Centro de Investigación y de Estudios Avanzados
del Instituto Politécnico Nacional

Unidad Zacatenco

Departamento de Computación

Tratamiento Numérico de Problemas Paramétricos Multi-Objetivo y Algoritmos Meméticos Evolutivos

Tesis que presenta

Víctor Adrián Sosa Hernández

para obtener el Grado de

Maestro en Ciencias

en Computación

Director de la Tesis:

Dr. Oliver Steffen Schütze

México, D. F. **Diciembre 2013**

Centro de Investigación y de Estudios Avanzados
del Instituto Politécnico Nacional

Campus Zacatenco

Computer Science Department

On the Numerical Treatment of Parametric Multi-Objective Optimization Problems and Memetic Evolutionary Algorithms

Submitted by

Víctor Adrián Sosa Hernández

as fulfillment of the requirement for the degree of

Master in

Computer Science

Advisor:

Dr. Oliver Steffen Schütze

Mexico, D. F. **December 2013**

Resumen

En un problema 'clásico' de optimización multi-objetivo (POM), la tarea principal es el optimizar varios objetivos al mismo tiempo. El conjunto solución de un POM, llamado conjunto de Pareto, típicamente forma un objeto de $(k-1)$ dimensiones, donde k es el número de objetivos involucrados en el problema. Un problema de optimización multi-objetivo dependiente de parámetros (POMP) se constituye de un problema multi-objetivo donde algunos parámetros $\lambda \in \mathbb{R}^l$ son externos y no pueden ser influenciados por el diseño de un objeto (por ejemplo, la dirección del viento en el diseño de un carro 'óptimo'). El conjunto solución de un POMP es en la mayoría de los casos de $k-1+l$ dimensiones. Recientemente, los algoritmos evolutivos basados en indicadores (AEBIs) han capturado el interés de muchos investigadores para tratar POMs, esto es debido a que tales algoritmos entregan una buena aproximación del conjunto de soluciones y usualmente tienen un mejor desempeño comparado con algoritmos basados en dominancia de Pareto. Sin embargo, estos métodos continúan teniendo la desventaja de necesitar un alto número de evaluaciones de la función objetivo para obtener un representación adecuada del conjunto de soluciones.

El alcance de este trabajo de tesis es el tratamiento numérico de POMs y POMPs utilizando búsqueda local y estrategias meméticas. Para llevar a cabo esto, el reciente desarrollado Método de Búsqueda Dirigida (MBD) (en ingles 'Directed Search Method (DS)') será adaptado a cada contexto. En el caso de POMPs, se adaptará el MBD para realizar un movimiento sobre el espacio λ, contando o no con la información del gradiente. Finalmente para los POMs, el objetivo es adaptar el MBD dentro del contexto de aproximaciones utilizando el hipervolumen, y de esa manera solucionar el problema de la 'lenta convergencia'. Se presentará un nuevo algoritmo de búsqueda local, el cual será integrado dentro de un algoritmo basado en el hipervolumen, para dar lugar a un nuevo algoritmo memético. Mostraremos que estas nuevas estrategias puede ser altamente competitivas considerando algoritmos del estado del arte sobre estos campos.

Abstract

In a 'classical' multi-objective optimization problem (MOP) the task is to optimize several conflicting objectives concurrently. The solution set of a MOP, the so-called Pareto set, typically forms a $(k-1)$-dimensional manifold, where k is the number of objectives involved in the MOP. A parameter dependent multi-objective optimization problem (PMOP) consists of a MOP where in addition several parameters $\lambda \in \mathbb{R}^l$ are external and cannot be influenced for the design of an object (e.g., the side wind in the design of an 'optimal' car). The solution set of a PMOP is typically $k-1+l$ dimensional. Recently, indicator based evolutionary algorithms (IBEAs) have caught the interest of many researchers for the treatment of MOPs, since they deliver the desired approximation of the solution set and due to a usually better performance compared to dominance based algorithms. Nevertheless, these methods still suffer the drawback that many function evaluations are required to obtain a suitable representation of the solution set.

The scope of this thesis project is the numerical treatment of both MOPs and PMOPs by means of local search and memetic strategies. For this, the recently developed Directed Search method will be adapted to the given contexts. In the case of PMOPs, we adapt the DS to perform a movement over λ-space with and without gradient information. Finally, for MOPs we aim to adapt the DS for the context of hypervolume approximations to overcome the problem of 'slow convergence'. We present a new local search algorithm and a new memetic algorithm using the SMS-EMOA. We will show that the novel local search strategies and the new memetic strategy will be highly competitive to state-of-the-art algorithms in these fields.

Agradecimientos

Primero que nada quiero agradecerle a Dios, por cada día durante estos dos años de mi vida, en los cuales me permitió superarme para lograr mis metas y seguir haciendo mis sueños realidad.

Doy las gracias al CONACyT por la beca que me brindó para realizar mis estudios de maestría, así como también por el apoyo recibido para realizar una estancia de investigación en el extranjero.

Igualmente quiero agradecerle al CINVESTAV-IPN y en especial al departamento de computación por aceptarme en su programa de maestría. Le doy gracias a mis profesores por compartir sus conocimientos conmigo, al personal administrativo Erika, Sofy y Felipa por realizar siempre un excelente trabajo y por ayudarme en cada trámite que tuve que hacer, también agradezco a los auxiliares de investigación por su ayuda y apoyo dentro del departamento. Finalmente, a mis amigos y compañeros dentro del departamento les doy gracias por su compañía, consejo y buenos momentos.

Quiero darle un especial agradecimiento al Dr. Oliver Schütze, por la oportunidad que me brindó de ser su tesista y alumno durante la maestría. Le doy las gracias por su guía durante el primer año de maestría siendo su tutorado y posteriormente por brindarme en todo momento su apoyo para la realización de este proyecto de tesis. Gracias por su tiempo y dedicación para revisar este trabajo.

Agradezco a la Dra. Heike Trautmann (Universidad de Müster) y al Dr. Günter Rudolph (TU-Dortmund) por darme la oportunidad de realizar una estancia de investigación en la Universidad de Dortmund (TU-Dortmund) en Alemania. Gracias por su valiosa ayuda y consejo en la realización de este trabajo.

Quiero darle las gracias a los doctores Luis Gerardo de la Fraga y Amilcar Meneses Viveros por formar parte del comité evaluador de mi trabajo de tesis. Sus comentarios ayudaron a mejorar este trabajo escrito.

A mis padres Isabel y Víctor por su amor, apoyo, y consejo que me han dado

durante estos 24 años que llevo en este mundo. Nunca terminaré de agradecerles y pagarles todo lo que han hecho por mí, por esa razón tengan por seguro que seguiré creciendo como persona, y me seguiré superando día tras día para seguir llenándolos de orgullo y satisfacciones.

Le doy las gracias a mis hermanos Andrés y Valeria, por su apoyo, ya que ellos también son una razón para yo seguir adelante. No dejen de luchar y superarse nunca.

A toda mi familia, abuelos, tíos y primos les agradezco ya que durante toda mi vida me han acompañado y enseñado muchas cosas importantes.

Agradezco a Pamela por su ayuda, apoyo, amistad y consejo. Eres mi mejor amiga.

Contents

List of Figures

List of Tables

List of Algorithms

List of Acronyms

SOP scalar optimization problem

MOP multi-objective optimization problem

PMOP parameter dependent multi-objective optimization problem

DS directed search method

PC predictor-corrector method

EA evolutionary algorithm

NBI normal boundary intersection method

CHIM convex hull of individual minima

MOEA multi-objective evolutionary algorithm

IBEA indicator-based evolutionary algorithm

HVDS hypervolume based on the directed search method

AHD averaged Hausdorff distance

SMS-EMOA S metric selection evolutionary multi-objective algorithm

IVP initial value problem

1 | Introduction

All along, humankind has the desire for being always better according to what has been done in the past, but getting better is not a simple task. Nowadays, if one wonders what 'better' is, the answer would not come quickly to our minds. Likely, it would be crucial to define firstly one or more features over which the decision will be done in order to select the 'best' among two or more elements. As an example, one can imagine the process to construct a motorcycle. Such a vehicle involves many characteristics regarding its construction, but for now 'quality' and 'cost' have been chosen to convey our ideas in a straightforward manner.

Figure 1.1: 2013 Honda CBR 1000 RR. This image was taken from [1].

Honda^{TM} for instance is one of the most important motorcycle makers that produces a certain number of motorcycle models, each model has different features in-

volved that depend on the quality of the materials that are made. In order to find a proper way to produce motorcycles, one has to look carefully at that process. People, who are in charge of the whole production process, are interested in how their motorcycles would be attractive for their clients due to their quality, but without forgetting the production cost that will affect directly the profit. So, by selecting one of the produced models (in this case the one in Figure 1.1), we notice that our selected features might be seen now as goals for the company. Analyzing deeply the problem, we realize that both goals are in conflict, what generates two different settings. The first setting is when the production cost is reduced in a significant way, therefore, the motorcycle quality will be punished in order to obtain a cheaper product. This possible setting will provoke higher profit by one sale but also losses for the company since people may look for another motorcycle with better quality. Otherwise, in the second setting, the aim is to maximize quality. This will cause that the production cost increase what represents fewer profit, as well as, the motorcycle cost will not be affordable for most of the clients which generates less sales. Due to the above explanations our decision becomes difficult. The latter is because, we can see that there is not only one solution for our problem but rather an entire set of solutions. Summarizing the previous ideas, an important question comes now to our minds, 'how should we construct the motorcycle in order to benefit the company by increasing the profit and the sales in the market?'. To answer that question, it is important first to have a set of good solutions and the 'how' is answered by one field called *optimization*.

Arguably, optimization may help us to find elements which are the best choices of a given problem. For instance, it would find proper solutions regarding the balance between cost and quality in the previous example. We can say that optimization is an important and active field for research, since it plays an important role in everyday life, for example in activities such as decision-making, designing of goods like our example, system control and others. Two important sub-fields of optimization are: (i) single-objective optimization which tries to solve problems with only one objective under some circumstances and (ii) multi-objective optimization that is when a problem has multiple objectives which are in conflict with each other and all objectives have to be optimized concurrently. The latter is defined as a multi-objective optimization problem (MOP).

Due to its importance and relation with many things in the real world, it is im-

portant to improve the knowledge related to optimization. Here, we mention some application domains that involve MOPs: engineering applications (electrical , aeronautical , robotics and control), industrial applications (scheduling, management, design and manufacture), scientific applications (chemistry, physics, medicine and computer science), etc [2].

Entering deeper into the subject, it is known that the solution set of a MOP, the so-called Pareto set, typically forms a $(k-1)$-dimensional manifold, where k is the number of objectives involved in the problem [3]. Currently, there are many methods and techniques to tackle MOPs, such as scalarization methods [4, 5], subdivision techniques [6], and the so-called multi-objective evolutionary algorithms (MOEAs) [7, 8]. For the treatment of MOPs especially MOEAs have caught the interest of many researchers (see, e.g., book of Deb [9] and references therein), since they are applicable to a wide range of problems. Other remarkable feature is their global nature what makes the algorithm in principle not dependent on the initial candidate set. Finally, we can say that they typically allow us to compute a finite size representation of the Pareto set in a single run of. Among MOEAs, there is a recent trend in the design of algorithms that are based on a particular performance indicator, the so-called indicator-based evolutionary algorithms (IBEAs) [10, 11]. Reasons for that include the improvement of the numerical treatment of the problem (e.g., the speed up of the convergence rate) and the fact that such optimal archives (i.e., optimal w.r.t. the given indicator) are in certain cases most appropriate for the related decision making problem.

On the other hand, it is known that MOEAs tend to converge slowly, what is a severe drawback of this kind of algorithms. That drawback leads to use a relatively high number of function evaluations to obtain a suitable representation of the set of interest. As a possible remedy, researchers have proposed *memetic strategies* in the recent past (e.g., [12]). Memetic strategies hybridize local search strategies mainly coming from mathematical programming with MOEAs in order to obtain fast and reliable global search procedures. Ever since memetic strategies were proposed and many researchers have reported successful results, nevertheless, questions remain open, what keeps researchers working on this field.

Nowadays, optimization is still growing since other kinds of problems are arising related to MOPs, such as parameter dependent multi-objective optimization problems

(PMOPs) which occur in many real world applications. Problems of this kind depend on an external parameter $\lambda \in \mathbb{R}^l$ which describes the influence of the other parameters in the objective functions [13, 14]. The latter motivates researchers to develop new tools in order to tackle them.

1.1 Motivation

There is a necessity for developing more efficient algorithms to tackle both MOPs and PMOPs, since such problems are related to real world applications.

For MOPs a faster and reliable way to approximate the solution set is required, since in many situations the cost to evaluate a model is expensive computationally speaking. Memetic strategies, as we mentioned, have been developed in order to reach this goal. However, to find the proper balance between global and local search becomes in a difficult task. The latter is because, when a local search is performed, a significant improvement is expected to justify the effort.

Next to MOPs, by looking at the literature, we noticed that for PMOPs there is a lack of methods to approximate the entire solution sets. The state-of-art algorithms are not able to compute the solution sets in only one run but in multiple runs by changing the value of the external parameter. This approach leads to the consumption of a huge amount of function evaluations making the process difficult and expensive.

At present, there are many methods to help multi-objective evolutionary algorithms (MOEAs), among of them a novel method was recently proposed for the numerical treatment of MOPs. This method is called the 'Directed Search (DS)' [15] and contains two parts: A descent method that performs an iterative procedure to obtain a curve of dominating points until a boundary point is reached. Further on a new continuation method which allows to search along the Pareto set of a given MOP. The latter has an important advantage, namely that it does not require any second gradient information which allows to use DS for designing a new local search strategy to hybridize evolutionary approaches. In fact, the DS can be realized gradient free.

So, as we see, DS would be integrated into evolutionary algorithms to improve their performance, as well as, it will be interesting to adapt this method for more problems in order to become it more robust. Thus, along this work, the design of

new methods is aimed for providing new tools to treat MOPs and PMOPs in a clever way. The integration of local search procedures into MOEAs is our main focus in this work, since by this it is expected to increase the rate of convergence of the resulting algorithms.

1.2 The Problem

The scope of this thesis is to work with a certain kind of evolutionary approaches, the so-called indicator-based evolutionary algorithms which were briefly explained in the introduction. Algorithms of this sort use an indicator which is in simple words is a tool to measure the approximation quality of the obtained solution set. Within IBEAs the indicator not only assesses the final approximation of the algorithm but also helps the algorithm to select such elements which contribute most to its value. Since to hybridize EAs, in this case IBEAs, with local search strategies is our main objective, the difficulty of the problem increases. In other words, to improve the value according to a given indicator while the local search is being applied will be as our main problem.

Finally, the DS has not been adapted for PMOPs, in order to extend their applicability, that is why, we aim for performing a movement toward and along $\lambda-$space into a PMOP. To achieve the previous aim represents a big challenge for us, since the underlying theory has to be developed. To emphasize the presented challenges for this work, we list them below:

- Design a local search technique for IBEAs using DS.
- Consider the external parameter $\lambda \in \mathbb{R}^l$ in order to extend the theory related to DS to tackle PMOPs.
- Design a predictor-corrector method that will be capable of moving in the 'λ-direction' of a given PMOP.
- Consider for both MOP and PMOP models which do not provide first or second derivative information.

1.3 General and Particular Aims

General Aim

To design a novel memetic strategies that are able to treat MOP and PMOP, in order to advance the state-of-the-art.

Particular Aims

- To adapt DS for PMOPs with and without gradient information.
- To design a novel continuation method both along f-space and λ-space for PMOPs.
- To design a new local search technique using DS for an IBEA.
- To integrate the new local search technique into a general IBEA in order to improve its performance.

1.4 Final Contributions

In order to summarize the work done along this thesis project, we list our contributions below:

- Local search algorithms for the treatment of MOPs and PMOPs
  - The Directed Search Method for PMOPs (λ-DS)
  - The Hypervolume based Directed Search (HVDS)
- Memetic multi-objective evolutionary algorithms
  - Memetic version of the S metric selection evolutionary multi-objective algorithm (SMS-EMOA) using HVDS only at initial stages
  - Memetic version of the SMS-EMOA using HVDS as an operator
- Numerical results and comparison against the state-of-the-art methods

- Collaboration with the TU-Dortmund University, Germany
- Contribution at the international conference 'Genetic and Evolutionary Computation Conference' (GECCO) 2013 in Amsterdam, The Netherlands: Victor Adrian Sosa-Hernandez, Oliver Schütze, Günter Rudolph, Heike Trautmann. Directed Search Method for Indicator-based Multi-Objective Evolutionary Algorithms, pp 1699-1702, 2013
- Contribution at the international conference 'EVOLVE- A bridge between Probability, Set Oriented Numerics, and Evolutionary Computation' 2013 in Leiden, The Netherlands: Victor Adrian Sosa-Hernandez, Oliver Schütze, Günter Rudolph, Heike Trautmann. The Directed Search Method for Pareto Front Approximations with Maximum Dominated Hypervolume, pp 189-205, 2013

1.5 Organization of the Thesis

The remainder of this thesis is organized as follows: In Chapter 2, we state some theoretical background to understand the presented work. Furthermore, we review some methods used to solve multi-objective optimization problems. Then, we introduce the Directed Search Method. Finally, we present the related work to PMOPs. In Chapter 3, we introduce the underlying ideas of the λ-DS and we also propose a modification of the λ-DS that is gradient free. We explain the influence of the external parameter λ and also this chapter is used to show some numerical results for the λ-DS with gradient information. Chapter 4 is used to explain the design of the new local search strategy called HVDS. We present, how it is integrated into an IBEA and first numerical results. Finally, we draw some conclusions and give some paths for future work in Chapter 5.

2 | Background

In this chapter, we provide the required background and related work to understand the ideas presented in this thesis project. We start by describing important concepts related to the theoretical background. Next to this, we present the state-of-the-art that involves the main techniques to solve MOPs, as well as, we dedicate this chapter also to list some tools to measure the performance over a generated approximation. Then, we present what a memetic strategy is and we mention some works related to PMOPs to get more acquainted to them. Finally, we present the local search technique used throughout this work, the Directed Search Method.

2.1 Theoretical Background

In this section, we start to talk about single-objective optimization problems to collect enough knowledge to face the problems which are going to be treated in this thesis, MOPs and PMOPs. Next, we are going to define these problems to know about what we have to do to solve them. Since to define Pareto optimality is very important in order to select such elements that are better to solve the problems, we also present the related theory.

2.1.1 Single-Objective Optimization Problem

First of all, it is important to know what a single-objective optimization problem (SOP) is, in order to understand MOPs and PMOPs. A SOP is defined as follows:

Definition 1. *A general single-objective optimization problem is defined as minimi-*

zing (or maximizing) a function as follows:

$$\min_{\mathbf{x}\in S} f(\mathbf{x}) \tag{2.1}$$

$$\begin{aligned} s.t \;\; g_i(\mathbf{x}) \leq & 0 \;\; i = 1, \ldots, p \\ h_j(\mathbf{x}) = & 0 \;\; j = 1, \ldots, q. \end{aligned}$$

where $f : \mathbb{R}^n \to \mathbb{R}$, $g_i(x) : \mathbb{R}^n \to \mathbb{R}$, *and* $h_i(x) : \mathbb{R}^n \to \mathbb{R}$. $S \subset \mathbb{R}^n$ *is defined as the feasible region which is restricted by the constraints*

$$S = \{\mathbf{x} : g_i(\mathbf{x}) \leq 0 \; and \; h_j(\mathbf{x}) = 0\}. \tag{2.2}$$

It is important to mention that for the remainder of this thesis, we only will instanciate unconstrained minimization problems. Given a function $f : S \subseteq \mathbb{R}^n \to \mathbb{R}$, the value $f(\mathbf{x}^*)$ is called a global minimum if and only if

$$f(\mathbf{x}^*) \leq f(\mathbf{x}) \;\; \forall \mathbf{x} \in S. \tag{2.3}$$

Commonly, a single-objective optimization problem may have a unique solution as we show in Figure 2.1.

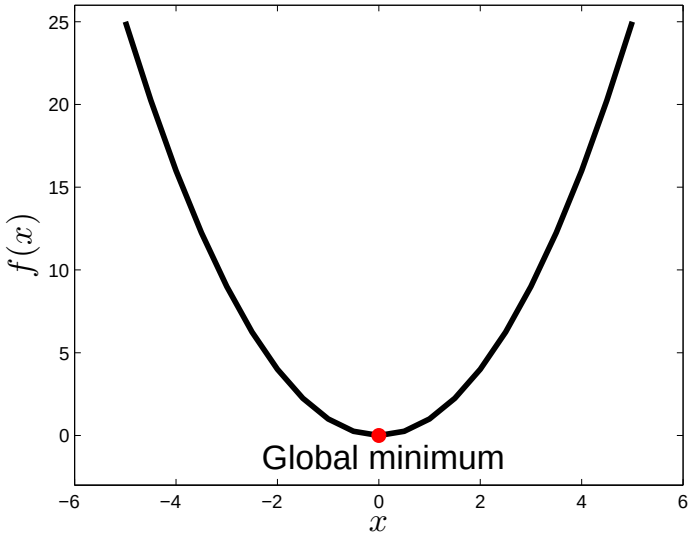


Figure 2.1: Example of a single-objective optimization problem with a unique solution.

2.1.2 Multi-Objective Optimization Problem

In the following, we define a continuous multi-objective optimization problem:

$$\min_{\mathbf{x}\in S}\{F(\mathbf{x})\}, \tag{2.4}$$

where $S \subseteq \mathbb{R}^n$ is the feasible region, which is a subset of the decision variable space and the function F is defined as the vector of objectives functions

$$F : S \to \mathbb{R}^k, \;\; F(\mathbf{x}) = (f_1(\mathbf{x}), \ldots, f_k(\mathbf{x})), \tag{2.5}$$

where $k \geq 2$ and where each objective $f_i : S \to \mathbb{R}$ is for sake of simplicity sufficiently smooth [16].

2.1.3 Parameter dependent Multi-Objective Optimization Problem

As we mention in the introduction, other kind of problems related to MOPs are arising, one example of them are the parameter dependent multi-objective optimization problems (PMOPs) which are defined as follows:

$$\min_{\mathbf{x}\in S} F_\lambda(\mathbf{x}), \tag{2.6}$$

where F_λ is defined as a vector of objective functions dependent on an external parameter λ

$$\begin{aligned} &F_\lambda : S \to \mathbb{R}^k, \\ &F_\lambda(\mathbf{x}) = (f_1(\mathbf{x}), \cdots, f_k(\mathbf{x})), \end{aligned} \tag{2.7}$$

and where $S \subset \mathbb{R}^n$. $\lambda \in \mathbb{R}^l$ specifies an external parameter, or parameters, to the objective functions.

2.1.4 Decision Variables

One of the most important parts of an optimization problem are the decision variables. They are numerical quantities that control the obtained values by evaluating them in a certain model as it is depicted in Figure 2.2. As a simple example, we can remember

our initial problem 'to construct a motorcycle', where we looked to make the decision at two characteristics of the motorcycle 'quality' and 'cost'. The interior features that define these two objectives can be seen as our decision variables, for instance, the capacity of the engine, the horsepower, the torque, etc.

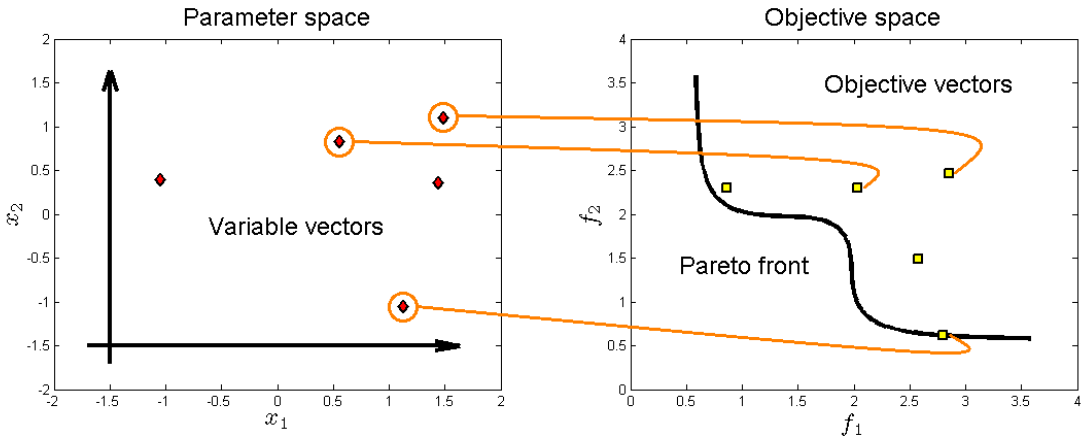


Figure 2.2: Mapping variable vectors by evaluating them to the objective space.

To give a more theoretical definition, we denote a quantity of a decision variable as x_i, where $i = \{1, 2, \ldots, n\}$. The vector $\mathbf{x}$ of n decision variables is represented by:

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}. \tag{2.8}$$

2.1.5 Jacobian Matrix

Since, we are going to deal with MOPs, it is convenient to represent the derivative of the model. In this case $\nabla F(\mathbf{x})$, by using a notation which is called the Jacobian matrix of F in $\mathbf{x}$, denoted by $J(\mathbf{x})$. The members of this matrix are given by all the partial derivatives of F:

$$J(\mathbf{x}) = \frac{\partial F}{\partial x}(\mathbf{x}) = \begin{pmatrix} \frac{\partial f_1}{\partial x_1}(\mathbf{x}) & \ldots & \frac{\partial f_1}{\partial x_n}(\mathbf{x}) \\ \vdots & & \vdots \\ \frac{\partial f_k}{\partial x_1}(\mathbf{x}) & \ldots & \frac{\partial f_k}{\partial x_n}(\mathbf{x}) \end{pmatrix} = \begin{pmatrix} \nabla f_1(\mathbf{x})^T \\ \vdots \\ \nabla f_k(\mathbf{x})^T \end{pmatrix}, \tag{2.9}$$

where $\nabla f_i(\mathbf{x})$ denotes the gradient of objective i, i.e.,

$$\nabla f_i = \begin{pmatrix} \frac{\partial f_i}{\partial x_1}(\mathbf{x}) \\ \vdots \\ \frac{\partial f_i}{\partial x_n}(\mathbf{x}) \end{pmatrix}. \tag{2.10}$$

2.1.6 Pareto Optimality

Optimality is a crucial concept in optimization, therefore, it is important to define it for MOPs. In 1896, Vilfredo Pareto proposed an important definition called later Pareto optimality [17] which is a generalization of the work of Edgeworth in 1881 [18]. In Figure 2.3 it is graphically defined and in the following we are going to define it theoretically.

Definition 2. *(a) Let* $\mathbf{v}$, $\mathbf{w} \in \mathbb{R}^k$. *Then the vector* $\mathbf{v}$ *is less than* $\mathbf{w}$ ($\mathbf{v} <_p \mathbf{w}$), *if* $v_i < w_i$ *for all* $i \in \{1, 2, \ldots, k\}$. *The relation* $\leq_p$ *is defined analogously.*

(b) A vector $\mathbf{y} \in \mathbb{R}^n$ *is dominated by a vector* $\mathbf{x} \in \mathbb{R}^n$ ($\mathbf{x} \prec \mathbf{y}$) *with respect to the definition of a multi-objective optimization problem if*

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

else $\mathbf{x}$ *is called non-dominated by* $\mathbf{y}$.

(c) A point $\mathbf{x} \in S$ *is called (Pareto) optimal or a Pareto point if there exists no* $\mathbf{y} \in S$ *which dominates* $\mathbf{x}$.

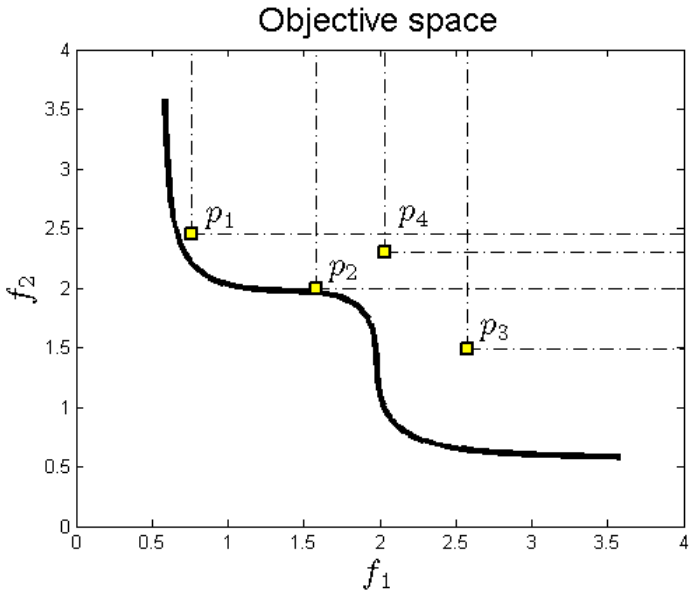


Figure 2.3: Dominated and non-dominated solutions over the objective space. The solution denoted by p_4 is the only one that is dominated by a other solution in this case p_2. Solutions p_1, p_2, and p_3 cannot be compared which means that they are mutually non-dominating.

If all the objectives $f_1(\mathbf{x}), f_2(\mathbf{x}), \ldots, f_k(\mathbf{x})$ are differentiable, the following theorem of Kuhn and Tucker [19] states the necessary condition for Pareto optimality for unconstrained MOPs.

Theorem 1. *Let $\mathbf{x}^*$ be a Pareto point of a MOP; then there exists a vector $\alpha \in \mathbb{R}^k$ with $\alpha_i \geq 0$, $i = 1, \ldots, k$, and $\sum_{i=1}^{k} \alpha_i = 1$ such that*

$$\sum_{i=1}^{k} \alpha_i \nabla f_i(\mathbf{x}^*) = 0. \tag{2.11}$$

The theorem claims that the vector of zeros can be written as a convex combination of the gradients of the objectives at every Pareto point. We notice that Equation (2.11) is not a sufficient condition for Pareto optimality, but we can say that points which satisfy Equation (2.11) are certainly 'Pareto candidates'.

Definition 3. *A point $\mathbf{x} \in \mathbb{R}^n$ is called a Karush-Kuhn-Tucker point* [1] *(KKT point) if*

[1] Named after the works of Karush [20] and Kuhn and Tucker [19].

there exist scalars $\alpha_1, \alpha_2, \ldots, \alpha_k \geq 0$ *such that* $\sum_{i=1}^{k} \alpha_i = 1$ *and that Equation (2.11) is satisfied.*

The set of all global Pareto points is called the Pareto set, denoted by P_S; an example is shown in Figure 2.4 and the image of the Pareto set $F(P_S)$ is called the Pareto front, in Figure 2.5 an example is presented.

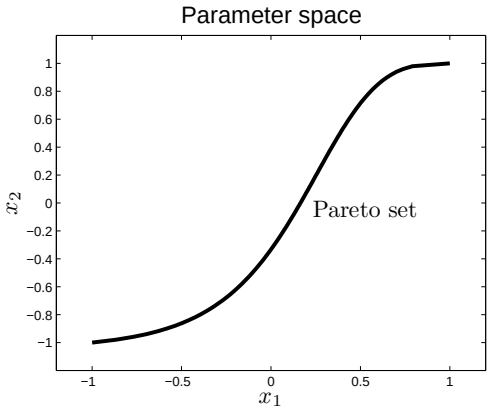


Figure 2.4: Example of a Pareto set.

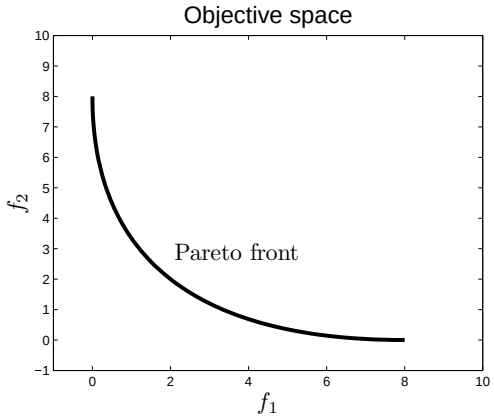


Figure 2.5: Example of a Pareto front.

A multi-objective optimization problem contains a possibly uncountable set of solutions as we can observe in Figure 2.4. Such solutions when are evaluated, produce vectors whose components represent trade-offs in objective space (see Figure 2.5). Thus, to choose an acceptable solution or solutions from the final solution set is the task for a decision maker [21].

Theorem 1 can be used to give a qualitative description of the Pareto set. For this, define the following map:

$$\tilde{F} : \mathbf{R}^{n+k} \to \mathbf{R}^{n+1}$$
$$\tilde{F}(\mathbf{x}, \alpha) = \begin{pmatrix} \sum_{i=1}^{k} \alpha_i \nabla f_i(\mathbf{x}) \\ \sum_{i=1}^{k} \alpha_i - 1 \end{pmatrix}. \tag{2.12}$$

If $\mathbf{x}$ is a Pareto point there exists by Theorem 1 a vector $\alpha^* \in \mathbb{R}^k$ such that $\tilde{F}(\mathbf{x}^*, \alpha^*) = 0$. Hence, the Pareto set and the according set of weight vectors are contained in the preimage $\tilde{F}^{-1}(0)$, and we expect by the Implicit Function Theorem [22] that this set forms a set of dimension $k-1$. This is indeed the case under certain assumptions on the MOP [3].

In the case of PMOPs, the family of Pareto sets (one for each value of λ) is thus defined as

$$P_{S,\Lambda} := \{(\mathbf{x}, \lambda) \in \mathbb{R}^{n+l},\ \text{s.t } \mathbf{x} \text{ is a Pareto point of } F_\lambda,\ \lambda \in \Lambda\} \tag{2.13}$$

where its image denoted by $F(P_{S,\Lambda})$ will represent the family of Pareto fronts.

Unlike MOPs, we notice that PMOPs have a complete family of Pareto sets and Pareto fronts, since they include the influence of an external parameter. As an example, we consider the following PMOP taken from [6] to show how the solution sets looks like. Figures 2.6 and 2.7 present a graphical representation of this problem in both parameter space and objective space.

$$F_\lambda : \mathbb{R}^n \to \mathbb{R}^k$$

$$F_\lambda(x) := (1-\lambda)F_1(x) + \lambda F_2(x), \tag{2.14}$$

where $\lambda \in \mathbb{R}$ and

$$F_1, F_2 : \mathbb{R}^2 \to \mathbb{R}^2$$

$$F_1(x_1, x_2) = \begin{pmatrix} (x_1-1)^4 + (x_2-1)^2 \\ (x_1+1)^2 + (x_2+1)^2 \end{pmatrix},$$

$$F_2(x_1, x_2) = \begin{pmatrix} (x_1-1)^2 + (x_2-1)^2 \\ (x_1+1)^2 + (x_2+1)^2 \end{pmatrix}.$$

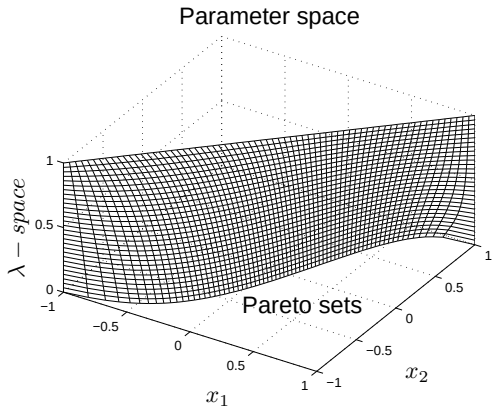


Figure 2.6: Family of Pareto sets of a PMOP.

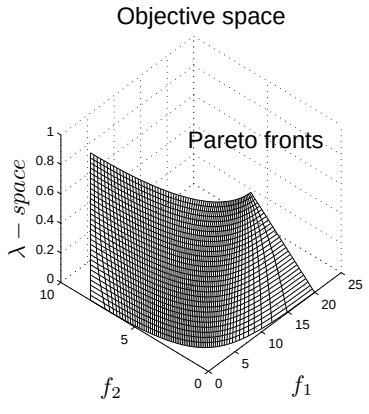


Figure 2.7: Family of Pareto fronts of a PMOP.

2.2 Mathematical Programming Techniques for MOPs

To give a brief insight into the developments related to MOPs and PMOPs, we have to start by describing the main available methods to find the entire set of optimal solutions for a multi-objective optimization problem. These methods are based on a mathematical formulation.

2.2.1 Scalarization Methods

One way to tackle MOPs is to use scalarization methods, i.e., to transform the original problem (defined in Equation (2.4)) into a scalar optimization problem of the form:

$$\min_{\mathbf{x}\in S} f_\alpha(\mathbf{x}), \tag{2.15}$$

where $f_\alpha : S \to \mathbb{R}$ and $\alpha \in \mathbb{R}^k$ is an external parameter. Note that for a given value of α the solution of Equation (2.15) is typically a single point rather than a $(k-1)$-manifold. Finite size Pareto set/front approximations can hence only be obtained by choosing a clever sequence of optimization problems using Equation (2.15) which calls for a suitable set $A := \alpha^{(1)}, \ldots, \alpha^{(m)} \subset \mathbb{R}^k$ of external parameters.

The general advantage of the use of scalarization methods is that they can be tackled by any solver for scalar optimization problems. On the other hand, it is not always ensured that the resulting set of minimizers forms a suitable approximation of the Pareto set/front (e.g., in terms of the spread along the set of interest).

Weighted Sum Method Probably, the first scalarization method is the 'Weighted Sum Method' [23, 4]. The underlying idea is to assign to each objective a certain weight $\omega_i \geq 0$, and to minimize the resulting weighted sum. Given a problem of the form of Equation (2.4), the weighted sum problem can be stated as follows:

$$\min_{\mathbf{x}\in S} \sum_{i=1}^{k} \omega_i f_i(\mathbf{x}), \tag{2.16}$$

where $\omega_i \geq 0$ for all $i = 1, \ldots, k$ and $\sum_{i=1}^{k} \omega_i = 1$ [16].

The main advantage of the Weighted Sum Method is that one can expect to find Pareto optimal solutions by solving Equation (2.16). It can be shown that for

problems where the Pareto front is convex all points on this set can be reached by solving Equation (2.16) for a particular value of ω. Nevertheless, the main drawback of this method is that optimal solutions $\mathbf{p}$ where $F(\mathbf{p})$ lies on a concave portion of the Pareto front, can hardly be found.

ϵ-Constraint Method The 'ϵ-Constraint Method' [24] consists in selecting one objective $f_i,\ i \in \{1,\ldots,k\}$, to be optimized, while the other objectives are treated as constraints by setting an upper bound to each of them. This leads to the following optimization problem:

$$\min f_i(\mathbf{x}), \tag{2.17}$$

$$\text{s.t } f_j(\mathbf{x}) \leq \epsilon_j,\ \mathbf{x} \in S \text{ for all } j = 1,\ldots,k,\ j \neq e.$$

Theorem 2. *A vector $\mathbf{x} \in S$ is Pareto optimal if and only if it is a solution of the ϵ-constraint problem (defined in Equation (2.17)) for every $i = 1,\ldots,k$, where $\epsilon_j = f_j(\mathbf{x}^*)$ for $j = 1,\ldots,k,\ j \neq i$.*

Unlike the Weighted Sum Method , by using the ϵ-Constraint Method, it is possible to find optimal solutions even in non-convex regions. However, the proper choice of the values of ϵ may get difficult. Further, there is no a specific criterion to select the objective to minimize.

Weighted Tchebycheff Method The aim of the 'Weighted Tchebycheff Method' [25] is to find a point whose image is as close as possible to a given reference point $\mathbf{z} \in \mathbb{R}^k$. For the distance assignment the weighted Tchebycheff metric is mostly used: Let $\omega \in \mathbb{R}^k$ with $\omega_i \geq 0$, $i = 1,\ldots,k$, and $\sum_{i=1}^{k} \omega_i = 1$, and let $\mathbf{z} = [z_1,\ldots,z_k]^T$, then the Weighted Tchebycheff Method reads as follows:

$$\min_{\mathbf{x}\in S} \max_{i=1,\ldots,k} \omega_i |f_i(\mathbf{x}) - z_i|. \tag{2.18}$$

Note that the solution of Equation (2.18) depends on $\mathbf{z}$ as well as on ω. The main advantage of the Weighted Tchebycheff Method is that by a proper choice of these vectors every point on the Pareto front can be reached.

Theorem 3. *The solution of Equation (2.18) is weakly Pareto optimal if $\omega \in \mathbb{R}^k_+$.*

Theorem 4. *Let $\mathbf{x}^* \in S$ be Pareto optimal. Then there exists $\omega \in \mathbb{R}^k_+$ such that $\mathbf{x}^*$ is a solution of Equation (2.18), where $\mathbf{z}$ is chosen as the utopia vector of the MOP.*

The utopia vector $F* = [f_1^*, \ldots, f_k^*]^T$ of a MOP consists of the minimal objective values f_i^* of each function f_i. On the other hand, the proper choices of $\mathbf{z}$ and ω might also present delicate problems for a particular problem.

Normal Boundary Intersection Method In 1998 an alternative scalararization method was proposed by Das and Dennis [26], the Normal Boundary Intersection method (NBI). The NBI method 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 minima of each objective.

2. Points $\mathbf{y}_i$ from the CHIM are selected and the point $\mathbf{z}_i^* \in S$ is computed such that the image $F(\mathbf{x}_i^*)$ has the maximal distance from $\mathbf{y}_i$, in the direction that is normal to the CHIM and points toward the origin.

To be more precise, let $\mathbf{x}_i^*$ be a global minimizer of the i-th objective, let $\mathbf{F}_i^* := F(\mathbf{x}_i^*)$, and denote

$$\Phi := [\mathbf{F}_1^*, \ldots, \mathbf{F}_k^*] \in \mathbb{R}^{k \times k}. \tag{2.19}$$

Then the CHIM is defined as

$$\text{CHIM} = \left\{ \Phi\omega : \omega \in \mathbb{R}^k : \sum_{i=1}^{k} \omega_i = 1, \omega_i \geq 1, i = 1, \ldots, k \right\}. \tag{2.20}$$

The optimization problem in the second step is called the NBI-subproblem. Given an initial value $\Phi\omega = \sum_{i=1}^k \omega_i \mathbf{F}_i^*$ and the direction $d \in \mathbb{R}^k$ which is orthogonal to the CHIM and points toward the origin, the NBI-subproblem can be stated in mathematical terms as follows:

$$\max_{\mathbf{x},t} t \tag{2.21}$$

$$\text{s.t } F(\mathbf{x}_0) + td = F(\mathbf{x}),$$

$$\mathbf{x} \in S.$$

The usage of Equation (2.21) can be helpful since there are scenarios where the aim is to steer the search in a certain direction given in objective space [26, 27]. However, solutions of Equation (2.21) do not have to be Pareto optimal. A graphical representation of the NBI method is depicted in Figure 2.8.

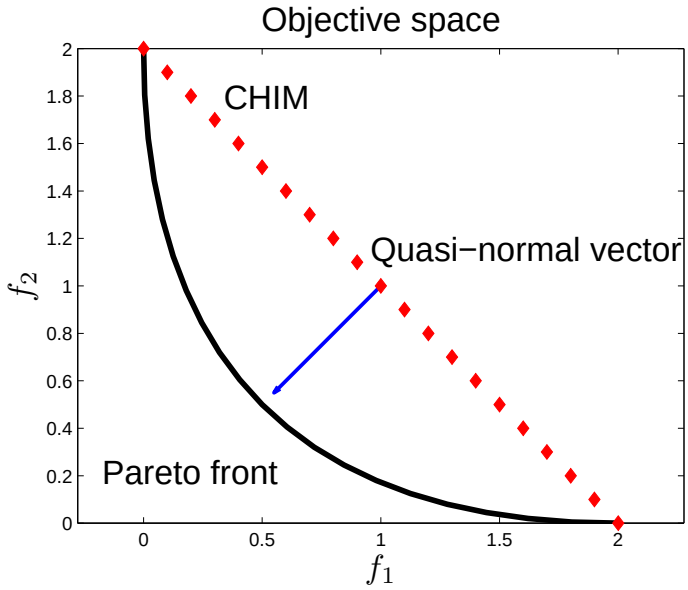


Figure 2.8: Example of the convex hull of individual minima and the quasi-normal vector for a particular MOP.

2.2.2 Set Oriented Methods

Set oriented methods such as subdivision techniques [6, 28, 29, 30] generate an entire set in each iteration step in a suitable sense. To perform this, the process starts with a division over a box previously defined, then it generates outer approximations of the set of interest until a desired granularity of the box is reached. To define the algorithm, we have to consider a finite collection of discrete dynamical systems of the type:

$$\mathbf{x}_{j+1} = f_m(\mathbf{x}_j),\ j = 0, 1, 2, \ldots, \tag{2.22}$$

where we assume for simplicity that each $f_m : \mathbb{R}^n \to \mathbb{R}^n$ $m = 1, \ldots, s$ is continuous. Hence, the aim is to compute invariant sets for the entire collection of dynamical systems. To be more precise, the goal is to approximate a subset $A \subset \mathbb{R}^n$ such that:

$$f_m(A) = A \text{ for } m = 1, \ldots, s. \tag{2.23}$$

Another method of this sort is the cell mapping approach [31]. This method starts with the idea to see a numerical approach to solve a MOP as a dynamical system. This method treats the search space as a discrete search space instead of continuous. Such method proposes to increase a discretization of the search space by dividing it into bigger hypercubes. The evolution of the dynamical system is then reduced to a new function, which is not in $\mathbb{R}^n$, but on the cell space. In the case of solving only functions that are strictly deterministically defined, we have the so-called simple cell mapping. This technique allows us to obtain the attractors and basins of attractions of a dynamical system. In [32, 33] new applications of simple cell mapping are presented in the context of multi-objective optimization.

2.2.3 Descent Directions

To use descent directions is another way to perform a local search toward the set of optimal solutions. Given a point $\mathbf{x} \in \mathbb{R}^n$, a vector $\nu \in \mathbb{R}^n$ is called a descent direction if a search in that direction leads ideally to improve all the objective values. In other words, v is a descent direction of a MOP at a point $\mathbf{x} \in \mathbb{R}^n$ if there exists a $\tilde{t} \in \mathbb{R}_+$ such that

$$F(\mathbf{x} + t\nu) <_p F(\mathbf{x}), \ \forall t \in (0, \tilde{t}). \tag{2.24}$$

If all objectives of a MOP are differentiable, then Equation (2.24) is equivalent to

$$\nabla f_i(\mathbf{x})^T \nu < 0, \ i = 1, \ldots, k. \tag{2.25}$$

Hence, if a descent direction ν is given for an initial point $\mathbf{x}$, a solution $\mathbf{x}_{new}$ that dominates $\mathbf{x}$ can be found by performing a line search, i.e.,

$$\mathbf{x}_{new} = \mathbf{x} + t\nu, \tag{2.26}$$

where $t \in \mathbb{R}_+$ is a (sufficiently small) step size. In [34, 28, 35, 36] some descent directions have been presented. For the purpose of this work, we introduce two descent directions in the following.

Averaged Descent Direction In [28] a descent direction is presented, in the case of having bi-objective optimization problems (i.e., $k = 2$). This direction was already

integrated into a MOEA in the work of Lara et al. in [37]. The Average Descent Direction takes advantage of the descent cone properties of a MOP and it reads as follows:

Theorem 5. *Let* $\mathbf{x} \in \mathbf{R}^n$, $f_1, f_2 : \mathbb{R}^n \to \mathbb{R}$ *define a bi-objective MOP, and* $\nabla f_i \neq 0$ *for* $i = 1, 2$. *Then, the direction*

$$\nu = -\frac{1}{2}\left(\frac{\nabla f_1(\mathbf{x})}{||\nabla f_1(\mathbf{x})||} + \frac{\nabla f_2(\mathbf{x})}{||\nabla f_2(\mathbf{x})||}\right), \tag{2.27}$$

where $||\cdot|| = ||\cdot||_2$, *is a descent direction at* $\mathbf{x}$ *for the MOP.*

One of the main advantages to compute this descent direction is that it only needs the gradient information unlike the other approaches where in most of the cases one has to solve a linear quadratic optimization problem. Nonetheless, the method still has a clear disadvantage namely that it cannot be generalized for more than two objective functions.

Descent direction of Schäffler, Schultz and Weinzierl In [34] Schäffler, Schultz and Weinzierl presented an approach for the computation of a descent direction for unconstrained MOPs. This descent direction is defined as follows:

Theorem 6. *Let a given MOP as defined in Equation (2.4) and the map* $q : \mathbb{R}^n \to \mathbb{R}$ *be defined by*

$$q(\mathbf{x}) = \sum_{i=1}^{k} \tilde{\alpha}_i \nabla f_i(\mathbf{x}), \tag{2.28}$$

where $\tilde{\alpha}$ *is a solution of*

$$\min_{\alpha \in \mathbb{R}} \left\{ \left\| \sum_{i=1}^{k} \alpha_i \nabla f_i(\mathbf{x}) \right\|_2^2 : \ \alpha_i \geq 0, \ i = 1, \ldots, k, \ \sum_{i=1}^{k} \alpha_i = 1 \right\}. \tag{2.29}$$

Then the following statements hold:

- *Either* $q(\mathbf{x}) = 0$ *or* $-q(\mathbf{x})$ *is a descent direction.*
- *For each* $\tilde{\mathbf{x}} \in \mathbf{R}^n$, *there exists a neighborhood* $N(\tilde{\mathbf{x}})$ *and a constant* $L_{\tilde{\mathbf{x}}} \in \mathbb{R}_0^+$ *such that*

$$||q(\mathbf{x}) - q(\mathbf{y})||_2 \leq L_{\tilde{\mathbf{x}}} ||\mathbf{x} - \mathbf{y}||_2, \ \forall \mathbf{x}, \mathbf{y} \in N(\tilde{\mathbf{x}}). \tag{2.30}$$

Note that if $q(x) = 0$, then $\mathbf{x}$ is a KKT point. While q is computed a test for a first order necessary condition for optimality is being performed also.

2.2.4 Multi-Objective Continuation Methods

Since the set of interest, the Pareto set/front, forms at least locally a manifold, it can make sense to perform a search along this set once locally optimal solutions are detected. Such methods are particularly advantageous if the Pareto front is connected. In the following we describe the core of the predictor-corrector (PC) methods as described in [38] for general implicit defined manifolds and in [3] for multi-objective optimization.

Crucial for the PCs presented in [38, 3] is the map $\tilde{F}$ as described in Equation (2.12) that transforms a MOP problem into a root finding problem: The set of interest is now

$$M := \{(\mathbf{x}, \alpha) \in \mathbb{R}^{n+k} | \tilde{F}(\mathbf{x}, \alpha) = 0\}. \tag{2.31}$$

To describe a PC, it is important to state some technical details. The tangent space of M at $(\mathbf{x}, \alpha) \in M$ is given by

$$T_{(\mathbf{x},\alpha)}\partial M = \ker J(\mathbf{x}, \alpha) = \{\mathbf{u} \in \mathbb{R}^{n+k} | J(\mathbf{x}, \alpha)\mathbf{u} = 0\}, \tag{2.32}$$

where $J(\mathbf{x}, \alpha)$ denotes the Jacobian matrix of $\tilde{F}$ at $(\mathbf{x}, \alpha)$ and $\ker A$ the kernel of a matrix A. Well-spread tangent vectors can e.g., be obtained by computing a QR-decomposition of $J(\mathbf{x}, \alpha)^T$:

Let $Q = (Q_N, Q_K) \in \mathbb{R}^{(n+k)\times(n+k)}$ be an orthogonal matrix with $Q_N \in \mathbb{R}^{(n+k)\times(n+1)}$ and $Q_K \in \mathbb{R}^{(n+k)\times(k-1)}$ and $R = (R_1, 0, \ldots, 0)^T \in \mathbb{R}^{(n+k)\times(n+1)}$, where $R_1 \in \mathbb{R}^{(n+1)\times(n+1)}$ is a right upper triangular matrix such that

$$\tilde{F}'(\mathbf{x}, \alpha)^T = \begin{pmatrix} \sum_{i=1}^{k} \alpha_i \nabla^2 f_i(\mathbf{x}) & 0 \\ \nabla f_1(\mathbf{x})^T & 1 \\ \vdots & \vdots \\ \nabla f_k(\mathbf{x})^T & 1 \end{pmatrix} = QR. \tag{2.33}$$

Hereby, $\nabla^2 f(\mathbf{x}) \in \mathbb{R}^{n\times n}$ denotes the Hessian of f at $\mathbf{x}$. Then the columns of Q_K build an orthonormal basis of $T_{\mathbf{x},\alpha}\partial M$.

In the following, we formulate the general idea of PCs: Given a point $(\mathbf{x}, \alpha) \in M$, further points along M are computed in using the next two steps (for a graphical representation see Figure 2.9):

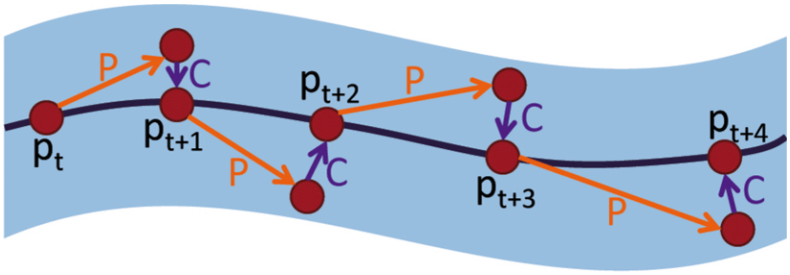


Figure 2.9: A predictor-corrector method. This image was taken from [39].

Predictor To predict a set $\{p^{(1)}, \ldots, p^{(s)}\} \subset \mathbb{R}^{n+k}$ of distinct and well-distributed points that are near both to (x^*, α^*) and M, we can use Equation (2.33), i.e., by choosing

$$p^{(i)} = (\mathbf{x}^*, \alpha^*) + t_i q_i, \tag{2.34}$$

where $t_i \in \mathbb{R} \setminus 0$ is a step size and q_i the i-th column vector of Q_K.

Corrector Then, for $i = 1, \ldots, s$, starting from the predicted point $p^{(i)}$, compute by (typically few) iterative steps an approximated element $(\mathbf{x}^{(i)}, \alpha^{(i)}) \in M$, i.e., such that $\tilde{F}(\mathbf{x}^{(i)}, \alpha^{(i)}) \approx 0$. This can e.g., be done by applying root finding methods such as the Gauss-Newton method to $\tilde{F}$.

A drawback of the above procedure is that one needs to compute or approximate the Hessians of the objectives. To overcome this problem, in [40] a continuation method is presented using the DS without computing Hessian information that makes the process cheaper. In [41] a novel gradient free continuation method based on the DS is presented, this method allows to steer the search along the solution set (Pareto set) of a multi-objective optimization problem without using any gradient or Hessian information.

2.2.5 The Directed Search Method

The Directed Search Method has been proposed for differentiable MOPs and it allows us to steer the search process from a given point into a desired direction in objective space [40] as we show in Figure 2.10. To be more precise, given a point $\mathbf{x}_0 \in \mathbb{R}^n$, and a vector $\mathbf{d} \in \mathbb{R}^k$ representing the desired search direction in image space, a search

direction $\nu \in \mathbb{R}^n$ in parameter space is sought such that

$$\lim_{t \to 0} \frac{f_i(\mathbf{x}_0 + t\nu) - f_i(\mathbf{x}_0)}{t} = d_i, \quad i = 1, \ldots, k. \tag{2.35}$$

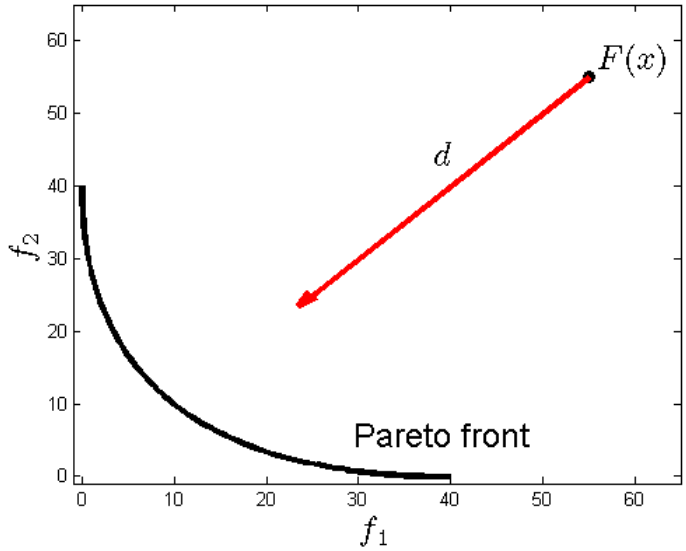


Figure 2.10: The directed search method.

The Equation (2.35) can be stated in matrix vector notation, therefore such a direction vector ν solves the following system of linear equations:

$$J(\mathbf{x}_0)\nu = \mathbf{d}, \tag{2.36}$$

where $J(\mathbf{x})$ denotes the Jacobian of F at $\mathbf{x}$ as we state in Equation (2.9). Since typically $k << n$, we can assume that the linear system in Equation (2.36) is (highly) under-determined. Among the solutions of Equation (2.36), the one with the least 2-norm can be viewed as the greedy direction for the given context what means by using this direction we are going to obtain the maximum gain. The wanted solution is given by

$$\nu_+ := J(\mathbf{x})^+ \mathbf{d}, \tag{2.37}$$

where $J(\mathbf{x})^+$ denotes the pseudo-inverse of $J(\mathbf{x})$ (we refer e.g. to [42] for an efficient computation of ν_+). If one proceeds the search in direction $\mathbf{d}$ in the same manner, this

is identical to the numerical solution of the following initial value problem (starting from solution $\mathbf{x}_0 \in \mathbb{R}^n$):

$$\begin{aligned} \mathbf{x}(0) &= \mathbf{x}_0 \in \mathbb{R}^n \\ \dot{\mathbf{x}}(t) &= \nu_+(\mathbf{x}(t)), \quad t > 0. \end{aligned} \tag{2.38}$$

If $\mathbf{d}$ is a 'descent direction' (i.e., $d_i \leq 0$ for all $i = 1, \ldots, k$ and there exists an index j such that $d_j < 0$), a numerical solution of (2.38) can be viewed as a hill climber for MOPs. In Figure 2.11, we can observe the curve of points computed by DS using a direction d.

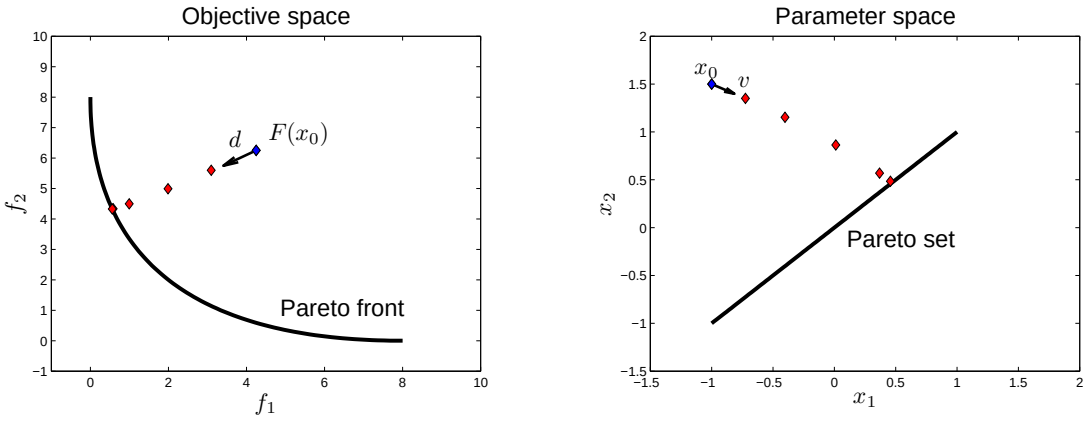


Figure 2.11: A curve of dominating points obtained by the Directed Search Method.

The end point $\mathbf{x}^*$ of the solution curve of (2.38) does not necessarily have to be a Pareto point, but it is a boundary point in objective space, i.e., $F(\mathbf{x}^*) \in \partial F(\mathbb{R}^n)$ (where $\partial F(\mathbb{R}^n)$ denotes the set of bounds of a given MOP) which means that the gradients of the objectives in $\mathbf{x}^*$ are linearly dependent (and hence, that $\text{rank}(J(\mathbf{x}^*)) < k$). This fact can be used to check numerically if a current iteration is near to a boundary point: for the condition number of the Jacobian it holds

$$\kappa_2(J(\mathbf{x})) = \sqrt{\frac{\lambda_{max}(J(\mathbf{x})^T J(\mathbf{x}))}{\lambda_{min}(J(\mathbf{x})^T J(\mathbf{x}))}} \to \infty \quad \text{for} \quad \mathbf{x} \to \mathbf{x}^*, \tag{2.39}$$

where $\lambda_{max}(A)$ and $\lambda_{min}(A)$ denote the largest and the smallest eigenvalue of matrix A, respectively. (Roughly speaking, the condition number indicates how 'near' the rows of $J(\mathbf{x})$, i.e., the gradients of the objectives, are to be linearly independent: The higher the value of $\kappa_2(J(\mathbf{x}))$, the closer $J(\mathbf{x})$ is to a matrix with rank less than k.) Further, one can check the (approximated) end point $\mathbf{x}^*$ numerically for optimality

by checking if $\| \sum_{i=1}^{k} \tilde{\alpha}_i \nabla f_i(\mathbf{x}^*) \|_2 \leq tol$, where $tol > 0$ is a given tolerance and $\tilde{\alpha}$ solves the k-dimensional quadratic optimization problem stated in Equation (2.29).

The hill climber described above shares many characteristics with the one described in [43], where also possible choices for $\mathbf{d}$ are discussed.

Next to the DS descent method there exists the possibility to use the method to search along the Pareto set as a predictor-corrector method which performs the movement along Pareto set of a given MOP, different to the other approaches, the 2nd derivative information is not necessary. For the predictor the method uses the orthogonal vector α to the Pareto front which is the convex weight related with a given local Pareto point $\mathbf{x}$ such that $\sum_{i=1}^{k} \alpha_i \nabla f_i(\mathbf{x}) = 0$ and further we assume that $\text{rank}(J(\mathbf{x})) = k - 1$. In this case α is orthogonal to the Pareto front, i.e, $\alpha \perp T_y \partial F(\mathbb{R}^n)$ where T_y denotes the tangent space of $\partial F(\mathbb{R}^n)$. Thus, the next idea is a search orthogonal to α as is depicted in Figure 2.12, for that reason the QR-factorization of α is necessary to obtain the orthonormal basis of the tangent space. In this way, we obtain an orthogonal matrix $Q = (q_1, \ldots, q_k) \in \mathbb{R}^{k \times k}$ and q_i, $i = 1, \ldots, k$, its column vectors, and $R = (r_{11}, 0, \ldots, 0)^T \in \mathbb{R}^{k \times 1}$ with $r_{11} \in \mathbb{R} \backslash \{0\}$. Since $\alpha = r_{11} q_1 (\alpha \in \text{span}\{q_1\})$ and Q is orthogonal, the vectors $(q_2, \ldots, q_k)$ form the orthonormal basis of the orthogonal hyperplane to α.

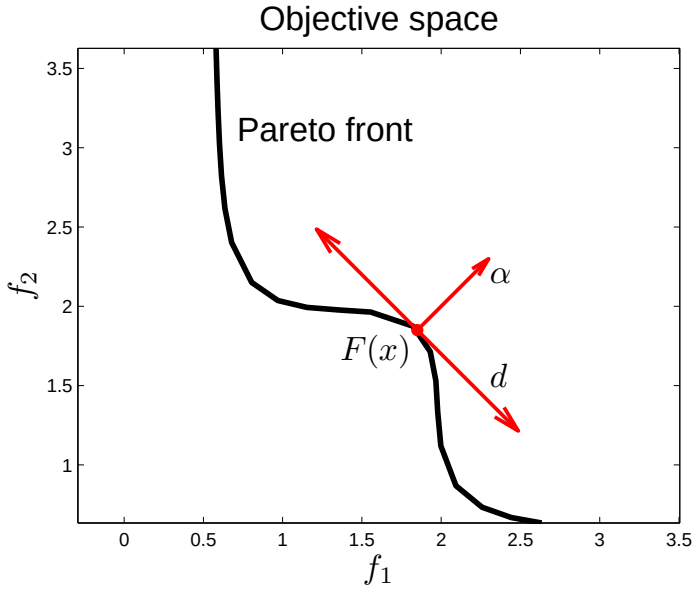


Figure 2.12: A movement over the Pareto front using a direction $\mathbf{d}$ by getting the vector α.

Using

$$J(\mathbf{x})\nu_i = q_i,\ i = 2, \ldots, k \tag{2.40}$$

we can obtain directions ν_i to have a predictor point without any second gradient information. To perform the corrector step, we can use the predictor point p and solve Equation (2.38).

2.3 Performance Indicators

In order to assess the produced outcome set by a multi-objective optimization algorithm, one needs a performance indicator to carry out this task. Most indicators mainly look for two important concepts: Spread and convergence. For this section, we are going to present several indicators.

2.3.1 Hypervolume

The so-called *dominated hypervolume* (or S metric) of a population is a commonly accepted measure [44] for assessing the quality of an approximation. This indicator refers to the size of the space covered or size of dominated space. The hypervolume is described as the Lebesgue measure Λ of the union of hypercubes defined by a non-dominated point $v^{(i)}$ and a reference point R expressed as $\bigcup[v^{(i)}, R]$. In Figure 2.13 a graphical representation of hypervolume is shown.

Definition 4. *Let* $v^{(1)}, v^{(2)}, \ldots v^{(\mu)} \in \mathbb{R}^k$ *be a non-dominated set and* $R \in \mathbb{R}^k$ *such that* $v^{(i)} \prec R$ *for all* $i = 1, \ldots, \mu$. *The value*

$$Hyp(v^{(1)}, \ldots, v^{(\mu)}; R) = \Lambda_d \left(\bigcup_{i=1}^{\mu} [v^{(i)}, R] \right) \tag{2.41}$$

is termed the dominated hypervolume *with respect to* reference point R, *where* $\Lambda_d(\cdot)$ *denotes the Lebesgue measure in* $\mathbb{R}^k$.

This measure has a number of appealing properties but determining its value is getting the more tedious the larger the number of objectives is considered [45]. In case of two objectives ($k = 2$) and lexicographically ordered non-dominated set

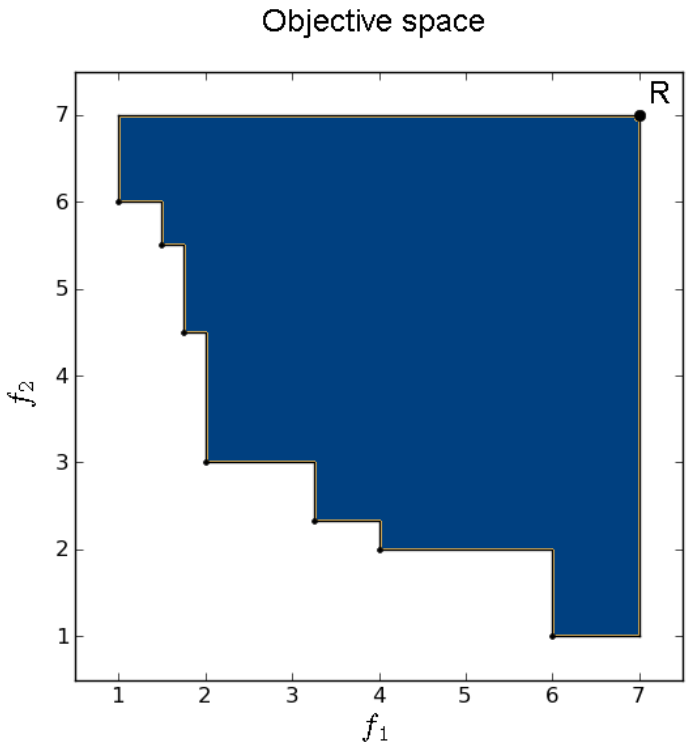


Figure 2.13: Dominated hypervolume for a two objective setting.

$v^{(1)}, v^{(2)}, \ldots v^{(\mu)}$ the calculation of (2.41) reduces to

$$\text{Hyp}(v^{(1)}, \ldots, v^{(\mu)}; R) = \left[r_1 - v_1^{(1)}\right] \cdot \left[r_2 - v_2^{(1)}\right] + \sum_{i=2}^{\mu} \left[r_1 - v_1^{(i)}\right] \cdot \left[v_2^{(i-1)} - v_2^{(i)}\right]. \tag{2.42}$$

By analyzing this indicator some advantages are explained in the following:

- If R is given, $\text{Hyp}(A)$ can be computed without further knowledge,
- Hyp is Pareto compliant. That is if $(b < a)$ and $A_1 = \{a, a_2, \ldots, a_j\}$ and $A_2 = \{b, a_2, \ldots, a_j\}$, it follows that $\text{Hyp}(A_2) \geq \text{Hyp}(A_1)$ [21].

On the other hand, also this indicator has some disadvantages such as:

- Hyp is dependent on the choice of R,

- the computation of Hyp(A) for $k > 2$ is time consuming, since the complexity was estimated as $O(n^{k+1})$ with n being the number of variables in parameter space and k the number of objectives [46].
- the final distribution according to Hyp is not evenly distributed along the Pareto front.

2.3.2 Hausdorff Distance

Another indicator is the Hausdorff distance [47], which is described next, step by step:

(a) The distance between two points $a, b \in \mathbb{R}^n$ is defined as

$$\text{dist}(a, b) := ||a - b||, \tag{2.43}$$

where $|| \cdot ||$ is a norm on $\mathbb{R}^n$.

(b) The distance between a point $b \in \mathbb{R}^n$ and a set $A \subset \mathbb{R}^n$ is defined as

$$\text{dist}(b, A) := \inf_{a \in A} ||b - a||. \tag{2.44}$$

(c) The semi-distance between two sets A, $B \subset \mathbb{R}^n$ is defined as

$$\text{dist}(B, A) = \sup_{b \in B} \text{dist}(b, A) = \inf_{a \in A} \sup_{b \in B} ||a - b||. \tag{2.45}$$

Note that dist(A, B) does not have to be equal to dist(B, A), i.e., dist is not symmetric. As example consider $A \subset B$, then dist$(A, B) = 0$ but not necessarily dist$(B, A) = 0$.

(d) The Hausdorff distance between A, $B \subset \mathbb{R}^n$

$$d_H(A, B) = \max(\text{dist}(A, B), \text{dist}(B, A))$$

$$= \max(\inf_{a \in A} \sup_{b \in B} ||a - b||, \inf_{b \in B} \sup_{a \in A} ||b - a||). \tag{2.46}$$

One of its principal advantages is that d_H defines a metric on the set of compact subsets. However, it is important to mention that d_H is sensitive to single outliers. Such outliers can be generated when using stochastic method such as MOEAs for the approximation of the Pareto set/front.

2.3.3 Generational Distance and Inverted Generational Distance

The Generational Distance (GD) introduced by Van Veldhuizen and Lamont [48] is a value representing how 'far' the known Pareto front A is from the true Pareto front $F(P_s)$. This indicator is defined as follows:

Let

$$A = \{a_1, \ldots, a_j\} \tag{2.47}$$

be an archive with j vectors $a_i \in \mathbb{R}^n$, then

$$\mathrm{GD}(A, F(P_s)) := \frac{1}{j}(\sum_{i=1}^{j} d_i^p)^{1/p}, \tag{2.48}$$

where $p = 2$ and d_i is the Euclidean distance (in objective space) between the image of a_i and the nearest member of the true Pareto front.

In Cruz and Coello's work [49], the Inverted Generational Distance (IGD) is proposed. IGD uses as a reference a discretization of the true Pareto front $F(P_s)$, and compares each of its elements with respect to the front A produced by an algorithm. This intends to reduce some of the problems that occur with the generational distance metric when an algorithm generates very few non-dominated solutions. In the following IGD is defined:

Let $\{y_1, \ldots, y_z\}$ be a discretization of Pareto front, $z \in \mathbb{N}$, and A is defined as in (2.47), then

$$\mathrm{IGD}(A, F(P_s)) := \frac{1}{z}(\sum_{i=1}^{z} ds_i^p)^{1/p}, \tag{2.49}$$

where ds denotes the Euclidean distance from y_i to A.

The disadvantage for both methods is that if we increase the size of one of this sets the values for both GD and IGD get lower (i.e., the approximation appears to be 'better').

2.3.4 Averaged Hausdorff Distance

In [50] a new performance indicator Δ_p is proposed, which can be seen as an 'averaged Hausdorff distance' (AHD) between the image of A and the true Pareto front and

the approximation. This new metric is composed by the indicators GD and IGD but with little changes that we describe below:

Given a candidate set $A = \{a_1, \ldots, a_j\}$ and a discretization of the Pareto front $F(P_s) = \{y_1, \ldots, y_z\}$, then

$$\mathrm{GD}_p(A, F(P_s)) := \left(\frac{1}{j} \sum_{i=1}^{j} d_i^p \right)^{1/p} \tag{2.50}$$

is the averaged semi-distance from the image of A to the discretization of the Pareto front and

$$\mathrm{IGD}_p(A, F(P_s)) := \left(\frac{1}{z} \sum_{i=1}^{z} ds_i^p \right)^{1/p} \tag{2.51}$$

is the averaged semi-distance of the y_i's to the image of A. Finally, we define Δ_p as:

$$\Delta_p := \max(GD_p(A), IGD_p(A)). \tag{2.52}$$

Hence, it holds $\Delta_\infty = d_H$ and Δ_p with $p < \infty$ can be considered as the averaged Hausdorff distance.

2.4 Evolutionary algorithms

In order to obtain an approximation of the Pareto set, one has to perform a series of separate runs in the case of the traditional mathematical programming techniques. In contrast, subdivision techniques and evolutionary algorithms (EA) can help us to find several members of the Pareto-optimal set in a single run of the algorithm. The EAs that tackle MOPs, are called multi-objective evolutionary algorithms (MOEAs) [2, 51, 21]. A multi-objective evolutionary algorithm normally shares the structure and concepts analogous to its genetic counterparts. Normally, a structure or individual is an encoded solution to some problem and the set of individuals is called population. Within a MOEA there exist evolutionary operators that act over the population generating solutions with higher and higher fitness. This fitness is going to determine which element will survive for the next generation. The main operators are mutation, recombination and selection [21]. A basic structure of an MOEA is shown in Algorithm 1.

Algorithm 1 Generic Evolutionary Algorithm

Require: A model of a problem to solve.

Ensure: A candidate set for the given problem.

1: Generate (randomly) an initial population.
2: **while** the stopping-criteria is not reached **do**
3: Compute the fitness for each individual.
4: Apply the evolutionary operators (recombination and mutation) to generate the offspring.
5: Select the elements for the next iteration.
6: **end while**

MOEAs arise aiming to generate solutions over a MOP such that the solutions belong to the Pareto set. Some examples for this sort of approaches are NSGA-II [7], MOEA/D [8], SMS-EMOA [52], Δ_p-EMOA[10].

2.4.1 Indicator-Based Evolutionary Algorithms

There is a trend to use a performance indicator to select such elements whose contribution helps to improve the value of this measure. These MOEAs are called indicator-based evolutionary algorithms (IBEAs). Nowadays, they have caught the interest of many researchers to treat multi-objective optimization problems. This is due to the fact that they yield a good approximation of the solution set and usually yield a better performance than dominance-based algorithms over a given indicator. Nevertheless, as we mentioned in the first chapter, since they are a sort of MOEAs, they also still suffer the same problem namely the slow convergence. Examples of this kind of approaches are the SMS-EMOA [53] using the hypervolume, the one in [10] which uses the AHD, among others. For the purpose of this thesis it is important to explain the SMS-EMOA, since we use this algorithm to integrate one of our proposed local search procedures.

SMS-EMOA The S metric selection evolutionary multi-objective algorithm (SMS-EMOA) was proposed in [53] by Emmerich, Beume and Naujoks. In this algorithm the hypervolume governs the selection operator, which allows to get at the final a good

approximation according to the hypervolume. The main idea is to integrate new points in the population by replacing such elements whose contribution to hypervolume is the worst, Figure 2.14 shows graphically this contribution. In the following, we are going to explain this algorithm step by step.

Algorithm 2 SMS-EMOA algorithm

Require: A model of a problem to solve.

Ensure: A candidate for the given problem.

1: Initialize randomly a population $P \subset S$ with μ elements.
2: **while** the stopping-criteria is not reached **do**
3: Generate an offspring $\mathbf{x} \in S$ from P by variation.
4: Integrate $\mathbf{x}$ into the population $P := P \cup \{\mathbf{x}\}$.
5: Build the ranking of fronts $G_1, \ldots, G_h$ from P.
6: Compute the hypervolume contribution for each point $m \in G_h$.
7: Find and discard from P the point m^* with the least hypervolume contribution $P := P \backslash \{m^*\}$.
8: **end while**

The first important thing of this algorithm is that it updates the population by creating only one new individual for each iteration. The algorithm starts with an initial population μ of n elements. As second step a new individual is generated by means of random variation operators 'crossover and mutation'. The third step includes the new element in the population, therefore the size of the population is now $n + 1$. In order to take a decision of which elements are kept in the population the algorithm builds a ranking taken from the NSGA-II [7]. This ranking separates the population into h different fronts $(G_1, \ldots, G_h)$ with respect to the degree of non-dominance. Finally, the algorithm computes the contribution of each element that belongs to the front G_h according to the hypervolume. G_h is the worst ranked front, and the element with the least hypervolume contribution will be discarded from the population. This process is repeated until a certain number of iterations will be reached. SMS-EMOA keeps elements that are dominated what overcome to lose diversity for the succeeding populations. A pseudo code of SMS-EMOA can be read in Algorithm 2.

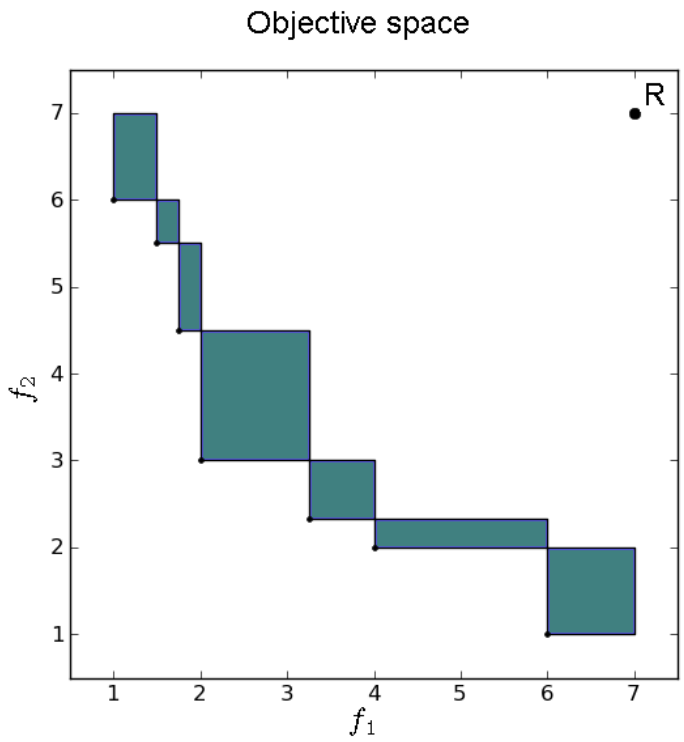


Figure 2.14: Hypervolume contribution for each point in the approximation.

2.5 Memetic Strategies

An important drawback of the multi-objective evolutionary algorithms is their rate of convergence, which is typically slow. That is why, the idea of exploiting all available knowledge about a problem is a way to accelerate the search process [54].

Memetic algorithms were introduced by Moscato [55] in 1989, who took the roots of this term from the word 'meme' presented by Dawkins in [56]. In this work it was established that meme refers to the unit of transmission in the context of cultural evolution. Speaking now about evolutionary algorithms, the term refers to improve certain elements into the population with a local search mechanism.

The local search procedures improve in this case one or some elements during the global search procedure performed by the MOEA. The combination between a local strategy and a global search using MOEAs is called memetic evolutionary algorithm. Many researchers have reported successful results by the hybridization of the MOEAs.

However, there exist many open questions such as:

- When we should apply the procedure?
- Which elements should be improved?, and
- How much we should improved these elements?

An interesting strategy can be seen in [57] where the authors presented a local search procedure called the 'Hill Climber with Sidestep' (HCS). Such mechanism allows us to perform a movement toward and along the Pareto front with and without gradient information. Within the work the authors integrated HCS into two state-of-the-art algorithms the NSGA-II [7] and SPEA2 [58]. The way to apply the local search is different for each algorithm, for NSGA-II the HCS is applied only to such elements that are the best of a given generation in order to pull the members toward the Pareto front, while the local search into SPEA2 is acting as an operator with a lower probability. Another example of a memetic strategy is proposed in [59]. In this work, the authors in particular use the DS as a local search procedure but its gradient free version in a general purpose evolutionary algorithm, in this case the MOEA/D [8]. This work makes an effort to demonstrate the advantage to integrate the DS into evolutionary algorithms in order to improve their performance which is also the aim for this work.

2.6 Parameter dependent Multi-Objective Optimization

Up to now a question still missing to answer 'How could a parameter dependent multi-objective optimization problem be treated?'. In this section, we are going to give a brief state-of-the-art of the field, since it is important for this thesis to convey what has been done a what is missing to do in this topic. Probably, the first study in the field of parameter dependent optimization was the work of A.S. Manne in the year of 1953 [60]. Since then, the study around this field keeps growing. In the following we summarize some important work within literature.

To start this summary, we are going to mention an interesting work titled 'Non-Linear Parametric Optimization' present by Bank et al. in [60]. Through this volume is intended to explain theoretical terms, notation about a parameter dependent single-objective problems and it is possible also to find important definitions, explanations about the properties of a parametric problem, and methods to solve this sort of problems. In [61] the term of 'Fitness Dynamism' is explained which occurs when the fitness landscape being optimized is non-stationary, in other words the structures as individuals or schemata have a fitness varying over time. Although, this work contains important concepts, the prior goal is to try to solve dynamic single-objective problems by using an evolutionary algorithm and not a PMOP. By following the same line in [62] a multi-population evolutionary algorithm is proposed to tackle time-dependent optimization problems which changes the optimum over time. Next to these works in [63] the author states that within static environments, the adaptation is done for the optimal solutions, but in the case of dynamic environment, the adaptation should be done toward the characteristics of environmental changes that can be adapted. The last idea remarks the importance to adapt a solution in order to get solutions when the function changes. To perform such adaptation a dynamic Pareto optimum genetic algorithm is proposed. Different to the previous approaches in [64] similarities between dynamic optimization and multi-objective optimization are presented in order to intend to solve financial time series by a multi-objective ranking method using Pareto optimality with an evolutionary algorithm.

In the following, we present some works which consider now multiple objectives. In Farina, Deb, and Amato's work [13] it is defined what dynamic multi-objective optimization is. Also, the work contains some test case applications and many results related with problems which depend of an external parameter are presented. They also establish a classification of the dynamic multi-objective optimization problems. Finally, they proposed a dynamic MOEA to get several solutions of the Pareto set and front including the influence of one external parameter. A similar work could be seen in [65]. The work in [14] makes a good insight into PMOPs, but only treats problems with uniquely one external parameter by using numerical path following algorithms. It is also included geometrical properties of the solution set, parameter dependent multi-objective optimization problem studies and connections to bifurcation theory.

Analyzing all these things, we can realize that there are many things to do in

order to improve the state-of-the-art related to MOPs and PMOPs.

3 | Directed Search for Parameter dependent MOPs

We consider again our first example 'the motorcycle problem', but now changing the goals to speed and safety. We can say that these two goals can be affected by external factors as the side wind, humidity, heat and so on. Such factors can be seen as external parameters to the original problem. These external parameters will affect the solution set of the problem. Therefore, an entire family of solution sets arises (i.e., one Pareto set for each value of the external parameter).

Here, we present a new version of the Directed Search Method to treat parameter dependent multi-objective problems (PMOPs) as defined in Equation (2.6). This new version is called λ-DS. Further, we design a novel continuation method to steer a search process over λ-space using λ-DS. To overcome the problem to need gradient information, we present also a gradient free version of this method. Finally, some numerical results and comparisons are shown to test our novel method.

3.1 λ-Directed Search Method

As it was presented in Chapter 2, the Directed Search Method showed its potential to tackle MOPs, what motivates us to extend its applicability to treat PMOPs. In order to present the ideas of DS for PMOPs, we are going to modify Equation (2.7) for our needs. In particular, it is advantageous (at least formally) to treat λ as a 'normal' parameter leading to the following problem:

$$F : \mathbb{R}^{n+l} \to \mathbb{R}^{k+l}$$

$$F(\mathbf{x},\lambda) = \begin{pmatrix} f_1(\mathbf{x},\lambda) \\ \vdots \\ f_k(\mathbf{x},\lambda) \\ \lambda \end{pmatrix} = \begin{pmatrix} g_1(\mathbf{x},\lambda) \\ \vdots \\ g_{k+l}(\mathbf{x},\lambda) \end{pmatrix}. \tag{3.1}$$

Using Equation (3.1), we are able to write Equation (2.35) which is the original DS equation, but now including the influence of the external parameter λ. Hence, given a point $(\mathbf{x},\lambda) \in \mathbb{R}^{n+l}$ in parameter space, and a vector $d \in \mathbb{R}^{k+l}$ in objective space, we are interested in finding a direction vector ν such that:

$$\lim_{t\to 0} \frac{g_i((\mathbf{x},\lambda)+t\nu) - g_i(\mathbf{x},\lambda)}{t} = \langle \nabla g_i(\mathbf{x},\lambda), \nu \rangle = d_i, \;\; i = 1,\ldots,k+l, \tag{3.2}$$

where $\nu = (\nu_f, \nu_\lambda)^T \in \mathbb{R}^{n+l}$, $v_f \in \mathbb{R}^n$, $v_\lambda \in \mathbb{R}^l$ and $d = (d_f, d_\lambda)^T \in \mathbb{R}^{k+l}$, $d_f \in \mathbb{R}^k$, $d_\lambda \in \mathbb{R}^l$.

The Jacobian matrix of the map described in Equation (3.1) is given by

$$J(\mathbf{x},\lambda) = \begin{pmatrix} \nabla_x f_1(\mathbf{x},\lambda) & \nabla_\lambda f_1(\mathbf{x},\lambda) \\ \vdots & \vdots \\ \nabla_x f_k(\mathbf{x},\lambda) & \nabla_\lambda f_k(\mathbf{x},\lambda) \\ 0 & I_l \end{pmatrix} := \begin{pmatrix} J_x & J_\lambda \\ 0 & I_l \end{pmatrix} \in \mathbb{R}^{(k+l)\times(n+l)}, \tag{3.3}$$

where $J_x \in \mathbb{R}^{k\times n}$, $J_\lambda \in \mathbb{R}^{k\times l}$, and I_l is the $(l \times l)$-identity matrix.

Using the Jacobian matrix, as in the classical DS, we can state Equation (3.2) in matrix vector notation as

$$\begin{pmatrix} J_x & J_\lambda \\ 0 & I_l \end{pmatrix} \begin{pmatrix} \nu_f \\ \nu_\lambda \end{pmatrix} = \begin{pmatrix} d_f \\ d_\lambda \end{pmatrix}. \tag{3.4}$$

By solving Equation (3.4), we notice that

$$I_l \nu_\lambda = d_\lambda \quad \Rightarrow \quad \nu_\lambda = d_\lambda \tag{3.5}$$

and

$$J_x \nu_f + J_\lambda \nu_\lambda = d_f \quad \Rightarrow \quad J_x \nu_f = d_f - J_\lambda d_\lambda \tag{3.6}$$

where the influence of λ appears since now it is treated as a 'normal' parameter.

Remark 1. *Since the influence of λ has an unknown and maybe unwanted behavior, one could use another approach, namely to obtain ν by solving*

$$\begin{pmatrix} J_x & 0 \\ 0 & I_l \end{pmatrix} \begin{pmatrix} \nu_f \\ \nu_\lambda \end{pmatrix} = \begin{pmatrix} d_f \\ d_\lambda \end{pmatrix}, \tag{3.7}$$

which means, to perform the movement in λ-direction while keeping the 'classical' search in f-direction, in other words using the classical DS to get the direction ν(i.e., ignoring the influence of λ).

Either using Equation (3.4) or Equation (3.7) a direction $\nu \in \mathbb{R}^{n+l}$ can be computed by solving a system of linear equations. In practice, usually the number of parameters is higher than the number of objectives for a PMOP, which implies that the solution of the system is not going to be unique. To overcome this situation, the solution with the smallest Euclidean norm that solves the system can be chosen leading to

$$\nu_+ = J(\mathbf{x}_0, \lambda_0)^+ d, \tag{3.8}$$

where $J(\mathbf{x}_0, \lambda_0)^+ \in \mathbb{R}^{(n+l)\times(k+l)}$ denotes the pseudo-inverse of $J(\mathbf{x}_0, \lambda_0)$.

The next result states the pseudo-inverse of J under certain (mild) assumptions:

Proposition 1. *Let $J := \begin{pmatrix} J_x & J_\lambda \\ 0 & I_l \end{pmatrix}$ be as in Equation (3.3), further let $rank(J_x) = k$, and J_λ be a invertible matrix. Then the pseudo-inverse of J is given by*

$$J^+ = \begin{pmatrix} J_x^+ & -J_x^+ J_\lambda \\ 0 & I_l \end{pmatrix}. \tag{3.9}$$

Proof. Let $J := \begin{pmatrix} J_x & J_\lambda \\ 0 & I_l \end{pmatrix}$ where $rank(J) = k + l$ then as defined in [66]

$$J^+ = J^T (J J^T)^{-1}. \tag{P1.1}$$

It is

$$JJ^T = \begin{pmatrix} J_x & J_\lambda \\ 0 & I_l \end{pmatrix} \begin{pmatrix} J_x^T & 0 \\ J_\lambda^T & I_l \end{pmatrix} = \begin{pmatrix} J_x J_x^T + J_\lambda J_\lambda^T & J_\lambda \\ J_\lambda^T & I_l \end{pmatrix}. \tag{P1.2}$$

We know that the inverse of a block matrix is given by

$$\begin{pmatrix} A & B \\ C & D \end{pmatrix}^{-1} = \begin{pmatrix} S_D^{-1} & -A^{-1}BS_A^{-1} \\ -D^{-1}CS_A^{-1} & S_A^{-1} \end{pmatrix}^{-1}, \tag{P1.3}$$

where

$$S_A = D - CA^{-1}B, \tag{P1.4}$$

$$S_D = A - BD^{-1}. \tag{P1.5}$$

Writing JJ^T as in Equation (P1.3), we obtain

$$A = J_x J_x^T + J_\lambda J_\lambda^T, \tag{P1.6}$$

$$B = J_\lambda, \tag{P1.7}$$

$$C = J_\lambda^T, \tag{P1.8}$$

$$D = I_l. \tag{P1.9}$$

Using Equations (P1.6), (P1.7), (P1.8) and (P1.9), we get for S_A and S_D

$$S_A = I_l - J_\lambda^T (J_x J_x^T + J_\lambda J_\lambda^T)^{-1} J_\lambda, \tag{P1.10}$$

$$S_D = J_x J_x^T + J_\lambda J_\lambda^T - J_\lambda I_l J_\lambda^T = J_x J_x^T + J_\lambda J_\lambda^T - J_\lambda J_\lambda^T = J_x J_x^T. \tag{P1.11}$$

By replacing these values in the members of Equation (P1.3), we get the following:

$$S_D^{-1} = (J_x J_x^T)^{-1}, \tag{P1.12}$$

$$\begin{aligned}
-A^{-1}BS_A^{-1} &= -((J_x J_x^T + J_\lambda J_\lambda^T)^{-1} J_\lambda (I_l - J_\lambda^T (J_x J_x^T + J_\lambda J_\lambda^T)^{-1} J_\lambda)^{-1}) \\
&= -((I_l - J_\lambda^T (J_x J_x^T + J_\lambda J_\lambda^T)^{-1} J_\lambda) J_\lambda^{-1} (J_x J_x^T + J_\lambda J_\lambda^T))^{-1} \\
&= -(J_\lambda^{-1}(J_x J_x^T + J_\lambda J_\lambda^T) - J_\lambda^T (J_x J_x^T + J_\lambda J_\lambda^T)^{-1} J_\lambda J_\lambda^{-1} (J_x J_x^T + J_\lambda J_\lambda^T))^{-1} \\
&= -(J_\lambda^{-1}(J_x J_x^T + J_\lambda J_\lambda^T) - J_\lambda^T)^{-1} \\
&= -(J_\lambda^{-1} J_x J_x^T + J_\lambda^{-1} J_\lambda J_\lambda^T - J_\lambda^T)^{-1} \\
&= -(J_\lambda^{-1} J_x J_x^T + J_\lambda^T - J_\lambda^T)^{-1} \\
&= -(J_\lambda^{-1} J_x J_x^T)^{-1} \\
&= -((J_x J_x^T)^{-1} J_\lambda)
\end{aligned}$$

$$-A^{-1}BS_A^{-1} = -((J_xJ_x^T)^{-1}J_\lambda), \tag{P1.13}$$

$$-D^{-1}CS_A^{-1} = -(I_lJ_\lambda^T(J_xJ_x^T)^{-1}) = -(J_\lambda^T(J_xJ_x^T)^{-1}), \text{ and} \tag{P1.14}$$

$$S_A^{-1} = (I_l - J_\lambda^T(J_xJ_x^T + J_\lambda J_\lambda^T)^{-1}J_\lambda)^{-1}, \tag{P1.15}$$

therefore

$$(JJ^T)^{-1} = \begin{pmatrix} (J_xJ_x^T)^{-1} & -((J_xJ_x^T)^{-1}J_\lambda) \\ -(J_\lambda^T(J_xJ_x^T)^{-1}) & (I_l - J_\lambda^T(J_xJ_x^T + J_\lambda J_\lambda^T)^{-1}J_\lambda)^{-1} \end{pmatrix}. \tag{P1.16}$$

Finally, by solving Equation (P1.1), we obtain:

$$J^+ = J^T(JJ^T)^{-1}$$

$$= \begin{pmatrix} J_x^T & 0 \\ J_\lambda^T & I_l \end{pmatrix} \begin{pmatrix} (J_xJ_x^T)^{-1} & -((J_xJ_x^T)^{-1}J_\lambda) \\ -(J_\lambda^T(J_xJ_x^T)^{-1}) & (I_l - J_\lambda^T(J_xJ_x^T + J_\lambda J_\lambda^T)^{-1}J_\lambda)^{-1} \end{pmatrix} \text{ since } J_x^+ = J_x^T(J_xJ_x^T)^{-1},$$

$$= \begin{pmatrix} J_x^+ & -(J_x^+J_\lambda) \\ 0 & I_l \end{pmatrix}.$$

Hence, $J^+ = \begin{pmatrix} J_x^+ & -(J_x^+J_\lambda) \\ 0 & I_l \end{pmatrix}$ as claimed. □

At last, we complete all the neccesary elements to compute ν_+, therefore, we can write Equation (3.8) using our obtained result in Equation (3.9) as follows:

$$\nu_+ = J^+d = \begin{pmatrix} J_x^+ & -(J_x^+J_\lambda) \\ 0 & I_l \end{pmatrix} \begin{pmatrix} d_f \\ d_\lambda \end{pmatrix} = \begin{pmatrix} J_x^+d_f - J_x^+J_\lambda d_\lambda \\ d_\lambda \end{pmatrix} \tag{3.10}$$

Thus, taking the direction ν_+, an initial point $(\mathbf{x}_0, \lambda_0)$ in parameter space and a given stepsize t, we can expect the largest progress in direction d in objective space, this new solution is obtained by computing

$$\begin{pmatrix} \mathbf{x}_{new} \\ \lambda_{new} \end{pmatrix} = \begin{pmatrix} \mathbf{x}_0 \\ \lambda_0 \end{pmatrix} + t\begin{pmatrix} \nu_f \\ \nu_\lambda \end{pmatrix}, \tag{3.11}$$

where $t \in \mathbb{R}_+$.

The theory related to the DS to tackle PMOPs has been established, but it delivers us only one point up to here. In the following, we will discuss a descent and a continuation method based on the above observations.

3.2 λ-DS Descent Method

In the above section, we have seen how to compute a direction $v \in \mathbb{R}^{n+l}$ of a given point $(\mathbf{x}_0, \lambda_0)$ to steer the search in d-direction. Analog to the classical DS, we have to solve an initial value problem (IVP) to obtain a curve of points toward the solution set. To be more precise, given a point $(\mathbf{x}_0, \lambda_0) \in \mathbb{R}^{n+l}$ and a direction $d \in \mathbb{R}^{k+l}$ where $d_f \leq_p 0$ the use of Equation (3.8) leads to the (numerical) solution of the following initial value problem:

$$\begin{aligned} z(0) &= (\mathbf{x}_0, \lambda_0) \in \mathbb{R}^{n+l} \ . \\ \dot{z}(t) &= J(\mathbf{x}(t), \lambda(t))^{+} d \ . \end{aligned} \tag{3.12}$$

To investigate the result obtained by Equation (3.12), we start the following analysis by investigating the end point of Equation (3.12). Given $z_f = (\mathbf{x}_f, \lambda_f)$ such that $J^{+}(z_f)d = 0$, it is

$$\begin{pmatrix} J_x^{+} d_f - J_x^{+} J_\lambda d_\lambda \\ d_\lambda \end{pmatrix} = 0. \tag{3.13}$$

In Equation (3.13), we notice that $d_\lambda = 0$ and $J_x^{+} d_f = 0$. Therefore, this implies that $\mathbf{x}_f$ is a KKT point of F_λ with KKT weight $\alpha = -d_f$.

In the following, we investigate solutions of Equation (3.12) qualitatively. Let $\gamma : [0, t_f] \to \mathbb{R}^{n+l}$ be such a solution, and let t_c be the smallest value of $t \geq 0$ such that

$$\nexists \nu \in \mathbb{R}^{n+l} : J(\dot{z}(t))\nu = d. \tag{3.14}$$

We will call t_c the critical value and $\gamma(t_c)$ the critical point of Equation (3.12).

A critical point $z_c = (\mathbf{x}_c, \lambda_c)$ is the end of the movement in d_f-direction. To prove that our new approach work in a proper manner, we investigate the relation with a widely used mathematical method the Normal Boundary Intersection method (NBI). By adapting the NBI subproblem to the current context, it can be stated as follows:

$$\begin{aligned} &\max_{\mathbf{x},\lambda,t} \ t \\ &F(\mathbf{x}, \lambda) = F(\mathbf{x}_0, \lambda_0) + td. \\ &\lambda \in \Lambda, \ \mathbf{x} \in S \end{aligned} \tag{3.15}$$

Proposition 2. *Let* $(\mathbf{x}^*, \lambda^*)$ *be the critical point of Equation* (3.12)*, then it is a local solution of Equation* (3.15)*.*

Proof. Let $g(\mathbf{x}, \lambda, t) := t$ and $h_i(\mathbf{x}, \lambda, t) := f_i(\mathbf{x}_0, \lambda_0) + td_i - f_i(\mathbf{x}, \lambda)$. We can assume that $(\mathbf{x}^*, \lambda^*)$ is not a local solution of Equation (3.15). Then, there exist $\nu = (\tilde{\nu_f}, \tilde{\nu_\lambda}, \nu_{n+l+1}) \in \mathbb{R}^{n+l+1}$, $\tilde{\nu_f} \in \mathbb{R}^n$, $\tilde{\nu_\lambda} \in \mathbb{R}^l$ and $t^* \in \mathbb{R}$ such that

$$\langle \nabla g(\mathbf{x}^*, \lambda^*, t^*), \nu) \rangle = \left\langle \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}, \begin{pmatrix} \tilde{\nu_f} \\ \tilde{\nu_\lambda} \\ \nu_{n+l+1} \end{pmatrix} \right\rangle < 0, \text{ and} \tag{P2.1}$$

$$\langle \nabla h_i(\mathbf{x}^*, \lambda^*, t^*), \nu) \rangle = \left\langle \begin{pmatrix} -\nabla_x f_i(\mathbf{x}^*, \lambda^*) \\ -\nabla_\lambda f_i(\mathbf{x}^*, \lambda^*) \\ d_i \end{pmatrix}, \begin{pmatrix} \tilde{\nu_f} \\ \tilde{\nu_\lambda} \\ \nu_{n+l+1} \end{pmatrix} \right\rangle = 0,\ i = 1, .., k \tag{P2.2}$$

By Equation (P2.1), it follows that $\nu_{n+l+1} \neq 0$, and by Equation (P2.2), we have that

$$\langle \nabla f_i(\mathbf{x}^*, \lambda^*), \tilde{\nu} \rangle = \nu_{n+l+1} d_i,\ i = 1, \ldots, k, \tag{P2.3}$$

where $\tilde{\nu} = (\tilde{\nu_f}, \tilde{\nu_\lambda})^T$.

Hence, for $\tilde{\nu} := \frac{1}{\nu_{n+l+1}} \tilde{\nu}$ it is $J(\mathbf{x}^*, \lambda^*)\tilde{\nu} = d$ which contradicts that $(\mathbf{x}^*, \lambda^*)$ is a critical point of Equation (3.12). □

Remark 2. *Otherwise, local solutions of Equation* (3.15) *are also potential critical points of Equation* (3.12)*: Let* $\mathbf{x}^{**}$ *be a solution of Equation* (3.15) *and assume that there exists a* $\nu \in \mathbb{R}^{n+l}$ *such that* $J(\mathbf{x}^{**}, \lambda^{**})\nu = d$*. Then,* $\tilde{\nu} = (\nu, -1) \in \mathbb{R}^{n+l+1}$ *satisfies Equations* (P2.1) *and* (P2.2) *which is in contradiction to the assumption of* $(\mathbf{x}^{**}, \lambda^{**})$*.*

3.3 A Continuation Method Based on the λ-DS Approach

In Chapter 2, it was explained how DS is able to perform a search toward and along the Pareto front of a MOP. Thus, in the context of PMOPs, we are particularly interested in movements over λ-space since this is still missing. The DS theory for PMOPs has been presented, thus we are in the position to propose a new predictor-corrector method for the continuation along λ-space which means to move from a point $F(\mathbf{x}, \lambda)$ on the Pareto front of the MOP for a value of λ orthogonal to this front and along the family of fronts.

Analog to the proposition of the classical continuation method using DS, what we are looking for is an approach that allows us to perform a movement over the family of Pareto fronts, obviously without computing any Hessian matrix as for the tradional continuation methods. The central idea of this new continuation method over λ-space is to use a predictor direction in the λ-DS in order to get a predictor point over λ-space. The predictor direction used for this is going to be based on the geometry of the family of Pareto fronts, what we are going to explain in the following. Next to this, we have to find a way to correct back to the family of Pareto fronts, since those predictor points would be a bit far of the solution set, what leads to the corrector method. In the following, we will discribe all these steps to realize such a continuation method.

To start, we are going to introduce the normal vector η to the linearization of the family of Pareto fronts which will allow us to compute the predictor direction. In [3] it is stated that if $\mathbf{x}$ is a Pareto point with a related convex weight α, then α is orthogonal to the linearization of the tangent space at $F(\mathbf{x})$. Therefore, to adapt this definition to the new setting, we have now to build the complete system by adding the weight associated to λ-space, in this case β as follows:

$$\left\langle \begin{pmatrix} J_x & J_\lambda \\ 0 & I_l \end{pmatrix} \begin{pmatrix} \nu_f \\ \nu_\lambda \end{pmatrix}, \begin{pmatrix} \alpha \\ \beta \end{pmatrix} \right\rangle = 0. \tag{3.16}$$

To get η, we have to solve Equation (3.16) for β as follows:

$$= \langle \begin{pmatrix} J_x\nu_f + J_\lambda\nu_\lambda \\ \nu_\lambda \end{pmatrix}, \begin{pmatrix} \alpha \\ \beta \end{pmatrix} \rangle$$

$$= \langle J_x\nu_f + J_\lambda\nu_\lambda, \alpha\rangle + \langle \nu_\lambda, \beta\rangle$$

$$= \langle J_x\nu_f, \alpha\rangle + \langle J_\lambda\nu_\lambda, \alpha\rangle + \langle \nu_\lambda, \beta\rangle (\text{ from classical DS } \langle J_x\nu_f, \alpha\rangle = 0)$$

$$= \langle \nu_\lambda, {J_\lambda}^T\alpha\rangle + \langle \nu_\lambda, \beta\rangle = 0$$

$$\beta = -{J_\lambda}^T\alpha. \tag{3.17}$$

Using the obtained result of Equation (3.17), we study the normal vector η of the tangent space of $F(P_{S,\Lambda})$ at $F(\mathbf{x}, \lambda)$ (a graphical representation of η is presented in Figure 3.1), because we are in the setting to make the following conjecture about it:

$$\eta = \begin{pmatrix} \alpha \\ -{J_\lambda}^T\alpha \end{pmatrix}, \tag{3.18}$$

further

$$\langle \begin{pmatrix} J_x & J_\lambda \\ 0 & I_l \end{pmatrix} \begin{pmatrix} \nu_f \\ \nu_\lambda \end{pmatrix}, \begin{pmatrix} \alpha \\ -{J_\lambda}^T\alpha \end{pmatrix} \rangle = 0 \text{ for all } \nu \in \mathbb{R}^{n+l}. \tag{3.19}$$

The previous conjecture leads us to the following result, which is going to be crucial for our continuation method over λ-space.

Proposition 3. *Let $\mathbf{y}^*$ be a (locally) efficient point and $(\mathbf{x}^*, \lambda^*)$ a corresponding (locally) Pareto optimal point (i.e. $g(\mathbf{x}^*, \lambda^*) = \mathbf{y}^*$) of an unconstrained parametric multi-objective optimization problem. Let*

$$p_\eta(\mathbf{x}, \lambda) = \sum_{i=1}^{k+l} \eta_i g_i(\mathbf{x}, \lambda) \tag{3.20}$$

denote a convex combination of the objectives for which $(\mathbf{x}^*, \lambda^*)$ is a stationary point (and therefore fulfills the Karush-Kuhn-Tucker condition).

Then for the vector η we have that:
η is an element of the orthogonal complement to the vector subspace (image $g'(\mathbf{x}^*, \lambda^*)) \subset \mathbb{R}^{k+l}$, where $g'(\mathbf{x}^*, \lambda^*)$ is the Jacobian matrix of g at the point $(\mathbf{x}^*, \lambda^*)$.

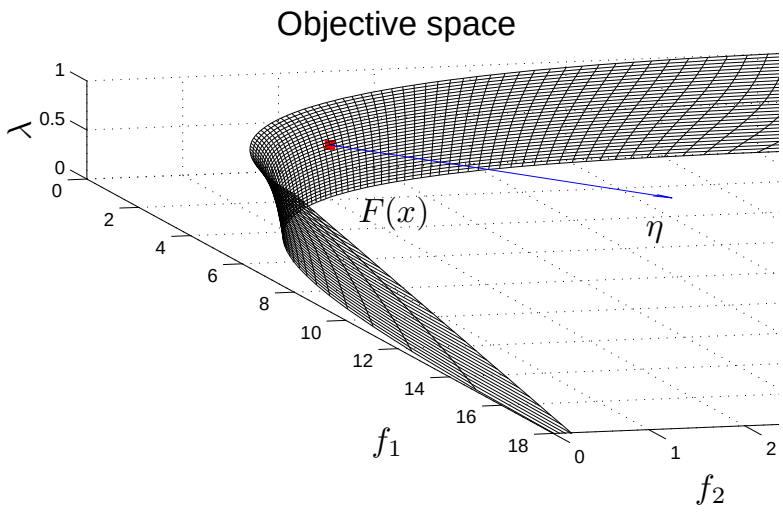


Figure 3.1: Orthogonal vector to the linearization of the family of Pareto fronts, the arrow represents η and the surface is the family of Pareto fronts.

Proof. From the first-order Karush-Kuhn-Tucker condition we get:

$$\nabla p_\eta(\mathbf{x}^*, \lambda^*) = 0 \Longleftrightarrow \sum_{i=1}^{k+l} \eta_i \nabla g_i(\mathbf{x}^*, \lambda^*) = 0 \Longleftrightarrow \eta^T \begin{pmatrix} \nabla g_1(\mathbf{x}^*, \lambda^*)^T \\ \vdots \\ \nabla g_{k+l}(\mathbf{x}^*, \lambda^*)^T \end{pmatrix} = 0. \quad (3.21)$$

Because of $\begin{pmatrix} \nabla g_1(\mathbf{x}^*, \lambda^*)^T \\ \vdots \\ \nabla g_{k+l}(\mathbf{x}^*, \lambda^*)^T \end{pmatrix} = g'(\mathbf{x}^*, \lambda^*)$ it follows that η is orthogonal to the columns of the Jacobian matrix $g'(\mathbf{x}^*, \lambda^*)$ and thus to the image of the linear mapping $g'(\mathbf{x}^*, \lambda^*)$. □

As a corollary one obtains the statement $rank\ g'(\mathbf{x}^*, \lambda^*) < k + l$, i.e. the linear mapping $g'(\mathbf{x}^*, \lambda^*) : \mathbb{R}^{n+l} \to \mathbb{R}^{k+l}$ is not surjective in a Pareto optimal point $(\mathbf{x}^*, \lambda^*)$. If $rank\ g'(\mathbf{x}^*, \lambda^*) = k+l-1$, from the Proposition 3 we can conclude furthermore the uniqueness of the assignment of a weight vector η (and therefore of a scalar-valued function p_η) to a Pareto optimal point $(\mathbf{x}^*, \lambda^*)$.

If in an appropriate neighborhood of $g(\mathbf{x}^*, \lambda^*)$ the image set $g(\mathbb{R}^{n+l})$ behaves like a bordered differentiable manifold of dimension $k + l$, the geometrical meaning of the

vector η can be put in even more concrete forms:

Proposition 4. *Let $\mathbf{y}^*$ be a globally efficient point and $(\mathbf{x}^*, \lambda^*)$ an associated globally Pareto optimal point (i.e. $g(\mathbf{x}^*, \lambda^*) = \mathbf{y}^*$) of an unconstrained parametric multi-objective optimization problem. Let p_η denote a convex combination of the objectives for which $(\mathbf{x}^*, \lambda^*)$ is a stationary point (and therefore fulfills the Karush-Kuhn-Tucker condition). In addition be:*

- *rank $g'(\mathbf{x}^*, \lambda^*) = k + l - 1$*
- *There is an open neighborhood $N(\mathbf{y}^*)$ of $\mathbf{y}^*$, so that $g(\mathbb{R}^{n+l}) \cap N(\mathbf{y}^*) =: M$ is a bordered differentiable manifold of dimension k.*

Then we have:

(a) $\mathbf{y}^* \in \partial M$, where the $(k-1)$-dimensional border manifold of M is denoted by ∂M.

(b) η is oorthogonal to the tangent plane $T_{\mathbf{y}^*}\partial M$ of ∂M in $\mathbf{y}^*$.

Proof. Assertion (a) will be proven by contradiction. Assume therefore that $\mathbf{y}^*$ is not an element of ∂M. It follows that $\mathbf{y}^*$ is an inner point of M, i.e. that there is a δ-neighborhood $N(\mathbf{0}, \delta)$ of $\mathbf{0} \in \mathbb{R}^{k+l}$ with $\mathbf{y}^* + N(\mathbf{0}, \delta) \subseteq M$. Now choose a vector $\nu \in \mathbb{R}_+^{k+l}$. Then there is a $\lambda \in \mathbb{R}$, $\lambda > 0$ with $(-\lambda) \cdot \nu \in N(\mathbf{0}, \delta)$, and $\tilde{\mathbf{y}} := (\mathbf{y}^* - \lambda\nu) \in M \subset g(\mathbb{R}^{n+l})$ is true. Because of $\mathbf{y}^* - \tilde{\mathbf{y}} = \lambda\nu \in \mathbb{R}_+^{k+l}$ we can conclude that $\tilde{\mathbf{y}} \leq \mathbf{y}^*$ (and $\tilde{\mathbf{y}} \in g(\mathbb{R}^{n+l})$), in contradiction to the global efficiency of $\mathbf{y}^*$.

The assertion (b) follows from the Proposition 3, if one can show that $T_{y^*}\partial M = \mathit{image}\ g'(\mathbf{x}^*, \lambda^*)$. We will prove this by contradiction. Therefore, be $T_{y^*}\partial M \neq \mathit{image}\ g'(\mathbf{x}^*, \lambda^*)$. As we assumed that both $T_{y^*}\partial M$ and $\mathit{image}\ g'(\mathbf{x}^*, \lambda^*)$ are $(k-1)$-dimensional vector subspaces of the $\mathbb{R}^{k+l}$, it follows that $\mathit{image}\ g'(\mathbf{x}^*, \lambda^*) \subseteq T_{y^*}\partial M$, and hence: There is a vector vector $\delta\mathbf{y} \in \mathit{image}\ g'(\mathbf{x}^*, \lambda^*)$, so that $\delta\mathbf{y}$ can be represented by $\delta\mathbf{y} = \xi + \vartheta$, where $\xi \in T_{y^*}\partial M$, $\vartheta \in (T_{y^*}\partial M)^\perp$, $\vartheta \neq 0$. Let us denote the corresponding inverse-image vector by $\delta(\mathbf{x}, \lambda)$, i.e. $g'(\mathbf{x}^*, \lambda^*) \cdot \delta(\mathbf{x}, \lambda) = \delta\mathbf{y}$.

Now for a sufficiently small $a \in \mathbb{R}_+$ consider the curve

$$\tau : \begin{cases} (-a, +a) \to \mathbb{R}^{k+l} \\ t \mapsto g((\mathbf{x}^*, \lambda^*) + t \cdot \delta(\mathbf{x}, \lambda)) \end{cases} . \tag{3.22}$$

By virtue of $\tau'(0) = g'(\mathbf{x}^*, \lambda^*) \cdot \delta(\mathbf{x}, \delta) = \delta\mathbf{y} = \xi + \vartheta$ either $+\tau'(0)$ or $-\tau'(0)$ is an element of the outward directed tangent space of the bordered manifold M in the point $\mathbf{y}^*$. For one of the two possible signs of t and for sufficiently small $|t|$ (so that $\tau(t)$ is represented sufficiently well by the linear approximation $\tau(0) + \tau'(0) \cdot t$) the image points of the curve τ are therefore situated outside M. This is a contradiction to the definition of M, so that the assumption $T_{y^*}\partial M \neq$ image $g'(\mathbf{x}^*, \lambda^*)$ must be false. □

Thanks to the previous results, which confirm our conjecture over η, we are now able to perform a movement in $\lambda-$direction given a point $(\mathbf{x}_0, \lambda_0) \in P_{S,\Lambda}$ and $\eta \in \mathbb{R}^{k+l}$ by using λ-DS.

For a predictor point using a direction d, we can assume that:

(I) d is perpendicular with respect to the tangent space of Pareto front of F_λ and

(II) d is perpendicular to the orthogonal vector of the linearization of the family of Pareto fronts η.

By analyzing the previous assumptions, we know that $d_f \perp span\{\alpha\}^\perp$ which implies that $d_f = \mu\alpha$ and since $\langle d, \eta \rangle = 0$, we can write the product by replacing η in the following form:

$$\alpha^T J_\lambda d_\lambda = \mu ||\alpha||_2^2 \tag{3.23}$$

where $d_\lambda \in \mathbb{R}^l$ solves Equation (3.23).

Therefore, to select a predictor direction d_{pred} to perform the search using λ-DS, one has to choose

$$d_{pred} = (\alpha, d_\lambda)^T, \tag{3.24}$$

such that d_λ satisfies Equation (3.23).

Remark 3. *For the special case when the dimension of λ is equal to 1 ($\lambda \in \mathbb{R}$), we are going to have two possibilities for d:*

- *if* $\alpha^T J_\lambda \neq 0$*, the value of the predictor direction is going to be set by*

$$d_\lambda = \frac{\mu||\alpha||_2^2}{\alpha^T J_\lambda} \tag{3.25}$$

and

- *if* $\alpha^T J_\lambda = 0$*, it will imply that* $\mu = 0$ *which leds to set* $d_f = 0$ *and since* d *has to be orthogonal to* $\eta = \binom{\alpha}{0}$ *implies that* $d = \binom{0}{\alpha}$.

In order to compute a new point over λ-space using λ-DS, the following point

$$p := (x_0, \lambda) + t\nu \tag{3.26}$$

can be chosen as a predictor point, where t represents the stepsize, which is going to be computed as the classical DS by using the following equation:

$$h_i = \frac{\epsilon}{\langle \nabla g_i, \nu \rangle}, \ i = 1, ...k + l, \tag{3.27}$$

where ϵ represents a small distance between our current point and the desired point, being the other elements the ones that we already have, hence the step size is going to be selected by the following equation:

$$t := \min_{i=1,...,k+l} h_i. \tag{3.28}$$

Given a predictor point p, the next step is to obtain a corrector point which is the subsequent solution along the curve. To perform this task, we are going to take p as initial point and solving numerically Equation (3.1) fixing the value of λ, to be more precise $d_\lambda = 0$. To compute the corrector point two approaches figure:

1. To use λ-DS descent method using $d_f = -\alpha$.

2. To solve the following minimization problem by fixing the value of α

$$\min_x \sum_{i=1}^{k} \alpha_i f_i(x, \lambda). \tag{3.29}$$

Now, two questions come to our minds, which approach do we have to use and why? The answer is going to appear in the final section of this chapter, since up to here, we do not have all the elements to provide a complete answer.

Algorithm 3 presents a possible way to implement the continuation method over λ-space from an initial KKT point $(\mathbf{x}, \lambda)$. The algorithm is composed by all the previous results in order to make it clear, being easy to program it.

Algorithm 3 Continuation method over λ-space using λ-DS for $\lambda \in \mathbb{R}$

Require: An initial solution $(\mathbf{x}, \lambda)$, a convex weight α, a threshold $\epsilon \in \mathbb{R}_+$, and the number of samples $ns \in \mathbb{Z}_+$.

Ensure: A set of candidate solutions $(\mathbf{x}_i, \lambda_i)$.

1: $i := 0$.
2: **while** $i < ns$ **do**
3: Compute d_λ as in Equation (3.25).
4: Set d_{pred} as in Equation (3.24).
5: Compute ν as in Equation (3.8).
6: Compute t using Equations (3.27) and (3.28).
7: $p_i = (\mathbf{x}_i, \lambda_i) - \text{sgn}(d_\lambda) t\nu$.
8: Compute $(\mathbf{x}_{i+1}, \lambda_{i+1})$ by using Equation (3.29) or using Equation (3.4) setting $d_f = -\alpha$ and $d_\lambda = 0$, using in both p_i as the initial point and fixing the value of α.
9: $i := i + 1$.
10: **end while**

3.4 On Stochastic Local Search in PMOPs

In the following section, we are going to explain the behavior of the stochastic local search for PMOPs. To do this, we have to investigate the relation of search directions $\nu \in \mathbb{R}^{n+l}$ in parameter space and the related movement in objective space at a given point $(\mathbf{x}, \lambda) \in \mathbb{R}^{n+l}$. To start, we can express the latter by $J(\mathbf{x}, \lambda)\nu$, which means that:

- If a line search along ν is performed, then a new point is given by $(\mathbf{x}, \lambda)_{new} = (\mathbf{x}, \lambda) + t\nu$, where t is the step size.

- If $(\mathbf{x}_{new}, \lambda_{new})$ is chosen from a neighborhood $N(\mathbf{x}_0, \lambda_0)$ of $(\mathbf{x}_0, \lambda_0)$, then we are in the same setting. To see this, we define $\nu := (\mathbf{x}_{,new}\, \lambda_{new}) - (\mathbf{x}_0, \lambda_0)$, and we have that $(\mathbf{x}_{new}, \lambda_{new}) = (\mathbf{x}_0, \lambda_0) + \nu$.

Looking at the second statement, in order to see what the influence of λ produces within local search, we will consider the extreme cases. Firstly, when our point is far

away from the solution set and then when it is near. To perform our experiments, we consider the PMOP stated in Equation (2.14).

3.4.1 Point far away from the Pareto set

For this setting, we can assume a given point $(\mathbf{x}, \lambda)$ which is far away from the Pareto set. In this case, the objective gradients nearly point into the same direction (considering only $\mathbf{x}$), what told us that in the extreme case by setting

$$g := \nabla_x f_1(\mathbf{x}, \lambda) \tag{3.30}$$

then

$$\nabla_x f_i(\mathbf{x}, \lambda) = \mu_i \nabla f_1(\mathbf{x}, \lambda) \; i = 1, \ldots, k. \tag{3.31}$$

So, for a search direction $\nu \in \mathbb{R}^{n+l}$, we have that

$$J\nu = \begin{pmatrix} J_x\nu_f + J_\lambda\nu_\lambda \\ \nu_\lambda \end{pmatrix}, \tag{3.32}$$

where $J_\lambda\nu_\lambda$ represents the influence of λ. By decomposing Equation (3.32) the following result arises

$$J_x\nu_f = \begin{pmatrix} g^T\nu \\ \vdots \\ g^T\nu \end{pmatrix} = g^T\nu_f \begin{pmatrix} \mu_1 \\ \vdots \\ \mu_k \end{pmatrix}. \tag{3.33}$$

In order to see the impact of λ within local search, we can consider the approach from Equation (3.4), the one that uses a direction for λ-space. By taking 100 points randomly sampled from the neighborhood of a point $(\mathbf{x}_0, \lambda_0)$ and varying the value of both components $\mathbf{x}$ and λ, we can see the movement produced by the influence of λ in Figure 3.2 and in Figure 3.3 how the points were taken in parameter space. For these figures, we used the following settings: $(\mathbf{x}, \lambda) = (10.0, 45.2, 1.0)^T$ and the radio for the test points is 2.0 in $\mathbf{x}$ and 0.3 in λ, the generated points are defined far away from the Pareto set.

Now, considering the approach described in Equation (3.7), we can eliminate the influence of λ, which means to sample the points only by varying the value of $\mathbf{x}$. In this case, we take the same point defined above, using now, a radio of 2.0 in $\mathbf{x}$ and

0.0 in λ for the samples. In Figures 3.4 and 3.5, we can see the new movement and how the samples were taken for this case.

By analyzing both approaches, we can conclude that the influence of λ does not help in the search toward the family of Pareto sets, since it affects in a bad way the movement, on the other hand, if we remove this influence, we can detect a clear movement in direction $d_f = \pm(\mu_1, \ldots, \mu_k)^T \in \mathbb{R}^k$ in objective space what is not the case in parameter space. The latter told us that if we choose randomly a point $(\mathbf{x}_{new}, \lambda_{new})$ from an small neighborhood of $\mathbf{x}$ without altering λ, we can expect that the difference $F(\mathbf{x}_{new}, \lambda_{new}) - F(\mathbf{x}, \lambda)$ is a multiple of d_f. In the case that $F(\mathbf{x}_{new}, \lambda_{new}) - F(\mathbf{x}, \lambda) >_p 0$ which means that $(\mathbf{x}, \lambda)$ dominates $(\mathbf{x}_{new}, \lambda_{new})$, we can flip the search and use $(\hat{\mathbf{x}}, \lambda) := (\mathbf{x}, \lambda) - \nu$. For PMOPs, we have to pay attention since the concept of dominance does not apply for λ-space, that is why, the promising results are in the side to delete this component for the search. Finally, just to see the difference between movements, we present both in Figure 3.6. As a final conclusion, the influence of λ would be deleted, being the option to use the classical DS the best way to reach the family of Pareto sets when we are far away.

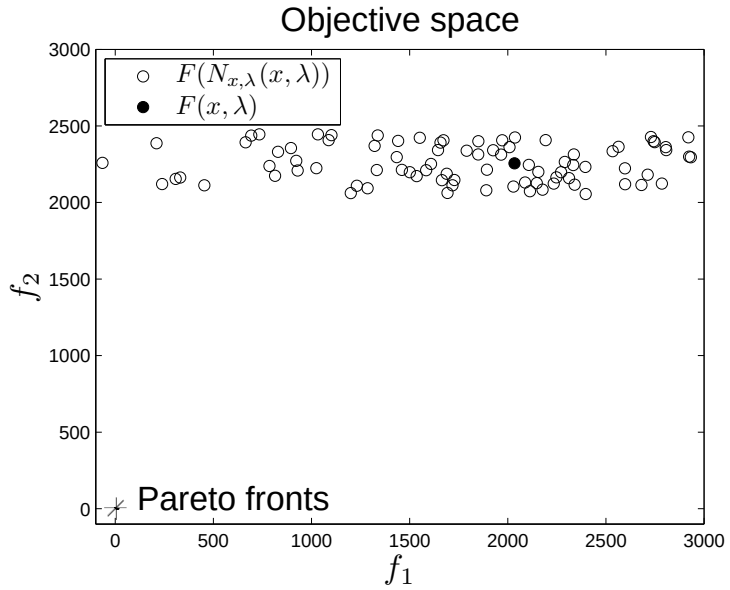


Figure 3.2: 100 randomly test points in the neighborhood of $(\mathbf{x}, \lambda)$ altering both components $\mathbf{x}$ and λ in objective space.

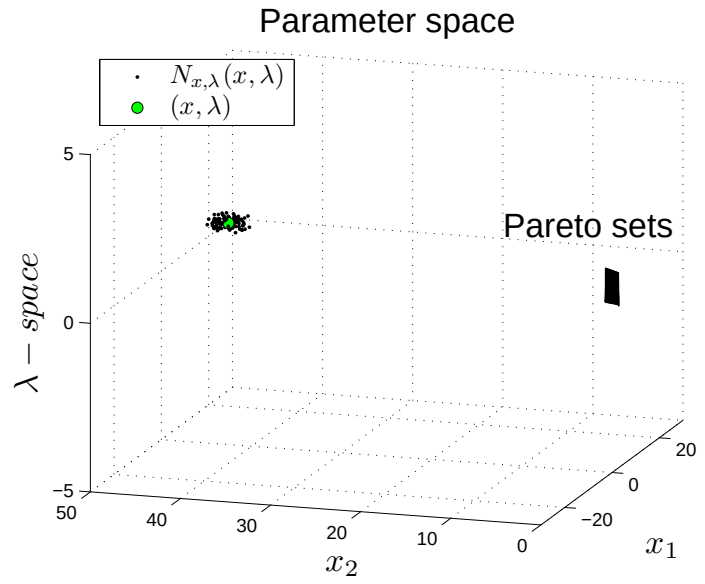


Figure 3.3: 100 randomly test points around a given point $(\mathbf{x}, \lambda)$ altering both components $\mathbf{x}$ and λ in parameter space.

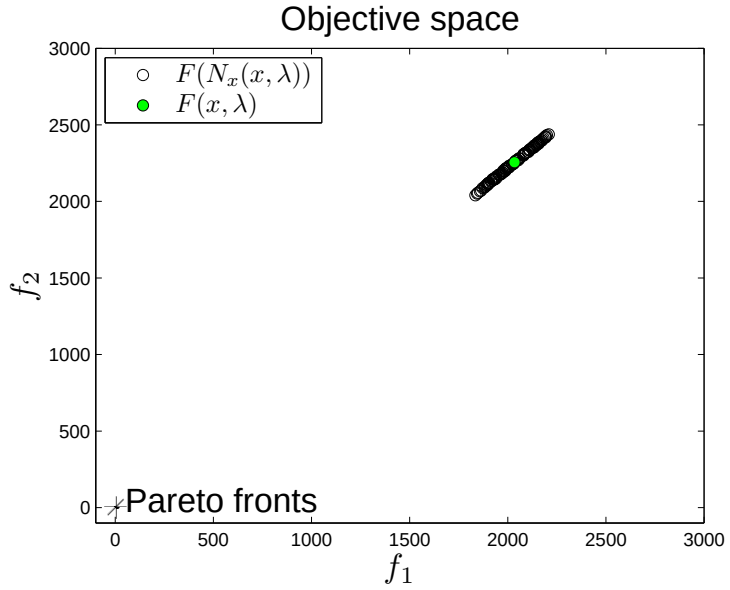


Figure 3.4: 100 randomly test points in the neighborhood of $(\mathbf{x}, \lambda)$ altering just $\mathbf{x}$ in objective space.

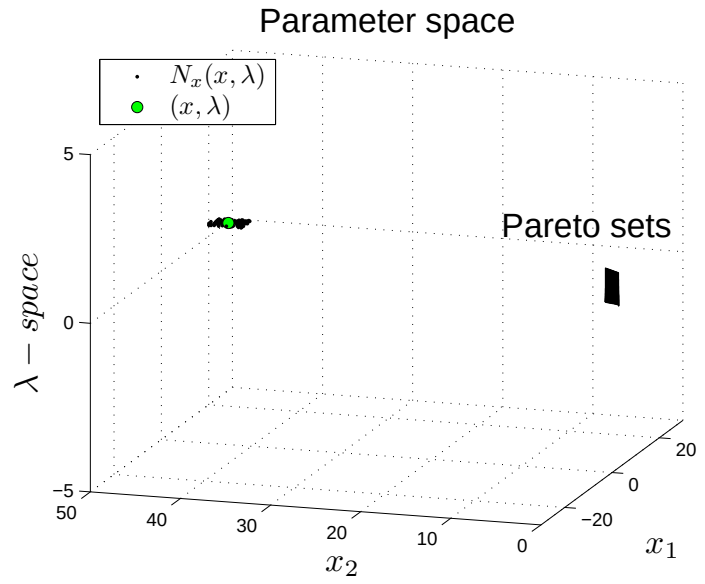


Figure 3.5: 100 randomly test points around a given point $(\mathbf{x}, \lambda)$ altering just $\mathbf{x}$ in parameter space.

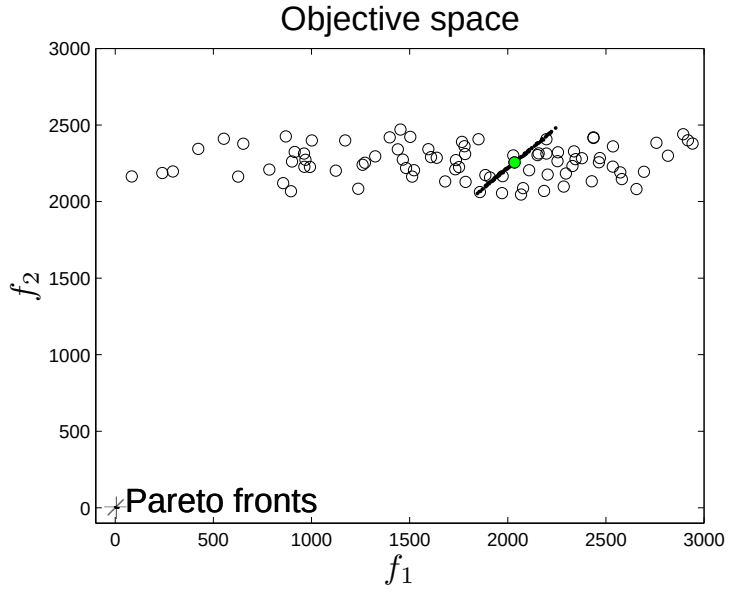


Figure 3.6: Comparison to see how the influence of λ affects the model.

3.4.2 Point near the Pareto set

Considering the other extreme situation, i.e., to consider a point $(\mathbf{x}, \lambda)$ that lies on the Pareto set. Thus, there exists a convex weight $\alpha \in \mathbb{R}^k$ such that $\sum_{i=1}^{k} \alpha_i \nabla_x f_i(\mathbf{x}, \lambda) = 0$ (which can be written as $J_x(\mathbf{x}, \lambda)^T \alpha = 0$). Further, we assume that the rank of $J_x(\mathbf{x}, \lambda)$ is $k - 1$. Then, it holds for any $\nu_f \in \mathbb{R}^n$:

$$\langle J_x(\mathbf{x}, \lambda)\nu_f, \alpha \rangle = \langle \nu_f, J_x(\mathbf{x}, \lambda)^T \alpha \rangle = 0. \tag{3.34}$$

Hence, we have that either $J_x(\mathbf{x}, \lambda)\nu_f = 0$ or a movement along ν_f leads to a movement along the family of Pareto fronts since $J_x(\mathbf{x}, \lambda)\nu_f \neq 0$ is orthogonal to α. The produced movement along the family is not along λ-space and it is regardless of the choice of ν_f. To see this movement, we take a Pareto point $(\mathbf{x}, \lambda) = (-0.38022, -0.75514, 0.21053)^T$ and sample points using the following parameters $r_x = 2.0$ for $\mathbf{x}$ and $r_\lambda = 0.0$ for λ. The result is showed in Figures 3.7 and 3.8. Just as a brief comment, if we choose a radio $r_\lambda \neq 0$, the result does not change, as it happened with the previous case when the point is far away, that is why, we do not show figures changing this value.

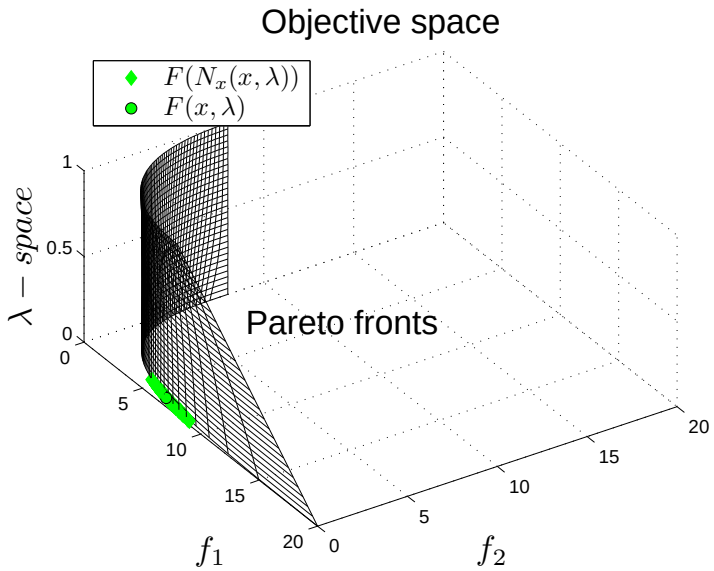


Figure 3.7: 100 randomly test points in the neighborhood of $(\mathbf{x}, \lambda)$ altering just $\mathbf{x}$ in objective space.

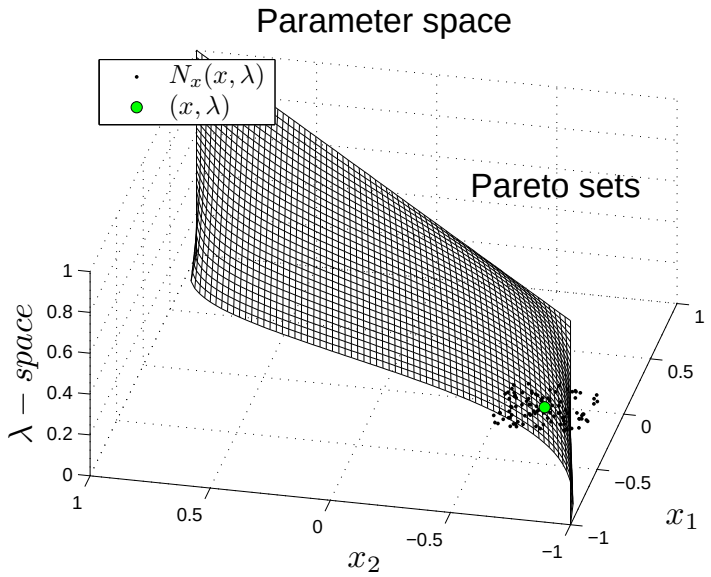


Figure 3.8: 100 randomly test points around a given point $(\mathbf{x}, \lambda)$ altering just $\mathbf{x}$ in parameter space.

3.5 The Gradient Free Directed Search Method for PMOPs

As we know a possible drawback of our new approach the λ-DS is the computation of the gradient. The latter can be solved by approximating the gradient which comes, however, with additional function evaluations that will increase the cost of the method. Nervertheless, this addition effort would come by free when use together an evolutionary approach since several solutions exist in the neighborhood of a given point $(\mathbf{x}, \lambda)$.

Assume a PMOP described as follows

$$F(x,\lambda) = \begin{pmatrix} g_1(\mathbf{x},\lambda) \\ \vdots \\ g_{k+l}(\mathbf{x},\lambda) \end{pmatrix}, \tag{3.35}$$

a candidate solution $(\mathbf{x}, \lambda) \in \mathbb{R}^{n+l}$, and r search directions $\nu_i \in \mathbb{R}^{n+l}$, $i = 1, \ldots, r$.

We can define the matrix $\mathcal{F}(\mathbf{x}, \lambda) \in \mathbb{R}^{(k+l)\times r}$ as follows:

$$\mathcal{F}(\mathbf{x}, \lambda) := (\langle \nabla g_i(\mathbf{x}, \lambda), \nu_j \rangle) \quad \begin{matrix} i = 1, \ldots, k+l \\ j = 1, \ldots, r \end{matrix}, \tag{3.36}$$

where every entry m_{ij} of $\mathcal{F}$ is defined by the directional derivative of objective g_i in direction ν_j, i.e., $m_{ij} = \langle \nabla g_i(\mathbf{x}, \lambda), \nu_j \rangle$. By using Equation (3.36) the following result is crucial for the discrete λ-DS version.

Proposition 5. *Let* $(\mathbf{x}, \lambda)$, $\nu_1, \ldots, \nu_r$ *are linear independent and* $\nu_i \in \mathbb{R}^{n+l}$, $w \in \mathbb{R}^r$, *and* $\nu := \sum_{i=1}^{r} w_i \nu_i$. *Then*

$$\mathcal{F}(\mathbf{x}, \lambda) w = J(\mathbf{x}, \lambda)\nu. \tag{3.37}$$

Proof. It is

$$\mathcal{F}(\mathbf{x}, \lambda) w = \begin{pmatrix} \langle \nabla g_1(\mathbf{x}, \lambda), \nu_1 \rangle & \ldots & \langle \nabla g_1(\mathbf{x}, \lambda), \nu_r \rangle \\ \vdots & \vdots & \vdots \\ \langle \nabla g_k + l(\mathbf{x}, \lambda), \nu_1 \rangle & \ldots & \langle \nabla g_k + l(\mathbf{x}, \lambda), \nu_r \rangle \end{pmatrix} \begin{pmatrix} w_1 \\ \vdots \\ w_r \end{pmatrix} \tag{3.38}$$

and

$$J(\mathbf{x}, \lambda)\nu = J(\mathbf{x}, \lambda)(\sum_{i=1}^{r} w_i \nu_i) = \sum_{i=1}^{r} w_i \begin{pmatrix} \nabla g_1(\mathbf{x}, \lambda)^T \\ \vdots \\ \nabla g_k + l(\mathbf{x}, \lambda)^T \end{pmatrix} \nu_i. \tag{3.39}$$

Hence, for the s-th component of both products it holds

$$(\mathcal{F}(\mathbf{x}, \lambda))_s = \sum_{i=1}^{r} w_i \langle \nabla g_s(\mathbf{x}, \lambda), \nu_i \rangle = (J(\mathbf{x}, \lambda)\nu)_s, \tag{3.40}$$

and the desired identity follows. □

Hence, the gradient free version of our approach λ-DS can be done by following the next steps: Given an initial point $(\mathbf{x}_0, \lambda_0)$ which has been selected for local search, as well as r test points $(\mathbf{x}_i, \lambda_i)$, $i = 1, \ldots, r$, we can approximate the entries of $\mathcal{F}$ by using the following equation:

$$m_{ij} = \langle \nabla g_i(\mathbf{x}, \lambda)\nu_j \rangle = \frac{g_i(\mathbf{x}_j, \lambda_j) - g_i(\mathbf{x}_0, \lambda_0)}{||(\mathbf{x}_j, \lambda_j) - (\mathbf{x}_0, \lambda_0)||_2}. \tag{3.41}$$

Thus, instead of solving Equation (3.4) to get the search direction ν, it can alternatively be computed by solving:

$$\mathcal{F}(\mathbf{x}, \lambda)w = d, \tag{3.42}$$

and setting

$$\nu := \sum_{i=1}^{r} w_i \nu_i. \tag{3.43}$$

Remark 4. *To compute a direction ν that solves Equation (3.42), it is only needed to have $r = k + l$ test points. Therefore, if $r > k + l$ test points are available in the neighborhood of $(\mathbf{x}_0, \lambda)$, this direction comes for free.*

3.6 Numerical Results

In this final section, we will present some numerical results and discussion related to the λ-DS continuation method using the example defined in Equation (2.14).

It is important to define firstly the approach that we are going to use to compare our proposed method, in this case by looking at the literature, a classical continuation method is defined in [3] where the zero set is traced using the KKT equations:

$$\hat{F}(\mathbf{x}, \alpha) = \begin{pmatrix} \sum_{i=1}^{k} \alpha_i \nabla f_i(\mathbf{x}) \\ \sum_{i=1}^{k} \alpha_i - 1 \end{pmatrix}, \tag{3.44}$$

obviously, we have to make a modification to adapt this classical continuation method to the current context, in this case, since the value of $\alpha \in \mathbb{R}^k$ is going to be fixed, to get the zero set, the following optimization problem has to be solved:

$$\tilde{F}(\mathbf{x}, \lambda) = \sum_{i=1}^{k} \alpha_i \nabla_x f_i(\mathbf{x}, \lambda) = 0. \tag{3.45}$$

Since to find predictor directions at a point $(\mathbf{x}, \lambda)$ implies to linearize the solution set in parameter space, second derivative information will be neccesary, since the linearization has to be realized by a QR-factorization of $J(\mathbf{x}_0, \lambda)$, what is going to become the computation of new points expensive for this method. A graphical representation of this method,in both parameter and objective space, can be seen in

Figure 3.9 and Figure 3.10. As we notice the continuation keeps the value of α to perform the search along λ-space.

Remark 5. *To make our comparison fair, we assume that for both approaches automatic differentiation (AD) will be used to compute the derivatives. By using AD as shown in [67], we can estimate* 5 *function evaluations, while for the computation of the Hessian matrix the value of function evaluations is set by*

$$k(4+6n), \tag{3.46}$$

where k is the number of objectives and n the number of variables.

Firstly, we compare the proposed continuation methods which differ in how to compute the corrector point. The first one uses the λ-DS to do the task and the second one solves Equation (3.45) to obtain it. To see the results, we present Table 3.1:

Information	**PCLDS1**	PCLDS2
# of $F(\mathbf{x}, \lambda)$	**17**	192
# of $J(\mathbf{x}, \lambda)$	**17**	12
# of $H(\mathbf{x}, \lambda)$	**0**	0
# of total $F(\mathbf{x}, \lambda)$	**102**	252

Table 3.1: Comparison of the two possibilities to perform a corrector point by using our continuation method.

In Table 3.1, we can observe that PCLDS1, the approach that uses λ-DS to get a corrector point, is cheaper than PCLDS2, the approach that solves Equation (3.29) to obtain a corrector point. The main reason of this is that λ-DS is able to reach quickly a boundary point what represents an advantage if it will be used into evolutionary algorithms. Nevertheless, speaking about accuracy and quality, if we look at Figure 3.11 and Figure 3.12 , we notice that the produced points by PCLDS1 are less accurate than the ones produced by PCLDS2. Another important detail about PCLDS2 is that the quality of the approximation is better, if we compare both approaches against the produced points by the classical continuation. So, we conclude that depending on the problem that we have at hand is the approach that we will use to obtain a corrector point.

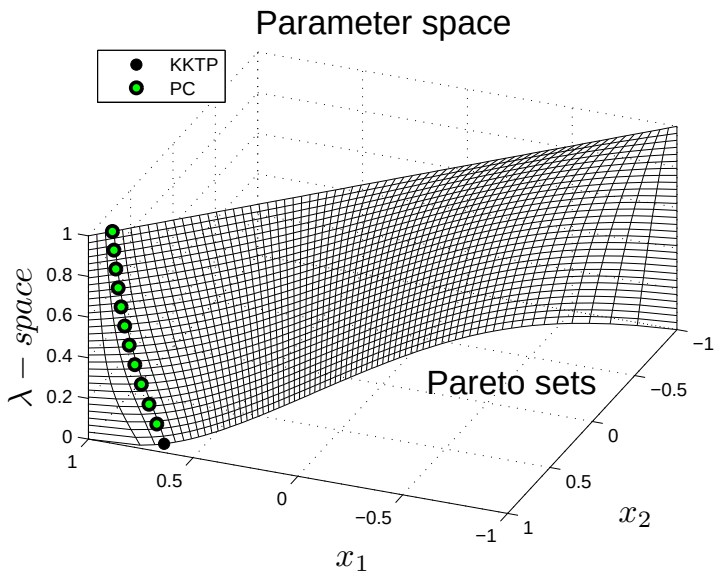


Figure 3.9: Continuation method using the classical approach. KKTP is the initial point, which is a KKT point and PC are the points over the curve.

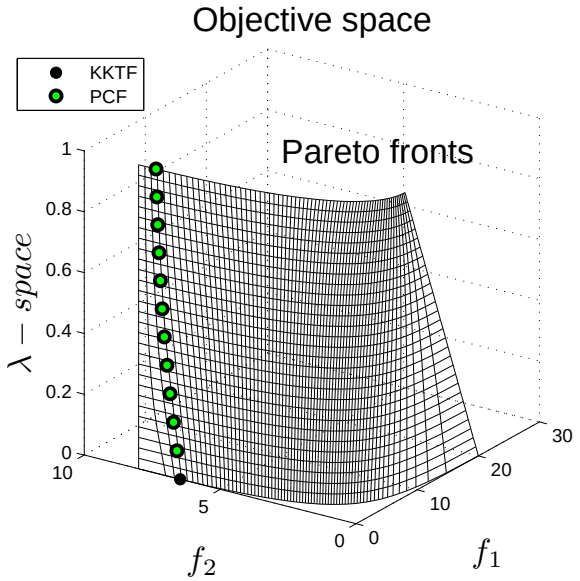


Figure 3.10: Continuation method using the classical approach. KKTF is the image of the initial point and PCF are the points over the curve.

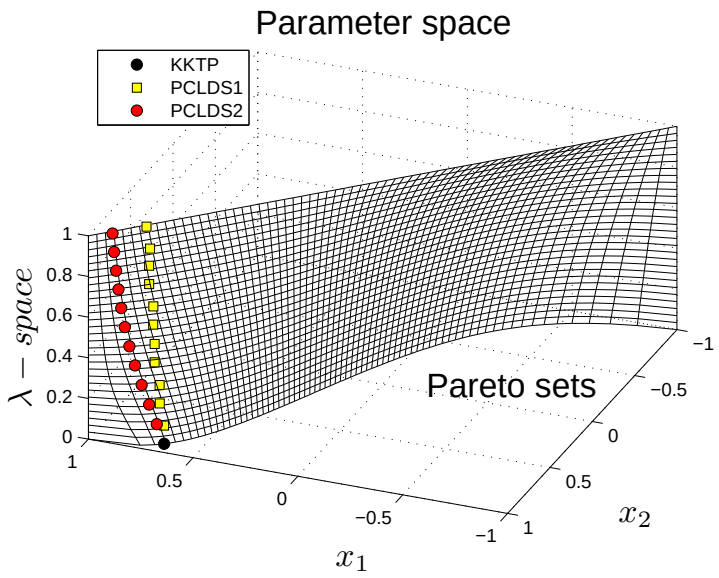


Figure 3.11: λ-DS continuation method using both corrector approaches in parameter space.

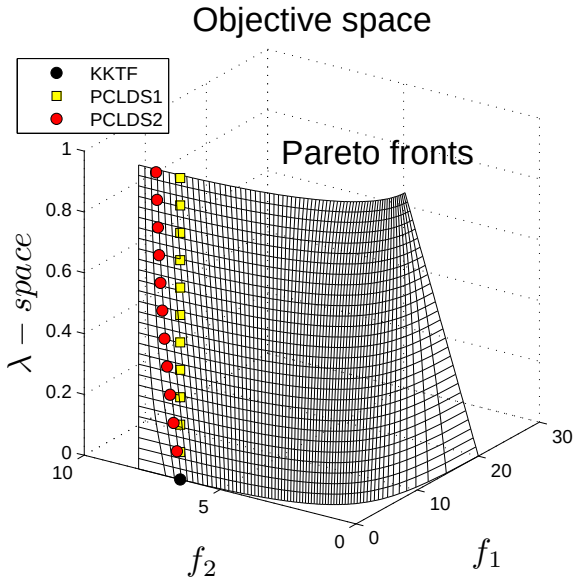


Figure 3.12: λ-DS continuation method using both corrector approaches in objective space.

Now, for comparison purposes, we take PCLDS2 to compare it against the classical continuation method using KKT Equations, due to the approximation quality is similar, what we do not obtain by using PCLDS1. Hence, in Table 3.2, we present the results of each approach by counting the number of total function evaluations.

According Table 3.2, it is important to mention that our approach comes with less cost than the classical one, since we obtain the same result, but λ-DS does not need any second derivative information.

	KKT Equations	**PCLDS2**
# of $F(\mathbf{x}, \lambda)$	87	**192**
# of $J(\mathbf{x}, \lambda)$	216	**12**
# of $H(\mathbf{x}, \lambda)$	87	**0**
# of total $F(\mathbf{x}, \lambda)$	3951	**252**

Table 3.2: Comparison between our novel approach and the classical continuation method.

4 | Directed Search for Hypervolume based MOEAs

In the previous chapter, we presented theory, results and discussion for our extension of the Directed Search method to tackle parameter dependent multi-objective optimization problems. Nevertheless, another important objective of this thesis is still missing to be covered namely how to support IBEAs by using the DS. In this chapter, we are going to present the design of a novel local search strategy using hypervolume approximations in the context of MOPs. Further, we will integrate such local search strategies into SMS-EMOA to improve its performance.

As we know, when local search strategies are used together with MOEAs, they explore the solution set in a fine-grained way in order to improve the solution at hand. The DS has been introduced in Chapter 2, it exploits gradient information to perform a search toward and along the Pareto front, however, as a local searcher it has not been used to improve the value of a given indicator. In literature, one of the most well-accepted and used indicators is the dominated hypervolume. In the following pages, we are going to present a new local search strategy which aims to maximize the value of the produced hypervolume in a clever way. Such strategy is called the hypervolume based Directed Search (HVDS). The algorithm will be firstly introduced as a standalone procedure and will then be integrated into an IBEA to obtain a new memetic evolutionary algorithm.

4.1 The Hypervolume based on Directed Search

The Hypervolume based Directed Search (HVDS) is introduced as a new local search strategy for bi-objective optimization problems. This strategy proposes a division of the objective space into three different regions, to know the position of an element from a given approximation. Our approach is able to perform movements toward the Pareto front and along the solution set. This movement is according to the position of the selected element within the three regions. The HVDS decides automatically in which region a point is, being the only condition to be used to have gradient information available.

4.1.1 Division of the objective space

The HVDS uses a division of the objective space to locate a point $F(\mathbf{x})$, in order to perform the necessary movement in every region of the search. By some observations, we can divide the objective space into the following regions:

- **Region** I The objective vector $F(\mathbf{x})$ is 'far away' from the Pareto (denoted by $F(\mathbf{x}) \in I$).

- **Region** II $F(\mathbf{x})$ is now 'in between', i.e., not far away nor near to the Pareto front.

- **Region** III When $F(\mathbf{x})$ is 'near' or already on the Pareto front.

In Figure 4.1, we can observe how these regions would look like (including the reference point $R = (r_1, r_2)^T \in \mathbb{R}^2$ used to measure the hypervolume), leading to the following desired movements:

1. When a vector $F(\mathbf{x})$ is in Region I a greedy search toward the rough location of the Pareto front is desired.

2. If a vector $F(x)$ is not close nor far away and it dominates a given reference point R. A descent direction has to be selected such that the movement in that direction maximizes the hypervolume.

3. Given a vector $F(x)$ in Region III, a movement toward the Pareto front will lead to non-significant improvements of the dominated hypervolume. Hence, a search along the Pareto front is desired, in order to maximize the contribution to the hypervolume of that point.

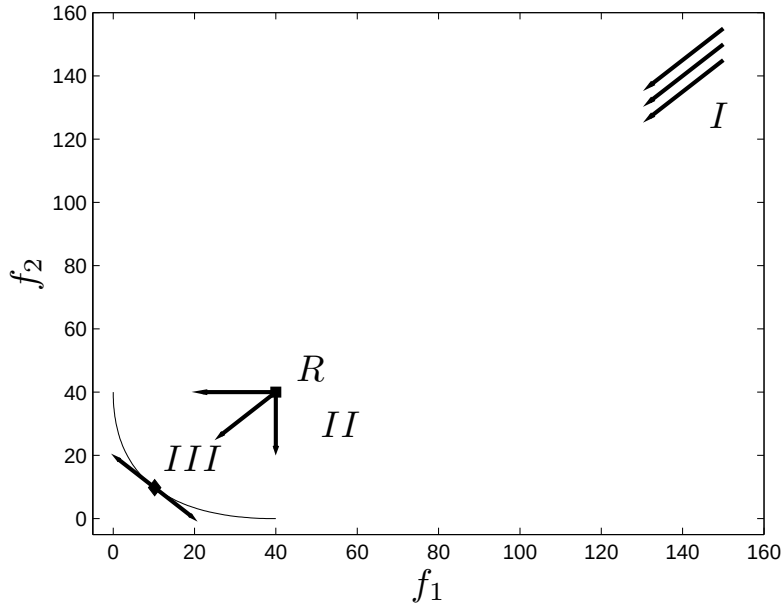


Figure 4.1: Division of the objective space into three different regions.

To decide in which of the three regions an objective vector $F(\mathbf{x})$ is, we use the properties of the descent cone of a MOP. Those properties were defined in the work of Brown and Smith [68]. Here, the authors consider the size of the descent at the initial and final stages of an optimization process to state the following:

- When a point $\mathbf{x}$ is far from the local optima, the objectives gradients are aligned and the descent cone is almost equal to the half-spaces associated with each objective.
- When a point $\mathbf{x}$ is close to the Pareto set, the individual gradients are almost contradictory, what provokes that the size of the descent cone is extremely narrow.

A graphical illustration of the above defined properties can be found in Figure 4.2 (this figure was taken from [68]). In order to take advantage of this properties

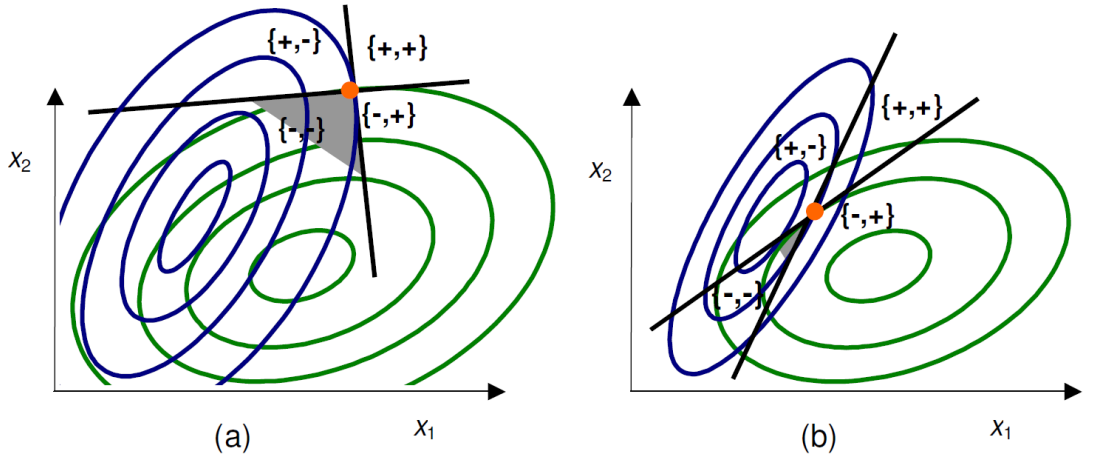


Figure 4.2: A descent cone is depicted for a 2 parameter and 2 objective MOP. The process is during initial (a) and final (b) stages of convergence.

to assign an objective vector $F(x)$ to one region, the angle between gradients will be considered.

Let

$$g_i := \nabla f_i(\mathbf{x}), \quad i = 1, 2, \tag{4.1}$$

and $g_1, g_2 \neq 0$, then the angle between g_1 and g_2 is defined by

$$\cos\alpha = \frac{g_1^T g_2}{||g_1||\ ||g_2||} \ \in \ [-1, 1]. \tag{4.2}$$

In other words, if $cos\ \alpha = 1$, both gradients point into the same direction ($\downarrow\downarrow$) which happens, roughly speaking, if $\mathbf{x}$ is infinitely far from the Pareto set. If $cos\ \alpha = 0$, the gradients are orthogonal to each other ($\leftarrow\downarrow$). Finally, when $cos\ \alpha = -1$, the gradients point into opposite directions ($\downarrow\uparrow$) which happens if $\mathbf{x}$ is on the Pareto set (i.e., zero distance).

Since $\cos\alpha \in [-1, 1]$, our three defined regions can be numerically detected, by setting two values $a, b \in (-1, 1)$ with $b < a$. Leading to the following:

$$\begin{aligned} F(x) \in I &\ : \Leftrightarrow \cos\ \alpha\ \geq\ a, \\ F(x) \in II &\ : \Leftrightarrow \cos\ \alpha\ \in\ (b, a), \\ F(x) \in III &\ : \Leftrightarrow \cos\ \alpha\ \leq\ b, \end{aligned}$$

In order to study which are the proper values for a and b, we introduce the following MOP:

$$f_1, f_2 : \mathbb{R}^n \to \mathbb{R}$$
$$f_i(\mathbf{x}) = ||\mathbf{x} - w_i||_2^2 \tag{4.3}$$

where

$$w_1 = (1, 1, 1, 1, ...) \in \mathbb{R}^n$$
$$w_2 = (1, -1, 1, -1, ...) \in \mathbb{R}^n$$

In Figures 4.3 and 4.4 a color-map of the previous problem is presented using $n = 2$ and $\mathbf{x} \in [-5, 5]^2$. We can clearly see the locations of both the Pareto set and Pareto front. Since this color-map of a cone size of a MOP into a certain range behaves similar for other MOPs, we are in the position to define a and b in a general manner.

To select a, we start with the fact that all what is dominated for a given reference point R does not contributes to the hypervolume, therefore, it should be in Region I. The latter is because shooting a point $\mathbf{x}$ in a greedy direction will make that $F(\mathbf{x})$ dominates R, in order to be considered when compute the hypervolume. In this case, we recommend to select $a \in [0.4, 0.6]$. For the remainder of this work, we set $a = 0.5$ to perform our experiments.

The value of b can be chosen by considering how an angle is approaching to -1 because the individual gradients are almost contradictory. So, if we select b very close to -1, we guarantee that only points which are in Region III will be considered to perform a movement along the front. The recommended interval is $b \in (-1, -0.9]$, being our choice $b = -0.9$.

Finally, we can observe the result of the objective space division for our example in Figure 4.5. As final observation, we can choose also a according to the selected reference point R for a given problem, this leads to have a better division of the objective space.

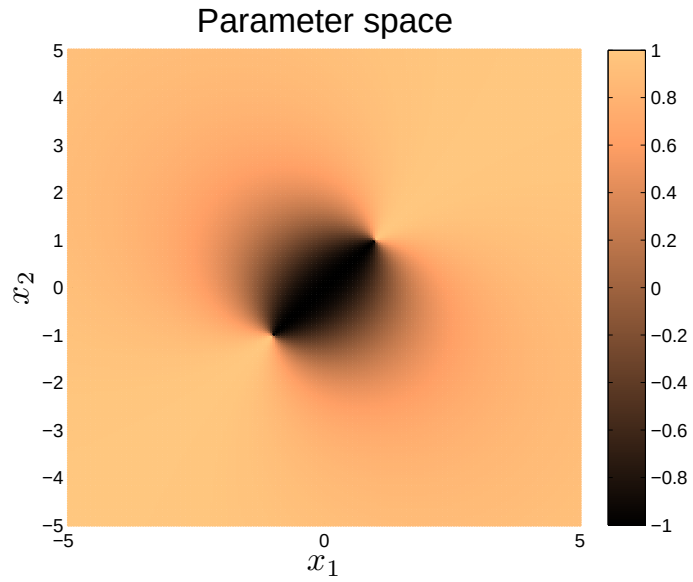


Figure 4.3: Color-map of the parameter space, the dark blue region represents the Pareto set.

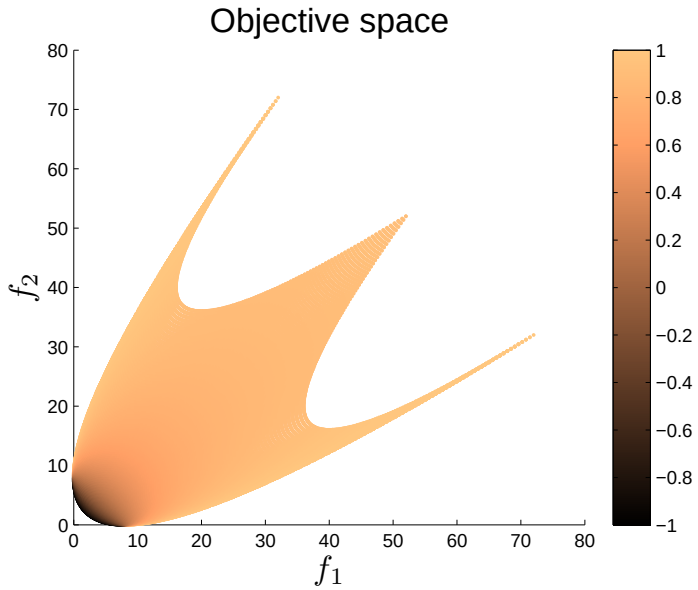


Figure 4.4: Color-map of the objective space, the dark blue region represents the Pareto front.

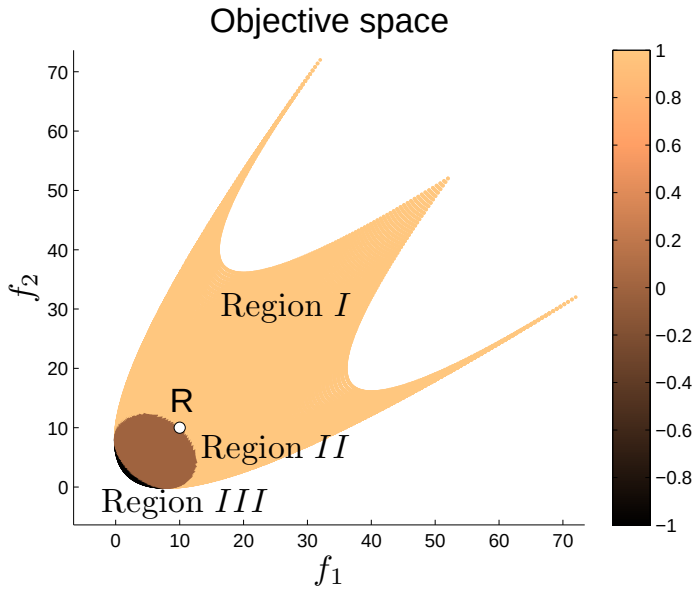


Figure 4.5: Region color divisions together with a reference point R.

4.1.2 One Element Archives

We will consider in the following consider for sake of simplicity archives consisting of one element. Subsequent considerations will show that the consideration of general archives can be put back in many cases to such one element archives.

Given an archive $A = \{\mathbf{x}\}$ with $\mathbf{x} \in S$. Furthermore, a reference point $R = (r_1, r_2)^T \in \mathbb{R}^2$ for hypervolume calculations. The three local search regions are described as follows:

Local search in Region I For this region large improvements are required, since points there do not contribute to the hypervolume. In [37] was proposed to use Equation (2.27) as a greedy descent direction for bi-objective optimization problems. This descent direction performs large improvements in objective space, because when $\mathbf{x}$ is far away both gradients point almost in the same direction. Due to the previous fact, for Region I, we propose to use Lara's direction coupled with an Armijo-like step size (as in [28]) for a point located in this region. Thereby, a new point will be given by $\mathbf{x}_{new} = \mathbf{x} + t\nu$, using Equation (2.27) to compute ν.

Local search in Region II Given a point $\mathbf{x}$ such that $F(\mathbf{x})$ is in Region II. The desired improvement can be done by finding a search direction $d_{II} <_p 0$ such that a movement in that direction maximizes the hypervolume according to a given reference point R. Since the DS allows to move in any direction in objective space, we propose to use it in the following way:

Being y_{new} the image of a new point x_{new} obtained using DS, it is given then by

$$y_{new} = F(\mathbf{x}) + td_{II}, \tag{4.4}$$

where $t \in \mathbb{R}$ is a given (fixed) step size and d_{II} has to be chosen such that it solves the following two-dimensional problem

$$\max_{d\in\mathbb{R}^2} H(d) = (r_1 - f_1(\mathbf{x}) - td_1) \times (r_2 - f_2(\mathbf{x}) - td_2), \tag{4.5}$$
$$\text{s.t.} ||d||_2^2 = 1$$

If we replace the Euclidean norm by the infinity norm in the constraint of Equation (4.5) (which drops the assumption that the movement is done with an equal step in objective space) a straightforward computation shows that

$$d_{II,\infty} = F(\mathbf{x}) - R, \tag{4.6}$$

solves the modified problem. It is important to say that by using $d_{II,\infty}$ (which is easier to calculate) coupled with the DS yields no difference in the performance of the algorithm. Figure 4.6 shows a graphical representation to select d_{II}.

Local search in Region III In the case that the image $F(\mathbf{x})$ of a given point be in **Region** III, a movement along the Pareto front is sought, in order to increase the single point contribution to the hypervolume. To achieve this objective, we propose to linearize the Pareto front at $F(\mathbf{x})$ and compute the optimal step size along d_{III} to produce the maximum hypervolume contribution. That direction d_{III} can be computed in the following way:

Let $\mathbf{x}$ be a Karush Kuhn Tucker (KKT) point. It is known that the corresponding weight vector α s.t. $\sum_{i=1}^{2} \alpha_i \nabla f_i(\mathbf{x}) = 0$ is orthogonal to the linearized Pareto front at $F(\mathbf{x})$, and α solves the following quadratic optimization problem (see [34]):

$$\min_{\alpha\in\mathbb{R}^2} \left\{ \|\alpha_1 \nabla f_1(x) + \alpha_2 \nabla f_2(\mathbf{x})\|_2^2 : \ \alpha_i \geq 0, \ i = 1, 2, \ \alpha_1 + \alpha_2 = 1 \right\} \tag{4.7}$$

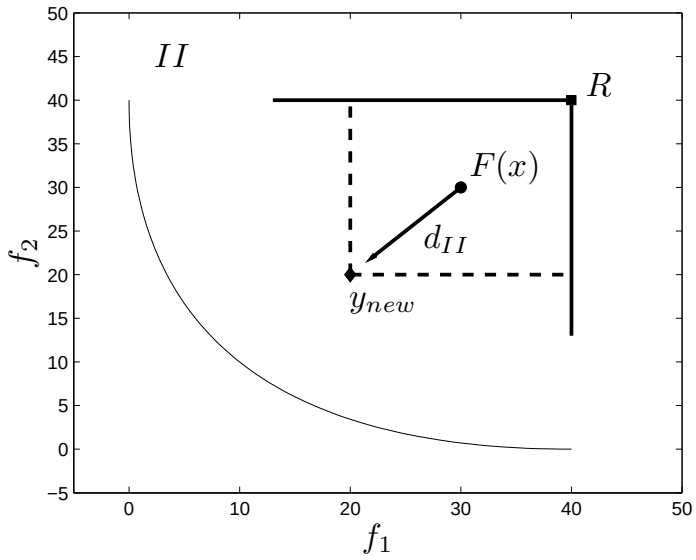


Figure 4.6: Direction for a point in Region II to perform the local search.

Hence, for a bi-objective problem, we can compute a solution $\tilde{\alpha}$ by solving Equation (4.7) and set

$$d_{III} = \begin{pmatrix} -\tilde{\alpha}_1 \\ \tilde{\alpha}_2 \end{pmatrix} \tag{4.8}$$

as our new direction to perform the movement along the Pareto front using the DS.

By having d_{III}, it leads to the following one-dimensional problem to maximize the hypervolume, where the step size t is now the unknown:

$$\max_{t\in\mathbb{R}} H(t) = (r_1 - f_1(\mathbf{x}) - td_1) \times (r_2 - f_2(\mathbf{x}) - td_2), \tag{4.9}$$

this problem has an analytic solution when the values of the weight vector α has no entries equal to zero. The previous ideas draw the result below.

Proposition 6. *Let $\alpha >_p 0$, then the global maximizer of Equation (4.9) is given by*

$$t^* = \frac{d_1 r_2 + d_2 r_1 - d1 f_2(x) - d_2 f_1(x)}{2 d_1 d_2}. \tag{4.10}$$

Proof. If $\alpha_p >_p 0$, then it follows by Equation (4.8) that $d_1, d_2 \neq 0$. The first derivative of H is given by

$$H'(t) = 2td_1d_2 + d_2 f_1(\mathbf{x}) - r_1 d_2 + d_1 f_2(\mathbf{x}) - d_1 r_2. \tag{P6.1}$$

Setting this to zero leads to

$$t^* = \frac{d_1 r_2 + d_2 r_1 - d1 f_2(\mathbf{x}) - d_2 f_1(\mathbf{x})}{2 d_1 d_2}. \tag{P6.2}$$

Further, the second derivative at t^* is given by

$$H''(t^*) = 2 d_1 d_2 < 0. \tag{P6.3}$$

The negativity holds since $\alpha >_p 0$ and by construction of d_{III}, and the claim follows. □

We stress that the above solution holds for the linearized problem which is of course a simplification of the problem at hand. We have observed that the step size t^* leads to satisfying results in particular if (i) the Pareto front is almost linear, and (ii) if the reference point R and the current objective vector $F(\mathbf{x})$ are not too far away from each other. For practical implementations, it is advisable to define a maximal step size t_{max} to bound the search. Also note that the step size t^* is defined for a search in objective space while the new iteration $\mathbf{x}_{new} = \mathbf{x} + t_x \nu$ is obtained via a line search in parameter space. For this, we follow the suggestion made in [69] to make the match $t_x = t^*$ that works particularly well for small values of t^*. Finally, we note that the above consideration is made for KKT points. However, these computations work also well if the candidate solution $\mathbf{x}$ is near to the Pareto set. In particular, d_{III} points along the Pareto front as we show in Figure 4.7.

Algorithm 4 summarizes the above discussion and presents the HVDS as standalone algorithm.

Algorithm 4 HVDS as standalone algorithm for one element archives

Require: An starting point $\mathbf{x}_0$, values a and b for region assignment and a reference point $R \in \mathbb{R}^2$.

Ensure: A new solution $\mathbf{x}_i$.

$i := 0$.
repeat
  Compute the angle between gradients θ as in Equation (4.2).
  if $\theta > a$ **then**
    Compute ν_I as in Equation (2.27) $\{F(\mathbf{x}_i) \in I\}$
    Compute $t_I \in \mathbb{R}_+$.
    Set $\mathbf{x}_{i+1} = \mathbf{x}_i + t_I \nu_I$.
  else if $\theta \in (b, a)$ **then**
    Set $d_{II} = F(\mathbf{x}_i) - R$ $\{F(\mathbf{x}_i) \in II\}$.
    Get $\nu_{II} = J(\mathbf{x}_i)^+ d_{II}$ as in Equation (2.37).
    Compute $t_{II} \in \mathbb{R}_+$.
    Set $\mathbf{x}_{i+1} = \mathbf{x}_i + t_{II} \nu_{II}$.
  else
    Get the convex weight α according to Equation (4.7). $\{F(x_i) \in III\}$
    Set $d_{III} = (-\alpha[2], \alpha[1])^T$ as in Equation (4.8).
    Get $\nu_{III} = J(\mathbf{x}_i)^+ d_{III}$ as in Equation (2.37).
    Compute t_{III} as in Equation (4.10).
    Set $\mathbf{x}_{i+1} = \mathbf{x}_i + t_{III} \nu_{III}$.
  end if
  $i := i + 1$.
until $t_{III} = 0$ or a maximum number of iterations is reached

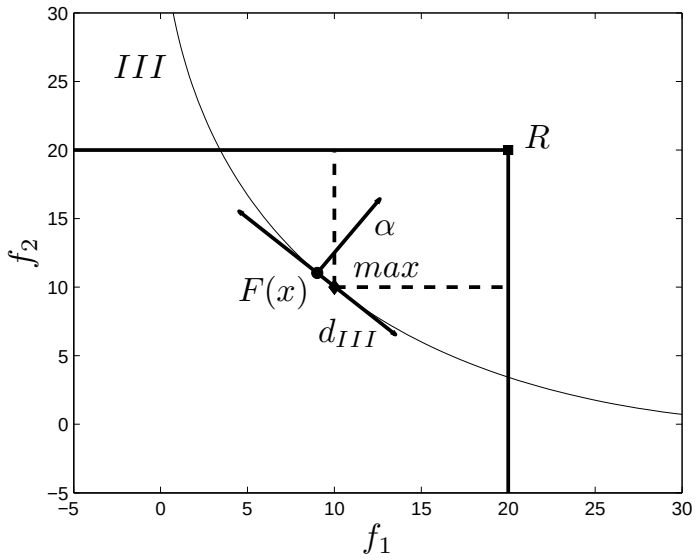


Figure 4.7: Direction for a point in Region III to move along the Pareto front.

4.1.3 General Archives

Now we consider the general case where the archive contains l elements, i.e., $A = \{\mathbf{x}_1, \ldots, \mathbf{x}_l\}$. The 'optimal' search direction for a given point $\mathbf{x} \in A$ depends in some cases on the location of the other elements of A. We face now the following problem, if we increase the contribution of one point $\mathbf{x} \in A$, another point in A could reduce its contribution, what it is not desired for us. However, we can overcome this problem by reducing all cases to the one element case with appropriate adjustments to the algorithm.

In the following, we consider the local search in all three distance regions for an archive A with l elements.

Local search in Region I If a point $\mathbf{x} \in A$ that is chosen for local search is far away from the Pareto front, a large movement is desired regardless of the location of the other elements of A. Hence, we propose to proceed as for the one element case using Equation(2.27) to get a greedy direction.

Local search in Region II Since we consider two-objective problems, the images in objective space of A can be sorted by one of the objective values which we assume in the following. Let $\mathbf{x}_i \in A$ be given for local search a direction d_{II} for the DS can be obtained using the following observation. Figure 4.8 shows a scenario that suggests that the hypervolume contribution of $\mathbf{x}_{new}$ will be obtained via a modification of $\mathbf{x}_i$. Such modification will be restricted to the region between $F(\mathbf{x}_{i-1})$ and $F(\mathbf{x}_{i-1})$. Hence, for $i \in \{2, \ldots, l-1\}$ we propose to choose a new reference point given by

$$R_{F(\mathbf{x}_i)} = \begin{pmatrix} f_1(\mathbf{x}_{i+1}) \\ f_1(\mathbf{x}_{i-1}) \end{pmatrix}, \tag{4.11}$$

and to proceed analog to the one element case using the direction

$$d_{II,\mathbf{x}_i} = F(\mathbf{x}_i) - R_{F(\mathbf{x}_i)}. \tag{4.12}$$

For the extreme points (i.e., $i \in \{1, l\}$) we proceed again with $R_{F(\mathbf{x}_i)} = F(\mathbf{x}_i) - R$.

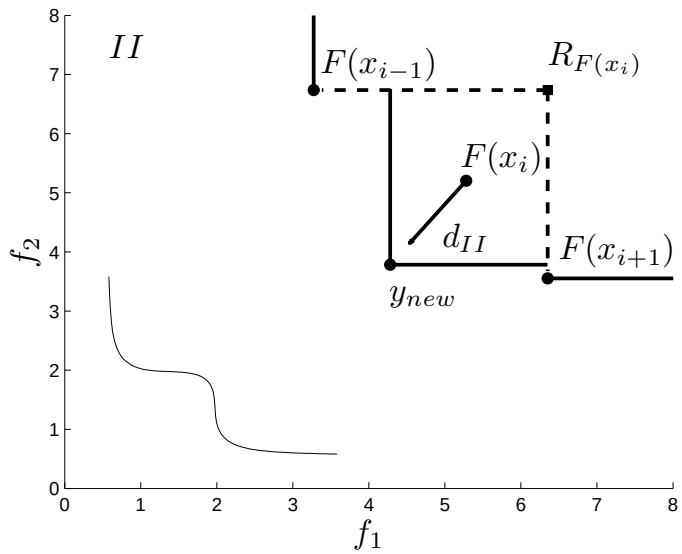


Figure 4.8: Local search in Region II for multiple archive entries.

Local search in Region III Analog to the above discussion we can proceed with points in the third distance region. To be more precise, we propose to use the reference point $R_{F(\mathbf{x}_i)}$ for intermediate points (i.e., $i \in \{2, \ldots, l-1\}$) and the original point R for the extreme archive entries to perform a movement along the solution set.

4.2 Integrating HVDS into SMS-EMOA

Here we present two attempts to integrate the local search mechanism HVDS into an EMOA in order to obtain a fast and reliable algorithm to obtain hypervolume approximations of a given MOP. We have chosen to take the state-of-the-art algorithm SMS-EMOA ([52]), however, we stress that HVDS can in principle be hybridized with any other hypervolume based EMOA.

4.2.1 HVDS at the initial stage

As first attempt, we propose to integrate the new local search mechanism as follows: After the update of the archive in iteration step i, m_i elements of the population P_i are chosen for local improvement via HVDS, X_{LS} will represent the set of the chosen elements. Since it is assumed that HVDS actually improves the hypervolume value of a given element, no consideration of the hypervolume contributions is necessary (which is a time-consuming task), but the new iterates replaces the initial point. This first version of the algorithm only considers to update the selected points at the beginning, in other words m elements are chosen after initializing the population P and then we replace them after applying HVDS. Algorithm 5 shows the pseudo-code of this new hybrid SMS-EMOA-HVDS.

A possible drawback of this first algorithm would be when a multi-modal problem be given, since we spend all the local search effort at the initial stage , so there is a high probability to get stuck in a local minima.

4.2.2 HVDS as operator

Due to the possible drawback to apply the HVDS only at the beginning, we introduce a new algorithm to overcome the problem. One of the properties of SMS-EMOA is that it only produces one offspring at each generation. Most times, this offspring replaces the element with the worst hypervolume contribution. Hence, by using that observation, we are in the position to take this element in order to apply the HVDS. It is important to mention that we have to add one extra parameter, in this case, the probability for the application of HVDS. For this new version the HVDS will act as

an operator with a certain probability to be applied. The following algorithm only includes the stage to produce an offspring into the SMS-EMOA using HVDS.

Algorithm 5 SMS-EMOA-HVDS

Require: A given MOP to solve and a reference point $R \in \mathbb{R}^2$ to compute the hypervolume.

Ensure: An approximation of the Pareto front.

1: Initialize a population $P \subset Q$ with μ elements at random.
2: Choose the set $X_{LS} \subset P$ with $|X_{LS}| = m$.
3: Sort elements of X_{LS} by their f_1 value.
4: **for all** $i = 1, \ldots, m$ **do**
5: $\quad \mathbf{x}_{i,0} = i$th element of X_{LS}.
6: $\quad \tilde{\mathbf{x}}_i$ =HVDS($\mathbf{x}_{i,0}$, a, b, R).
7: $\quad P := P \cup \{\tilde{\mathbf{x}}_i\} \setminus \{\mathbf{x}_{i,0}\}$.
8: **end for**
9: **repeat**
10: $\quad$ Generate offspring $\mathbf{x} \in Q$ from P by variation.
11: $\quad P := P \cup \{\mathbf{x}\}$.
12: $\quad$ Build ranking $G_1, \ldots, G_h$ from P.
13: $\quad$ Compute the hypervolume contribution for each $\mathbf{x} \in G_h$.
14: $\quad$ Denote by $\mathbf{x}^*$ the element with the least hypervolume contribution
15: $\quad P{:=}P \setminus \{\mathbf{x}^*\}$
16: **until** stopping criterion fulfilled
17: **return** P

Algorithm 6 HVDS as operator into SMS-EMOA

Require: A certain probability pr.

Ensure: A tunned new element $\tilde{\mathbf{x}}$.

1: Generate offspring $\mathbf{x} \in Q$ from P by variation.
2: **if** $rnd() <= pr$ **then**
3: $\quad \tilde{\mathbf{x}}$ =HVDS($\mathbf{x}$, a, b, R).
4: **else**
5: $\quad \tilde{\mathbf{x}} = \mathbf{x}$.
6: **end if**
7: $P := P \cup \{\tilde{\mathbf{x}}\}$.

4.3 Numerical Results

In the following, we present results of the HVDS. Here the HVDS is tested and compared as a standalone algorithm using both versions (for one element and for general ones). Then, we present results when HVDS is integrated into SMS-EMOA using the two proposed approaches.

4.3.1 HVDS as Standalone Algorithm

First we test the ability of the HVDS as standalone algorithm. For this, we will use the problem stated in Equation (4.3) and another model stated below, which are uni-modal problems.

$$f_1, f_2 : \mathbb{R}^2 \to \mathbb{R}$$
$$\begin{aligned} f_1(x) &= \frac{1}{2}(\sqrt{1+(x_1+x_2)^2} + \sqrt{1+(x_1-x_2)^2} + x_1 - x_2) + \lambda \cdot e^{-(x_1-x_2)^2}, \\ f_2(x) &= \frac{1}{2}(\sqrt{1+(x_1+x_2)^2} + \sqrt{1+(x_1-x_2)^2} - x_1 + x_2) + \lambda \cdot e^{-(x_1-x_2)^2}, \end{aligned} \tag{4.13}$$

where $\lambda = 0.85$. MOP (4.3) ([70], denoted by 'Convex') has a convex Pareto front, and the front of MOP (4.13) ([14], 'Dent') is convex-concave (see Figure 4.9).

First, we test the HVDS for one element archives. For the sake of a comparison, we define a simple hill climber as follows: For a given point $\mathbf{x}$, a further candidate solution $\mathbf{y}$ is taken from a neighborhood of $\mathbf{x}$. As next iterate, the solution with the highest

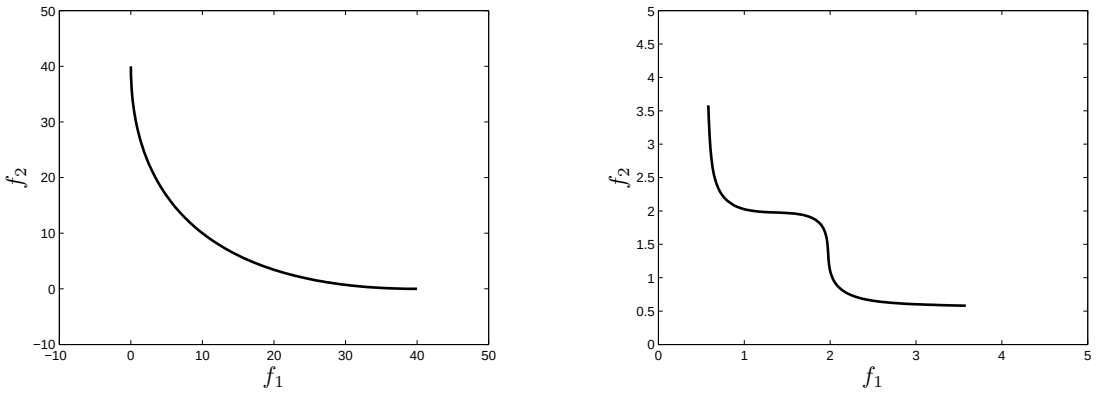


Figure 4.9: Pareto fronts of MOP (4.3) (left) and MOP (4.13) (right).

hypervolume value is taken and the search is continued in the same manner. We have chosen this strategy since it relates to a stochastic local search procedure within hypervolume-based MOEAs. Figures 4.11, 4.12, 4.13, and 4.14 show exemplary runs for both methods on each problem. Figure 4.10 shows the hypervolume against the number of function evaluations for both problems and methods. Here we count five function evaluations for the cost of one gradient evaluation which would be the case when using automatic differentiation [67]. In both cases, HVDS is able to get higher hypervolume values in the early stage of the algorithm. Also, we can see the smart movement over all the objective space while the hill climber search without direction. For Dent, the algorithm is even able to terminate after 130 function evaluations at the optimal hypervolume value.

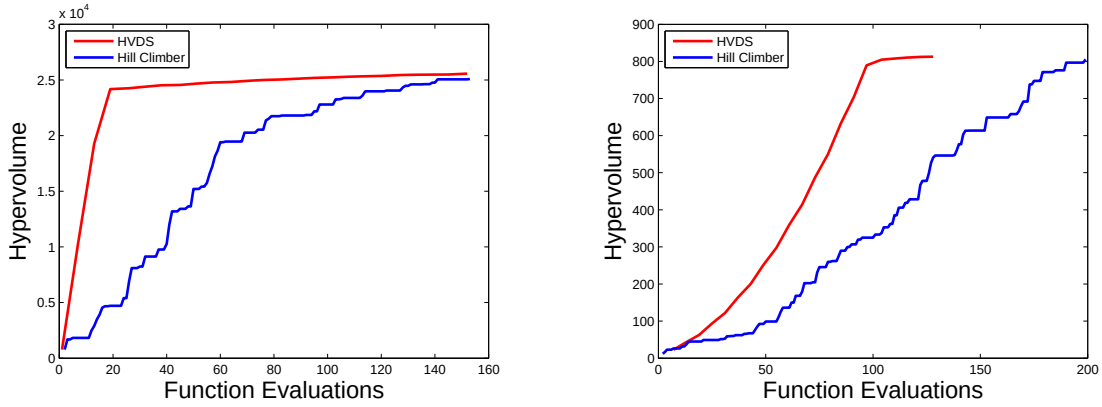


Figure 4.10: Hypervolume comparison of the HVDS against the hill climber on Convex (left) and Dent (right). The results are averaged over 20 test runs.

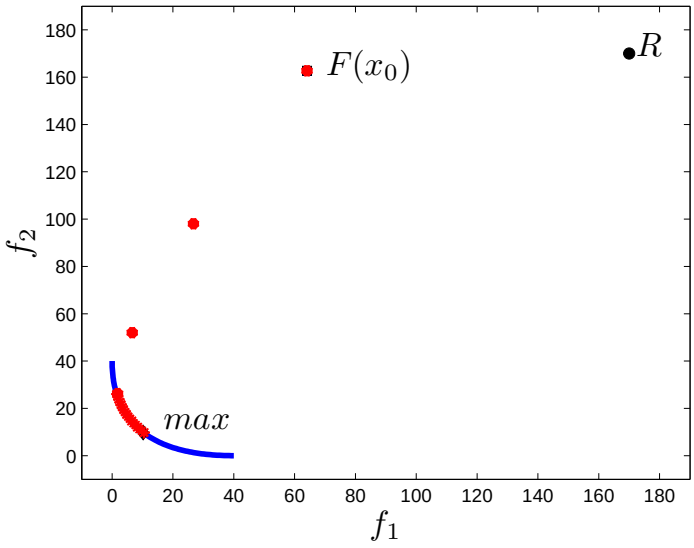


Figure 4.11: Result of the HVDS on Convex.

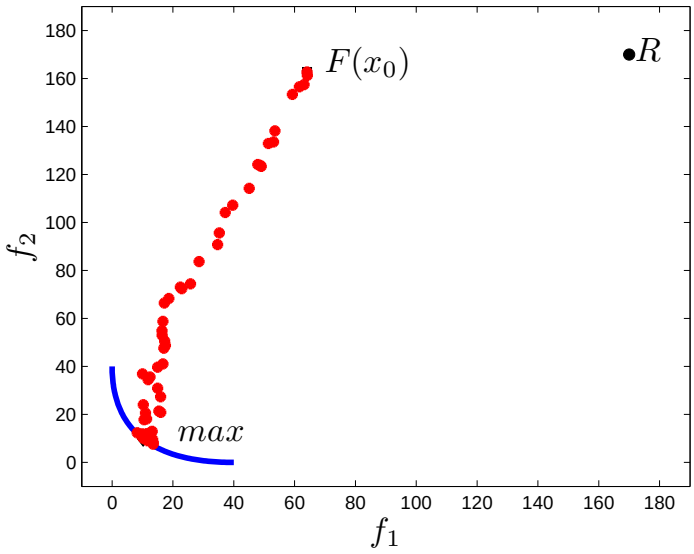


Figure 4.12: Result of the hypervolume hill climber on Convex.

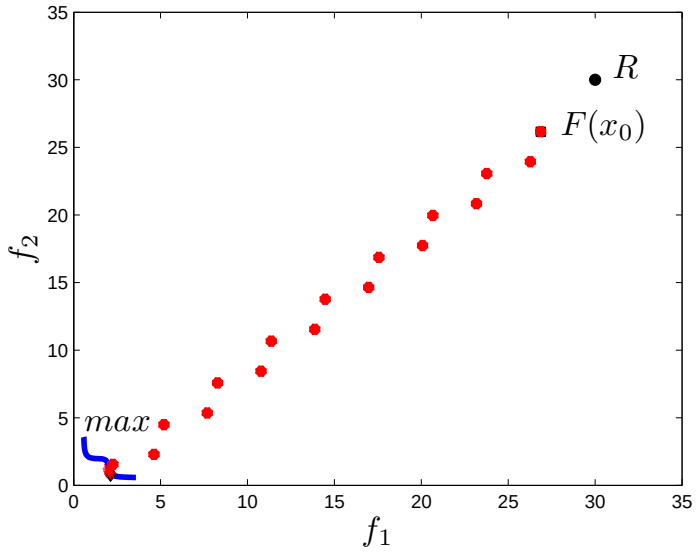


Figure 4.13: Result of the HVDS on Dent.

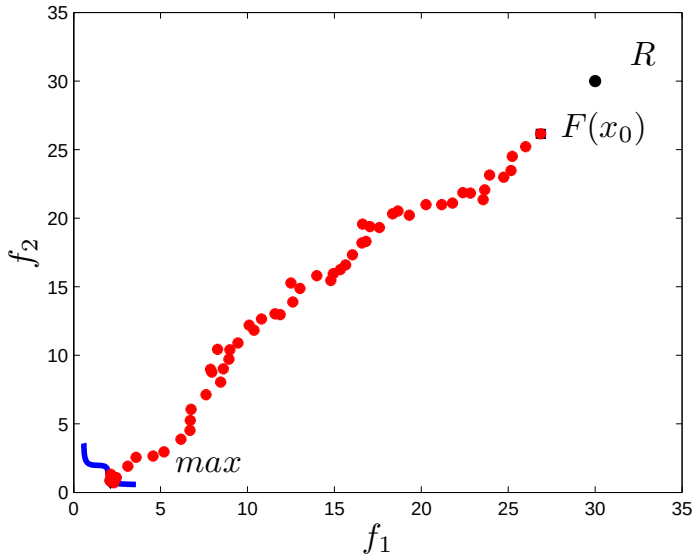


Figure 4.14: Result of the hypervolume hill climber on Dent.

Next, we make a first attempt to investigate the ability of the HVDS within set based search, that is to consider general archives. For this, we have made the following adaption of the standalone HVDS as presented in Algorithm 4: Instead of one starting point $\mathbf{x}_0$ we choose an initial population $X = \{\mathbf{x}_0^{(1)}, \ldots, \mathbf{x}_0^{(5)}\}$ consisting of five elements random elements. The iteration step is then performed individually for all elements (i.e., $\mathbf{x}_{i+1}^{(j)} = \mathbf{x}_i^{(j)} + t\nu$ as described in Algorithm 4) using the choice of the reference point as proposed in Equation (4.11) for intermediate points and the normal reference point R for extreme points.

Figures 4.15 and 4.16 show some numerical results of the application of the HVDS over a given set of points. Table 4.1 presents a comparison of HVDS against SMS-EMOA with population size $\mu = 5$. It is shown that HVDS is able to get better hypervolume approximations. However, it has to be noted that for problem Dent none of the methods converges toward the optimal archive but the values get stuck on the value shown in Table 4.1 even for a higher budget of function evaluations. This might be due to the fact that only *one* point is iterated at each step.

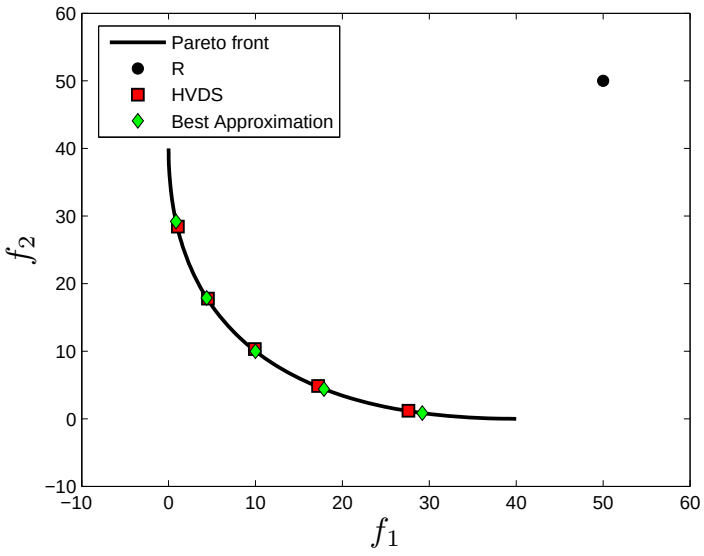


Figure 4.15: Numerical results of the 5 element HVDS on Convex.

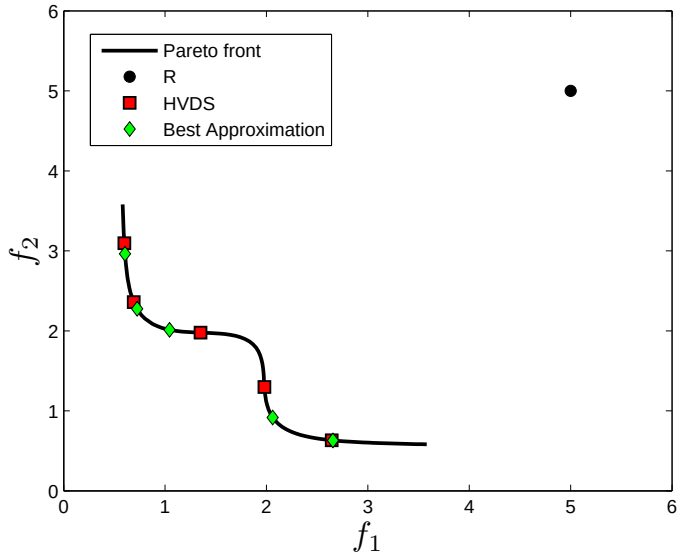


Figure 4.16: Numerical results of the 5 element HVDS on Dent.

	HVDS		SMS-EMOA		
	# Iterations	Hypervolume	# Iterations	Hypervolume	Best Value
Convex	1400	**2100.1424**	1400	1992.9788	2107.6523
Dent	885	**16.6941**	900	16.5721	16.8225

Table 4.1: Comparison of the 5 element HVDS and the SMS-EMOA with $\mu = 5$. The results are averaged over 20 test runs.

4.3.2 HVDS within SMS-EMOA

Finally, we investigate the potential of HVDS as local searcher coupled with SMS-EMOA. For the computations using Algorithm 5, we have realized the memetic algorithm as follows: To pull the current archive to the Pareto set, we have chosen to run two HVDS runs in the beginning of the search because the relatively high cost of this search (i.e., we have taken $m_1 = 3$ together with a budget of 50 iterations and $m_i = 0$ for $i > 1$). To perform the tests, the modified suite of Zitzler-Deb-Thile (ZDT) problems presented in [71] has been selected with the difference that we do not need the twice differentiability property. We add also Convex and Dent to the test problems. Those problems are described in Table 4.2 where is also included the

parameters for HVDS. Finally, for the SMS-EMOA setting, we establish a probability of 0.9 for crossover and $1/\mu$ for mutation.

Problem	Functions	Domain	# Variables	HVDS Param.
ZDT1	$f_1(x) = x_1$ $f_2(x) = g(x)(2 - \sqrt{f_1(x)/g(x)})$ $g(x) = 1 + \frac{9}{n-1}\sum_{i=1}^{n} x_i^2$	$[0,1] \times [-1,1]^n$	30 Variables	$a = 0.5$ $b = -0.9$ $R = [11, 11]$
ZDT2	$f_1(x) = x_1$ $f_2(x) = g(x)(2 - (f_1(x)/g(x))^2)$ $g(x) = 1 + \frac{9}{n-1}\sum_{i=1}^{n} x_i^2$	$[0,1] \times [-1,1]^n$	30 Variables	$a = -0.5$ $b = -0.9$ $R = [11, 11]$
ZDT3	$f_1(x) = x_1$ $f_2(x) = g(x)(2 - \sqrt{f_1(x)/g(x)}) -$ $(f_1(x)/g(x))sin(10\pi f_1(x)))$ $g(x) = 1 + \frac{9}{n-1}\sum_{i=1}^{n} x_i^2$	$[0,1] \times [-1,1]^n$	30 Variables	$a = 0.5$ $b = -0.9$ $R = [11, 11]$
ZDT4	$f_1(x) = x_1$ $f_2(x) = g(x)(2 - \sqrt{f_1(x)/g(x)})$ $g(x) = 1 + 10(n-1) +$ $+ \sum_{i=1}^{n}(x_i^2 - 10cos(4\pi f_1(x)))$	$[0,1] \times [-5,5]^n$	5 Variables	$a = 0.5$ $b = -0.9$ $R = [11, 11]$
Convex	Model defined in Equation (4.3)	$[-10, 10]^n$	10 Variables	$a = 0.5$ $b = -0.9$ $R = [50, 50]$
Dent	Model defined in Equation (4.13)	$[-100, 100]^n$	2 Variables	$a = -0.3$ $b = -0.96$ $R = [5, 5]$

Table 4.2: Test problems.

Table 4.3 and Figure 4.18 show some numerical results on the above test problems. Box-plots of the respective HV values after the final iteration are given in Figure 4.17. In 4 out of 6 cases the new hybrid is superior to its base EMOA while the differences in location of the HV values are not statistically significant for Dent and ZDT4. The latter is certainly due to the choice of the local search since the two runs got stuck in local minima, and hence, the effort was lost. Further variants of local search, e.g., the application of more but shorter HVDS runs, will be tested for HVDS as operator.

	SMS-EMOA			SMS-EMOA-HVDS		
	Average	Deviation	Median	Average	Deviation	Median
Convex	2003.867	68.956	2021.200	**2161.668**	18.039	2164.803
Dent	17.234	0.031	17.241	**17.245**	0.023	17.248
ZDT1	105.015	0.948	105.002	**108.965**	1.654	109.512
ZDT2	97.592	2.965	96.176	**107.463**	3.563	109.207
ZDT3	113.771	1.857	114.330	**116.097**	1.948	117.576
ZDT4	**76.536**	13.485	82.107	71.552	15.770	71.352

Table 4.3: HV results of SMS-EMOA with and without HVDS as local searcher after 2500 iterations of the algorithm (using the same number of function evaluations). The values are obtained from 20 test runs. Bold quantities are the ones with highest hypervolume.

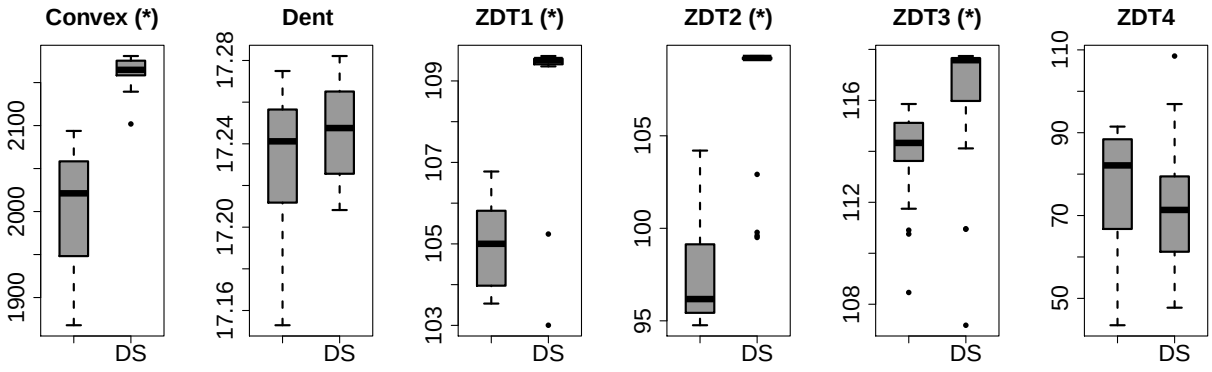
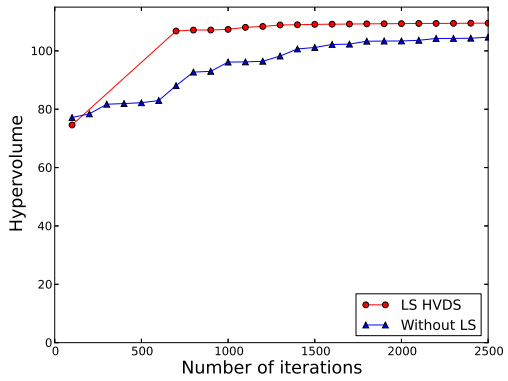
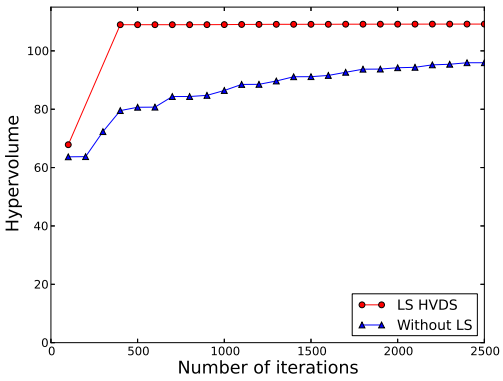


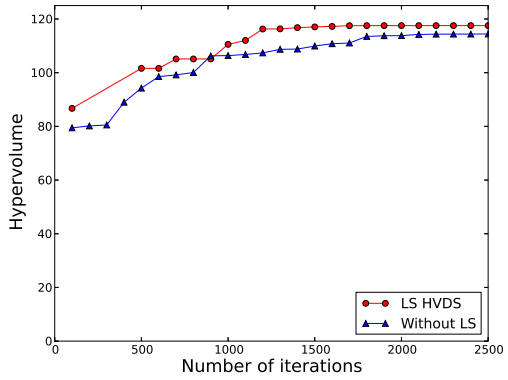
Figure 4.17: Box-plots of the HV at the final iteration of the SMS-EMOA and its hybrid variant using Algorithm 5 on the considered test problems. Statistically significant differences due to the Wilcoxon-Rank-Sum Test with $\alpha = 0.05$ are marked with (*).



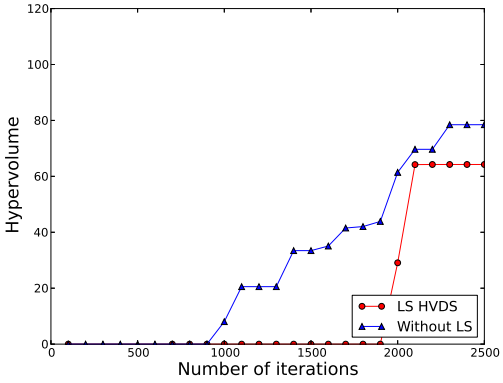
(a) ZDT1



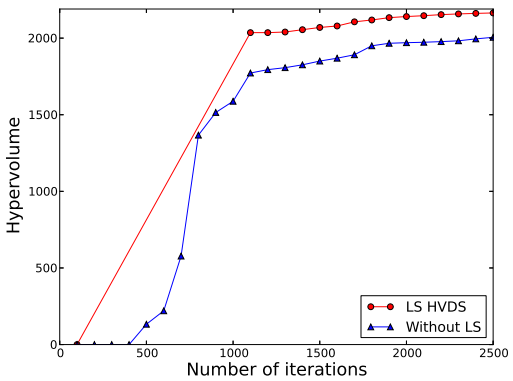
(b) ZDT2



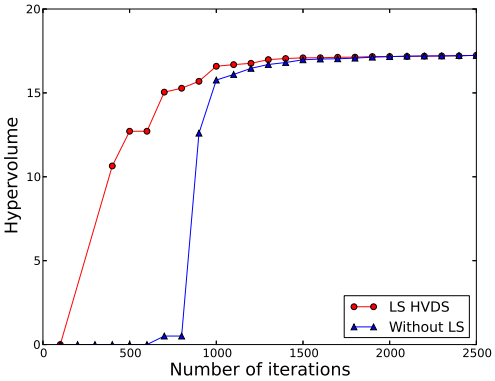
(c) ZDT3



(d) ZDT4



(e) Convex



(f) Dent

Figure 4.18: Hypervolume results of SMS-EMOA and its hybrid variant using Algorithm 5 on some benchmark models.

As can be seen in Table 4.3, the first memetic version of SMS-EMOA has troubles when it tries to tackle multi-modal problems as ZDT4. In order to overcome this problem, we perform new experiments using HVDS as an operator. Such change only needs a certain probability to apply the operator over the offspring that SMS-EMOA produces, as we already mentioned. Nevertheless, the probability has to be low, for uni-modal problems we have chosen $p = 0.1$ and for multi-modal problems $p = 0.001$ for the entire running. Numerical results using the HVDS as operator into SMS-EMOA are presented as follows:

				SMS-EMOA			SMS-EMOA-HVDS		
	Mod	Func	Var	Average	Dev	Median	Average	Dev	Median
Convex	U	2500	10	2003.867	68.956	2012.466	**2153.466**	27.275	2166.596
Dent	M	2500	2	111.772	6.729	105.506	**114.692**	4.469	115.87
ZDT1	U	2500	30	105.015	0.948	104.891	**109.125**	0.537	109.375
ZDT2	U	2500	30	97.592	2.965	96.172	**108.016**	1.495	108.643
ZDT3	M	2500	30	113.771	1.857	114.238	**116.998**	1.251	117.610
ZDT4	M	7500	10	62.963	14.061	61.275	**83.699**	7.220	85.353
ZDT6	M	7500	10	93.509	1.698	93.287	**99.843**	3.339	99.072
DTLZ1	M	25000	15	108.605	6.356	109.865	**113.569**	3.383	114.651
DTLZ2	U	1000	30	118.775	0.202	118.786	**120.043**	0.048	120.045
DTLZ3	M	25000	15	67.01	35.209	58.026	**103.095**	9.25	105.208
DTLZ4	U	1000	30	111.772	6.729	105.506	**114.692**	4.469	115.87

Table 4.4: HV results of SMS-EMOA with and without HVDS as operator (using the same number of function evaluations). The values are obtained over 20 test runs. Bold quantities are the ones with highest hypervolume.

Table 4.4 shows new results by using HVDS as an operator. Here, we can observed that for multi-modal problems (ZDT3, ZDT4, ZDT6, DTLZ1, and DTLZ3) the second memetic version of SMS-EMOA computes a higher hypervolume than SMS-EMOA without HVDS using the same number of function evaluations. To perform the new computations, we include into the test problems five new models ZDT6, DTLZ1, DTLZ2, DTLZ3, and DTLZ4 (DTLZ problems where taken from [72]) which are defined in Table 4.5. Considering the results obtained by using Algorithm 5 this new

version outperforms the older having better results according to the hypervolume.

As a final remark, all DTLZ problems were tested for only two objectives.

Problem	Functions	Domain
ZDT6	$f_1(x) = 1 - \exp -4x_1$ $f_2(x) = g(x)(2 - (f_1(x)/g(x))^2)$ $g(x) = 1 + \frac{9}{n-1}\sum_{i=1}^{n} x_i^2$	$[0,1] \times [-1,1]^n$
DTLZ1	$f_1(x) = (1 + g(x))0.5\prod_{i=1}^{k-1} x_i$ $f_{m=2:k-1}(x) = (1 + g(x))0.5(\prod_{i=1}^{k-m} x_i)(1 - x_{k-m+1})$ $f_k(x) = (1 + g(x))0.5(1 - x_1)$ $g(x) = 100(n + \sum_{i=1}^{n}((x_i - 0.5)^2 - \cos(20\pi(x_i - 0.5))))$	$[0,1]^n$
DTLZ2	$f_1(x) = (1 + g(x))\prod_{i=1}^{k-1}\cos(x_i\frac{\pi}{2})$ $f_{m=2:k-1}(x) = (1 + g(x))(\prod_{i=1}^{k-m}\cos(x_i\frac{\pi}{2}))\sin(x_{k-m+1}\frac{\pi}{2})$ $f_k(x) = (1 + g(x))\sin(x_1\frac{\pi}{2})$ $g(x) = \sum_{i=1}^{n}(x_i - 0.5)^2$	$[0,1]^n$
DTLZ3	$f_1(x) = (1 + g(x))\prod_{i=1}^{k-1} cos(x_i\frac{\pi}{2})$ $f_{m=2:k-1}(x) = (1 + g(x))(\prod_{i=1}^{k-m}\cos(x_i\frac{\pi}{2}))\sin(x_{k-m+1}\frac{\pi}{2})$ $f_k(x) = (1 + g(x))sin(x_1\frac{\pi}{2})$ $g(x) = 100(n + \sum_{i=1}^{n}((x_i - 0.5)^2 - \cos(20\pi(x_i - 0.5))))$	$[0,1]^n$
DTLZ4	As DTLZ2, except all $x_i \in x$ are replaced by x_i^α, where $\alpha > 0$	$[0,1]^n$

Table 4.5: Definition of the new test problems for the second memetic version of the SMS-EMOA using HVDS as an operator.

Finally, we show in the following figures (Figures 4.19, 4.20, 4.21, 4.22, 4.23, 4.24, 4.25, 4.26, 4.27, 4.28, and 4.29) a graphical comparison of this second memetic version against the standard version of SMS-EMOA. For this we select the averaged running of each algorithm. We can see clearly that the memetic version of the SMS-EMOA achieve better results using the same number of function evaluations.

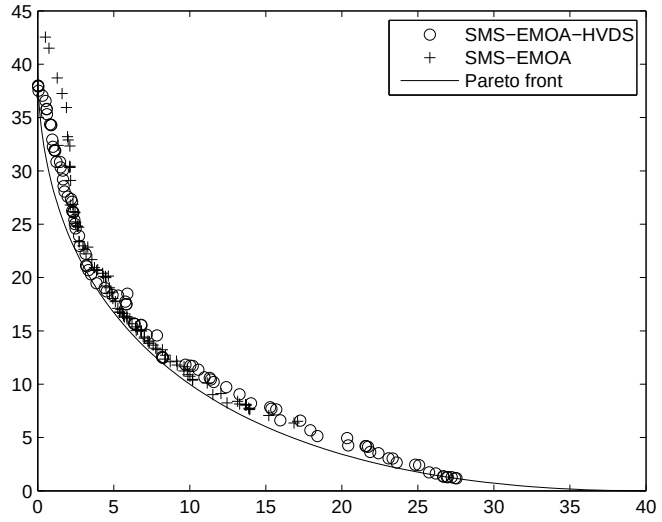


Figure 4.19: SMS-EMOA-HVDS solving Convex.

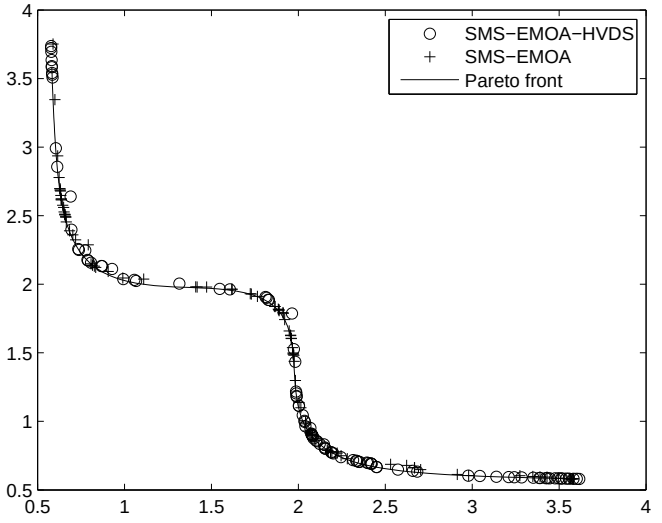


Figure 4.20: SMS-EMOA-HVDS solving Dent.

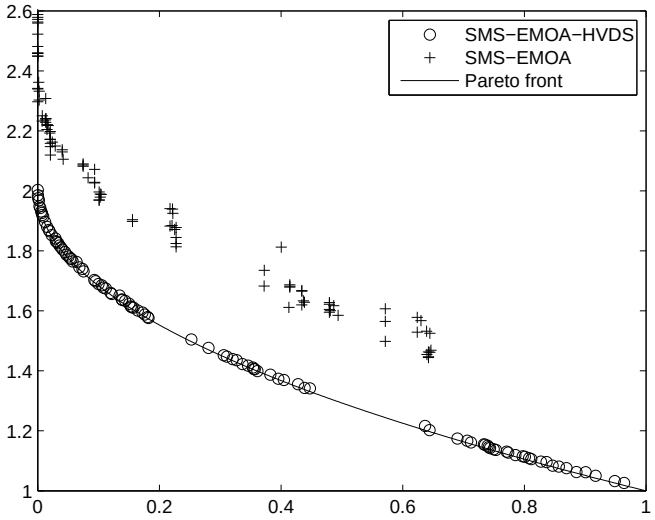


Figure 4.21: SMS-EMOA-HVDS solving ZDT1.

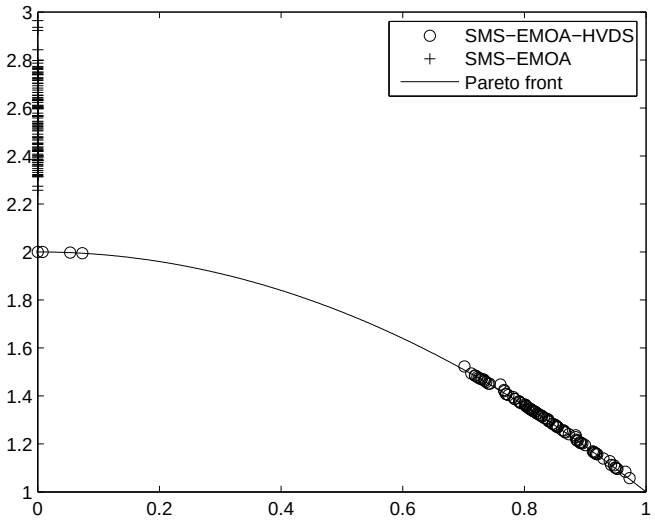


Figure 4.22: SMS-EMOA-HVDS solving ZDT2.

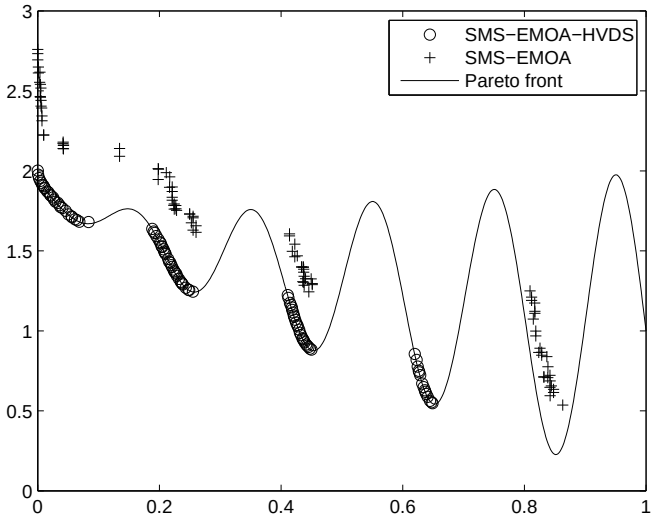


Figure 4.23: SMS-EMOA-HVDS solving ZDT3.

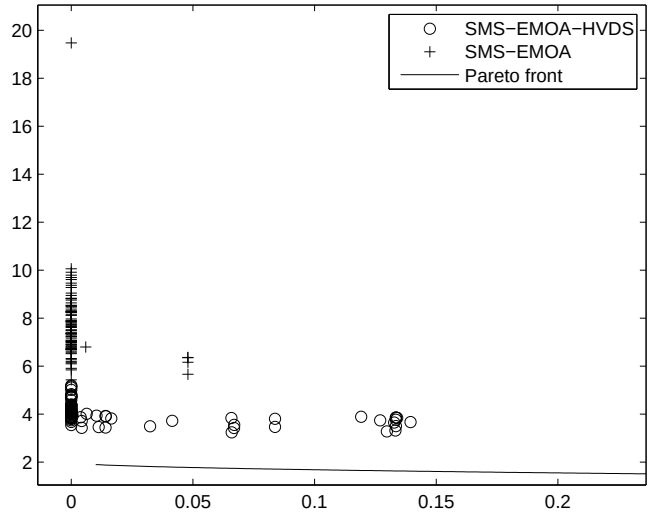


Figure 4.24: SMS-EMOA-HVDS solving ZDT4.

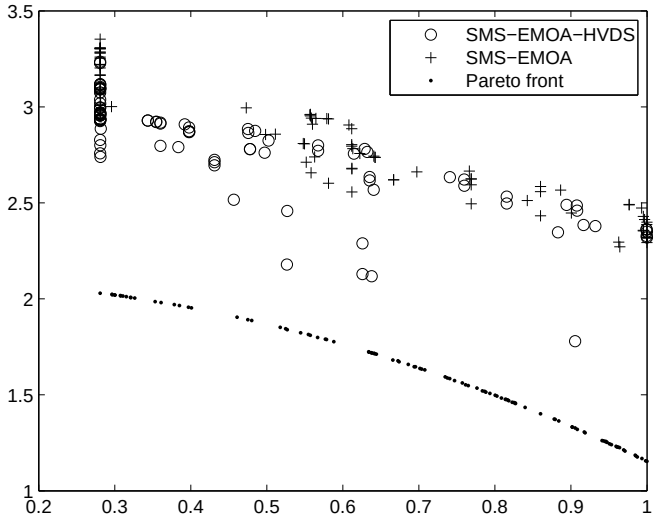


Figure 4.25: SMS-EMOA-HVDS solving ZDT6.

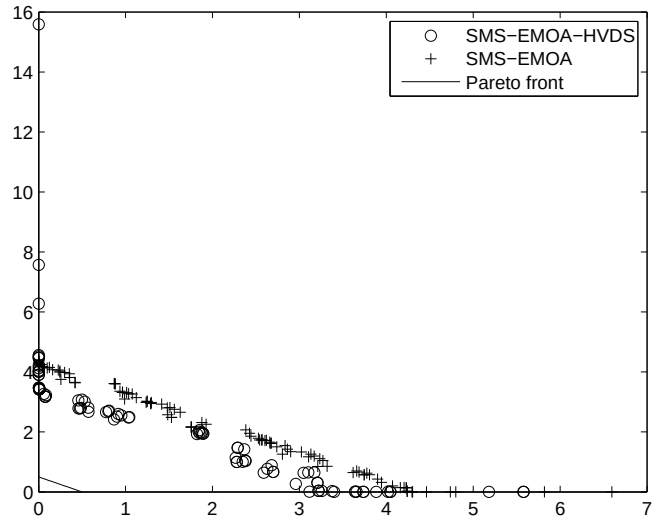


Figure 4.26: SMS-EMOA-HVDS solving DTLZ1.

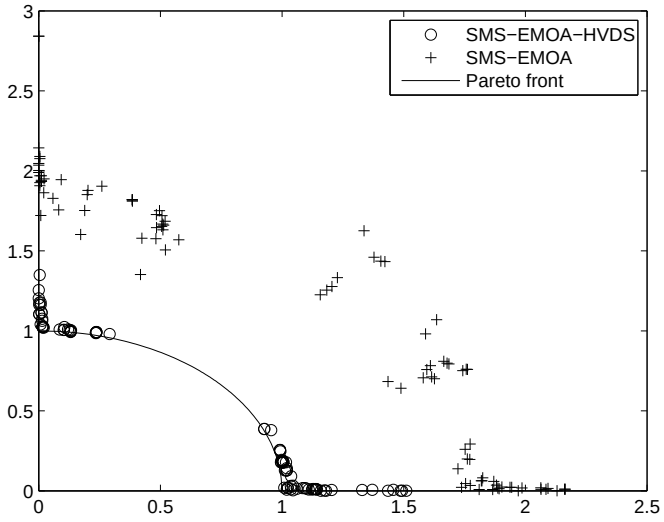


Figure 4.27: SMS-EMOA-HVDS solving DTLZ2.

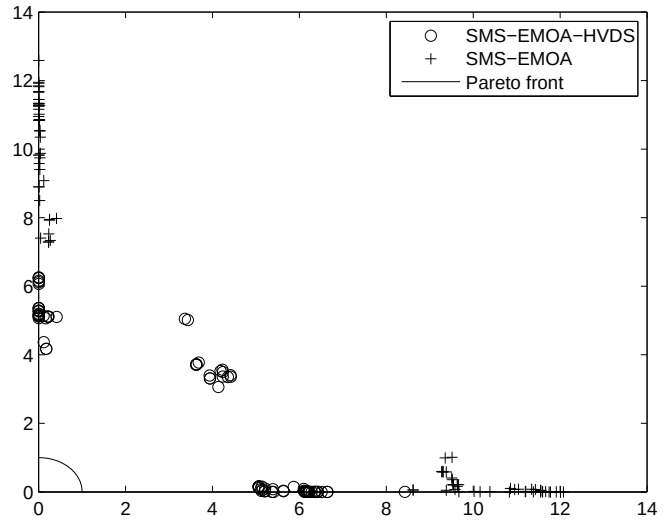


Figure 4.28: SMS-EMOA-HVDS solving DTLZ3.

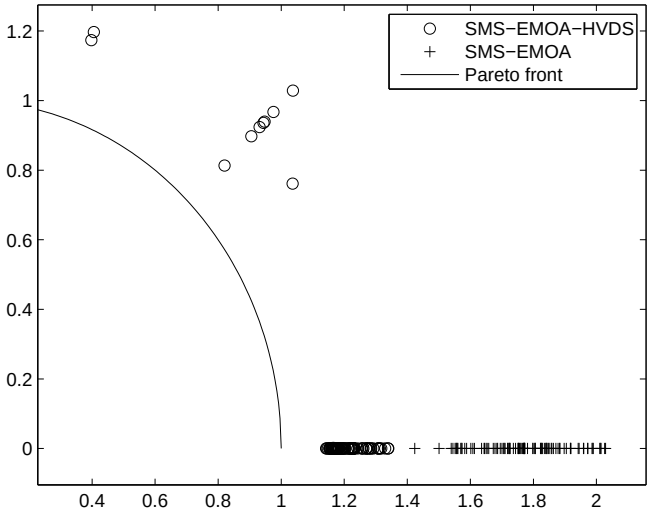


Figure 4.29: SMS-EMOA-HVDS solving DTLZ4.

5 | Conclusions and Future Work

One of the two main goals of this work has been the design of local search strategies for the numerical treatment of both MOPs and PMOPs. The other important goal was to design a novel memetic indicator based evolutionary algorithm to tackle MOPs. In the following, we list the conclusions obtained by this work:

- In the case of PMOPs, we presented the λ-DS which is an extension of the DS. We showed that it is advantageous to treat λ as a normal parameter what allows to state the λ-DS formulation. By using λ-DS, it is possible to get a direction ν by solving a system of linear equations that steers the search into any direction in objective space. Therefore, the DS has been successfully adapted to the context of PMOPs, leading to the algorithm λ-DS.

- We have presented a descent method using λ-DS. This method allows us to perform a search toward the family of Pareto fronts. We showed a relation with a widely use mathematical method, the Normal Boundary Intersection Method. Such relation was proven for both contexts. Thereby, a critical point of the descent method based on the λ-DS is a local solution of the NBI-subproblem, and vice versa.

- We studied the behavior of the stochastic local search for PMOPs. In this study, we discovered the influence of the parameter λ into a given problem. In the case of being far away, we concluded that the influence of this external parameter affects in a bad way the search toward the family of Pareto fronts. On the other hand, when a point is already over the family of Pareto fronts, the influence of λ does not affect the desired movement. Therefore, we propose to use the classical DS to reach a point over the family of Pareto fronts.

- We designed a predictor-corrector method over λ-space. For this PC, we defined the orthogonal vector to the linearization of the family of Pareto fronts. This vector allowed to compute a predictor direction. For the corrector step, we presented two different options. Thus, the proposed approach allows us to perform a search along λ-space without using any second gradient information.

- We presented two corrector methods. In Table 3.1, we showed a comparison between these two methods. The PCLDS1 appears to be cheaper but it is less accurate than PCLDS2. The PCLDS2 can produce a better continuation method speaking about quality. Nevertheless, there is no clear winner since both approaches have their advantages. Hence, we recommend to choose an option according to the context.

- We have shown in Table 3.2 a fair comparison against a classical continuation method. To show this, we had to adapt the KKT equations to the give context. In this comparison, the λ-DS was the clear winner, being the one that uses less function evaluation to perform such a movement in λ-space.

- Analog to the DS, we showed that the λ-DS could be used without using gradient information. The latter can e.g. be done using neighborhood information. We can say that this approach would be particularly advantageous when using evolutionary algorithms, since this neighborhood information would be obtained without cost.

- To tackle MOPs, we presented the HVDS for hypervolume approximations. We presented a division of the objective space based on the angle between the gradients. This division allows us to assign a given point for local search into three different distance regions. For each of the distance regions, we designed a specific movement. Such a movement increases the hypervolume contribution of a given solution point $\mathbf{x}$. Therefore, the HVDS is able reach points on the Pareto front at the same time that the hypervolume increases.

- We developed the HVDS to manage one ($A = \{\mathbf{x}\}$) and general ($A = \{\mathbf{x}_1, \ldots, \mathbf{x}_\mu\}$) element archives. A comparison for the HVDS as standalone algorithm was presented in Figure 4.10 and Table 4.1. This comparison showed the behavior of

the HVDS over the three distance regions. The HVDS obtained the highest hypervolume using the same budget of function evaluations. Thereby, the HVDS is a perfect choice to be coupled within an IBEA that uses the hypervolume.

- We designed a novel memetic hypervolume based evolutionary algorithm using the SMS-EMOA. This algorithm outperformed the basic version of SMS-EMOA. A problem arose in the case of multi-modal problems by using our method only at the beginning. To solve this problem, we presented a second version of the memetic algorithm using the HVDS as an operator. This second version of SMS-EMOA-HVDS was able to tackle uni-modal as well as multi-modal problems. Hence, we have successfully adapted the HVDS into an IBEA.

5.1 Future Work

There are many interesting aspects to be covered for future work. In the case of PMOPs, to test the λ-DS on more parameter dependent models is desired since our first results remarks a clear advantage over classical continuation approaches. A study of the behavior of the λ-DS for $l > 1$ (where l represents the number of extenal parameters) is still missing. Next, it would be desirable to design a novel evolutionary algorithm which obtains over one single run an approximation of the complete family of solutions. The latter idea gives us a perfect candidate to integrate the λ-DS in order to improve its performance. Finally, it would also be interesting to design an indicator to measure the quality of an approximation of the family of Pareto fronts.

For the HVDS, other important improvements would be interesting. First of all, the current study was restricted to unconstrained bi-objective problems which has to be generalized for sake of a broader applicability. To investigate other ways to assign elements over objective space will also allows to improve our current method. Further, it would be desirable to use the gradient free version of the Directed Search method in the case of Regions II and III, in order to exploit the neighborhood information available when a MOEA is used. Finally, it might be interesting to adapt the method to other indicators, e.g, AHD [73], $R2$ [74], preference indicators [75, 76], among others.

Bibliography

[1] Honda motorcycles, http://www.honda.mx/motos/, September 2013.

[2] C. A. Coello Coello. Evolutionary Multi-Objective Optimization: A Historical View of the Field. *Computational Intelligence Magazine, IEEE*, 1(1):28–36, 2006.

[3] C. Hillermeier. *Nonlinear Multiobjective Optimization: A Generalized Homotopy Approach*. Springer, 2001. ISBN-13:9783764364984.

[4] S. Gass and T. Saaty. The Computational Algorithm for the Parametric Objective Function. *Naval Research Logistics Quarterly*, 2(1):39–45, 1955.

[5] K. Belkeziz and M. Pirlot. Proper Efficiency in Nonconvex Vector-Maximization-Problems. *European Journal of Operational Research*, 54(1):74–80, 1991.

[6] O. Schütze. *Set Oriented Methods for Global Optimization*. Dissertation, Universität Paderborn, December 2004.

[7] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan. A Fast and Elitist Multiobjective Genetic Algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*, 6(2):182–197, 2002.

[8] Q. Zhang and H. Li. MOEA/D: A Multiobjective Evolutionary Algorithm Based on Decomposition. *IEEE Transactions on Evolutionary Computation*, 11(6):712–731, 2007.

[9] K. Deb. *Multi-Objective Optimization using Evolutionary Algorithms*. John Wiley & Sons, Chichester, UK, 2001. ISBN 0-471-87339-X.

[10] K. Gerstl, G. Rudolph, O. Schütze, and H. Trautmann. Finding Evenly Spaced Fronts for Multiobjective Control via Averaging Hausdorff-Measure. *Proceedings*

of 8th International Conference on Electrical Engineering, Computer Science and Automatic Control (CCE 2011), 2011.

[11] M. T. M. Emmerich, A. H. Deutz, and J. W. Kruisselbrink. On quality indicators for black-box level set approximation. In *EVOLVE - A Bridge between Probability, Set Oriented Numerics and Evolutionary Computation*, volume 447 of *Studies in Computational Intelligence*, pages 157–185. Springer Berlin, Heidelberg, 2013.

[12] A. Jaszkiewicz, H. Ishibuchi, and Q. Zhang. Multiobjective memetic algorithms. In *Handbook of Memetic Algorithms*, volume 379 of *Studies in Computational Intelligence*, pages 201–217. Springer Berlin, Heidelberg, 2012.

[13] M. Farina, K. Deb, and P. Amato. Dynamic Multiobjective Optimization Problems: Test Cases, Approximations, and Applications. *IEEE Transactions on Evolutionary Computation*, 8(5):425–442, 2004.

[14] K. Witting. *Numerical Algorithms for the Treatment of Parametric Multiobjective Optimization Problems and Applications*. Dissertation, Universität Paderborn, February 2012.

[15] O. Schütze, A. Lara, and C. A. Coello Coello. The Directed Search Method for Unconstrained Multi-Objective Optimization Problems. Technical Report TR-OS-2010-01, CINVESTAV-IPN, 2010.

[16] K. M. Miettinen. *Nonlinear Multiobjective Optimization*. Kluwer Academic Publishers, 1999.

[17] V. Pareto. Cours d'economie politique, volume i and ii. f. rouge, lausanne. 1896.

[18] F. Y. Edgeworth. Mathematical psychics. 1881.

[19] H. Kuhn and A. Tucker. Nonlinear programming. In *Proceedings of the Second Berkeley Symposium on Mathematical Statistics and Probability*, page 481–492. University of California Press, Berkeley and Los Angeles, 1951.

[20] W. E. Karush. *Minima of functions of several variables with inequalities as side conditions*. PhD thesis, Dept. Math., Univ. Chicago, 1939.

[21] C. A. Coello Coello, G. B. Lamont, and D. A. Van Veldhuizen. *Evolutionary Algorithms for Solving Multi-Objective Problems*. Springer, 2nd edition edition, 2007.

[22] J.R. Munkres. *Analysis On Manifolds*. Advanced Books Classics. Westview Press, 1997.

[23] L.A. Zadeh. Optimality and non-scalar-valued performance criteria. *IEEE Transactions on Automatic Control*, 8:59–60, 1963.

[24] Y. Y. Haimes, L.S. Lasdom, and D.A. Wismer. On a Bicriterion Formulation of the Problems of integrated system identification and system optimization. *IEEE Transactions on Systems, Man, and Cybernetics*, 1:296–297, 1971.

[25] Bowman and V. Joseph. On the relationship of the tchebycheff norm and the efficient frontier of multiple-criteria objectives. In HervÃ© Thiriez and Stanley Zionts, editors, *Multiple Criteria Decision Making*, volume 130 of *Lecture Notes in Economics and Mathematical Systems*, pages 76–86. Springer Berlin Heidelberg, 1976.

[26] I. Das and J. Dennis. Normal-boundary intersection: A new method for generating the pareto surface in nonlinear multicriteria optimization problems.

[27] G. Stehr, H. Graeb, and K. Antreich. Performance trade-off analysis of analog circuits by normal-boundary intersection. In *Proceedings of the 40th annual Design Automation Conference*, DAC '03, pages 958–963, New York, NY, USA, 2003. ACM.

[28] O. Schütze M. Dellnitz and T. Hestermeyer. Covering pareto sets by multilevel subdivision techniques.

[29] M. Dellnitz, O. Schütze, and S. Sertl. Finding zeros by multilevel subdivision techniques. *IMA Journal of Numerical Analysis*, 22(2):167–185, 2002.

[30] P. Deuflhard, M. Dellnitz, O. Junge, and C. Schütte. Computation of essential molecular dynamics by subdivision techniques. In *Computational molecular dynamics: challenges, methods, ideas*, pages 98–115. Springer, 1999.

[31] C.S. Hsu. Cell-to-cell mapping: A method of global analysis for nonlinear systems. *Applied mathematical sciences*.

[32] C. Hernándeza, O.Schütze, and J. Q. Sun. Computing the Set of Approximate Solutions of a Multi-Objective Optimization Problem by Means of Cell Mapping techniques. In *EVOLVE - A Bridge between Probability, Set Oriented Numerics, and Evolutionary Computation IV*, Advances in Intelligent Systems and Computing. Springer, 2013.

[33] C. Hernández, Y. Narajani, Y. Sardahi, W. Liang, and O. Schütze. Simple cell mapping method for multiobjective optimal pid control design. *International Journal of Dynamics and Control*, 2013.

[34] S. Schäffler, R. Schultz, and K. Weinzierl. A stochastic method for the solution of unconstrained vector optimization problems. *Journal of Optimization Theory and Applications*, 114(1):209–222, 2002.

[35] J. Fliege and B. Svaiter. Steepest descent methods for multicriteria optimization. *Mathematical Methods of Operations Research*, 51(3):479–494, 2000.

[36] P. Bosman and D. de Jong. Exploiting gradient information in numerical multi–objective evolutionary optimization. In *Proceedings of the 2005 conference on Genetic and evolutionary computation*, GECCO '05, pages 755–762, New York, NY, USA, 2005. ACM.

[37] A. Lara, C. A. Coello Coello, and O. Schütze. A painless gradient-assisted multi-objective memetic mechanism for solving continuous bi-objective optimization problems. In *Evolutionary Computation (CEC), 2010 IEEE Congress on*, pages 1–8. IEEE, IEEE Press, 2010.

[38] E. Allgower and Georg K. *Numerical Continuation Methods*. Springer, 1990.

[39] D. Mayerich and J. Keyser. Hardware accelerated segmentation of complex volumetric filament networks. *IEEE Transactions on Visualization and Computer Graphics*, 15(4):670–681, 2009.

[40] O. Schütze, A. Lara, and C. A. Coello Coello. The directed search method for unconstrained multi-objective optimization problems. In *Proceedings of the*

EVOLVE – A Bridge Between Probability, Set Oriented Numerics, and Evolutionary Computation, 2011.

[41] E. Salinas Márquez and O. Schütze. Gradient-free continuation method for box-constrained multi-objective optimization problems. *EVOLVE extended abstract proceedings*, 2013.

[42] J. Nocedal and S. Wright. *Numerical Optimization*. Springer Series in Operations Research and Financial Engineering. Springer, 2006.

[43] P. A.N. Bosman and E. D. de Jong. Exploiting gradient information in numerical multi-objective evolutionary optimization. In Hans-Georg Beyer et al., editor, *2005 Genetic and Evolutionary Computation Conference (GECCO 2005)*, volume 1, pages 755–762, New York, USA, June 2005. ACM Press.

[44] E. Zitzler and L. Thiele. Multiobjective Optimization Using Evolutionary Algorithms - A Comparative Case Study. In *Conference on Parallel Problem Solving from Nature (PPSN V)*, pages 292–301, Berlin, 1998. Springer.

[45] N. Beume. S-metric calculation by considering dominated hypervolume as Klee's measure problem. *Evolutionary Computation*, 17(4):477–492, 2009.

[46] J. Knowles and D. Corne. Properties of an adaptive archiving algorithm for storing nondominated vectors. *IEEE Transactions on Evolutionary Computation*, (7):100–116, 2003.

[47] D. P. Huttenlocher, G. A. Klanderman, and W. J. Rucklidge. Comparing Images Using the Hausdorff Distance.

[48] D. A. Van Veldhuizen and G. B. Lamont. On measuring multiobjective evolutionary algorithm performance. In *2000 Congress on Evolutionary Computation*, volume 1, pages 204–211, Piscataway, New Jersey, July 2000. IEEE Service Center.

[49] C. A. Coello Coello and N. Cruz Cortés. Solving Multiobjective Optimization Problems using an Artificial Immune System. *Genetic Programming and Evolvable Machines*, 6(2):163–190, June 2005.

[50] O. Schütze, X. Esquivel, A. Lara, and C. A. Coello Coello. Measuring the Averaged Hausdorff Distance to the Pareto Front of a Multi-Objective Optimization Problem. Technical Report TR-OS-2010-02, CINVESTAV-IPN, 2010.

[51] A. Abraham, L. Jain, and R. Goldberg. *Evolutionary Multiobjective Optimization-Theoretical Advances and Applications*. Springer, 2005.

[52] N. Beume, B. Naujoks, and M. Emmerich. SMS-EMOA: Multiobjective selection based on dominated hypervolume. *European Journal of Operational Research*, 2006.

[53] N. Beume, B. Naujoks, and M. Emmerich. An EMO Algorithm Using the Hypervolume Measure as Selection Criterion. *In Coello, C. A. C.,Aguirre, A. H., and Zitzler, E., editors, EMO*, pages 62–76, 2005.

[54] M. A. Montes de Oca, C. Cotta, and F. Neri. Local Search. In *Handbook of Memetic Algorithms*, volume 379 of *Studies in Computational Intelligence*, pages 29–41. Springer Berlin, Heidelberg, 2012.

[55] P. Moscato. On Evolution, Search, Optimization, Genetic Algorithms and Martial Arts: Towards Memetic Algorithms. Technical Report 826, California Institute of Technology, 1989.

[56] R. Dawkins. *The Selfish Gene*. Popular Science, 1989.

[57] A. Lara, G. Sanchez, C. A. Coello Coello, and O. Schütze. HCS: A New Local Search Strategy for Memetic Multiobjective Evolutionary Algorithms. *IEEE Transactions on Evolutionary Computation*, 14(1):112–132, 2010.

[58] E. Zitzler, M. Laumanns, , and L. Thiele. SPEA2: Improving the strength pareto evolutionary algorithm. In *Evol. Methods Design, Optimization Control Applicat. Ind. Problems (EUROGEN '01)*, pages 95–100, 2002.

[59] S. J. Alvarado-García. *On Pointwise Iterative Local Search Techniques for Evolutionary Multiobjective Optimization*. Dissertation, CINVESTAV, December 2012.

[60] B. Bank, J. Guddat, D. Klatte, B. Kummer, and K. Tammer. *Non-Linear Parametric Optimization*. Akademie-Verlag, Berlin, 1982.

[61] C. Philippe, E. Cathy, and G. Alessio. An Evolutionary Approach for Time dependant Optimization. *International Journal on Artificail Intelligence Tools*, 1993.

[62] J. Branke, T. Kaussler, C. Schmidt, and H. Schmeck. A Multi-Population Approach to Dynamic Optimization Problems. Technical report, University of Karlsruhe, 2000.

[63] K. Yamasaki. Dynamic Pareto Optimum GA against the changing environments. Technical report, Department of Information Science Tokyo University of Information Sciences, 2001.

[64] Y. Kazuko, K. Kazuhisa, and S. Masuteru. Dynamic Optimization by Evolutionary Algorithms Applied to Financial Time Series. In *IEEE Congress on Evolutionary Computing*, volume 2, pages 2017–2022, 2002.

[65] A. A. Tantar, E. Tantar, and P. Bouvry. A classification of dynamic multi-objective optimization problems. In *Proceedings of the 13th annual conference companion on Genetic and evolutionary computation*, GECCO '11, pages 105–106, New York, NY, USA, 2011. ACM.

[66] R.B. Bapat. Generalized inverses. In *Linear Algebra and Linear Models*, Universitext, pages 31–36. Springer London, 2012.

[67] A. Griewank. *Evaluating Derivatives: Principles and Techniques of Algorithmic Differentiation*. Number 19 in Frontiers in Appl. Math. SIAM, Philadelphia, PA, 2000.

[68] M. Brown and R. E. Smith. Directed multi-objective optimisation. *International Journal of Computers, Systems and Signals*, 6(1):3–17, 2005.

[69] E. Mejia and O. Schütze. A predictor corrector method for the computation of boundary points of a multi-objective optimization problem. In *International Conference on Electrical Engineering, Computing Science and Automati Control (CCE 2010)*, pages 1–6, 2007.

[70] O. Schütze, A. Lara, and C. A. Coello Coello. On the influence of the number of objectives on the hardness of a multiobjective optimization problem. *IEEE Transactions on Evolutionary Computation*, 15(4):444–455, 2011.

[71] E. Zitzler, K. Deb, and L. Thiele. Comparison of Multiobjective Evolutionary Algorithms: Empirical Results. *Evol. Comput., vol. 8, no. 2, pp.173-195*, 2000.

[72] K. Deb, L. Thiele, M. Laumanns, and E. Zitzler. Scalable Multi-Objective Optimization Test Problems. In *Congress on Evolutionary Computation (CEC 2002)*, pages 825–830. IEEE Press, 2002.

[73] O. Schütze, X. Esquivel, 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*, 2012. forthcoming, see also http://delta.cs.cinvestav.mx/~schuetze/technical_reports/index.html.

[74] D. Brockhoff, T. Wagner, and H. Trautmann. On the properties of the r2 indicator. In *Proceedings of the fourteenth international conference on Genetic and evolutionary computation conference*, GECCO '12, pages 465–472, New York, NY, USA, 2012. ACM.

[75] T. Wagner, H. Trautmann, and D. Brockhoff. Preference Articulation by Means of the R2 Indicator. In *Evolutionary Multi-criterion Optimization (EMO 2013)*, volume 7811, pages 81–95, Sheffield, Royaume-Uni, March 2013.

[76] A. Auger, J. Bader, D. Brockhoff, and E. Zitzler. Articulating User Preferences in Many-Objective Problems by Sampling the Weighted Hypervolume. In G. Raidl et al., editors, *Genetic and Evolutionary Computation Conference (GECCO 2009)*, pages 555–562, New York, NY, USA, 2009. ACM.