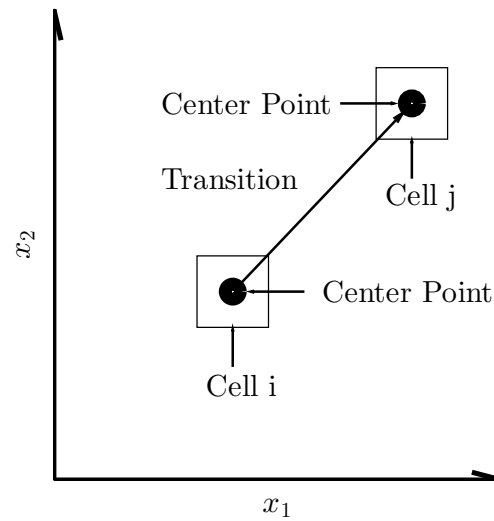


Figure 3.3: Shows how the mapping is done from the center of $cell_i$ to $cell_j$.

Algorithm 3 Cell integer to cell coordinates

Require: $cell, N$

Ensure: z

- 1: $cell \leftarrow cell - 1$
 - 2: **for** $i \leftarrow 1 : length(N)$ **do**
 - 3: $coord(i) \leftarrow rem(cell, N(i)) + 1$
 - 4: $cell \leftarrow fix(cell/N(i))$
 - 5: **end for**
 - 6: $z \leftarrow coord$
-

Algorithm 4 Cell coordinates to cell number.

Require: z, N

Ensure: $ncell$

- 1: $z \leftarrow z - 1$
 - 2: $ncell \leftarrow z(1)$
 - 3: $b \leftarrow N(1)$
 - 4: **for** $i \leftarrow 2 : length(N)$ **do**
 - 5: $ncell \leftarrow ncell + z(i) * b$
 - 6: $b \leftarrow b * N(i)$
 - 7: **end for**
 - 8: $ncell \leftarrow ncell + 1$
-

3.1.3 Analysis of SCM

The SCM method can be viewed as a graph problem $g = (x, e)$, where x is the set of vertex of the graph and e is the set of edges. In the following we describe the SCM as a graph problem.

1. Create graph: The first step consists of creating a graph g . The vector x is equal to the set S of cells. The vector e is the mapping of a cell i to a cell j , i.e. for a cell i compute its center point and then follow the dynamical system to a cell j . Which is the representation of the transition between vertexes.
2. Find cycles: The second step is to use the graph g to compute the global properties of the dynamical system. For this we could use either breadth search first or deep search first (both methods are equivalent in this case, since we only have transition for each vertex) and these methods can also give us the size of the cycles if we use the colored version of the algorithms.

This view of the SCM method helps us to study the method in terms of space and time complexity. In the following, we state some of the more important properties of the method.

- The out-degree [41] of the graph g is 1.
- The generation of the graph is done in $O(|x|)$, since for each vertex we only have one edge going out.
- The search for cycles is done in $O(|x| + |e|)$, since each vertex and edge is visited once.
- If we use an adjacency list representation for the graph g , $O(|x| + |e|)$ is the memory needed to store the graph.
- The number of vertexes x is given by $N_1 \times N_2 \times \dots \times N_n$, which is exponential in the number of dimensions. This is the weakest point of the SCM method, thus the method is restricted to low dimensional problems.

3.2 Adapting the SCM Method for MOPs

So far, the SCM method is designed for general dynamical systems as shown in Algorithm 2. In order to apply it to the context of multi-objective optimization, we have to define a suitable dynamical system. For this, we have chosen to take models that are derived from *descent direction methods*.

3.2.1 Dynamical system

In this section, we have picked the descent direction from [18] to be used with the SCM method. Using Equation (2.16), the dynamical system

$$\dot{x}(t) = v(x(t)) \quad (3.3)$$

can now be used since it defines a pressure toward the Pareto set/front of the MOP at hand. The other descent direction methods described previously can be used in a similar form.

3.2.2 Step size control

Once we have decided which descent direction ν to use, we are left with the problem of choosing an appropriate step length t .

Typically, the computation of the step size is not easy as we have two conflicting wishes. First, we would like to choose a step size t that lowers the objective function as much as possible with a given descent direction ν . Ideally, we would like to have the global minimizer of the function

$$\phi(t) = F(x + t\nu), \quad t > 0, \quad (3.4)$$

however, doing this we would spend a lot of function evaluation solving this problem and this is in contrast to the second wish, which is to make this decision as cheap as possible in terms of time and number of function evaluations.

As an alternative to the global minimizer of the function ϕ , we can use an inexact step size control as the ones proposed in [2, 42].

In the particular case of the SCM for multi-objective optimization, it has several advantages for solving the step size control problem since it has more information at hand. For instance, we have the size of the cell which is given by h and we also have that we start at the center of each cell. With this information, we already have a value for sufficient decrease. If there exists a $t\nu_i \geq \frac{h_i}{2}$, $i = 1, \dots, n$, then we ensure that we leave the current cell, which it is what the SCM method needs to keep working.

Now, to decide if the step size t is accepted, we can use a dominance test, if the image of the new point $F(x_{i+1}) \prec F(x_i)$ then we accept the step size t else we can use backtracking until we find an appropriate step size or the step size is so small that it would be enough to leave the current cell.

We are left with the choice of the initial step size t_0 . For this we could choose an arbitrary initial value and if this value is not accepted, we could rely in the backtracking to help us find an appropriate value for t_0 . However we can also compute the nearest neighbor given the descent direction ν from the current cell center. This idea is shown in Equation (3.5).

$$t_0 = \max \left(\frac{h_i}{\nu_i} \right) + \epsilon, \quad \forall i | \nu_i \neq 0. \quad (3.5)$$

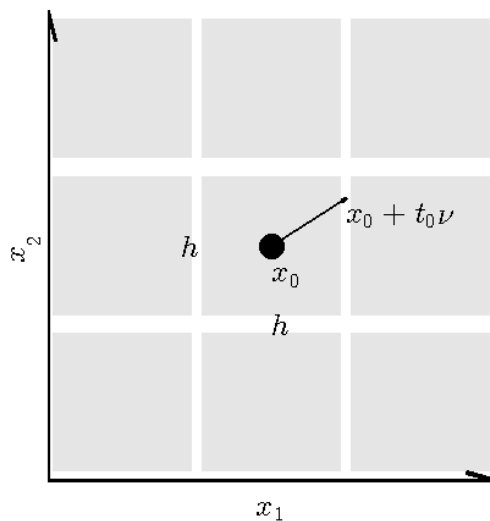


Figure 3.4: Shows the setting of the step size control problem for the SCM method.

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 neighbor cell, 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.

3.2.3 Finding Pareto optimal solutions

In order to be able to store those cells that are candidates to be Pareto optimal (to be more precise, cells that potentially contain a part of the Pareto set), we use a set called *cPs*.

As stated before the SCM method is useful to compute the global attractors and the domain of attraction of the given dynamical system. To compute the Pareto set and Pareto front, we use the first one. The solution of Equation (2.16) leads to an optimal solution which can be also seen as a fixed point. Thus, every time the SCM method finds a fixed point, we state that point is a candidate optimal solution. Moreover, even if we do not reach the stopping criteria of Equation (2.16), SCM method can also detect when we cycle around solutions. In this case all the solutions of the cycle are marked as candidate optimal solutions.

In terms of global properties gathered by the SCM method, the candidate optimal solutions are 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.

It is important to notice that because of the dynamical system periodic groups with size greater than 1 should not appear, however, due to discretization errors and too large step sizes periodic groups greater than 1 may be generated (i.e., an oscillation around the Pareto set can occur). Thus, cells that are involved in the current periodic group are also considered to be candidates.

Now, we have all the elements needed to solve a MOP by means of SCM method.

3.3 Description the SCM method for MOPs

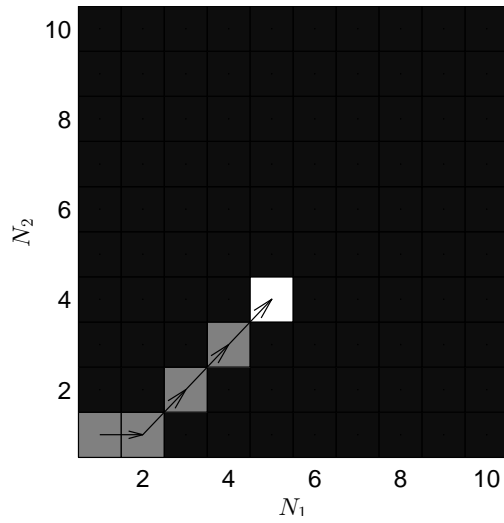
In this section, we put together all the above discussion to write the SCM method for multi-objective optimization. Algorithm 5 shows the key elements of the SCM method for multi-objective optimization.

Now, we describe the behavior of the SCM with an example on MOP (A.1) with a 10×10 grid. Figure 3.5 shows the SCM for the iterations 1, 3, 10 and 50. The result is 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. That is, the path formed by the cells (1, 1), (2, 1), (3, 2), (4, 3), and (5, 4). Cell (5, 4) is an endpoint 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 therefor they should have the same group number. Since, this is the first group motion discovered, we give 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 belong should have the same group number from before (group number 2).

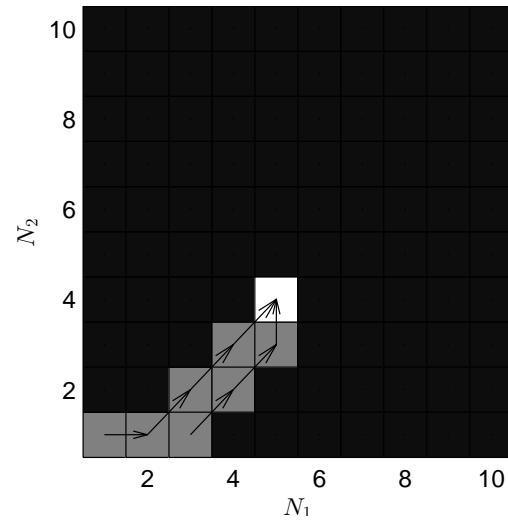
Then, we choose a new starting cell and continue until we finish processing all the cells. As we process the cells, we gather more information of the problem. For this example we have 8 periodic motions with the same number of optimal solutions. Figure 3.6 shows the result of the SCM for multi-objective optimization in cell space.

On the previous example we have used a 10×10 grid. However, different grid sizes can be used. The grid size has an influence on the number of function evaluations to be used and also on the quality of the solutions. Thus, the bigger the grid size, the better the result. On the other hand, it also means more computational effort. Figure 3.7 shows SCM for grid size of 5×5 , 10×10 and 20×20 . We can see even for a small grid size the SCM succeeds to find the cells that contain the optimal solutions. We can also see that as we increase the grid size, we obtain a better approximation of both Pareto set and Pareto front. Further, we can see that although as we increase the grid size, we have a better detail on the mapping, we can see the same patten on the mapping even for the 5×5 grid.

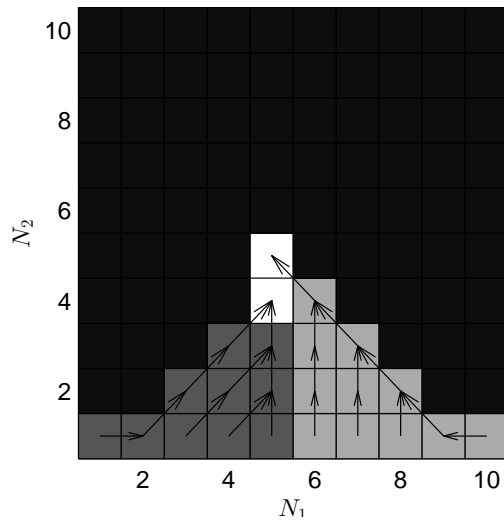
Figure 3.8 shows the result of SCM on different academic models. For the first two model we use a 10×10 grid and for the third model we use a 20×20 grid.



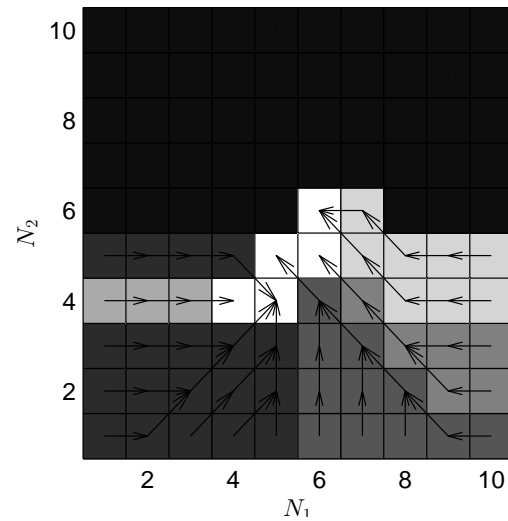
(a) Iteration 1



(b) Iteration 3



(c) Iteration 10



(d) Iteration 50

Figure 3.5: Iteration of SCM. The white cells represent the optimal solution found so far. The arrows show the path from the starting cell to the optimal solution. The darker cells represent unexplored regions.

Algorithm 5 Simple Cell Mapping for MOPs.

Require: $DynamicalSystem, F, ub, lb, N, h, N_c$
Ensure: z, C, Gr, Pe, St, cPs

```

1:  $current\_group \leftarrow 1$ 
2:  $cPs = \{\}$ 
3:  $Gr(i) \leftarrow 0, \forall i \in N_c$ 
4: for all  $p_{cell} \in N_c$  do
5:    $cell \leftarrow p_{cell}$ 
6:    $i \leftarrow 0$ 
7:   while  $newcell = true$  do
8:      $x_i \leftarrow$  center point of  $cell$ 
9:     if  $Gr(cell) = 0$  then
10:       $\nu \leftarrow$  compute with Equation (3.3)
11:       $t \leftarrow$  compute with Equation (3.5)
12:       $p_{i+1} \leftarrow x_i + \nu t$ 
13:       $ncell \leftarrow$  cell where  $p_{i+1}$  is located
14:       $C(cell) \leftarrow ncell$ 
15:       $cell \leftarrow ncell$ 
16:       $i \leftarrow i + 1$ 
17:     end if
18:     if  $Gr(cell) > 0$  then
19:        $Gr(C^j(p_{cell})) \leftarrow Gr(cell), j \leftarrow 0, \dots, i$ 
20:        $Pe(C^j(p_{cell})) \leftarrow Pe(cell), j \leftarrow 0, \dots, i$ 
21:        $St(C^j(p_{cell})) \leftarrow St(cell) + i - j, j \leftarrow 0, \dots, i$ 
22:        $cell \leftarrow C(cell)$ 
23:        $newcell \leftarrow false$ 
24:     end if
25:     if  $Gr(cell) = -1$  then
26:        $current\_group \leftarrow current\_group + 1$ 
27:        $Gr(C^k(p_{cell})) \leftarrow current\_group, k \leftarrow 0, \dots, i$ 
28:        $j \leftarrow i^{th}$  value when period appears
29:        $Pe(C^k(p_{cell})) \leftarrow i - j, k \leftarrow 0, \dots, i$ 
30:        $St(C^k(p_{cell})) \leftarrow j - k, k \leftarrow 0, \dots, j - 1$ 
31:        $St(C^k(p_{cell})) \leftarrow 0, k \leftarrow j, \dots, i$ 
32:        $cPs \leftarrow cPs \cup cell_k, k \leftarrow j, \dots, i$ 
33:        $cell \leftarrow C(cell)$ 
34:        $newcell \leftarrow false$ 
35:     end if
36:   end while
37: end for

```

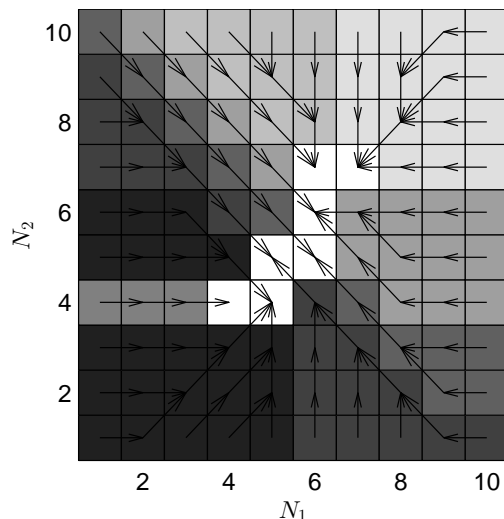


Figure 3.6: SCM on MOP (A.1) in page 81 with a 10×10 grid. White cells represent optimal solutions. Further, cells with the same color belong to the same domain of attraction. Finally, the arrows represent the mapping between cells.

3.4 Numerical Results

In this section, we present the numerical results of the SCM method for multi-objective optimization on three academic functions and we compare the results with the two state-of-the-art algorithms MOEA/D and NSGA-II (the evolutionary computations are averaged over 20 independent runs). Tables 3.1 and 3.2 show the Δ_p values of the candidate sets (respectively their images) to P_Q ($F(P_Q)$) for all further examples.

3.4.1 Problem 1

First, we consider MOP (A.1 in page 81). Figure 3.9 shows the numerical result obtained by the SCM algorithm and the two evolutionary algorithms. A budget of 10,000 function evaluations was used, the grid for SCM was set to $N = [40, 40]$ (9,600 function evaluations). From the figure, we see SCM covers all the Pareto set/front with a good spread. We can also see that the results from SCM to the evolutionary algorithms are competitive. From the tables 3.1 and 3.2, NSGA-II has the best approximation in parameter space, while the SCM has the best approximation in objective space according to Δ_p .

3.4.2 Problem 2

Next, we consider MOP (A.2 in page 82). Figure 3.10 shows some numerical results. A budget of 10,000 function evaluations was used, the grid for SCM was $N = [40, 40]$ (9,600 function evaluations). From the figure, we see the SCM method does not only finds the global optimal solution, but also the 8 regions that are nearly optimal, which is a potential drawback of the method, since in this case we are only interested in global optimal solutions. From the tables, MOEA/D has the best approximation in parameter space, while the NSGA-II has the best approximation in objective space according to Δ_p . In this case, since SCM computes also the local optimal solutions the Δ_p values are not as good as the ones from the evolutionary algorithms.

3.4.3 Problem 3

Further, we consider MOP (A.3 in page 83). Figure 3.11 shows a numerical result of the SCM and the evolutionary strategies for $n = 3$. A budget of 10,000 function evaluations was used, the grid for SCM was $N = [10, 10, 10]$ (6,000 function evaluations). Figure 3.12 shows a numerical result of the SCM and the evolutionary strategies for $n = 5$. A budget of 600,000 function evaluations was used, the grid for SCM was $N = [10, 10, 10, 10, 10]$ (600,000 function evaluations). From the figure, we see the SCM method is the only one that finds the four components of the Pareto set. We also see the SCM computes a good approximation of the Pareto front. From the tables, SCM has the best approximation in parameter space, while the NSGA-II has the best approximation in objective space according to Δ_p .

3.4.4 Problem 4

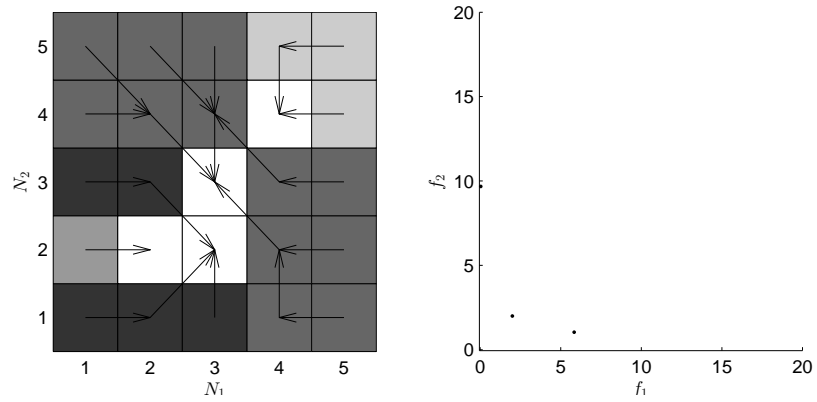
Now, we consider MOP (A.6). Figure 3.13 shows a numerical result of the SCM and the evolutionary strategies. A budget of 6,000 function evaluations was used, the grid for SCM was $N = [20, 10, 5]$ (6,000 function evaluations). For $n = 3$, we see the SCM method together with NSGA-II yield the best results for this problem. From the tables, SCM has the best approximation in parameter space, while the NSGA-II has the best approximation in objective space according to Δ_p . For $n = 5$, SCM is still able to compute the sets of interest, however the computational cost increases exponentially as we increase the number of parameters.

Table 3.1: Δ_p values for the distances of the candidate solution set to $P_{Q,\epsilon}$, the best solutions in **boldface**.

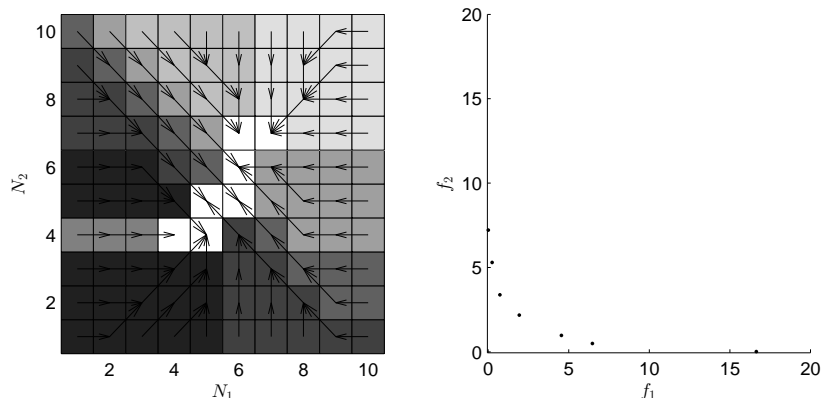
MOP	MOEA/D	NSGA-II	SCM
A.1 (Convex)	0.1376	0.1373	0.2550
A.2 (Rudolph)	0.0400	0.0273	7.8876
A.3 (Schäeffler $n = 3$)	35.0855	22.0992	4.0927
A.3 (Schäeffler $n = 5$)	35.3288	3.8959	3.8458
A.6 (PID)	15.8382	21.3082	13.6784

Table 3.2: Δ_p values for the distances of the images of the candidate sets to $F(P_{Q,\epsilon})$.

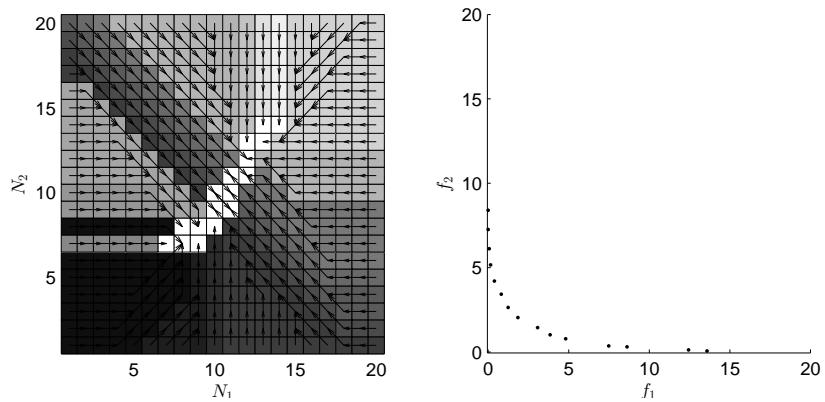
MOP	MOEA/D	NSGA-II	SCM
A.1 (Convex)	3.6766	1.7580	1.7385
A.2 (Rudolph)	0.0667	0.0367	0.1632
A.3 (Schäeffler $n = 3$)	49.2080	2.9000	4.5000
A.3 (Schäeffler $n = 5$)	59.5702	3.6123	7.9329
A.6 (PID)	22.9232	1.7408	3.3082



(a) 5×5 Grid



(b) 10×10 Grid



(c) 20×20 Grid

Figure 3.7: SCM for different grid sizes on MOP (A.1 in page 81). In the left, we present cell space. The cells of the same color belong to the same domain of attraction. Next, white cells that contain the Pareto set. Further, the black arrow represent the mapping between cells. In the right, we present objective space. The black dots represent the approximation of the Pareto front.

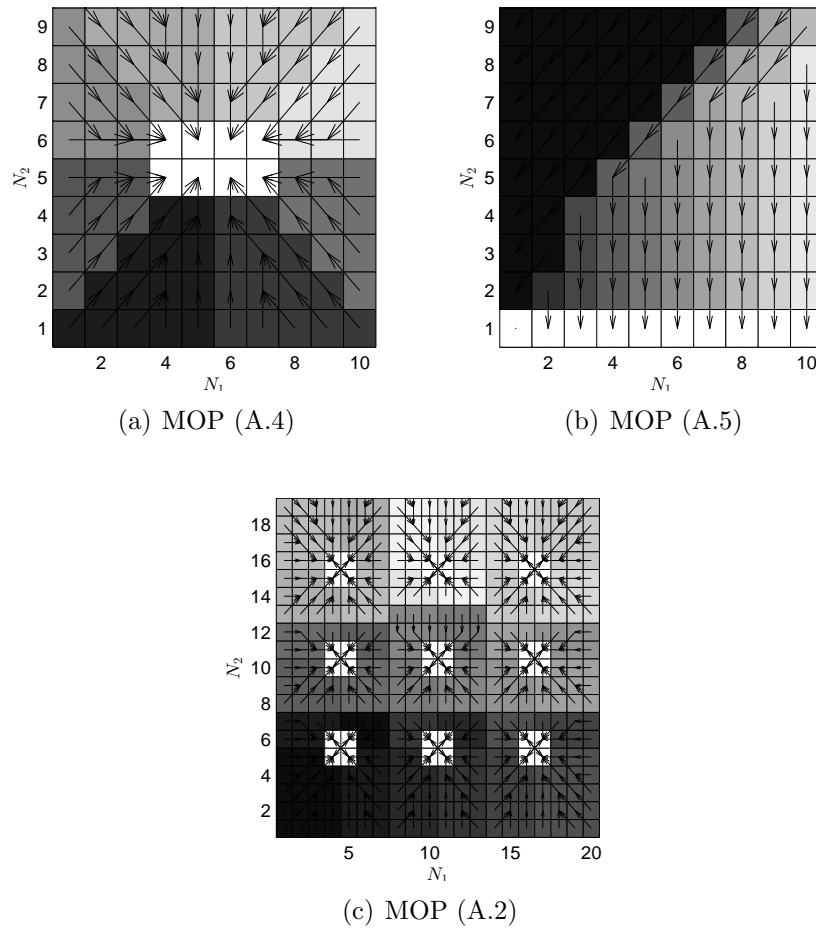
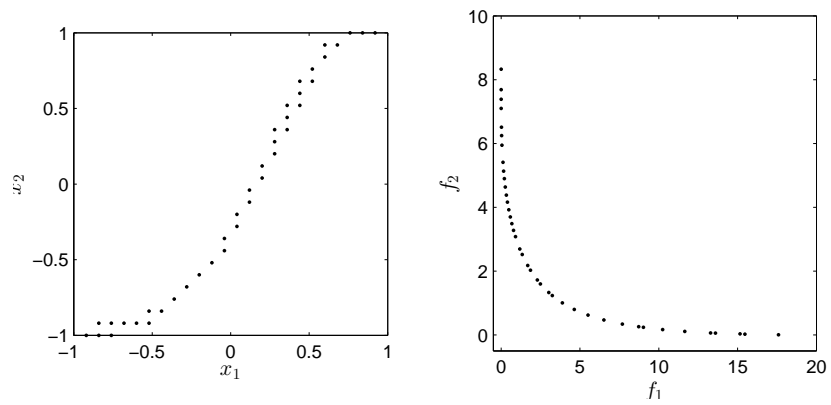
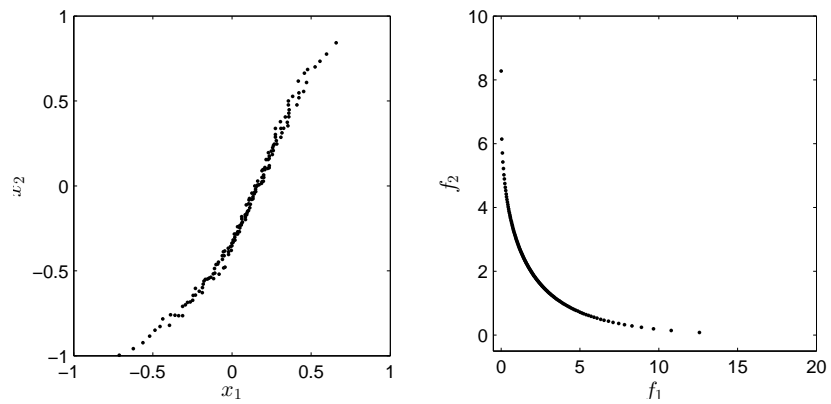


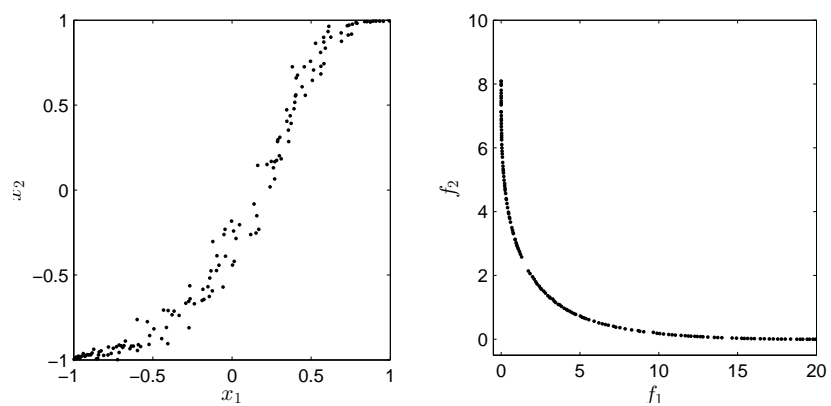
Figure 3.8: SCM for three Academic models on cell space. The cells of the same color belong to the same domain of attraction. Next, white cells that contain the Pareto set. Further, the black arrow represent the mapping between cells. In the right, we present objective space.



(a) Cell Mapping

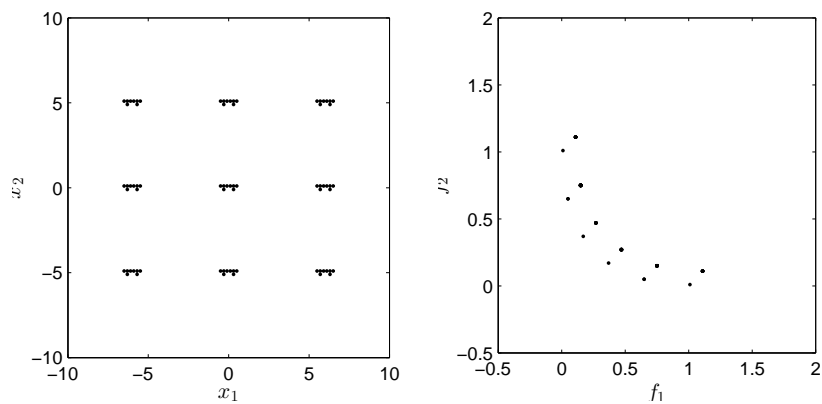


(b) MOEA/D

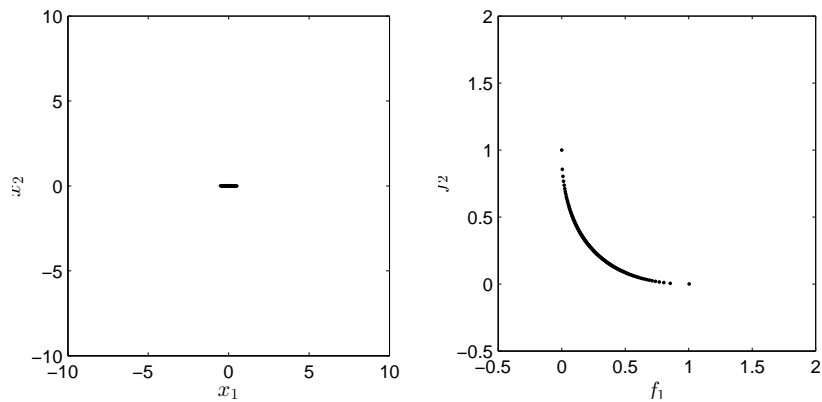


(c) NSGA-II

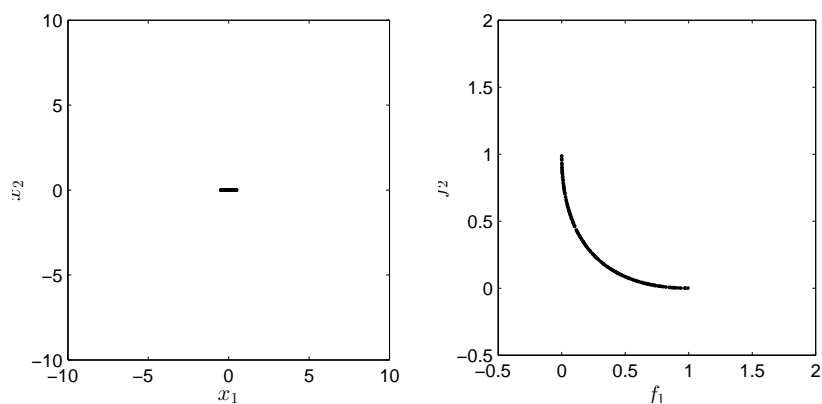
Figure 3.9: Numerical results for MOP (A.1). Black points indicate Pareto optimal candidates.



(a) Cell Mapping

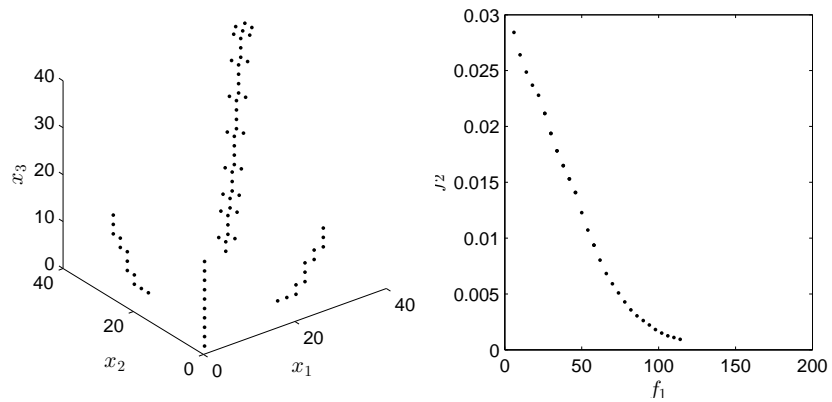


(b) MOEA/D

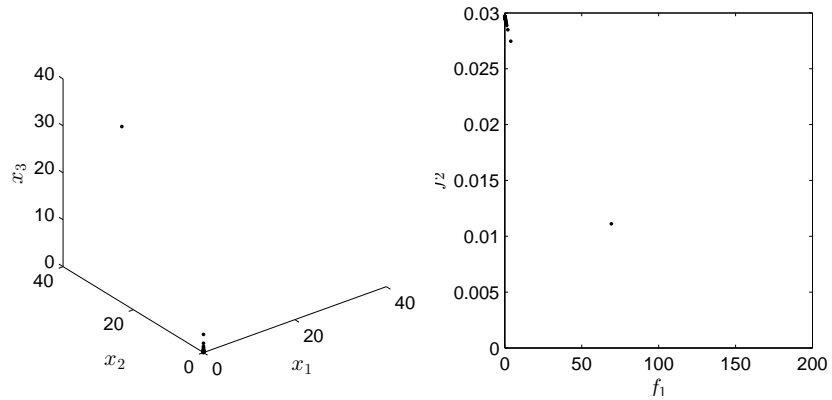


(c) NSGA-II

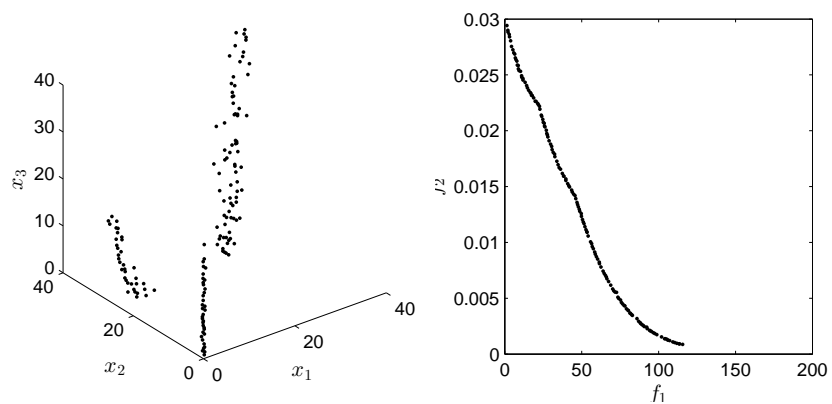
Figure 3.10: Numerical results for MOP (A.2). Black points indicate Pareto optimal candidates.



(a) Cell Mapping

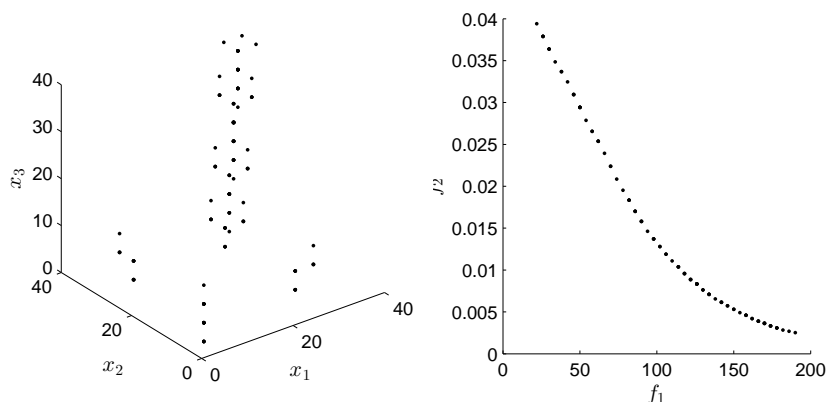


(b) MOEA/D

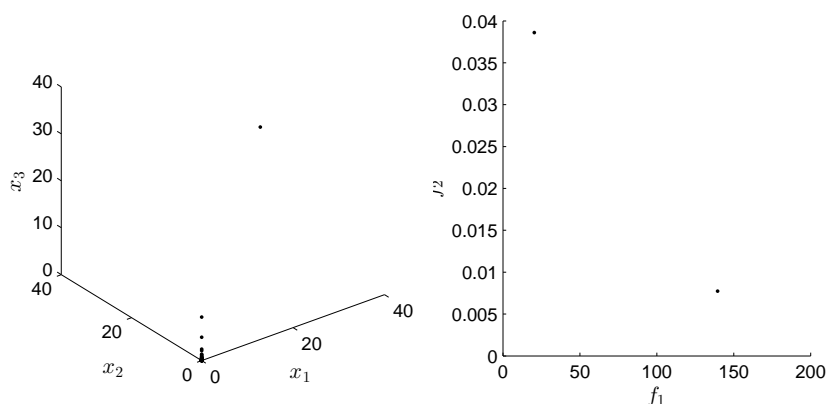


(c) NSGA-II

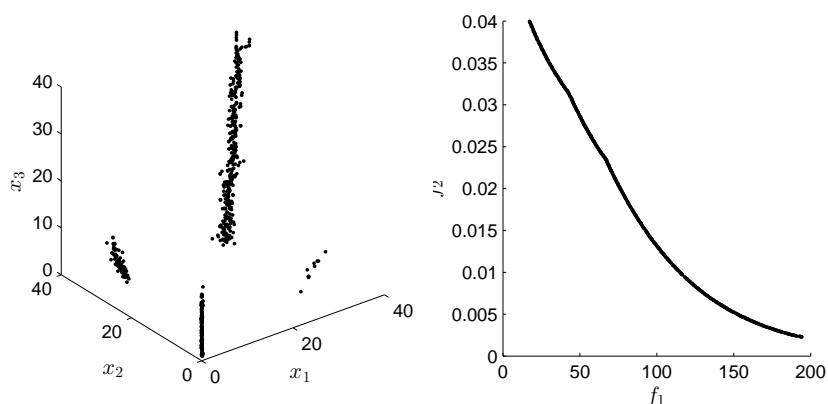
Figure 3.11: Numerical results for MOP (A.3) with $n = 3$. Black points indicate Pareto optimal candidates.



(a) Cell Mapping

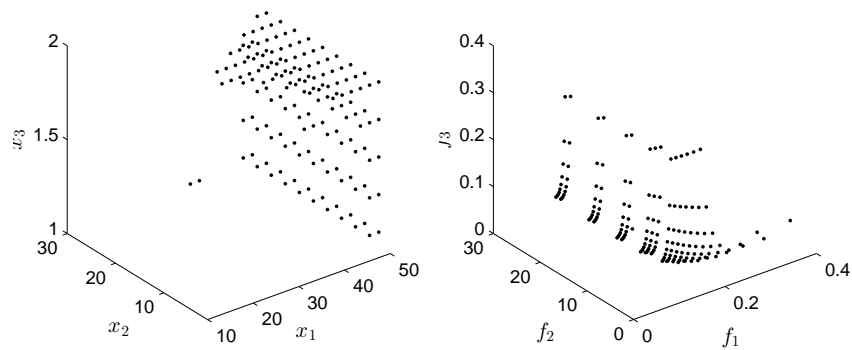


(b) MOEA/D

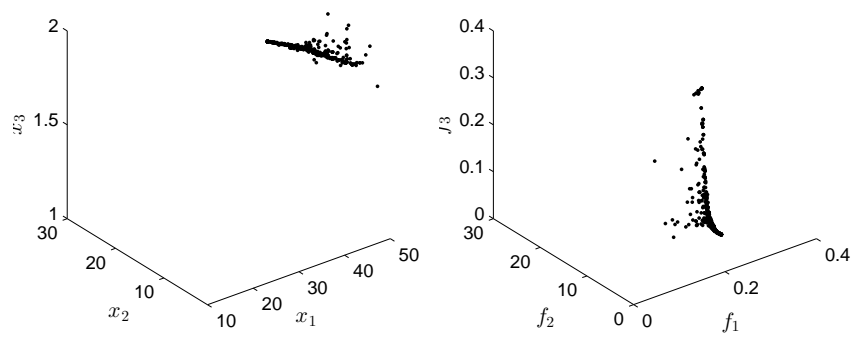


(c) NSGA-II

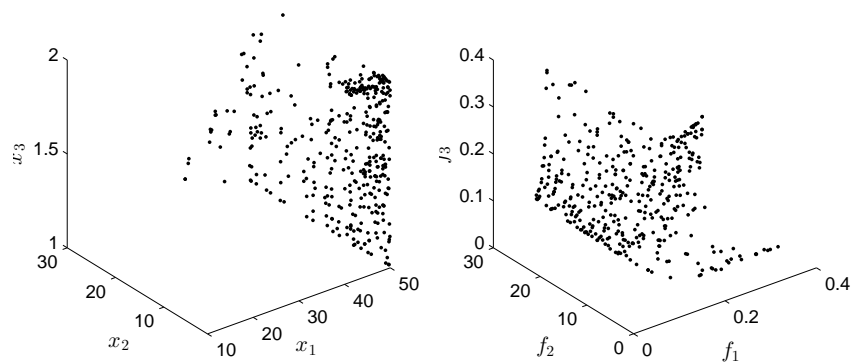
Figure 3.12: Numerical results for MOP (A.3) $n = 5$. Black points indicate Pareto optimal candidates.



(a) Cell Mapping



(b) MOEA/D



(c) NSGA-II

Figure 3.13: Numerical results for MOP (A.6 in page 86). Black points indicate Pareto optimal candidates.

Chapter 4

Simple Cell Mapping for the Approximation of the Set of Approximate Solutions

Here we address the problem of computing the set of approximate solutions of a given MOP. This set is of potential interest for the decision maker (DM) since it might give him/her additional solutions to the optimal ones for the realization of the project related to the MOP.

In a variety of applications in industry and finance a problem arises that several objective functions have to be optimized concurrently leading to multi-objective optimization problems. For instance, in space mission design, there are two crucial aims for the realization of a transfer: Minimization of flight time and fuel consumption of the spacecraft [43, 44, 45]. In this case, it makes sense to consider in addition to the ‘optimal’ trajectories also approximate solutions since by this the decision maker is offered a much larger variety of possibilities, and to present one way to compute this enlarged set of interest with reasonable effort.

As a motivating example, consider the MOP which is a model taken from [27] for the transfer from Earth to Mercury, and the following two points x_i with images $F(x_i), i = 1, 2$:

$$x_1 = (782, 1288, 1788), F(x_1) = (0.462, 1001.7)$$

$$x_2 = (1222, 1642, 2224), F(x_2) = (0.463, 1005.3)$$

The two objectives are the propellant mass fraction i.e., the portion of the vehicle’s mass which does not reach the destination and the flight time (in days). In the domain, the first parameter is of particular interest: It determines the departure time from the Earth (in days after 01/01/2000). $F(x_1)$ is less than $F(x_2)$ in both components, and thus, x_1 can be considered to be ‘better’ than x_2 . However, note that the difference in image space is small: the mass fraction of the two solutions differs by 0.001 which makes 0.1% of the total mass, and the flight time differs by four days for a transfer which takes almost three years. In case the DM is willing to

accept this deterioration, it will offer him/her a second choice in addition to x_1 for the realization of the transfer: While the two solutions offer ‘similar’ characteristics in image space this is not the case in the design space since the starting times for the two transfers differ by 440 days.

The identification of the two solutions would be a fundamental requirement during the preliminary design of a space mission. In fact, in order to increase the reliability of the design, the mission analysts would need to identify one or more back-up solutions, possibly with identical cost, for each baseline solution. Furthermore, for each mission opportunity (i.e. each launch date) rather than an optimal solution, it is generally required to identify a set of nearly optimal ones, possibly all with similar cost.

As stated before the SCM method is restricted to small dimensions of the parameter space. However, there exist small dimensional problems where a thorough investigation is desirable. Such models e.g. arise in preliminary space mission design (e.g., [43, 44, 45]) or in the design of electrical circuits (e.g., [46, 47]).

In this study, we make a first attempt to adapt the SCM to the problem at hand. Due to their global approach, these methods are well-suited for the thorough investigation of small problems, including the computation of the set of approximate solutions. Finally, we present three academic bi-objective optimization problems including a comparison to a related evolutionary approach.

4.1 Archiving Techniques

In this section, we present the archiving procedure ArchiveUpdateTight2 [32]. This procedure will be used together with the SCM method in order to compute the set of approximate solutions P_{Q_ϵ} .

Definition 7 (Tight Sets [32]) 1. A set $A_\epsilon \subset Q$ is called Δ_M -tight ϵ -approximation Pareto set of Problem (2.1) if A_ϵ is ϵ -approximate Pareto set and

$$\text{dist}(F(P_Q), F(A_\epsilon)) \leq \Delta_M.$$

2. A set $A_\epsilon^* \subset Q$ is called a Δ_M -tight ϵ -Pareto set if A_ϵ^* is an ϵ -Pareto set of Problem (2.1) and

$$d_H(F(P_Q), F(A_\epsilon^*)) \leq \Delta_M.$$

Definition 8 ([32]) Let $\epsilon \in \mathbb{R}_+^*$

1. A set $A_\epsilon \subset Q$ with $|A_\epsilon| \geq 2$ is called a (Δ_M, Δ_m) -tight ϵ -approximation Pareto set if A_ϵ^* is an Δ_M -tight ϵ -approximation Pareto set and

$$\text{dist}(F(a), F(A \setminus \{a\})) \geq \Delta_m, \forall a \in A.$$

2. (Δ_M, Δ_m) -tight ϵ -Pareto set is defined analogously.

ArchiveUpdateTight2 leads to a sequence of archives $A_l, l \in \mathbb{N}$, such that there exists with probability one, an integer $l_0 \in \mathbb{N}$ such that A_l is an $\Delta - tight$ ϵ -approximate Pareto set for all $l \geq l_0$ and $\lim_{l \rightarrow \infty}$. Algorithm 6 describes the procedure of ArchiveUpdateTight2.

Algorithm 6 ArchiveUpdateTight2

Require: P, A_0

Ensure: A

```

1:  $A \leftarrow A_0$ 
2: for all  $p \in P$  do
3:   if  $\nexists a \in A : a \prec_\epsilon p$  or  $\nexists a \in A : a \prec p$  and  $\forall a \in A : d_\infty(F(a), F(p)) > \Delta$  then
4:      $A \leftarrow A \cup \{p\}$ 
5:   end if
6:   for all  $a \in A$  do
7:     if  $p \prec a$  then
8:        $A \leftarrow A \cup \{p\} \setminus \{a\}$ 
9:     end if
10:  end for
11: end for

```

Given an archive A_0 the archiver accepts a solution p if the one of the following conditions is true.

- If all of the conditions in line 3 of Algorithm 6 is true
- If there exists an element $a \in A_0$ which is dominated by p

4.2 Description of the SCM Method for the Set of Approximate Solutions

After one run of the SCM 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 presented above.

The integration of both algorithms is as follows: Once the group number of the current periodic motion is discovered, we use Algorithm 7 to compute the set $P_{Q,\epsilon}$. Algorithm 7 updates the archive first with the periodic group of the current periodic motion and continues with the rest of the periodic motion. Once it finds a cell which is not in $P_{Q,\epsilon}$ it stops the procedure. The reason for this can be easily seen. Since each periodic group is a curve of dominated points, once a point $x_j \notin P_{Q,\epsilon}$ the other points would not be either, since by construction these points are dominated by x_j .

In the worst case Algorithm 7 visits all the cells. If this is the case, all solutions in Q are in the set $P_{Q,\epsilon}$. Thus, the presented approach would be equivalent to apply

Algorithm 7 Post-processing to get $P_{Q,\epsilon}$

Require: $P_{Q,\epsilon}, C, pcell, i$

Ensure: $P_{Q,\epsilon}$

```

1:  $j \leftarrow i - 1$ 
2:  $isP_{Q,\epsilon} \leftarrow \mathbf{true}$ 
3: while  $isP_{Q,\epsilon}$  do
4:    $x_j \leftarrow$  center point of  $C^j(pcell)$ 
5:    $P_{Q,\epsilon} \leftarrow ArchiveUpdateTight2(P_{Q,\epsilon}, x_j, F(x_j), \epsilon, h)$ 
6:   if  $x_j \notin P_{Q,\epsilon}$  then
7:      $isP_{Q,\epsilon} \leftarrow \mathbf{false}$ 
8:   end if
9:    $j \leftarrow j - 1$ 
10: end while

```

the archiving technique to all centers points of the cells. Typically, this is not the case and the number of cells run in the archive is much lower than the total number of cells.

Now, we describe the behavior of the SCM for approximate solutions with an example on MOP (A.1 in page 81) with a 10×10 grid and $\epsilon = 0.5$. Figure 4.1 shows the SCM for the iterations 1, 3, 10 and 50. The result is in cell space. Note than even if a cell is marked to be an approximate solution, it is possible than in later iterations this cell is removed from the set.

It is also important to notice the impact of the ϵ value on the final solution. If a value of $\epsilon = 0$ is chosen then the solution is equivalent to the optimal one. As we increase the value of ϵ more solutions become accepted. Figure 4.2 shows the result of SCM on MOP A.1 with different ϵ values.

Figure 4.3 shows the final result of the SCM for the set of approximate solutions in cell space. We can see that the set of approximate solutions is bigger than the set of optimal solutions shown in Figure 3.6.

4.3 Numerical Results

In this section, we present the numerical results of the SCM method for the approximation of $P_{Q,\epsilon}$ on three academic functions and we compare the results with two of algorithms of the state-of-the-art MOEA/D and NSGA-II (the results of the evolutionary computations are averaged over 20 independent runs). Tables 4.1 and 4.2 show the Δ_p values of the candidate sets (respectively their images) to $P_{Q,\epsilon}$ ($F(P_{Q,\epsilon})$) for all further examples.

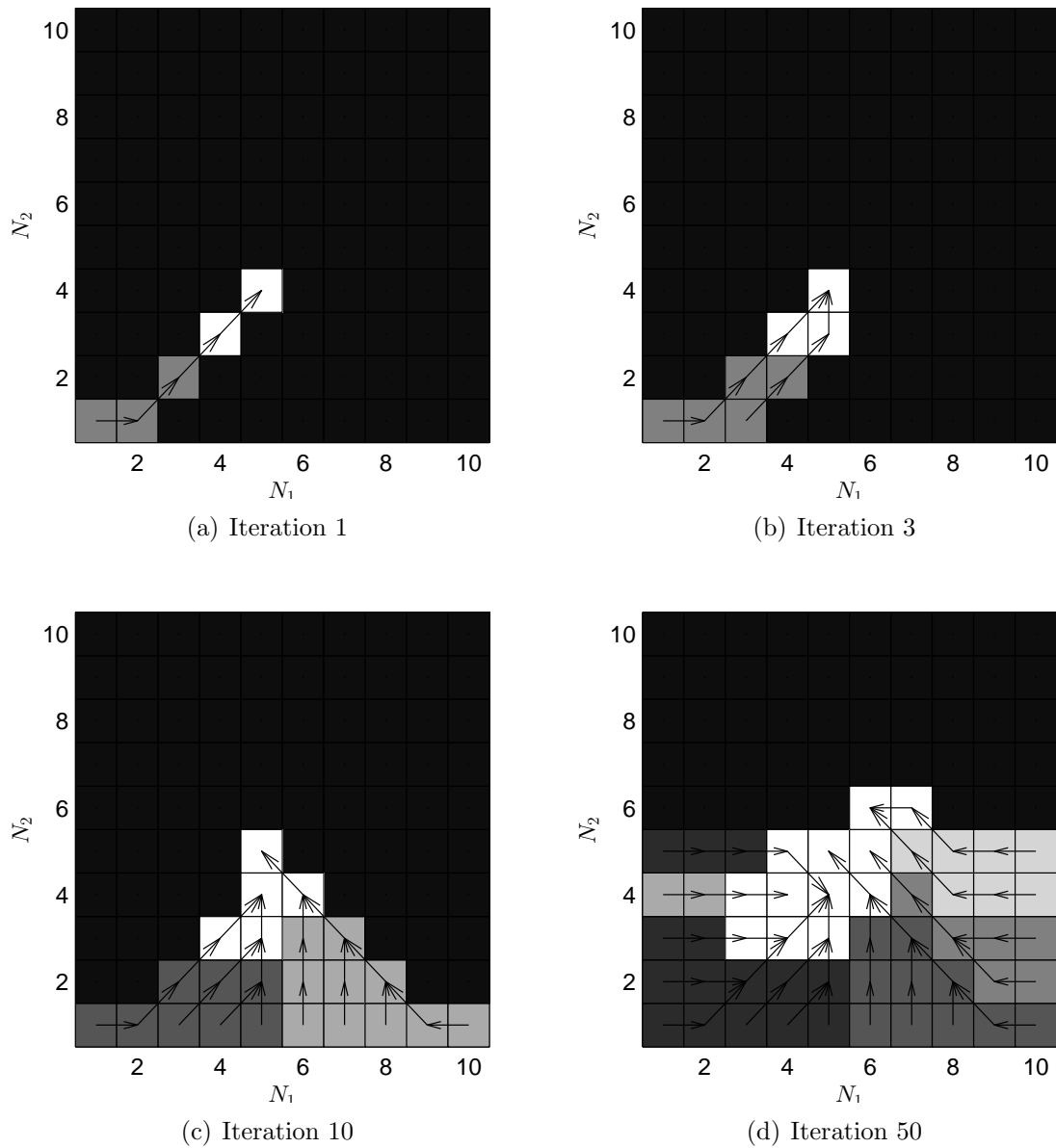


Figure 4.1: Iteration of SCM for $P_{Q\epsilon}$. The white cells represent the near optimal solutions. Finally, the darker cells represent unexplored regions.

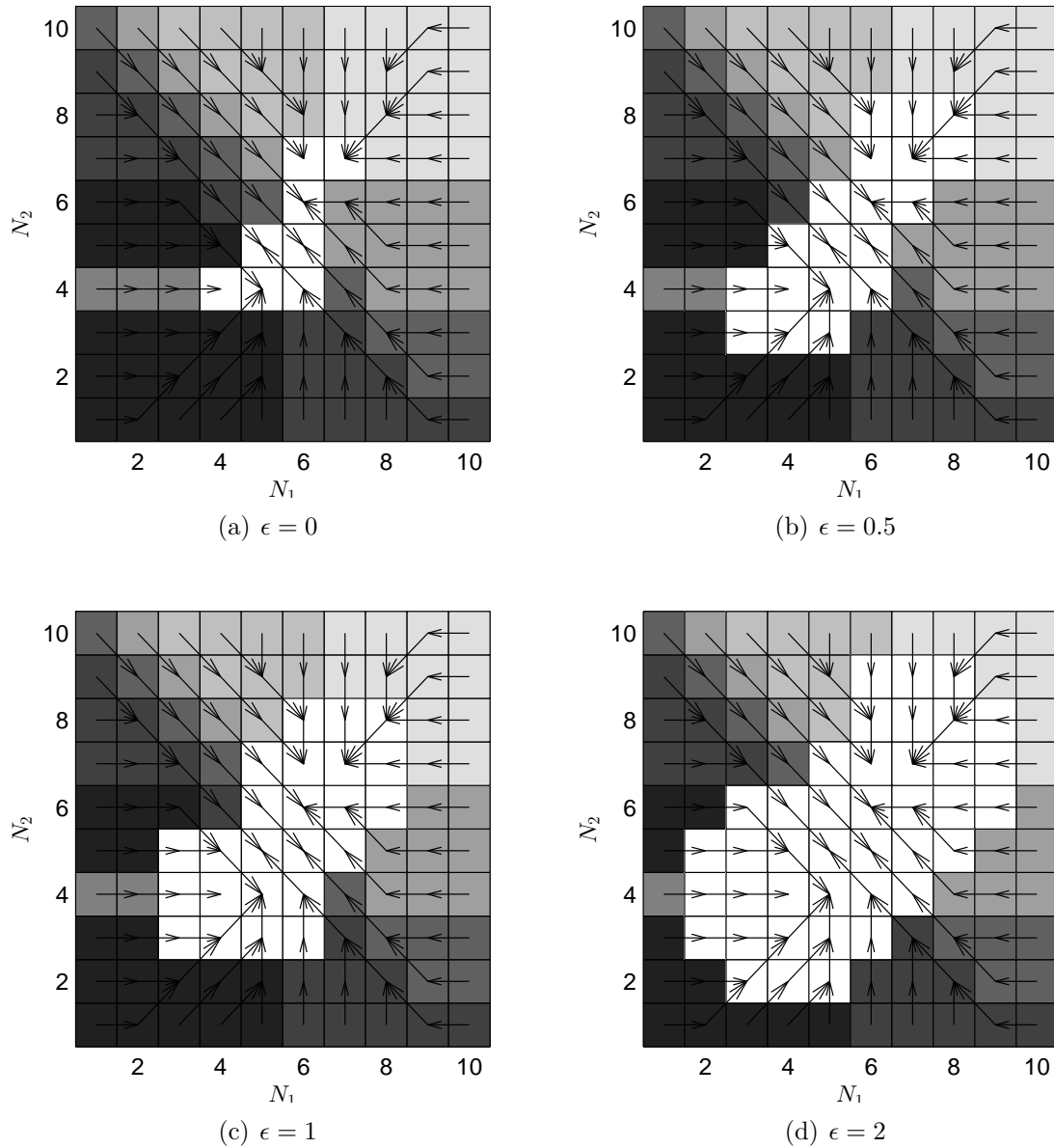


Figure 4.2: Impact of ϵ values. The white cells represent the near optimal solutions. Finally, the darker cells represent unexplored regions.

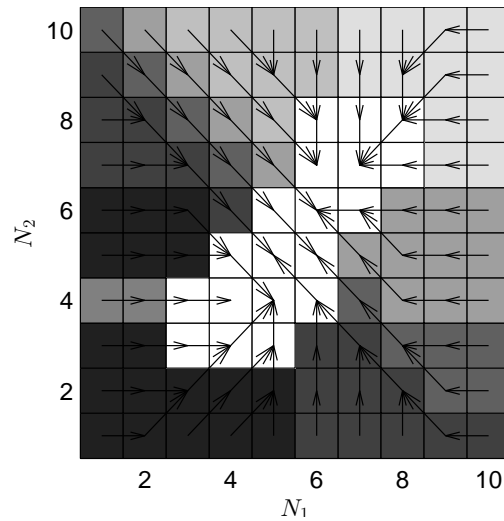


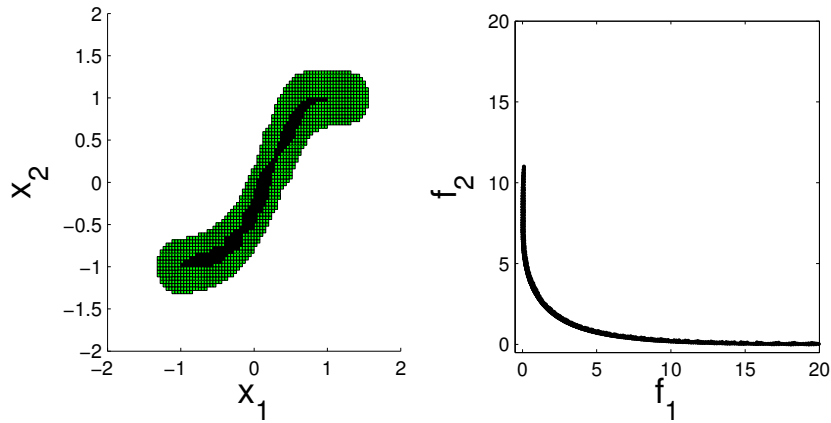
Figure 4.3: SCM on MOP (A.1 in page 81) with a 10×10 grid and $\epsilon = 0.5$. White cells represent near optimal solutions

4.3.1 Problem 1

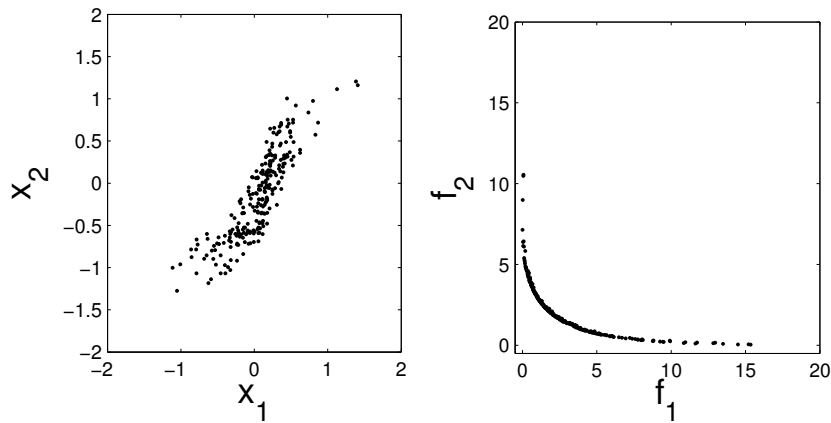
First, we consider the MOP (A.1 in page 81). Figure 4.4 shows the numerical result obtained by the SCM algorithm and the two evolutionary algorithms. A budget of 10,000 function evaluations was used, the grid for SCM was set to $N = [40, 40]$ (9,600 function evaluations). Apparently, SCM is able to get the best approximation of the set of interest, in particular in parameter space.

4.3.2 Problem 2

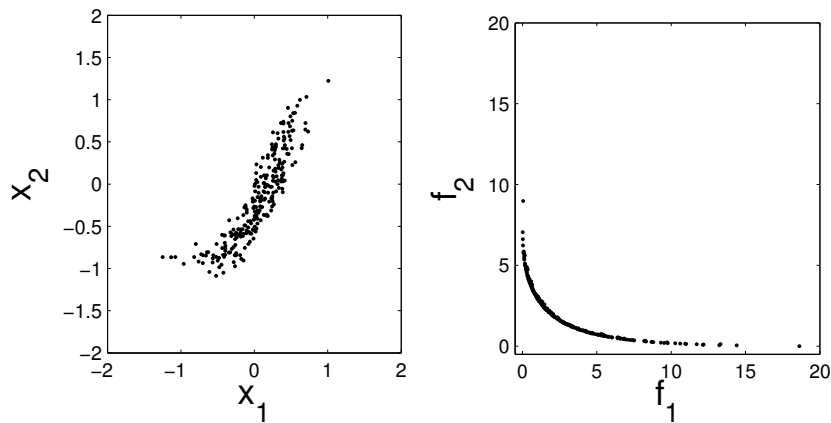
Next, we consider the MOP (A.2 in page 82). Figure 4.5 shows some numerical results. A budget of 60,000 function evaluations was used, the grid for SCM was $N = [100, 100]$ (60,000 function evaluations). SCM computes a covering of the entire set of interest, while the evolutionary strategies do not always detect all components. As a hypothetical decision making problem assume the DM is interested in the performance $Z = [0.2132, 0.2932]$ (measured in objective space) and is willing to accept a deterioration of $\epsilon = [0.1, 0.1]$. Then, e.g. the representatives of the cells those images are within the target regions can be presented to the DM leading here to the following 22 candidate solutions:



(a) Cell Mapping



(b) MOEA/D



(c) NSGA-II

Figure 4.4: Numerical results for MOP (A.1). Black cells indicate Pareto optimal candidates and gray cells regions in $P_{Q,\epsilon}$ that are not optimal.

$$\begin{aligned}
& (-6.04, -5.00), \quad (-0.04, -5.00), \quad (5.96, -5.00), \quad (-6.04, -0.04), \\
& (-0.12, -0.04), \quad (-0.04, -0.28), \quad (-0.04, -0.20), \quad (-0.04, -0.12), \\
& (-0.04, -0.04), \quad (-0.12, 0.04), \quad (-0.04, 0.04), \quad (-0.04, 0.12), \\
& (-0.04, 0.20), \quad (-0.04, 0.28), \quad (0.04, -0.12), \quad (0.04, -0.04), \\
& (0.04, 0.04), \quad (0.04, 0.12), \quad (5.96, -0.04), \quad (-6.04, 5.00), \\
& (-0.04, 5.00), \quad (5.96, 5.00).
\end{aligned}$$

The solutions are well-spread and come in this case from all nine components of $P_{Q,\epsilon}$. Since these components are located in different regions of the parameter space, the DM is hence given a large variety for the realization of his/her project.

4.3.3 Problem 3

Finally, we consider the MOP (A.3 in page 82). Figure 4.6 shows a numerical result of the SCM and the evolutionary strategies. A budget of 60,000 function evaluations was used and the grid for SCM was $N = [20, 20, 20]$ (48,000 function evaluations). Also in this case, SCM obtains the best result in particular measured in parameter space (compare also to Tables 1 and 2). Figure 4.7 shows the resulting boxes of another hypothetical decision making problem where we have chosen $Z = [23, 0.02231]$ and $\epsilon = [2, 0.0004]$. Also here, the DM is offered an entire range of solutions with different parameter values.

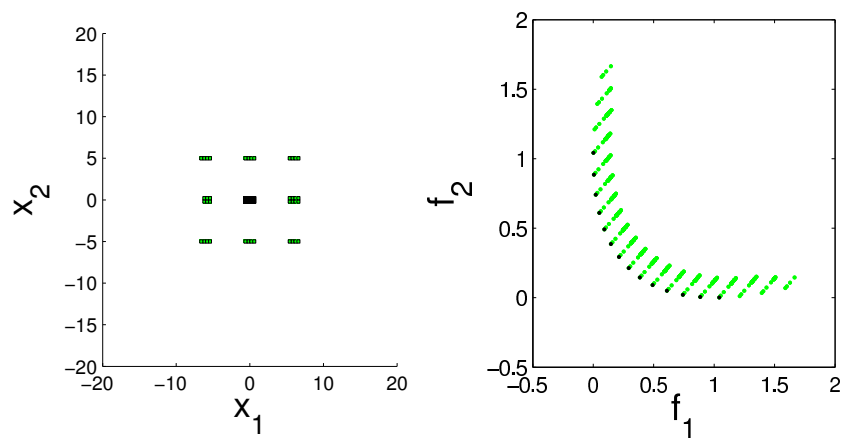
The results indicate that the cell mapping technique is able to reliably compute the set of approximate solutions, and is faster than the evolutionary approaches. The method, however, is restricted to small dimensions of the decision space.

Table 4.1: Δ_p values for the distances of the candidate solution set to $P_{Q,\epsilon}$, the best solutions in boldface. For MOP (A.2) (Rudolph) and MOP (A.3) (Schäffler).

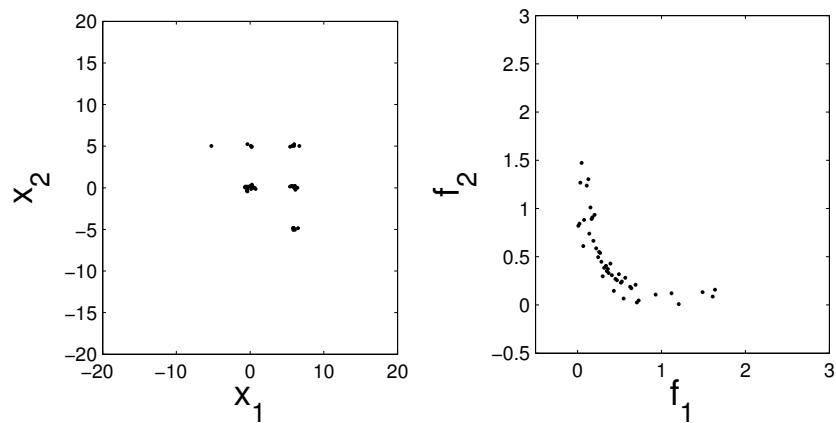
MOP	MOEA/D	NSGA-II	SCM
A.1(Convex)	0.5141	0.4628	0.0849
A.2(Rudolph)	5.0507	7.4737	0.0632
A.3(Schäffler)	10.8365	10.9384	0.8660

Table 4.2: Δ_p values for the distances of the images of the candidate sets to $F(P_{Q,\epsilon})$.

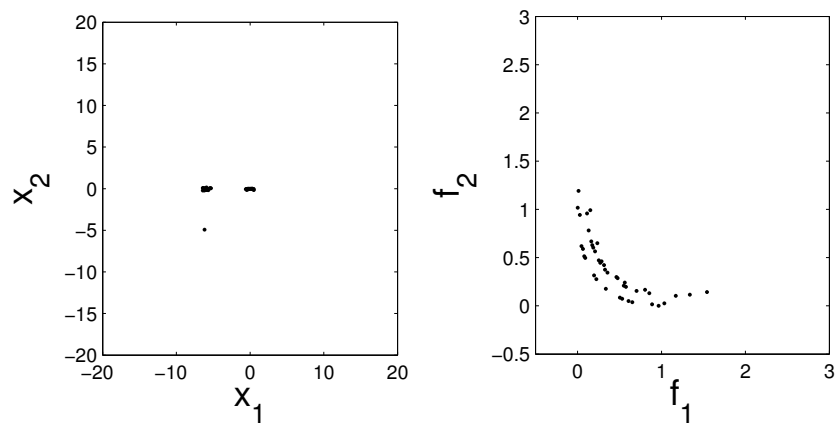
MOP	MOEA/D	NSGA-II	SCM
A.1(Convex)	7.8902	8.0027	2.4250
A.2(Rudolph)	0.4276	0.6317	0.0524
A.3(Schäffler)	5.8152	2.6852	1.5000



(a) Cell Mapping

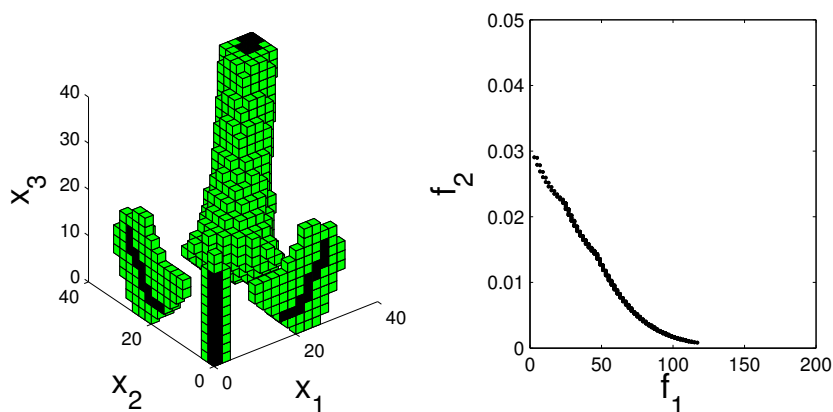


(b) MOEA/D

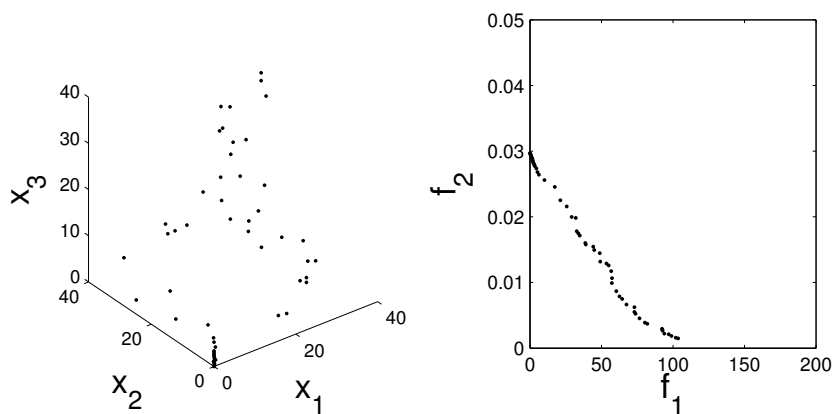


(c) NSGA-II

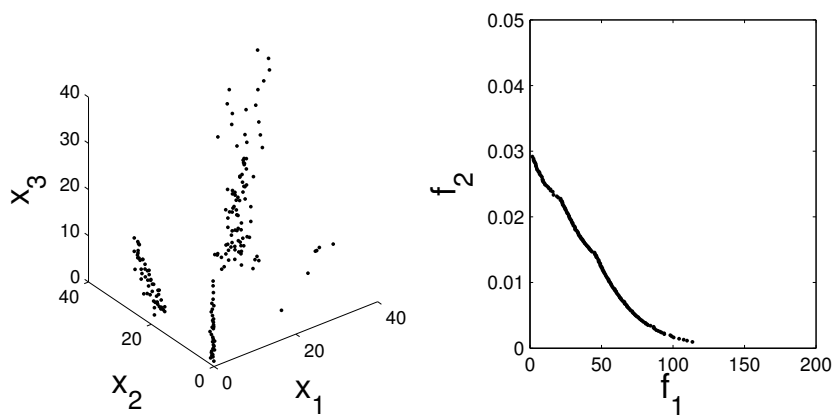
Figure 4.5: Numerical results for MOP (A.2). Black cells indicate Pareto optimal candidates and gray cells regions in $P_{Q,\epsilon}$ that are not optimal.



(a) Cell Mapping



(b) MOEA/D



(c) NSGA-II

Figure 4.6: Numerical results for MOP (A.3). Black cells indicate Pareto optimal candidates and gray cells regions in $P_{Q,\epsilon}$ that are not optimal.

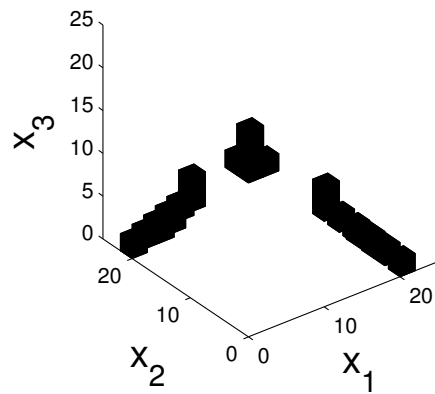


Figure 4.7: Potential solutions for the hypothetical decision making problem related to MOP (A.2).

Chapter 5

Analysis of Descent Direction Methods

So far, we have only used the descent direction proposed in [18]. Although this descent direction yields good results, we have other candidates to use along with SCM. In this chapter, we make a comparison of the descent direction methods by means of the SCM.

First, we present the problem at hand (Section 5.1), next we make a empirical comparison of the methods on an academic function to give an intuition of the study (Section 5.2). We conclude this chapter with a study of the domains of attraction by means of the SCM on three academic functions (Section 5.3).

5.1 The Problem

Since, we have many options of descent direction methods, when one is faced with the task of choosing the best descent direction method for the given application, an important question arises. This is, if the methods have a preference in finding a segment of the Pareto front, that is given a random point x_0 , we do not have the same probability to find any Pareto optimal point.

This is of particular interest for memetic evolutionary algorithms. This algorithms combines evolutionary algorithms with local search with the goal of improving their results. One possible option for local search is to use descent direction methods. However, if the current descent direction methods have a preference towards a segment of the Pareto front. It might lead to a lost of diversity on the evolutionary algorithm, which is highly unwanted. Therefore, it is important to study the descent direction methods and to identify to which kind of problems is better to use a given descent direction method.

To answer this question, we use one of the features of the SCM. Until now, we have only explored the periodic groups found by the SCM. However, the SCM also gives as result the domains of attraction for all periodic groups found. We propose to use this feature to compute the domain of attraction for all Pareto optimal points. With

this information we are able give a fair comparison of the descent direction methods by measuring the size of the domain of attraction of all Pareto optimal points.

5.2 Empirical Comparison of DDMs

In this section, we present a first comparison of the descent direction methods. We have picked MOP (A.1). The comparison is as follows:

We choose 10 random points and then apply the descent direction methods [2, 16, 18] to all the points for the same number of steps and an initial step size of 1. Figure 5.1 shows the comparison of the descent direction methods using one and five steps. From the figure we can see Fliege and Schäffler yield the same results and also for the two methods we can see a small preference to the end points of the Pareto set/front. However, it is not enough to conclude on this matter. Table 5.1 shows the results for Δ_p on the comparison that uses 5 steps.

Table 5.1: Δ_p values for the distances of the candidate solution set to $P_{Q,\epsilon}$ and $F(P_{Q,\epsilon})$ for MOP (A.1).

	Lara [18]	Fliege [2]	Schäffler [16]
Parameter space	1.9671	2.1573	2.1573
Objective space	15.6873	14.2295	14.2295

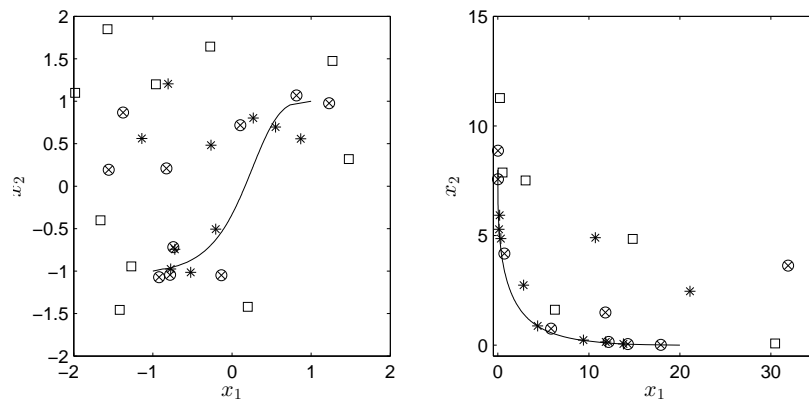
5.3 Comparison of DDMs by SCM

In this section, we present the results of the comparison of the descent direction methods on three academic functions. We use slightly different methods to make the comparison in parameter space and objective space.

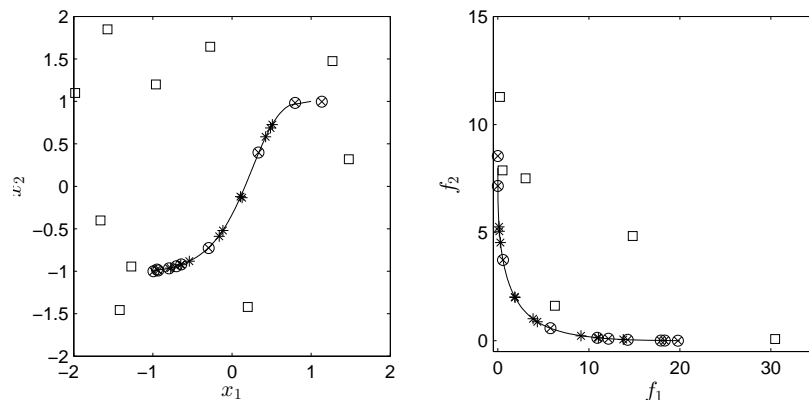
We first look into the comparison in parameter space. Once we have the results of the SCM for MOO, we cluster the cells by their group number (Gr) and then compute the size of the domain of attraction of each group motion which is given by the number of cells that have the same group number.

Since the number of group motions can be very high (almost each Pareto optimal point has a different group motion), it is convenient to group the group motions that are the closer together. For this we use a new discretization $N_Q \in \mathbb{N}^n$ for the feasible region Q where $N_Q \leq N$. With N_Q we can now group the group motions by computing for each Pareto optimal a new cell number with N_Q .

Now we look into objective space. Although we could use the results of the domains of attraction from parameter space, we would have the problem of choosing which of the Pareto optimal points would represent the region in objective space and also how big this region would be. Thus, we propose to also make a discretization of the objective space $N_{F(Q)} \in \mathbb{N}^k$ in the region bounded by the ideal vector and the



(a) One Step



(b) Five Steps

Figure 5.1: Comparison of three descent direction methods. The squares represent the initial points, asterisks represent Lara [18], crosses represent Schäffler [16] and circles represent Fliege [2] a) Shows the result for one iteration of the methods. b) Shows the result for five iterations of the methods.

nadir vector. Then, we can group the groups motions in the same way as in parameter space.

5.4 Numerical Results

In the following, we present the comparison for the descent direction methods. For all the figures a lighter color means a bigger domain of attraction, while a darker color means a smaller domain of attraction.

5.4.1 Problem 1

First, we look into MOP (A.1). Figure 5.2 shows the result of SCM on cell space with a 10×10 grid. Here we can see the mapping for the different descent directions along with the optimal solutions found. Figure 5.3 shows the result obtained by the SCM algorithm, the grid for SCM was set to $N = [100, 100]$, $N_Q = [10, 10]$ and $N_{F(Q)} = [10, 10]$. Apparently, all the methods show a preference towards the extreme points of the both Pareto set and Pareto front. However, in Lara's method is not as clear as it is for the other two methods.

5.4.2 Problem 2

Next, we look into MOP (A.4). Figure 5.4 shows the result of SCM on cell space with a 21×21 grid. This allows to visually compare the difference on mapping for the descent direction methods. Figure 5.5 shows the result obtained by the SCM algorithm, the grid for SCM was set to $N = [100, 100]$, $N_Q = [20, 20]$ and $N_{F(Q)} = [20, 20]$. In this case, it is clearer a preference towards the extreme points of the sets.

5.4.3 Problem 3

Finally, we look into MOP (A.4). Figure 5.6 shows the result of SCM on cell space with a 21×21 grid. This allows to visually compare the difference on mapping for the descent direction methods. Figure 5.7 shows the result obtained by the SCM algorithm, the grid for SCM was set to $N = [100, 100]$, $N_Q = [20, 20]$ and $N_{F(Q)} = [20, 20]$. For Fliege and Schäffler it shows a preference for the optimal points in the left and the size of the domain of attraction is smaller as we move to the right. We can also notice that for this problem Fliege and Schäffler give different slightly different results, opposite to the previous problems where both were the same. In Lara we can see a clear preference towards any section of the Pareto set/front.

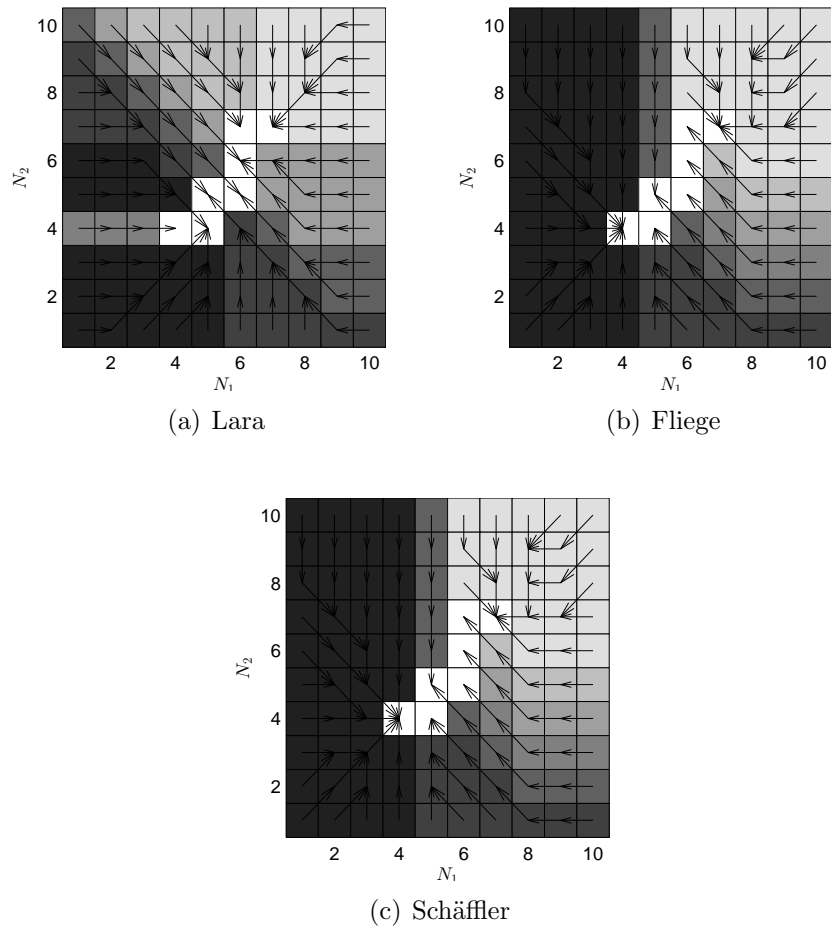
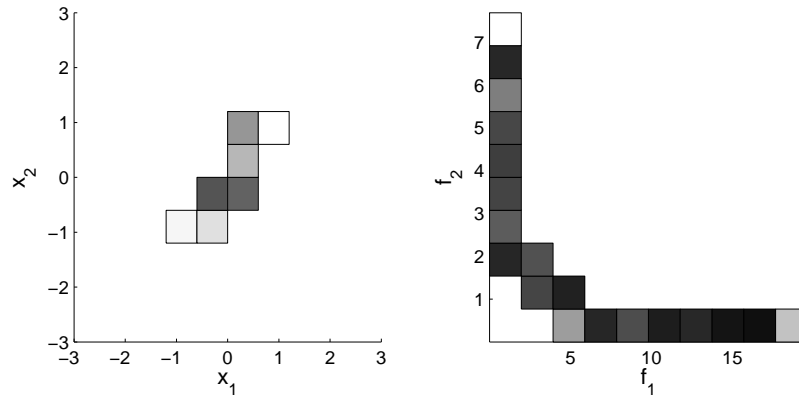
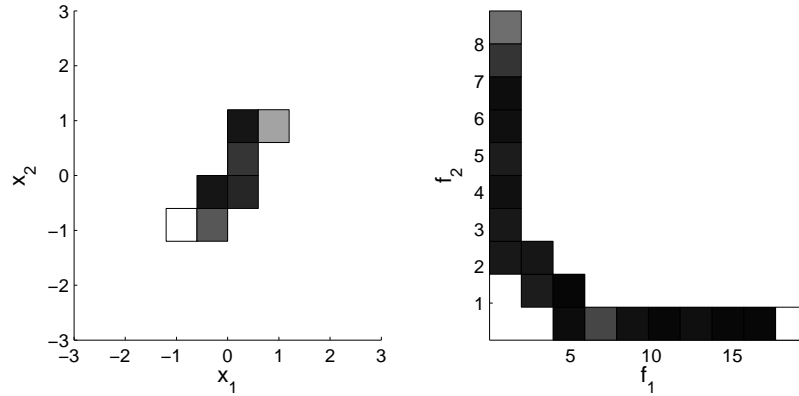


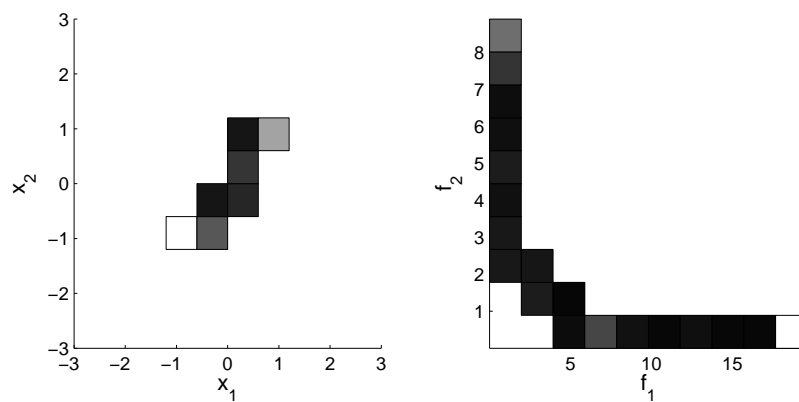
Figure 5.2: SCM on MOP (A.1) in cell space with different descent direction methods.



(a) Lara



(b) Fliege



(c) Schäffler

Figure 5.3: Comparison of three descent direction methods for MOP (A.1). We show parameter space (left) with $N_Q = 10 \times 10$ and objective space (right) $N_{F(Q)} = 10 \times 10$.

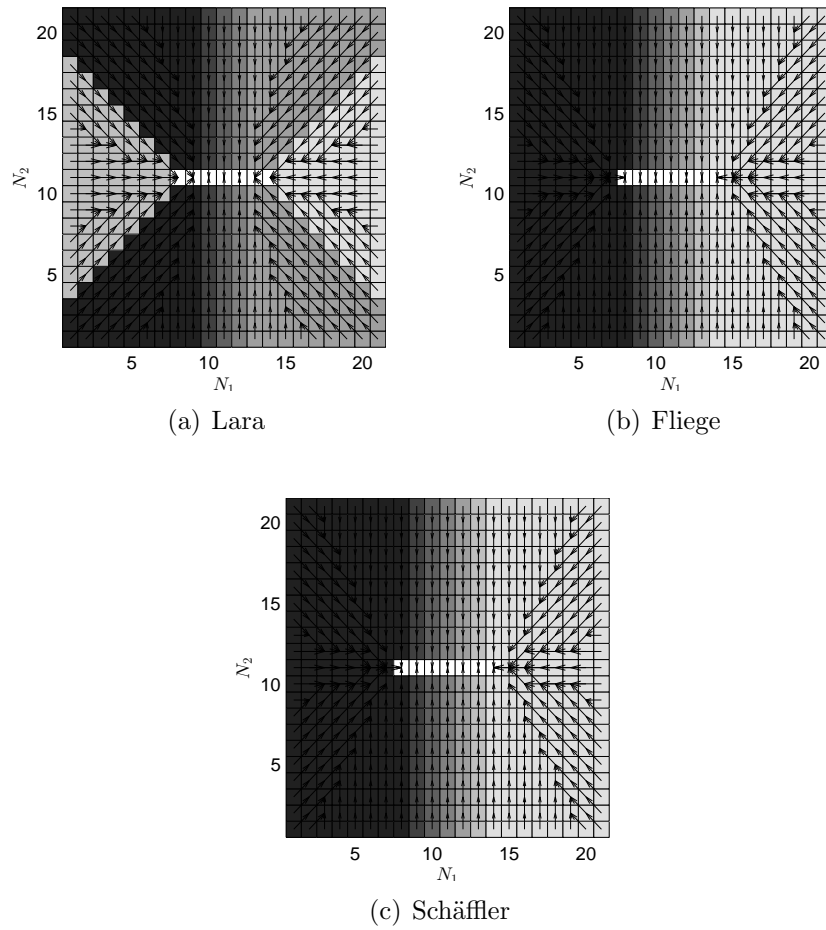
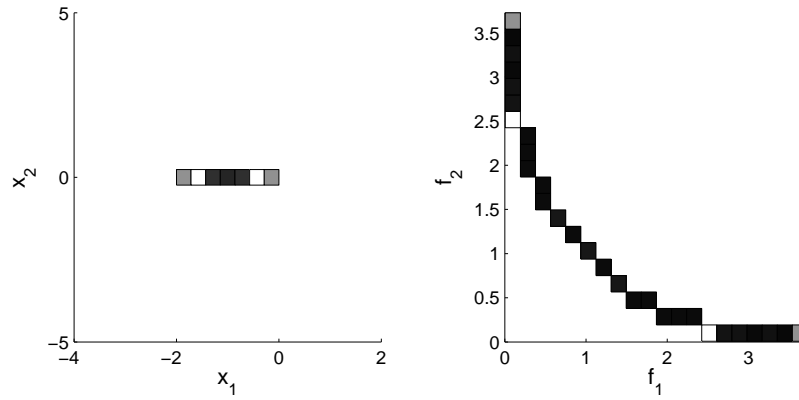
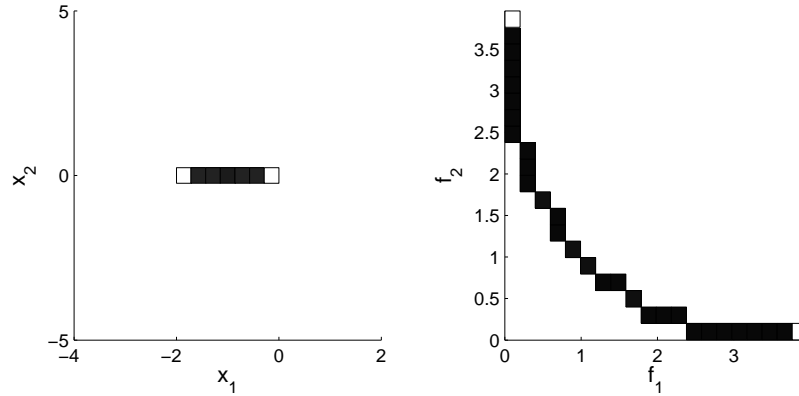


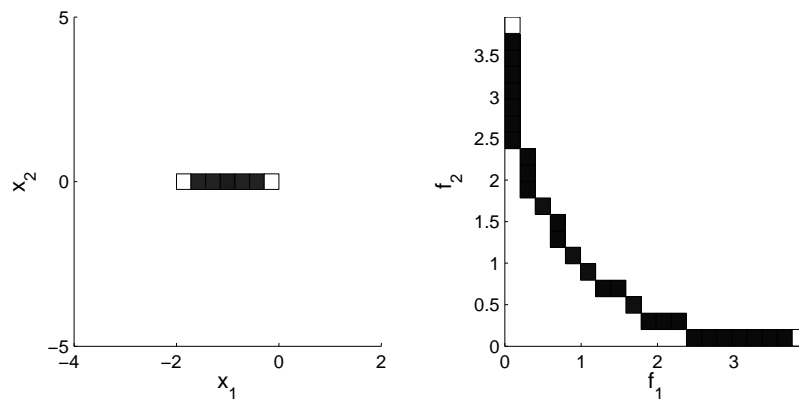
Figure 5.4: SCM on MOP (A.4) in cell space with different descent direction methods.



(a) Lara



(b) Fliege



(c) Schäffler

Figure 5.5: Comparison of three descent direction methods for MOP (A.4). We show parameter space (left) with $N_Q = 20 \times 20$ and objective space (right) $N_{F(Q)} = 20 \times 20$.

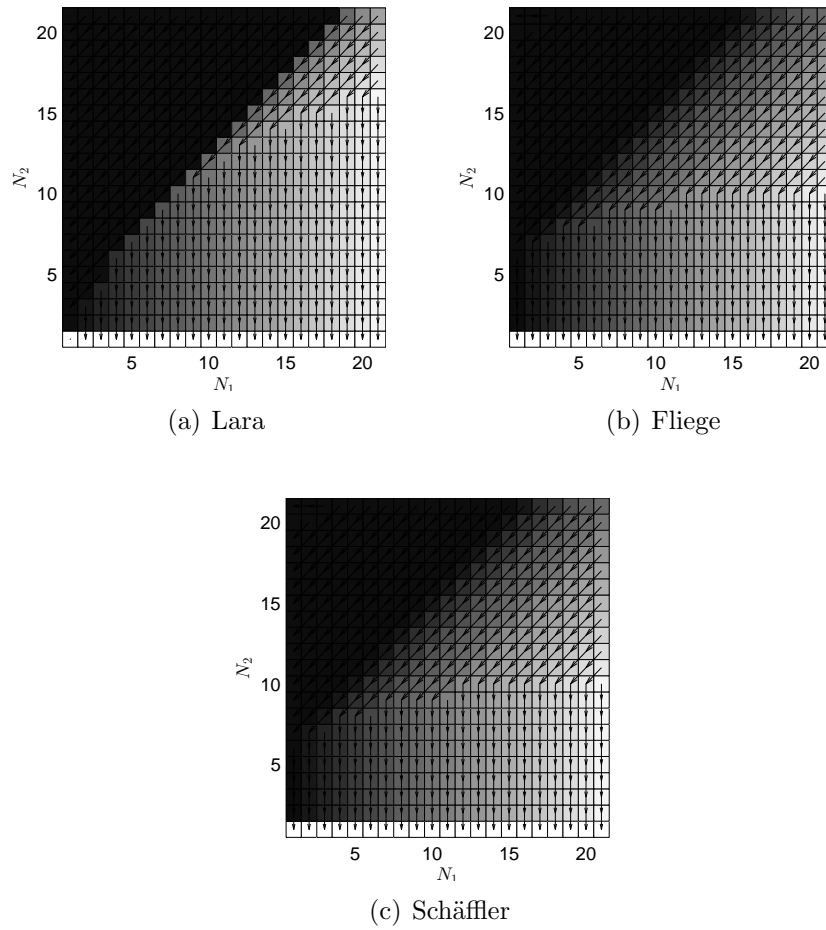
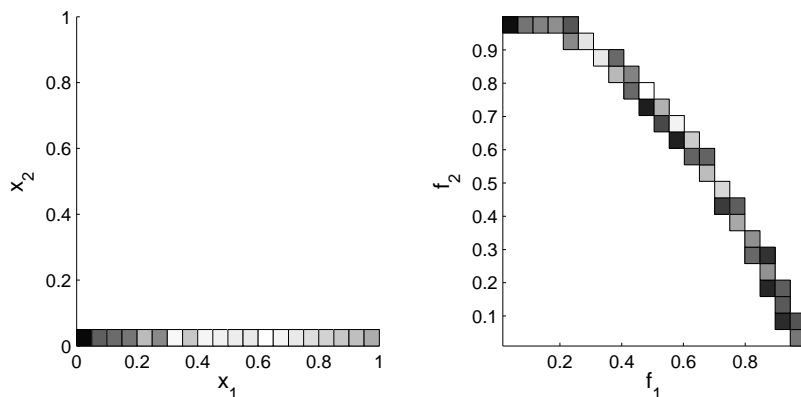
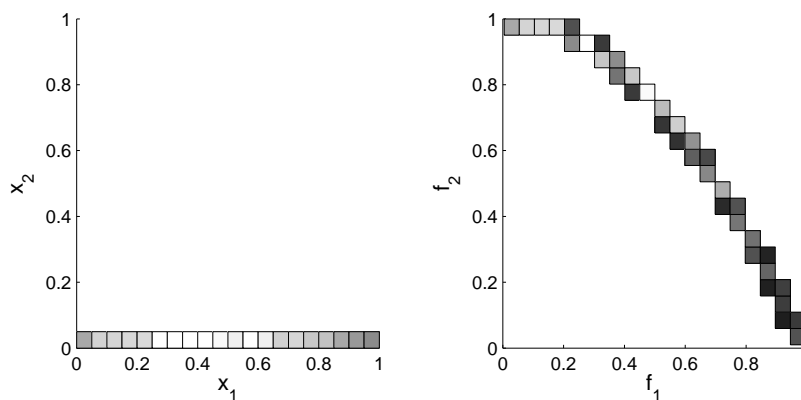


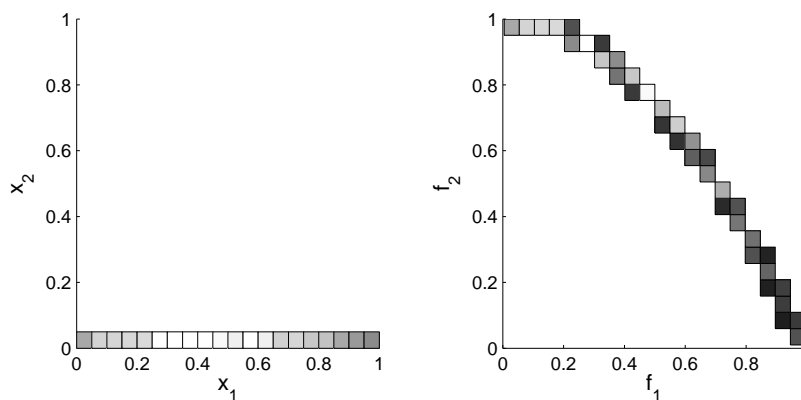
Figure 5.6: SCM on MOP (A.5) in cell space with different descent direction methods.



(a) Lara



(b) Fliege



(c) Schäffler

Figure 5.7: Comparison of three descent direction methods for MOP (A.5). We show parameter space (left) with $N_Q = 20 \times 20$ and objective space (right) $N_{F(Q)} = 20 \times 20$.

Chapter 6

Refine and Recover

One of the potential drawbacks of the SCM is given by the size of the cells. The choice of the size of the cells is critical for the algorithm, since cells too small may lead to a high number of function evaluations, while cells too big may lead to results that are not significant enough or even in the lost of information.

In this chapter, we address the problem of refining the solutions once the SCM has computed the optimal solutions (Section 6.1) and then we address the problem of how to recover optimal solutions once they have been discarded by the SCM due the discretization of the feasible region Q (Section 6.2).

6.1 Refine

Although there is a risk on using a big size of cell, the SCM can be used to compute a raw picture of promissory regions where optimal solutions can be found. With this preliminary result, now we can focus in these regions to perform a fine search. In this section, we present several methods that can be used in this context. First, we present a method that given the center points of each cell that belongs to an optimal solution, it searches for the best point within its cell (Section 6.1.1). Next, we present a variation of the SCM method that takes a subspace of the feasible region Q and then performs the SCM (Section 6.1.2). Further, we examine in how to use the subdivision techniques together with the SCM (Section 6.1.3). Finally, we present a comparison between the methods proposed in this section and the SCM for multi-objective optimization (Section 6.1.4).

6.1.1 Refine in Cell

As we mentioned previously the center point method has the disadvantage that the center point may not be significant of the whole cell. When a cell is marked as an optimal solution, it means that within the cell, we can find an optimal solution. However in the SCM, we always take the center point to represent the cell, thus it can be the case that there are better solutions rather than the center point.

The idea of this method is the following: Given the center point x_c of a cell and its boundaries lb_{cell} and ub_{cell} , we use the dynamical systems with lb_{cell} and ub_{cell} as box constraints and we use x_c as initial point. Thus, we will find a new point x_{c+j} after j steps such that $x_{c+j} \prec x_c$. Algorithm 8 shows the key elements of this approach called rinc.

Algorithm 8 Refine in Cell (rinc)

Require: *DynamicalSystem, cPs, maxiter, tolerance*

Ensure: *refined_solution*

```

1: refined_solution  $\leftarrow \{\}$ 
2: for all cell  $\in$  cPs do
3:   i  $\leftarrow 1$ 
4:    $x_c \leftarrow$  compute center of cell with Equation (3.1)
5:    $lb_{cell}, ub_{cell} \leftarrow$  compute lower and upper bounds of cell
6:   while  $i \leq maxiter$  and  $stopcriteria(x_c) > tolerance$  do
7:      $x_c \leftarrow DynamicalSystem(x_c)$ 
8:   end while
9:   refined_solution  $\leftarrow refined\_solution \cup \{x_c\}$ 
10: end for

```

This method has the property that it will output the same number of solutions as the input had. However, it has two potential drawbacks. First, although several optimal solutions may exist in the same cell but the method only give one for each cell. Further, due the step size control, periodic groups bigger than one may appear. Thus a cell marked as optimal solution may not in fact be one, in this case the method is not capable of detecting it and will give as output a point in the boundary of a cell.

6.1.2 SCM-Refine

As we can see in Algorithm 2, the SCM takes as input a feasible region bounded by box constraints lb and ub . We propose to give as input a set of feasible cells $cell_i$ such that $i \in n$ and everything outside this set is considered to be the sink cell. This idea allows the SCM to skip cells that are not interesting by a given criteria, i.e. the cell is not feasible.

Thus, once we have the result of the SCM using only box constraints, we can select the cells that are optimal solutions. Next, we divide these cells by a given factor c and then use the SCM only with these cells and compute a new set of optimal solutions. Algorithm 9 shows the key elements of this method (SCM-R).

This method has the advantages that we are able to compute more than one optimal solution for each cell. Also the method can be used several times until it meets a criteria. For instance, on the size of the cells or the number of solutions. Finally, this method compute not only the new set of optimal solutions but also the domains of attraction of the new set, which can be both an advantage of a drawback, if we are interested or not in the domains of attraction.

Algorithm 9 SCM-Refine

Require: $DynamicalSystem, cPs, div, h, N$ **Ensure:** cPs

```

1:  $S \leftarrow \{\}$ 
2:  $h_i \leftarrow h_i / div_i$ 
3:  $N_i \leftarrow N_i * div_i$ 
4:  $rang_i \leftarrow -div_i + 1, -div, \dots, 0, i = 1, \dots, |div|$ 
5:  $comb \leftarrow rang_1 \times \dots \times rang_{|div|}$ 
6: for all  $cell \in cPs$  do
7:    $z \leftarrow$  compute cell coordinates of  $cell$  with Algorithm (3)
8:    $corner_i \leftarrow z_i div_i, i = 1, \dots, |div|$ 
9:    $new\_cells \leftarrow comb_i + corner, i = 1, \dots, |div|$ 
10: end for
11:  $S \leftarrow S \cup new\_cells$ 
12:  $z, C, Gr, Pe, St, cPs \leftarrow SCM(DynamicalSystem, F, ub, lb, N, h, N_c, S)$ 

```

6.1.3 SCM with subdivision techniques

The subdivision techniques are an effective way to compute the optimal solutions of a given MOP and the subdivision techniques has also been proved to converge to the optimal solutions [40]. However, in the realization of the algorithm, when the cells are too big (for instance in the initial iterations), it is more difficult to decide if a given cell fulfills the selection criteria. In this section, we propose to use the SCM (SCM-ST) to preprocess the feasible region Q .

Again, we select the optimal solutions given by the SCM. Next, we convert the cells to a (center, radius) representation. Finally, we use the new set of cells as initial parameter for the subdivision techniques.

This method has also the advantage of computing more than one optimal solution for each original cell. Also, if we are not interested in the domains of attraction of the new set of optimal solutions, this method only computes the optimal solutions. Thus, it is cheaper in terms of memory than the previous method, since we do not compute the global properties of the dynamical system.

6.1.4 Comparison of the methods

In this section, we present a numerical comparison of the refinement methods presented above for MOP (A.1). A budget of 10,000 function evaluations was used as in Section 3.4. First a result with SCM was computed. The grid for rinc was set to $N = [35, 35]$ (7350 function evaluations) leading to 21 optimal solutions and for the other cases was set to $N = [20, 20]$ (2400 function evaluations) leading to 11 optimal solutions.

Next, for rinc 17 steps of the dynamical system were used for each optimal solution found. In the case of both SCM-ST and SCM-R a division factor of 2 was used and

the stopping criteria was set to be less than the limit of function evaluations to be used. SCM-ST does a search equivalent to a grid of $N = [640, 320]$ and SCM-R does a search to a grid of $N = [640, 640]$. If we would like to use SCM on grids of these size then we would use 1228800 and 2457600 respectively.

Figure 6.1 shows the results. Table 6.1 shows the Δ_p values of the candidate sets to P_Q and their image $F(P_Q)$. From the results, we can see that SCM-R and SCM-ST are similar. Also if we compare this results with Table 3.1 in page 40, we can see that the three refinement methods show better result than the SCM and further SCM-R and SCM-ST show better results than MOEA/D and NSGA-II.

Table 6.1: Δ_p values for the distances of the candidate solution set to $P_{Q,\epsilon}$ and $F(P_{Q,\epsilon})$, the best solutions in boldface.

	P_Q	$F(P_Q)$
Refine in Cell	0.1869	1.6355
SCM-R	0.1328	1.8150
SCM-ST	0.1328	1.8336

6.2 Recover

It may be the case that in the course of the SCM cells get lost, although they contain optimal solutions. For instance, this would be the case due discretization errors. We now describe a couple of algorithms using a process which allows to recover those optimal solutions that have previously been lost.

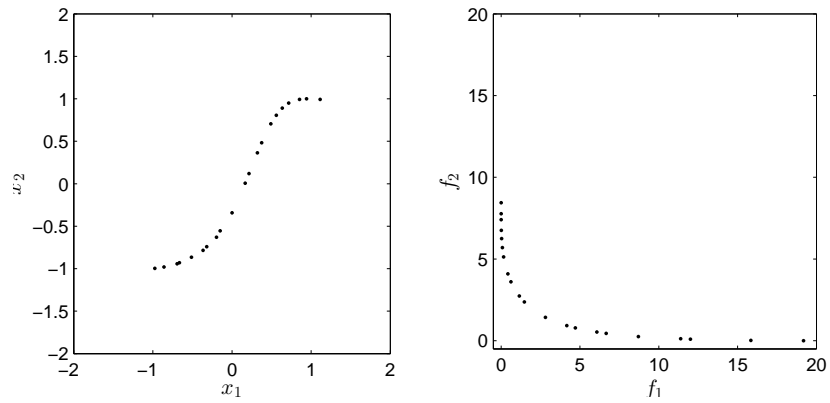
The idea behind these methods is the following: If the size of the cells is small enough, we expect that there would be neighboring boxes that also contain optimal solutions. Then we could perform a local search around these cells and keep those that are optimal solutions and we continue doing this while new solutions are found.

First, we look into a continuation method adapted to the SCM context (Section 6.2.2). Further, we look into a method that moves the boxes towards optimal solutions (Section 6.2.1). Finally, we present a comparison between these methods (Section 6.2.3).

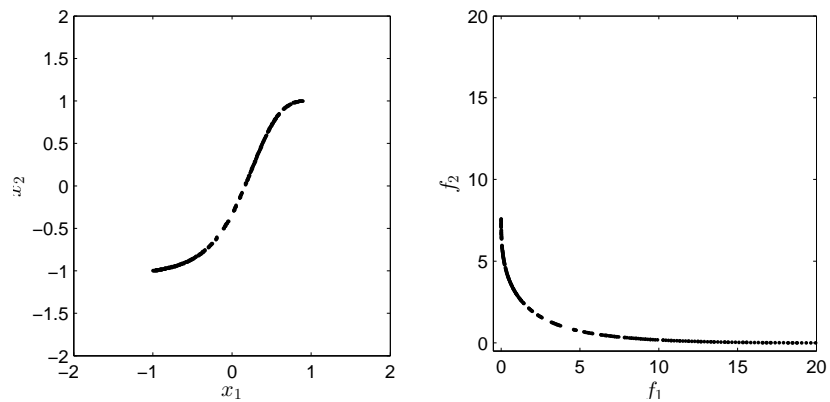
6.2.1 Moving Boxes

Given the set of optimal solutions found by SCM, we now choose for each cell several test points at random and then follow the dynamical system for one step. Then, we compute the cell where the new points end and add the new cells. This process is repeated until there are not any new cells. Algorithm 10 shows this process.

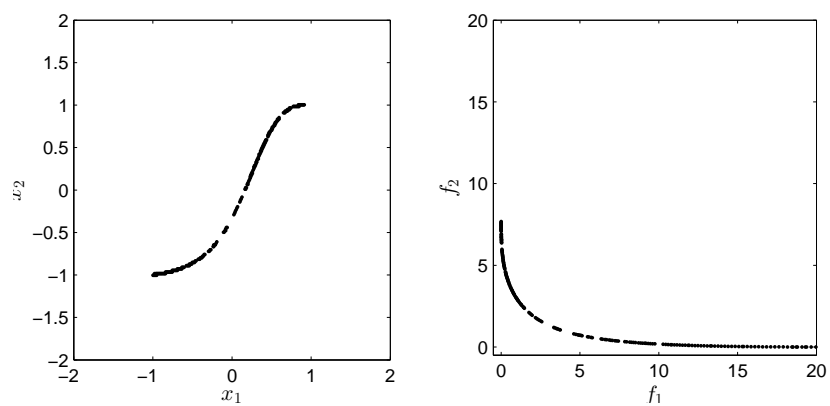
This method has the disadvantage that if big sections of the solutions are lost, then it is very likely that the method will not be able to recover all the solutions. Also, if the set cPs (in Algorithm 10) contains cells that does not belong to the set



(a) Refine in Cell



(b) SCM-Refine



(c) SCM with Subdivision Techniques

Figure 6.1: Numerical results for MOP (A.1). Black cells indicate Pareto optimal candidates.

Algorithm 10 Moving Boxes**Require:** $DynamicalSystem, cPsh, N, no_test$ **Ensure:** cPs

```

1:  $S \leftarrow \{\}$ 
2: for all  $cell \in cPs$  do
3:   for  $i$  to  $no\_test$  do
4:      $x_j \leftarrow rand(cell_{lb_j}, cell_{ub_j}), j = 1, \dots, |lb|$ 
5:      $x_{new} \leftarrow DynamicalSystem(x)$ 
6:      $new\_cell \leftarrow$  compute cell of  $x_{new}$  with Algorithm 4
7:     if  $new\_cell \notin cPs$  then
8:        $cPs \leftarrow cPs \cup new\_cell$ 
9:     end if
10:  end for
11: end for

```

of optimal solutions then the method would give as output a path to the optimal solution from the initial cell.

6.2.2 Continuation

Once we have the set of optimal solutions computed by the SCM, we would like to move along the Pareto front. We use a continuation method based on the directed search [17].

Algorithm 11 SCM Continuation**Require:** cPs **Ensure:** cPs

```

1: for all  $cell \in cPs$  do
2:    $x_c \leftarrow$  compute center of  $cell$  with Equation (3.1)
3:    $\alpha_{cell} \leftarrow$  compute  $\alpha_{cell}$ 
4:    $q, r \leftarrow QR - factorization(\alpha_{cell})$ 
5:    $directions \leftarrow q_i \cup -q_i, i = 2, \dots, |q|$ 
6:    $x_i \leftarrow DirectedSearch(x_c, directions_i), i = 1, \dots, |directions|$ 
7:    $new\_cells \leftarrow$  compute cell index of  $x_i$  with Algorithm 4
8:    $cPs \leftarrow cPs \cup new\_cells$ 
9: end for

```

6.2.3 Comparison of the methods

In this section, we present a numerical comparison of the recover methods presented above for MOP (A.1). First a result with SCM was computed and then to we remove

$\frac{2}{3}$ of the resulting cells to represent the possible lost of information. The grid for SCM was set to $N = [20, 20]$.

Figure 6.2 shows the results for moving boxes with 10 random points and for SCM-continuation. Table 6.2 shows the Δ_p values of the candidate sets to P_Q and their image $F(P_Q)$. From the results, we can see both are similar.

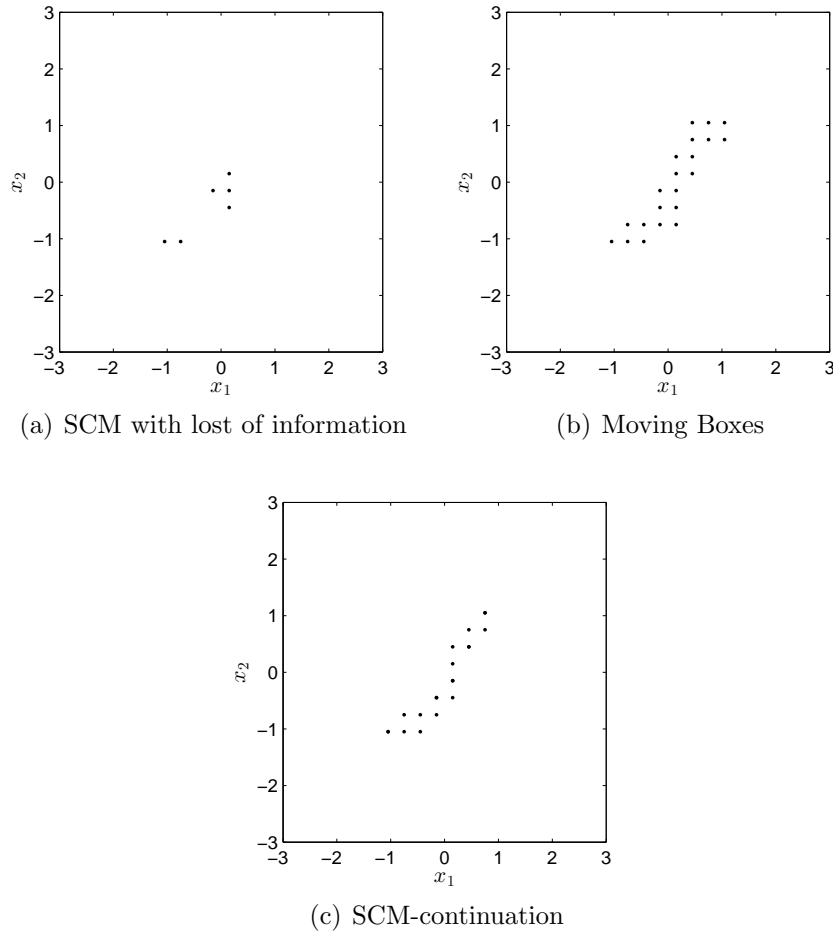


Figure 6.2: Numerical results for MOP (A.1). Black cells indicate Pareto optimal candidates.

Table 6.2: Δ_p values for the distances of the candidate solution set to $P_{Q,\epsilon}$ and $F(P_{Q,\epsilon})$, the best solutions in **boldface**.

	P_Q	$F(P_Q)$
Moving boxes	0.3536	1.8635
SCM-continuation	0.2550	2.6770

Chapter 7

Conclusions and Future Work

7.1 Conclusions

The main goal of this thesis has been to design set oriented methods for the numerical treatment of multi-objective optimization problems. The proposed approach was to adapt the simple cell mapping method to the given context. With this approach we obtained a thorough investigation for low dimensional problems.

We have used the simple cell mapping techniques to address the problem of computing the set of approximate solutions of a given multi-objective optimization problem. This set is of interest for the decision maker since it might enhance the set of options for him/her when compared to the set of optimal solutions, the Pareto set. To compute the set of approximate solutions we have adapted cell mapping techniques that were originally designed for the global analysis of dynamical systems. Since methods of that kind divide the search space into n -dimensional cells, where n is the dimension of the decision space of the MOP, they are well-suited for the problem at hand since they allow for an efficient approximation of the set of interest. We have tested the novel method on three academic functions and have compared it against two evolutionary methods. The results indicate that the cell mapping technique is able to reliably compute the set of approximate solutions, and is faster than the evolutionary approaches.

Further, we have made a comparison of some of the existing descent direction methods. This comparison is of particular interest for memetic evolutionary algorithms, since the results indicate that the studied methods have a potential bias towards the extreme points of the Pareto front. This should be taken to account when designing a memetic evolutionary algorithm based on this methods, since it can lead to unwanted issues as a lost on diversity.

Finally, we have also hybridized the simple cell mapping method with other techniques to partially overcome some of the drawbacks of the method. In particular we have addressed the refinement of the solutions once a prominent region has been found and the potential lost of information due discretization errors. The results show that this approach is highly competitive with the evolutionary algorithms for

low dimension problems.

7.2 Future Work

There are several points to be addressed. First of all, it would be desirable to extend the applicability of the method to higher dimensional problems. For this it seems promising to parallelize the simple cell mapping method.

Next, the integration of constraint handling techniques has to be addresses which has been left out in this study.

It would also be interesting to extend the refinement and recovering techniques to the other sets of interest such as approximate solutions and domains of attractions.

For our study, we have used academic models. However, we plan to apply the new method to real-wold engineering problems.

In the case of the comparison of descent direction methods, it would be desirable to improve it and to design a framework to help us study these methods and also to either improve them or to design a bias free descent direction method.

Finally, we plan to use the simple cell mapping to address robust multi-objective optimization, since it seems to be a good candidate given the thorough investigation provided by simple cell mapping.

Appendix A

Appendix A: Test Functions

Problem 1

This MOP taken from [48] that is given by two objective functions $f_1, f_2 : \mathbb{R}^2 \rightarrow \mathbb{R}$,

$$\begin{aligned} f_1(x_1, x_2) &= (x_1 - 1)^2 + (x_2 - 1)^4, \\ f_2(x_1, x_2) &= (x_1 + 1)^2 + (x_2 + 1)^2. \end{aligned} \tag{A.1}$$

Here, we have chosen for the domain $Q = [-3, 3] \times [-3, 3]$. The Pareto set P_Q forms a curve connecting the end points $x_1 = (-1, -1)^T$ and $x_2 = (1, 1)^T$. Figure A.1 shows the Pareto set and Pareto front.

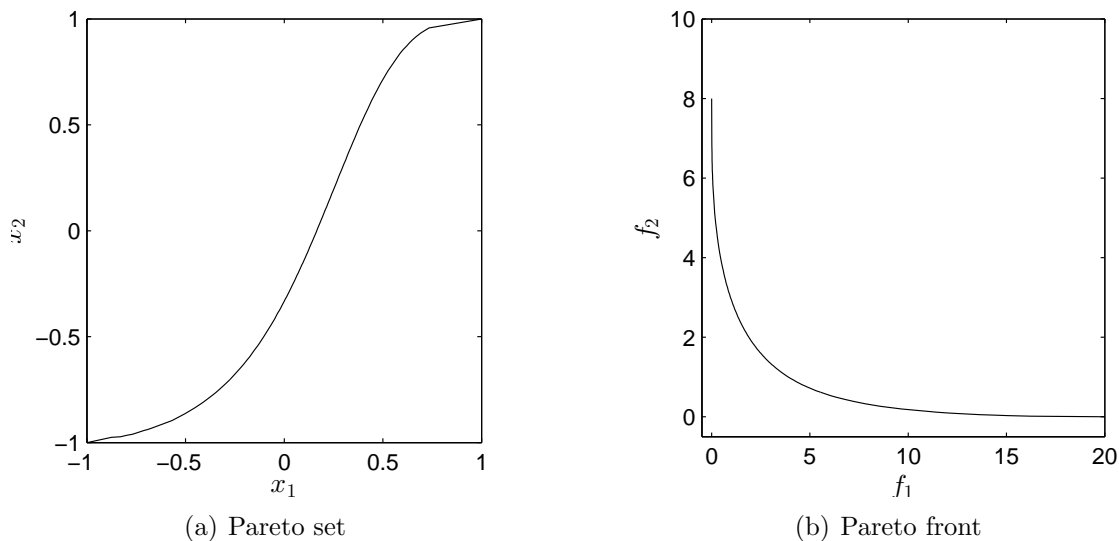


Figure A.1: Pareto set and Pareto front of MOP (A.1).

Problem 2

Next, we consider the problem $F : \mathbb{R}^2 \rightarrow \mathbb{R}^2$ proposed in [49].

$$F(x_1, x_2) = \begin{pmatrix} (x_1 - t_1(c + 2a) + a)^2 + (x_2 - t_2b)^2 + \delta_t \\ (x_1 - t_1(c + 2a) - a)^2 + (x_2 - t_2b)^2 + \delta_t \end{pmatrix}, \quad (\text{A.2})$$

where

$$t_1 = \text{sgn}(x_1) \min \left(\left\lceil \frac{|x_1| - a - c/2}{2a + c} \right\rceil, 1 \right), t_2 = \text{sgn}(x_2) \min \left(\left\lceil \frac{|x_2| - b/2}{b} \right\rceil, 1 \right),$$

and

$$\delta_t = \begin{cases} 0 & \text{for } t_1 = 0 \text{ and } t_2 = 0 \\ 0.1 & \text{else} \end{cases}.$$

Using $a = 0.5$, $b = 5$, $c = 5$ the Pareto set of MOP (A.2) is connected and further there exist eight other connected components that are locally optimal. For $\epsilon > 0.1$, the set $P_{Q,\epsilon}$ consists of nine sets that contain these components. Figure A.2 shows the Pareto set and Pareto front.

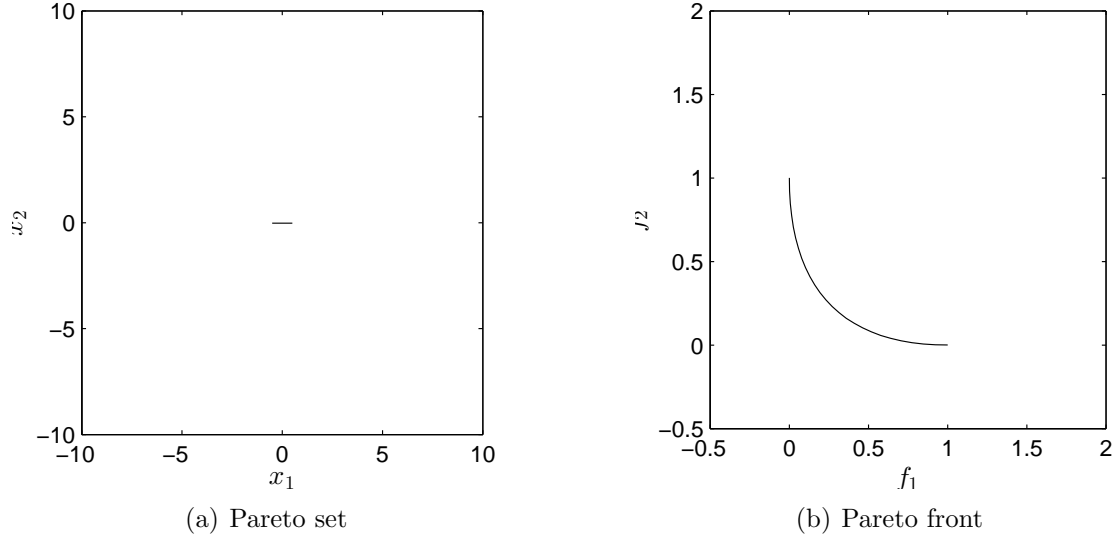


Figure A.2: Pareto set and Pareto front of MOP (A.2).

Problem 3

The production model proposed in [16] is defined as follows:

$$\begin{aligned}
 f_1, f_2 &: \mathbb{R}^n \rightarrow \mathbb{R} \\
 f_1(x) &= \sum_{j=1}^n x_j, \\
 f_2(x) &= 1 - \prod_{j=1}^n (1 - w_j(x_j)),
 \end{aligned} \tag{A.3}$$

where

$$w_j(z) = \begin{cases} 0.01 \cdot \exp(-(\frac{z}{20})^{2.5}) & \text{for } j = 1, 2 \\ 0.01 \cdot \exp(-\frac{z}{15}) & \text{for } j > 2 \end{cases}$$

Objective f_1 is related to the cost of a given product and f_2 to its failure rate. For the domain $Q = [0, 40]^3$ the Pareto set consists of four connected components. Figure A.2 shows the Pareto set and Pareto front.

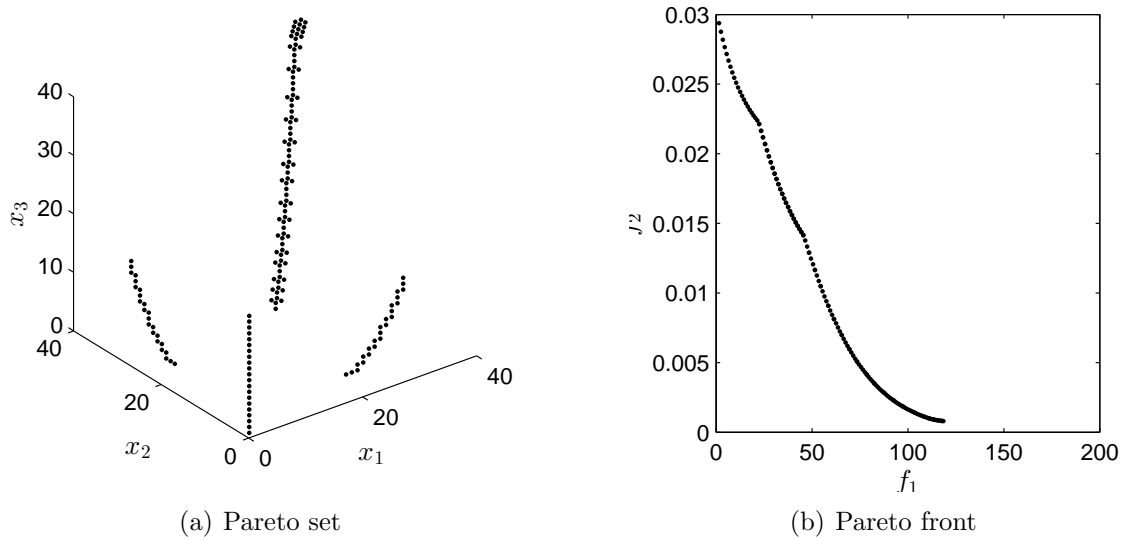


Figure A.3: Pareto set and Pareto front of MOP (A.3).

Problem 4

The model proposed in [50] is defined as follows:

$$\begin{aligned} f_1(x, y) &= x^2 + y^2, \\ f_2(x, y) &= (x + 2)^2 + y^2 \end{aligned} \tag{A.4}$$

Here, we have chosen for the domain $Q = [-5, 5] \times [-5, 5]$. The Pareto set P_Q forms

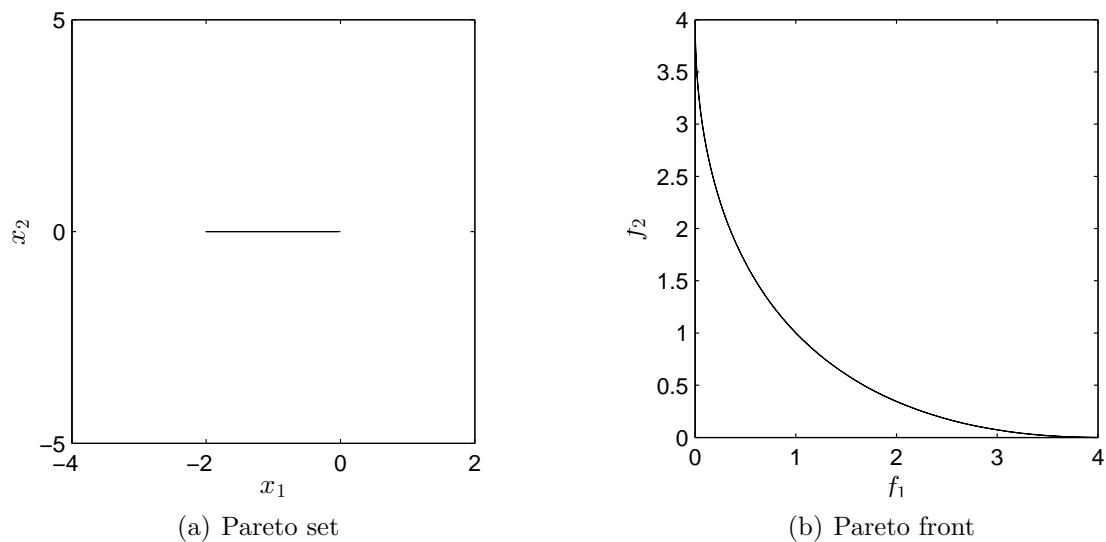


Figure A.4: Pareto set and Pareto front of MOP (A.4).

a curve connecting the end points $x_1 = (-2, -0)^T$ and $x_2 = (0, 0)^T$. Figure A.4 shows the Pareto set and Pareto front.

Problem 5

The model proposed in [51] is defined as follows:

$$\begin{aligned}
 f_1(x) &= x_1, \\
 f_2(x, g) &= g(x)h(x), \\
 &\text{and} \\
 g(x) &= 1 + x_2^2 \\
 h(x) &= \begin{cases} 1 & \text{if } f_1 \leq g, \\ 0 & \text{otherwise} \end{cases}
 \end{aligned} \tag{A.5}$$

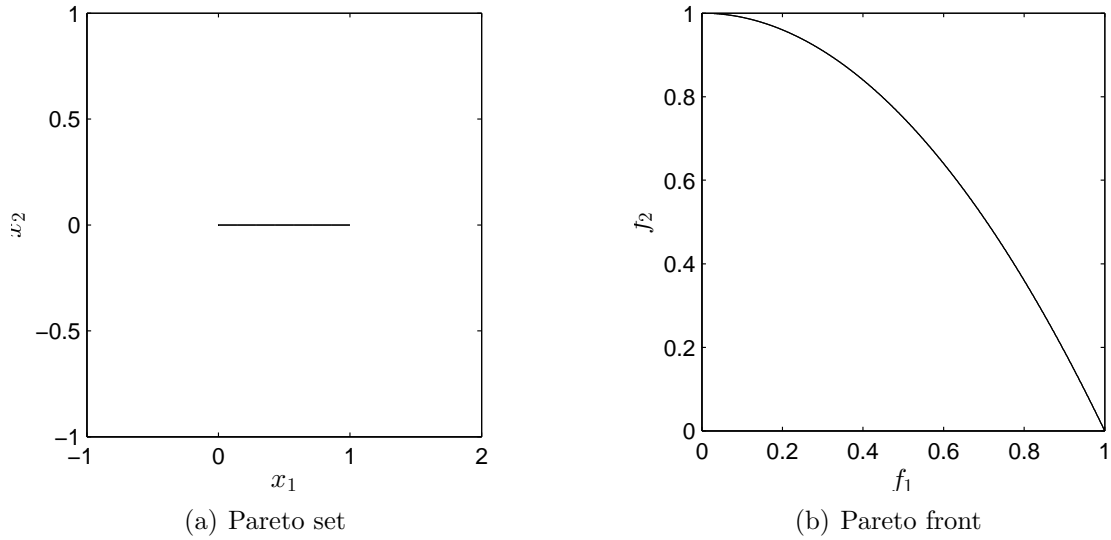


Figure A.5: Pareto set and Pareto front of MOP (A.5).

Here, we have chosen for the domain $Q = [0, 1] \times [0, 1]$. The Pareto set P_Q forms a curve connecting the end points $x_1 = (0, 0)^T$ and $x_2 = (1, 0)^T$. Figure A.5 shows the Pareto set and Pareto front.

Problem PID

We consider a second order oscillator subject to a proportional-integral-derivative (PID) control [52].

$$\ddot{x} + 2\zeta\omega_n\dot{x} + \omega_n^2x = \omega_n^2u(t), \quad (\text{A.6})$$

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

$$u(t) = k_p[r(t) - x(t)] + k_i \int_0^t [r(\hat{t}) - x(\hat{t})] d\hat{t} - k_d\dot{x}(t), \quad (\text{A.7})$$

$r(t)$ 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,

$$Q = \{\mathbf{k} \in [10, 50] \times [1, 30] \times [1, 2] \subset \mathbb{R}^3\}. \quad (\text{A.8})$$

Peak time and overshoot are common in time domain control design objectives [53, 54, 55]. We consider the multi-objective optimization problem to design the control gain \mathbf{k} ,

$$\min_{\mathbf{k} \in Q} \{t_p, M_p, e_{IAE}\}, \quad (\text{A.9})$$

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

$$e_{IAE} = \int_0^{T_{ss}} |r(\hat{t}) - x(\hat{t})| d\hat{t}. \quad (\text{A.10})$$

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 $T_{ss} = 20s$.

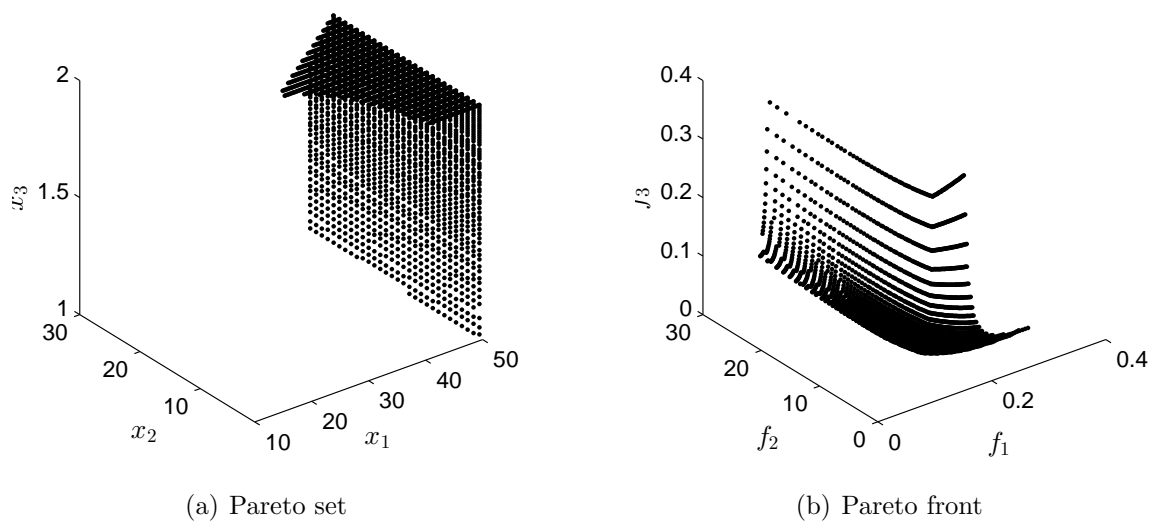


Figure A.6: Pareto set and Pareto front of MOP (A.6). Here, we have chosen for the domain $Q = [10, 1, 1] \times [50, 30, 2]$. Figure A.6 shows the Pareto set and Pareto front.

Bibliography

- [1] I. Das and J. Dennis. Normal-boundary intersection: A new method for generating the Pareto surface in nonlinear multicriteria optimization problems. *SIAM Journal of Optimization*, 8:631–657, 1998.
- [2] J. Fliege and B. F. Svaiter. Steepest Descent Methods for Multicriteria Optimization. *Mathematical Methods of Operations Research*, 51(3):479–494, 2000.
- [3] K. Klamroth and J. Tind and M. Wiecek. Unbiased Approximation in Multicriteria Optimization. *Mathematical Methods of Operations Research*, 56:413–437, 2002.
- [4] J. Fliege. Gap-free computation of Pareto-points by quadratic scalarizations. *Mathematical Methods of Operations Research*, 59:69–89, 2004.
- [5] E.L. Allgower and K. Georg. *Numerical Continuation Methods*. Springer, 1990.
- [6] C. Hillermeier. *Nonlinear Multiobjective Optimization - A Generalized Homotopy Approach*. Birkhäuser, 2001.
- [7] O. Schütze and A. Dell’Aere and M. Dellnitz. On Continuation Methods for the Numerical Treatment of Multi-Objective Optimization Problems. In Jürgen Branke and Kalyanmoy Deb and Kaisa Miettinen and Ralph E. Steuer, editor, *Practical Approaches to Multi-Objective Optimization*, number 04461 in Dagstuhl Seminar Proceedings. Internationales Begegnungs- und Forschungszentrum (IBFI), Schloss Dagstuhl, Germany, 2005. <<http://drops.dagstuhl.de/opus/volltexte/2005/349>>.
- [8] V. Pereyra. Fast computation of equispaced Pareto manifolds and Pareto fronts for multiobjective optimization problems. *Math. Comput. Simul.*, 79(6):1935–1947, 2009.
- [9] M. Dellnitz and O. Schütze and T. Hestermeyer. Covering Pareto Sets by Multi-level Subdivision Techniques. *Journal of Optimization Theory and Applications*, 124:113–155, 2005.
- [10] J. Jahn. Multiobjective search algorithm with subdivision technique. *Computational Optimization and Applications*, 35(2):161–175, 2006.

- [11] O. Schütze and M. Vasile and O. Junge and M. Dellnitz and D. Izzo. Designing optimal low thrust gravity assist trajectories using space pruning and a multi-objective approach. *Engineering Optimization*, 41(2):155–181, 2009.
- [12] K. Deb. *Optimization using Evolutionary Algorithms*. John Wiley & Sons, 2001.
- [13] C. Coello Coello and G. Lamont and D. Van Veldhuizen. *Evolutionary Algorithms for Solving Multi-Objective Problems*. Springer, Second edition, 2007.
- [14] O. Schütze and M. Laumanns and C. A. Coello Coello and M. Dellnitz and E.-G. Talbi. Convergence of Stochastic Search Algorithms to Finite Size Pareto Set Approximations. *Journal of Global Optimization*, 41(4):559–577, 2008.
- [15] C.S. Hsu. *Cell-to-cell mapping: A method of global analysis for nonlinear systems*. Applied mathematical sciences. Springer-Verlag, 1987.
- [16] S. Schäffler and R. Schultz and K. Weinzierl. A stochastic method for the solution of unconstrained vector optimization problems. *Journal of Optimization, Theory and Application*, 114(1):209–222, 2002.
- [17] O. Schütze and A. Lara and C. A. Coello. The Directed Search Method for Unconstrained Multi-Objective Optimization Problems. Technical Report COA-R1, CINVESTAV-IPN, 2010.
- [18] A. Lara López. *Using Gradient Based Information to build Hybrid Multi-objective Evolutionary Algorithms*. PhD thesis, 2012.
- [19] V. Pareto. *Manual of Political Economy*. The MacMillan Press, 1971 (original edition in French in 1927).
- [20] K. M. Miettinen. *Nonlinear Multiobjective Optimization*. Kluwer Academic Publishers, 1999.
- [21] P. Loridan. ϵ -Solutions in Vector Minimization Problems. *Journal of Optimization, Theory and Application*, 42:265–276, 1984.
- [22] O. Schütze and C. A. Coello Coello and E.-G. Talbi. Approximating the ϵ -efficient set of an MOP with stochastic search algorithms. In A. Gelbukh and A. F. Kuri Morales, editor, *Mexican International Conference on Artificial Intelligence (MICAI-2007)*, pages 128–138. Springer-Verlag Berlin Heidelberg, 2007.
- [23] H. Kuhn and A. Tucker. Nonlinear programming. *Proc. Berkeley Symp. Math Statist. Probability*, page 481–492, 1951.
- [24] Y.Y. Haimes and L.S. Lasdon and D.A. Wismer. On bicriterion formulation of the problems of integrated system identification and system optimization. *IEEE Transactions on Systems, Man, and Cybernetics*, 1(3):296–297, 1971.

- [25] N. Srinivas and K. Deb. Multiobjective function optimization using nondominated sorting genetic algorithms. *Evolutionary Computation Journal*, 2:221–248, 1995.
- [26] Q. Zhang and H. Li. MOEA/D: A Multi-objective Evolutionary Algorithm Based on Decomposition. *IEEE Trans. on Evolutionary Computation*, 11:712–731, 2007.
- [27] O. Schütze, M. Vasile, and C. A. Coello Coello. Approximate solutions in space mission design. In *Proceedings of the 10th International Conference on Parallel Problem Solving From Nature*, page 805–814, 2008.
- [28] E. Zitzler and L. Thiele and M. Laumanns and C. M. Fonseca and V. Fonseca. Performance Assessment of Multiobjective Optimizers: An Analysis and Review. *IEEE Transactions on Evolutionary Computation*, 7(2):117–132, 2003.
- [29] D. A. Van Veldhuizen. *Multiobjective Evolutionary Algorithms: Classifications, Analyses, and New Innovations*. PhD thesis, Department of Electrical and Computer Engineering. Graduate School of Engineering. Air Force Institute of Technology, Wright-Patterson AFB, Ohio, May 1999.
- [30] O. Schütze and X. Esquivel and A. Lara and C. A. Coello Coello. Using the averaged Hausdorff distance as a performance measure in evolutionary multi-objective optimization. *IEEE Transactions on Evolutionary Computation*, 16(4):504–522, 2012.
- [31] C. A. Coello Coello and Nareli Cruz Cortés. Solving Multiobjective Optimization Problems using an Artificial Immune System. *Genetic Programming and Evolvable Machines*, 6(2):163–190, 2005.
- [32] O. Schütze and M. Laumanns and E. Tantar and C. A. Coello Coello and E.-G. Talbi. Computing Gap Free Pareto Front Approximations with Stochastic Search Algorithms. *Evolutionary Computation*, 18(1).
- [33] C. S. Hsu. A discrete method of optimal control based upon the cell state space concept. *Journal of Optimization Theory and Applications*, 46(4):547–569, 1985.
- [34] F. H. Bursal and C. S. Hsu. Application of a cell-mapping method to optimal control problems. *International Journal of Control*, 49(5):1505–1522, 1989.
- [35] L. G. Crespo and J. Q. Sun. Stochastic Optimal Control of Nonlinear Dynamic Systems via Bellman’s Principle and Cell Mapping. *Automatica*, 39(12):2109–2114, 2003.
- [36] H. Flashner and T.F. Burns. Spacecraft momentum unloading: The cell mapping approach. *Journal of Guidance, Control and Dynamics*, 13:89–98, 1990.

- [37] W. H. Zhu and M. C. Leu. In *Planning optimal robot trajectories by cell mapping*, pages 1730–1735, 1990.
- [38] F.Y. Wang and P.J.A Lever. A cell mapping method for general optimum trajectory planning of multiple robotic arms. *Robotics and Autonomous Systems*, 12:15–27, 1994.
- [39] J. Y. Yen. Computer disk file track accessing controller design based upon cell to cell mapping. 1994.
- [40] M. Dellnitz and O. Schüetze. Multilevel Subdivision Techniques for Scalar Optimization Problems. In J.-Q. Sun, editor, *Global Analysis of Nonlinear Dynamics*, pages 221–252. Springer, 2012.
- [41] Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, Clifford Stein. *Introduction to Algorithms*. MIT Press, 2009.
- [42] E. I. Mejía Estrada. The Directed Search Method for Constrained Multi-Objective Optimization Problems. Master’s thesis, 2012.
- [43] M. Dellnitz and S. Ober-Blöbaum and M. Post and O. Schütze and B. Thiere. A multi-objective approach to the design of low thrust space trajectories using optimal control. *Celestial Mechanics and Dynamical Astronomy*, 105:33–59, 2009.
- [44] O. Schütze and M. Vasile and O. Junge and M. Dellnitz and D. Izzo. Designing optimal low thrust gravity assist trajectories using space pruning and a multi-objective approach. *Engineering Optimization*, 41:155–181, 2009.
- [45] O. Schütze and M. Vasile and C. A. Coello Coello. Computing the set of epsilon-efficient solutions in multiobjective space mission design. *Journal of Aerospace Computing, Information, and Communication*, 8:53–70, 2011.
- [46] M. Blesken and U. Rückert and D. Steenken and K. Witting and M. Dellnitz. Multiobjective Optimization for Transistor Sizing of CMOS Logic Standard Cells Using Set-Oriented Numerical Techniques. In *27th Norchip Conference*, 2009.
- [47] M. Blesken and A. Chebil and U. Rückert and X. Esquivel and O. Schütze. Integrated Circuit Optimization by Means of Evolutionary Multi-objective Optimization. In *Genetic and Evolutionary Computation Conference (GECCO-2011)*, pages 807–812, 2011.
- [48] O. Schütze. *Set Oriented Methods for Global Optimization*. PhD thesis, University of Paderborn, 2004. <<http://ubdata.uni-paderborn.de/ediss/17/2004/schuetze/>>.

- [49] G. Rudolph and B. Naujoks, and M. Preuss. Capabilities of EMOA to detect and preserve equivalent Pareto subsets. In *Proceedings of Evolutionary Multi-Objective Optimization (EMO2007)*, page 36–50.
- [50] R. Kumar and P. Rockett. Improved Sampling of the Pareto-Front in Multiobjective Genetic Optimizations by Steady-State Evolution: A Pareto Converging Genetic Algorithm. *Evolutionary Computation*, 10(3):2109–2114, 2002.
- [51] R. O. Day. *Explicit Building Block Multiobjective Evolutionary Computation: Methods and Application*. PhD thesis, 2005.
- [52] C. Hernández and Y. Naranjani and Y. Sardahi and W. Liang and O. Schütze and J.-Q. Sun. Simple Cell Mapping Method for Multiobjective Optimal PID Control Design. *International Journal of Dynamics and Control*, In review, 2013.
- [53] G. P. Liu and S. Daley. Optimal-tuning PID controller design in the frequency domain with application to a rotary hydraulic system. *Control Engineering Practice*, 7(7):821–830, 1999.
- [54] G. P. Liu and S. Daley. Optimal-tuning nonlinear PID control of hydraulic systems. *Control Engineering Practice*, 8(9):1045–1053, 2000.
- [55] S. Panda. Multi-objective PID controller tuning for a FACTS-based damping stabilizer using Non-dominated Sorting Genetic Algorithm-II. *International Journal of Electrical Power & Energy Systems*, 33(7):1296–1308, 2011.



**CENTRO DE INVESTIGACIÓN Y DE ESTUDIOS AVANZADOS
DEL INSTITUTO POLITÉCNICO NACIONAL**

Departamento de Computación

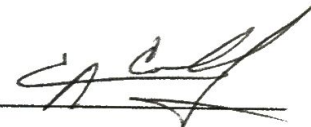
Después de haber efectuado la revisión del trabajo de tesis titulado

Cell-to-Cell Mapping for Global Multi-Objective Optimization

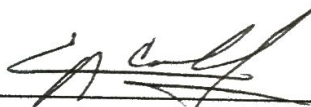
Realizado por el alumno Carlos Ignacio Hernández Castellanos, bajo la dirección del Doctor Oliver Steffen Schütze.

- 1 El documento final de tesis cumple con los requisitos para el Programa de Maestría en Ciencias.
- 2 Solicitamos que la fecha de Examen de Grado se lleve a cabo el día 14 de agosto a las 13:00 hrs.
- 3 Firman la presente el día 25 de julio del año 2013:

Oliver Steffen Schütze.

P.A. 

Dr. Carlos Artemio Coello Coello



Dr. Luis Gerardo de la Fraga

