Centro de Investigación y de Estudios Avanzados
del Instituto Politécnico Nacional

Unidad Zacatenco

Departamento de Computación

Pareto Tracer: Método Predictor Corrector para Problemas de Optimización Multiobjetivo

Tesis que presenta

Adanay Martín Pérez

para obtener el Grado de

Maestro en Ciencias

en la Especialidad de

Computación

Director de la Tesis

Dr. Oliver Schütze

México, D.F. **Diciembre 2014**

Centro de Investigación y de Estudios Avanzados
del Instituto Politécnico Nacional

Campus Zacatenco

Computer Science Department

Pareto Tracer: A Predictor Corrector Method for Multi-objective Optimization Problems

Submitted by

Adanay Martín Pérez

as fulfillment of the requirement for the degree of

Master in

Computer Science

Advisor

Dr. Oliver Schütze

Mexico, D.F. **December 2014**

Resumen

En un gran número de aplicaciones prácticas se requiere la optimización simultánea de varios objetivos. Dado que los criterios a optimizar suelen ser contradictorios, no resulta sorprendente que estos problemas posean un conjunto de soluciones (denominado conjunto de Pareto) en lugar de un óptimo absoluto. Además, bajo ciertas condiciones se tiene que dicho conjunto es típicamente un objeto suave (similar a una superficie o sólido). Por consiguiente, a fin de facilitar la toma de decisiones, nuestra meta es obtener una representación finita del conjunto de Pareto. Alternativamente, se podría restringir la búsqueda a una zona específica en situaciones en las que el conjunto de interés sea excesivamente grande.

En esta tesis se propone un nuevo método predictor corrector para resolver problemas de optimización multi-objetivo mediante técnicas de continuación. El algoritmo, denominado *Pareto Tracer*, fue diseñado para trazar la curva (u objeto dimensional) de soluciones de Pareto (locales) en problemas con un número arbitrario de objetivos. Adicionalmente, *Pareto Tracer* es capaz de lidiar con restricciones tanto de caja como de igualdad mediante una modificación del método de Newton utilizado como corrector. Se proponen además dos variantes basadas respectivamente en métodos quasi-Newton y de gradiente descendente que no requieren el uso de segundas derivadas. El desempeño de la nueva propuesta se discute primero teóricamente y luego se evalúa en varios ejemplos de prueba.

Abstract

In many real-world applications, the problem arises that several objectives have to be optimized concurrently leading to a multi-objective optimization problem. Since these goals are typically contradictory, it comes as no surprise that the solution set—the so-called *Pareto set*—of such problems does (in general) not consist of one single solution. Moreover, under some mild conditions, this set is typically a smooth manifold (like a surface or solid). For the decision maker it is hence desired to obtain a suitable finite size representation of the optimal set, or to explore (locally) the possibilities around a given solution in situations where the entire Pareto set is too large.

In this thesis, we devise a novel predictor corrector method to address multi-objective optimization problems by continuation. The algorithm, Pareto Tracer, is capable *to trace* the manifold of (local) Pareto solutions of a problem with in principle any number of objectives. Our proposal can cope with box and equality constraints for which a Newton-like method was designed to deal with restrictions. Furthermore, two Hessian-free realizations of the Pareto Tracer were developed based respectively on quasi-Newton and gradient descent approaches. We discuss the algorithm first theoretically and demonstrate further on its strength on several examples.

A mi abuela Juanita, quién siempre llevo conmigo...

Agradecimientos

Quisiera agradecer especialmente a mi asesor de tesis, el Dr. Oliver Schütze, cuya dedicación a este trabajo sobrepasó todas mis expectativas como estudiante. Cuyos relevantes aportes, críticas y comentarios han sido esenciales en el desarrollo de esta investigación. Por haberme brindado la oportunidad de recurrir a sus conocimientos y experiencia en el área y por mostrarme el apasionante mundo de la investigación científica se ha ganado mi lealtad y total admiración.

Le agradezco eternamente a mi familia por creer en mí y apoyarme incondicionalmente. A mis padres Olguita y Amauris por todo su cariño. A mi hermana Anabel cuya fortaleza es mi fuente de inspiración y a mi sobrina Alejandra por todo su amor y por ser una niña maravillosa.

A Joaquín e Israel por abrirme las puertas de sus casas.

Agradezco a CONACyT la beca de maestría que me otorgó durante dos años para cursar estudios en el CINVESTAV-IPN y por el apoyo recibido para participar en un congreso en el extranjero. Gracias por hacer realidad un sueño.

Agradezco a también a todos los que fueron mis profesores y compañeros de clase, con los que compartí dos años llenos de experiencias y me enseñaron a conocer este país encantador. A Luis y Manoat(e)l. A Petlachi, Imanti, Alex, Jesús... A mis amigos Yiny y Molinet por estar siempre presentes y compartir conmigo la aventura de venir a estudiar a México.

Agradezco a Sofía, a Felipa y a todo el personal del Departamento de Computación por su apoyo y consideración.

Al Dr. Debrup Chakraborty por sus clases magistrales. Al Dr. Carlos A. Coello Coello por su excelencia como investigador y por enseñarme el verdadero concepto de puntualidad. A mis sinodales, el Dr. Luis Gerardo de la Fraga y el Dr. Francisco Rodríguez Henríquez con quién además tuve la oportunidad de pasar mi examen de ingreso en Cuba. Al Dr. José Guadalupe Rodríguez por confiar en mí y por su invaluable guía en el proceso de aceptación en esta maestría.

Gracias a todos los que de una forma u otra formaron parte de esta experiencia inolvidable.

Contents

List of Figures

List of Tables

List of Algorithms

List of Acronyms

MOMP multi-objective mathematical programming. 22, 23

MOP multi-objective optimization problem. vii, xvii–xix, xxi, 9, 12–16, 22, 23, 25, 28–31, 34, 36–38, 40, 43, 47, 48, 54, 56–59, 67, 69, 70, 72, 79–81, 83, 85, 88, 89, 92, 97, 99–103, 105–107, 110–112, 114–117, 122, 123, 125–129

MP mathematical programming. 16, 22

NBI Normal Boundary Intersection. xiii, xvii, 24–28, 41–43

NDS Newton direction subproblem. 55–57, 69, 70, 91–94, 100, 114, 115, 117, 126

NLP nonlinear programming. 16, 18

NLS nonlinear least squares. 17, 20

NNC Normalized Normal Constraint. xiii, xvii, 27, 28, 42, 43

NNEC normalized normal equality constraint. 42

ODE ordinary differential equation. 30, 31, 43

ODE NNEC ODE Normalized Normal Equality Constraint. 42

PC predictor corrector. vii, xvii, xix, 5, 6, 36–38, 40, 43, 45, 47, 58, 60, 73–77, 79–81, 96, 97, 99, 100, 125

PP Physical Programming. 24, 28

PPF Pareto Path Following. 45, 129

PT Pareto Tracer. v, vii, xvii–xix, xxi, 5–7, 47, 58–60, 62–66, 71–81, 83, 92, 93, 95–103, 106, 115, 116, 118–123, 125–130

QCLP quadratically constrained linear programming. 17, 21, 35, 55, 70, 91, 115, 127

QCQP quadratically constrained quadratic programming. 17, 21, 55

QN quasi-Newton. v, vii, xvii, xviii, 4–6, 16, 19, 32, 34, 35, 47, 67, 69–81, 83, 93–103, 106, 116–123, 127, 129

QP quadratic programming. 16, 21, 35, 94, 127

SD steepest descent. 4, 6, 16, 18, 31, 32, 34, 35, 44, 45, 67, 69–77, 79–81, 93–100, 103, 105, 106, 112, 116, 117, 119–122, 127–129

SDE stochastic differential equation. 30

SOCP second-order cone programming. 17, 21, 55–57, 70, 92, 94, 115

SOP single-objective optimization problem. 9, 10, 13, 16, 22, 23, 28, 32, 34, 40, 70, 128, 129

SP semidefinite programming. 17, 21

WS weighted sum. 23, 24

ZDT Zitzler-Deb-Thile. 77

ZS Zigzag Search. xvii, 44, 45

Chapter 1

Introduction

In many situations in everyday life, we face the problem that we want to optimize certain quantity. In business applications, for instance, it would be desirable to minimize costs and to maximize profits, while transportation management systems call for the minimization of distances or travel times. Basically, we can understand *optimization* as the process of selecting the best choice (with regard to some criteria) among a set of possible alternatives. However, together with this elementary notion of optimization, the question arises of what we consider to be *the best choice.* In a typical scenario, but not necessarily indicative for all situations, we have a single goal in mind. Think, for instance, of situations where we need the cheapest supplier of some service or the fastest way to reach a specific location. The solutions to these problems can be described quantitatively in terms of price and time, respectively. Thus, we can establish an ordering between the available choices to select the best option, which is, accordingly to the last examples, the supplier with the lowest price or the travel plan that takes you to your destiny in the minimum amount of time.

A less intuitive scenario, nevertheless present in a great variety of applications, is when several objectives have to be optimized concurrently. As a general example, two common goals in product design are certainly to maximize the quality of the product and to minimize its cost. Undoubtedly, high quality designs will be very attractive to the customers. However, producing such nice models may increase the cost per unit which goes along with fewer profits for the manufacturer, or alternatively, the necessity to launch expensive products to the market. The last choice, on the other hand, may lead several clients to reconsider its initial best and look for more accessible options. A decision that, in the end, will be translated again in less benefits.

It is thereby mandatory that we arrive at a consensus over the meaning of an optimal solution according to more than one objective. For this purpose, we employ the notion of Pareto efficiency or Pareto optimality[1], which states that a solution becomes better if at least one of its goals is improved while the others do not get worse.

[1] Named after the work of Vilfredo Pareto (1848-1923) in [1].

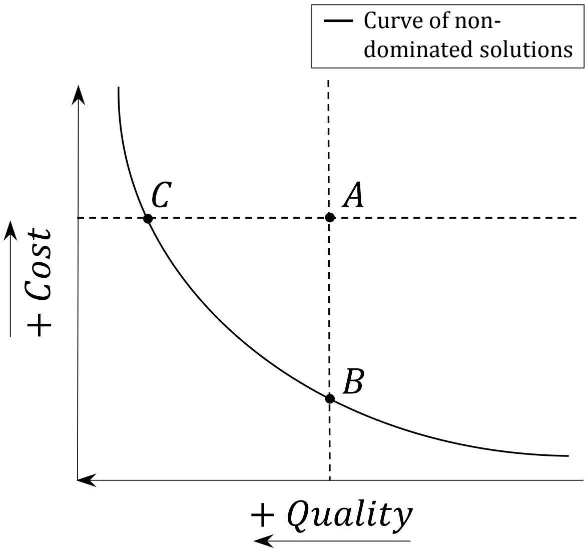


Figure 1.1: Fictitious curve of non-dominated solutions of the well-known problem 'quality versus cost'.

This term is also known as Pareto dominance. We will get deeper into this concept in the next chapters, but for now it is good enough to think of an optimal solution as one that is non-dominated (non-improved) by any other. Moreover, since our goals are typically contradictory, it comes as no surprise that the solution set, the so called Pareto set, respectively its image, the Pareto front, does in general not consist of one single solution. To make it clear, we go back to our last example where the optimal product designs are those which cannot improve its quality without spending more money. Figure 1.1 shows a fictitious curve[2] of non-dominated points, i.e, each point on the curve represents an optimal solution in objective space. From the picture, it is easy to see that choosing the design A is not a good idea: we can save money by taking B which is a model of the same quality, or alternatively, if we choose C, we end up with a higher quality product by spending the same amount of money. We cannot, however, decide between B and C: we will obviously prefer C, but may be unwilling to pay that much money.

Pareto efficiency is therefore a practical criterion for evaluating our possibilities: an outcome that is not Pareto optimal can be potentially improved thus should be avoided. Hence, our task is to restrict the attention to the set of non-dominated choices. Then, a specialized designer can make tradeoffs within this set, rather than considering the full range of possibilities.

[2] We will see later in this thesis that under certain smoothness assumptions, the solution set of a problem like this forms—at least locally—a curve.

1.1 Motivation

Multi-objective optimization methods[3] have been finding ever wider use in different areas of science, engineering, industry, or economics, and are an essential tool in practical scenarios of decision making. Several strategies in this field have been proposed and tested over the past decades. There are, for instance, scalarization methods [2, 3, 4, 5, 6, 7] that transform (by parametrization) the original multi-objective problem into a *classical* single-objective one. By choosing a clever sequence of parameters, a suitable finite size approximation of the entire Pareto set can be obtained. However, a remarkable drawback of this scheme lies in the selection of such a convenient sequence of parameters. Further, there exist many set oriented methods such as specialized evolutionary strategies [8, 9, 10], subdivision techniques [11, 12], or cell mapping techniques [13, 14]. These methods have in common that they use sets in an iterative manner and are thus able to deliver an approximation of the solution set in one run of the algorithm. The limitation here, though, has a lot to do with the efficiency. Evolutionary algorithms, on one side, can be seen as stochastic global search methods inspired by genetics and natural selection, but are well-known by its *relatively* slow converge rates. Besides, the optimality of the solutions can be guaranteed only in a probabilistic sense. Subdivision or cell mapping techniques, on the other side, are typically computationally expensive. Although, they have a global scope and are well suited for parallelization [15]. Another group encloses the descent direction methods [16, 17, 18, 19, 20, 21] that are generally fast local convergent algorithms focused in finding only one optimal point. Thus, not surprisingly, some of them are hybridized with a set oriented approach in the hope to gather the best of both worlds (see [11, 22, 23, 24, 25, 26]). Finally, the methods based on continuation strategies [27, 28, 29, 20, 30, 31, 32] exploit the fact that, under some mild conditions, the Pareto set/front is typically a smooth object (like a surface or solid), so these procedures can be applied to detect further solutions in the neighborhood of a given optimal point. Literally, the idea here is *to follow* the curve (or manifold[4]) of Pareto points which turns to be a very efficient manner to detect large connected portions of the solution set. These methods are also of local nature, implying that we can get trapped in local optima or miss sections of the Pareto set if it is not connected. Thus, the necessity to combine local schemes with global strategies arises once again in this context. Some examples are given in [33, 34, 35, 36, 22, 23].

We are not, yet, concerned with any hybridization technique. Rather, we prefer to focus first on local continuation methods, where our main motivation—apart of the given outline—is that most of the current designs are either limited to bi-

[3] We refer here to continuous optimization, i.e., all the functions involved (objectives as well as constraints) are continuously differentiable.

[4] We will utilize the term 'manifold' for objects of any dimensions.

objective problems, or they completely exclude second-order information. This exclusion, though apparently convenient, directly attempts against the speed of convergence of the method. The single exception is the approach proposed by Hillermeier in [28]. However, as we will see later on, the efficiency of this method (at least in terms of function evaluations) can be improved as well as the quality of its output (this time in terms of the distribution of solutions in objective space). Besides, the latter procedure was not designed to work with inequality constraints and requires the exact second derivatives of the problem. Even when we already know that this knowledge leads to better convergence rates, many real-world applications may have no access to that kind of data. It is therefore one more challenge to provide a continuation algorithm capable to (efficiently) approximate second-order information. To the best of our knowledge, no such a strategy already exists in the context of multi-objective continuation.[5] On the other hand, descent direction methods are also part of our targets since local searches are widely used by continuation schemes. A Newton method for multi-objective problems has been designed by Fliege et al. in [16] that uses second derivatives. Recently, a quasi-Newton approach was suggested in [17] capable to approximate this information by means of successive BFGS updates. Other gradient-based approaches have been proposed in the literature as described in [18, 19, 20, 21]. However, almost none of these can deal with constraints. The exceptions are the one given in [21] that works exclusively with box constrained problems, and the multi-objective steepest descent method of Fliege and Svaiter [18] which manages general inequalities.

1.2 The Problem

Our primary goal, in this thesis work, is *to trace* the manifold of optimal solutions of a given continuous multi-objective problem. We will consider as well situations that may be subject to equality and/or inequality constraints. More specifically, given an initial efficient point of the problem in hand, the task is to explore the surrounding neighborhood in order to find further solutions. We will *continue* this process in an iterative manner until our set of solutions evolves to a suitable discretization of one connected component of the Pareto set/front. By *suitable discretization* of the solution set, we understand here one that resembles an even distribution of points in objective space. The theory and tools provided by this research are all of local nature, so we will assume that the starting point is sufficiently close or belongs to the set of interest. In addition, we will only consider problems with one connected optimal set. A global approach for our methods is left out of this study to be subject of a future analysis.

[5] We can always take the method of Hillermeier using approximations by finite differences. However, this approach turns to be very inefficient even for medium sized problems.

1.3 Aims of the Thesis

As stated above, we are interested in utilizing continuation techniques to approximate the set of efficient solutions of a problem with multiple goals. Specifically, we aim to design a *predictor corrector* method for the numerical treatment of such multi-objective problems. The theoretical framework of classical predictor corrector strategies for the solution of undetermined systems of equations is given in [37, 38]. Basically, these procedures consist of a succession of two main steps: a predictor step where a movement along the linearized solution set is performed, and a following corrector step concerned with returning that predicted point back to the manifold of solutions. This step, undoubtedly, shall not be required if the Pareto set is linear. The predictor corrector scheme has been applied before to the context of multi-objetive optimization in [27, 28, 29]. However, only the method in [28] can deal with more than two objectives. Two other approaches are proposed as well in [20] and [30] but these works do not make an explicit use of the techniques explained in [37, 38]. Instead, they are based exclusively on the gradient information of the optimization problem, so shall only be seen as first steps toward procedures with better rates of convergence.

The Pareto Tracer, as we call our approach, should be able to deliver an even discretization of the Pareto front. The efficiency of the method will be measured in terms of function evaluations (also for comparison purposes). The justification for this measure is that it is indicative of the costs required to solve problems that are inherently expensive, i.e., the evaluation of the functions involved is very costly. Additionally, a better quality of the solutions is desired such that—as commonly pursued in multi-objective optimization—an evenly spread distribution of the optimal points in objective space is produced. For this, the Pareto Tracer needs to steer the search in parameter space based on a prescribed result in objective space. A property that, on the other hand, most of the continuation methods lack. We will also deal with equality as well as inequality constrained problems. With this purpose, a novel Newton-like descent direction method will be designed to be capable of handling general restrictions. The latter procedure is intended for (but not restricted to) the internal use by the Pareto Tracer during the corrector phase. Moreover, in the absence of second-order information, a succession of clever updates will take place via quasi-Newton methods. This represents a clear advantage over other approaches where the use of exact second derivatives is mandatory[6] or is completely excluded. The theoretical analysis of the convergence rate of our proposals will not be covered in this work. However, we present plenty of numerical results that support our claim that the Pareto Tracer is an affordable alternative for the numerical treatment of multi-objective optimization problems. As pointed out before, the study of the asymptotic time complexity is also omitted, but it is highly recommended for a future analysis in order to better comprehend the strengths and weaknesses of our methods.

[6] We are also including in this statement finite differences approximations.

1.4 Final Contributions

This thesis contributes to the area of continuous multi-objective optimization. Specifically, it introduces a new predictor corrector method for problems with several objectives that we call Pareto Tracer. In addition, a Newton algorithm capable of handling general constraints is proposed and used as corrector by the novel continuation approach. The latter is a method of feasible directions, meaning that the initial point and the subsequent iterations must remain feasible (with respect to the inequalities). Thus, one problem that certainly arises in the context of continuation is that a predicted point cannot be guaranteed to be feasible. Given this inconvenient, we have decided to limit the Pareto Tracer to the special case of equality and box constrained problems. Further, we utilize gradient projections to ensure the feasibility of predictors. The Newton approach, on the other hand, comes together with a modified Armijo condition concerning situations where there is no descent direction that additionally points toward the feasible region (this time regarding the equalities). A step length control that considers this adjustment is essential to suffice those cases where all objectives cannot be improved simultaneously in order to satisfy the equality constraints. Several numerical examples have been presented together with theoretical evidence that help to explain the reliability of our method. However, a proper analysis of convergence and computational complexity is still missing. Furthermore, a superiority of our design regarding the distribution of solutions arises from the steering property: the Pareto Tracer is able to guide the search in parameter space given a desired movement in objective space.

We also cope with two Hessian-free realizations based respectively on gradient descent and quasi-Newton methods. The numerical results indicate that the quasi-Newton variant may be the best Hessian-free choice we could expect. However, the implementation of the steepest descent version turns to be much more computationally efficient. Moreover, we show by several examples that secants (i.e., the line that passes through two consecutive solutions) could be a great approximation of tangent directions (for the bi-objectives case) since the derivatives can be sometimes nearly orthogonal to the Pareto set. The quasi-Newton approach, on the other hand, gets significantly improved if the BFGS updates are applied on predictors as done through the corrector iterations. This trick will provide a better initial guess for the Hessians to the subsequent corrector, and will possibly help to preserve the second-order information gained along all the trajectory.

In conclusion, we consider the Pareto Tracer a highly competitive algorithm for the treatment of multi-objective problems that fulfill the smoothness assumptions required by continuation methods. Furthermore, it could be a powerful tool for a wide range of real-life applications, specially for those where the function evaluations are too costly and resource-intensive. A preliminary study of this work can be found in [39], where the discussion is limited to unconstrained bi-objective problems.

1.5 Organization of the Thesis

The remainder of this paper is organized as follows. In Chapter 2, we present the theoretical background required to understand the ideas developed along this thesis. Additionally, we provide the reader with an outline of the related work proposed by other researchers in order to place our contributions in the proper context. The Pareto Tracer method is introduced and extensively described in Chapter 3 for unconstrained problems. Then, in Chapter 4, the corresponding modifications to deal with equality restrictions are covered in detail. The Newton method for constrained problems is also introduced in this chapter. Chapter 5 is then dedicated to the handling of inequality constraints. Further modifications to the Pareto Tracer with this purpose are illustrated and the Newton method is (once more) extended to deal with inequalities. We finally conclude in Chapter 6 where we also propose some ideas to follow in a future work.

Chapter 2

Background and Related Work

We begin this chapter by a presentation of a brief but essential background on the concepts required to understand the ideas developed in this thesis. Here, we will address two different classes of optimization problems: single-objective optimization problems (SOPs) and multi-objective optimization problems (MOPs). Though the focus of this work is on MOPs, a thorough understanding of the theory and available methods to handle SOPs is crucial to efficiently treat as well problems with more than one objective. In the second part of this chapter we provide the reader with an outline of the related work proposed by other researchers in order to place our contributions in the proper context.

2.1 Theoretical Background

This section presents a theoretical background on the key definitions and principles that constitute the foundations of SOPs and MOPs.

2.1.1 Single-objective Optimization

A general continuous SOP is defined as

$$\begin{aligned} &\min_{x \in \mathbb{R}^n} f(x) \\ &\text{s.t.} \quad g_j(x) \leq 0, \qquad j = 1, \ldots, m, \\ &\qquad\;\; h_j(x) = 0, \qquad j = 1, \ldots, p, \end{aligned} \tag{2.1}$$

where $f : \mathbb{R}^n \to \mathbb{R}$ is called the objective function, $g_j : \mathbb{R}^n \to \mathbb{R}$, $j = 1, \ldots, m$, are the inequality constraints and $h_j : \mathbb{R}^n \to \mathbb{R}$, $j = 1, \ldots, p$, are the equality constraints. We assume along this work that the objective function and the constraints are continuously differentiable. Alternatively, we can write (2.1) as

$$\min_{x \in \mathcal{X}} f(x), \tag{2.2}$$

where $\mathcal{X} = \{x \in \mathbb{R}^n \mid g(x) \leq 0 \text{ and } h(x) = 0\}$ is called the feasible region and $g : \mathbb{R}^n \to \mathbb{R}^m$ and $h : \mathbb{R}^n \to \mathbb{R}^p$ are defined as the vector functions of inequalities and equalities, respectively. At a feasible point x, the inequality constraint g_j, $j \in \{1,\ldots,m\}$ is said to be active if $g_j(x) = 0$, and inactive if the strict inequality $g_j(x) < 0$ is satisfied. Additionally, we define the gradient $\nabla f(x) \in \mathbb{R}^n$ of a multivariable function f as the vector consisting of the function's partial derivatives

$$\nabla f(x) = \begin{pmatrix} \frac{\partial f}{\partial x_1}(x) \\ \vdots \\ \frac{\partial f}{\partial x_n}(x) \end{pmatrix}, \tag{2.3}$$

and the Hessian matrix $\nabla^2 f(x) \in \mathbb{R}^{n\times n}$ as the square matrix of the second-order partial derivatives

$$\nabla^2 f(x) = \begin{pmatrix} \frac{\partial f}{\partial^2 x_1}(x) & \frac{\partial f}{\partial x_1 \partial x_2}(x) & \ldots & \frac{\partial f}{\partial x_1 \partial x_n}(x) \\ \frac{\partial f}{\partial x_2 \partial x_1}(x) & \frac{\partial f}{\partial^2 x_2}(x) & \ldots & \frac{\partial f}{\partial x_2 \partial x_n}(x) \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial f}{\partial x_n \partial x_1}(x) & \frac{\partial f}{\partial x_n \partial x_2}(x) & \ldots & \frac{\partial f}{\partial^2 x_n}(x) \end{pmatrix}. \tag{2.4}$$

Since we are dealing with a single objective, the concept of minimizer becomes intuitive and simple to understand.

Definition 2.1.1

(*a*) *A point x^* is a global minimizer of f if $f(x^*) \leq f(x)$ for all $x \in \mathcal{X}$.*

(*b*) *A point x^* is a local minimizer of f if $f(x^*) \leq f(x)$ is satisfied in a feasible neighborhood of x^*.*

Closely related to the definition of optimality are the first-order necessary conditions or Karush-Kuhn-Tucker (KKT)[1] conditions for SOPs, which are summarized in the following theorem.

Theorem 2.1.1

Suppose that x^* is a local solution of (2.1) *that satisfies some regularity conditions (see below). Then, there exist Lagrange multipliers λ_j, $j = 1,\ldots,p$, and γ_j, $j = 1,\ldots,m$, such that the following conditions are satisfied:*

$$\nabla f(x^*) + \sum_{j=1}^{p} \lambda_j \nabla h_j(x^*) + \sum_{j=1}^{m} \gamma_j \nabla g_j(x^*) = 0, \tag{2.5a}$$

$$h_j(x^*) = 0, \qquad j = 1,\ldots,p, \tag{2.5b}$$

$$g_j(x^*) \leq 0, \qquad j = 1,\ldots,m, \tag{2.5c}$$

$$\gamma_j \geq 0, \qquad j = 1,\ldots,m, \tag{2.5d}$$

$$\gamma_j g_j(x^*) = 0, \qquad j = 1,\ldots,m. \tag{2.5e}$$

[1] Named after the work of Karush [40] and Kuhn and Tucker [41].

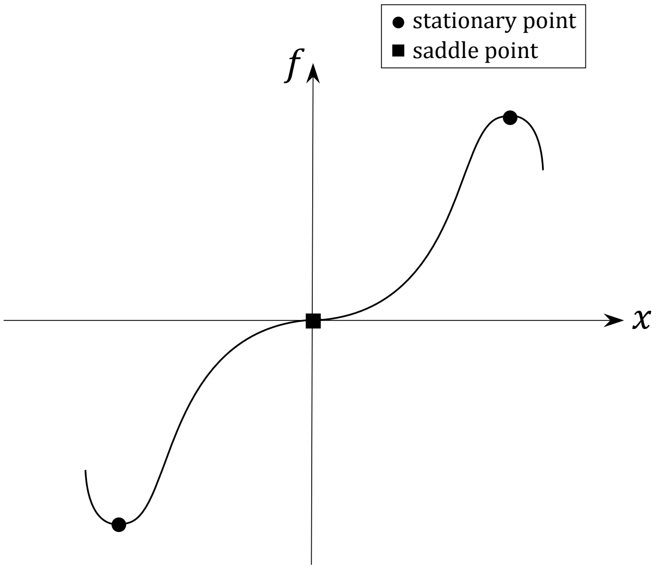


Figure 2.1: Critical points of an univariate function.

One of the most widely used regularity conditions is the linear independence constraint qualification (LICQ), which states that the gradients of the equality constraints and the gradients of the active inequality constraints are linear independent at x^*.

The KKT conditions, as stated above, are only necessary conditions. Hence, the points satisfying these requirements are merely candidates for being a local minimum. For instance, KKT points of unconstrained problems are those where the derivative vanishes (also called critical points) but not all are necessarily optima. Critical points of smooth functions can be stationary: local minima (the Hessian is positive definite) and local maxima (the Hessian is negative definite), or saddle points, i.e., there is a change of curvature (the Hessian is indefinite). See Figure 2.1 for an example. Most optimization algorithms focus only on the detection of KKT points including those considered here, therefore the study of the sufficient conditions for optimality is left out of the scope of this thesis.

Finally, we would like to emphasize a particular case of inequality constraints: those that impose lower and upper bounds on the variables, commonly defined as

$$l_i \leq x_i \leq u_i, \qquad i = 1, \ldots, n, \tag{2.6}$$

or alternatively, we can say that $x \in \mathcal{B}$ where

$$\mathcal{B} = \{x \in \mathbb{R}^n \mid l_i \leq x_i \leq u_i, \quad i = 1, \ldots, n\}. \tag{2.7}$$

If some component of x lacks a lower or an upper bound, we set the appropriate component of l and u to $-\infty$ or $+\infty$, respectively. Considering the interesting properties given by the rectangular shape of the feasible region (sometimes called *box*) and the numerous scientific applications that depend on box constrained optimization, this remains a very active area of research. Ultimately, box restrictions can be expressed in standard form

$$\begin{matrix} -x_i + l_i \leq 0 \\ x_i - u_i \leq 0 \end{matrix}, \qquad i = 1, \ldots, n, \tag{2.8}$$

and handled as general inequalities, but the latter can turn out to be inefficient since $2n$ constraints would be added to the original problem.

2.1.2 Multi-objective Optimization

In the following, we define a general continuous MOP as

$$\begin{aligned} \min_{x \in \mathbb{R}^n} \ & F(x) \\ \text{s.t.} \ & g_j(x) \leq 0, \qquad j = 1, \ldots, m, \\ & h_j(x) = 0, \qquad j = 1, \ldots, p, \end{aligned} \tag{2.9}$$

where $F : \mathbb{R}^n \to \mathbb{R}^k$ is the vector of objective functions $f_1, \ldots, f_k : \mathbb{R}^n \to \mathbb{R}$, $g_j : \mathbb{R}^n \to \mathbb{R}$, $j = 1, \ldots, m$, are the inequality constraints and $h_j : \mathbb{R}^n \to \mathbb{R}$, $j = 1, \ldots, p$, are the equality constraints. We also assume that all objectives and constraints are continuously differentiable. The feasible region in parameter space is defined as for the single-objective case by $\mathcal{X} = \{x \in \mathbb{R}^n \mid g(x) \leq 0 \text{ and } h(x) = 0\}$ where $g : \mathbb{R}^n \to \mathbb{R}^m$ and $h : \mathbb{R}^n \to \mathbb{R}^p$ are defined as the vector functions of inequalities and equalities respectively. Additionally, we will call $\mathcal{F} = F(\mathcal{X})$ the feasible objective region and $\partial\mathcal{F}$ the boundary of $\mathcal{F}$.

Since no single solution would generally minimize every objective simultaneously, the concept of optimality for MOPs is given by the notion of Pareto efficiency [1], sometimes called Pareto dominance.

Definition 2.1.2

(*a*) *A point $y \in \mathbb{R}^n$ is dominated by a point $x \in \mathbb{R}^n$ ($x \prec y$) with respect to F if*

$$F(x) \leq F(y) \text{ and } F(x) \neq F(y),^{2}$$

else y is said to be non-dominated by x ($x \not\prec y$).

(*b*) *A feasible point $x^* \in \mathcal{X}$ is Pareto efficient if $\nexists y \in \mathcal{X}$ such that $y \prec x$.*

(*c*) *A feasible point $x^* \in \mathcal{X}$ is weak Pareto efficient if $\nexists y \in \mathcal{X}$ such that $F(y) < F(x)$.*

(*d*) *A point $x^* \in \mathcal{X}$ is locally (weak) efficient if it is (weak) efficient in a feasible neighborhood of x^*.*

[2] All operators are taken componentwise when convenient.

To be more precise, the Pareto optimality concept employs a partial order induced by $\mathbb{R}^k_+$ (the Paretian cone or non-negative orthant on $\mathbb{R}^k$) defined by

$$F(x) \leq F(y) \Leftrightarrow F(y) - F(x) \in \mathbb{R}^k_+,$$

where $\mathbb{R}_+$ denotes the set of non-negative real numbers. Then, our task is to search for minimal points induced by such partial order. On the other hand, the definition of weak Pareto efficiency is induced by $\mathbb{R}^k_{++}$, the interior of the Paretian cone, i.e.,

$$F(x) < F(y) \Leftrightarrow F(y) - F(x) \in \mathbb{R}^k_{++},$$

where $\mathbb{R}^k_{++}$ denotes the set of strictly positive real numbers. A well-known property of partial orders is that not every pair of elements needs to be related: for some pairs, it may be that neither element dominates the other. Thus, it is natural that the solution set of MOPs does in general not consist of one single solution but rather of an entire set of solutions. Note that, in contrast, the concept of optimality for SOPs is given by the complete order induced by $\mathbb{R}_+$.

Definition 2.1.3

(*a*) *The set of Pareto optimal points* $\mathcal{P} = \{x \in \mathcal{X} \mid \nexists y \in \mathcal{X} : y \prec x\}$ *of* F *is called the Pareto set of* F.

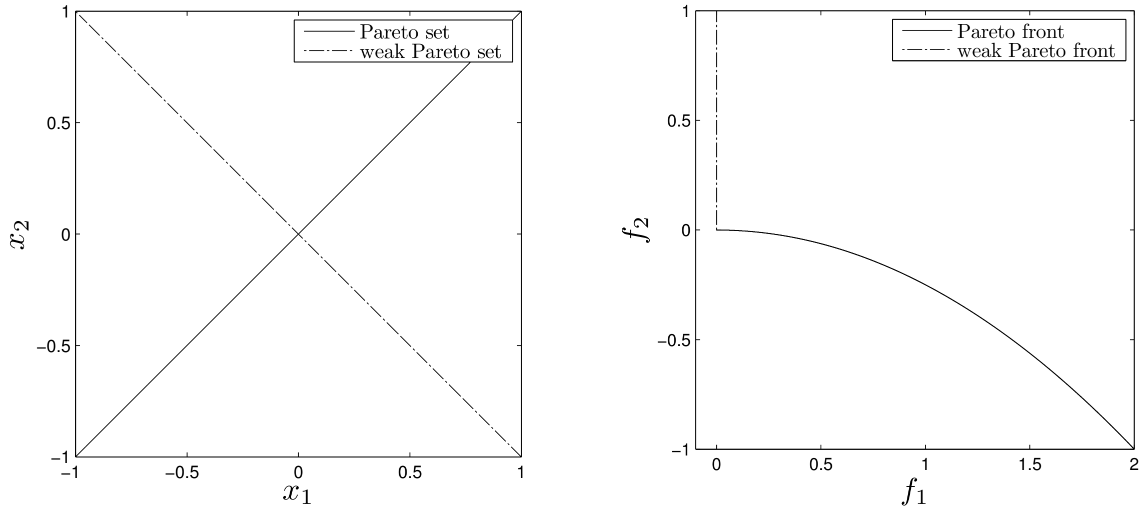
(*b*) *The image* $F(\mathcal{P})$ *of the Pareto set is called the Pareto front of* F *and is denoted by* $\mathcal{F}^*$.

For a better understanding we provide a graphic example. Consider the following MOP where the domain set was taken as $Q = [-1, 1]^2$:

$$\begin{aligned} f_1(x) &= |x_1 + x_2| \\ f_2(x) &= -x_1 x_2. \end{aligned} \tag{2.10}$$

The Pareto optimal set of (2.10) is given by $\mathcal{P} = \{(x_1, x_2) \mid x_1 = x_2\}$ and the weak Pareto set is given by $\mathcal{P}_{\mathcal{W}} = \{(x_1, x_2) \mid x_1 = x_2\} \cup \{(x_1, x_2) \mid x_1 = -x_2\}$ as illustrated in Figure 2.2.

Subsequently, the first-order optimality conditions for MOPs can also be seen as an extension of the KKT conditions for SOPs, for which $k = 1$ and $\alpha = 1$. These are summarized in the following theorem.



(a) decision space (b) objective space

Figure 2.2: Weak and optimal Pareto set and front of the MOP (2.10).

Theorem 2.1.2
Suppose that x^* is a local solution of (2.9) *that satisfies some regularity conditions, e.g., the LICQ condition. Then, there exist Lagrange multipliers α_j, $j = 1, \ldots, k$, λ_j, $j = 1, \ldots, p$, and γ_j, $j = 1, \ldots, m$, such that the following conditions are satisfied*

$$\sum_{j=1}^{k} \alpha_j \nabla f_j(x^*) + \sum_{j=1}^{p} \lambda_j \nabla h_j(x^*) + \sum_{j=1}^{m} \gamma_j \nabla g_j(x^*) = 0, \tag{2.11a}$$

$$h_j(x^*) = 0, \qquad j = 1, \ldots, p, \tag{2.11b}$$

$$g_j(x^*) \leq 0, \qquad j = 1, \ldots, m, \tag{2.11c}$$

$$\alpha_j \geq 0, \qquad j = 1, \ldots, k, \tag{2.11d}$$

$$\textstyle\sum_{j=1}^{k} \alpha_j = 1, \tag{2.11e}$$

$$\gamma_j \geq 0, \qquad j = 1, \ldots, m, \tag{2.11f}$$

$$\gamma_j g_j(x^*) = 0, \qquad j = 1, \ldots, m. \tag{2.11g}$$

This time, in order to characterize critical points for MOPs, we first introduce the definition of a *descent direction*.

Definition 2.1.4
At a point $x \in \mathbb{R}^n$, a descent direction for a function $f : \mathbb{R}^n \to \mathbb{R}$ is defined as a vector $\nu \in \mathbb{R}^n$ such that

$$\nabla f(x)^T \nu < 0. \tag{2.12}$$

Similarly, for a vector function $F : \mathbb{R}^n \to \mathbb{R}^k$ with $F(x) = (f_1(x), \dots, f_k(x))^T \in \mathbb{R}^k$, ν must satisfy

$$\nabla f_j(x)^T \nu < 0, \qquad \forall j = 1, \dots, k, \tag{2.13}$$

meaning that all objectives can be decreased simultaneously.

To see this, take the univariate function $f_\nu : \mathbb{R} \to \mathbb{R}$ such that

$$f_\nu(t) = f(x + t\nu). \tag{2.14}$$

Given that ν is a descent direction, it is that $f'_\nu(0) = \nabla f(x)^T \nu < 0$. Then, for all $t \in (0, \bar{t}]$ where $\bar{t}$ is sufficiently small, we have that $f_\nu(t) < f_\nu(0)$. The extension to the multi-objective case follows immediately from applying the previous reasoning componentwise.

Consequently, critical points for unconstrained MOPs are sometimes defined as those points $\hat{x} \in \mathbb{R}^n$ such that for every vector $\nu \in \mathbb{R}^n$ there is a $j \in \{1, \dots, k\}$ satisfying

$$\nabla f_j(\hat{x})^T \nu \geq 0. \tag{2.15}$$

Thus, in order to improve some of the objectives, at least one must worsen resulting in a non-dominated solution. Still, for more than one objective, critical points are not necessarily optima (note that weak Pareto points are also critical: there is a $j \in \{1, \dots, k\}$ such that $\nabla f_j(\hat{x})^T \nu = 0$). Accordingly, noncritical points are those for which a descent direction exists, therefore, dominating solutions can be found.

Another criteria widely used involves the Jacobian of F, more precisely, $\hat{x}$ is a critical point of F if $\operatorname{rank}(J(\hat{x})) < k$, where the Jacobian of F at an arbitrary point x is defined as

$$J(x) = \begin{pmatrix} \nabla f_1(x)^T \\ \vdots \\ \nabla f_k(x)^T \end{pmatrix} \in \mathbb{R}^{k \times n}. \tag{2.16}$$

Then, noncritical points are those where the Jacobian has full rank. However, this criteria applies for both minimization and maximization problems.

Finally, and perhaps most importantly in the context of our work, is that Pareto candidates (or KKT points) typically form a $(k-1)$-dimensional differentiable manifold [28]. This and subsequent applications will be subject to an in-depth discussion throughout the entire thesis.

2.2 Related Work

In this section we describe some of the strategies that are currently used to solve SOPs and MOPs. We restrict our discussion to mathematical programming (MP) techniques—our area of research—although there are many other approaches that also show excellent results in a great variety of problems.

As we pointed out before, this thesis is mainly concerned with solving MOPs. However, most of the methods currently used for this matter utilize SOPs in one way or another. Thus, the study of this subarea is an excellent starting point for a better comprehension of the techniques used to deal with MOPs.

2.2.1 Single-objective Optimization

SOPs are handled so far from different perspectives. The most traditional one comprises the MP techniques covering a great set of algorithms. We refer the reader to [42, 43, 44] and references therein for a survey on other approaches such as evolutionary algorithms (EAs). MP can be divided into two major groups: unconstrained and constrained optimization. These methods can also be classified according to the nature of the objective function and the constraints, giving rise to different subfields in the area.

We start our discussion by the most generic class of problems: when at least one of the constraints or the objective is a general nonlinear function. This case classifies as a nonlinear programming (NLP) problem, which primarily arise in areas of physics and engineering. One of the most popular techniques of NLP is the Nelder-Mead [45] or downhill simplex method since it fits in situations where the derivatives are unknown. Additionally, the constrained case is normally handled by replacing the problem by one or more unconstrained problems using penalty, barrier, or augmented Lagrangian functions [46]. Of special interest for this work are the steepest descent [46], Newton [46] and quasi-Newton [47] methods, for which we dedicate a separate subsection.

Following, and among the most significant categories, we can mention linear programming (LP), where both, the objective and the constraints, are linear. The simplex [48] and interior point [49] methods are probably the most widely used algorithms in this subfield, having a great application in marketing, operation research and finances. Problems with linear constraints where the objective function is quadratic fall into the category of quadratic programming (QP). These problems are particularly relevant because they arise as subproblems in optimization methods with general constraints, e.g., in sequential quadratic programming methods. Examples of QP techniques include interior point methods for QP [50] and active set methods [46].

When the constraints as well as the objective function are quadratic, we are faced with a quadratically constrained quadratic programming (QCQP) problem. It is of interest to point out that the general case of a QCQP problem is NP-hard, still, they can be treatable by means of reformulation-linearization techniques and semidefinite programming (SP) [51] or second-order cone programming (SOCP) [52] relaxations. A particular case of QCQP is when the objective function is linear, referred to as quadratically constrained linear programming (QCLP). The latter forms an important part of many of the techniques employed along this work and is handled in all cases by means of SOCP reformulations. Therefore, a special subsection is dedicated to this category. This is also the case of nonlinear least squares (NLS) that we later approach in more detail.

Line Search Strategies

There are two basic iterative approaches to find a local minimum of a function: line search strategies and trust region frameworks [46]. We focus here only on the former, which starts by finding a descent direction and then computes a step size that determines the maximal *suitable* movement along the given direction. Thus, a new iteration is defined as

$$\bar{x} = x + t\nu, \tag{2.17}$$

where $t > 0$ is the desired step length, $\nu \in \mathbb{R}^n$ the descent direction, and x the current iteration point. The ideal scenario is to be able to find the minimizer of the univariate function

$$f_\nu(t) = f(x + t\nu). \tag{2.18}$$

Nevertheless, such exact line search procedures may be too costly and time consuming, so it seems to be wise to select an inexact strategy to deliver *acceptable* solutions to our problem. One measure of the quality of our solution can be given by the Wolfe conditions [46], even though, there are other policies. The most intuitive prerequisite for a step size to be acceptable is that it produces *sufficient decrease* in the objective function. This rule is also called the Armijo condition [46] and asserts that

$$f(\bar{x}) \leq f(x) + c_1 t \nabla f(x)^T \nu, \tag{2.19}$$

where $c_1 \in (0, 1)$. A second rule can be introduced in order to reject *too small* step lengths. We may require that t also satisfies

$$\nabla f(\bar{x})^T \nu \geq c_2 \nabla f(x)^T \nu, \tag{2.20}$$

where $c_2 \in (c_1, 1)$. The previous two conditions together are called the Wolfe conditions.

An important result here due to Zoutendijk [53] is that a line search scheme under the Wolfe conditions converges to a critical point provided that some moderate requirements are fulfilled. First, it should be that $|\nabla f(x)^T \nu| \geq \epsilon$ for some $\epsilon > 0$ which indicates that the angle between the derivative and the search direction is bounded away from 90°. Secondly, it is also required that f is continuously differentiable and bounded from below, and ∇f is Lipschitz continuous (see Definition 2.2.1). A procedure like this yields a sequence of candidate solutions x_i with $\lim_{i\to\infty} \nabla f(x_i) = 0$. Thus, the search can be stopped when

$$\|\nabla f(x)\| \leq \varepsilon, \tag{2.21}$$

where $\varepsilon > 0$ is a small tolerance and the operator $\|\cdot\|$ refers to the Euclidean norm[3].

Definition 2.2.1
A function $f : \mathcal{X} \in \mathbb{R}^n \to \mathbb{R}$ is called Lipschitz continuous on $\mathcal{X}$ if there exists $L > 0$ such that

$$\|\nabla f(x) - \nabla f(y)\| \leq L\|x - y\| \quad \forall x, y \in \mathcal{X}, \tag{2.22}$$

where L is called the Lipschitz constant and $\|\cdot\|$ defines any norm. The extension to a multi-objective function $F : \mathcal{X} \in \mathbb{R}^n \to \mathbb{R}^k$ is straightforward.

Further, Equation (2.12) for descent directions defines not only one vector but instead a cone of descent directions. The latter has encouraged the development of several procedures that mainly differ by the selection of the search direction to explore. Thus, in the following, we will cover some of these methods but centering our attention on the selection of the descent direction to use.

Steepest Descent Method

Probably the most basic strategy to solve NLP problems without restrictions is to take steps proportional to the most greedy search direction at each point, i.e., the steepest descent (SD) direction [46]

$$\nu = -\frac{\nabla f(x)}{\|\nabla f(x)\|}. \tag{2.23}$$

However, this method shows severe drawbacks since it requires a great number of iterations for functions that have long narrow valleys. In such cases, the nonlinear conjugate gradient (CG) method [54] is preferable. The latter can reach the solution in at most n iterations for pure quadratic functions but for general cases it may have a slower progress that resembles the behavior of the SD method.

Newton Method

Another approach within the unconstrained NLP category is the Newton method [46], where the search direction is computed by

$$\nu = -\nabla^2 f(x)^{-1} \nabla f(x). \tag{2.24}$$

[3] The operator $\|\cdot\|$ refers to the Euclidean norm unless specified otherwise.

This is a robust and efficient procedure that shows at most quadratic convergence, but unfortunately, at the expense of requiring the Hessians to compute the successive steps. Moreover, the Hessian matrix is not necessarily positive definite in all the domain space, therefore, the Newton direction may be a non-descent direction. Yet, there are several ways to handle this issue as described in [46] which basically consist of a modification of the Hessian to keep it positive definite. In this context, one of the most attractive practices is to use modified Cholesky factorizations as the algorithm proposed by Gill, Murray and Wright [55, 56].

An additional concern when designing practical Newton methods is the computational time since it has a complexity of $\mathcal{O}(n^3)$ where n is the number of variables. To accomplish this goal it is common to use inexact Newton methods, for instance, the Newton-CG also known as truncated Newton [46]. The latter computes the search direction by an adapted version of the CG method that ends as soon as a direction of negative curvature is encountered.

Quasi-Newton Methods

For problems where no Hessian information is provided, best suitable techniques can be used: the quasi-Newton (QN) methods [47]. QN methods build a quadratic model of the problem at each iteration approximating Hessians by calculating the differences between the gradients of the successive steps. These approximations are good enough to produce at most superlinear convergence which makes these procedures an affordable choice on a large variety of problems. QN methods can also be seen from two different perspectives: a line search strategy and a trust region framework, but we will focus only on the former. Here, the search direction is given by

$$\nu = -B^{-1}\nabla f(x), \tag{2.25}$$

where $B \approx \nabla^2 f(x)$. Note that the only difference with the classical Newton method is the approximation of the Hessian by B.

One remarkable point about this theory is that B^{-1} does not need to be calculated at every step. Instead, it may only be updated which reduces the complexity time of each iteration to $\mathcal{O}(n^2)$. Another idea is to update the Cholesky decomposition of B which is widely regarded as more reliable [57]. Based on the type of update, several QN methods can be defined, but in general, all must comply with the secant equation which is

$$\bar{B}s = y, \tag{2.26}$$

where $\bar{B}$ denotes the updated Hessian at the new iteration point $\bar{x}$, $s = \bar{x} - x$ and $y = \nabla f(\bar{x}) - \nabla f(x)$. The secant equation is underdetermined for $n > 1$, thus, it does not have a unique solution. Most updates are based on finding a symmetric solution to (2.26) that minimizes some norm of the difference between $\bar{B}$ and B. Additionally, in the context of a line search, it is advisable to provide positive definite approximations to ensure that the search directions are descent.

The most popular update is the BFGS, named for its discoverers Broyden, Fletcher, Goldfarb, and Shanno, and is given by

$$\bar{B} = B - \frac{Bss^T B}{s^T Bs} + \frac{yy^T}{y^T s}. \tag{2.27}$$

The BFGS update satisfies the secant equation and the symmetry property but it does not ensure the required positive definiteness of the matrices in the context of line searches. Observe that if we premultiply s^T at both sides of (2.26) we obtain

$$s^T \bar{B} s = s^T y > 0, \tag{2.28}$$

a necessary and sufficient condition for the positive definiteness of $\bar{B}$ called the curvature condition. This requirement is satisfied by imposing the Wolfe condition (2.20) on the step size control described earlier.

Nonlinear Least-Squares

The problem of NLS fitting requires the minimization of the squared residuals of k functions:

$$\min_{x \in \mathbb{R}^n} f(x) = \frac{1}{2}\|r(x)\|^2, \tag{2.29}$$

where $r : \mathbb{R}^n \to \mathbb{R}^k$ is the vector of residual functions $r_1, \ldots, r_k : \mathbb{R}^n \to \mathbb{R}$. One effective method to deal with this class of problems is the Gauss-Newton [58] that can be viewed as a modification of the Newton method. We start by defining the first and second derivatives of $f(x)$ as

$$\nabla f(x) = J(x)^T r(x), \quad \text{and} \tag{2.30a}$$

$$\nabla^2 f(x) = J(x)^T J(x) + \sum_{j=1}^{k} r_j(x) \nabla^2 r_j(x), \tag{2.30b}$$

respectively. Here, $J(x)$ denotes the Jacobian of $r(x)$. Then, the second term of $\nabla^2 f(x)$ is neglected and the search direction is computed as

$$\nu = -(J(x)^T J(x))^{-1} J(x) r(x) = -J(x)^{+} r(x), \tag{2.31}$$

where $J(x)^+$ denotes the Moore-Penrose pseudo inverse of $J(x)$. The quadratic convergence of the Newton method is commonly observed assuming that $J(x)$ has full rank, though, it depends to a great extent on how much more significant the first term in (2.30b) is with respect to the second. There are other algorithms as the Levenberg-Marquardt that is certainly more robust and can be viewed as a Gauss-Newton using a trust region approach. The latter was recently modified in [59] to support problems with box constraints.

Second-Order Cone Programming

SOCP [60] is a relatively young area in optimization. The task is to minimize a linear function over the intersection of an affine set and the product of second-order (Lorentz) cones, hence, the problem is defined by

$$\begin{aligned} \min_{x\in\mathbb{R}^n} \; & f^T x \\ \text{s.t.} \; & \|A_j x + b_j\| \leq c_j^T x + d_j, \qquad j = 1, \ldots, k, \\ & Ex + e = 0, \end{aligned} \tag{2.32}$$

where $f, c_j \in \mathbb{R}^n$, $A_j \in \mathbb{R}^{(n_j-1)\times n}$, $b_j \in \mathbb{R}^{n_j-1}$, $d_j \in \mathbb{R}$, $E \in \mathbb{R}^{p\times n}$ and $e \in \mathbb{R}^p$. The inequality constraints in (2.32) are called second-order cone constraints of dimension n_j since

$$\|A_j x + b_j\| \leq c_j^T x + d_j \Leftrightarrow \begin{pmatrix} A_j \\ c_j^T \end{pmatrix} x + \begin{pmatrix} b_j \\ d_j \end{pmatrix} \in K_{n_j},$$

where K_{n_j} is the standard second-order cone of dimension n_j defined as

$$K_{n_j} = \left\{ \begin{pmatrix} u \\ v \end{pmatrix} \mid u \in \mathbb{R}^{n_j-1}, v \in \mathbb{R} \text{ and } \|u\| \leq v \right\}.$$

This area of research has been finding ever wider use as there are several families of problems that can be recast as SOCP, in particular, all its subclasses: LP $\subset$ QP $\subset$ QCQP $\subset$ *SOCP* $\subset$ SP. See, for instance, [61] and references therein. Admitting that it is not a good idea shifting LP and QP to SOCP, it is an accepted practice to reformulate QCQP applications as SOCP to be solved by specialized primal-dual interior point methods. SP reformulations are also used, though more expensive. There are other approaches, e.g., the one proposed in [62] but are not considered in this work. A more detailed explanation will be given throughout the rest of this paper where we will be confronted with several instances of QCLP.

One final consideration is that the dual problem of (2.32) belongs as well to the category of SOCP and is given by

$$\begin{aligned} \max \; & -b^T u - d^T v - e^T y \\ \text{s.t.} \; & A^T u + C^T v + E^T y = f, \\ & \|u_j\| \leq v_j, \qquad j = 1, \ldots, k, \end{aligned} \tag{2.33}$$

with variables $u^T = (u_1^T, \ldots, u_k^T)$, where $u_j \in \mathbb{R}^{n_j-1}$, $v \in \mathbb{R}^k$ and $y \in \mathbb{R}^p$. Further, it is $A^T = (A_1^T, \ldots, A_k^T)$ and $C^T = (c_1, \ldots, c_k)$. Actually, as in LP, the duality in SOCP comes in a symmetric sense, meaning that the dual of the dual is the primal.

2.2.2 Multi-objective Optimization

We finally turn our attention to many of the strategies designed so far for the numerical treatment of MOPs. As stated earlier, we are seeking for a set of solutions, not a single point, so multi-objective optimization programs certainly have to deal with two goals: convergence toward the Pareto set as well as a good distribution of the solutions along this set. Hence, it should come as no surprise that multicriteria tradeoffs emerge once again in our designs. To overcome this compromise between spread and proximity, various quality indicators have been proposed over the past few years. These are commonly used to guide the search of a multi-objective evolutionary algorithm (MOEA) although have also application as performance measures for comparative studies. A good survey on this topic is provided in [63].

In order to shortly review the state-of-the-art methods and best practices in multi-objective optimization, we focus our discussion (as for the single-objective case) on multi-objective mathematical programming (MOMP) techniques. However, we shall call attention to a prominent class of algorithms that has made a big impact on the scientific community: the MOEAs, being quite successful in a great number of applications of the real world as wireless sensor networks, resource allocation, data mining, bioinformatics, among others [8, 9, 10]. These are population based methods capable to deliver an approximation of the solution set in one run of the algorithm. The latter, together with their global approach and a derivative-free implementation, are the main reasons of the MOEAs' popularity. Nevertheless, one major drawback are their relatively slow convergence rates and the fact that the optimality of the solutions cannot be guaranteed (it is only known that the generated solutions are non-dominated by any other).

As a remedy, the hybridization of MOEAs and MOMP has gained increasing popularity since they aim to take the best of both worlds. These techniques, referred to as memetic algorithms (MAs) [26, 64], attempt to exploit the global mechanisms of MOEAs and the power of MOMP methods as local searchers in order to get fast and reliable optimization procedures. One weakness of MAs is that MOMP techniques usually require gradient or even Hessian information at each candidate solution which we don't have at hand in a great variety of problems. On the other side, approximations of gradients, e.g., by finite differences (FD), increase in a big amount the computational cost of the process since they take a lot of additional function evaluations. For instance, forward FD requires n additional function evaluations to approximate one derivative, where n is the number of variables of the problem. Therefore, the development of derivative-free MP techniques has gained the interest of many researchers. In spite of most progress has been made for SOPs, some gradient-free memetic approaches has recently been proposed for MOPs [65, 22, 66].

Another group of well-known set oriented methods are subdivision techniques [11, 12] or cell mapping techniques [13, 14] that were initially proposed to determine the global behavior of nonlinear dynamical systems and later extended to deal with MOPs. The former is based on strategies to successively partition the space into boxes that may be discarded by some optimality cut-off test, while the main idea of the latter consists of a discretization of the space into hypercubes inspired by the finite precision arithmetic of the machine. Admitting that this set of methods can be computationally expensive, they have a global scope and are well suited for parallelization techniques [15].

From now on and for a better comprehension and clarity, we dedicate a separate subsection to three of the main subcategories of MOMP: scalarization methods, descent direction methods and finally (and the center of our attention) continuation methods.

Scalarization Methods

One of the most basic and followed philosophies to handle MOPs is to transform the original problem into a single-objective one that can be solved by means of standard optimization techniques. There are different strategies of reformulations [2, 3, 5] but the key idea is to require that an optimal solution of the new SOP is also a Pareto optimal solution of the original MOP. Further, a selection of a clever sequence of these SOPs will also be necessary such that a suitable finite size approximation of the entire Pareto set is obtained.

Two classic scalarization techniques are the weighted sum (WS) [67, 4] and the ϵ-constraint [68, 69] methods. The first one is based on the following subproblem

$$\min_{x\in\mathbb{R}^n} \sum_{j=1}^{k} w_j f_j(x), \tag{2.34}$$

where a sufficient condition for a solution of (2.34) to be a Pareto point is the selection of positive weights. It is also common to normalize the coefficients of the subproblem such that $\sum_{j=1}^{k} w_j = 1$. The WS scheme is simple and easy to implement, but also has clear disadvantages. First of all, the question arises of what sequence of weights is more appropriate taking into account that a uniform distribution of weights does not guarantee a uniformly distributed set of Pareto optimal solutions, and secondly—but not less important—is that only solutions in the convex subset of the Pareto front are to be found [70, 71].

Following is the ϵ-constraint method [68, 69] that overcomes some of the convexity problems of the WS strategy. Here, the proposed SOP to be solved is defined by

$$\begin{aligned} &\min_{x\in\mathbb{R}^n} f_p(x) \\ &\text{s.t.} \quad f_j(x) \leq \epsilon_j, \qquad j \in \{1,\ldots,k\} \setminus \{p\}, \end{aligned} \tag{2.35}$$

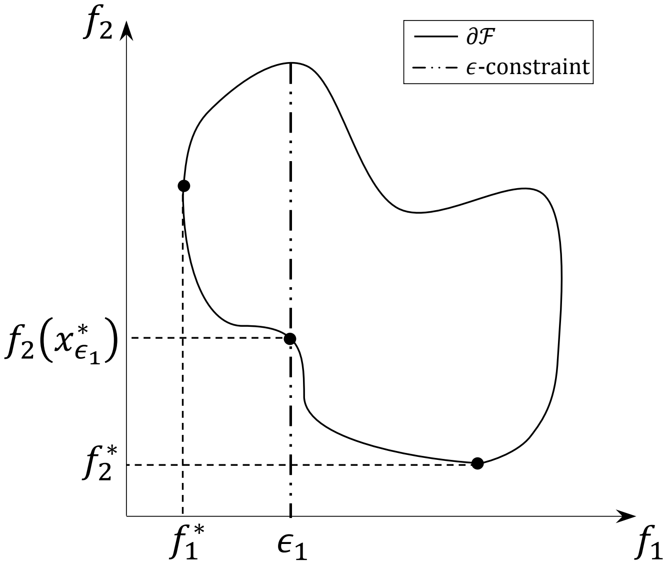


Figure 2.3: $x^*_{\epsilon_1}$ is the minimum found by the ϵ-constraint method where the primary objective is f_2 and f_1 is bounded above by ϵ_1. f_1^* and f_2^* are the minimum values of f_1 and f_2, respectively.

where f_p is regarded as the primary objective. See Figure 2.3 for a bi-objective example. This time, a solution x^* of the subproblem (2.35) is a Pareto optimal point if and only if it is also a solution of the subproblem for each value of $p \in \{1, \ldots, k\}$ such that $f_j(x^*) = \epsilon_j$ for all $j \neq p$. An advantage over the WS approach is the possibility to reach the entire set of Pareto efficient solutions, but again, how to choose the ϵ-values is not an easy task and remains under investigation.

Later, the Physical Programming (PP) design was proposed in [72] which looks like a more reliable choice [73]. PP does not involve the use of weights but requires the designer to divide the objectives into regions or intervals to be assigned to one of the following categories: ‘unacceptable’, ‘highly undesirable’, ‘undesirable’, ‘tolerable’, ‘desirable’, and ‘highly desirable’. Each criterion is also associated with a class-function: ‘smaller is better’, ‘larger is better’, and ‘center is better’. These categories and class-functions are then used to define an aggregate function that attempts to comply with the user’s preferences.

In the following, we explain in more detail two other scalarization techniques relevant to our work since they better follow the spirit of continuation-type methods.

Normal Boundary Intersection (NBI) The NBI method [6] is a scalarization strategy specially designed for the generation of uniformly distributed optimal solutions in the objective space. The first step is to construct what the authors call the convex hull of individual minima (CHIM): given an even distribution of vectors in the unit k-dimensional hypercube, define

$$\text{CHIM} = \left\{ \Phi w \mid \sum_{j=1}^{k} w_j = 1, \quad w_j \geq 0, \quad j = 1, \ldots, k \right\}, \tag{2.36}$$

where the matrix $\Phi \in \mathbb{R}^{k \times k}$ is usually called the pay-off matrix and is given by

$$\Phi = \left(F_1^* - F^*, \ldots, F_k^* - F^* \right). \tag{2.37}$$

Here, $F_j^* = F(x_j^*)$ and x_j^* represents the individual global minima of the j-th objective. Additionally, the vector $F^* = (f_1^*, \ldots, f_k^*)^T$ is referred to as the shadow minimum where $f_j^* = f_j(x_j^*)$ for $j = 1, \ldots, k$.

The main idea of the method relies on the observation that the intersection between the boundary of the feasible objective region and the normal to any point in the CHIM is probably a Pareto optimal point. Then, the original MOP is replaced by a sequence of *NBI subproblems* (each defined by a point in the CHIM) as follows

$$\begin{array}{ll} \displaystyle\max_{(x,t) \in \mathbb{R}^n \times \mathbb{R}} & t \\ \text{s.t.} & \Phi w + t\hat{n} = F(x),^{4} \\ & \quad g_j(x) \leq 0, \qquad j = 1, \ldots, m, \\ & \quad h_j(x) = 0, \qquad j = 1, \ldots, p, \\ & \quad l \leq x \leq u. \end{array} \tag{2.38}$$

If we do not have the exact normal to the CHIM simplex, we can compute a quasi-normal direction $\hat{n}$ given by

$$\hat{n} = -\Phi e, \tag{2.39}$$

where e is the vector of all ones. Figure 2.4 provides a more comprehensible description of the scheme adopted by the NBI algorithm.

Among the advantages of this approach it is highlighted that the method is capable of producing an evenly spaced set of Pareto optimal points without requiring exact Hessians, instead, secant approximations can be used as BFGS updates. However, the algorithm depends on a uniform distribution of the input parameters (weights) which is a non-trivial task [74, 75], augmented with the fact that an efficient computation of the NBI subproblems hardly depends on the selection of the initial guesses. The latter is softened by defining an ordering of the weights such that one can use the previously computed solution as the starting point for the current computations, which gives the NBI the nature of continuation-type methods. However, establishing such an order becomes seriously complicated for problems with more than two objectives. Additionally, and as a consequence of reducing the entire feasible objective region to a simplex, the NBI may leave out some areas of the front located outside the shadow projected by the CHIM in the direction of the normal. See Figure 2.5 for such an example.

[4] Actually $F(x)$ is replaced by $F(x) - F^*$ such that all objective functions are non-negative.

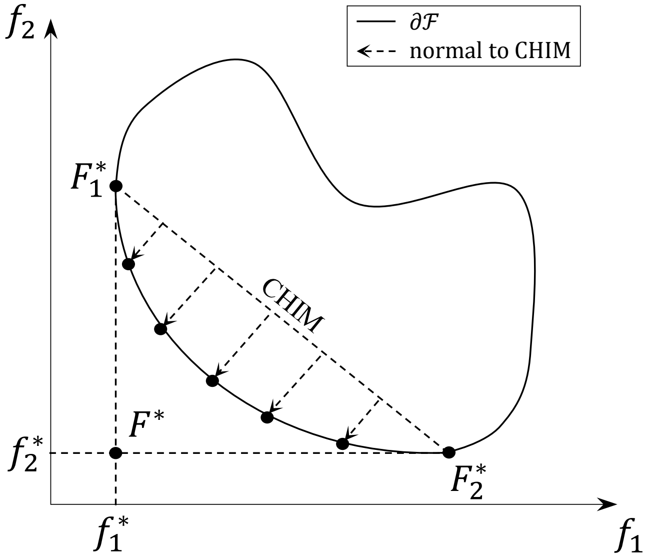


Figure 2.4: Scheme followed by the NBI. Each intersection of a normal with the boundary may define a new solution.

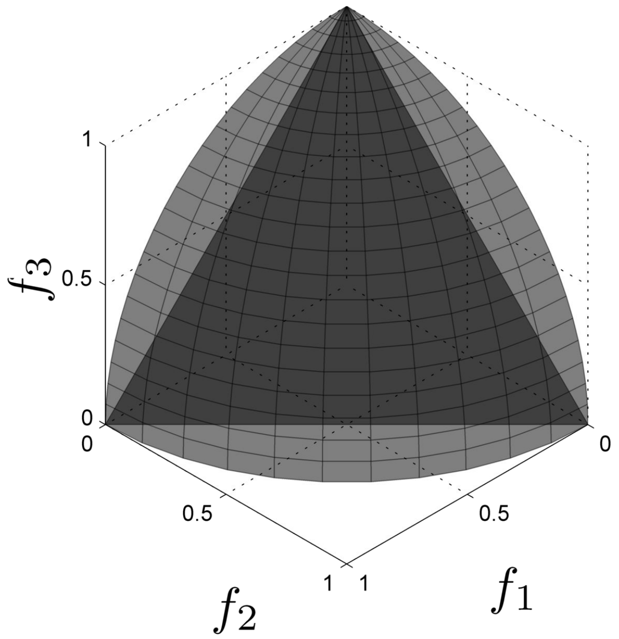


Figure 2.5: Unobtainable Pareto front sections under the NBI scheme. The dark triangle represents the CHIM and the Pareto front is a quarter of a sphere (lighter surface). It is easy to see that a projection of the triangle onto the sphere won't necessarily cover all the border sections.

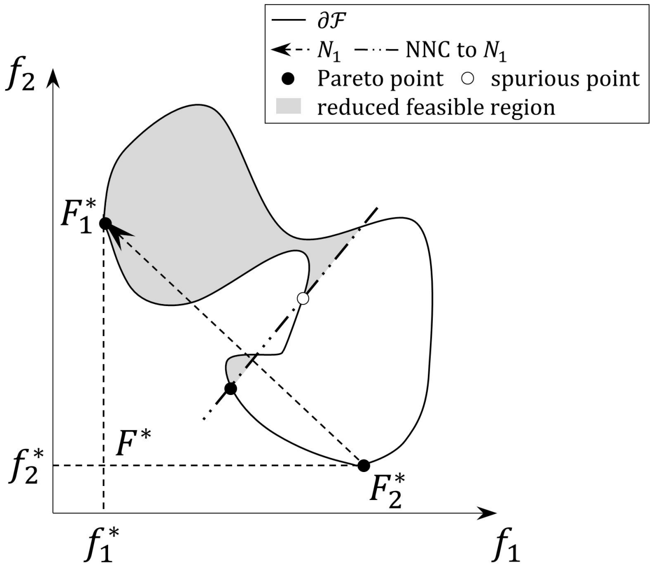


Figure 2.6: Scheme followed by the NNC method. Each intersection of a NNC to N_1 with the boundary may define a new solution. Note that the spurious point is not even local optimum.

Normalized Normal Constraint (NNC) The NNC method [7] was proposed with the aim of improving some of the limitations of the NBI design. Here, the CHIM is renamed to utopia line or hyperplane but for the sake of clarity we keep the notation Φ. Again, an evenly spaced discretization of the utopia hyperplane is ensured by the use of uniformly distributed convex weights, and accordingly, one subproblem is solved for each hyperplane point:

$$\begin{aligned} \min_{x\in\mathbb{R}^n} \quad & f_k(x) \\ \text{s.t.} \quad & N_j(F(x)-\Phi w) \leq 0, \qquad j=1,\ldots,k-1, \\ & g_j(x) \leq 0, \qquad j=1,\ldots,m, \\ & h_j(x) = 0, \qquad j=1,\ldots,p, \\ & l \leq x \leq u, \end{aligned} \tag{2.40}$$

where

$$N_j = F_k^* - F_j^*. \tag{2.41}$$

The main modification is given by the exchange of the NBI equality constraint by $(k-1)$ inequalities that progressively reduce the feasible domain. See Figure 2.6 for a graphical description. The term *normalized* is given due to the following normalization scheme

$$f_j(x) \leftarrow \frac{f_j(x)-f_j^*}{f_j^n - f_j^*}, \qquad j=1,\ldots,k, \tag{2.42}$$

where

$$f_j^n = \max\left(f_j(x_1^*),\ldots,f_j(x_k^*)\right), \qquad j=1,\ldots,k. \tag{2.43}$$

The point $F^n = (f_1^n, \dots, f_k^n)^T \in \mathbb{R}^k$ is usually called the nadir point.

Note by Figure 2.6 that the NNC algorithm suffers from the same limitation of the NBI approach: a subproblem optimal point is not necessarily Pareto efficient. Nevertheless, the use of inequality constraints seems to be less likely to generate non Pareto points over the use of the NBI equality since the latter forces the solution to lie on the normal. Moreover, the NBI method works with a family of quasi-normals while the NNC scheme defines true normals to the utopia hyperplane which possibly leads to a better distribution of the obtained solutions. In general, due to the similarities between both strategies, these methods share several features (advantages and disadvantages) that won't be listed here in detail, but we would like to mention a suggested procedure in [76] to overcome the drawback of missing sections of the front. The proposal consists of extending the utopia hyperplane (by allowing the weights to fall beyond zero and one) such that the entire Pareto front can be reached. In spite of additional optimization problems have to be solved, this and the introduction of a *Pareto filter* to exclude dominated points from the solution set, constitute a great help to improve the results of this excellent optimization method.

In general, there have been other attempts to enhance the last two algorithms, but we will simply close this subsection by recommending the reading of one in particular: the Directed Search Domain (DSD) method [77, 78]. This algorithm was developed with the intention to combine the advantages of PP, NBI and NNC procedures. Additionally, it introduces the idea of shrinking the search domain in order to partially reduce the tendency of the above-mentioned methods to compute redundant solutions.

Descent Direction Methods

Now it is the turn to discuss a different class of algorithms that focuses only on finding one optimal point. These methods do not use any ordering or scalarization of the objective functions, rather, a line search is performed in a direction that decreases all objectives simultaneously, i.e., a descent direction (see Definition 2.1.4). Thus, in order to solve the problem, this kind of procedures follows a curve of dominated points until no further improvement can be achieved. Line search strategies for MOPs follow the same plan described in Section 2.2.1 for SOPs. The difference here is that almost all methods have its own strategy not only for the selection of the search direction but also for the step length control and the stopping criteria. Thus, we discuss each particularity at the proper time. There are also developments for MOPs based on trust region frameworks (see [79, 80]) but we leave this topic out since our main interest here is on line search strategies.

Since we are now dealing with MOPs, an interesting aspect worth discussing in some detail is about how to use descent directions to obtain an approximation of the entire solution set. In [16, 17, 21, 81] the authors solve several instances of their respective programs for different starting points equally spatially distributed or randomly uniformly distributed in the parameter space. The experiments show that the choice of a large number of such initial points seems to give satisfactory approximations of the whole local Pareto fronts from a heuristic point of view. Nevertheless, a substantial number of redundant solutions are expected in this type of practice which is definitely problematic when the efficiency has priority. Furthermore, even an equally spaced distribution of points in the parameter space does not guarantee an equally distributed discretization of the front. For this purpose, it seems more promising to combine descent direction methods with MOEAs—to design new MAs—as treated for instance in [25, 24, 22] and references therein. Hopefully, the MOEA should provide a well distributed set of solutions while the local search will help to improve the convergence toward the efficient set. Another approach is to use descent directions inside a global subdivision technique [11] where the underlying idea is to write down an iteration scheme which—interpreted as a discrete dynamical system—possesses the Pareto set as an attractor. Continuation methods are also good candidates to discover connected portions of the solution manifold by utilizing procedures of this class, but we cover this area more seriously in the next subsection.

In the sequel, some popular descent direction methods are described with the hope of highlighting the main similarities, differences and distinctive characteristics of developments in this subfield.

A Stochastic Descent Direction Method One of the first approaches suggested in this category is the one proposed by Schäffler, Schultz and Weinzierl in [19]. This method develops from the KKT equations for unconstrained optimization. Thus, the search direction here is defined by

$$q(x) = -\sum_{j=1}^{k} \alpha_j \nabla f_j(x) = -J(x)^T \alpha, \tag{2.44}$$

where $\alpha \in \mathbb{R}^k$ is the solution of the following norm minimization problem

$$\begin{array}{ll} \displaystyle\min_{\alpha \in \mathbb{R}^k} & \| \sum_{j=1}^{k} \alpha_j \nabla f_j(x) \|^2 \\ \text{s.t.} & \sum_{j=1}^{k} \alpha_j = 1, \\ & \qquad \alpha_j \geq 0, \qquad j = 1, \dots, k. \end{array} \tag{2.45}$$

If x is a KKT point, $q(x)$ is zero and the components of α coincide with the Lagrange multipliers, otherwise $q(x)$ is a descent direction at x of the considered MOP. Thus, the method can stop when the norm of $q(x)$ is under a given tolerance. These results are used later to state that the unique solution of the following initial value problem (IVP) is a curve consisting of dominated points:

$$\begin{array}{ll} x(0) = x_0, & t = 0, \\ \dot{x}(t) = q(x(t)), & t > 0. \end{array} \tag{2.46}$$

In order to solve (2.46) numerically, the authors propose to use the Euler method [82], a standard approach in the numerical analysis of ordinary differential equations (ODEs). For the step size control, they use a simple backtracking strategy in which the step length t is divided by two in case the next iteration point is dominated by the current point.

An additional consideration here is that this method was not originally applied in its deterministic form. Instead, a stochastic differential equation (SDE) is derived from (2.46) for which typical solutions stay close to the Pareto set for a relatively long period of time. For a throughout discussion on SDEs, see for instance [83, 84] and references therein. The numerical treatment of this SDE leads to a heuristic algorithm for the computation of a large number of Pareto optimal solutions. In contrast, the deterministic version of the method was later integrated into a global subdivision technique [11] where the dynamical system iterations are obtained via suitable discretizations of the ODE given by (2.46).

The main advantage of the considered method is the simplicity of its idea followed by efficient computations: the norm minimization problem has only k variables and $k+1$ constraints and the descent direction merely requires a matrix-vector product. The linear convergence rate, however, is possibly its main drawback which may lead to a huge number of function evaluations for more complicated MOPs.

The Directed Search Descent Method The Directed Search (DS) method [20] is an iterative procedure based on the idea of steering the search along a predefined direction in objective space. That is, given a point $x \in \mathbb{R}^n$ in parameter space and a direction $d \in \mathbb{R}^k$ in objective space, a direction vector $\nu \in \mathbb{R}^n$ is sought such that

$$\lim_{t \searrow 0} \frac{f_j(x+t\nu) - f_j(x)}{t} = \langle \nabla f_j(x), \nu \rangle = d_j, \qquad j = 1, \ldots, k. \tag{2.47}$$

With this, the relative change of each objective value for an infinitesimal step size is given by the components of d. In matrix notation, we have for Equation (2.47)

$$J\nu = d,^{5} \tag{2.48}$$

whose solution is a descent direction provided that all components of d are negative. Typically for MOPs, the number of parameters is higher than the number of objectives ($n \gg k$) so we can assume that the linear system (2.48) is (highly) underdetermined implying that its solution is not unique. One choice is to take

$$\nu_+ = J^+ d, \tag{2.49}$$

which is the solution of (2.48) (if $\operatorname{rank}(J) = k$) with the smallest Euclidean norm. This alternative has the advantage that it leads to the largest decay in direction d

[5] It can be taken $J = J(x)$ for the Jacobian matrix and $\nu = \nu(x)$ for any descent direction vector when convenient.

(in objective space) for a line search with a *small* step in direction ν_+ (in parameter space). Further, the trajectory followed by this procedure is identical to the solution curve of the subsequent IVP

$$\begin{aligned} x(0) &= x_0, & t &= 0, \\ \dot{x}(t) &= \nu_+(x(t)), & t &> 0, \end{aligned} \tag{2.50}$$

whose limit point x^* is also a critical point of the considered MOP. The step length control is similar to the one previously described, but this time the method stops when $\operatorname{rank}(J(x)) < k$. The latter is relaxed by using the condition number of the Jacobian to stop when

$$\kappa_2 = \sqrt{\frac{\lambda_{\max}(JJ^T)}{\lambda_{\min}(JJ^T)}} \geq \varepsilon, \tag{2.51}$$

where $\lambda_{\max}(A)$ and $\lambda_{\min}(A)$ are the largest and smallest eigenvalues of a matrix $A \in \mathbb{R}^{k \times k}$, respectively. If J is rank deficient, it holds that $\lambda_{\min}(JJ^T) = 0$. Then, we may expect that $\kappa_2 \to \infty$ as $x(t) \to x^*$ so ε is taken as a very large number.

One advantage of the DS is that it can be realized without gradient information when used as a local searcher in a MA [23]. In this context, the computation of the search direction may be obtained for free if neighborhood information is available, i.e., if there are points of the population containing other points inside certain neighborhood radius. Furthermore, the method is also coupled into a continuation strategy that is covered later in this chapter.

A Descent Direction Method for Handling Box Constraints The following method [21] looks for Pareto optimal points of box constrained MOPs as limit points of the trajectory solutions of suitable IVPs for ODEs. The proposed IVP reads as

$$\begin{aligned} x(0) &= x_0, & t &= 0, \\ \dot{x}(t) &= h(x(t)), & t &> 0, \end{aligned} \tag{2.52}$$

where the vector $h(x)$ is described as a kind of steepest descent direction and is given by

$$h(x) = -\mathcal{D}_l(x)^2 \mathcal{D}_u(x)^2 J(x)^T d(x). \tag{2.53}$$

The auxiliary vector $d(x) \in \mathbb{R}^k$ is computed by means of

$$d(x) = \left(J(x)\mathcal{D}_l(x)^2 \mathcal{D}_u(x)^2 J(x)^T\right)^* e, \tag{2.54}$$

where e is the vector of all ones and A^* denotes the adjoint matrix of $A \in \mathbb{R}^{k \times k}$.[6] Finally, the diagonal scaling matrices are defined as follows

[6] $AA^* = A^*A = \det(A)I$, where I denotes the identity matrix.

$$\left(\mathcal{D}_l(x)^\alpha\right)_{ij} = \begin{cases} (x_i - l_i)^\alpha, & i = j \\ 0, & i \neq j \end{cases}, \qquad i, j \in \{1, \ldots, n\}, \tag{2.55a}$$

$$\left(\mathcal{D}_u(x)^\beta\right)_{ij} = \begin{cases} (u_i - x_i)^\beta, & i = j \\ 0, & i \neq j \end{cases}, \qquad i, j \in \{1, \ldots, n\}, \tag{2.55b}$$

for some $\alpha, \beta > 0$.

Here, a point $\hat{x} \in \mathcal{B}$ is said to be a critical point of the box constrained problem if the matrix $\left(J(\hat{x})\mathcal{D}_l(\hat{x})^2\mathcal{D}_u(\hat{x})^2J(\hat{x})^T\right) \in \mathbb{R}^k$ is singular. Note that for points strictly inside the box, the last condition holds if and only if the Jacobian matrix is rank-deficient. Thus, a possible stop condition could be to check whether the determinant of the above matrix falls under a small threshold. On the other hand, the step length is chosen by a bisection strategy to ensure that the next iteration point is dominant and remains inside the box.

So far, we are not aware of any integration of this method into an EA or continuation strategy. Alternatively, one can run multiple instances of the program starting at different points with the aim of obtaining several solutions. An improved version of this approach is given in [81].

Newton, Quasi-Newton and Steepest Descent Methods This space is dedicated to the multi-objective versions of three well-known algorithms for SOPs—the Newton, quasi-Newton (QN) and steepest descent (SD) methods—since they share many ideas that can be unified in a single theory.

We start with the Newton method [16] which seems to be the best choice when we have the exact Hessians of all functions available. The core of the method is to solve—at each iteration of a line search—one minimization subproblem to obtain the direction to follow. The indicated subproblem reads as

$$\begin{aligned} \min_{(\nu,\delta)\in\mathbb{R}^n\times\mathbb{R}} \quad & \delta \\ \text{s.t.} \quad & \nabla f_j(x)^T\nu + \tfrac{1}{2}\nu^T\nabla^2 f_j(x)\nu \leq \delta, \qquad j = 1, \ldots, k, \end{aligned} \tag{2.56}$$

where $\nu \in \mathbb{R}^n$ stands for the search direction. Note that the left hand side of each constraint in (2.56) is a quadratic approximation of each function decrease, so basically we are trying to reduce all objectives as much as possible where δ is a measure of the expected decrease. Assuming that all Hessians are positive definite, the subproblem (2.56) is convex and has a unique minimizer, which is a descent direction provided that x is noncritical. A cheap stopping criterion for the method is supported by Proposition 2.2.1.

Proposition 2.2.1
For each $x \in \mathbb{R}^n$ the following statements are equivalent:

(a) *the point x is noncritical,*

(b) $\nu \neq 0$, *and*

(c) $\delta < 0$,

where ν and δ are respectively the solution and the minimum function value of the direction subproblem (2.56).

Then, for zero values of δ, ν is also zero and x is a critical point. Thus, we can stop when

$$\delta \geq -\varepsilon, \tag{2.57}$$

where $\varepsilon > 0$ is a small tolerance (generally taken as $\varepsilon = 5\sqrt{\text{eps}}$). A suggested idea in [16] for the step length control is to take a $t > 0$ such that it satisfies the Armijo condition [46] taken componentwise

$$F(x + t\nu) \leq F(x) + ct\delta e, \tag{2.58}$$

where $e \in \mathbb{R}^k$ is the vector of all ones and $0 < c < 1$ is a constant (generally taken as 0.1). The second term in the right hand side of (2.58) is inferred taking into account that

$$\delta = \max_{j=1,\dots,k} \nabla f_j(x)^T \nu + \frac{1}{2}\nu^T \nabla^2 f_j(x)\nu, \tag{2.59}$$

and by the positive definiteness of the Hessians it follows that

$$\nabla f_j(x)^T \nu \leq \delta, \qquad \forall j = 1, \dots, k. \tag{2.60}$$

Putting all together and under suitable local assumptions, full Newton steps are always accepted and the generated sequence converges superlinear (or at most quadratically)[7] to a local solution. Last, in order to see the relation to the single-objective version of the method, we utilize the Lagrangian of the subproblem (2.56) that reads as

$$\mathcal{L}((\nu, \delta), \alpha) = \delta + \sum_{j=1}^{k} \alpha_j \left(\nabla f_j(x)^T \nu + \frac{1}{2}\nu^T \nabla^2 f_j(x)\nu - \delta \right) \tag{2.61}$$

with derivatives

$$\nabla_\nu \mathcal{L} = \sum_{j=1}^{k} \alpha_j \left(\nabla f_j(x) + \nabla^2 f_j(x)\nu \right) = 0, \quad \text{and} \tag{2.62a}$$

$$\nabla_\delta \mathcal{L} = 1 - \sum_{j=1}^{k} \alpha_j = 0. \tag{2.62b}$$

[7] Additionally, for quadratic convergence, all Hessians must be Lipschitz continuous.

In particular, from (2.62a) we obtain

$$\nu = -\left(\sum_{j=1}^{k} \alpha_j \nabla^2 f_j(x)\right)^{-1} \sum_{j=1}^{k} \alpha_j \nabla f_j(x) \tag{2.63}$$

which coincides with the Newton direction for the SOP resulting from weighting the objectives by the a-priori unknown Lagrange multipliers. Also see that in case we are dealing with a single-criterion, $\alpha = 1$ and ν matches the Newton direction for SOPs (2.24).

A clear limitation of this approach is the requirement of the exact Hessians of all objectives. Although, if we don't have second-order information available, we can choose between the QN [17] and the SD [18] methods for MOPs. The first one looks very similar to the Newton approach. At each iteration it solves

$$\begin{aligned} \min_{(\nu,\delta)\in\mathbb{R}^n\times\mathbb{R}} \quad & \delta \\ \text{s.t.} \quad & \nabla f_j(x)^T \nu + \tfrac{1}{2}\nu^T B_j(x)\nu \leq \delta, \qquad j = 1,\dots,k, \end{aligned} \tag{2.64}$$

that differs only from the Newton direction subproblem in the approximation of each Hessian by a BFGS update (2.27). However, the trouble here is that the uniqueness of the solution depends on the positive definiteness of the Hessians. One way out for SOPs is to impose the Wolfe conditions on the step length control (defined by the equations (2.19) and (2.20)), but these requirements may be too hard (not to say impossible) to satisfy for many objectives simultaneously. Thus, as for the Newton method, it is assumed that the problem is strictly convex. It is also worth mentioning that some other advantages of the QN framework are not (yet) available for MOPs. This is the case of the reduction of the computational complexity of the iterations: for SOPs there is a reduction from $\mathcal{O}(n^3)$ to $\mathcal{O}(n^2)$ while for MOPs the complexity of the direction subproblem is not simplified. Still, another similarity to the single-objective version arises: a Hessian-free approach comes with the cost of a convergence downgrade from quadratic to superlinear[8].

On the other hand, we can completely disregard the second-order information of the problem and use the SD [18] method. This time the subproblem to solve for a descent direction is given by

$$\begin{aligned} \min_{(\nu,\delta)\in\mathbb{R}^n\times\mathbb{R}} \quad & \tfrac{1}{2}\|\nu\|^2 + \delta \\ \text{s.t.} \quad & \nabla f_j(x)^T \nu \leq \delta, \qquad j = 1,\dots,k, \end{aligned} \tag{2.65}$$

where the term $\frac{1}{2}\|\nu\|^2$ is added to the objective function in order to keep the problem bounded. With this, the convergence rate is reduced to linear but the computations become more efficient. Note that the SD direction subproblem belongs to the category

[8] Assuming that the derivatives are Lipschitz continuous.

of QP while the Newton and QN respective subproblems fit into QCLP which is a more general class. Finally, the Lagrangian of (2.65) reads as

$$\mathcal{L}((\nu,\delta),\alpha) = \frac{1}{2}\|\nu\|^2 + \delta + \sum_{j=1}^{k} \alpha_j \left(\nabla f_j(x)^T \nu - \delta\right) \tag{2.66}$$

with derivatives

$$\nabla_\nu \mathcal{L} = \nu + \sum_{j=1}^{k} \alpha_j \nabla f_j(x) = 0, \quad \text{and} \tag{2.67a}$$

$$\nabla_\delta \mathcal{L} = 1 - \sum_{j=1}^{k} \alpha_j = 0, \tag{2.67b}$$

from which we obtain

$$\nu = -J^T \alpha. \tag{2.68}$$

Further, the dual problem of (2.65) is given by

$$\begin{array}{ll} \max\limits_{\alpha \in \mathbb{R}^k} & -\frac{1}{2}\alpha^T J J^T \alpha \\ \text{s.t.} & \sum_{j=1}^{k} \alpha_j = 1, \\ & \qquad \alpha_j \geq 0, \qquad j = 1,\ldots,k. \end{array} \tag{2.69}$$

Observe that the dual formulation of the SD direction subproblem is equivalent to the direction taken by the method of Schäffler et al. [19] explained earlier.

From these three methods, the SD is the only one that can handle inequality constraints. For this, a LP version of (2.65) is utilized given by

$$\begin{array}{lll} \min\limits_{(\nu,\delta) \in \mathbb{R}^n \times \mathbb{R}} & \delta & \\ \text{s.t.} & \nabla f_j(x)^T \nu \leq \delta, & j = 1,\ldots,k, \\ & \nabla g_j(x)^T \nu \leq \delta, & j \in \mathcal{I}, \\ & \quad \|\nu\|_\infty \leq 1. & \end{array} \tag{2.70}$$

Note that the change consists of removing the quadratic term of the objective function in (2.65) and adding one more constraint to keep the problem bounded. By the choice of the infinity norm, the additional inequality is also linear. Hereby, $\mathcal{I} = \{j \mid g_j(x) > -\epsilon, \quad j = 1,\ldots,m\}$ denotes the set of active and nearly active constraints where $\epsilon > 0$ is a small value. Further, the step length control has to satisfy the Armijo condition while keeping the subsequent iterations feasible.

As far as we know, neither of these methods has been coupled into a global approach such as EAs or subdivision or cell mapping techniques, leaving room for future research in these lines of investigation. This is also the case for the development of continuation methods based on these strategies, which forms an important part of the contributions of this work.

Continuation Methods

The prime motivation to apply continuation for the numerical treatment of MOPs comes from the fact that under certain mild assumptions, it can be induced from the KKT conditions that the Pareto set (respectively its image, the Pareto front), of a continuous MOP is a piecewise continuous $(k-1)$-dimensional manifold [28], where k is the number of objectives. Thus, specialized techniques capable to perform a search along the manifold of solutions—if one (or more) solution is at hand—promise to be very efficient applied to this context.

Continuation methods can be seen from the point of view made in [38, 37]. That is, the set of methods and techniques for the numerical approximation of a curve[9] implicitly defined by an undetermined system of equations. A particularly attractive branch—because of their robustness and flexibility—are the so called predictor corrector (PC) or pseudo arc length continuation methods, which also constitute the spine of this work. Thus, we will first state the main steps of PC techniques for tracing one-dimensional solution sets of underdetermined nonlinear equations. For a more thorough discussion as well as extensions to higher dimensional solution sets we refer e.g. to [38, 37, 85]. Assume we are given the equation

$$H(x) = 0, \tag{2.71}$$

where $H : \mathbb{R}^{N+1} \to \mathbb{R}^N$ is sufficiently smooth. If x is a solution of (2.71) with $\text{rank}(H'(x)) = N$, then it follows by the Implicit Function Theorem (IFT) [28] that there exist a value $\epsilon > 0$ and a curve $c : (-\epsilon, \epsilon) \to \mathbb{R}^{N+1}$ such that $c(0) = x$ and

$$H(c(s)) = 0, \qquad \forall s \in (-\epsilon, \epsilon). \tag{2.72}$$

Differentiating (2.72) leads to

$$H'(c(s)) \cdot c'(s) = 0. \tag{2.73}$$

Hence, tangent vectors $c'(s)$ (and thus, linearizations of the solution curve at $x = c(s)$) can be found via computing kernel vectors of $H'(x)$. This is done in literature by a QR factorization [86] of $H'(x)^T$: if

$$H'(x)^T = QR = (q_1, \ldots, q_{N+1})\, R \tag{2.74}$$

for an orthogonal matrix $Q \in \mathbb{R}^{(N+1)\times(N+1)}$ and a right upper triangular matrix $R \in \mathbb{R}^{(N+1)\times N}$, then the last column vector q_{N+1} of Q is such a desired kernel vector. The orientation of the curve (note that both $+q_{N+1}$ and $-q_{N+1}$ are desired linearizations, but point in opposite directions) can be inferred by monitoring the sign of

$$\det \begin{pmatrix} H'(x) \\ q_{N+1}^T \end{pmatrix}. \tag{2.75}$$

[9] The definition in [38, 37] is given for problems with one-dimensional solution sets. For problems with higher dimensional solution sets, ‘curve’ can simply be replaced by ‘manifold’ in the definition.

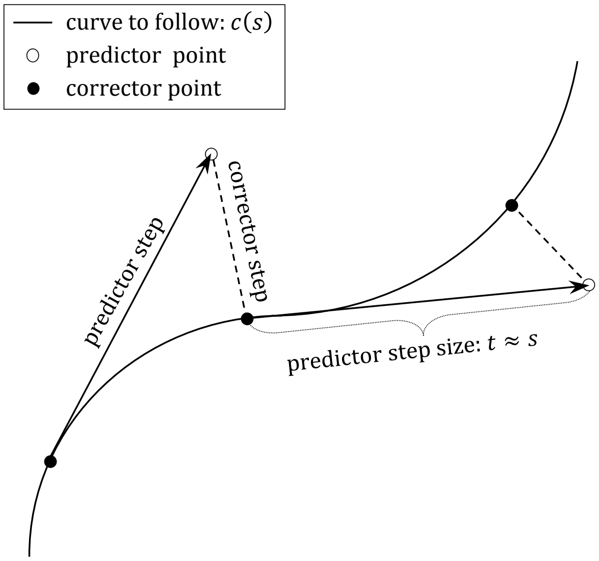


Figure 2.7: Scheme followed by predictor corrector methods.

More specifically, we change the sign of the computed predictor whenever the determinant in (2.75) is negative. Having the vector pointing along the linearized solution curve, a movement in that direction can be performed leading to a predictor point p. In a following corrector step, one can get back to c by using (2.71) e.g. via a Gauss-Newton or a Levenberg-Marquardt method [58] starting with p. In this manner, a sequence of solutions that are aligned along the curve $H^{-1}(0)$ can be obtained. Figure 2.7 provides a more illustrative example of the working principle of PC methods.

The relation between the classical continuation theory and multi-objective optimization is thus easily derived: the first-order optimality conditions for MOPs (see Theorem 2.1.2) lead to an undetermined system of equations (that we call indistinctly KKT system or KKT equations). Thus, applying the IFT on the underlying system and under some mild conditions, it is inferred that the optimal solution set is a $(k-1)$-dimensional object that can be tracked e.g. by PC methods. A more detailed argument on this subject is provided in [28]. The algorithms we review next closely follow the spirit of the continuation theory (in particular, PC methods) although not all are oriented to explicitly solve the KKT system of equations. We separate them into three groups: the classical methods that focus on the solution of the KKT system, the scalarization-based methods which formulate a scalar subproblem—based on the objectives and the constraints of the original MOP—whose corresponding KKT system has to be solved, and the gradient-based methods that use the gradients to predict a suitable point that later is corrected until an optimal solution is found. The last two groups—though implicitly—end up with a set of solutions for the KKT equations where the Lagrange multipliers may (if necessary) be determined a-posteriori.

Before getting into the details of the specific methods, we would like to emphasize that these are very efficient procedures in terms of function evaluations, though, mainly because of its local nature. This implies that we can get trapped into local optima or miss sections of the solution set if it is not connected. Thus, the necessity of hybridizing local methods with global strategies arises once again for the solution of MOPs. So far, continuation methods have been used as recovering operators applied after global subdivision techniques with the aim to repair sections of the front that may has been discarded. See for instance [34] which is mentioned later in the appropriate context. A different approach is based on a Delaunay tessellation of the search space [87, 88] which utilizes simplices instead of hypercubes—as done by cell mapping and subdivision techniques—to discretize the search space. Another alternative is to combine continuation strategies with metaheuristics. One example that integrates a MOEA, a recovering technique and a continuation approach is the Evolutionary Recover Algorithm (ERA) proposed in [35]. Additionally, a very attractive idea was developed in [36] that consists of a curve-based MOEA combined with a gradient-based PC method (see below). However, continuation methods for MOPs are, in general, not derivative-free. To our knowledge, a PC algorithm based on the derivative-free local search given in [22] and a gradient-free version of the DS method [23] are the only proposals in this field. Both strategies are integrated into a metaheuristic.

Classical Continuation Methods We call *classical* continuation methods those strategies based on the use of standard continuation routines on the system of equations defined by the first-order optimality conditions for MOPs. According to our knowledge, there are only two approaches in this category: the method of Hillermeier (immediately described) and the approach of Martin et al. [29] which extends the former to handle inequality constraints, although, restricted to bi-objective optimization problems (BOPs).

The Method of Hillermeier A direct application of PCs on the KKT equations of a MOP was first introduced by Hillermeier in [28]. Consider the map

$$\tilde{F}(x, \alpha, \lambda) = \begin{pmatrix} \sum\limits_{j=1}^{k} \alpha_j \nabla f_j(x) + \sum\limits_{j=1}^{p} \lambda_j \nabla h_j(x) \\ h(x) \\ \sum\limits_{j=1}^{k} \alpha_j - 1 \end{pmatrix} = 0. \tag{2.76}$$

The set of KKT points of a nonlinear equality constrained MOP is contained in the zero set of $\tilde{F}$ which motivates the continuation along $\tilde{F}^{-1}(0)$. For BOPs, the method proceeds as the general technique described above but with the following change: instead of computing the determinant given in (2.75) to orientate the continuation, the author proposes to check whether the condition

$$[x - \tilde{x}] \cdot q \geq 0 \tag{2.77}$$

is met, where $x = (x, \alpha, \lambda) \in \mathbb{R}^{n+k+p}$ is the current corrector point, $\tilde{x}$ is the previously computed solution, and q is the tangent vector. If that is not the case, the direction of q is flipped. Then, a suitable step length that guarantees a uniform spread of solutions on the front is sought. That is, for two consecutive solutions $\tilde{x}$ and x, it is desirable that

$$\|F(x) - F(\tilde{x})\| \approx \tau, \tag{2.78}$$

where $\tau > 0$ is a user specified value. For this, one can take the step size

$$t = \frac{\tau}{\|Jq\|} \tag{2.79}$$

inferred from linear approximations of the objectives—by first-order Taylor expansions [89]—which is close to reality for small step sizes ($\tau \ll 1$). Additionally, some conditions need to be assumed here in order to apply the IFT to $\tilde{F}$: the LICQ constraint qualification must hold as well as the regularity of the matrix $\left(\sum_{j=1}^{k} \alpha_j \nabla f_j(x) + \sum_{j=1}^{p} \lambda_j \nabla h_j(x)\right)$ in (2.76). The multi-objective case is handled by taking kernel vectors of $\tilde{F}'$ obtained via a QR factorization [86] of $\tilde{F}'^T$. Given that

$$\tilde{F}'(x, \alpha, \lambda)^T = QR = (q_1, \ldots, q_{n+p+k})\, R \tag{2.80}$$

for an orthogonal matrix $Q \in \mathbb{R}^{(n+k+p)\times(n+k+p)}$ and a right upper triangular matrix $R \in \mathbb{R}^{(n+k+p)\times(n+p+1)}$, the last $k-1$ column vectors of Q forms an orthonormal basis of the linearized solution set in the compound (x, α, λ)-space. Thus, one can e.g. move in the directions of the computed orthonormal vectors $q_{n+p+2}, \ldots, q_{n+p+k}$ to obtain predictors that are grid-aligned along the tangent space of the optimal manifold in decision space. A problem of this election, though, is that after mapping the computed predictors to the objective space, the grid-alignment is probably not kept. Finally, the continuation algorithm is stopped if one of the Lagrange multipliers α_j, $j \in \{1, \ldots, k\}$ is negative, which is indicative that a non-optimal solution has been found.

The Hillermeier algorithm has been successfully applied in [34] within a global subdivision technique to recover missing parts of the front that may have been discarded during the subdivision phase. Recall that subdivision strategies are based on successive partitions of the space into boxes that may be removed if no solution is found within them. Nevertheless, this cut-off test does not guarantee that boxes containing optimal points are not excluded. Thus, after the subdivision phase, a recovering process is performed consisting of a continuation algorithm starting at the solutions corresponding to the retained set of boxes.

On the other hand, one of the main drawbacks of classical approaches is the requirement of the exact Hessians of all the functions involved—objectives as well as equalities—followed by a lack of strategies to handle inequality constraints. To overcome the former limitation, a modification is proposed in [90] based on successive approximations of the tangent space. This proposal allows finding promising predictor points with lower effort (in particular for high dimensional models) since no Hessian of the objectives have to be calculated. Additionally, an extension to tackle inequality constrained problems is given by Martin et al. in [29] which employs a modification of ParCont [91] a rigorous PC method based on interval analysis and parallelotope domains. The fundamental idea of the suggested procedure rests on a certified continuation on the first-order conditions of optimality coupled with an active set management strategy inspired by the method of Rakowska et al. [27] (explained below) which is, however, restricted to bi-objective problems.

Scalarization-based Continuation Methods This class of methods is (so far) mainly concerned with BOPs and can be seen as a particular instance of parametric optimization. The idea is to transform the MOP into a SOP by inducing an additional parameter that controls the trade-off between the optimal solutions (as in scalarization methods). The difference here lies in the way the resulting parametric SOP is treated: instead of solving several instances of the SOP for a sequence of a-priori computed parameters, a continuation-like method is used to track the solution manifold of the parametric SOP.

The Method of Rakowska, Haftka and Watson One interesting approach in this class is proposed by Rakowska et. al in [27] which is based on a previous work by the same authors on parametric optimization [92]. The method relies on a scalarization subproblem based on the weighted sum of the objectives as defined below.

$$\begin{array}{lll} \min\limits_{(x,\alpha)\in\mathbb{R}^n\times\mathbb{R}} & (1-\alpha)f_1(x)+\alpha f_2(x) & \\ \text{s.t.} & g_j(x)\leq 0, & j=1,\dots,m, \\ & h_j(x)=0, & j=1,\dots,p.^{10} \end{array} \tag{2.81}$$

The goal here is to trace the path of KKT points of (2.81) for different values of the parameter $\alpha \in [0,1]$. This task is achieved by a PC approach on the KKT system of equations of (2.81) combined with a strategy to identify and maintain the set of active constraints. The latter is specially important since the solution curve is not necessarily smooth at points corresponding to changes in the active set. Thus, each smooth segment of the optimal curve is detected by solving the reduced KKT system

[10] This method was not originally proposed to handle equalities but they can be included with no additional modification. Furthermore, for the sake of simplicity, the box constraints are assumed to be general inequalities.

of equations involving only the active constraints:

$$(1-\alpha)\nabla f_1(x) + \alpha\nabla f_2(x) + \sum_{j=1}^{p}\lambda_j \nabla h_j(x) + \sum_{j\in\mathcal{I}}\gamma_j \nabla g_j(x) = 0, \tag{2.82a}$$

$$h_j(x) = 0, \quad j = 1,\dots,p, \tag{2.82b}$$

$$g_j(x) = 0, \quad j \in \mathcal{I}, \tag{2.82c}$$

where $\mathcal{I} = \{j \mid g_j(x) > -\epsilon, \quad j = 1,\dots,m\}$ denotes the set of active and nearly active constraints for a small tolerance $\epsilon > 0$. The logic behind this strategy is that the active constraints in each smooth segment are fixed, so they can be treated as equalities. Then, in order to detect *switching points* (transition points to a new segment), it is necessary to identify if:

- a previously positive Lagrange multiplier γ_j, $j \in \{1,\dots,m\}$ becomes zero, or
- a previously negative inequality g_j, $j \in \{1,\dots,m\}$ becomes zero.

Since it is known that the Lagrange multipliers associated with inactive constraints should be zero, the first item may be indicative that one of the current active constraints becomes inactive. The second item is associated with the opposite scenario: when one of the inactive constraints becomes active. In such cases, it is necessary to restart the continuation algorithm with a new set of active constraints, although, a special analysis should be made in cases for which both conditions are satisfied: γ_j and g_j become equal to zero for the same index $j \in \{1,\dots,m\}$. A better discussion on how to handle different types of singularities is discussed in [93, 92].

The orientation of the path following is given by the sign of $\partial\alpha/\partial s$ where s is the arc length of the curve. Finally, second-order optimality conditions are utilized to determine whether the solutions of the KKT system are indeed minima, in which case we can decide either stop the procedure or to follow the critical points until the solutions become optimal again. In spite of the fact that this method can satisfactory handle inequality constraints, it is restricted to convex BOPs with known Hessians, which is, indeed, a disadvantage.

The Method of Pereyra, Saunders and Castillo The prime goal of this procedure [32] is to provide an equispaced Pareto front discretization for BOPs (same goal as the NBI design). The method follows the idea of formulating equispacing constraints from a previous paper [94] and extend it for the management of constrained problems. Here, an arc length continuation is performed with a parameter α and an additional equality constraint, one that is explicitly designed to request an equally spaced outcome. Specifically, the task is to solve the subproblem

$$\begin{array}{lll} \min\limits_{(x,\alpha)\in\mathbb{R}^n\times\mathbb{R}} & (1-\alpha)f_1(x)+\alpha f_2(x) & \\ \text{s.t.} & \|F(x)-F(\tilde{x})\|^2=\tau^2, & \\ & g_j(x)\leq 0, & j=1,\dots,m, \\ & h_j(x)=0, & j=1,\dots,p, \\ & l\leq x\leq u, & \\ & 0\leq\alpha\leq 1, & \end{array} \tag{2.83}$$

where $\tilde{x}$ is the previously computed solution, and $\tau > 0$ is an estimation of the distance between the consecutive solutions. Note that the subproblem (2.83) is a convex combination of the objectives, which is well-known by its difficulty to select α-values that generate an evenly distributed set of solutions, though, it is precisely this issue what the equispacing constraint is intended to handle.

Just as the NBI method, this strategy requires the individual global minima of each objective, but this time to estimate the space between points on the front, given by

$$\tau = \frac{c\|F_1^* - F_2^*\|}{l-1}, \tag{2.84}$$

where $c \geq 1$ and l is the number of desired solutions. Then, the method stops after performing exactly $l-2$ steps. Here, one complication arises from our ignorance of the exact arc length: we may fall too short of or overshoot the end-point. Furthermore, the equispacing constraint provides no insight of the orientation of the continuation, thus, in order to prevent going backward we need to impose constraints on the objectives (e.g., defining the one that decreases certainly provides an orientation).

In a more direct comparison to the NBI and the NNC methods, a clear advantage is the non-necessity of an a-priori known set of uniformly distributed weights. Instead, the parameter α is implicitly determined at each iteration. However, extending this approach to the general case would be clearly more challenging: (i) it is also required an ordering of the subproblems (at least defining a previous solution is mandatory), (ii) the complexity of the subproblems increases since the parameter α augments its dimension and, (iii) additional distance constraints would be necessary. Again, no second derivatives of the functions are crucial, but the inclusion of the equispacing constraint may turn the subproblem treated here nonconvex, even if the original objectives and constraints are convex.

There are other approaches in this category that we won't be covered here. However, we will simply highlight one of them: the ODE Normalized Normal Equality Constraint (ODE NNEC) method [31]. The emphasis is given to this one in particular since the associated scalar subproblem is not based on the weighted sum of the objectives. Instead, the NNC subproblem is used with a small change: the NNC (an inequality) is replaced by a NNEC (an equality). This change makes the approach a kind of hybrid between the NBI and the NNC methods. Additionally, the idea of managing the inequality constraints by tracking the active set changes is incorporated from the method of Rakowska et al. [27]. The difference is that instead of solving the corresponding KKT equations by classical PC methods, the authors propose to use an ODE integrator able to detect the changes of activity of the inequalities.

Gradient-based Continuation Methods The aim of the set of methods described here is as well to look for further solutions by following the optimal manifold of a MOP. However, they do not (explicitly) solve the system of equations induced by the first-order optimality conditions (as in classical continuation methods) neither use any scalarization approach (as in scalarization-based continuation methods). Instead, they perform a step along the front by following one of the gradients (or a linear combination of them) in order to compute a promising predictor point. Then, the task is to correct the predicted point back to the solution manifold by using exclusively the derivatives of the objectives. This strategy leads to Hessian-free PC methods with the additional advantage that the iterations are generally very efficient in terms of computational time complexity and memory usage. Still, since only first-order information is used, a great number of iterations is expected for problems with Pareto sets showing a high degree of curvature. Thus, these methods should be seen as first steps toward procedures with better rates of convergence.

The Directed Search Continuation Method The novel idea of the following PC method [20] is the suggestion of a new predictor direction based on the geometry of the Pareto front. Given a (local) Pareto point $x \in \mathbb{R}^n$ of an unconstrained MOP and the associated Lagrange multipliers $\alpha \in \mathbb{R}^k$, it is known (see e.g. [28]) that α is orthogonal to the linearized Pareto front at $F(x)$. Thus, a movement in the orthogonal space to α (in objective space) could be a promising search direction to obtain new predictor points. For this purpose, a QR factorization [86] of α is utilized such that

$$\alpha = QR, \tag{2.85}$$

where $Q = (q_1, \dots, q_k) \in \mathbb{R}^{k \times k}$ is an orthogonal matrix and $R = (r_{11}, 0, \dots, 0)^T \in \mathbb{R}^k$ with $r_{11} \in \mathbb{R}_{++}$. Here, the last $k-1$ columns of Q forms an orthonormal basis of the kernel space of α (since the columns of Q are orthogonal and $\alpha = r_{11}q_1$ by (2.85)). Taking the previous reasoning into account, a possible well-spread set of predictor directions ν_i may be the ones that satisfy

$$J\nu_i = q_i, \qquad i = 2, \dots, k.^{11} \tag{2.86}$$

The directions in (2.86) coincide with the search direction taken by the DS method discussed above. Thus, the computation of predictors this way can be seen as further applications of the DS method on Pareto points, where a fixed step length—that is a problem dependent parameter—is chosen. Next, the corrector phase is handled once

[11] Additionally, the negative vectors $-\nu_i$, $i = 2, \dots, k$, are also taken.

again by the DS method where the search direction in objective space is taken as $-\alpha$. Finally, to deal with problems with more than two objectives, a coupling of this approach with the recovering algorithm in [34] is achieved with promising results. The latter is, however, not necessary for the bi-objective case where the use of data structures can be avoided. The idea is to perform first a continuation all along in the direction to optimize one of the objectives, and then (by reversing the orientation) a continuation all along in the direction to optimize the other objective. Thus, the current solution is guaranteed to be new and the knowledge of the previously obtained solutions is not required. More precisely, suppose that an improvement according to f_2 is desired. Then, in order to give the right orientation to the continuation, the coefficient q_{22} is used in the computation of predictors such that

$$p = x - \operatorname{sgn}(q_{22}) t\nu, \tag{2.87}$$

where $t \in \mathbb{R}_{++}$ stands for a suitable step length. The continuation is thus stopped when the extreme vector of weights[12] $\alpha = (0, 1)^T$ is reached. If, on the contrary, f_1 should be improved, we flip the sign of the second term in (2.87) and stop when the current multipliers are near enough to $\alpha = (1, 0)^T$. A derivative-free version of the DS method that exploits neighborhood information to approximate gradients has been proposed in [23].

The Zigzag Search Method The Zigzag Search (ZS) method is another gradient-based continuation algorithm for the numerical treatment of BOPs that was designed in [30]. This method works by alternating the gradients of the objectives in a local search along the front, leading to a kind of zigzag behavior capable of producing a discretization of the solution curve. More precisely, the predictor and corrector steps are renamed to *zig* and *zag*, which are successively executed after a previous FFPO (find the first Pareto optimum) algorithm.

Let's suppose that we start at a Pareto point $x_1 \in \mathbb{R}^n$ and the continuation is oriented to decrease f_2. The zig step performs a movement following the direction of $\nabla f_1(x_1)$ to obtain a candidate solution p with a higher value of f_1 and a lower value of f_2. Then, in a consecutive zag step, we move in the projected direction of $-\nabla f_2(p)$ onto the orthogonal hyperplane to $\nabla f_1(p)$, obtaining, this time, a solution x_2 with a lower value of f_2 and a slightly changed value of f_1. The zag step is repeated until no further improvement can be achieved on f_2 and a new Pareto optimal point is found. Finally, the procedure stops if there are no more Pareto optimal solutions found at the current zigzag iteration. For a better understanding, we refer the reader to Figure 2.8.

The idea proposed by this method contributes to computationally efficient implementations (note that no optimization subproblem or equation system has to be solved) but a clear limitation here is that the steepest descent behavior of the approach may

[12] The terms 'weights', 'Lagrange multipliers', 'KKT multipliers', and just 'multipliers' can be interchanged whenever it leads to no ambiguity.

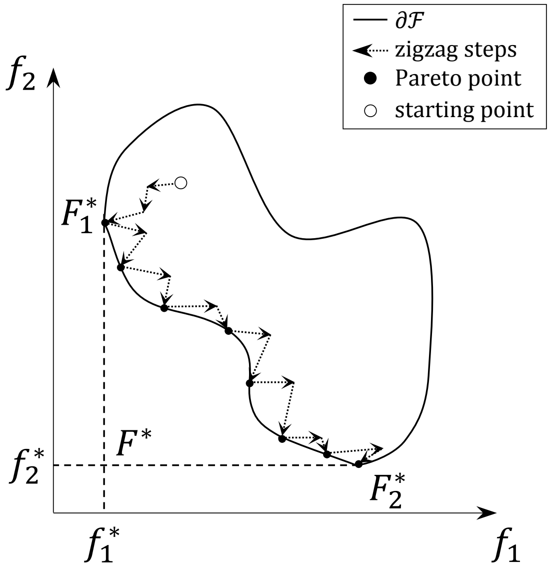


Figure 2.8: One run of the Zigzag Search method.

result in a potential loss of performance making the method zigzag along and toward the front. Additionally, the extension to problems with more than two objectives seems to be a non-trivial task.

There is another gradient-based PC approach for BOPs whose reading is recommended since it is used in a very interesting MA. This is the Pareto Path Following (PPF) method [36] which performs a predictor step guided by the gradient of one of the objectives (as done by the ZS method) and then utilizes the Pareto Descent local search [95] (a kind of steepest descent optimizer) as corrector. It also incorporates a procedure to manage inequalities by means of gradient projections: the Pareto Descent Repair Operator proposed in [96]. However, the most attractive contribution of the approach is the integration of the PPF into a curve-based MOEA. The resulting MA considers the disconnected fragments of the solution curve as the units of search, where the genetic crossover combines solutions from different curves to find undiscovered sections of the front. The role of the local search is therefore to discover connected solutions. Additionally, two metrics are proposed to assess the performance of the algorithm: an inter-curve coverage (interested in the discovery of every portion of the optimal curve) and an intra-curve coverage (focused in a satisfactory sampling of each connected portion).

Chapter 3

The Pareto Tracer Method

In this chapter, we present a novel PC method for the treatment of MOPs where we utilize classical continuation techniques on a map $\tilde{F}$ that is motivated by the KKT equations for MOPs. More precisely, for every optimal solution x (in decision space of a MOP) there exists an associated convex weight (or Lagrange multipliers) α such that (x, α) satisfies the first-order conditions of optimality and that this tuple is contained in the zero set of $\tilde{F}$. In [28], $\tilde{F}$ is used to perform a PC method in the compound (x, α)-space. The additional consideration of the weight space, however, comes with a possible increase of the nonlinearity of the solution set. If, for instance, the Pareto set is linear—which is the ideal case for PC methods since then no corrector step has to be performed—the related solution set does *not* have to be linear in the (x, α)-space leading to additional corrector steps (see the next section for such an example). In our algorithm we utilize $\tilde{F}$, but separate decision and weight space whenever possible leading to significant savings in the overall computational cost. For the corrector, we use the multi-objective Newton method proposed by Fliege et al. in [16] which assumes the objectives to be strictly convex. The method is also applicable to nonconvex problems, however, without guarantee of convergence. The resulting PC method, the Pareto Tracer, is in principle applicable to MOPs with any number of objectives and can be made Hessian-free via quasi-Newton or gradient descent methods. For simplicity, we will restrict this chapter to the unconstrained case

$$\min_{x \in \mathbb{R}^n} F(x), \tag{3.1}$$

and discuss in the next chapters problems with different types of constraints.

The remainder of this chapter is organized as follows. In Section 3.1 we address issues related to the computation of predictor points while Section 3.2 is dedicated to the corrector phase. In Section 3.3 we present the Pareto Tracer, and discuss Hessian-free realizations using ideas from quasi-Newton and gradient descent methods in Section 3.4. In Section 3.5, we show some numerical results, and finally conclude in Section 3.6.

3.1 Predictor

The first task for the computation of predictor points in continuation methods is typically to determine the tangent space to the curve (or manifold) of interest at a given solution. Let $x \in \mathbb{R}^n$ be a KKT point of (3.1) and $\alpha \in \mathbb{R}^k$ its associated Lagrange multipliers, that is, $\alpha_j \geq 0$, $j = 1, \ldots, k$, with $\sum_{j=1}^k \alpha_j = 1$ and

$$\sum_{j=1}^{k} \alpha_j \nabla f_j(x) = J^T \alpha = 0. \tag{3.2}$$

Crucial for our considerations is the map

$$\tilde{F}(x, \alpha) = \left(\begin{array}{c} \sum\limits_{j=1}^{k} \alpha_j \nabla f_j(x) \\ \sum\limits_{j=1}^{k} \alpha_j - 1 \end{array} \right) = 0 \tag{3.3}$$

motivated by the KKT conditions for unconstrained MOPs (see Theorem 2.1.2). In [28], kernel vectors of $\tilde{F}'$ are obtained via a QR factorization of $\tilde{F}'^T$ which yields an orthonormal basis of the linearized solution set in the compound (x, α)-space. The additional consideration of the weight space, however, comes with a possible increase of the nonlinearity of the solution set. The latter is mainly due to the difference of magnitudes between the parameters of the problem and the Lagrange multipliers (i.e., $0 \leq \alpha_j \leq 1$, $j = 1, \ldots, k$). A more illustrative example is given in Figure 3.1 where the Pareto set of the BOP (3.69) is displayed together with its corresponding values of α_1. Note that while the Pareto set is a line, the augmented optimal curve in (x, α)-space is not. Considering that predictor points are obtained through linearizations of the solution set, following the (generally nonlinear) trajectory of tuples (x, α) will inevitably lead to the computation of correctors. Instead, moving in a tangent direction to the Pareto set (in this case) will end up with a predictor that is already on the desired curve, so no corrector step would be necessary.

Inspired by the previous discussion, we choose another way to proceed which allows to separate decision from weight space yielding tangent vectors to the Pareto set. In Chapter 2, we learned that $\tilde{F}'$ is orthogonal to the solution manifold of $\tilde{F}$ at a given solution point. Thus, as in [28], we are seeking for kernel vectors of $\tilde{F}'$ which are in turn tangent to the augmented optimal set. However, in order to be able to decouple x and α, QR decompositions of $\tilde{F}'^T$ are replaced this time by direct computations of kernel vectors $(\nu, \mu) \in \mathbb{R}^{n+k}$ such that

$$\tilde{F}'(x, \alpha) \begin{pmatrix} \nu \\ \mu \end{pmatrix} = \begin{pmatrix} \sum_{j=1}^k \alpha_j \nabla^2 f_j(x) & \nabla f_1(x) \ldots \nabla f_k(x) \\ 0 & 1 \quad \ldots \quad 1 \end{pmatrix} \begin{pmatrix} \nu \\ \mu \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}. \tag{3.4}$$

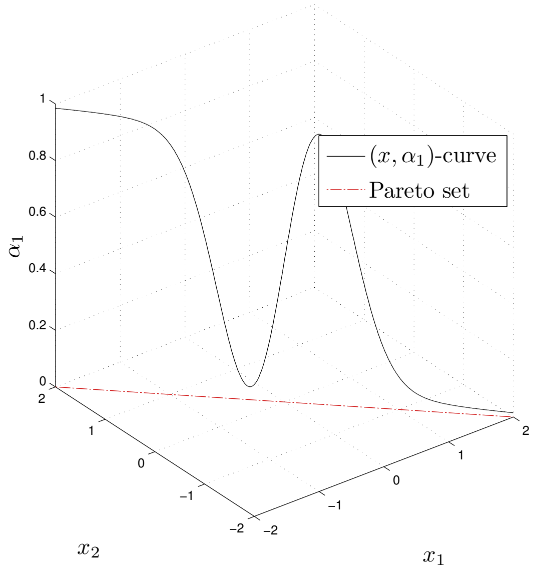


Figure 3.1: Pareto set of the BOP (3.69) and its corresponding Lagrange multiplier α_1.

Here, by the second equation in (3.4) we see that

$$\sum_{j=1}^{k} \mu_j = 0. \tag{3.5}$$

Assume for now that we are given a vector $\mu \neq 0$ such that the above property is fulfilled. Then, we obtain by the first equation in (3.4)

$$\sum_{j=1}^{k} \alpha_j \nabla^2 f_j(x) \nu = -\sum_{j=1}^{k} \mu_j \nabla f_j(x) = -J^T \mu. \tag{3.6}$$

Additionally, under the requirement that the matrix

$$W_\alpha = \sum_{j=1}^{k} \alpha_j \nabla^2 f_j(x) \in \mathbb{R}^{n \times n} \tag{3.7}$$

is regular, the vector ν_μ that satisfies (3.4) can be expressed as

$$\nu_\mu = -W_\alpha^{-1} J^T \mu. \tag{3.8}$$

Let's show next that if $\text{rank}(J) = k - 1$, the vector ν_μ is zero if and only if μ is also zero. Since W_α is regular, we have that $\nu_\mu = 0$ if and only if $J^T \mu = 0$. By the rank nullity theorem [97] $\dim(\ker(J^T)) = 1$ and from the KKT conditions of optimality $J^T \alpha = 0$ (see Equation (3.2)). Thus, $J^T \mu = 0$ if and only if $\mu = c\alpha$ for some $c \in \mathbb{R}$. Suppose that the previous condition is met, then by (3.5) we have that $0 = \sum_{j=1}^k \mu_j = c \sum_{j=1}^k \alpha_j = c$ which implies that $\mu = 0$. Note that by the same argument it is possible to prove that under our rank assumptions the Jacobian matrix is not orthogonal to the linearized Pareto set.

The following discussion shows that ν_μ is a tangent vector to the Pareto set at x. Let

$$g_\alpha(x) = \sum_{j=1}^k \alpha_j f_j(x). \tag{3.9}$$

Then, a curve along the set of KKT points through x_0 with associated weight α_0 can be described via

$$H(x,t) = \nabla g_{\alpha(t)}(x) = 0, \qquad t \in \mathbb{R}_+. \tag{3.10}$$

Since μ represents the change in α-space we can set $\alpha(t) = \alpha_0 + t\mu$, and it is thus $H(x_0, t_0 = 0) = 0$. Consequently, for the tangent vector of $H^{-1}(0)$ at (x_0, t_0) it holds

$$t_H(x_0,t_0) = -\left(\frac{\partial H}{\partial x}\right)^{-1} \frac{\partial H}{\partial t} = -\left(\sum_{j=1}^k \alpha_j \nabla^2 f_j(x)\right)^{-1} \left(\sum_{j=1}^k \mu_j \nabla f_j(x)\right) = \nu_\mu. \tag{3.11}$$

The second step of the predictor phase is to decide the orientation of the movements along the tangent space. In order to accomplish this task, we will show immediately that μ can be used to steer the search in objective space. First of all, given a direction $\nu \in \mathbb{R}^n$ in parameter space of (3.1) the corresponding movement in objective space for infinitesimal step sizes is given by

$$d = J\nu. \tag{3.12}$$

To see this, consider the j-th component of the right hand side of (3.12):

$$(J\nu)_j = \langle \nabla f_j(x), \nu \rangle = \lim_{t \searrow 0} \frac{f_j(x+t\nu) - f_j(x)}{t}, \qquad j = 1, \ldots, k. \tag{3.13}$$

Secondly, since x is a KKT point, almost all values of ν lead to a movement along the linearized Pareto front:

$$\langle J\nu, \alpha \rangle = \langle \nu, J^T \alpha \rangle \overset{(3.2)}{=} 0. \tag{3.14}$$

Taking into account that α is orthogonal to the Pareto front [28], Equation (3.14) implies that either $J\nu = 0$ or a movement along the linearized front at $F(x)$ is performed. It is hence desired to steer the search into a direction $d \in \mathbb{R}^k \setminus \{0\}$ in objective space such that d is orthogonal to α. Thus, the task is to find the proper orientation vector $\mu_d \in \mathbb{R}^k$ that satisfies

$$J\nu_{\mu_d} = d. \tag{3.15}$$

If we substitute ν_{μ_d} in Equation (3.15) by its expression given by (3.8), we obtain

$$J\nu_{\mu_d} = -JW_\alpha^{-1}J^T\mu_d = d. \tag{3.16}$$

However, the solution of the linear system (3.16) either does not exist, or is not unique. Recall that J is rank deficient at optimal points so it is $JW_\alpha^{-1}J^T$. Let's take

$$A = -JW_\alpha^{-1}J^T \in \mathbb{R}^{k\times k}. \tag{3.17}$$

Given that $\text{rank}(J) = k-1$, it follows that also $\text{rank}(A) = k-1$. Further, since $J^T\alpha = 0$, then $A\alpha = 0$, and thus $\text{kernel}(A) = \text{span}\{\alpha\}$. By construction $d \perp \alpha$ (i.e., d and α are linearly independent). Then, Equation (3.16) has one and thus an infinite number of solutions. More precisely, suppose that μ_d^+ is one particular solution, e.g.,

$$\mu_d^+ = A^+ d. \tag{3.18}$$

Then, the solution of (3.16) can be expressed as

$$\mathbb{M} = \mu_d^+ + \mathbb{R}\alpha = \left\{\mu_d^+ + c\alpha \mid c \in \mathbb{R}\right\}, \tag{3.19}$$

and in order to find a $\mu_d \in \mathbb{M}$ that satisfies the condition (3.5), we do

$$0 = e^T\mu_d = e^T(\mu_d^+ + c\alpha) = e^T\mu_d^+ + ce^T\alpha = e^T\mu_d^+ + c, \tag{3.20}$$

where $e \in \mathbb{R}^k$ is the vector of all ones. Such a solution is therefore

$$\mu_d = \mu_d^+ - (e^T\mu_d^+)\alpha. \tag{3.21}$$

It is not required, however, to explicitly determine μ_d. Note that

$$\begin{aligned} \nu_{\mu_d} &= -W_\alpha^{-1}J^T\mu_d = -W_\alpha^{-1}J^T(\mu_d^+ + c\alpha) = -W_\alpha^{-1}J^T\mu_d^+ - cW_\alpha^{-1}J^T\alpha \\ &= -W_\alpha^{-1}J^T\mu_d^+. \end{aligned} \tag{3.22}$$

We can thus compute a set of vectors $\nu_1, \ldots, \nu_{k-1}$ such that the corresponding movements $J\nu_i$, $i = 1, \ldots, k-1$, in objective space form an orthonormal basis of the linearized Pareto front at $F(x)$: let

$$\alpha = QR = (q_1, \ldots, q_k)R \tag{3.23}$$

be a QR factorization of α, where q_i is the i-th column vector of Q, then set

$$d_i = q_{i+1}, \qquad i = 1, \ldots, k-1, \tag{3.24}$$

to obtain an orthonormal basis of the tangent space to the Pareto front. Later, for each direction d_i in objective space, we can determine $\mu_{d_i}^+$ via (3.18) to be utilized in the computation of the corresponding $\nu_i = \nu_{\mu_{d_i}}$ by (3.22). If it is alternatively desired to obtain orthogonal vectors that span the linearized Pareto *set*, one can e.g. realize this via computing an orthonormal basis of the image of

$$M = -W_\alpha^{-1} J^T \in \mathbb{R}^{n \times k}. \tag{3.25}$$

Note that since $M\mu$ is tangent to the Pareto set for any $\mu \neq 0$ such that $\sum_{j=1}^k \mu_j = 0$, it follows that M spans the tangent space to the Pareto set. Finally, we summarize the results of the above discussion in the following theorem.

Theorem 3.1.1
Let x be a KKT point of (3.1) *with associated Lagrange multipliers $\alpha \in \mathbb{R}^k$. Further, let* $\operatorname{rank}(J) = k-1$, *and W_α be regular. Let $d_1, \ldots, d_{k-1}$ be as in* (3.24), *and $\mu_{d_i}^+$ as in* (3.18). *Then, the vectors*

$$\nu_{\mu_{d_i}} = -W_\alpha^{-1} J^T \mu_{d_i}^+, \qquad i = 1, \ldots, k-1, \tag{3.26}$$

point along the linearized Pareto set at x such that the vectors

$$J\nu_{\mu_{d_i}}, \qquad i = 1, \ldots, k-1, \tag{3.27}$$

form an orthonormal basis of the linearized Pareto front at $F(x)$.

For the special case $k = 2$ there are—after normalization—only two choices for μ with the property (3.5):

$$\mu^{(1)} = (1, -1)^T \quad \text{and} \quad \mu^{(2)} = (-1, 1)^T. \tag{3.28}$$

If both objectives are strictly convex, we can directly steer the search without using (3.18): given μ, we have for the corresponding movement in objective space

$$\begin{aligned} d_j = (J\nu_\mu)_j &= -\nabla f_j(x)^T W_\alpha^{-1} J^T \mu \\ &= -\sum_{i=1}^2 \mu_i \nabla f_j(x)^T W_\alpha^{-1} \nabla f_i(x). \end{aligned} \tag{3.29}$$

By the strict convexity of the objectives it follows that W_α^{-1} is positive definite. Further, since x is a KKT point of a BOP, the two objective gradients point into opposite directions. That is, there exists a $c < 0$ such that $\nabla f_2(x) = c\nabla f_1(x)$. Thus, we obtain for the first component of d

$$\begin{aligned} d_1 &= -\mu_1 \nabla f_1(x)^T W_\alpha^{-1} \nabla f_1(x) - \mu_2 \nabla f_1(x)^T W_\alpha^{-1} \nabla f_2(x) \\ &= -\mu_1 \underbrace{\nabla f_1(x)^T W_\alpha^{-1} \nabla f_1(x)}_{>0} - \mu_2 \underbrace{c}_{<0} \underbrace{\nabla f_1(x)^T W_\alpha^{-1} \nabla f_1(x)}_{>0}. \end{aligned} \tag{3.30}$$

Evaluating the choices for μ in (3.28) we see that

$$d_1 = c_1 \mu_1, \tag{3.31}$$

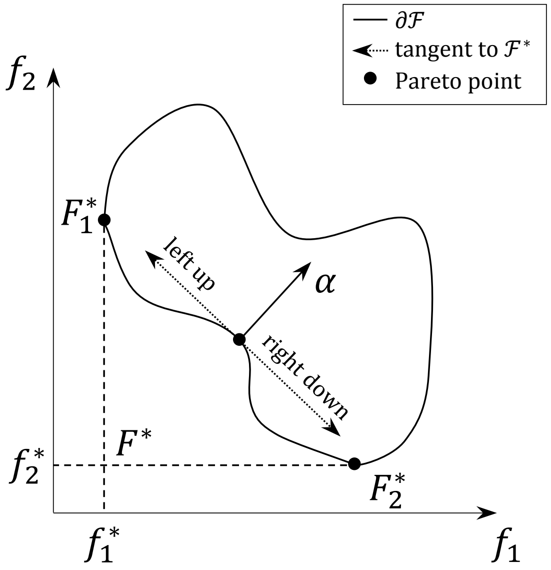


Figure 3.2: For BOPs, we can orientate the continuation into two possible ways: to minimize the first objective (left up), or to minimize the second one (right down).

where $c_1 < 0$. Analog, we obtain $d_2 = c_2\mu_2$, for a $c_2 < 0$. Since $d \neq 0$ it follows by (3.14) that a movement along the linearized Pareto front is performed. More precisely, for $\mu^{(1)}$ we obtain a 'left up' movement along the (local) Pareto front and for $\mu^{(2)}$ a 'right down' movement orthogonal to α. See Figure 3.2 for a graphical description.

Finally, to obtain the predictor $p = x + t\nu_\mu$ (for a given μ) at the current KKT point x, we need to select the step size t. In this study we are particularly interested in an evenly distributed set of solutions along the Pareto front. That is, at least for two consecutive solutions $\tilde{x}$ and x we want that

$$\|F(\tilde{x}) - F(x)\| \approx \tau, \tag{3.32}$$

where $\tau > 0$ is a user specified value. For this, we follow the suggestion made in [28] and take the step size

$$t = \frac{\tau}{\|J\nu_\mu\|}. \tag{3.33}$$

3.2 Corrector

The goal of the corrector phase is to ensure that the resulting solution is on the efficient set (i.e., it is at least a KKT point). However, a predictor cannot—in general—guarantee even local optimality, unless of course, we are following a flat Pareto set. Therefore, our new target is to bring (or correct) that predicted point back to the solution manifold which is certainly *not* an easy assignment. Nevertheless, due to the smoothness assumptions we consider in this study, a good property of predictors is that they remain close to the Pareto set provided that we do not move *too far* along the tangent. Then, it seems that the set of methods that fit best in this context are those based on descent directions (see Chapter 2) since they do not use any ordering or scalarization of the objectives avoiding the use of extra parameters. Moreover, they are exclusively designed to find only one optimal solution—hopefully the *closest* one to the initial point—so running one of these algorithms starting at the predictor promises to be an efficient alternative to find such a (corrected) optimal solution. Here, we have decided to take the Newton method [16] for unconstrained MOPs as corrector since it is (to our knowledge) the only one that can handle second-order information. Another option could be minimizing the norm of the map $\tilde{F}$ as done e.g. in [28] but one problem of this choice is the tight coupling between parameter and weight space and the possible convergence to non-optimal solutions (e.g., one with negative Lagrange multipliers or that maximizes the objectives).

Having made our selection, we dedicate this section to several details related to the implementation of the Newton method. As we saw in the previous chapter, the Newton direction is defined as the solution of

$$\begin{array}{ll} \min\limits_{(\nu,\delta)\in\mathbb{R}^n\times\mathbb{R}} & \delta \\ \text{s.t.} & \nabla f_j(x)^T\nu + \frac{1}{2}\nu^T\nabla^2 f_j(x)\nu \leq \delta, \qquad j = 1,\dots,k, \end{array} \tag{3.34}$$

where δ serves as a measure of the expected decrease in objective space produced by a line search in direction ν in parameter space. Particular attention should be paid to the additional variable δ which plays an important role here since it is used in the step length control and the stopping criteria. More precisely, an acceptable step size may be decided by a backtracking procedure based on a bisection strategy together with a modification of the (componentwise) Armijo condition. Basically, the idea is to take a $t > 0$ such that it simultaneously satisfies

$$F(x + t\nu) \leq F(x) + ct\delta e, \tag{3.35}$$

where $e \in \mathbb{R}^k$ is the vector of all ones and $0 < c < 1$ is a constant (generally taken as $c = 0.1$). In case the above condition is not met by the current value of t, it should be divided by two and, as a recommendation, the initial step length $t = 1$ should always be taken. Finally, the method is stopped when no more progress is expected in objective space (i.e. $\delta = 0$ which implies that $\nu = 0$). A more relaxed stopping criteria can be defined as

$$\delta \geq -\varepsilon, \tag{3.36}$$

where $\varepsilon > 0$ is a small tolerance (generally taken as $\varepsilon = 5\sqrt{\text{eps}}$). In [16], it has been shown that under suitable local assumptions (the objectives are strictly convex and the second derivatives are Lipschitz continuous) full Newton steps are always accepted and the generated sequence yields quadratic convergence toward a KKT point[1].

[1] The same rate of convergence is also observed by applying a Gauss-Newton or Levenberg-Marquardt method to minimize the norm of the map $\tilde{F}$ as done e.g. in [28].

We will, in the following, proceed as one step of a standard line search strategy: starting at a given point $x \in \mathbb{R}^n$, we will focus first in determining a search direction $\nu \in \mathbb{R}^n$, and later in computing a new iteration point by performing a suitable step ($t > 0$) along the given direction (unless, of course, the stop condition is satisfied). Our first task is thus to solve the subproblem (3.34) that we will call Newton direction subproblem (NDS). Note that this problem is in the category of quadratically constrained quadratic programming (QCQP), o more specifically, quadratically constrained linear programming (QCLP).[2] One way to handle such problems is by shifting them to second-order cone programming (SOCP) to be solved by specialized primal-dual interior point methods. Recall that SOCP focuses on minimizing a linear function subject to second-order (Lorentz) cone constraints. Hence, this area deals with problems of the form

$$\begin{aligned} &\min_{x\in\mathbb{R}^n} f^T x \\ &\text{s.t.} \quad \|A_j x + b_j\| \le c_j^T x + d_j, \qquad j = 1,\dots,k, \end{aligned} \tag{3.37}$$

where $f, c_j \in \mathbb{R}^{n+1}$, $A_j \in \mathbb{R}^{(n_j-1)\times(n+1)}$, $b_j \in \mathbb{R}^{n_j-1}$ and $d_j \in \mathbb{R}$. Consequently, our aim is now to transform the NDS into the form required by (3.37). To accomplish this, we define the minimization problem

$$\begin{aligned} &\min_{(\nu,\delta)\in\mathbb{R}^n\times\mathbb{R}} \Big(0,\dots,0,1\Big)\begin{pmatrix}\nu\\ \delta\end{pmatrix} = \delta \\ &\text{s.t.} \quad \left\| \begin{pmatrix} \frac{1}{2}\nabla f_j^T & -\frac{1}{2} \\ \frac{1}{\sqrt{2}}L_j^T & 0 \end{pmatrix}\begin{pmatrix}\nu\\ \delta\end{pmatrix} + \begin{pmatrix}\frac{1}{2}\\ 0\end{pmatrix} \right\| \le \Big(-\tfrac{1}{2}\nabla f_j^T, \tfrac{1}{2}\Big)\begin{pmatrix}\nu\\ \delta\end{pmatrix} + \frac{1}{2}, \quad j = 1,\dots,k, \end{aligned} \tag{3.38}$$

where $L_j L_j^T = \nabla^2 f_j(x)$ and $\nabla f_j = \nabla f_j(x)$. Note that (3.38) belongs to SOCP with $f = (0,\dots,0,1)^T \in \mathbb{R}^{n+1}$,

$$A_j = \begin{pmatrix} \frac{1}{2}\nabla f_j^T & -\frac{1}{2} \\ \frac{1}{\sqrt{2}}L_j^T & 0 \end{pmatrix} \in \mathbb{R}^{(n+1)\times(n+1)},$$

$b_j = (^1\!/_2, 0, \dots, 0)^T \in \mathbb{R}^{n+1}$, $c_j = (-^1\!/_2\nabla f_j^T, ^1\!/_2)^T \in \mathbb{R}^{n+1}$, and $d_j = {}^1\!/_2 \in \mathbb{R}$. Note also that both the NDS and the problem (3.38) have the same objective function. Then, the equivalence between them depends on the equivalence between both sets of constraints as shown below:

[2] Actually, every QCQP can be transformed to a QCLP by the standard trick of adding one more variable and one more constraint.

$$\begin{aligned}
\left\| \begin{pmatrix} \frac{1}{2}\nabla f_j^T & -\frac{1}{2} \\ \frac{1}{\sqrt{2}} L_j^T & 0 \end{pmatrix} \begin{pmatrix} \nu \\ \delta \end{pmatrix} + \begin{pmatrix} \frac{1}{2} \\ 0 \end{pmatrix} \right\| &\leq \left(-\tfrac{1}{2}\nabla f_j^T, \tfrac{1}{2}\right) \begin{pmatrix} \nu \\ \delta \end{pmatrix} + \frac{1}{2} \\
\left\| \begin{pmatrix} \frac{\nabla f_j^T \nu - \delta + 1}{2} \\ \frac{1}{\sqrt{2}} L_j^T \nu \end{pmatrix} \right\| &\leq \frac{-\nabla f_j^T \nu + \delta + 1}{2} \\
\frac{(\nabla f_j^T \nu - \delta + 1)^2}{4} + \frac{1}{2}\nu^T L_j L_j^T \nu &\leq \frac{(-f_j^T \nu + \delta + 1)^2}{4} \\
\frac{1}{2}\nu^T \nabla^2 f_j(x)^T \nu &\leq \delta - \nabla f_j(x)^T \nu \\
\nabla f_j(x)^T \nu + \frac{1}{2}\nu^T \nabla^2 f_j(x)^T \nu &\leq \delta.
\end{aligned}$$

One important detail here is that in order to cast the NDS to SOCP, it is necessary to perform a Cholesky decomposition of all Hessians to obtain the set of $L_j \in \mathbb{R}^{n \times n}$ such that $L_j L_j^T = \nabla^2 f_j(x)$ for all $j = 1, \ldots, k$. However, in practical implementations of Newton methods it is a common practice to transform the Hessian to keep it positive definite. In our context, the positive definiteness of the Hessians is also a requirement for the NDS to be convex and have a unique minimizer. One way to achieve such transformations is by means of modified Cholesky decompositions, so it may be possible that Hessian factorizations be anyway a necessary task.[3] With this purpose, we choose the algorithm of Gill, Murray and Wright [55].

Another aspect to have in mind when solving the NDS is that we need the Lagrange multipliers related to the corrector x^* for the computation of the tangent to the Pareto set and subsequently the next predictor(s) (see Equation (3.8)). Recall that the dual problem of (3.37) is given by

$$\begin{aligned}
\max\ & -b^T u - d^T v \\
\text{s.t.}\ \ & A^T u + C^T v = f, \\
& \|u_j\| \leq v_j, \qquad j = 1, \ldots, k,
\end{aligned} \tag{3.39}$$

with variables $u^T = (u_1^T, \ldots, u_k^T)$, where $u_j \in \mathbb{R}^{n_j - 1}$ and $v \in \mathbb{R}^k$. Further, it is $A^T = (A_1^T, \ldots, A_k^T)$ and $C^T = (c_1, \ldots, c_k)$. Here, it is easy to see (from the first-order optimality conditions) that some of the dual variables of the NDS at x^* coincide with the multipliers coming from the KKT conditions of the considered MOP at x^*. The same also applies to the dual variables of the problem (3.38) since (as shown above) it is equivalent to the NDS. More precisely, at the last iteration of the Newton method, one can set

$$\alpha_j = v_j, \qquad j = 1, \ldots, k, \tag{3.40}$$

which are required for computing (later) tangent vectors to the Pareto set at x^* (see Equation (3.8)). Alternatively, one can deal with the dual problem (3.39) and ask

[3] Hessian modifications do not guarantee the convergence of the Newton method. Nevertheless, such modifications do ensure that the resulting ν is a descent direction provided that x is noncritical.

the solver for the primal variables ν and δ.[4] In our implementations, we have chosen the latter approach since (3.39) fits better in the standard input accepted by SeDuMi [98, 99], the solver of our preference. There are other options as SDPT3 [100, 101] but in our experiments we have found that SeDuMi is faster, though, SDPT3 is reported as more reliable. Another non-free software packages that can deal with SOCP are Gurobi [102] and MOSEK [103]. Finally, an alternative much easier to use is CVX [104, 105] that was designed as a modeling system for convex programming which does not require a fixed format for the input. CVX is capable to automatically convert the problem from its original form (as e.g. the form of the NDS) into the format required for any of the solvers we previously mentioned.[5] However, we only used this choice at the earlier stages (as a proof of concept) since this is the slowest option probably due to the *on the fly* conversion feature.

Our main concern when deciding to use the Newton method as corrector was the computational cost of one iteration of this method compared to one iteration of the Gauss-Newton or Levenberg-Marquardt methods used to minimize the norm of $\tilde{F}$. In general, primal-dual interior point methods for SOCP are slower, although they can be solved as well in polynomial time. Strictly speaking, no formal asymptotic bounds for the complexity of SOCP optimizers are usually available and, if they exist, are overly pessimistic. In practice, the treatment of special problem structures (e.g., sparsity) often allows faster running times and, in general, the observed speed is much better than the theoretical worst case. Moreover, no formalism has been established so far about the asymptotic lower bound for such methods. For all of these reasons and considering that the Newton method has better theoretical properties than solving the system of equations $\tilde{F}$, we have decided to use it as corrector and consider the computational cost in terms of function evaluations. A more thorough study on this topic is encouraged for a future work.

Once that we have the search direction, we can compute a new iteration point $\bar{x} \in \mathbb{R}^n$ such that

$$\bar{x} = x + t\nu, \qquad t > 0, \tag{3.41}$$

and continue in this manner until the stop condition is satisfied. Due to numerical issues, it may be convenient to set a maximum number of iterations $N \in \mathbb{N}$ (where $\mathbb{N}$ stands for the set of strictly positive natural numbers). Furthermore, we can also establish a threshold ϵ (that we take as ϵ = eps) to stop the execution when the absolute change in parameter space is unimportant. No further discussion concerning the implementation of the step length control or the stopping criteria is considered necessary so we immediately proceed to illustrate the pseudo code of the Newton method for unconstrained MOPs in Algorithm 1. Following, in Algorithm 2, we show the pseudo code of a bisection backtracking strategy based on the Armijo condition.

[4] Recall that in SOCP the dual of the dual is the primal. Then, to obtain the final expression of the dual problem, we only need to substitute A, b, C and d in (3.39) by their formulas given above.

[5] The conversion can only be successfully achieved if the convexity of the problem can be verified.

Algorithm 1 Newton method for unconstrained MOPs

```
Require: $\bar{x} \in \mathbb{R}^n$, $\varepsilon > 0$, $N \in \mathbb{N}$, $\epsilon > 0$
Ensure: KKT point $x^*$ with associated weight $\alpha \in \mathbb{R}^k$ (if not terminated in Line 2)
 1: set $i = 0$
 2: while $i < N$ and $\|\bar{x} - x\| > \epsilon$ do                      ▷ main loop
 3:     set $x = \bar{x}$
 4:     solve the subproblem (3.38) to obtain $\nu$ and $\delta$, and    ▷ search direction
 5:     ask the solver for the dual variables to set $\alpha = v$, or
 4:     solve the dual subproblem (3.39) to set $\alpha = v$, and
 5:     ask the solver for the primal variables $\nu$ and $\delta$
 6:     if $\delta \geq -\varepsilon$ then                              ▷ stop condition
 7:         set $x^* = x$ and return
 8:     else
 9:         obtain a suitable step length $t > 0$ by Algorithm 2        ▷ line search
10:         compute $\bar{x} = x + t\nu$ and set $i = i + 1$            ▷ next iteration
11:     end if
12: end while
```

Algorithm 2 Line search based on the Armijo condition with bisection

```
Require: $\bar{t} > 0$, $\nu \in \mathbb{R}^n$, $\delta < 0$, $c \in (0,1)$, $\epsilon > 0$
Ensure: $t^* \in \mathbb{R}_+$ satisfiying the Armijo condition
while $\bar{t} > \epsilon$ do                                           ▷ main loop
    set $t = \bar{t}$
    if $F(x + t\nu) \leq F(x) + ct\delta e$ then                        ▷ Armijo condition
        set $t^* = t$ and return
    else
        compute $\bar{t} = t/2$                                         ▷ bisection
    end if
end while
set $t^* = 0$
```

3.3 The Pareto Tracer Method

We are now in the position to put predictor and corrector methods together into a continuation algorithm that we call Pareto Tracer (PT). The resulting pseudo code is presented below in Algorithm 3. As the name suggests, the PT is designed to trace the curve (or manifold) of Pareto points of a MOP. The procedure perfectly fits into the category of PC methods where arc length approximations are used to provide a discretization of the desired set (Line 10). It also follows the spirit of classical continuation (as done e.g. in [28] and [29]) but goes beyond of purely solving the KKT system of equations.[6] Additionally, the method seeks for a separation of decision and weight space which allows to compute predictors in a more efficient manner. As a by-product, a new set of multipliers $\mu_1, \ldots, \mu_k$ is introduced with the role of steering the search. If, for instance, uniform spread of points on the Pareto front is sought, we may want to obtain predictors such that the corresponding movements in objective space are *as well distributed as possible* (e.g., they are orthogonal to each other (Line 6)). At any case, by computing the appropriate orientation vector μ (Line 8), we can steer the search into any direction given in objective space (lines 9 and 11). Another difference of the PT with the proposals in [28] and [29] is the election of the Newton method as corrector instead of directly solving $\tilde{F}$ (lines 1 and 13). As discussed in the previous section, this choice is not based on an improvement of the computational complexity of the corrector algorithm. Our intention here is to investigate other possibilities that fit better into the context of optimization since—as pointed out before—the zero set of the map $\tilde{F}$ contains solutions that are not even local optima.

Algorithm 3 Pareto Tracer method for unconstrained MOPs

```
Require: $p \in \mathbb{R}^n$, $\tau > 0$
Ensure: finite size approximation of $\mathcal{P}$ and $\mathcal{F}^*$
 1: starting at $p$, obtain $x_0$ and $\alpha_0$ by Algorithm 1                ▷ first corrector
 2: set $\mathcal{P} = \mathcal{P} \cup x_0$, $\mathcal{F}^* = \mathcal{F}^* \cup F(x_0)$, and $\mathcal{W} = \mathcal{W} \cup \alpha_0$
 3: set $l = 1$
 4: while $l \leq |\mathcal{P}|$ do                                            ▷ main loop
 5:     set $x_0 = \mathcal{P}_l$ and $\alpha_0 = \mathcal{W}_l$
 6:     compute $d_i, i = 1, \ldots, k-1$, as in (3.24)                       ▷ objective directions
 7:     for $i = 1, \ldots, k-1$ do
 8:         compute $\mu_{d_i}^+$ as in (3.18)                                ▷ orientation vector
 9:         compute $\nu_i = \nu_{\mu_{d_i}}$ as in (3.26)                    ▷ tangent vector
10:         compute $t_i$ as in (3.33)                                        ▷ step length
11:         compute $p_{i_1} = x_0 + t_i\nu_i$ and $p_{i_2} = x_0 - t_i\nu_i$  ▷ predictors
12:         for $i_j = i_1, i_2$ do
13:             starting at $p_{i_j}$, obtain $x_{i_j}$ and $\alpha_{i_j}$ by Algorithm 1   ▷ correctors
14:             if $F(x_{i_j}) \notin \mathcal{F}^*$ then
15:                 set $\mathcal{P} = \mathcal{P} \cup x_{i_j}$, $\mathcal{F}^* = \mathcal{F}^* \cup F(x_{i_j})$, and $\mathcal{W} = \mathcal{W} \cup \alpha_{i_j}$
16:             end if
17:         end for
18:     end for
19:     set $l = l + 1$
20: end while
```

[6] Actually, in [29], the map $\tilde{F}$ is based on the Fritz John conditions of optimality.

In addition, the PT has to deal with a well-known problem of PC methods—in particular for $k > 2$—consisting of identifying the part of the solution manifold that is already covered by the algorithm. To achieve this, it is crucial to provide an efficient data structure to store the computed solutions since the time complexity of verifying whether or not a solution belongs to the set (Line 14) is a critical part of the overall performance of the method. To this end, we choose the data structure proposed in [34] which is based on a subdivision of the search space into boxes. Other ideas can be found in [106, 107, 108, 87, 88] and references therein. The only difference of our approach with the one in [34] is that we construct the partition in objective space and not in parameter space. The change is realized because we aim in this work for an evenly spread approximation of the Pareto front whereas in [34] a covering of the Pareto set is considered. In the following, we explain some of the details regarding the implementation of this data structure. Let's suppose that every function is restricted to a certain range in objective space

$$l_j^{\mathcal{F}} \leq f_j(x) \leq u_j^{\mathcal{F}}, \qquad j = 1, \ldots, k. \tag{3.42}$$

The objective space is then given by

$$Q_{\mathcal{F}} = [l_{\mathcal{F}}, u_{\mathcal{F}}] = [l_1^{\mathcal{F}}, u_1^{\mathcal{F}}] \times [l_2^{\mathcal{F}}, u_2^{\mathcal{F}}] \times \cdots \times [l_k^{\mathcal{F}}, u_k^{\mathcal{F}}] \in \mathbb{R}^{k \times 2}. \tag{3.43}$$

Under this hypothesis, we can tessellate the space into *small* boxes where each box $\mathcal{B} \subset \mathbb{R}^k$ is represented by a center $c \in \mathbb{R}^k$ and a radius $r \in \mathbb{R}^k$ such that

$$\mathcal{B}(c, r) = \left\{ F(x) \in Q_{\mathcal{F}} : |f_j(x) - c_j| \leq r_j, \quad \forall j = 1, \ldots, k \right\}. \tag{3.44}$$

The underlying idea of this approach is to maintain a collection of boxes where each box is restricted to contain only one solution, and where the covering box $\mathcal{B}(F(x))$ of a given solution $F(x)$ can be unequivocally determined. Thus, a box can be represented by both $\mathcal{B}(c, r)$ and $\mathcal{B}(F(x))$. This strategy leads to a finite size discretization of the objective space with two important consequences: (i) the possibility of a good spread of solutions and (ii) the possibility of determining (in a finite amount of time) whether or not a solution is already computed, that is, whenever its corresponding box belongs to the collection. For this purpose, however, the definition given in (3.44) is useless since several boxes can share a common boundary. Thus, in order to unequivocally determine the covering box of a given solution, we redefine (3.44) as

$$\mathcal{B}(c, r) = \left\{ F(x) \in Q_{\mathcal{F}} \left| \begin{matrix} c_j - r_j \leq f_j(x) < c_j + r_j, \text{ if } c_j + r_j < u_j^{\mathcal{F}} \\ c_j - r_j \leq f_j(x) \leq c_j + r_j, \text{ if } c_j + r_j = u_j^{\mathcal{F}} \end{matrix}, \ \forall j = 1, \ldots, k \right. \right\}. \tag{3.45}$$

Let's now consider a partition of the space starting at $\mathcal{B}(c, r) = Q_{\mathcal{F}}$ with radius $r = (u_{\mathcal{F}} - l_{\mathcal{F}})/2$ and center $c = l_{\mathcal{F}} + r$. A box $\mathcal{B}(c, r)$ can be subdivided with respect to the i-th coordinate leading to two new boxes that completely cover the old one. That is, $\bar{\mathcal{B}}(\bar{c}, \hat{r})$ and $\mathcal{B}^{+}(c^{+}, \hat{r})$ where

$$\hat{r}_j = \begin{cases} r_j, & i \neq j \\ ^{r_j}/_{2}, & i = j \end{cases} \quad \text{and} \quad c_j^{\mp} = \begin{cases} c_j, & i \neq j \\ c_j \mp {}^{r_j}/_{2}, & i = j \end{cases}. \tag{3.46}$$

Subsequently, after h subdivisions of the current set of boxes and by cyclically choosing the reference coordinate, i.e.,

$$j_i = (i-1) \bmod k + 1, \qquad i = 1, \dots, h, \tag{3.47}$$

we finish with a partition of $Q_{\mathcal{F}}$ into boxes of radius

$$r_j = \begin{cases} \dfrac{u_j^{\mathcal{F}} - l_j^{\mathcal{F}}}{2^{\left\lfloor \frac{h-1}{k} \right\rfloor + 2}}, & j \leq (h-1) \bmod k + 1 \\ \dfrac{u_j^{\mathcal{F}} - l_j^{\mathcal{F}}}{2^{\left\lfloor \frac{h-1}{k} \right\rfloor + 1}}, & j > (h-1) \bmod k + 1 \end{cases}, \qquad j = 1, \dots, k, \tag{3.48}$$

where for every point $F(x) \in Q_{\mathcal{F}}$ there exists exactly one covering box with center in

$$c_j = \begin{cases} l_j^{\mathcal{F}} + 2r_j \left\lfloor \dfrac{f_j(x) - l_j^{\mathcal{F}}}{2r_j} \right\rfloor + r_j, & f_j(x) \notin u_j^{\mathcal{F}} \\ l_j^{\mathcal{F}} + 2r_j \left\lfloor \dfrac{f_j(x) - l_j^{\mathcal{F}}}{2r_j} \right\rfloor - r_j, & f_j(x) \in u_j^{\mathcal{F}} \end{cases}, \qquad j = 1, \dots, k. \tag{3.49}$$

We can thus maintain a collection $\mathcal{B}(\mathcal{F}^*)$ to store the boxes corresponding to the solution's images in $\mathcal{F}^*$. An efficient implementation of this data structure can be achieved by considering $\mathcal{B}(\mathcal{F}^*)$ a binary tree where each level corresponds to one subdivision step and each leaf to one box. For a better understanding Figure 3.3 shows a binary tree representation of five boxes obtained after three subdivision steps. Note that each step is completely determined by the tree structure and the initial box $Q_{\mathcal{F}}$.[7] Considering $\mathcal{B}(Q_{\mathcal{F}}, h)$ the full set of boxes obtained after h subvidisions, it is not hard to see that $\mathcal{B}(Q_{\mathcal{F}}, h)$ is defined by a complete binary tree of height h [109] and that any arbitrary tree $\mathcal{B}(\mathcal{F}^*) \subset \mathcal{B}(Q_{\mathcal{F}}, h)$ is usually not complete. Moreover, using this scheme, the memory requirements seem to grow only linearly in the number of objectives and solutions. On the other hand, the operation of insertion as well as verifying whether a box (leaf) belongs to the set (tree) can be implemented in $\mathcal{O}(h)$. Algorithm 4 shows the pseudo code of a method that combines the operations 'insert' and 'contains' into a new one called 'recover'. The method ensures that the covering box of a given solution is added to the tree (recovered) and returns whether the corresponding leaf was *really* inserted or already belonged to the collection. Then, we can replace the condition of the if-statement in Line 14 of Algorithm 3 by an appropriate call to Algorithm 4. An additional optimization here could be to bypass predictor points when their associated boxes have previously been considered (i.e., they already contain a corrector). Considering that predictors and correctors are supposed to be close, this will avoid the computation of a corrector that may be later discarded by the 'recover' algorithm in Line 14.

[7] We do not actually need to store any information at the nodes of the tree since the tree structure by itself is good enough to represent the entire collection of boxes.

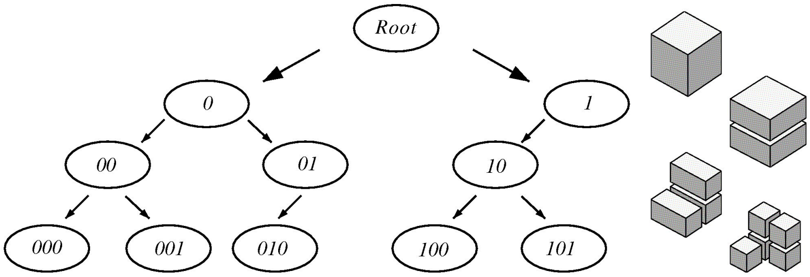


Figure 3.3: The data structure used for the representation of the solution set.

Algorithm 4 Recover algorithm

```
Require: binary tree $\mathcal{B}(\mathcal{F}^*) \subset \mathcal{B}(Q_{\mathcal{F}}, h)$, $F(x) \in \mathbb{R}^k$
Ensure: $\mathcal{B}(F(x)) \in \mathcal{B}(\mathcal{F}^*)$ is 'recovered'
starting at the root of $\mathcal{B}(\mathcal{F}^*)$
for $i = 1, \ldots, h$ do                                  ▷ main loop
    compute $j = (i-1) \bmod k + 1$                        ▷ current coordinate
    compute $c_j = l_j^{\mathcal{F}} + (u_j^{\mathcal{F}} - l_j^{\mathcal{F}})/2$   ▷ current center
    if $f_j(x) < c_j$ then                                 ▷ left branch
        if there is no left child then
            create a new left child
        end if
        move one level down to the left child
        set $u_j^{\mathcal{F}} = c_j$                      ▷ shrinking the space
    else                                                   ▷ right branch
        if there is no right child then
            create a new right child
        end if
        move one level down to the right child
        set $l_j^{\mathcal{F}} = c_j$                      ▷ shrinking the space
    end if
end for
compute $r = (u_{\mathcal{F}} - l_{\mathcal{F}})/2$ and $c = l_{\mathcal{F}} + r$   ▷ covering box of $F(x)$
return $\mathcal{B}(F(x)) = \mathcal{B}(c, r)$ and $\begin{cases} 1 & \text{if a new leaf was created} \\ 0 & \text{if the corresponding leaf already existed} \end{cases}$
```

Another aspect related to the performance of the PT method has to do with the information we keep or recompute regarding the current set of solutions. Note that the set of Pareto points $\mathcal{P}$ is treated as a queue where the solutions are being stored as they are being computed. Initially, $\mathcal{P}$ contains only one point (Line 2) and at each iteration $l = 1, 2, \dots,$ at most $m = 2(k-1)$ new points—obtained after computing m predictors and their respective correctors starting at $\mathcal{P}_l$—are added to the queue (Line 15). For each point in $\mathcal{P}$, the corresponding function and α values are stored respectively in $\mathcal{F}^*$ and $\mathcal{W}$ to be used at the proper time. However, first and second-order information will also be required for the computation of new solutions. Here, we should decide whether to store the Jacobian and Hessian matrices of the solutions held in $\mathcal{P}$, or alternatively recompute them again when requested (e.g., to compute tangent vectors at the current solution (Line 9)). In our implementations, we utilized the first choice since our goal so far is to reduce the number of function evaluations.

The last task to complete the PT algorithm is to determine the value of h (i.e., the number of subdivisions of the objective space). Since we aim in this work for an evenly spread distribution of solutions where the distance between neighbor points is approximately $\tau > 0$ (see Equation (3.32)), one strategy could be to perform the least number of subdivisions such that

$$\tau \geq 2 \max_{j=1,\dots,k} r_j = 2 \max r. \tag{3.50}$$

With this, our intention is to ensure—to the extent possible—that a new solution does not fall into the same box as the solution from which the local exploration departed. In the following, we will show that

$$h = k \left\lceil \log_2 \frac{\max\left(u_{\mathcal{F}} - l_{\mathcal{F}}\right)}{\tau} \right\rceil \tag{3.51}$$

satisfies such a requirement. First, observe that

$$\left\lfloor \frac{h-1}{k} \right\rfloor = \left\lfloor \left\lceil \log_2 \frac{\max\left(u_{\mathcal{F}} - l_{\mathcal{F}}\right)}{\tau} \right\rceil - \frac{1}{k} \right\rfloor = \left\lceil \log_2 \frac{\max\left(u_{\mathcal{F}} - l_{\mathcal{F}}\right)}{\tau} \right\rceil - 1. \tag{3.52}$$

Also, since h is a multiple of k, we have from (3.48)

$$2r = \frac{u_{\mathcal{F}} - l_{\mathcal{F}}}{2^{\left\lfloor \frac{h-1}{k} \right\rfloor + 1}}. \tag{3.53}$$

Then, combining (3.52) and (3.53), we get

$$2 \max r = \frac{\max\left(u_{\mathcal{F}} - l_{\mathcal{F}}\right)}{2^{\left\lfloor \frac{h-1}{k} \right\rfloor + 1}} \leq \frac{\max\left(u_{\mathcal{F}} - l_{\mathcal{F}}\right)}{\frac{\max(u_{\mathcal{F}} - l_{\mathcal{F}})}{\tau}} \leq \tau. \tag{3.54}$$

Yet, there is one more trick that can help to refine the results: standing at the box corresponding to the current solution, we can check the points existing in the neighboring boxes and bypass the result if it is *too close* to one of its neighbors. More precisely, two points $F(x)$ and $F(\hat{x})$ are considered to be close if

$$\|F(x) - F(\hat{x})\| < c\tau, \tag{3.55}$$

where $c \in (0, 1]$ is a small constant and $F(\hat{x}) \in \mathcal{N}(F(x))$. Further, $\mathcal{N}(F(x))$ denotes the set of points whose covering boxes have a common boundary with $\mathcal{B}(F(x))$. This situation occurs relatively frequently since the points are not necessarily located at the center of the boxes and may be near to the boundaries. For the sake of efficiency, we can also discard predictors p if there is some corrector x such that $F(x) \in \mathcal{N}(F(p))$ and $\|F(p) - F(x)\| < c\tau$. The best results—in terms of costs and distribution—were obtained in our experiments for a value of $c = 0.7$. However, this strategy comes with some drawbacks. First, the index of the solutions in $\mathcal{P}$ (respectively in $\mathcal{F}^*$) should be stored at the leafs of the tree to be able to identify the point existing inside a given box. Second, the number of neighboring boxes is exponential in k so this strategy is limited to problems with a few objectives.

Algorithm 5 Pareto Tracer method for unconstrained BOPs

```
Require: $p \in \mathbb{R}^n$, $\tau > 0$, $\zeta > 0$
Ensure: finite size approximation of $\mathcal{P}$ and $\mathcal{F}^*$
starting at $p$, obtain $x$ and $\alpha$ by Algorithm 1          ▷ first corrector
set $\mathcal{P} = \mathcal{P} \cup x$, and $\mathcal{F}^* = \mathcal{F}^* \cup F(x)$
for $j = 1, 2$ do
    set $\mu = \mu^{(j)}$ as in (3.28)                           ▷ orientation vector
    repeat
        compute $\tilde{\nu} = \nu_\mu$ as in (3.8)                ▷ tangent vector
        if $(J\tilde{\nu})_j < 0$ then                          ▷ orientation adjustment
            set $\nu = \tilde{\nu}$
        else
            set $\nu = -\tilde{\nu}$
        end if
        compute $t$ as in (3.33)                                 ▷ step length
        compute $p = x + t\nu$                                   ▷ predictor
        starting at $p$, obtain $x$ and $\alpha$ by Algorithm 1   ▷ corrector
        set $\mathcal{P} = \mathcal{P} \cup x$, and $\mathcal{F}^* = \mathcal{F}^* \cup F(x)$
    until $\alpha_j \geq 1 - \zeta$ or $f_j$ could not be improved   ▷ stop condition
end for
```

Finally, the representation of the solution set gets significantly simplified for BOPs: since the Pareto fronts of such problems are typically one-dimensional, there are only 'left up' or 'right down' movements to be considered in objective space. The idea is thus to perform first a continuation all along in the direction to optimize one of the objectives, and then (by reversing the orientation) a continuation all along in the direction to optimize the other objective. This ensures that the image of the new solution does not belong to $\mathcal{F}^*$ avoiding the use of additional data structures or any other strategy to verify this fact. In Section 3.1, we saw that if both objectives are convex, the choice of $\mu^{(1)}$ in (3.28) leads to a movement oriented to minimize the first function (left up), while the choice of $\mu^{(2)}$ ensures a movement in order to minimize the second one (right down). In case the convexity requirement cannot be guaranteed, we can implement the following orientation adjustment: let's suppose—without loss of generality—that an improvement according to f_1 is sought. Then, we can check at every step the condition

$$\nabla f_1(x)^T \nu < 0, \tag{3.56}$$

and reverse the orientation if it is not satisfied. Algorithm 5 shows the pseudo code of a version of the PT that has been optimized to deal exclusively with BOPs. Note that the orientation adjustment is implemented from Line 7 to Line 11. Another matter of importance in the context of BOPs is the selection of a stopping criterion for the continuation. When dealing with more than two objectives, the 'recover' algorithm provides such a criterion. Here, an application of the Newton method starting at a predictor that goes *beyond* the boundaries of the Pareto set/front will step back in order to reach (again) the manifold of optimal solutions. In this case, the obtained corrector will probably belong to a box that is already in $\mathcal{B}(\mathcal{F}^*)$ and consequently will be discarded. Then, the PT eventually stops when the neighborhood of all solutions in $\mathcal{P}$ has been explored (Line 4 of Algorithm 3). Since for the bi-objective case we obviously desire to avoid the overhead of additional data structures, we are forced to consider other alternatives to stop the method. One possibility (given that we have the Lagrange multipliers available) is to stop when

$$\alpha_j \geq 1 - \zeta, \qquad j \in \{1, 2\}, \tag{3.57}$$

where $\zeta > 0$ is a small value (that we take as $\zeta = 10^{-3}$) and j refers to the function we seek to minimize (see Line 16). There are other criteria that can be used to improve the results. For instance, we can verify whether the function that is supposed to decrease *actually* decreases, i.e.,

$$f_j(x) < f_j(\tilde{x}) - \varepsilon, \qquad j \in \{1, 2\}, \tag{3.58}$$

where $\varepsilon > 0$ is a small tolerance (e.g., $\varepsilon = 5\sqrt{\text{eps}}$) and $\tilde{x}$ is the previous solution. Otherwise, x is probably a weak optima and we stop. Additionally, we can check that the condition

$$f_i(x) \geq f_i(\tilde{x}), \qquad i \in \{1, 2\}, \tag{3.59}$$

is met all the way, where the i-th objective is the one that should increase. *Too short* steps may also be indicative that we are near to one of the extremes of the Pareto set/front (applicable to the general case) so we could also stop when

$$\|F(x) - F(\tilde{x})\| \leq c\tau, \tag{3.60}$$

where $c \in (0, 1)$ can be taken this time as e.g. $c = 0.1$. Further stopping criteria as well as degeneracy or singularity detection are left out of this study and recommended for a future work.

3.4 Hessian-free Realizations

As the reader has probably noted from the analysis above, one disadvantage of the PT is the requirement of the exact Hessians of all objective functions. Although second-order algorithms usually show good convergence rates, the issues related to the computation of Hessians are twofold. First of all, we may not have this information available and the use of approximations (e.g., by FD) could be extremely expensive in terms of function evaluations (our main concern regarding performance). Secondly, Hessian matrices demand a storage capacity of $\mathcal{O}(n^2)$ which is resource-intensive even for medium sized problems. In this section, we therefore face a new challenge: the development of Hessian-free realizations of the PT. In the following, we will first address different alternatives to compute predictors and will later on discuss the computation of correctors.

3.4.1 Predictor

The most basic mechanism to let Hessians out of the equation is to approximate them by the identity matrix ($I \in \mathbb{R}^{n \times n}$). By doing so, we get from (3.8)

$$\nu_\mu = -W_\alpha^{-1} J^T \mu = -\sum_{j=1}^{k} \alpha_j I J^T \mu = -J^T \mu. \tag{3.61}$$

Given $d \neq 0$, we can use (as in Section 3.1)

$$\mu_d^+ = -(JJ^T)^+ d, \tag{3.62}$$

which comes from (3.18) by using the same trick as before on W_α^{-1}. Then,

$$\nu_{\mu_d} = J^T (JJ^T)^+ d = J^+ d. \tag{3.63}$$

This means that the predictor directions coincide with the election made by the Directed Search algorithm proposed in [20], and will be a linear combination of the derivatives of the problem. Specifically for $k = 2$, we have at Pareto points: $\nabla f_1(x) = c \nabla f_2(x)$ for a value of $c < 0$. Then, it can be easily derived that the use of the equations above is equivalent to move in the direction of $-\nabla f_1(x)$ if we want the continuation oriented to minimize the first function, and otherwise, to move in the direction of $-\nabla f_2(x)$ if we need to improve the second objective. Recall from Chapter 2 that this strategy for BOPs has been used before in [36] and [30], from which (3.61) is a generalization. This choice of predictor directions is very attractive for several reasons: (i) it completely discards Hessians, (ii) the computations turn to be very efficient since only a matrix-vector multiplication is required[8], and (iii) under

[8] We would also need to compute the pseudo inverse of J if we desire to steer the search into a direction d given in objective space. However, this is not mandatory: the components of μ can be chosen e.g. at random such that they sum up to zero.

the assumption that $\mathrm{rank}(J) = k-1$, the Jacobian is not orthogonal to the linearized solution set so a movement toward the Pareto set is expected. In spite of this fact, however, the derivatives can be sometimes nearly orthogonal to the Pareto set. See Figure 3.4 for such examples. Thus, it seems more reasonable to take as predictor directions the secant between the last two points[9], i.e.,

$$\nu = \tilde{x} - x, \tag{3.64}$$

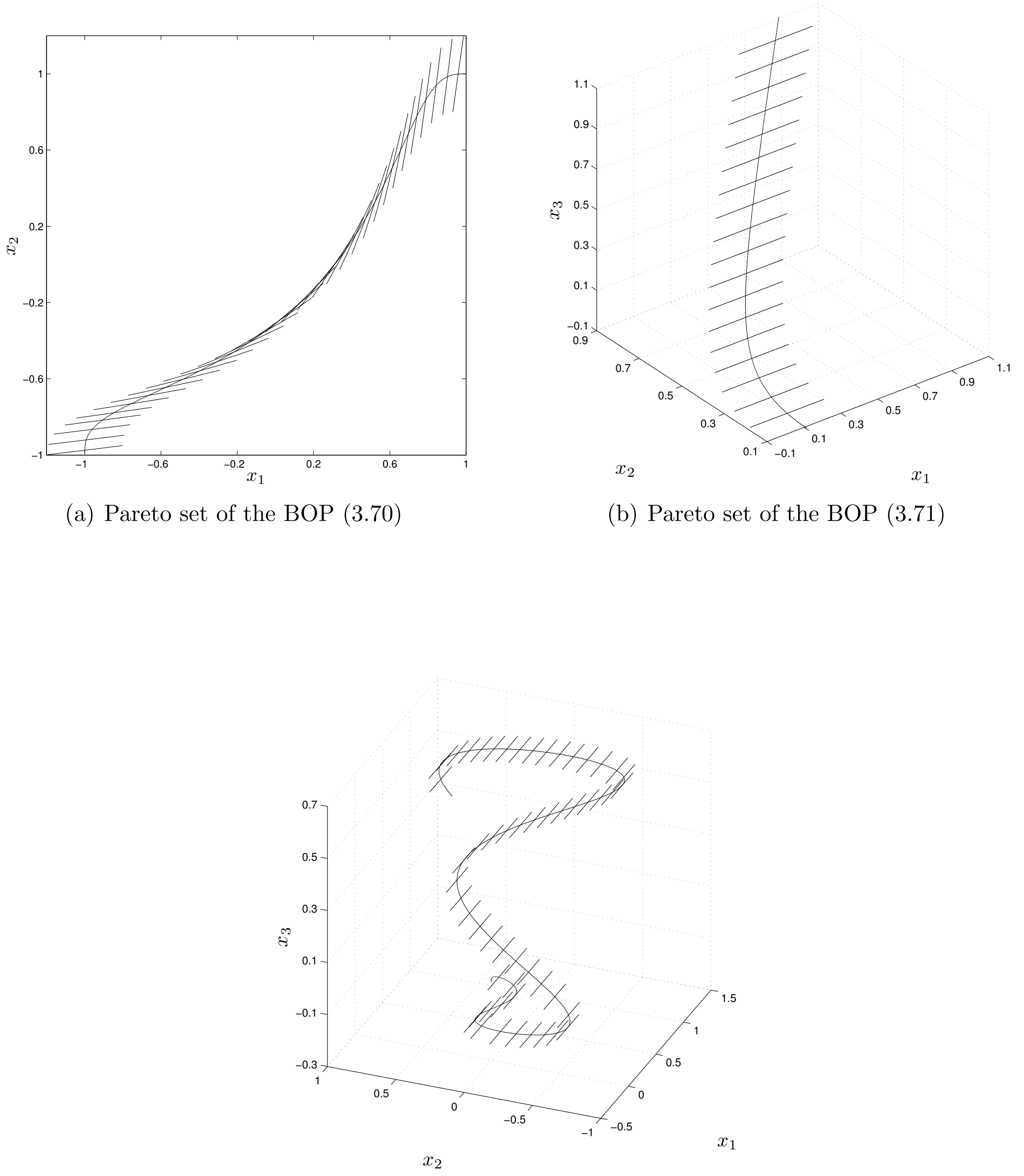
where x is the current corrector and $\tilde{x}$ is a previous point on the solution curve. With this, both direction and orientation come essentially for free while (implicitly) some curvature information is captured. Moreover, for MOPs with linear Pareto sets, derivatives and secants coincide with the direction of the solution set, so it seems that the use of secants is not worse than the use of derivatives as approximations of tangent vectors. Actually, in all our experiments, secant predictors lead to much better results than taking gradients, except for linear Pareto sets where the results were identical. However, extending the concept of secants for $k = 3$ is not a trivial task. Then, for comparison purposes, we define the steepest descent approximation of the tangent space as follows: for BOPs we take the secant as in (3.64), and for the general case we utilize (3.63) to obtain an approximate tangent direction.

The computation of the tangent space can also be made Hessian-free when utilizing elements from quasi-Newton methods [17, 47]. That is, at a point $x \in \mathbb{R}^n$, we replace the matrix W_α in the formulations of ν_{μ_d} and μ_d^+ (Equations (3.22) and (3.18), respectively) by

$$B_\alpha = \sum_{j=1}^{k} \alpha_j B_j \in \mathbb{R}^{n\times n}, \tag{3.65}$$

where $B_j \approx \nabla^2 f_j(x)$. In the first iteration, we choose $B_j = B_\alpha = I$ leading to $\nu_{\mu_d} = J^+ d$ as in (3.63). In subsequent iterations, we typically obtain better Hessian approximations due to the successive BFGS updates performed in the previous corrector steps if the QN method [17] is used instead of the Newton method (see the next heading). Figure 3.5 shows an example of an approximated tangent computed by utilizing the approximated Hessians B_1 and B_2 at a corrector point on the Pareto set of a BOP. We compared this choice of *quasi-Newton predictors* for BOPs with the use of *secant predictors* and obtained in all our test cases better results (fewer function evaluations) except for problems with linear Pareto sets where the results were the same (no corrector step was needed). However, the differences here between both choices of predictors were not so dramatic as the differences observed when comparing *secant* and *gradient predictors* (by using the formula (3.63)). These results encourage a further study on the use of secants for situations where efficiency has priority. Note, though, that for efficiency here we refer to asymptotic time complexity which would be reduced from $\mathcal{O}(n^3)$ to $\mathcal{O}(n)$ if we decide to use secants instead

[9] Initially, we have no option but to take derivatives.



(a) Pareto set of the BOP (3.70)

(b) Pareto set of the BOP (3.71)

(c) Pareto set of the BOP (3.73)

Figure 3.4: The derivatives of the objective functions along the Pareto set of three BOPs.

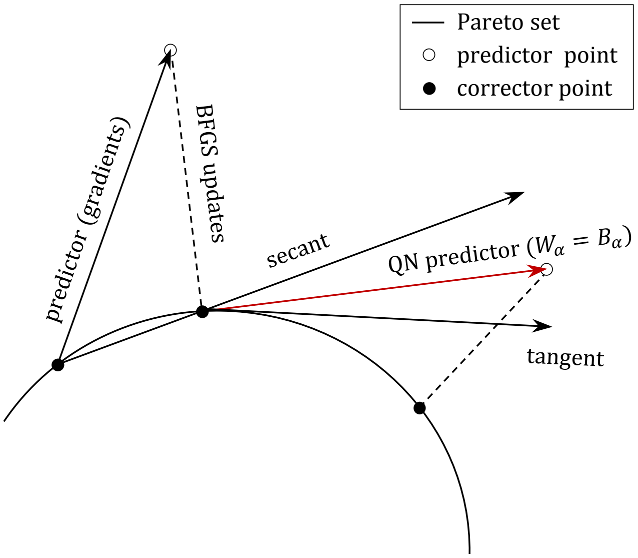


Figure 3.5: The second-order information gathered during the corrector phase is utilized to approximate the predictor direction. The more accurate the second-order information, the better the tangent estimation, and in consequence, the fewer corrector steps is expected.

of QN approximations. Nevertheless, this would not improve the overall performance of the continuation method since the corrector iterations clearly require more than $\mathcal{O}(n^3)$ to perform. In conclusion, the QN approximation of the tangent space will be given by the formulas (3.22) and (3.18) where W_α is replaced by B_α as in (3.65) and the approximated Hessians B_j, $j = 1, \ldots, k$, are obtained as a by-product of a QN method applied as the previous corrector.

3.4.2 Corrector

For the corrector phase we can (again) completely disregard the Hessian information and use the SD method [18] for MOPs. Note that the Newton direction subproblem becomes equivalent to

$$\begin{array}{lll} \min\limits_{(\nu,\delta)\in\mathbb{R}^n\times\mathbb{R}} & \frac{1}{2}\|\nu\|^2 + \delta & \\ \text{s.t.} & \nabla f_j(x)^T \nu \leq \delta, & j = 1, \ldots, k, \end{array} \tag{3.66}$$

if we approximate the Hessians as the identity matrix. Subproblem (3.66) is utilized by the SD method to compute the search direction (see Chapter 2 for more details). Further, the step length control and the stopping criteria remain the same as for the Newton version. This is also the case of the QN method [17], a more promising approach that utilizes BFGS updates to approximate Hessians. Here, the search direction subproblem is given by

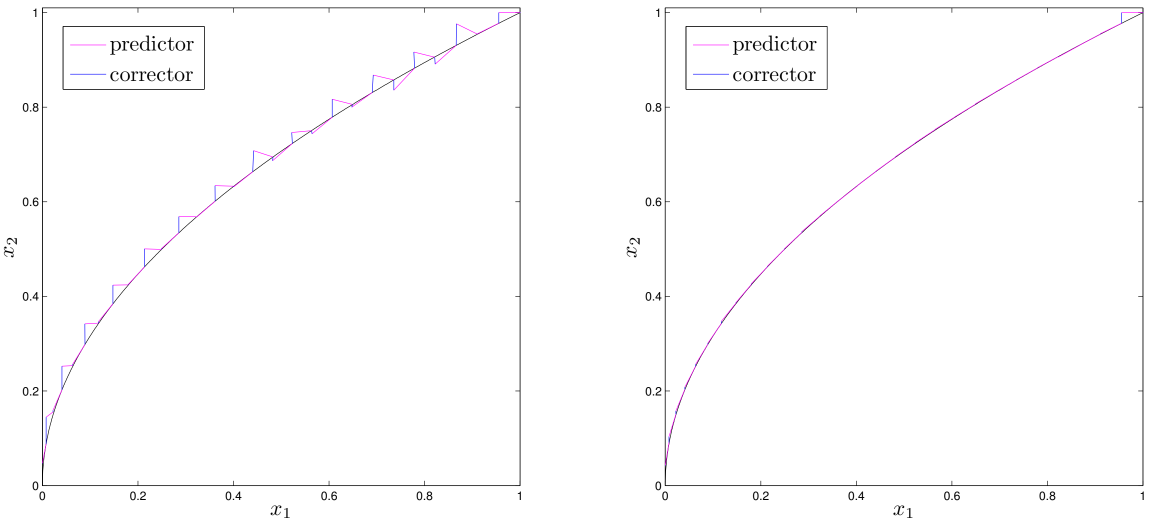
$$\begin{array}{ll} \min\limits_{(\nu,\delta)\in\mathbb{R}^n\times\mathbb{R}} & \delta \\ \text{s.t.} & \nabla f_j(x)^T\nu + \frac{1}{2}\nu^T B_j \nu \leq \delta, \qquad j = 1,\ldots,k, \end{array} \tag{3.67}$$

which differs only from the Newton direction subproblem by the use of $B_j \approx \nabla^2 f_j(x)$. Then, the current Hessians are updated at each iteration based on the information we have just computed to be used in the next step, i.e.,

$$\bar{B}_j = B_j - \frac{B_j s s^T B_j}{s^T B_j s} + \frac{y_j y_j^T}{y_j^T s}, \qquad j = 1,\ldots,k, \tag{3.68}$$

where $\bar{B}_j$ denotes the j-th updated Hessian at the new point $\bar{x}$, $s = \bar{x} - x$, and $y_j = \nabla f_j(\bar{x}) - \nabla f_j(x)$. As usual, in the first iteration we take the Hessians as the identity matrix. We will omit the pseudo code of these two methods since they are a small variation of Algorithm 1. For the SD version, we just replace the subproblem to solve by (3.66). Note also that the dual problem of (3.66) is given in Chapter 2 by (2.69). For the QN approach, we additionally have to introduce the BFGS updates at $\bar{x}$ after Line 10. The improvement of the QN method over the SD is that a superlinear convergence rate can be proven under some mild conditions while the SD method is at most linearly convergent. One disadvantage, though, is that (3.66) is a quadratic program simpler to solve than the QN direction subproblem (3.67) which belongs to QCLP. In addition, (3.66) guarantees a descent direction while (3.67) requires the positive definiteness of each B_j for the same purpose (including the uniqueness of the solution). As explained in the previous chapter for SOPs, the updated Hessian is forced to be positive definite by imposing the Wolfe conditions—(2.19) and (2.20)—on the step length control, but for MOPs it seems that these rules may be impossible to satisfy for all objectives together. We then solve the problem as in Section 3.2 for the Newton method: by applying a modified Cholesky factorization on each B_j to keep them positive definite. After all, recall also from Section 3.2 that in order to solve the quadratically contrained program (3.67) we have to shift it to SOCP. Thus, a decomposition of the form $B_j = L_j L_j^T$ for all $j = 1,\ldots,k$, will be anyway a requirement.

Given this scenario, it seems wiser to store and update the Cholesky factors L_j, $j = 1,\ldots,k$, rather than maintaining the full Hessians. Note that the operation of update can be done in $\mathcal{O}(n^2)$ [57, 110] compared to the $\mathcal{O}(n^3)$ required by the modified Cholesky decomposition [55]. However, the potential advantage of this optimization gets obscure since sooner or later we would have to multiply the Cholesky factors in order to compute the inverse of $B_\alpha = \sum_{j=1}^k \alpha_j B_j$ for the next predictor. The implementation and analysis of this and similar strategies has been left out of the scope of this thesis. In general, no improvement on the time complexity for the QN approach could be achieved over the Newton method. On the contrary, additional resources have to be employed for the computation of the k BFGS updates given in (3.68).



(a) Zigzag behavior resulting from QN correctors taking I as initial guess for the Hessians.

(b) Results of taking QN correctors with an updated initial guess for the Hessians.

Figure 3.6: Variants of the QN version of the PT on the BOP (3.71) with $n = 100$.

Figure 3.6 (a) shows one run of the QN version of the PT on the BOP (3.71). For this choice we observed a kind of zigzag behavior: whenever the predicted point is far from the curve, we have to perform many corrector steps and obtain at the end good approximations of the Hessians, which yields in consequence well-estimated predictors. Such exact predictors, however, may lead to just a few corrector iterations resulting in a relatively bad approximation of the Hessians, so we obtain this time a bad guess for the new predictor. Then, many corrector steps will be performed again leading to a good predicted point in the upcoming step, and so on. This phenomenon is mainly observed in problems with a large number of variables since starting at the identity matrix, n updates may be needed to approach the true Hessians [47]. As an alternative, the information gathered for the previous corrector can be utilized to update the Hessians at the current predictor. To be more precise, we apply the BFGS update (3.68) taking $\bar{x}$ as the current predictor ($\bar{x} = p$) and x as the previous corrector. With this, we expect to offer a better initial guess for the Hessians to the QN method employed as corrector. Furthermore, given that both points are close to each other, we expect that this strategy allows to keep the second-order information gained along all the continuation process. Figure 3.6 (b) shows the result of this variation in contrast to the choice of taking the identity matrix as an initial estimation for the Hessians in every corrector step. From the graphical results, it is easy to guess the starting point of the continuation. We will also omit the pseudo code of the SD and QN versions of the PT but would like to make one final remark: the Hessian updates at the predictor points should be placed after Line 11 in Algorithm 3 and after Line 13 in Algorithm 5 to be utilized by the following corrector.

3.5 Numerical Results

Here we present some numerical results of the PT and compare it against the method of Hillermeier [28] since both are close in spirit and deliver comparable results. The latter is not the case for other methods, mainly because their applications lead to approximations of different magnitude which hinders a fair comparison. For the PT we use the Newton, quasi-Newton, and steepest descent realizations called PT-N, PT-QN, and PT-SD in the following. The values of the parameters have been taken as specified when they were first introduced. On the other hand, our implementation of the method of Hillermeier follows strictly the instructions given in [28] explained also in Chapter 2. Additionally, we use the data structure presented in Section 3.3 to handle problems with more than two objectives. For the corrector, we resolved to use the Matlab implementation of the Levenberg-Marquardt method setting the default tolerances for the first order optimality measure (10^{-6}) and for the relative change in the step length (10^{-6}). In all cases, we have taken one initial Pareto optimal solution as starting point (the same for each of the algorithms).

In the following, we present six examples of unconstrained MOPs (four of them with two objectives and the other two with three objectives). For each problem, we show a table that records the number of solutions produced by each algorithm plus the average number of corrector and backtrack iterations and the total amount of function, Jacobian and Hessian evaluations required to obtain the corresponding set of solutions. The field associated to the number of backtrack iterations performed by the method of Hillermeier is left empty since the Levenberg-Marquardt method is based on a trust-region framework. Analogously, the fields corresponding to the number of Hessian evaluations required by the PT-QN and the PT-SD are left empty. In addition, the approximations of the Pareto set (respectively the Pareto front) obtained by the PT-QN are displayed graphically for each example.

3.5.1 Example 1

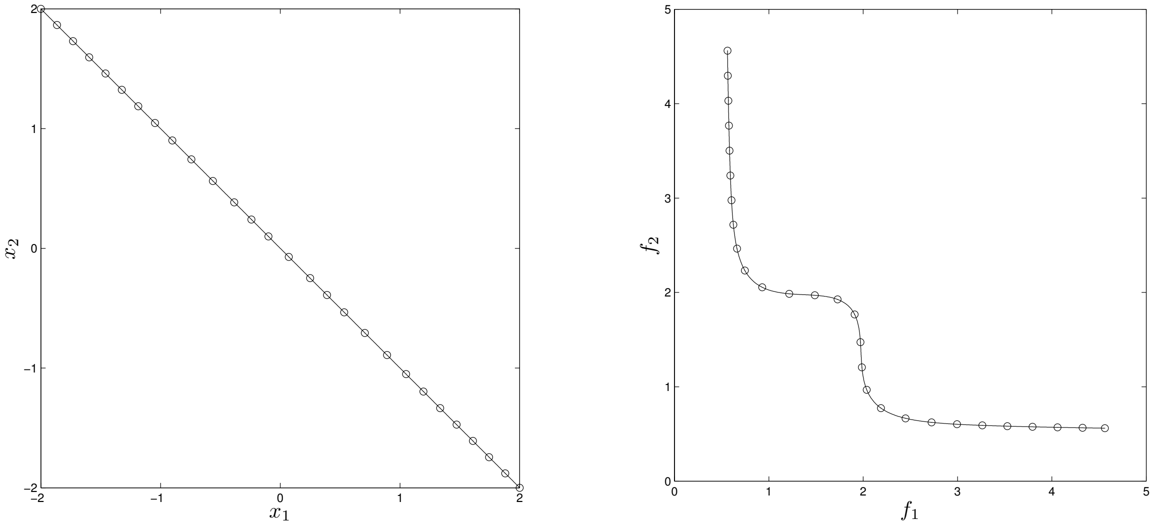
First we consider the BOP from [111] defined as

$$\begin{aligned} f_1(x_1,x_2) &= \frac{1}{2}\Big(\sqrt{1+(x_1+x_2)^2}+\sqrt{1+(x_1-x_2)^2}+x_1-x_2\Big)+\lambda\cdot e^{-(x_1-x_2)^2}, \quad \text{and}\\ f_2(x_1,x_2) &= \frac{1}{2}\Big(\sqrt{1+(x_1+x_2)^2}+\sqrt{1+(x_1-x_2)^2}-x_1+x_2\Big)+\lambda\cdot e^{-(x_1-x_2)^2}. \quad (3.69)\end{aligned}$$

The Pareto set of (3.69) is a line and the Pareto front is convex-concave for $\lambda = 0.85$ as chosen here. Since the Pareto set is unbounded, we have restricted the search to $[-2, 2]$ in both coordinate directions. As desired step length in objective space we have chosen $\tau = 0.27$. Figure 3.7 shows the numerical results of the PT-QN and Table 3.1 the costs (in function evaluations) for the different methods. Apparently, the cost of the Hillermeier method is higher than the cost of the PT-N. The reason for this is that the Pareto set is linear resulting in zero corrector steps for all the PT variants, while the solution set is nonlinear in (x, α)-space. See Figure 3.8 (a) where the α_1-value is plotted against the x-values. Consequently, the Hillermeier approach needs more corrector steps and the solutions—after being projected onto the x-space—may be *not* so well-distributed in the areas where the curvature of the augmented curve is higher (see Figure 3.8). The latter may cause that the method yields a different number of solutions (in this case it delivered 29 solutions while all the PT variants delivered 28 solutions). Another cause for this may be that the Hillermeier method can converge to points with negative Lagrange multipliers if e.g. it goes *beyond* the limits of the solution set. In this case, the solution is discarded while the PT can step back and obtain a new optimal point. Here, both the PT-QN and the PT-SD are the winners in terms of costs, and the costs are equal due to the linearity of the Pareto set.

	Hillermeier	PT-N	PT-QN	PT-SD
Solutions	29.0000	28.0000	28.0000	28.0000
Avg. corrector iterations	3.2333	0.0000	0.0000	0.0000
Avg. backtrack iterations	-	0.0000	0.0000	0.0000
Function evaluations	29.0000	28.0000	28.0000	28.0000
Jacobian evaluations	128.0000	28.0000	28.0000	28.0000
Hessian evaluations	256.0000	56.0000	-	-

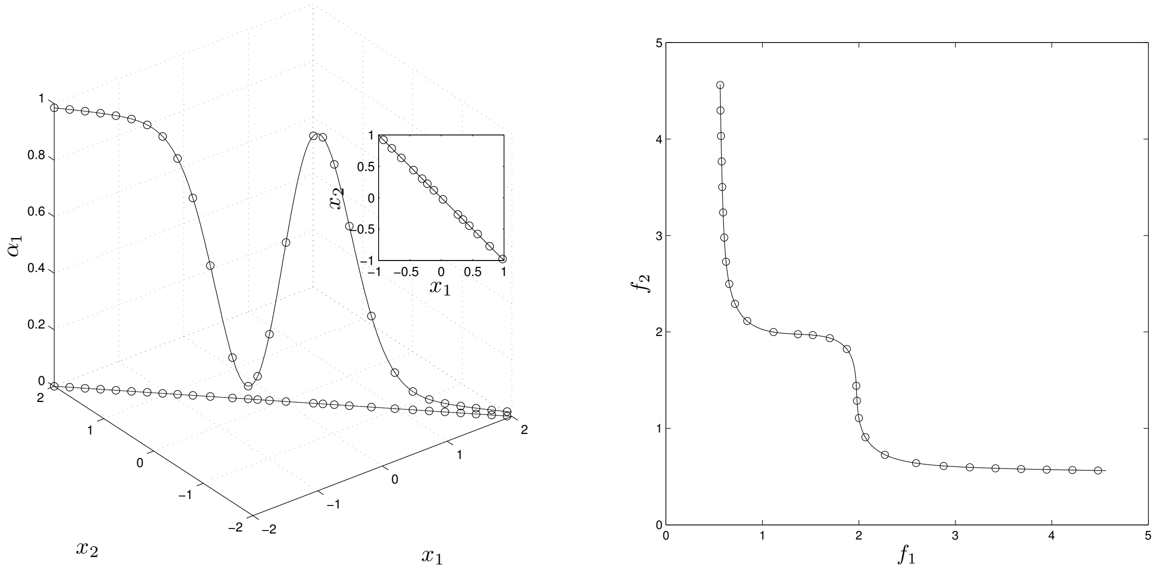
Table 3.1: Computational efforts of the four PC variants on the BOP (3.69).



(a) decision space

(b) objective space

Figure 3.7: Numerical results of the PT-QN on the BOP (3.69).



(a) decision space

(b) objective space

Figure 3.8: Numerical results of the Hillermeier method on the BOP (3.69). The projections of the solutions onto the x-space are also plotted in (a).

3.5.2 Example 2

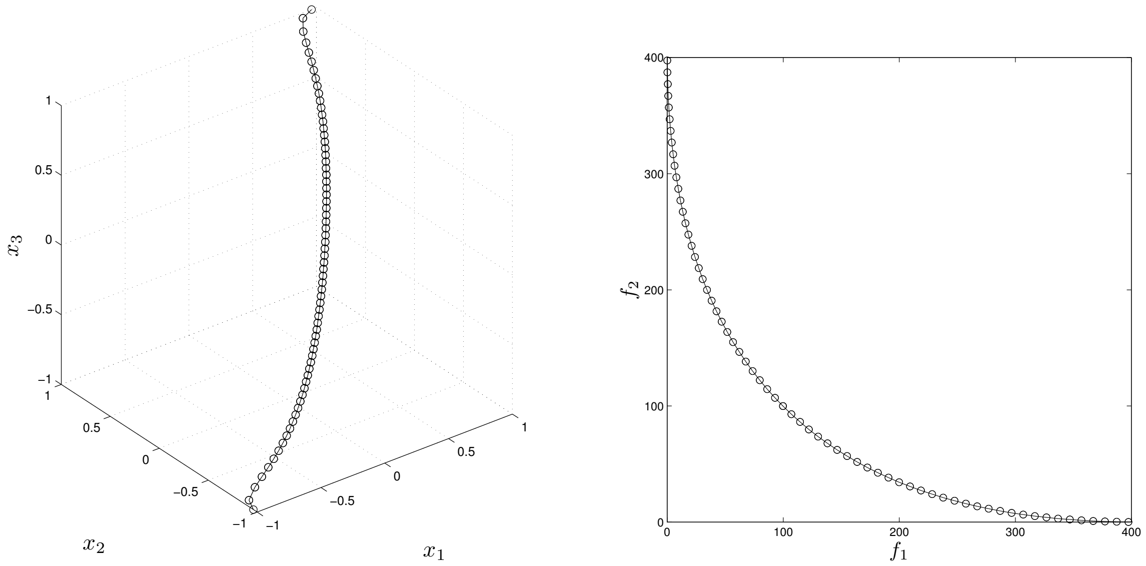
Next we consider the following example from [33] given by

$$f_j(x) = \sum_{\substack{i=1 \\ i \neq j}}^{100} (x_i - a_i^j)^2 + (x_j - a_j^j)^4, \qquad j = 1, 2, \tag{3.70}$$

where $a^1 = (1, \ldots, 1)^T \in \mathbb{R}^{100}$ and $a^2 = -a^1$. The Pareto set of this problem is a nonlinear curve connecting the two minima a^1 and a^2. See Figure 3.9 and Table 3.2 for the results of the PT-QN and the computational efforts, respectively for $\tau = 10$. Although the Pareto set is nonlinear, we see again that the PT-N needs less correctors and outperforms the classical PC method in terms of Jacobian and Hessian evaluations. However, the method of Hillermeier requires less evaluations of the objective functions. This behavior is because the latter is based on the system of equations $\tilde{F}$ that is terms only of the Jacobian and the Hessians of the problem. Thus, only one evaluation of the objectives at each solution found is required at the end of the procedure to deliver the Pareto front. Nevertheless, the costs of the PT-N can be further reduced via using its QN counterpart. Actually, the PT-QN takes a few more function and Jacobian evaluations than the PT-N, but taking in mind that no Hessian information is required, we consider this method the overall winner. On the other hand, since this model is more complicated than the previous one, the PT-SD needs much more corrector steps to bring a predicted point back to the solution curve.

	Hillermeier	PT-N	PT-QN	PT-SD
Solutions	69.0000	69.0000	69.0000	69.0000
Avg. corrector iterations	2.2647	0.9706	1.5441	10.4559
Avg. backtrack iterations	-	0.0000	0.0074	3.6166
Function evaluations	69.0000	135.0000	175.0000	1972.0000
Jacobian evaluations	223.0000	135.0000	174.0000	780.0000
Hessian evaluations	446.0000	270.0000	-	-

Table 3.2: Computational efforts of the four PC variants on the BOP (3.70).



(a) decision space

(b) objective space

Figure 3.9: Numerical results of the PT-QN on the BOP (3.70).

3.5.3 Example 3

We take this time one of the problems of the benchmark proposed in [112]

$$\begin{aligned} f_1(x) &= x_1 + \frac{2}{|J_1|}\sum_{j\in J_1}\left(x_j - x_1^{0.5\left(1+\frac{3(j-2)}{n-2}\right)}\right)^2, \quad \text{and} \\ f_2(x) &= 1-\sqrt{x_1} + \frac{2}{|J_2|}\sum_{j\in J_2}\left(x_j - x_1^{0.5\left(1+\frac{3(j-2)}{n-2}\right)}\right)^2, \end{aligned} \tag{3.71}$$

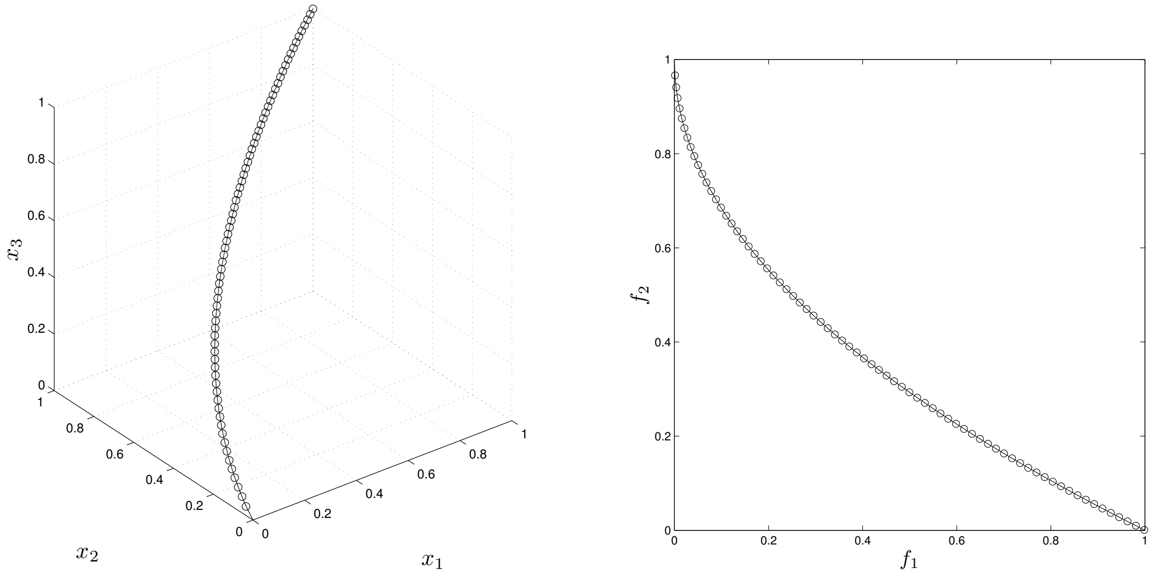
where $J_1 = \{j \mid j \text{ is odd and } 2 \leq j < n\}$ and $J_2 = \{j \mid j \text{ is even and } 2 \leq j < n\}$. The Pareto set is given by

$$x_1 \in [0,1] \quad \text{and} \quad x_j = x_1^{0.5\left(1+\frac{3(j-2)}{n-2}\right)}, \qquad j = 2,\ldots,n. \tag{3.72}$$

As in the first example, the optimal curve is unbounded so the domain search was chosen to be $Q = [0, 1] \times [-1, 1]^{n-1}$ and $n = 100$. The desired step length in objective space was set as $\tau = 0.02$. As can be seen in Table 3.3, the Hillermeier method shows the worst performance (except, of course, in terms of objective function evaluations). The results of the PT-QN are shown in Figure 3.10 and are again comparable to those of the PT-N considering that exact Hessians are not required. However, it is interesting to see that the PT-SD needs only two corrector iterations in average. This may be caused by the lack of second-order information which leads in turn to the computation of more backtrack steps, and eventually, more objective function evaluations. Furthermore, since the Wolfe conditions have been left out of the step length control, the backtracking procedure does not require additional Jacobian evaluations, so this indicator remains low for the PT-SD in this particular case.

	Hillermeier	PT-N	PT-QN	PT-SD
Solutions	74.0000	72.0000	72.0000	73.0000
Avg. corrector iterations	5.2267	0.5753	1.5616	2.0946
Avg. backtrack iterations	-	0.0000	0.1370	3.2532
Function evaluations	75.0000	116.0000	198.0000	535.0000
Jacobian evaluations	468.0000	116.0000	188.0000	230.0000
Hessian evaluations	936.0000	232.0000	-	-

Table 3.3: Computational efforts of the four PC variants on the BOP (3.71).



(a) decision space

(b) objective space

Figure 3.10: Numerical results of the PT-QN on the BOP (3.71).

3.5.4 Example 4

The following problem was taken as well from [112] and reads as

$$
\begin{aligned}
f_1(x) &= x_1 + \frac{2}{|J_1|} \sum_{j \in J_1} \left\{ x_j - 8x_1 \cos\left(6\pi x_1 + \frac{j\pi}{n}\right)\right\}^2, \quad \text{and} \\
f_2(x) &= 1 - \sqrt{x_1} + \frac{2}{|J_2|} \sum_{j \in J_2} \left\{ x_j - 8x_1 \sin\left(6\pi x_1 + \frac{j\pi}{n}\right)\right\}^2,
\end{aligned}
\tag{3.73}
$$

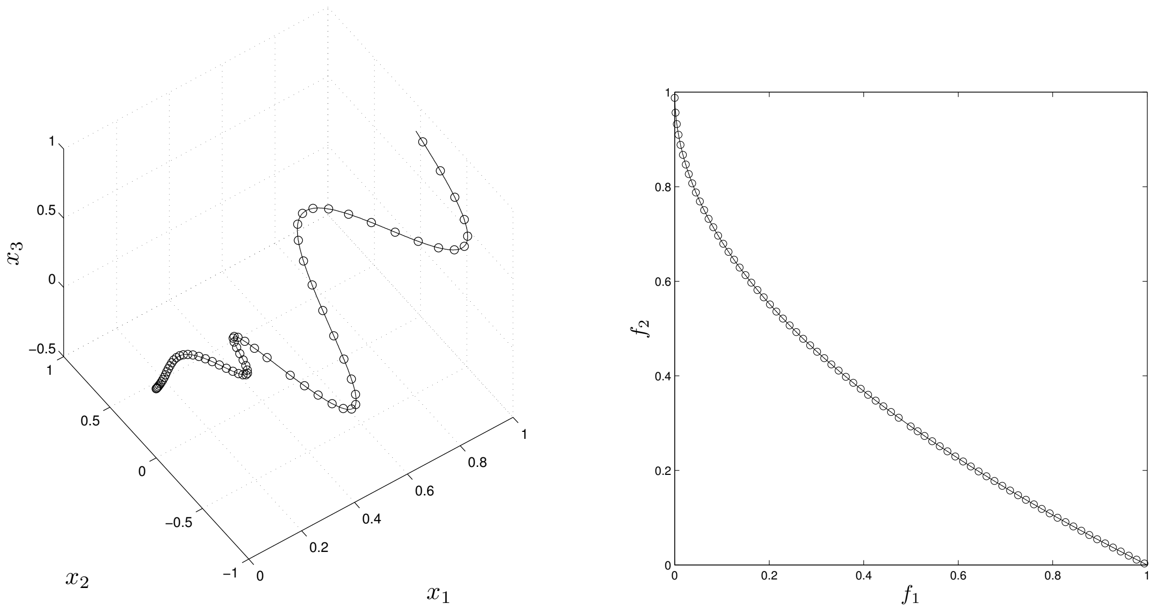
where J_1, J_2, and the domain set Q were chosen as above. We also set $n = 100$ and $\tau = 0.02$. The Pareto set is given by

$$
x_1 \in [0,1] \quad \text{and} \quad x_j = \begin{cases} 8x_1 \cos\left(6\pi x_1 + \frac{j\pi}{n}\right), & j \in J1 \\ 8x_1 \sin\left(6\pi x_1 + \frac{j\pi}{n}\right), & j \in J2 \end{cases}, \qquad j = 2, \dots, n. \tag{3.74}
$$

The problems of the benchmark suggested in [112] were designed to have a Pareto set with a high level of curvature in contrast to those of the suite Zitzler-Deb-Thile (ZDT) [113] and Deb-Thile-Laumanns-Zitzler (DTLZ) [114] which have a flat Pareto set. We select the BOP (3.73) among the most complicated models (see Figure 3.11). Even though, the comparison in Table 3.4 remains consistent with the previous examples: (i) the behavior of the PT-QN remains closer to that of the PT-N, and (ii) the consideration of the compound space (x, α) leads to a potential loss of performance due to the increase of the nonlinearity of the solution set. Observe also in the last column of Table 3.4 that the PT-SD yielded a much greater number of solutions. This is possibly because the Hessian information is completely discarded which makes this method to accumulate sometimes points at the extremes of the solution set. Most likely for the same reason, the phenomenon observed in the previous example arises here again: the PT-SD needs much more backtrack iterations which reduces the number of correctors. Still, this implies a high number of function evaluations but due to the non-consideration of the curvature condition in the step length control, the number of evaluations of the Jacobian remains low.

	Hillermeier	PT-N	PT-QN	PT-SD
Solutions	75.0000	74.0000	73.0000	79.0000
Avg. corrector iterations	6.1974	0.9595	2.4247	3.9241
Avg. backtrack iterations	-	0.0000	0.0000	2.9191
Function evaluations	76.0000	146.0000	251.0000	1156.0000
Jacobian evaluations	564.0000	146.0000	251.0000	390.0000
Hessian evaluations	1128.0000	292.0000	-	-

Table 3.4: Computational efforts of the four PC variants on the BOP (3.73).



(a) decision space (b) objective space

Figure 3.11: Numerical results of the PT-QN on the BOP (3.73).

3.5.5 Example 5

Now we consider the DTLZ2 test problem from [114] with three objectives given by

$$\begin{aligned} f_1(x) &= (1+g(x))\cos(x_1\pi/2)\cos(x_2\pi/2), \\ f_2(x) &= (1+g(x))\cos(x_1\pi/2)\sin(x_2\pi/2), \quad \text{and} \\ f_3(x) &= (1+g(x))\sin(x_1\pi/2), \end{aligned} \tag{3.75}$$

where

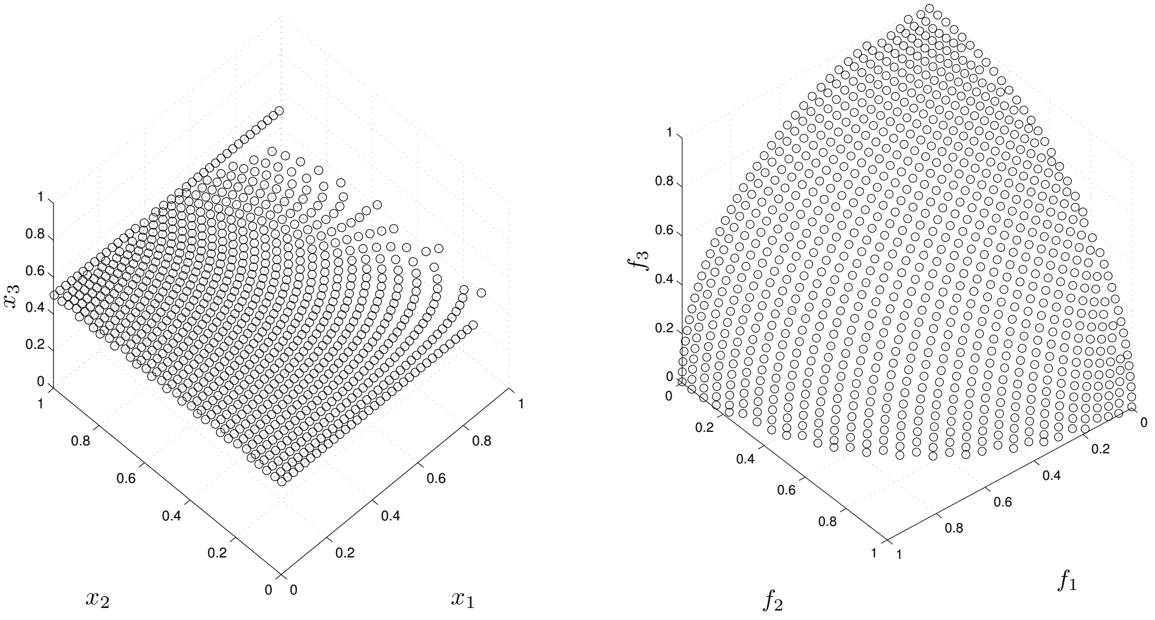
$$g(x) = \sum_{i=3}^{100}(x_i - 0.5)^2. \tag{3.76}$$

The Pareto set here is a flat rectangular surface as shown in Figure 3.12. Taking $\tau = 0.04$, we can see from Table 3.5 that as in the first example—which has a linear Pareto set—only the Hillermeier approach needs some correctors to find optimal points. Additionally, for $k > 2$, the number of function evaluations is not even close to the number of final solutions. This is due to the data structure utilized to track the part of the manifold that is currently covered by the continuation (see Section 3.3). Here, instead of evaluating every solution at the end of the method, each corrector point should be immediately evaluated to verify whether it already belongs to the current solution set. One way to avoid this is by building the partition in parameter space, but this would affect the distribution of points on the Pareto front and the comparison with the PT would not be fair. Observe also that the method of

Hillermeier found less solutions. As explained above, this is the result of discarding solutions with negative Lagrange multipliers.

	Hillermeier	PT-N	PT-QN	PT-SD
Solutions	908.0000	930.0000	930.0000	930.0000
Avg. corrector iterations	3.6380	0.0000	0.0000	0.0000
Avg. backtrack iterations	-	0.0000	0.0000	0.0000
Function evaluations	4545.0000	3721.0000	3721.0000	3721.0000
Jacobian evaluations	4531.0000	930.0000	930.0000	930.0000
Hessian evaluations	13593.0000	2790.0000	-	-

Table 3.5: Computational efforts of the four PC variants on the MOP (3.75).



(a) decision space

(b) objective space

Figure 3.12: Numerical results of the PT-QN on the MOP (3.75).

3.5.6 Example 6

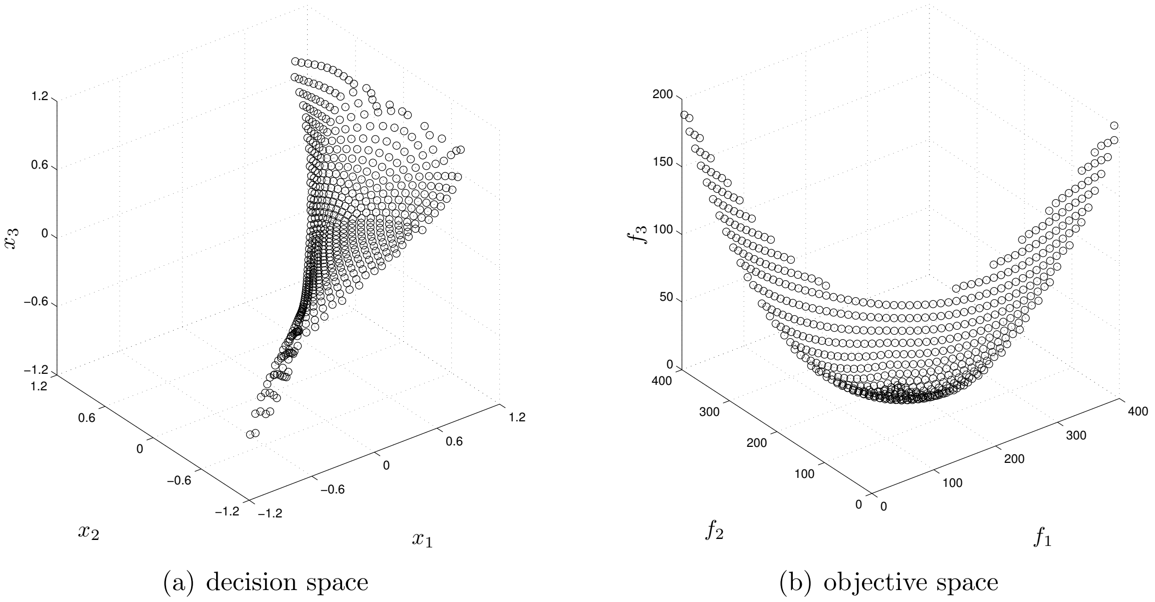
Finally, we reconsider the nonlinear problem given in the second example this time with three objectives, i.e.,

$$f_j(x) = \sum_{\substack{i=1 \\ i \neq j}}^{100} (x_i - a_i^j)^2 + (x_j - a_j^j)^4, \qquad j = 1, 2, 3, \tag{3.77}$$

where $a^1 = (1, \ldots, 1)^T \in \mathbb{R}^{100}$, $a^2 = -a^1$, and $a^3 = (1, -1, 1, -1, \ldots)^T \in \mathbb{R}^{100}$. Figure 3.13 shows the numerical results of the PT-QN for $\tau = 10$ and Table 3.6 the costs of the different methods to obtain this approximation. Due to the variations in the number of solutions produced by the different algorithms, it becomes hard to establish a fair comparison here. As we have previously commented, this variations may be given by many factors. For instance, poor second-order information, the nonlinearity of the model, and the convergence to suboptimal points among other possibilities. However, the PT-N continues to be the one that requires the fewest corrector iterations in average followed by its QN counterpart, while the SD version (as expected) shows the worst results.

	Hillermeier	PT-N	PT-QN	PT-SD
Solutions	728.0000	678.0000	669.0000	675.0000
Avg. corrector iterations	2.7542	1.3349	2.2254	11.5232
Avg. backtrack iterations	-	0000.0000	0.0272	2.9711
Function evaluations	3640.0000	3865.0000	4643.0000	29219.0000
Jacobian evaluations	3361.0000	2016.0000	2749.0000	10796.0000
Hessian evaluations	10083.0000	6048.0000	-	-

Table 3.6: Computational efforts of the four PC variants on the MOP (3.77).



(a) decision space

(b) objective space

Figure 3.13: Numerical results of the PT-QN on the MOP (3.77).

3.6 Discussion

In this chapter, we have presented a novel PC method to trace the manifold of KKT points of a given MOP. The method is in the spirit of classical continuation techniques for the solution of underdetermined systems of equations and has some similarities with the approach proposed by Hillermeier in [28]. The main difference between our method and the latter procedure is that the Pareto Tracer separates decision and weight space whenever possible leading to significant savings in the computational effort—measured in number of function evaluations—which is reported on some benchmark examples. As a by-product, a new set of multipliers $\mu_1, \dots, \mu_k$ is introduced to play the role of providing the degree of freedom necessary to steer the search in parameter space based on what we need in objective space. Additionally, the Newton method recently proposed by Fliege et al. in [16] was selected as corrector instead of applying a Gauss-Newton or Levenberg-Marquardt method to solve the system of equations given by the first-order optimality conditions. However, so far we cannot say much about the computational complexity of the Newton method, which is encouraged for a future work in order to provide a comparison with the available alternatives. Nevertheless, our aim here has been principally to investigate other possibilities for correctors that are not so hardly dependent on the set of Lagrange multipliers. Besides, the zero set of the KKT equations contains solutions that are not even local optima. The Pareto Tracer method is applicable to MOPs with in principle any number k of objectives and can be made Hessian free via using elements from QN methods. Here, we have proposed to ensure the positive definiteness of the BFGS updates through modified Cholesky factorizations—used as well by the Newton method—since the Wolfe conditions may be hard to satisfy for each objective at a time. Furthermore, a second role for the BFGS updates is suggested in the context of continuation: an update of the Hessians at predicted points with base on the information gathered from the previous corrector may help to preserve the second-order information gained along all the trajectory. For comparison purposes, we also provide a SD realization of the method and suggest to include in the comparison the approach in [90] that can approximate the tangent space in $\mathcal{O}(n^2)$ without the use of Hessians. Several optimizations for the bi-objective case were implemented as well. The most significant one—apart of excluding data structures—is the use of secants in the context of the PT-SD to compute predictor directions. As graphically illustrated by three examples in Figure 3.4, the derivatives can be nearly orthogonal to the Pareto set, which additionally encourages the use of QN realizations. Moreover, based on the numerical results, the PT-QN seems to be the overall winner in terms of function evaluations, so our next task would be clearly to intensify the investigation of this particular method in order to provide more efficient computations.

Chapter 4

Handling Equality Constrains

In this chapter, we will extend the Pareto Tracer method to handle equality constraints, i.e., we will consider MOPs of the form

$$\begin{aligned} &\min_{x \in \mathbb{R}^n} F(x) \\ &\text{s.t. } h_j(x) = 0, \qquad j = 1, \dots, p. \end{aligned} \tag{4.1}$$

Again, we utilize classical continuation techniques on a map $\tilde{F}$ that is motivated by the KKT conditions (now for equality constrained MOPs). The extended method can also separate decision from weight space leading to a reduction of the overall computational cost. The vector μ resulting from this separation will have—consistently with the unconstrained case—the role of steering the search into a direction given in objective space. For the corrector, we suggest a modification of the Newton method [16] to handle problems with equalities. The analysis of the convergence rate of the modifications is not considered in this paper but is highly recommended for a future work. We will, instead, give some theoretical and numerical evidence that the flow defined by the adapted Newton direction leads from an initial solution to a KKT point. In addition, an Armijo condition adequate to the new context had to be designed in order to respect the predicted improvements of the considered functions (objectives and equalities). Take in mind that it may be cases where there is no descent direction that additionally points toward the feasible region. Those situations claim for a careful review of the concept we have so far of *suitable* step lengths. Furthermore, two Hessian-free realizations of the Pareto Tracer were developed based respectively on quasi-Newton and gradient descent methods. Still, a detailed study of the asymptotic time complexity of the proposals is missing which is mandatory to provide a more fair comparison with other methods and possibly to optimize the computations.

The remainder of this chapter is organized as follows. Sections 4.1 and 4.2 are dedicated to the computation of predictors and correctors, respectively. The extended Pareto Tracer for equality constrained MOPs is presented in Section 4.3 and Hessian-free realizations are discussed later on in Section 4.4. Finally, some numerical results are given in Section 4.5 and the conclusions in Section 4.6.

4.1 Predictor

As usual during predictor steps, we start by computing the tangent space to the solution manifold at a given KKT point of (4.1), namely $x \in \mathbb{R}^n$. Here, we can proceed analog to the unconstrained case as we will show immediately. The auxiliary map $\tilde{F} : \mathbb{R}^{n+k+p} \to \mathbb{R}^{n+p+1}$ inspired by the first-order conditions of optimality (as in Equation (3.3)) reads as

$$\tilde{F}(x, \alpha, \lambda) = \begin{pmatrix} \sum_{j=1}^{k} \alpha_j \nabla f_j(x) + \sum_{j=1}^{p} \lambda_j \nabla h_j(x) \\ h(x) \\ \sum_{j=1}^{k} \alpha_j - 1 \end{pmatrix} = 0, \tag{4.2}$$

where $\alpha \in \mathbb{R}^k$ and $\lambda \in \mathbb{R}^p$ are the associated Lagrange multipliers such that $\alpha_j \geq 0$ for all $j = 1, \ldots, k$. See Theorem 2.1.2 for more details. Defining

$$W_{\alpha,\lambda} = \sum_{j=1}^{k} \alpha_j \nabla^2 f_j(x) + \sum_{j=1}^{p} \lambda_j \nabla^2 h_j(x) \in \mathbb{R}^{n \times n}, \quad \text{and} \tag{4.3}$$

$$H = \begin{pmatrix} \nabla h_1(x)^T \\ \vdots \\ \nabla h_p(x)^T \end{pmatrix} \in \mathbb{R}^{p \times n}, \tag{4.4}$$

we obtain

$$\tilde{F}'(x, \alpha, \lambda) = \begin{pmatrix} W_{\alpha,\lambda} & J^T & H^T \\ H & 0 & 0 \\ 0 & 1 \ldots 1 & 0 \end{pmatrix} \in \mathbb{R}^{(n+p+1) \times (n+k+p)} \tag{4.5}$$

whose kernel vectors are sought. For this, let $\nu \in \mathbb{R}^n$, $\mu \in \mathbb{R}^k$, $\xi \in \mathbb{R}^p$ such that

$$\begin{pmatrix} W_{\alpha,\lambda} & J^T & H^T \\ H & 0 & 0 \\ 0 & 1 \ldots 1 & 0 \end{pmatrix} \begin{pmatrix} \nu \\ \mu \\ \xi \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}. \tag{4.6}$$

Again, we choose a μ that satisfies (3.5), i.e., $\sum_{j=1}^{k} \mu_j = 0$, which reduces (4.6) to

$$\begin{pmatrix} W_{\alpha,\lambda} & H^T \\ H & 0 \end{pmatrix} \begin{pmatrix} \nu_\mu \\ \xi \end{pmatrix} = - \begin{pmatrix} J^T \\ 0 \end{pmatrix} \mu. \tag{4.7}$$

If $\mathrm{rank}(W_{\alpha,\lambda}) = n$ and $\mathrm{rank}(H) = p$, it follows that the solution of (4.7) is unique since then the matrix on the left hand side is regular. To show this, let's suppose that

$$\begin{pmatrix} W_{\alpha,\lambda} & H^T \\ H & 0 \end{pmatrix} \begin{pmatrix} \nu \\ \xi \end{pmatrix} = 0, \tag{4.8}$$

or equivalently

$$W_{\alpha,\lambda}\nu + H^T\xi = 0, \quad \text{and} \tag{4.9a}$$

$$H\nu = 0. \tag{4.9b}$$

From (4.9a), we obtain $\nu = -W_{\alpha,\lambda}^{-1}H^T\xi$, and substituting the value of ν in (4.9b) it is $-HW_{\alpha,\lambda}^{-1}H^T\xi = 0$ which implies that $\xi = 0$. Then, substituting ξ in (4.9a) leads to $W_{\alpha,\lambda}\nu = 0$ which implies in turn that $\nu = 0$. Thus, the linear system of equations (4.8) has only the trivial solution $\nu = 0$ and $\xi = 0$ implying that its associated matrix is regular. One difference here with the handling of unconstrained MOPs is that we have to compute the compound vector $(\nu_\mu, \xi) \in \mathbb{R}^{n+p}$ in order to obtain ν_μ. This means that we shall solve an equation system of size $\mathcal{O}(n+p)^2$ and later discard the component ξ of the solution. We can otherwise find a way to compute ξ in advance (as we did with μ) but this would be far more complicated. Moreover, since ξ does not have neither an associated *meaning* nor any degree of freedom, the invested time and efforts would possibly be in vain. Our next task is thus to prove that ν_μ is tangent to the Pareto set of the equality constrained MOP (4.1). Let

$$g_{\alpha,\lambda}(x) = \sum_{j=1}^{k} \alpha_j f_j(x) + \sum_{j=1}^{p} \lambda_j h_j(x). \tag{4.10}$$

Then, a curve along the set of KKT points through x_0 with associated weight (α_0, λ_0) can be described via

$$H(x,t) = \begin{pmatrix} \nabla g_{\alpha(t),\lambda(t)}(x) \\ h(x) \end{pmatrix} = 0, \qquad t \in \mathbb{R}_+. \tag{4.11}$$

Since μ represents the change in α-space we can set $\alpha(t) = \alpha_0 + t\mu$. Analogously, $\lambda(t) = \lambda_0 + t\xi$ and it is thus $H(x_0, t_0 = 0) = 0$. Consequently, for the tangent vector of $H^{-1}(0)$ at (x_0, t_0) it holds

$$\frac{\partial H}{\partial x} t_H(x_0, t_0) + \frac{\partial H}{\partial t} = 0$$

$$\begin{pmatrix} W_{\alpha,\lambda} \\ H \end{pmatrix} t_H(x_0, t_0) + H^T\xi + J^T\mu = 0$$

which is satisfied for $t_H(x_0, t_0) = \nu_\mu$. Note that we can also express (4.7) as

$$W_{\alpha,\lambda}\nu_\mu + H^T\xi = -J^T\mu, \quad \text{and} \tag{4.12a}$$

$$H\nu_\mu = 0, \tag{4.12b}$$

and by the sum of (4.12a) and (4.12b) it follows that

$$W_{\alpha,\lambda}\nu_\mu + H\nu_\mu + H^T\xi + J^T\mu = 0. \tag{4.13}$$

In our implementations, we have made one modification w.r.t. (4.7): instead of $W_{\alpha,\lambda}$ we have used W_α which means that we only consider first order approximations of the equality constraints. That is, we compute the predictor direction ν_μ such that

$$\begin{pmatrix} \nu_\mu \\ \xi \end{pmatrix} = -\begin{pmatrix} W_\alpha & H^T \\ H & 0 \end{pmatrix}^{-1} \begin{pmatrix} J^T \\ 0 \end{pmatrix} \mu. \tag{4.14}$$

We have observed that the change from (4.7) to (4.14) leads to nearly the same performance while the requirements of (4.14) are much lower since the Hessians of the equalities are omitted. A further study on the impact of eliminating the second-order terms of the equality constraints should be considered in the immediate future.

Now we are ready to orientate our tangent in parameter space based on a desired movement $d \in \mathbb{R}^k$ in objective space. Again, the set of multipliers $\mu_1, \ldots, \mu_k$ will play the predominant role in this task. The idea is to compute first a $\mu_d \in \mathbb{R}^k$ satisfying the linear system of equations

$$J\nu_{\mu_d} = d. \tag{4.15}$$

However, since the solution of (4.7) is a compound vector, we transform (4.15) into

$$(J \;\; 0) \begin{pmatrix} \nu_{\mu_d} \\ \xi \end{pmatrix} = d, \tag{4.16}$$

and substituting (ν_{μ_d}, ξ) in Equation (4.16) by its corresponding expression given by (4.7), we obtain the linear system of size $\mathcal{O}(k^2)$

$$-(J \;\; 0) \begin{pmatrix} W_{\alpha,\lambda} & H^T \\ H & 0 \end{pmatrix}^{-1} \begin{pmatrix} J^T \\ 0 \end{pmatrix} \mu_d = d. \tag{4.17}$$

For simplicity, we will take

$$\begin{aligned} A &= -(J \;\; 0) \begin{pmatrix} W_{\alpha,\lambda} & H^T \\ H & 0 \end{pmatrix}^{-1} \begin{pmatrix} J^T \\ 0 \end{pmatrix} \\ &= -JWJ^T \in \mathbb{R}^{k \times k}, \end{aligned} \tag{4.18}$$

where $W \in \mathbb{R}^{n \times n}$ is the left top submatrix of

$$\begin{pmatrix} W_{\alpha,\lambda} & H^T \\ H & 0 \end{pmatrix}^{-1}.$$

Since J is rank deficient at optimal points, $\text{rank}(A) < k$ and the solution of (4.17) is not unique or does not exist. On the other hand, note that from (4.7), the predictor directions can be expressed as

$$\nu_\mu = -WJ^T\mu, \tag{4.19}$$

from which it can be derived (using the fact that ν_μ is a tangent vector for any μ with components that sum up to 0) that the columns of $-WJ^T$ span the tangent space of the Pareto set (at least under our rank assumptions). Additionally, since x is a KKT point, by Theorem 2.1.2 it follows that

$$J^T\alpha + H^T\lambda = 0, \tag{4.20}$$

where $\alpha \in \mathbb{R}^k$ and $\lambda \in \mathbb{R}^p$ are the corresponding KKT multipliers. Then, by taking the inner product of ν_μ and (4.20), we obtain from (4.12b)

$$\nu_\mu^T J^T\alpha + \nu_\mu^T H^T\lambda = \nu_\mu^T J^T\alpha = 0. \tag{4.21}$$

Since the above equation is satisfied for an arbitrary ν_μ in the range of $-WJ^T$, it holds that $A\alpha = 0$. Then, under the assumption that $\text{rank}(J) = k-1$, we can proceed analogously to Chapter 3. That is, the solution of (4.17) is given by

$$\mathbb{M} = \mu_d^+ + \mathbb{R}\alpha, \tag{4.22}$$

where

$$\mu_d^+ = A^+d, \tag{4.23}$$

and the solution $\mu_d \in \mathbb{M}$ with components that sum up to zero is given by

$$\mu_d = \mu_d^+ - (e^T\mu_d^+)\alpha. \tag{4.24}$$

Next, following the instructions above, we can compute a set of vectors $\nu_1, \ldots, \nu_{k-1}$ such that the corresponding movements $J\nu_i = d_i$, $i = 1, \ldots, k-1$, in objective space form an orthonormal basis of the linearized Pareto front at $F(x)$, i.e., the set of directions $d_1, \ldots, d_{k-1}$ are chosen as in (3.24). Alternatively, one can obtain orthogonal vectors that span the linearized Pareto *set* via computing an orthonormal basis of the image of

$$M = -WJ^T \in \mathbb{R}^{n\times k}.$$

Finally, the step size as well as the special treatment for the bi-objective case can be taken as proposed in the previous chapter.

4.2 Corrector

In order to correct the predicted point back to the manifold of optimal solutions, we have decided to adapt the Newton method presented in [16] to the current context. Another option, as stated before, could be to solve the KKT system of equations. Based on the arguments given in Chapter 3, we prefer to study new alternatives and present numerical results that may inspire further investigation. So far, we have not considered the convergence rates of the modifications (which is certainly an interesting topic), but can give evidence that the flow defined by the Newton direction leads from any initial solution to a KKT point. That is, the resulting Newton method will also converge toward a KKT point if the step size is chosen adequately. In particular, we suggest to compute the Newton direction for (4.1) via solving

$$\begin{aligned} \min_{(\nu,\delta)\in\mathbb{R}^n\times\mathbb{R}} \quad & \delta \\ \text{s.t.} \quad & \nabla f_j(x)^T\nu + \tfrac{1}{2}\nu^T\nabla^2 f_j(x)\nu \leq \delta, \qquad j = 1,\dots,k, \\ & h_j(x) + \nabla h_j(x)^T\nu = 0, \qquad j = 1,\dots,p. \end{aligned} \tag{4.25}$$

The difference of (4.25) with the unconstrained version (3.34) is the additional constraint $h(x) + H\nu = 0$. Still, δ represents a measure of the expected decrease of the objectives by performing a line search in direction ν. To be more precise, given ν, we can compute δ as

$$\delta = \max_{j=1,\dots,k} \nabla f_j(x)^T\nu + \frac{1}{2}\nu^T\nabla^2 f_j(x)\nu, \tag{4.26}$$

from which it is inferred that the progress toward the optimal set is determined by the objective function that decreases the least. Additionally, we need a similar *merit* function to estimate how close we expect to be from the feasible region defined by $\mathcal{X} = \{x \in \mathbb{R}^n \mid h(x) = 0\}$. For this, we utilize

$$P_h(x) = \frac{1}{2}\sum_{j=1}^{p} h_j(x)^2 = \frac{1}{2}\|h(x)\|^2, \tag{4.27}$$

whose expected decrease in direction ν can be measured by the directional derivative

$$\nabla_\nu P_h(x) = h(x)^T H\nu = -\|h(x)\|^2. \tag{4.28}$$

By premultiplying $h(x)^T$ at both sides of the equality constrain in (4.25) we obtain the second equality above. The following result shows that this adaption can be seen as a particular penalization method for which the penalized MOP is given by

$$\min_{x\in\mathbb{R}^n} F_h : \mathbb{R}^n \to \mathbb{R}^k, \tag{4.29}$$

where F_h is the vector of penalized functions $f_1^h,\dots,f_k^h : \mathbb{R}^n \to \mathbb{R}$ defined as

$$f_j^h(x) = f_j(x) + CP(x), \qquad j = 1,\dots,k, \tag{4.30}$$

for some $C > 0$.

Proposition 4.2.1
Let $x \in \mathbb{R}^n$ be given and let the objective functions be strictly convex. Further, let (ν^*, δ^*) be the solution of (4.25).

(*a*) *If $\nu^* = 0$, then $\delta^* = 0$ and x is a KKT point of* (4.1).

(*b*) *If $\nu^* \neq 0$ and $\delta^* < 0$, then ν^* is a descent direction of* (4.29) *for $C = 0$ (i.e., a descent direction of the unconstrained MOP* (3.1)).

(*c*) *If $\nu^* \neq 0$ and $\delta^* \geq 0$, then $\|h(x)\|^2 \neq 0$ and ν^* is a descent direction of* (4.29) *for*

$$C > \frac{\delta^*}{\|h(x)\|^2} > 0. \tag{4.31}$$

Proof The Lagrangian of (4.25) is given by

$$\begin{aligned}\mathcal{L}((\nu,\delta),\alpha,\lambda) = \delta &+ \sum_{j=1}^{k} \alpha_j \left(\nabla f_j(x)^T \nu + \frac{1}{2}\nu^T \nabla^2 f_j(x)\nu - \delta \right) \\ &+ \sum_{j=1}^{p} \lambda_j \left(h_j(x) + \nabla h_j(x)^T \nu \right).\end{aligned} \tag{4.32}$$

Then, the KKT conditions for (4.25) at (ν^*, δ^*) read as

$$\nabla_\nu \mathcal{L} = \sum_{j=1}^{k} \alpha_j \left(\nabla f_j(x) + \nabla^2 f_j(x)\nu^* \right) + \sum_{j=1}^{p} \lambda_j \nabla h_j(x) = 0, \tag{4.33a}$$

$$\nabla_\delta \mathcal{L} = 1 - \sum_{j=1}^{k} \alpha_j = 0, \tag{4.33b}$$

$$\nabla f_j(x)^T \nu^* + \frac{1}{2}\nu^{*T} \nabla^2 f_j(x)\nu^* \leq \delta^*, \qquad j = 1, \dots, k, \tag{4.33c}$$

$$h_j(x) + \nabla h_j(x)^T \nu^* = 0, \qquad j = 1, \dots, p, \tag{4.33d}$$

$$\alpha_j \geq 0, \qquad j = 1, \dots, k, \tag{4.33e}$$

$$\alpha_j \left(\nabla f_j(x)^T \nu^* + \frac{1}{2}\nu^{*T} \nabla^2 f_j(x)\nu^* - \delta^* \right) = 0, \qquad j = 1, \dots, k. \tag{4.33f}$$

(*a*) Let $\nu^* = 0$. From (4.33b) and (4.33e) it follows that α is a convex weight. Further, from (4.33a) it follows that if $\nu^* = 0$ then

$$\sum_{j=1}^{k} \alpha_j \nabla f_j(x) + \sum_{j=1}^{p} \lambda_j \nabla h_j(x) = 0,$$

and from (4.33d) that $h(x) = 0$. Thus, x is a KKT point of (4.1). Finally, from (4.33f) we obtain $\delta^* = 0$.

(*b*) Let $\nu^* \neq 0$ and $\delta^* < 0$. Then by (4.33c) and since all Hessians are positive definite we have

$$\nabla f_j(x)^T \nu^* \leq \delta^* - \frac{1}{2}\nu^{*T}\nabla^2 f_j(x)\nu^* < 0, \qquad j = 1, \ldots, k. \tag{4.34}$$

Since h is not used here, the statement holds also for unconstrained problems which has already been proven in [16].

(*c*) Let $\nu^* \neq 0$ and $\delta^* \geq 0$. From (4.28) we have for each penalized function

$$\nabla f_j^h(x)^T \nu^* = \left(\nabla f_j(x) + C\sum_{j=1}^{p} h_j(x)\nabla h_j(x)\right)^T \nu^* = \nabla f_j(x)^T \nu^* - C\|h(x)\|^2.$$

Thus, it is $\nabla f_j^h(x)^T \nu^* < 0$, $\forall j = 1, \ldots, k$, for

$$C > \frac{\max\limits_{j=1,\ldots,k} \nabla f_j(x)^T \nu^*}{\|h(x)\|^2}.$$

Since by (4.26) $\delta^* > \max\limits_{j=1,\ldots,k} \nabla f_j(x)^T \nu^*$, we can take

$$C > \frac{\delta^*}{\|h(x)\|^2} > 0$$

to ensure that C is strictly positive. It remains to show that $\|h(x)\|^2 \neq 0$: from (4.33f) we conclude that either $\alpha_j = 0$ or

$$\nabla f_j(x)^T \nu^* \geq -\frac{1}{2}\nu^{*T}\nabla^2 f_j(x)\nu^*, \qquad j = 1, \ldots, k. \tag{4.35}$$

Then, by taking the inner product of ν^* and (4.33a) we obtain

$$\sum_{j=1}^{k} \alpha_j \left(\nabla f_j(x)^T \nu^* + \nu^{*T}\nabla^2 f_j(x)\nu^*\right) + \sum_{j=1}^{p} \lambda_j \nabla h_j(x)^T \nu^* = 0. \tag{4.36}$$

By using (4.35), the above formula reduces to the following inequality

$$\frac{1}{2}\nu^{*T} W_\alpha \nu^* + \sum_{j=1}^{p} \lambda_j \nabla h_j(x)^T \nu^* \leq 0. \tag{4.37}$$

If additionally $h(x) = 0$, in virtue of (4.33d) we end up with

$$\frac{1}{2}\nu^{*T} W_\alpha \nu^* \leq 0 \tag{4.38}$$

which is satisfied only if $\nu^* = 0$ since α is a convex weight and the Hessians are positive definite. This contradicts the assumption, and the claim follows. □

Since we are thus computing descent directions of (4.29) there are three choices: (i) we can improve F but not P_h, (ii) we can improve both, and (iii) we can improve P_h but not F. This is reflected by the step size control we propose in the following. The idea is to apply the same componentwise Armijo condition as above but on the following function

$$\tilde{F}_h(x) = \begin{cases} F(x), & \delta < 0 \text{ and } \|h(x)\| = 0, \\ (F(x), P_h(x))^T, & \delta < 0 \text{ and } \|h(x)\| \neq 0, \\ P_h(x), & \delta \geq 0. \end{cases} \tag{4.39}$$

Recall that we already proved (in Proposition 4.2.1 (c)) that if $\delta \geq 0$, then $\|h(x)\| \neq 0$ and the term $P_h(x)$ will decrease (see Equation (4.28)). If $\delta < 0$, either item (i) or (ii) holds and by Proposition 4.2.1 (b), ν^* is a descent direction of F. If $\|h(x)\|^2 = 0$, P_h cannot be improved any more and we neglect the penalization term. If $\|h(x)\|^2 \neq 0$, P_h and F can be decreased simultaneously by a line search in direction ν^*. Therefore, the step size control should monitor that a sufficient decrease is achieved for each objective including P_h. Here, we avoid the selection of a penalization factor by applying a componentwise Armijo condition on the augmented function resulting of considering P_h as the $(k+1)$-th objective. Finally, in case (iii), since $\delta \geq 0$, at least one of the objectives will worsen in direction ν^*. However, by Proposition 4.2.1 (c), $\|h(x)\|^2 \neq 0$ and ν^* is a descent direction of (4.29) for a value of $C > 0$ given in (4.31). Then, the choice of a step length that produces a sufficient decrease in P_h—excluding the objective functions—can be seen as the choice of $C \gg 0$. Putting all together, we take as an acceptable step length $t > 0$ one that satisfies

$$\tilde{F}_h(x + t\nu) \leq \tilde{F}_h(x) + ct\Delta\tilde{F}_h(x). \tag{4.40}$$

Here, $\Delta\tilde{F}_h(x)$ represents the expected decrease of $\tilde{F}_h$ in objective space given by

$$\Delta\tilde{F}_h(x) = \begin{cases} \delta e, & \delta < 0 \text{ and } \|h(x)\|^2 = 0, \\ (\delta e, -\|h(x)\|^2)^T, & \delta < 0 \text{ and } \|h(x)\|^2 \neq 0, \\ -\|h(x)\|^2, & \delta \geq 0, \end{cases} \tag{4.41}$$

where $e \in \mathbb{R}^k$ is the vector of all ones. In our implementations, we have chosen $c = 0.1$ as proposed in [16]. Last, Proposition 4.2.1 (a) suggests as a possible stopping criteria

$$\|\nu^*\| < \varepsilon, \tag{4.42}$$

for a small value $\varepsilon > 0$ (taken as $\varepsilon = 5\sqrt{\text{eps}}$).

We will next discuss some of the implementation details of the proposed method. First, we must deal with the Newton direction subproblem (NDS) that—as for the unconstrained case—is in the category of QCLP. Again, we will shift it into the form

$$\begin{aligned} \min_{x \in \mathbb{R}^n} \; & f^T x \\ \text{s.t.} \; & \|A_j x + b_j\| \leq c_j^T x + d_j, \qquad j = 1, \ldots, k, \\ & Ex + e = 0, \end{aligned} \tag{4.43}$$

where $f, c_j \in \mathbb{R}^{n+1}$, $A_j \in \mathbb{R}^{(n_j-1)\times(n+1)}$, $b_j \in \mathbb{R}^{n_j-1}$, $d_j \in \mathbb{R}$, $E \in \mathbb{R}^{p\times(n+1)}$ and $e \in \mathbb{R}^p$. The modified NDS is thus given by

$$\begin{aligned}
&\min_{(\nu,\delta)\in\mathbb{R}^n\times\mathbb{R}} \Big(0,\dots,0,1\Big)\begin{pmatrix}\nu\\ \delta\end{pmatrix} = \delta \\
&\text{s.t.} \quad \left\| \begin{pmatrix} \frac{1}{2}\nabla f_j^T & -\frac{1}{2} \\ \frac{1}{\sqrt{2}} L_j^T & 0 \end{pmatrix}\begin{pmatrix}\nu\\ \delta\end{pmatrix} + \begin{pmatrix}\frac{1}{2}\\ 0\end{pmatrix} \right\| \leq \Big(-\tfrac{1}{2}\nabla f_j^T, \tfrac{1}{2}\Big)\begin{pmatrix}\nu\\ \delta\end{pmatrix} + \frac{1}{2}, \quad j = 1,\dots,k, \\
&\qquad\qquad h(x) + (H\ 0)\begin{pmatrix}\nu\\ \delta\end{pmatrix} = 0,
\end{aligned} \tag{4.44}$$

where $L_j L_j^T = \nabla^2 f_j(x)$ and $\nabla f_j = \nabla f_j(x)$. Note that the only difference between (4.44) and (3.38) is the additional equality constraint. Then, the subproblem (4.44) belongs to SOCP with $f = (0,\dots,0,1)^T \in \mathbb{R}^{n+1}$,

$$A_j = \begin{pmatrix} \frac{1}{2}\nabla f_j^T & -\frac{1}{2} \\ \frac{1}{\sqrt{2}} L_j^T & 0 \end{pmatrix} \in \mathbb{R}^{(n+1)\times(n+1)},$$

$b_j = (^1\!/_2, 0, \dots, 0)^T \in \mathbb{R}^{n+1}$, $c_j = (-^1\!/_2\nabla f_j^T, ^1\!/_2)^T \in \mathbb{R}^{n+1}$, and $d_j = ^1\!/_2 \in \mathbb{R}$. Additionally, it is $E = (H\ 0) \in \mathbb{R}^{p\times(n+1)}$ and $e = h(x) \in \mathbb{R}^p$. No further discussion on the implementation of the step length control and the stopping criteria will be necessary. As well, the pseudo codes of the Newton method for equality constrained MOPs and the proposed linear search backtracking are quite similar to Algorithm 1 and Algorithm 2, respectively, so will be omitted. For the Newton method, just recall that the dual problem of (4.43) was given in Chapter 2 by (2.33). For the step length control, it will only be required to change the condition of the if-statement in Line 3 by the modified Armijo condition (4.40).

4.3 The Pareto Tracer Method

In this section, we are ready to describe the Pareto Tracer method for equality constrained MOPs. The resulting pseudo code is basically the procedure presented in Algorithm 3 to handle problems without restrictions. Thus, in the following, we will merely discuss some minor details that are relevant to the new context.

The Pareto Tracer is now capable of tracing the manifold of Pareto points of MOPs with equality constraints. Again, the method separates decision from weight space obtaining as a result a vector μ that can be used to steer the search given a direction d in objective space. In addition, we chose to utilize a modified version of the Newton method as corrector instead of solving the KKT system of equations. Several reasons for this election were given in the previous chapter and section which are supported by the promising numerical results presented later in this chapter. We can thus base the new implementations on Algorithm 3 by changing the lines 8 and 9 to use the formulas (4.24) and (4.14) for the computation of orientation and direction vectors, respectively. Further, the lines 1 and 13 should make a call to the appropriate corrector method.

In order to keep track of the covered section of the solution set, we can take the data structure proposed in [34] with the modifications explained in Chapter 3. Basically, the 'recovering' technique presented in Algorithm 4 remains untouched.

Finally, for equality constrained BOPs, the computations can also be significantly simplified: still, the Pareto fronts of such problems are typically one-dimensional (i.e., the inclusion of equality restrictions does not change the dimension of the solution manifold). We can find more details on this in [28]. Then, we will keep Algorithm 5 identical except for Line 6 that should include Equation (4.14) to compute the tangent vector, and Line 14 that should use the modified Newton method. The discussion regarding the stopping criteria for the continuation can also be incorporated without further changes.

4.4 Hessian-free Realizations

To cope with Hessian-free realizations of the PT that can additionally handle problems subject to equality constraints, we basically have two options: one exclusively based on gradients (the SD realization) and another one based on QN methods.

If we choose to completely disregard the second-order information (first option), we have to approximate the Hessians by the identity matrix ($I \in \mathbb{R}^{n\times n}$). Then, from (4.14) we obtain for the predictor

$$\begin{pmatrix} \nu_\mu \\ \xi \end{pmatrix} = -\begin{pmatrix} I & H^T \\ H & 0 \end{pmatrix}^{-1} \begin{pmatrix} J^T \\ 0 \end{pmatrix} \mu. \tag{4.45}$$

The orientation vector can in turn be computed as in Equation (4.24) where

$$\mu_d^+ = A^+ d, \tag{4.46}$$

and A is this time

$$A = -\left(J \;\; 0\right) \begin{pmatrix} I & H^T \\ H & 0 \end{pmatrix}^{-1} \begin{pmatrix} J^T \\ 0 \end{pmatrix}. \tag{4.47}$$

Next, for the corrector, the Newton direction subproblem (4.25) gets reduced to

$$\begin{aligned} \min_{(\nu,\delta)\in\mathbb{R}^n\times\mathbb{R}} \quad & \tfrac{1}{2}\|\nu\|^2 + \delta \\ \text{s.t.} \quad & \nabla f_j(x)^T \nu \le \delta, \qquad j = 1,\dots,k, \\ & h_j(x) + \nabla h_j(x)^T \nu = 0, \qquad j = 1,\dots,p. \end{aligned} \tag{4.48}$$

Note that (4.48) belongs to the QP category as the corresponding SD version of the direction subproblem for the unconstrained case. The stopping criteria and the step length control do not need to suffer any modification.

If, on the other hand, we aim for faster rates of convergence, a QN realization would be a better choice. Here, the formulas for direction (ν_μ) and orientation (μ) remains as in (4.14) and (4.24) respectively, except for the term W_α that is replaced by

$$B_\alpha = \sum_{j=1}^{k} \alpha_j B_j \in \mathbb{R}^{n\times n}, \tag{4.49}$$

where $B_j \approx \nabla^2 f_j(x)$. Since we have decided to approximate the equality constraints by linear functions, no approximation of the Hessians of the equalities is required and B_α remains exactly as in (3.65). Initially, we take $B_j = B_\alpha = I$ leading to the SD version of the predictor. For subsequent iterations, though, we expect better approximations of B_α coming from the corrector phase. For this, we define the search direction subproblem as

$$\begin{array}{lll} \min\limits_{(\nu,\delta)\in\mathbb{R}^n\times\mathbb{R}} & \delta & \\ \text{s.t.} & \nabla f_j(x)^T\nu + \frac{1}{2}\nu^T B_j \nu \leq \delta, & j = 1,\ldots,k, \\ & h_j(x) + \nabla h_j(x)^T \nu = 0, & j = 1,\ldots,p, \end{array} \tag{4.50}$$

which differs only from the Newton direction subproblem by the use of $B_j \approx \nabla^2 f_j(x)$. Consistently with the predictor step, no update of the Hessians of the equalities is required so we apply the BFGS formula (Equation (3.68)) only to the Hessians of the objective functions. This time, we cannot say much about the convergence rate of both SD and QN versions of the method, although a superiority of the QN approach can be observed in the numerical results presented below. About the computational complexity, the QN direction subproblem again belongs to SOCP which certainly requires more effort than solving the SD direction subproblem. Moreover, modified Cholesky factorizations would also be required on $B_1,\ldots,B_k$ to ensure that (4.50) has a unique solution which is also a descent direction. Under this scenario, a detailed analysis of the convergence rate of this modification is basically a necessity in order to justify—beyond numerical results—the additional overhead required by the QN realizations. Finally, the same trick of updating the Hessians at predicted points using the information gathered from the previous corrector seems to work likewise in this context. With this, the zigzag behavior illustrated in Figure 3.6 as a consequence of resetting the Hessian information at each predictor (by taking $\nabla^2 f_j(p) \approx I$, $j = 1,\ldots,k$) is partially or completely eliminated yielding improved results as for the unconstrained case.

4.5 Numerical Results

In the following, we present some numerical results of the PT for problems with equality constraints. Since the Hillermeier method can also handle this type of problems, we will use it again in a comparison with the three versions of the PT (that we will continue to call PT-N, PT-QN and PT-SD). The values of the parameters have been taken as in the previous chapter (they are all specified when they were first mentioned). As stated as well in Chapter 3, the implementation of the method of Hillermeier follows the instructions given in [28] and uses the Levenberg-Marquardt method as corrector setting the default tolerances for termination (10^{-6}). For all methods, we have taken the same initial Pareto optimal solution as starting point.

We will present next four examples (three of them with two objectives and the other one with three objectives). The respective number of solutions and average number of corrector and backtrack iterations are summarized in a table together with the total number of function, Jacobian and Hessian evaluations. We do not show the number of evaluations of the equality functions neither their respective number of Jacobian evaluations. These indicators, however, coincide or are close to the number of function and Jacobian evaluations of the objectives. Recall that in the case of the PT, the Hessians of the equalities are not utilized at all, which represents an advantage over the Hillermeier method. It is not clear, though, whether this information is required to guarantee a competitive rate of convergence. The backtrack information is omitted also for the method of Hillermeier since the Levenberg-Marquardt method is not a line search but a trust-region strategy. The fields related to the Hessian information are also left empty for the PT-QN and the PT-SD, respectively. Additionally, we present graphically the approximations of the solution set (Pareto set and front) obtained by the PT-QN.

4.5.1 Example 1

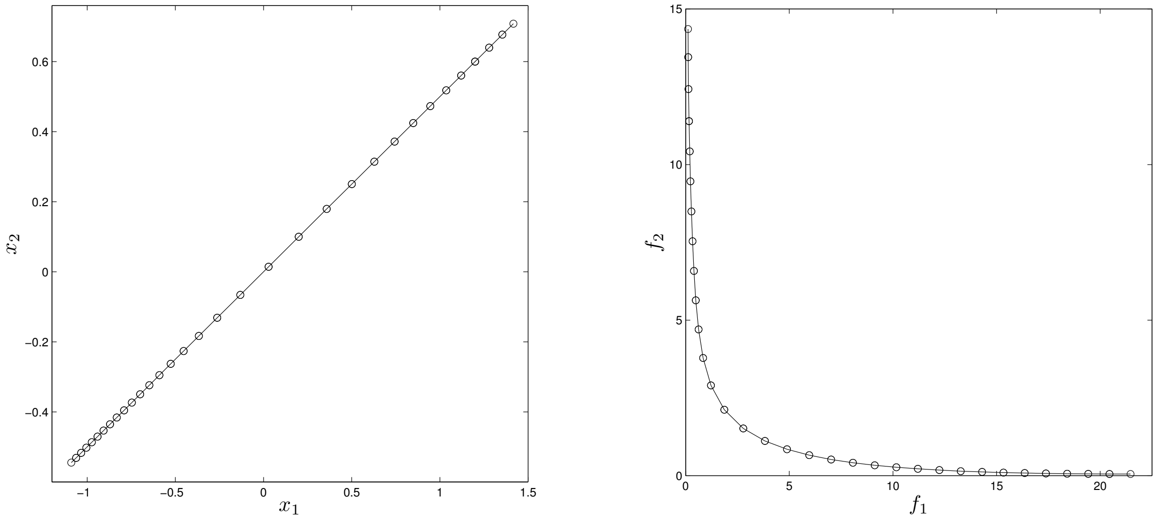
We start with the following problem from [33] subject to one linear equality constraint

$$\begin{aligned} f_j(x) &= \sum_{\substack{i=1\\ i\neq j}}^{100} (x_i - a_i^j)^2 + (x_j - a_j^j)^4, \qquad j = 1, 2, \\ \text{s.t.} \quad & \tfrac{1}{2}x_1 = x_2. \end{aligned} \tag{4.51}$$

Here, $a^1 = (1, \dots, 1)^T \in \mathbb{R}^{100}$, $a^2 = -a_1$, and the step length for the continuation in objective space was chosen as $\tau = 1$. The Pareto set of this example is a line so we expect that the PT requires no corrector at all. Our expectations are confirmed by the results illustrated in Table 4.1 where the costs of the algorithm of Hillermeier are also recorded in the first column. Further, Figure 4.1 shows graphically the numerical results of the PT-QN. Consistently with the previous chapter, we see that the strategy followed by Hillermeier leads to more corrector iterations due to the consideration of the compound space (x, α, λ) which certainly increases the nonlinearity of the optimal set (see the first numerical example of Chapter 3). Observe also in Table 4.1 that the PT-QN needs 0.3125 correctors in average. To clarify this result we would like to point out that the PT-QN actually needs one corrector at one of the extremes of the Pareto set. This happens due to our ignorance of the total arc length of the curve allowing predictors to be located *beyond* the limits of the solution manifold. In such cases, the algorithm steps back in one or more corrector iterations to reach (again) the Pareto set/front. The same situation occurs for the SD version of the method which in addition requires one backtrack iteration.

	Hillermeier	PT-N	PT-QN	PT-SD
Solutions	34.0000	33.0000	33.0000	33.0000
Avg. corrector iterations	3.4412	0.0000	0.3125	0.3125
Avg. backtrack iterations	-	0.0000	0.0000	0.3125
Function evaluations	34.0000	33.0000	34.0000	35.0000
Jacobian evaluations	152.0000	33.0000	34.0000	34.0000
Hessian evaluations	304.0000	66.0000	-	-

Table 4.1: Computational efforts of the four PC variants on the BOP (4.51).



(a) decision space

(b) objective space

Figure 4.1: Numerical results of the PT-QN on the BOP (4.51).

4.5.2 Example 2

Next we consider the BOP subject to one quadratic constraint

$$\begin{array}{ll} f_j(x) = \sum_{i=1}^{100}(x_i - a_i^j)^2, & j = 1, 2, \\ \text{s.t.} \quad \|x - c\|^2 = r^2, & \end{array} \tag{4.52}$$

where

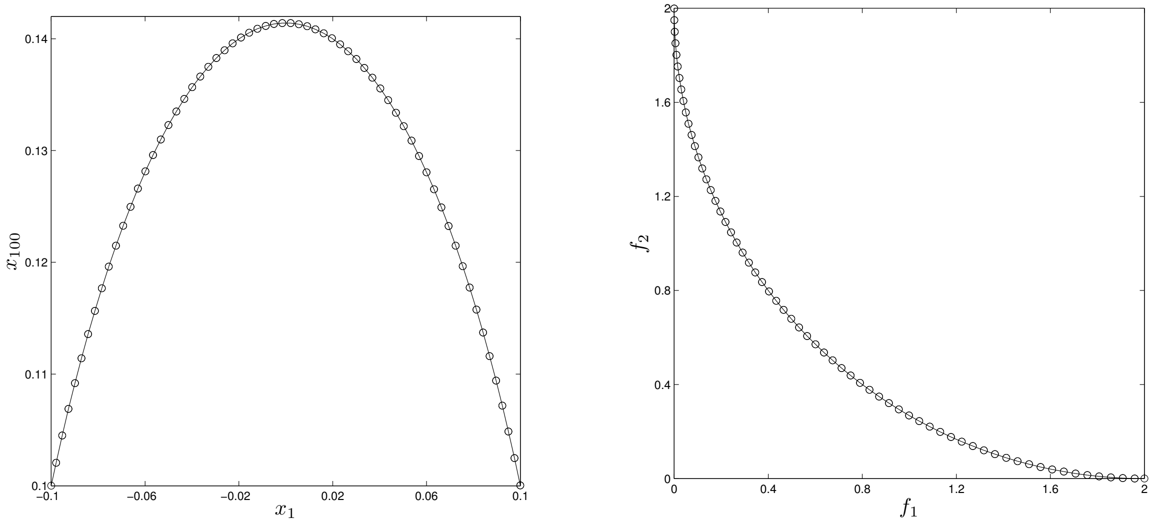
$$a^1 = \frac{1}{\sqrt{100}}(1, \ldots, 1)^T \in \mathbb{R}^{100}, \quad \text{and} \quad a^2 = \frac{1}{\sqrt{100}}(\underbrace{1, \ldots, 1}_{50\,\text{times}}, \underbrace{-1, ..., -1}_{50\,\text{times}})^T \in \mathbb{R}^{100}.$$

We also take $c = (0, ..., 0)^T \in \mathbb{R}^{100}$ and $r = 1$. See Figure 4.2 and Table 4.2 for the results of the PT-QN and the computational efforts, respectively for $\tau = 0.05$. Observe that even when the solution set is nonlinear, the PT-N needs less correctors and in consequence less Jacobian and Hessian evaluations than the classical method. Also recall that the Hillermeier approach is more likely to require less evaluations of the objectives since the map $\tilde{F}$ is in terms only of the Jacobian and the Hessians of the problem. For the latter, as the table shows, the number of function evaluations coincides with the number of solutions at least for the bi-objective case. Thus, given that we are considering several performance indicators, the reader may have noted that deciding the overall winner is also a MOP. In our opinion, the most reliable choice continues to be the PT-QN since without second-order information, the resulting number of function and Jacobian evaluations are not far from those of the PT-N, and in particular, the number of evaluations of the Jacobian is much lower than what the Hillermeier method requires. However, if the exact Hessians are available, taking the PT-N seems to be the best alternative since only the number of function evaluations is outperformed.[1] A completely atypical situation occurs here with the SD approach: it requires less correctors (including backtrack iterations) than the method of Hillermeier. This behavior is not observed in general but the opposite case: the PT-SD is usually the method showing the worst performance, unless of course, the considered solution set is linear or nearly linear.

	Hillermeier	PT-N	PT-QN	PT-SD
Solutions	63.0000	64.0000	64.0000	64.0000
Avg. corrector iterations	3.0000	1.0000	1.1270	1.8095
Avg. backtrack iterations	-	0.0000	0.0000	0.0079
Function evaluations	63.0000	127.0000	135.0000	179.0000
Jacobian evaluations	257.0000	127.0000	135.0000	178.0000
Hessian evaluations	514.0000	254.0000	-	-

Table 4.2: Computational efforts of the four PC variants on the BOP (4.52).

[1] The PT-QN needs no Hessian evaluations but instead requires BFGS updates. If evaluating the Hessians is very costly, the QN approach may be preferable. Otherwise, the use of exact second-order information (if available) may lead to better rates of convergence.



(a) decision space

(b) objective space

Figure 4.2: Numerical results of the PT-QN on the BOP (4.52).

4.5.3 Example 3

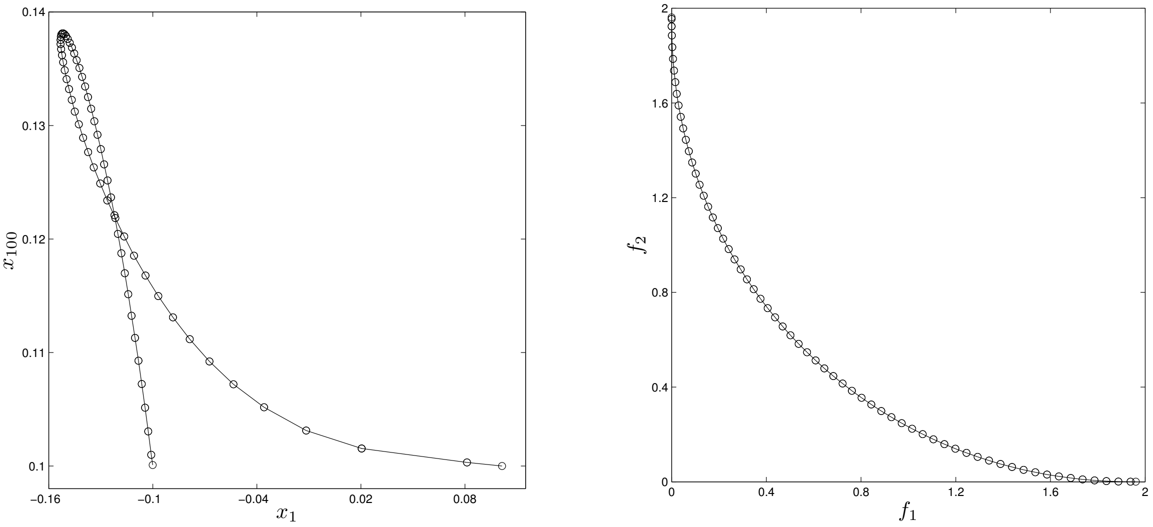
We will now reconsider the first example subject to a quadratic equality constraint. The problem thus reads as

$$\begin{aligned} f_j(x) &= \sum_{\substack{i=1\\ i\neq j}}^{100} (x_i - a_i^j)^2 + (x_j - a_j^j)^4, \qquad j = 1, 2, \\ \text{s.t.} \quad & \|x - c\|_2^2 = r^2, \end{aligned} \tag{4.53}$$

where a^1 and a^2 as well as c and r are taken as in the previous example. Figure 4.3 shows the numerical results of the PT-QN for $\tau = 0.05$ and Table 4.3 shows the computational costs for all methods. Again, the PT-N wins (except for the required number of function evaluations) over the classical approach due to the fewer number of correction steps that had to be performed. Since the Pareto set is nonlinear, the PT-QN is outperforming the PT-SD as well as all other methods in terms of total number of function evaluations (including objective functions, Jacobian and Hessians). In this case, a total of 189 function and 189 Jacobian evaluations (plus the detection of a Pareto optimal solution) were sufficient to obtain a suitable representation of the entire Pareto set/front of a problem of hundred variables. Nevertheless, these results are only of local nature unless (as it is the case) we are dealing with a convex problem.

	Hillermeier	PT-N	PT-QN	PT-SD
Solutions	63.0000	64.0000	65.0000	64.0000
Avg. corrector iterations	3.2031	1.7778	1.9375	5.2222
Avg. backtrack iterations	-	0.0000	0.0000	0.0676
Function evaluations	63.0000	176.0000	189.0000	534.0000
Jacobian evaluations	270.0000	176.0000	189.0000	393.0000
Hessian evaluations	540.0000	352.0000	-	-

Table 4.3: Computational efforts of the four PC variants on the BOP (4.53).



(a) decision space

(b) objective space

Figure 4.3: Numerical results of the PT-QN on the BOP (4.53).

4.5.4 Example 4

In order to test a MOP with three objectives, we take the following problem subject to a torus equality constraint

$$\begin{array}{ll} f_1(x) = (x_1-1)^2 + (x_2-1)^2 + (x_3-1)^2, & \\ f_2(x) = (x_1+1)^2 + (x_2+1)^2 + (x_3+1)^2, & \text{and} \\ f_3(x) = (x_1-1)^2 + (x_2+1)^2 + (x_3-1)^2, & \\ \quad \text{s.t.} \quad r^2 - (\sqrt{x_1^2+x_2^2} - R)^2 - x_3^2 = 0, & \end{array} \tag{4.54}$$

where $r = 0.3$ and $R = 0.5$. For $\tau = 0.15$, we can see from Table 4.4 that in accordance with the previous results, the PT-N is the method that takes the fewest number of correctors in average, followed very close by its QN counterpart which requires even less iterations than the approach of Hillermeier. The PT-SD in turn is the least efficient one taking in average more than nine iterations to correct the solutions back to the optimal manifold. However, when we consider the other indicators, the results do not look so good since all versions of the PT used more function evaluations than the algorithm of Hillermeier. The reasons for this are diverse. First, the number of objective evaluations required by the method of Hillermeier, though typically larger than the amount of solutions for $k \geq 3$, is more likely to be lower than the number of evaluations required by the Newton method. The difference here is that the latter evaluates the objectives at each iteration while the former only evaluates the potential solutions to verify whether they were already computed (see Section 3.3 of the previous chapter). In addition, the PT generated more solutions due its capacity of walking backward when the boundaries of the Pareto set are left behind. Nevertheless, we did not observe the best performance of the modified Newton method in this problem. For instance, when a predictor goes beyond the limits of the solution set, it may find a corrector at the opposite extreme of the manifold instead of *actually* stepping back. This occurs more or less frequent depending on the values of τ, i.e., larger values are more prone to cause this behavior. At any case, the method does not show a good rate of convergence if started at a non-efficient point where all the constraints are satisfied. The good news, however, is that—though slow—it follows the solution manifold of the equalities (as a consequence of the equality restriction introduced in the Newton direction subproblem) until it converges to a (local) optimal point. In conclusion, at least for this problem, the PT does (in general) not outperform the method of Hillermeier in terms of costs. Then, in order to overcome this limitation, we propose to work deeper in the convergence properties of our proposal and encourage the design of other equality constrained test problems that fulfill the smoothness assumptions required by continuation methods.

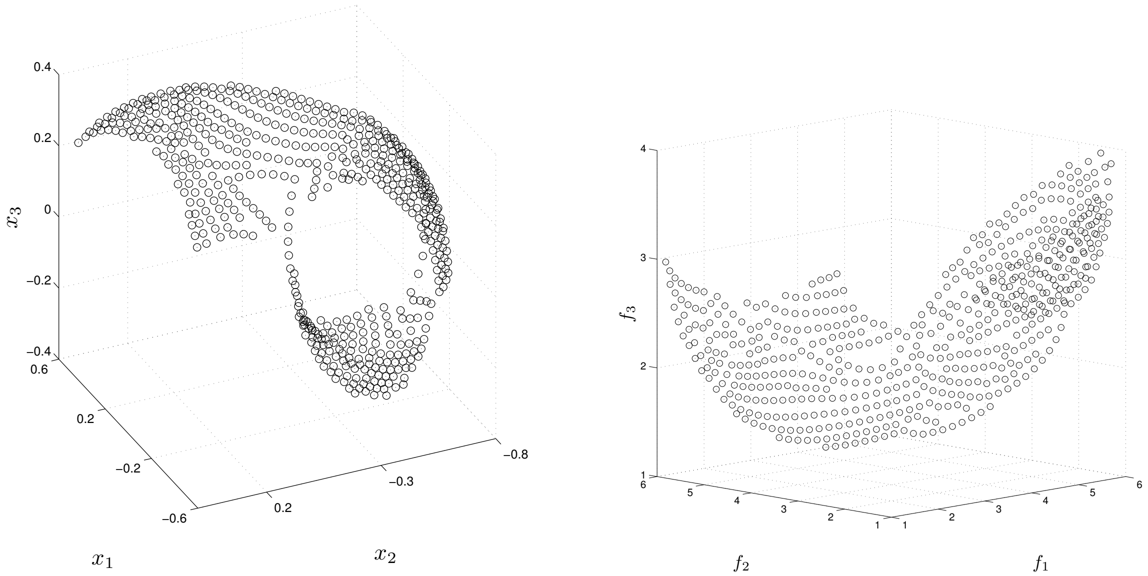
	Hillermeier	PT-N	PT-QN	PT-SD
Solutions	560.0000	600.0000	600.0000	963.0000
Avg. corrector iterations	3.5015	3.1415	3.1462	9.8294
Avg. backtrack iterations	-	0.1755	0.1766	0.6546
Function evaluations	2841.0000	6570.0000	6596.0000	95369.0000
Jacobian evaluations	2927.0000	3397.0000	3405.0000	13906.0000
Hessian evaluations	8781.0000	10191.0000	-	-

Table 4.4: Computational efforts of the four PC variants on the MOP (4.54).

On the other hand, if we analyze other quality indicators as the distribution of solutions in objective space, the PT is clearly the winner. Figure 4.4 shows the approximations of the Pareto set and front obtained by the method of Hillermeier, and in contrast, Figure 4.5 illustrates the numerical results of the PT-QN. Furthermore, this problem has several pieces of local optima connected to the global solution set.[2]

[2] This is the reason why the PT-SD found 963 solutions: it followed a larger piece of local optima.

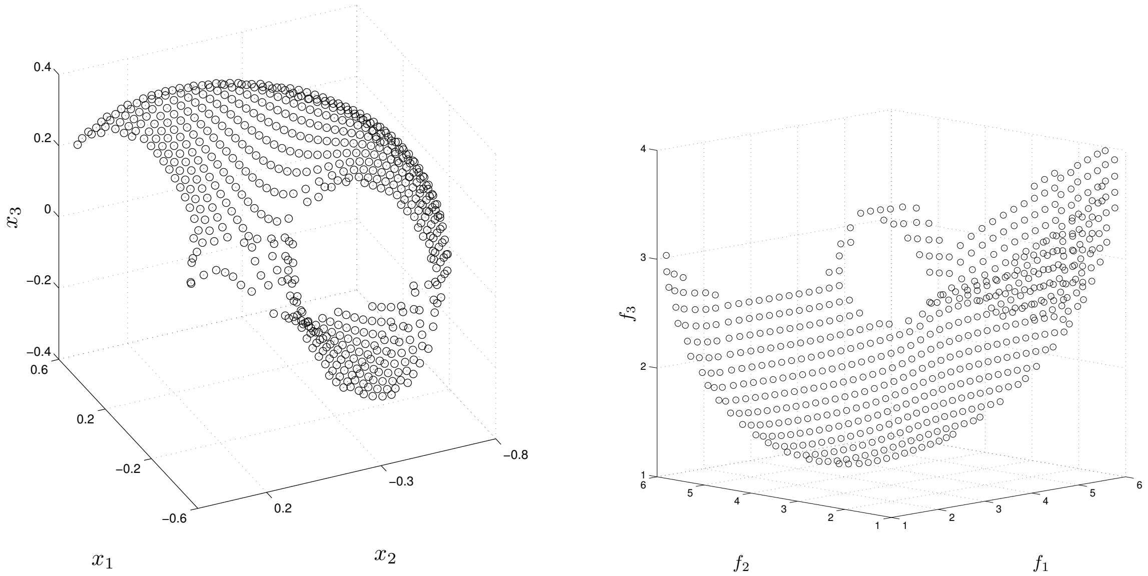
Thus, after applying a *dominance filter* (see e.g. [7, 76] and references therein) to the output of both the Hillermeier and the PT-QN methods, we present their respective results in Figure 4.6 and Figure 4.7.



(a) decision space

(b) objective space

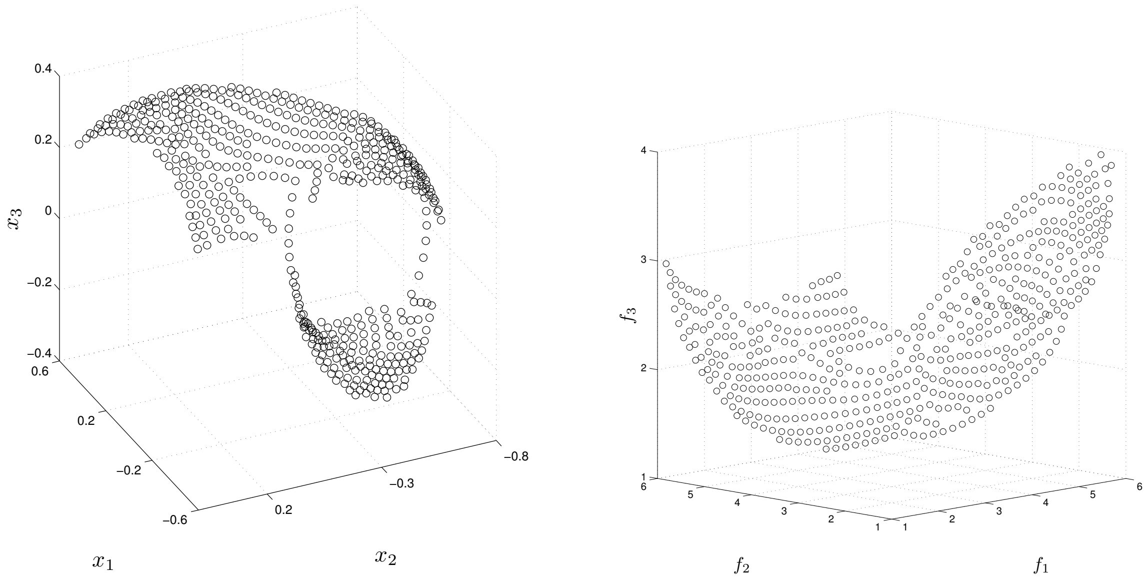
Figure 4.4: Numerical results of the Hillermeier method on the MOP (4.54).



(a) decision space

(b) objective space

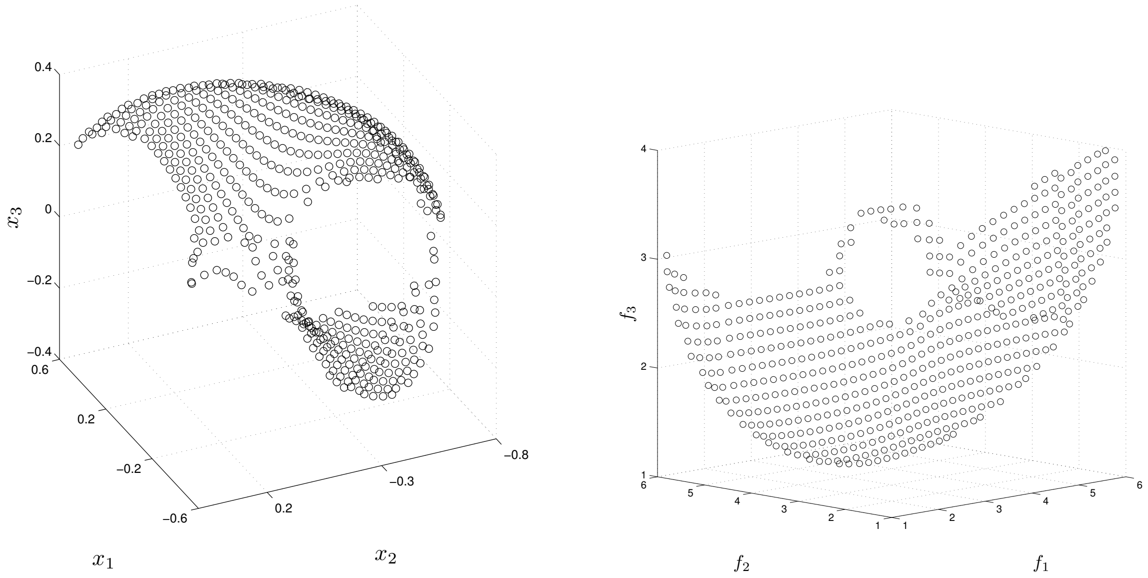
Figure 4.5: Numerical results of the PT-QN method on the MOP (4.54).



(a) decision space

(b) objective space

Figure 4.6: Filtered numerical results of the Hillermeier method on the MOP (4.54).



(a) decision space

(b) objective space

Figure 4.7: Filtered numerical results of the PT-QN method on the MOP (4.54).

4.6 Discussion

We have extended the Pareto Tracer method to handle MOPs subject to equality constraints. The method again utilizes classical continuation techniques on a map $\tilde{F}$ induced by the first-order optimality conditions for equality constrained MOPs. However, unlike the approach proposed in [28], the Pareto Tracer still separates parameters from KKT multipliers leading to a reduction of the total number of function evaluations in several test cases. The orientation vector μ—resulting from the above-mentioned separation—emerges as well in this context allowing to steer the search with a focus on an even distribution of points in objective space. For the corrector, a modified version of the Newton method [16] is proposed capable to deal with equality restrictions. The numerical results were very promising in several BOPs but the method presented problems in the last example that considers three objectives. This scenario encourages a further study of the convergence rate of the modifications in order to improve the efficiency of the new method. Note, though, that the Pareto Tracer continues to show the best quality of the solutions (with regard to distribution) due to its facility to steer the search in both decision and objective space. Another *advantage* of this proposal is that it completely disregards the Hessians of the constraints, even though we cannot (yet) establish the relation between this information and the rate of convergence of the corrector. In addition, the new version of the Newton method for equality constrained MOPs comes together with a modified Armijo condition concerning the estimated decrease of both the objectives and a merit function based on the equalities. This should suffice those cases where the objective functions cannot be improved in order to reach the feasible region. However, one drawback of the modified Newton direction (possibly the cause of the slow convergence observed in one of the examples) is that starting at a non-efficient point that satisfies the constraints, it will follow a path on the manifold defined by the equalities to finally converge to a (local) solution. On the other hand, two Hessian-free realizations of the algorithm were also developed based on quasi-Newton and steepest descent methods, respectively. The computational time complexity of these approaches is still an issue and recommended for a future investigation.

Chapter 5

Handling Inequality Constrains

Finally, this chapter is dedicated to the treatment of MOPs subject to inequality constraints. More precisely, we consider here problems of the form

$$\begin{aligned} &\min_{x\in\mathbb{R}^n} F(x), \\ &\text{s.t. } g_j(x) \leq 0, \qquad j = 1, \ldots, m. \end{aligned} \tag{5.1}$$

Handling such problems is not an easy topic in the context of continuation since the inequalities can turn the followed manifold non-smooth at points where the set of active constraints changes. One way to deal with those singularities is maintaining the set of restrictions that are currently active in order to detect and manage such changes of activity. We will denote this set at a point $x \in \mathbb{R}^n$ as

$$\mathcal{A}(x) = \{j | g_j(x) = 0, \quad j = 1, \ldots, m\}. \tag{5.2}$$

This has been implemented in [92] for parametric optimization and later in [27] for bi-objective optimization which inspired the works presented in [29] and [31] also for the bi-objective case. The general approach of Hillermeier [28], however, is not designed to handle problems that include inequality constraints. See Chapter 2 for more details. Here, we will utilize the fact (as did by the aforementioned techniques) that applying a continuation on the full KKT system of equations is equivalent to deal with a reduced system $\tilde{F}$ involving only the active constraints (see [92, 27, 29, 31] and references therein). We thus proceed for the predictor as for the equality constrained case but on the reduced map $\tilde{F}$ resulting from taking the active inequalities as equalities. In consequence, from the separation of parameters from Lagrange multipliers, the orientation vector μ emerges once more serving as a bridge between tangent directions in objective and decision space. For the corrector, a further modification of the Newton method [16] that considers the current active set is suggested. This approach, though, bears little relation to the modified Newton method proposed in Chapter 4 for equality constrained MOPs. Instead, it is closer in spirit to the steepest descent procedure designed by Fliege and Svaiter [18] to manage MOPs restricted by inequalities (see Chapter 2). Our proposal (as in [18]) is a method of feasible directions, meaning that the initial point and the subsequent iterations must be feasible.

This, regrettably, represents a clear limitation in the current context since so far we cannot guarantee that the predictor is a feasible point. Thus, for practical purposes, we will limit the Pareto Tracer to box constrained MOPs, i.e.,

$$\min_{l\leq x\leq u} F(x), \tag{5.3}$$

and utilize gradient projection methods to ensure the feasibility of the predicted points. Nevertheless, we will continue to show several theoretic aspects for the general case. A detailed analysis of the convergence rate for the suggested corrector is also delegated in this chapter for a future work. Though, we provide as well theoretical and numerical evidence that support our proposal. We also encourage an in-depth study of the time complexity of the extended Pareto Tracer for inequality constrained MOPs. Further, we recommend to include in the analysis the Hessian-free versions derived from quasi-Newton and steepest descent realizations of the method.

The remainder of this chapter is organized as follows. The computation of predictors for inequality constrained MOPs is discussed in Section 5.1. Additionally, in the same section, the particular case of MOPs subject to box constraints is considered. We cover the corrector phase in Section 5.2 for the general case first and then for problems with box restrictions. Subsequently, the Pareto Tracer for inequality constrained MOPs as well as its Hessian-free realizations are discussed in Section 5.3 and Section 5.4, respectively. Some numerical examples are presented in Section 5.5 and we finally conclude in Section 5.6.

5.1 Predictor

In this section, we will start by computing tangent directions to the Pareto set of general inequality constrained MOPs. As stated in [92] and [27], this task can be accomplished by considering the reduced KKT system of equations including exclusively the active constraints at the current solution. Due to numerical issues, we will take in our computations the set of active and nearly active constraints at a given point $x \in \mathbb{R}^n$, that is

$$\mathcal{I} = \mathcal{I}(x) = \{j \mid g_j(x) > -\epsilon, \quad j = 1, \ldots, m\} \supset \mathcal{A}(x) \tag{5.4}$$

for some $\epsilon > 0$ (taken as $\epsilon = \sqrt{\varepsilon} = \sqrt[4]{25 \times \text{eps}}$). Thus, such a reduced KKT map $\tilde{F} : \mathbb{R}^{n+k+s} \to \mathbb{R}^{n+s+1}$ would read as

$$\tilde{F}(x, \alpha, \gamma) = \begin{pmatrix} \sum_{j=1}^{k} \alpha_j \nabla f_j(x) + \sum_{j\in\mathcal{I}} \gamma_j \nabla g_j(x) \\ g_{j\in\mathcal{I}}(x) \\ \sum_{j=1}^{k} \alpha_j - 1 \end{pmatrix} = 0, \tag{5.5}$$

where $s = |\mathcal{I}|$. Further $\alpha \in \mathbb{R}^k$ and $\gamma \in \mathbb{R}^s$ are the associated Lagrange multipliers such that $\alpha_j \geq 0$ for all $j = 1, \ldots, k$. See Theorem 2.1.2 for more details on the first-order conditions of optimality. Analog to the equations (4.3) and (4.4) in Chapter 4, we define

$$W_{\alpha,\gamma} = \sum_{j=1}^{k} \alpha_j \nabla^2 f_j(x) + \sum_{j\in\mathcal{I}} \gamma_j \nabla^2 g_j(x) \in \mathbb{R}^{n\times n}, \quad \text{and} \tag{5.6}$$

$$G_{\mathcal{I}} = \left(\nabla g_j(x)^T\right)_{j\in\mathcal{I}} \in \mathbb{R}^{s\times n}, \tag{5.7}$$

leading to

$$\tilde{F}'(x,\alpha,\gamma) = \begin{pmatrix} W_{\alpha,\gamma} & J^T & G_{\mathcal{I}}^T \\ G_{\mathcal{I}} & 0 & 0 \\ 0 & 1\dots1 & 0 \end{pmatrix} \in \mathbb{R}^{(n+s+1)\times(n+k+s)}. \tag{5.8}$$

Since we are seeking for kernel vectors of (5.8)—which are tangent to the augmented solution manifold—we have to solve

$$\begin{pmatrix} W_{\alpha,\gamma} & J^T & G_{\mathcal{I}}^T \\ G_{\mathcal{I}} & 0 & 0 \\ 0 & 1\dots1 & 0 \end{pmatrix} \begin{pmatrix} \nu \\ \mu \\ \varsigma \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}, \tag{5.9}$$

where $\nu \in \mathbb{R}^n$, $\mu \in \mathbb{R}^k$, and $\varsigma \in \mathbb{R}^s$. Then, after choosing a value of μ that satisfies (3.5), i.e., $\sum_{j=1}^{k} \mu_j = 0$, the system (5.9) gets reduced to

$$\begin{pmatrix} W_{\alpha,\gamma} & G_{\mathcal{I}}^T \\ G_{\mathcal{I}} & 0 \end{pmatrix} \begin{pmatrix} \nu_\mu \\ \varsigma \end{pmatrix} = -\begin{pmatrix} J^T \\ 0 \end{pmatrix} \mu. \tag{5.10}$$

Note that (5.10) is identical to (4.7) except for the submatrices involved. Thus, we can use the same arguments as above to show that if $\text{rank}(W_{\alpha,\gamma}) = n$ and $\text{rank}(G_{\mathcal{I}}) = s$, the matrix in the left hand side of (5.10) is regular and the solution of the system is unique. Moreover, ν_μ will be a nontrivial tangent to the Pareto set of the MOP (5.1) unless there is a change of activity at x (or $\mu = 0$). Figure 5.1 illustrates a concrete example of this case taken from [31]. Consider the constrained BOP of two variables

$$\begin{aligned} & f_1(x) = x_1, \quad \text{and} \\ & f_2(x) = x_2, \\ & \text{s.t.} \qquad 14 - x_1^2 - x_2^2 \le 0, \\ & \qquad 5e^{-x_1} + 2e^{-0.5(x_1-3)^2} - x_2 \le 0, \\ & \qquad x_1 - 5 \le 0, \\ & \qquad x_2 - 5 \le 0. \end{aligned} \tag{5.11}$$

The points a, b, c, and d in the figure are called turning points, i.e., non-smooth points where the flow of the continuation locally changes. This may have several causes but in this particular context we will restrict to (i) transition points between the pieces of the solution set defined by the active constraints (b, c) and (ii) extreme points (a, d). Take e.g. c, where the first two constraints are active. At this point the first constraint becomes active while the second one changes to inactive along the solution manifold. Given the derivatives of both constraints at c, it is easy to see that there is no vector $\nu \neq 0$ such that $G_{\mathcal{I}}\nu = 0$. Thus, the system (5.10) can only be satisfied for $\nu_\mu = 0$. Nevertheless, the probability to reach exactly one of this turning points is extremely remote.

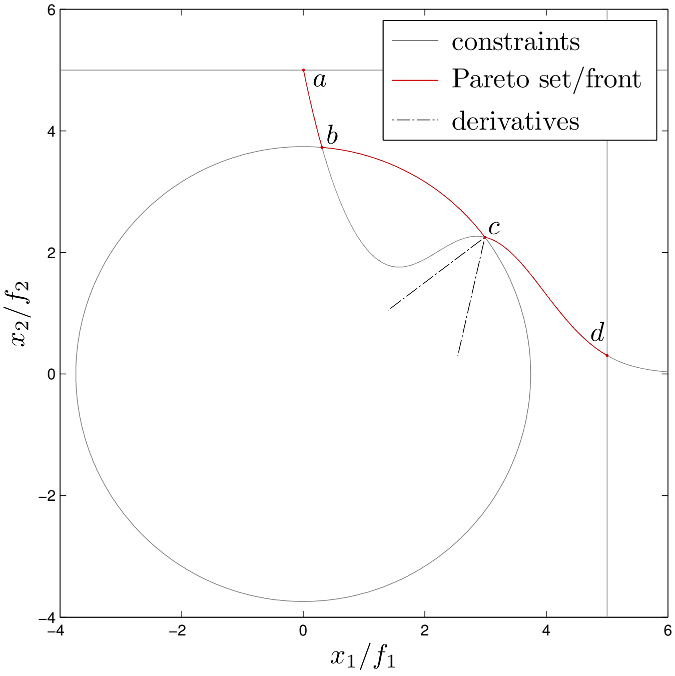


Figure 5.1: Turning points on the Pareto set and front of the BOP (5.11).

Additionally, in order to avoid dealing with the Hessians of the inequalities, we have made one modification to (5.10) (as done for (4.7) in Chapter 4) consisting of replacing $W_{\alpha,\gamma}$ by W_α in the computations. This means that only first-order approximations of the constraints are considered and therefore the tangent directions will be computed by

$$\begin{pmatrix} \nu_\mu \\ \varsigma \end{pmatrix} = -\begin{pmatrix} W_\alpha & G_{\mathcal{I}}^T \\ G_{\mathcal{I}} & 0 \end{pmatrix}^{-1} \begin{pmatrix} J^T \\ 0 \end{pmatrix} \mu. \tag{5.12}$$

In spite of a more careful analysis should be done on the impact of removing the second-order terms of the constraints, the numerical results so far are very promising since the change from (5.10) to (5.12) leads to nearly the same performance.

On the other hand, if we restrict the inequalities to box constraints, the auxiliary map $\tilde{F}$ would be given by

$$\tilde{F}(x,\alpha,\rho,\varrho) = \left(\begin{array}{c} \sum\limits_{j=1}^{k} \alpha_j \nabla f_j(x) - \sum\limits_{i\in\mathcal{I}_l} \rho_i e_i + \sum\limits_{i\in\mathcal{I}_u} \varrho_i e_i \\ (-x_i + l_i)_{i\in\mathcal{I}_l} \\ (x_i - u_i)_{i\in\mathcal{I}_u} \\ \sum_{j=1}^{k} \alpha_j - 1 \end{array} \right) = 0, \tag{5.13}$$

where

$$\begin{aligned} \mathcal{I}_l &= \{ i \mid -x_i + l_i > -\epsilon, \quad i = 1,\ldots,n \}, \quad \text{and} \\ \mathcal{I}_u &= \{ i \mid \ \ x_i - u_i > -\epsilon, \quad i = 1,\ldots,n \} \end{aligned} \tag{5.14}$$

are the sets of active and nearly active lower and upper box constraints, respectively. It is also

$$\mathcal{I}_{l,u} = \{ i \mid i \in \mathcal{I}_l \quad \text{or} \quad i \in \mathcal{I}_u \} \tag{5.15}$$

the set of active and nearly active box constraints. We take $r = |\mathcal{I}_{l,u}|$ and $(\rho,\varrho) \in \mathbb{R}^r$ as the Lagrange multipliers associated with the active box constraints. Furthermore, e_i stands for the i-th canonical vector of the appropriate length. For the sake of simplicity, we will define the matrix $I_{l,u} \in \mathbb{R}^{r\times n}$ as

$$(I_{l,u})_{j_i} = \begin{cases} -e_i^T, & i \in \mathcal{I}_l \\ \ \ e_i^T, & i \in \mathcal{I}_u, \end{cases}, \qquad j = 1,\ldots,r, \tag{5.16}$$

where $(A)_j$ denotes the j-th row of A. Then, $\tilde{F}'$ is given by

$$\tilde{F}'(x,\alpha,\rho,\varrho) = \begin{pmatrix} W_\alpha & J^T & I_{l,u}^T \\ I_{l,u} & 0 & 0 \\ 0 & 1\ldots1 & 0 \end{pmatrix} \in \mathbb{R}^{(n+r+1)\times(n+k+r)}. \tag{5.17}$$

To obtain a kernel vector of $\tilde{F}'$, let $\nu \in \mathbb{R}^n$, $\mu \in \mathbb{R}^k$, and $\eta \in \mathbb{R}^r$ such that

$$\begin{pmatrix} W_\alpha & J^T & I_{l,u}^T \\ I_{l,u} & 0 & 0 \\ 0 & 1\ldots1 & 0 \end{pmatrix} \begin{pmatrix} \nu \\ \mu \\ \eta \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}. \tag{5.18}$$

By choosing another time a μ with components that sum up to zero, Equation (5.18) is reduced to

$$\begin{pmatrix} W_\alpha & I_{l,u}^T \\ I_{l,u} & 0 \end{pmatrix} \begin{pmatrix} \nu_\mu \\ \eta \end{pmatrix} = \begin{pmatrix} -J^T\mu \\ 0 \end{pmatrix}. \tag{5.19}$$

If $\operatorname{rank}(W_\alpha) = n$ and $\operatorname{rank}(I_{l,u}) = r$, then the solution of (5.19) is unique since its associated matrix is regular.[1] Also from (5.19) we obtain

$$W_\alpha \nu_\mu + I_{l,u}^T \eta = -J^T \mu, \quad \text{and} \tag{5.20a}$$

$$I_{l,u}\nu_\mu = 0, \tag{5.20b}$$

and from (5.20b) that $\nu_\mu^i = 0$ for $i \in \mathcal{I}_{l,u}$. Thus, we only need to compute the j-th components of ν_μ such that $j \notin \mathcal{I}_{l,u}$. We also know that $(I_{l,u}^T\eta)_j = 0$ for $j \notin \mathcal{I}_{l,u}$. Then, the linear system of equations (5.19) is equivalent to

$$W_\alpha^{\mathcal{I}_c}\nu_\mu^{\mathcal{I}_c} = -J_{\mathcal{I}_c}^T\mu, \qquad \mathcal{I}_c = \{1, \ldots, n\} \setminus \mathcal{I}_{l,u}, \tag{5.21}$$

where $W_\alpha^{\mathcal{I}_c}$ is the matrix that results from W_α where the i-th row and column vectors are deleted for all $i \in \mathcal{I}_{l,u}$. Analog, $\nu_\mu^{\mathcal{I}_c}$ results from ν_μ and $J_{\mathcal{I}_c}$ from J by deleting the i-th element respectively the i-th column vector. Observe that if no constraint is active, the system (5.21) coincides with the formula for tangent vectors of unconstrained MOPs (see Equation (3.8)). Thus, in case $\nu_\mu^j = 0$ for all $j \notin \mathcal{I}_{l,u}$, we compute the predictor as for the unconstrained case (see Chapter 3) and then project it onto the box, i.e.,

$$p_i = \begin{cases} l_i, & \tilde{p}_i < l_i \\ \tilde{p}_i, & \tilde{p}_i \in [l_i, u_i] \\ u_i, & \tilde{p}_i > u_i, \end{cases}, \qquad i = 1, \ldots, n, \tag{5.22}$$

where $\tilde{p}$ is the computed unrestricted predictor. The latter is actually an unusual case that takes place when e.g. we are in a corner of the box (n constraints are active). In such situations, computing a direction without taking the constraints into account and then projecting the resulting predictor onto to the box (to keep it feasible) seems to be more promising than just stopping the method.

It is then time for the second phase in the computation of predictors: to provide the appropriate orientation to our tangent vectors depending on a desired result in objective space. That is, given a direction $d \in \mathbb{R}^k$ in objective space, we seek for an orientation vector $\mu_d \in \mathbb{R}^k$ such that

$$J\nu_{\mu_d} = d. \tag{5.23}$$

When dealing with the general case, we transform (5.23) into

$$(J\ \ 0)\begin{pmatrix} \nu_{\mu_d} \\ \varsigma \end{pmatrix} = d \tag{5.24}$$

since the solution of (5.10) is a compound vector. Then, substituting (ν_{μ_d}, ς) in Equation (5.24) by its corresponding expression given by (5.10) we obtain

$$-(J\ \ 0)\begin{pmatrix} W_{\alpha,\gamma} & G_{\mathcal{I}}^T \\ G_{\mathcal{I}} & 0 \end{pmatrix}^{-1}\begin{pmatrix} J^T \\ 0 \end{pmatrix}\mu_d = d, \tag{5.25}$$

[1] We used here the same argument to prove that the matrices corresponding to the systems (4.7) and (5.10), respectively, are regular.

which is a linear system of equations of size $\mathcal{O}(k^2)$. As in the previous two chapters, (5.25) has none or an infinite number of solutions since the matrix involved is rank deficient. Here, it is

$$\begin{aligned} A &= -\left(J \ \ 0\right) \begin{pmatrix} W_{\alpha,\gamma} & G_{\mathcal{I}}^T \\ G_{\mathcal{I}} & 0 \end{pmatrix}^{-1} \begin{pmatrix} J^T \\ 0 \end{pmatrix} \\ &= -JWJ^T \in \mathbb{R}^{k\times k}, \end{aligned} \tag{5.26}$$

where $W \in \mathbb{R}^{n\times n}$ is the left top submatrix of

$$\begin{pmatrix} W_{\alpha,\gamma} & G_{\mathcal{I}}^T \\ G_{\mathcal{I}} & 0 \end{pmatrix}^{-1}.$$

Analog to the discussion in Chapter 4, it is possible to prove that $A\alpha = 0$ and thus (5.25) has an infinity of solutions. Again, we will take

$$\mu_d = \mu_d^+ - (e^T\mu_d^+)\alpha, \tag{5.27}$$

where

$$\mu_d^+ = A^+d. \tag{5.28}$$

Let's suppose in the following that our problem comes only with box restrictions as in the MOP (5.3). If we substitute ν_{μ_d} in Equation (5.23) by its expression given in (5.21) for box constrained problems, we get

$$J\nu_{\mu_d} = -J_{\mathcal{I}_c}\left(W_\alpha^{\mathcal{I}_c}\right)^{-1} J_{\mathcal{I}_c}^T\mu_d = d. \tag{5.29}$$

Note that $J_{\mathcal{I}_c} \in \mathbb{R}^{k\times(n-r)}$, $W_\alpha^{\mathcal{I}_c} \in \mathbb{R}^{(n-r)\times(n-r)}$, and the r components of ν_{μ_d} corresponding to the active constraints are zero. Then, by a similar reasoning to that provided in Chapter 3, we will compute

$$\nu_{\mu_d}^{\mathcal{I}_c} = -\left(W_\alpha^{\mathcal{I}_c}\right)^{-1} J_{\mathcal{I}_c}^T\mu_d^+, \tag{5.30}$$

where

$$\mu_d^+ = A^+d = \left(J_{\mathcal{I}_c}\left(W_\alpha^{\mathcal{I}_c}\right)^{-1} J_{\mathcal{I}_c}^T\right)^+ d. \tag{5.31}$$

Finally, the predictor stage ends by computing a sequence of tangent vectors $\nu_1, \ldots, \nu_{k-1}$ that leads to a set of movements $d_1, \ldots, d_{k-1}$ in objective space belonging to an orthonormal basis of the linearized Pareto front at $F(x)$. The steps to achieve our goal are simple: (i) we compute a set of vectors $d_1, \ldots, d_{k-1}$ as in Equation (3.24), (ii) we compute the corresponding $\mu_{d_1}, \ldots, \mu_{d_{k-1}}$ as in (5.27) or (5.31) depending on whether we are dealing with the general case or a box constrained problem, and (iii) we obtain the tangent vectors $\nu_{\mu_{d_1}}, \ldots, \nu_{\mu_{d_{k-1}}}$ with the appropriate orientation by the respective formulas (5.12) or (5.30). In case we get empty vectors, assuming that x is a turning point, we can proceed as described above for box constrained problems: computing first the unconstrained tangent and then projecting the resulting predictor onto the box to keep it feasible. However, a more careful analysis should be done for the case of MOPs with general inequality constraints which is not provided in this work. On the other hand, we can instead seek for orthogonal vectors that span the linearized Pareto *set*. Then, we may compute an orthonormal basis of the image of

$$M = -WJ^T \in \mathbb{R}^{n\times k}, \tag{5.32}$$

or alternatively utilize

$$M = -\left(W_\alpha^{\mathcal{I}_c}\right)^{-1} J_{\mathcal{I}_c}^T \in \mathbb{R}^{(n-r)\times k} \tag{5.33}$$

for problems with box restrictions (recall that r components of the tangent vectors will be zero). To obtain a suitable step size and further optimizations of the computational costs for BOPs, we refer the reader to Chapter 3.

5.2 Corrector

In this section, we develop a new modification of the Newton method proposed by Fliege et al. in [16] to deal with problems subject to inequality restrictions. The new proposal resembles the steepest descent method for constrained MOPs designed by Fliege and Svaiter in [18]. However, we include second-order information of the objectives and a different design of the search direction subproblem by including the terms $g_j(x)$ for $j \in \mathcal{I}$ in the inequalities. More precisely, we suggest to compute the Newton direction for (5.1) by solving

$$\begin{array}{lll} \min\limits_{(\nu,\delta)\in\mathbb{R}^n\times\mathbb{R}} & \delta & \\ \text{s.t.} & \nabla f_j(x)^T\nu + \frac{1}{2}\nu^T\nabla^2 f_j(x)\nu \leq \delta, & j = 1,\ldots,k, \\ & g_j(x) + \nabla g_j(x)^T\nu \leq 0, & j \in \mathcal{I}. \end{array} \tag{5.34}$$

The difference of (5.34) with the unconstrained version (given by (3.34)) is the consideration of the additional inequalities $g_j(x) + \nabla g_j(x)^T\nu \leq 0$ corresponding to the active constraints, which also differ from the design in [18] by the term $g_j(x)$. Here, as in [18], we are proposing a method of feasible directions, meaning that the initial point and the subsequent iterations must remain feasible. Additionally, the variable δ still represents the expected decrease in objective space by performing a line search in direction ν, and will be utilized as well by the stopping criteria and the step length control. We will delegate the analysis of the convergence rate for a future study and will give in the following some theoretical arguments that help to support the convergence of the method toward KKT points.

Proposition 5.2.1
Let $x \in \mathbb{R}^n$ be a feasible point of (5.1) *and let the objective functions be strictly convex. Further, let (ν^*, δ^*) be the solution of* (5.34).

(*a*) *If $\nu^* = 0$, then $\delta^* = 0$ and x is a KKT point of* (5.1).

(*b*) *If $\nu^* \neq 0$, then ν^* is a descent direction for F at x.*

Proof The Lagrangian of (5.34) is given by

$$\begin{aligned}\mathcal{L}((\nu,\delta),\alpha,\gamma) = \delta &+ \sum_{j=1}^{k} \alpha_j \left(\nabla f_j(x)^T \nu + \frac{1}{2}\nu^T \nabla^2 f_j(x)\nu - \delta \right) \\ &+ \sum_{j\in\mathcal{I}} \gamma_j \left(g_j(x) + \nabla g_j(x)^T \nu \right).\end{aligned} \tag{5.35}$$

Then, the KKT conditions for (5.34) read as

$$\nabla_\nu \mathcal{L} = \sum_{j=1}^{k} \alpha_j \left(\nabla f_j(x) + \nabla^2 f_j(x)\nu^* \right) + \sum_{j\in\mathcal{I}} \gamma_j \nabla g_j(x) = 0, \tag{5.36a}$$

$$\nabla_\delta \mathcal{L} = 1 - \sum_{j=1}^{k} \alpha_j = 0, \tag{5.36b}$$

$$\nabla f_j(x)^T \nu^* + \frac{1}{2}\nu^{*T} \nabla^2 f_j(x)\nu^* \leq \delta^*, \qquad j = 1,\ldots,k, \tag{5.36c}$$

$$g_j(x) + \nabla g_j(x)^T \nu^* \leq 0, \qquad j \in \mathcal{I}, \tag{5.36d}$$

$$\alpha_j \geq 0, \qquad j = 1,\ldots,k, \tag{5.36e}$$

$$\alpha_j \left(\nabla f_j(x)^T \nu^* + \frac{1}{2}\nu^{*T} \nabla^2 f_j(x)\nu^* - \delta^* \right) = 0, \qquad j = 1,\ldots,k \tag{5.36f}$$

$$\gamma_j \geq 0, \qquad j \in \mathcal{I}, \tag{5.36g}$$

$$\gamma_j \left(g_j(x) + \nabla g_j(x)^T \nu^* \right) = 0, \qquad j \in \mathcal{I}. \tag{5.36h}$$

(*a*) Let $\nu^* = 0$. From (5.36b) and (5.36e) it follows that α is a convex weight and from (5.36a) that

$$\sum_{j=1}^{k} \alpha_j \nabla f_j(x) + \sum_{j\in\mathcal{I}} \gamma_j \nabla g_j(x) = 0.$$

Since x is feasible, $g_j(x) \leq 0$ for all $j = 1,\ldots,m$. Further, by (5.36g) we have that $\gamma_j \geq 0$ and by (5.36h) that $\gamma_j g_j(x) = 0$ for $j \in \mathcal{I}$. Thus, taking $\gamma_j = 0$ for $j \notin \mathcal{I}$ we conclude that x is a KKT point of (5.1). Finally, $\delta^* = 0$ follows from (5.36f).

(*b*) Let $\nu^* \neq 0$. Since x is feasible, it follows that $\nu = 0$ is a feasible direction for (5.34) and thus that $\delta^* \leq 0$. Since further all Hessians are positive definite, we obtain

$$\nabla f_j(x)^T \nu^* \leq \delta^* - \frac{1}{2}\nu^{*T}\nabla^2 f_j(x)\nu^* < 0, \qquad j = 1, \ldots, k,$$

and the claim follows. □

On the other hand, for box constrained MOPs, the Newton direction subproblem (5.34) reads as

$$\begin{aligned} \min_{(\nu,\delta)\in\mathbb{R}^n\times\mathbb{R}} \quad & \delta \\ \text{s.t.} \quad & \nabla f_j(x)^T\nu + \tfrac{1}{2}\nu^T\nabla^2 f_j(x)\nu \leq \delta, && j = 1, \ldots, k, \\ & -\nu_i - x_i + l_i \leq 0, && i \in \mathcal{I}_l, \\ & \nu_i + x_i - u_i \leq 0, && i \in \mathcal{I}_u, \end{aligned} \tag{5.37}$$

which can be stated as

$$\begin{aligned} \min_{(\nu,\delta)\in\mathbb{R}^n\times\mathbb{R}} \quad & \delta \\ \text{s.t.} \quad & \nabla f_j(x)^T\nu + \tfrac{1}{2}\nu^T\nabla^2 f_j(x)\nu \leq \delta, && j = 1, \ldots, k, \\ & I_{l,u}\nu + I_{l,u}(x + l + u) \leq 0. \end{aligned} \tag{5.38}$$

Recall that $x_i, i = 1, \ldots, n$, is not supposed to be active with respect to a lower and an upper bound at the same time. Note also that (5.38) becomes the unconstrained Newton direction subproblem (3.34) if no restriction is active, i.e., x is inside the box. At any case, the stopping criteria can be left as in Chapter 3 given by

$$\delta \geq -\varepsilon, \tag{5.39}$$

where $\varepsilon > 0$ is a small tolerance ($\varepsilon = 5\sqrt{\text{eps}}$). For the step length control, we take the Armijo rule for unrestricted MOPs but impose the following upper bound on the step size:

$$t_{\max} = \min_{i=1,\ldots,n} t_i, \tag{5.40}$$

where

$$t_i = \begin{cases} \dfrac{l_i - x_i}{\nu_i} & \nu_i < 0 \\ \dfrac{u_i - x_i}{\nu_i} & \nu_i > 0 \\ +\infty & \nu_i = 0 \end{cases}, \qquad i = 1, \ldots, n. \tag{5.41}$$

Then, as initial step length, one can take $t = \min(1, t_{\max})$ and the backtracking strategy will be identical to the procedure described in Algorithm 2 for the unconstrained case. The step length control for the general case will be considered in the future (although we may e.g. take the one suggested in [18]).

It only remains to give some instructions on how to solve the Newton direction subproblem. Since (5.38) belongs to the category of QCLP, we will have to shift it to SOCP. As previously done, we will leave the objective function δ as it is and transform the inequalities into second-order cone constraints, i.e.,

$$\|A_j x + b_j\| \leq c_j^T x + d_j, \tag{5.42}$$

where $f, c_j \in \mathbb{R}^{n+1}$, $A_j \in \mathbb{R}^{(n_j-1)\times(n+1)}$, $b_j \in \mathbb{R}^{n_j-1}$, $d_j \in \mathbb{R}$, and j is an index used here for convenience. The first k inequalities (associated with the objective functions of the original MOP) are handled as in the previous two chapters (see the SOCP instances (3.38) and (4.44)), and the last r inequalities (corresponding to the active constraints) can be left basically as they are since most of the solvers can mix linear inequality constraints with the original definition of SOCP. After all, they are a particular case of (5.42) with $A = 0 \in \mathbb{R}^{r\times(n+1)}$, $b = 0 \in \mathbb{R}^r$, $C = (c_j^T)_{j=1,\ldots,r} = -(I_{l,u}\ 0) \in \mathbb{R}^{r\times(n+1)}$,[2] and $d = (d_1, \ldots, d_r)^T = -I_{l,u}(x + l + u) \in \mathbb{R}^r$. The pseudo code of the Newton method for MOPs with box restrictions is basically the procedure described in Algorithm 1 after using the corresponding equations given in this section.

5.3 The Pareto Tracer Method

The Pareto Tracer method for MOPs with general inequality constraints is still incomplete. The main trouble here is that the corrector has been designed as a method of feasible directions and the predictor cannot be guaranteed to be a feasible point. If the inequalities are linear, we can go for a gradient projection method (to project the predictor onto the feasible polytope) but it would imply to solve an additional convex quadratic program. See for instance [46]. For general inequalities, various techniques (including heristics) can be used to determine an initial feasible point (a good starting on this topic is also [46]). The good news is that the predictor, though infeasible, is supposed to be close to the feasible region under some mild smoothness assumptions and provided that we are not stepping *too far* along the followed trajectory on the optimal manifold. In [92, 27, 29, 31] the alternative (for BOPs) has been to solve the reduced KKT system of equations based on the current active set. This technique, however, requires an a-priori knowledge of the active constraints at the corrector. The latter represents no challenge while following a single piece of the solution manifold where the active set is suppose to remain unchanged. The problem arises when a turning point is detected since all possible combinations of the potential active constraints should be considered in order to identify the next segment to follow.[3] Then, we prefer to leave these issues to be subject to a detailed analysis in

[2] Recall that we are dealing with a compound variable $(\nu, \delta) \in \mathbb{R}^{n+1}$ so being strict $(I_{l,u}\ 0)$ will be considered by the solver instead of just $I_{l,u}$.

[3] The technique actually includes (i) to determine the number s of constraints that are supposed to be active, (ii) to identify the set $\mathcal{Z}$ of all potential active constraints, and (iii) to test all possible s-combinations of constraints out of $\mathcal{Z}$ to identify the new active set.

the future and focus here in box constrained problems.

The pseudo code of the Pareto Tracer for box constrained MOPs is thus very close to Algorithm 3. We only have to change the lines 8 and 9 to use respectively the formula (5.31) to compute the desired orientation μ_d^+ and the formula (5.30) to compute the direction ν_{μ_d}. Additionally, if the tangent vector results to be zero, we have to repeat the process with the original formulas (3.18) and (3.26), respectively. Then, the resulting unconstrained predictor $\tilde{p}$ would have to be projected onto the box by Equation (5.22) in order to obtain a feasible point p. The lines 1 and 13 should also be changed to make a call to the new version of the Newton method capable to work with box restrictions. No further modification of the data structure used to keep track of the covered section of the Pareto front is required so the 'recovery' technique can be used as it is described in Algorithm 4. Finally, we can also apply the computational optimizations suggested in Chapter 3 for BOPs. Then, Algorithm 5 will remain identical except for Line 6 that should use Equation (5.21) to compute the tangent vector, and Line 14 that should make a call to the appropriate version of the Newton method.

5.4 Hessian-free Realizations

We will cover in this section the Hessian-free realizations of the current version of the PT. To be consistent with the work developed before, we will present in the following two approaches: one based on SD methods, and the other one based on QN methods. Considering first the general inequality case, we would obtain for predictor directions

$$\begin{pmatrix} \nu_\mu \\ \varsigma \end{pmatrix} = -\begin{pmatrix} I & G_{\mathcal{I}}^T \\ G_{\mathcal{I}} & 0 \end{pmatrix}^{-1} \begin{pmatrix} J^T \\ 0 \end{pmatrix} \mu, \tag{5.43}$$

if the second-order information is completely omitted from Equation (5.12), i.e., as in SD methods the Hessians are approximated by the identity matrix. The orientation vector will be computed as in Equation (5.27) where

$$\mu_d^+ = A^+ d, \tag{5.44}$$

and A is approximated by

$$A = -\left(J \;\; 0\right) \begin{pmatrix} I & G_{\mathcal{I}}^T \\ G_{\mathcal{I}} & 0 \end{pmatrix}^{-1} \begin{pmatrix} J^T \\ 0 \end{pmatrix}. \tag{5.45}$$

If we restrict the problem to accept only box constraints, the formulas for direction and orientation would be even more simplified. We will compute ν_μ by

$$\begin{aligned} \nu_\mu^{\mathcal{I}_c} &= -\left(W_\alpha^{\mathcal{I}_c}\right)^{-1} J_{\mathcal{I}_c}^T \mu = -\sum_{j=1}^{k} \alpha_j I_{n-r} J_{\mathcal{I}_c}^T \mu = -J_{\mathcal{I}_c}^T \mu, \\ \nu_\mu^{\mathcal{I}_{l,u}} &= 0, \end{aligned} \tag{5.46}$$

where $I_{n-r} \in \mathbb{R}^{(n-r)\times(n-r)}$ is the identity matrix and $\nu_\mu^{\mathcal{I}_{l,u}}$ stands for the components of ν_μ corresponding to the active constraints. Then, given d, we compute first

$$\mu_d^+ = -(J_{\mathcal{I}_c} J_{\mathcal{I}_c}^T)^+ d, \tag{5.47}$$

as done in (5.31), which leads to

$$\nu_{\mu_d}^{\mathcal{I}_c} = J_{\mathcal{I}_c}^T (J_{\mathcal{I}_c} J_{\mathcal{I}_c}^T)^+ d = J_{\mathcal{I}_c}^+ d. \tag{5.48}$$

The corrector, on the other hand, gets reduced to

$$\begin{array}{cll} \displaystyle\min_{(\nu,\delta)\in\mathbb{R}^n\times\mathbb{R}} & \frac{1}{2}\|\nu\|^2 + \delta & \\ \text{s.t.} & \nabla f_j(x)^T\nu \leq \delta, & j = 1,\ldots,k, \\ & g_j(x) + \nabla g_j(x)^T\nu = 0, & j \in \mathcal{I}, \end{array} \tag{5.49}$$

which is transformed into

$$\begin{array}{cll} \displaystyle\min_{(\nu,\delta)\in\mathbb{R}^n\times\mathbb{R}} & \frac{1}{2}\|\nu\|^2 + \delta & \\ \text{s.t.} & \nabla f_j(x)^T\nu \leq \delta, & j = 1,\ldots,k, \\ & I_{l,u}\nu + I_{l,u}(x + l + u) \leq 0, & \end{array} \tag{5.50}$$

for box constrained MOPs.

A more promising choice, after the results of the previous two chapters, is to utilize a QN approach. Here, the formulas for ν_μ and μ given for MOPs with general inequalities as well as for MOPs subject to box restrictions remain as they were originally specified, except for the term W_α that is replaced by

$$B_\alpha = \sum_{j=1}^{k} \alpha_j B_j \in \mathbb{R}^{n\times n}. \tag{5.51}$$

See (5.12) and (5.27) for the equations corresponding to direction and orientation in problems with inequality restrictions. The equivalent formulas for the box constrained case are given by (5.30) and (5.31), respectively. Again, no approximation of the Hessians of the inequalities is required since they are approximated by linear functions or they are actually linear (box constraints). Then, only the Hessians of the objectives will be updated (by means of BFGS updates (3.68)) at each iteration of the corrector method. For this, the Newton direction subproblem (see the subproblems (5.34) and (5.38) for inequality and box contrained MOPs, respectively) remains unchanged except for the term $\nabla^2 f_j(x)$ that is replaced by its approximation B_j for all $j = 1,\ldots,k$. The Hessians corresponding to predictor points are also updated based on the information gathered from the previous corrector, which improves (as expected) the performance of the method. We are still missing a proper analysis of the convergence rates of these procedures, but the numerical results presented in the next section point toward a superiority of the QN approach over the SD version. Additionally, a study of the asymptotic time complexity of each realization is highly recommended in order to provide a balance between convergence speed and computational cost.

5.5 Numerical Results

We will compare in the following the three versions of the PT for problems with box (and possibly equality) constraints. However, we will not include this time a comparison with the method of Hillermeier since it was not originally designed to work with inequalities. Also, we follow the routine of setting the parameters with the values specified at their first appearance and choose the same initial Pareto point for each of the algorithms.

To complete our tests, we will present immediately three constrained problems. The first two with two objectives and the last one with three objectives. In particular, the second example shows the performance of the PT on a problem with mixed box and inequality restrictions. As usual, a table will record the number of solutions encountered by each algorithm as well as the average number of corrector and backtrack iterations and the total amount of function, Jacobian and Hessian evaluations required. Despite we provide no comparison with any further method, the material presented here helps to understand the differences that arise from the alternative we choose to manage second-order information. It will be clearly illustrated that the results (in that sense) remain consistent with what was observed in the previous two chapters: the more accurate the Hessian information we have, the faster the convergence toward the Pareto set. We resolved (as in Chapter 4) that the number of function and Jacobian evaluations of the constraints provides no additional insights on the behavior of the PT since they are close to the amount of function and Jacobian evaluations of the objectives. Finally, we will continue to show the numerical results of the QN approach in a separate figure.

5.5.1 Example 1

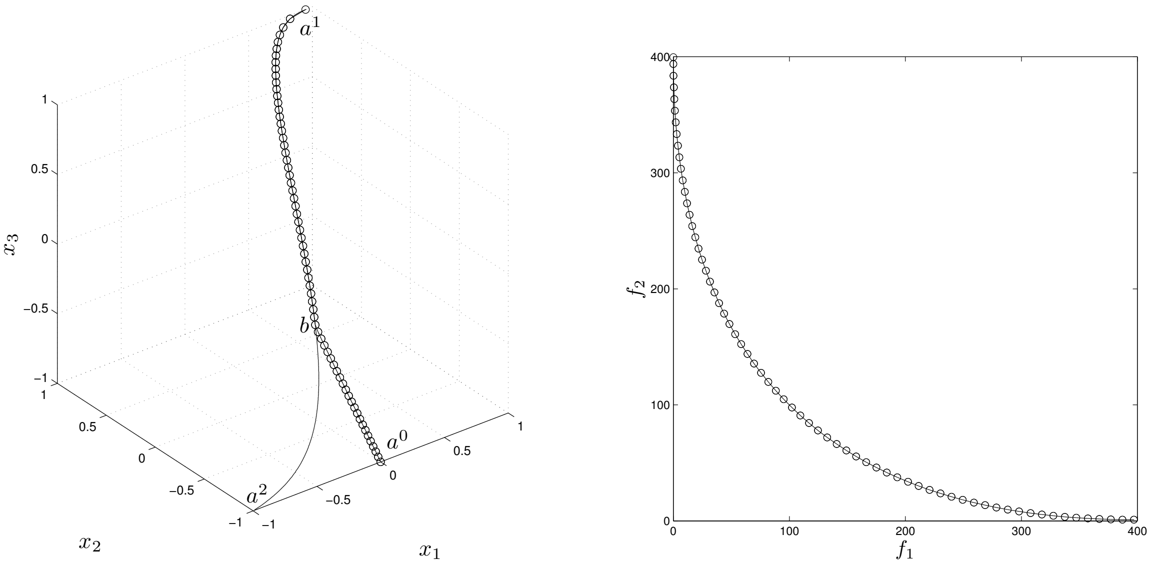
The first example is a BOP subject to one box inequality given by

$$\begin{aligned} f_1(x) &= (x_1 - a_1^1)^4 + \sum_{i=2}^{100} (x_i - a_i^1)^2, \quad \text{and} \\ f_2(x) &= \sum_{i=1}^{100} (x_i - a_i^2)^2, \\ \text{s.t.} \quad & x_1 \geq 0, \end{aligned} \tag{5.52}$$

where $a^1 = (1, \ldots, 1)^T \in \mathbb{R}^{100}$ and $a^2 = -a^1$. The solution set of the BOP (5.52) after removing the constraint is a nonlinear curve connecting the points a^1 and a^2 that gets projected onto the subspace defined by $x_1 = 0$ if we add the box restriction above. Call b the solution of the unconstrained problem where the first component is set to zero ($b_1 = 0$). Then, the Pareto set of the BOP (5.52) comes in two pieces: one linear segment from $a^0 = (0, -1, \ldots, -1) \in \mathbb{R}^{100}$ to b, and one curved segment from b to a^1. For a better understanding, Figure 5.2 shows the Pareto set/front obtained by the PT-QN using a step length of $\tau = 10$ in objective space. Here, the three versions of the PT performed no corrector step while following the fragment of the solution curve from a^0 to b. Table 5.1 records the respective computational efforts required by each algorithm. Undoubtedly, the differences observed between the methods arise from the continuation on the nonlinear segment that connects the points b and a^1. Specifically, the PT-N is the one that requires the least number of correctors at the cost of performing in this case 198 evaluations of the Hessians. The PT-QN is in second place, but considering the savings in Hessian evaluations and the small differences in function and Jacobian evaluations with respect to the PT-N, it can be considered the overall winner. The PT-SD is the one that requires more correctors, but on the other hand, its implementation does not have to deal with matrices of size $\mathcal{O}(n^2)$.

	PT-N	PT-QN	PT-SD
Solutions	66.0000	66.0000	66.0000
Avg. corrector iterations	0.5077	0.6923	1.6308
Avg. backtrack iterations	0.0000	0.0410	4.2153
Function evaluations	99.0000	116.0000	657.0000
Jacobian evaluations	99.0000	111.0000	172.0000
Hessian evaluations	198.0000	-	-

Table 5.1: Computational efforts of the three PT variants on the BOP (5.52).



(a) decision space

(b) objective space

Figure 5.2: Numerical results of the PT-QN method on the BOP (5.52).

5.5.2 Example 2

Next we consider a BOP of five variables subject to box restrictions and two equality constraints. This is a modification of the test problem presented in [6] that reads as

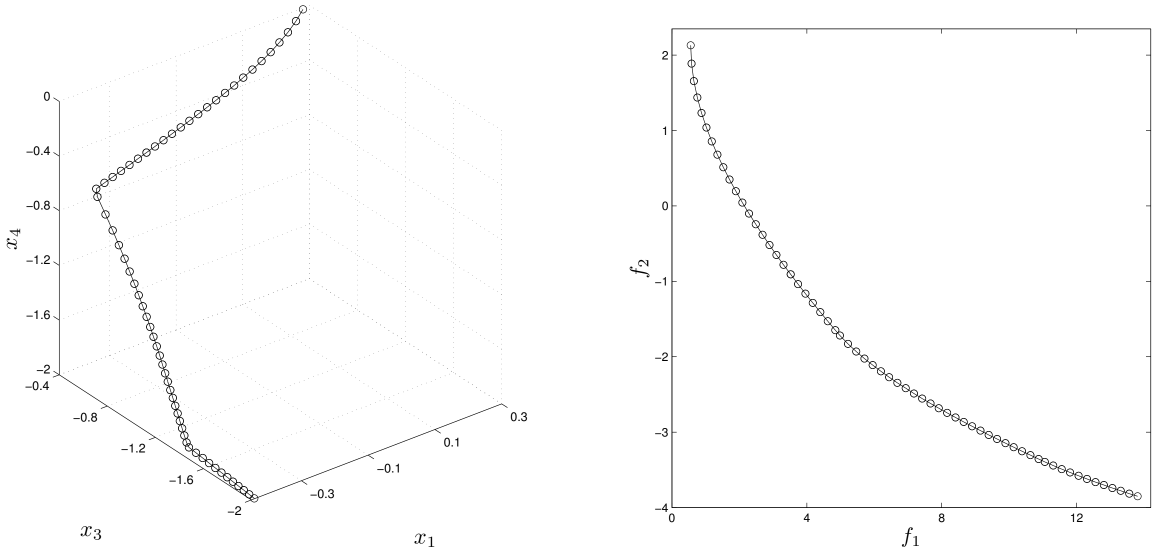
$$
\begin{aligned}
f_1(x) &= x_1^2 + x_2^2 + x_3^2 + x_4^2 + x_5^2, \quad \text{and} \\
f_2(x) &= 3x_1 + 2x_2 - \frac{x_3}{3} + 0.01(x_4 - x_5)^3, \\
\text{s.t.} \qquad & x_1 + 2x_2 - x_3 - 0.5x_4 + x_5 = 2, \\
& 4x_1 - 2x_2 + 0.8x_3 + 0.6x_4 + 0.5x_5^2 = 0, \\
& -2 \leq x_i \leq 2, \quad i = 1, \ldots, 5.
\end{aligned}
\tag{5.53}
$$

Figure 5.3 shows the numerical results of the PT-QN for $\tau = 0.25$ and Table 5.2 the computational costs of the three variants of the PT. We can see in the picture that the Pareto set of this problem is as well a piecewise curve induced by the inequality constraints. All versions of the PT were successful in finding the entire solution set. However, a strange event takes place here: the PT-QN converges slightly faster than the PT-N. A reasonable explanation for this particular scenario is that the Hessian of the second objective function is indefinite or negative definite in all the domain interval, so we apply a transformation (i.e., a modified Cholesky decomposition) on this matrix to keep it positive definite. Then, it is highly probable[4] that this (*necessary*) modification downgrades the convergence rate of the PT-N. This observation provides a further incentive for the use of QN methods in real-life applications where the Hessians are not necessarily positive definite and will have to be modified anyway.

	PT-N	PT-QN	PT-SD
Solutions	62.0000	62.0000	62.0000
Avg. corrector iterations	1.0323	0.9839	2.3226
Avg. backtrack iterations	0.0000	0.0000	0.2969
Function evaluations	127.0000	124.0000	296.0000
Jacobian evaluations	127.0000	124.0000	207.0000
Hessian evaluations	254.0000	-	-

Table 5.2: Computational efforts of the three PT variants on the BOP (5.53).

[4] We can only establish certainty after the proper convergence rate analysis of the corrector methods.



(a) decision space

(b) objective space

Figure 5.3: Numerical results of the PT-QN method on the BOP (5.53).

5.5.3 Example 3

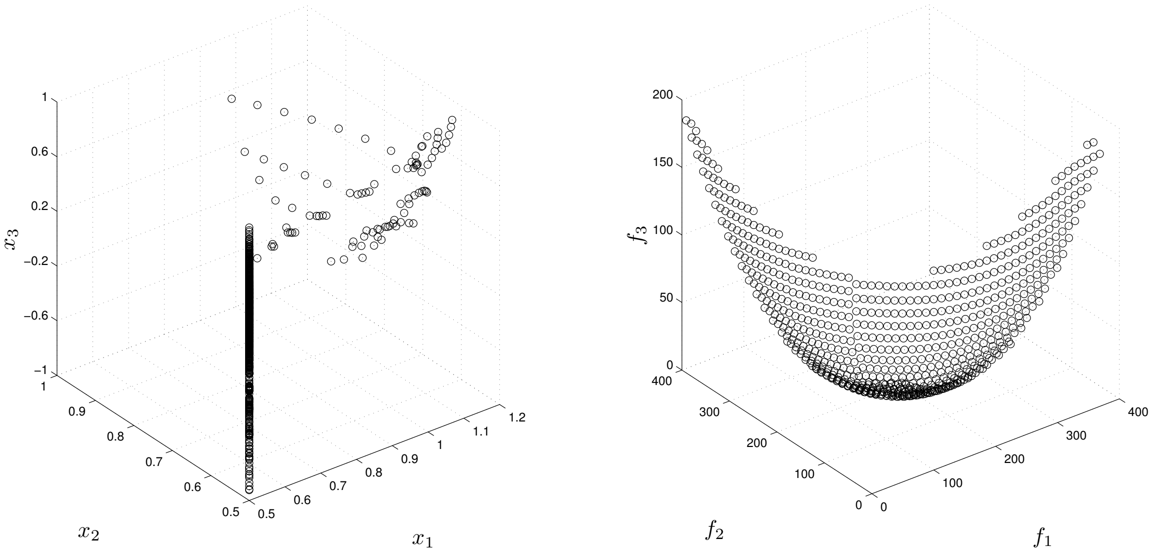
The last example is a three objectives problem defined by

$$\begin{aligned} f_j(x) = &\sum_{\substack{i=1\\ i\neq j}}^{100} (x_i - a_i^j)^2 + (x_j - a_j^j)^4, \qquad j = 1,2,3, \\ \text{s.t.} \quad & x_1 \geq 0.5, \quad \text{and} \\ & x_2 \geq 0.5, \end{aligned} \tag{5.54}$$

where $a^1 = (1, \ldots, 1)^T \in \mathbb{R}^{100}$, $a^2 = -a^1$, and $a^3 = (1, -1, 1, -1, \ldots)^T \in \mathbb{R}^{100}$. The Pareto set and front of the unconstrained version of this problem is presented in Figure 3.9 (last example of Chapter 3). Here, after introducing a lower bound for the first two variables, we obtain a similar Pareto front but a quite different Pareto set. See Figure 5.4 for a graphical illustration of the numerical results yielded by the PT-QN using a step length of $\tau = 10$ (the same was taken in Chapter 3 for the unconstrained case). It is interesting to see in the picture the remarkable differences between the distribution of points in parameter and objective space. Clearly, the PT had to be very precise in selecting the proper direction in decision space to produce such *well placed* results in objective space. Again, the variant that uses exact second derivatives converges faster toward the solution set. In contrast, the SD version is notably slower due to the total omission of the Hessian information. The best balance seems to be the QN approach that is most of the time capable to come close to the Newton behavior without any knowledge of the second-order information.

	PT-N	PT-QN	PT-SD
Solutions	711.0000	726.0000	644.0000
Avg. corrector iterations	1.1635	1.2933	8.1253
Avg. backtrack iterations	0.0000	0.0105	2.2218
Function evaluations	3870.0000	4111.0000	19297.0000
Jacobian evaluations	1907.0000	2081.0000	7502.0000
Hessian evaluations	5721.0000	-	-

Table 5.3: Computational efforts of the three PT variants on the MOP (5.54).



(a) decision space

(b) objective space

Figure 5.4: Numerical results of the PT-QN method on the MOP (5.54).

5.6 Discussion

In this chapter, we end up with a version of the Pareto Tracer that can handle both equality and box restrictions for problems with virtually any number of objectives. Additionally, the theoretical basis for the consideration of general inequality constraints has also been established. The strategies followed here are consistent with the work developed in the last two chapters and are similar in principle to classical continuation techniques. The map $\tilde{F}$ coming from the first-order conditions of optimality plays a key role in the derivation of tangent directions, although we actually use a reduced version of $\tilde{F}$ by taking the active inequalities as equalities and excluding the inactive constraints. Since for $k \geq 3$ there is an infinity of possible vectors within the tangent space to the Pareto set, it is essential to come up with a good *selection* mechanism for such predictor directions. This is precisely the inherent role associated with the *rotation* vector $\mu \in \mathbb{R}^k$ (coming also from $\tilde{F}$) which can be chosen based on a desired movement in objective space by solving an additional system of linear equations of size $\mathcal{O}(k^2)$. For the corrector, we have proposed a further modification of the Newton method [16] which can easily be coupled with the algorithm designed in Chapter 4 for equality constrained MOPs. This is, however, a method of feasible directions and it is still missing an effective strategy to deliver feasible predictors. Then, we have limited our implementations to problems subject to box constraints and utilize gradient projection methods to guarantee the feasibility of the predictors. The development of further strategies for the general case has been left for a future work. Other authors have proposed to use instead a Gauss-Newton or Levenberg-Marquardt method to solve the reduced system $\tilde{F}$ induced by the active set (see for instance [92, 27, 29, 31]). Our aim here, though, has been to exploit the ideas of the Newton method of Fliege et al. which is proven to have better convergence properties.[5] Besides, the alternatives suggested above hardly depend on an elaborated procedure to determine (a-priori) the active constraints at the following corrector. In addition, we developed two Hessian-free realizations of the Pareto Tracer based respectively on quasi-Newton and gradient descent methods, and the choice will depend on how we intend to manage the second-order information of the problem. The Hessians of the constraints are completely ignored since we approximate both equality and inequalities by linear functions. So far we have not been able to determine whether this linear approximations are good enough to provide a competitive rate of convergence for our methods. Thus, a detailed study of this topic as well as an analysis of the computational time complexity of the procedures are recommended as the next steps to follow.

[5] By convergence properties here we do not mean rate of convergence but the capacity of the Newton method to avoid being attracted to e.g. points of maximum.

Chapter 6

Conclusions and Future Work

This chapter summarizes the thesis work, discusses its findings and contributions, and points out its limitations. Then, in a second part, it also outlines directions for future research.

6.1 Conclusions

We have devised, in this thesis work, a novel PC method to trace the manifold of Pareto points of a given MOP. The Pareto Tracer, as we call it, follows the spirit of classical continuation techniques for the treatment of undetermined systems of equations. Numerical path following methods [38] have been utilized before in the context of multi-objective optimization. The most *classical* example is the approach of Hillermeier [28] which can be seen as an application of the Euler-Newton method [38] on the system of equations inferred from the KKT optimality conditions. Our proposal differs from the latter method in two major aspects. First of all, one of the main concerns of the Pareto Tracer focuses on a separation of the decision from the weight space. To be more precise, for every optimal solution, there exists an associated vector of weights (or Lagrange multipliers) such that they satisfy together the first-order conditions of optimality (see Theorem 2.1.2). It has been vastly illustrated along this work (see for instance Figure 3.1) that the additional consideration of the weight space comes with a possible increase of the nonlinearity of the solution set. If, for instance, we are dealing with a linear Pareto set, the corresponding augmented solution manifold does not have to be linear again, most likely because of the difference of magnitudes between the parameters of the problem and the Lagrange multipliers. Thus, considering that predictor points are obtained through linearizations of the solution set, following a (generally nonlinear) trajectory in the compound space (as done by the method of Hillermeier) will unavoidable lead to the computation of correctors even when the Pareto set is a line or a flat surface. As a byproduct of the separation process, we come up with a new set of multipliers that play the role of providing certain degree of freedom to steer the search in parameter space based on a desired movement in objective space. The latter property gives an advantage to the Pareto Tracer over the Hillermeier approach since it has a direct impact on the distribution of solutions in objective space.

The second major difference between the strategy proposed by Hillermeier and ours lies in the corrector phase. Here, the Newton method recently proposed by Fliege et al. in [16] was selected as corrector instead of applying a Gauss-Newton or a Levenberg-Marquardt method [58] to solve the KKT system of equations. The computational complexity of the Newton approach is yet to be determined, though, it can be solved in polynomial time. The prime motivation for our decision has been, however, to investigate other alternatives that are not so hardly dependent on the set of Lagrange multipliers and have, in addition, better theoretical properties. Recall that the corrector proposed by Hillermeier can converge to non-optimal solutions (e.g., one with negative Lagrange multipliers or that maximizes the objectives).

Our contributions also include the design of a Newton-like algorithm for MOPs subject to general constraints which is used as corrector by the Pareto Tracer when dealing with this type of problems. For the sake of simplicity, we have divided the procedure into one method that deals exclusively with equality restrictions (covered in Chapter 4) and another one designed to handle problems subject to inequality constraints (enclosed in Chapter 5). The integration of both algorithms into a single one is straightforward and has been omitted in favor of a greater clarity. The idea of the new approach is quite simple and mainly consists of a modification of the Newton direction subproblem given in [16] to include a linearization of the active constraints. Our proposal is a method of feasible directions, meaning that the initial point and the subsequent iterations must remain feasible (with respect to the inequalities). Thus, one problem that certainly arises in the context of continuation is that a predicted point cannot be guaranteed to be feasible. Given this inconvenient, we have decided to limit this work to the special case of box constrained problems and utilize gradient projections to ensure the feasibility of predictors.[1] A step length control strategy for MOPs with general inequalities is also missing. Basically, it should be one that satisfies the Armijo condition while keeping the iterations feasible (we may e.g. take the one suggested in [18]). However, we have focused so far exclusively on line searches dealing with box restrictions. Furthermore, the Armijo condition has been extended for equality constrained MOPs. Take in mind that there may be cases where there is no descent direction that additionally points toward the feasible region (regarding the equalities). Then, the line search should be designed in accordance to the predicted decrease of both the objectives and a merit function based on the equality restrictions (we take here $-\|h(x)\|^2$). This should suffice those cases where the functions cannot be improved simultaneously in order to reach the feasible region. The estimated reduction of the objectives, on the other hand, is given precisely by the Newton direction subproblem, where the minimum function value (denoted by δ)

[1] The extension to MOPs with linear constraints is straightforward but has been left to be developed in the future together with the general case.

represents such a measure. Several numerical examples have been presented together with theoretical evidence that help to support the convergence of the method toward KKT points. However, a detailed analysis of the convergence rate has been left out of the scope of this thesis. In particular, such analysis may help to understand the problems observed in the last example of Chapter 4 (subject to equality constraints), or even better, could help to improve the new method as well as these results.

We also cope with two Hessian-free realizations of the Pareto Tracer. The most basic one rests on gradient descent methods, i.e., the Hessians are approximated by the identity matrix. This implies that predictor directions would be a linear combination of the derivatives of the problem. This choice is very attractive by the efficiency of its computations, and has been used in [36] and [30] for the treatment of BOPs. Additionally, under the assumption that $\text{rank}(J) = k - 1$, the Jacobian is not orthogonal to the linearized manifold of solutions, so a movement toward the Pareto set is expected. In spite of this fact, the derivatives can be nearly orthogonal to the solution set as graphically illustrated by three examples in Figure 3.4. Thus, we take as predictor directions (at least for the bi-objective case) the secant between the last two points. With this, both direction and orientation come essentially for free while (implicitly) some curvature information is captured. Moreover, for problems with linear Pareto sets, derivatives and secants coincide with the direction of the solution set, so it seems that the use of secants is not worse than the use of derivatives as approximations of tangent vectors. For the corrector, we take the steepest descent method for MOPs proposed by Fliege and Svaiter in [18] and apply the modifications suggested here to deal with constrained problems. In consequence, our steepest descent approach for MOPs restricted by inequalities merely differs from the one in [18] by the inclusion of the terms $g_j(x)$ for $j \in \mathcal{I}$ in the constraints of the search direction subproblem. The computations get thus simplified to the solution of a QP problem in contrast to the QCLP instance utilized by the Newton method. The savings are, though, at the expense of a downgrade of the convergence rate from quadratic to linear (at least for the unconstrained case).

Mainly due to the last statement, our preferred choice for a Hessian-free realization is one that utilizes elements from quasi-Newton methods. A more clever management of the second-order information is achieved here by means of BFGS updates. The corrector phase is then accomplished by the quasi-Newton method for MOPs [17] which is taken as well with the appropriate modifications to handled problems with constraints. For the predictor, in turn, the exact Hessians are replaced in the proper formulations by their respective approximations obtained through the corrector steps. The improvement of the quasi-Newton method over the steepest descent approach is that a superlinear convergence rate can be proven under some mild conditions. On the other hand, the Newton method can guarantee at most quadratic convergence but at the cost of demanding the exact second derivatives of all objective functions. It seems that the quasi-Newton variant of the Pareto Tracer is the best Hessian-free choice we could expect. However, the implementation of the steepest descent version is much more efficient since it does not implicate matrices of size $\mathcal{O}(n^2)$. Another concern about the search direction subproblem is that the matrices involved must be positive definite to guarantee a descent direction as result (including the uniqueness of the solution). For SOPs, the updated Hessian is forced to be positive definite by imposing the Wolfe conditions on the step length control, but for MOPs it seems that these rules may be too hard (not to say impossible) to satisfy for all objectives simultaneously. We then solve the problem by applying a modified Cholesky factorization on each matrix to keep it positive definite. Actually, the same strategy is utilized in the Newton method since most real-life applications cannot ensure the positive definiteness of the Hessians in all the domain. Furthermore, a second role for the BFGS updates is suggested in the context of continuation: an update of the Hessians at predicted points with base on the information gathered from the previous corrector will provide a better initial guess to the subsequent corrector, and will possibly help to preserve the second-order information gained along all the trajectory. The Hessians of the constraints are not included in the updates since they are all approximated by linear functions. It is not clear, though, whether this information is required to provide a competitive rate of convergence.

Several optimizations of the computational costs for the bi-objective case were implemented as well. In particular, we avoid the use of data structures to keep track of the zone of the Pareto set/front that is already covered by the algorithm. This has been possible by performing an orientation adjustment of the continuation process for nonconvex problems. With this, we ensure to follow the optimal curve all along in the direction to minimize one of the objectives, and then, by reversing the orientation, we will go all along in the direction to minimize the other.

In conclusion, we consider the Pareto Tracer a highly competitive algorithm for the treatment of general MOPs that fulfill the smoothness assumptions required by continuation methods. Therefore, we intent to intensify the investigation in this line of research with special emphasis on Hessian-free realizations. A lot of work remains to be done as described in the next section, but we have shown that the use of continuation techniques is an affordable choice in a great variety of academic examples. This provides strong evidence that the Pareto Tracer could be as well a powerful tool for a wide range of real-life applications, particularly for those where the function evaluations are very costly and resource-intensive.

6.2 Future Work

There are several lines of research arising from this work that may be pursued in the future. We recommend to start by a proper extension of the Pareto Tracer to handle MOPs subject to general inequality restrictions. For linear constraints, we would simply have to solve a convex quadratic program to project the predictors onto the feasible polytope. Note that—as in box constrained problems—this projection would rarely be required: only when the step along the tangent passes by a corner of the polytope. Additionally, the (projected) predictor will be in most cases already a KKT point, but unfortunately, this approach will only work if absolutely all constraints are linear. The general (mixed) case would be much more challenging to work out. For this, it looks like an affordable choice to modify the method of feasible directions we propose here to accept infeasible starting points. Of course, this strategy should come with the proper modification of the step length control taking into account that not always all objectives can be improved in order to reach the feasible region. See Chapter 4 for a similar situation in the context of equality constrained MOPs.

A detailed analysis of the convergence rate and the asymptotic time complexity of the different variants of the Pareto Tracer are highly recommended for a better comprehension of the strengths and weaknesses of our proposals. In particular, we would like to improve the computational costs of the quasi-Newton approach over the Newton method. Recall that for SOPs the complexity of the iterations is reduced from $\mathcal{O}(n^3)$ to $\mathcal{O}(n^2)$ when we opt for quasi-Newton alternatives. In addition, such a study should (in our opinion) anticipate a possible extension of these methods for high dimensional problems (i.e., $n > 1000$). This is an almost unexplored area of research in multi-objective optimization that has a lot of practical interest. See for instance [115] and [116] in the context of evolutionary optimization, and [90] in multi-objective continuation. Here, we suggest to investigate the possibility of a *limited memory* version of the Pareto Tracer. As done with the Newton, quasi-Newton, and steepest descent methods, we could go for an adaptation of the *limited memory* BFGS (L-BFGS) [117, 118, 119, 120] for problems with more than one objective. The study of trust-region frameworks as alternatives to the line search strategies we manage here is another topic that could be of certain interest (see [79, 80, 65]).

We would also like to stress that the Pareto Tracer—as all continuation methods—is of local nature. It is thus conceivable to hybridize the algorithm with a global strategy such as specialized MOEAs in order to obtain a fast and reliable procedure. So far we have only tested our method on academic examples, but the true purpose of this research is to apply the new knowledge to real-world applications. We expect that the *hybridized* approach has more probabilities of success in this environment. Regarding this matter, a source of many interesting ideas is the proposal given in [36]. The latter is basically the integration of the Pareto Path Following, a gradient-based continuation method, into a curve-based MOEA. Although restricted to BOPs, this scheme comes with two new metrics to assess the performance of the algorithm: an inter-curve coverage (interested in the discovery of every portion of the optimal curve) and an intra-curve coverage (focused in a satisfactory sampling of each connected portion). Derivative-free realizations come in handy here and now, but this is probably one of the most challenging suggestions we have made for the future work. Some derivative-free quasi-Newton designs for SOPs are proposed in [121, 122, 123] which may serve as a good starting point.

Finally, one of the primary goals of continuation has a lot to do with visualization. Thus, a great choice and a nice complement to the Pareto Tracer would be the integration of an interpolation or triangulation strategy such that it can render a body or surface rather than plotting a set of isolated points. Two approaches related to this topic are proposed in [124] and [125].

Bibliography

[1] Vilfredo Pareto. *Manual of Political Economy.* The MacMillan Press, 1971 (original edition in French in 1927).

[2] Kaisa Miettinen. *Nonlinear Multiobjective Optimization.* International Series in Operations Research & Management Science. Springer, 1999.

[3] Gabriele Eichfelder. *Adaptive Scalarization Methods in Multiobjective Optimization.* Vector Optimization. Springer, 2008.

[4] R. Timothy Marler and Jasbir S. Arora. The weighted sum method for multi-objective optimization: new insights. *Structural and Multidisciplinary Optimization*, 41(6):853–862, 2010.

[5] Jörg Fliege. Gap-free computation of Pareto-points by quadratic scalarizations. *Mathematical Methods of Operations Research*, 59(1):69–89, 2004.

[6] Indraneel Das and John E. Dennis. Normal-Boundary Intersection: A new method for generating the Pareto surface in nonlinear multicriteria optimization problems. *SIAM Journal on Optimization*, 8(3):631–657, 1998.

[7] Achille Messac, Amir Ismail-Yahaya, and Christopher A. Mattson. The Normalized Normal Constraint method for generating the Pareto frontier. *Structural and Multidisciplinary Optimization*, 25(2):86–98, 2003.

[8] Aimin Zhou, Bo-Yang Qu, Hui Li, Shi-Zheng Zhao, Ponnuthurai Nagaratnam Suganthan, and Qingfu Zhang. Multiobjective evolutionary algorithms: A survey of the state of the art. *Swarm and Evolutionary Computation*, 1(1):32–49, 2011.

[9] Carlos A. Coello Coello, Gary B. Lamont, and David A. Van Veldhuizen. *Evolutionary Algorithms for Solving Multi-Objective Problems.* Springer, New York, 2nd edition, 2007.

[10] Kalyanmoy Deb. *Multi-Objective Optimization Using Evolutionary Algorithms.* Wiley Interscience Series in Systems and Optimization. John Wiley & Sons, Chichester, UK, 2001.

[11] Michael Dellnitz, Oliver Schütze, and Thorsten Hestermeyer. Covering Pareto sets by multilevel subdivision techniques. *Journal of Optimization Theory and Applications*, 124(1):113–136, 2005.

[12] Johannes Jahn. Multiobjective search algorithm with subdivision technique. *Computational Optimization and Applications*, 35(2):161–175, 2006.

[13] Carlos Hernández, Yousef Naranjani, Yousef Sardahi, Wei Liang, Oliver Schütze, and Jian-Qiao Sun. Simple cell mapping method for multi-objective optimal feedback control design. *International Journal of Dynamics and Control*, 1(3):231–238, 2013.

[14] Carlos Hernández, Jian-Qiao Sun, and Oliver Schütze. Computing the set of approximate solutions of a multi-objective optimization problem by means of cell mapping techniques. In Michael Emmerich, Andre Deutz, Oliver Schütze, Thomas Bäck, Emilia Tantar, Alexandru-Adrian Tantar, Pierre Del Moral, Pierrick Legrand, Pascal Bouvry, and Carlos A. Coello Coello, editors, *EVOLVE - A Bridge between Probability, Set Oriented Numerics, and Evolutionary Computation IV*, volume 227 of *Advances in Intelligent Systems and Computing*, pages 171–188. Springer International Publishing, 2014.

[15] Jesús Fernández Cruz, Oliver Schütze, Jian-Qiao Sun, and Fu-Rui Xiong. Parallel cell mapping for unconstrained multi-objective optimization problems. 227:133–146, 2014.

[16] Jörg Fliege, L. M. Graña Drummond, and Benar F. Svaiter. Newton's method for multiobjective optimization. *SIAM Journal on Optimization*, 20(2):602–626, 2009.

[17] Žiga Povalej. Quasi-Newton's method for multiobjective optimization. *Journal of Computational and Applied Mathematics*, 255(1):765–777, 2014.

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

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

[20] Oliver Schütze, Adriana Lara, and Carlos A. Coello Coello. The Directed Search method for unconstrained multi-objective optimization problems. In *EVOLVE - A Bridge between Probability, Set Oriented Numerics, and Evolutionary Computation*. Springer, 2012.

[21] Maria C. Recchioni. A path following method for box-constrained multiobjective optimization with applications to goal programming problems. *Mathematical Methods of Operations Research*, 58(1):69–85, 2003.

[22] Adriana Lara, Gustavo Sánchez, Carlos A. Coello Coello, and Oliver Schütze. HCS: A new local search strategy for memetic multiobjective evolutionary algorithms. *IEEE Transactions on Evolutionary Computation*, 14(1):112–132, 2010.

[23] Adriana Lara, Sergio Alvarado, Shaul Salomon, Gideon Avigad, Carlos A. Coello Coello, and Oliver Schütze. The gradient free Directed Search method as local search within multi-objective evolutionary algorithms. In Oliver Schütze, Carlos A. Coello Coello, Alexandru-Adrian Tantar, Emilia Tantar, Pascal Bouvry, Pierre Del Moral, and Pierrick Legrand, editors, *EVOLVE - A Bridge between Probability, Set Oriented Numerics, and Evolutionary Computation II*, volume 175 of *Advances in Intelligent Systems and Computing*, pages 153–168. Springer Berlin Heidelberg, 2013.

[24] Roman Denysiuk, Lino Costa, and Isabel Espírito Santo. A new hybrid evolutionary multiobjective algorithm guided by descent directions. *Journal of Mathematical Modelling and Algorithms in Operations Research*, 12(3):233–251, 2013.

[25] Roman Denysiuk, Lino Costa, and Isabel Espírito Santo. Generalized multiobjective evolutionary algorithm guided by descent directions. *Journal of Mathematical Modelling and Algorithms in Operations Research*, pages 1–17, 2014.

[26] Joshua Knowles and David Corne. Memetic algorithms for multiobjective optimization: Issues, methods and prospects. In William E. Hart, J.E. Smith, and N. Krasnogor, editors, *Recent Advances in Memetic Algorithms*, volume 166 of *Studies in Fuzziness and Soft Computing*, pages 313–352. Springer Berlin Heidelberg, 2005.

[27] Joanna Rakowska, Raphael T. Haftka, and Layne T. Watson. Multi-objective control-structure optimization via homotopy methods. *SIAM Journal on Optimization*, 3(3):654–667, 1993.

[28] Claus Hillermeier. *Nonlinear Multiobjective Optimization - A Generalized Homotopy Approach*. Birkhäuser, 2001.

[29] Benjamin Martin, Alexandre Goldsztejn, Laurent Granvilliers, and Christophe Jermann. On continuation methods for non-linear bi-objective optimization: towards a certified interval-based approach. *Journal of Global Optimization*, pages 1–14, 2014.

[30] Honggang Wang. Zigzag search for continuous multiobjective optimization. *INFORMS Journal on Computing*, 25(4):654–665, 2013.

[31] Andreas Potschka, Filip Logist, Jan F. Van Impe, and Hans Georg Bock. Tracing the Pareto frontier in bi-objective optimization problems by ODE techniques. *Numerical Algorithms*, 57(2):217–233, 2011.

[32] Víctor Pereyra, Michael Saunders, and José Castillo. Equispaced Pareto front construction for constrained bi-objective optimization. *Mathematical and Computer Modelling*, 57(9–10):2122–2131, 2013.

[33] Oliver Schütze. *Set Oriented Methods for Global Optimization*. PhD thesis, University of Paderborn, 2004. http://ubdata.uni-paderborn.de/ediss/17/2004/schuetze/.

[34] Oliver Schütze, Alessandro Dell'Aere, and Michael Dellnitz. On continuation methods for the numerical treatment of multi-objective optimization problems. In Jürgen Branke, Kalyanmoy Deb, Kaisa Miettinen, and Ralph E. Steuer, editors, *Practical Approaches to Multi-Objective Optimization*, number 04461 in Dagstuhl Seminar Proceedings, Dagstuhl, Germany, 2005. Internationales Begegnungs- und Forschungszentrum für Informatik (IBFI), Schloss Dagstuhl, Germany.

[35] Oliver Schütze, Carlos A. Coello Coello, Sanaz Mostaghim, El-Ghazali Talbi, and Michael Dellnitz. Hybridizing evolutionary strategies with continuation methods for solving multi-objective problems. *Engineering Optimization*, 40(5):383–402, 2008.

[36] Ken Harada, Jun Sakuma, Shigenobu Kobayashi, and Isao Ono. Uniform sampling of local Pareto-optimal solution curves by Pareto Path Following and its applications in multi-objective GA. In *Proceedings of the 9th Annual Conference on Genetic and Evolutionary Computation*, GECCO '07, pages 813–820, New York, NY, USA, 2007. ACM.

[37] Eugene L. Allgower and Kurt Georg. *Numerical Continuation Methods*. Springer, 1990.

[38] Eugene L. Allgower and Kurt Georg. *Numerical Path Following*. 1994.

[39] Adanay Martín and Oliver Schütze. A new predictor corrector variant for unconstrained bi-objective optimization problems. In Alexandru-Adrian Tantar, Emilia Tantar, Jian-Qiao Sun, Wei Zhang, Qian Ding, Oliver Schütze, Michael Emmerich, Pierrick Legrand, Pierre Del Moral, and Carlos A. Coello Coello, editors, *EVOLVE - A Bridge between Probability, Set Oriented Numerics and Evolutionary Computation V*, pages 165–179. Springer, 2014.

[40] William Karush. Minima of functions of several variables with inequalities as side conditions. Master's thesis, Department of Mathematics, University of Chicago, 1939.

[41] Harold W. Kuhn and Albert W. Tucker. Nonlinear programming. In *Second Berkeley Symposium on Mathematical Statistics and Probability*, pages 481–492, Berkeley and Los Angeles, 1951. University of California Press.

[42] Sima Noghanian, Abas Sabouni, Travis Desell, and Ali Ashtari. Global optimization: Differential evolution, genetic algorithms, particle swarm, and hybrid methods. In *Microwave Tomography*, pages 39–61. Springer New York, 2014.

[43] Carlos Segura, Carlos A. Coello Coello, Gara Miranda, and Coromoto León. Using multi-objective evolutionary algorithms for single-objective optimization. *4OR*, 11(3):201–228, 2013.

[44] Carlos A. Coello Coello. Theoretical and numerical constraint-handling techniques used with evolutionary algorithms: a survey of the state of the art. *Computer Methods in Applied Mechanics and Engineering*, 191(11–12):1245–1287, 2002.

[45] John A. Nelder and R. Mead. A simplex method for function minimization. *The Computer Journal*, 7(4):308–313, 1965.

[46] Jorge Nocedal and Stephen Wright. *Numerical Optimization*. Springer Series in Operations Research and Financial Engineering. Springer, 2006.

[47] John E. Dennis and Jorge J. Moré. Quasi-Newton methods, motivation and theory. *SIAM Rev.*, 19(1):46–89, 1977.

[48] George B. Dantzig. *Linear Programming and Extensions*. Landmarks in Physics and Mathematics. Princeton University Press, 1998.

[49] Narendra Karmarkar. A new polynomial-time algorithm for linear programming. In *STOC '84 Proceedings of the sixteenth annual ACM symposium on Theory of computing*, pages 302–311, 1984.

[50] Yurii Nesterov and Arkadii Nemirovskii. *Interior-Point Polynomial Algorithms in Convex Programming*. Studies in Applied and Numerical Mathematics. SIAM, Philadelphia, 1994.

[51] Kurt M. Anstreicher. Semidefinite programming versus the reformulation-linearization technique for nonconvex quadratically constrained quadratic programming. *Journal of Global Optimization*, 43(1–2):471–484, 2009.

[52] Sunyonga Kima and Masakazu Kojimab. Second order cone programming relaxation of nonconvex quadratic optimization problems. *Optimization Methods and Software*, 15(3–4):201–224, 2001.

[53] G. Zoutendijk. *Integer and Nonlinear Programming*, chapter Nonlinear Programming, Computational Methods, pages 37–86. North-Holland, Amsterdam, 1970.

[54] William W. Hager and Hongchao Zhang. A survey of nonlinear conjugate gradient methods. *Pacific Journal of Optimization*, 2(1):35–58, 2006.

[55] Philip E. Gill, Gene H. Golub, Walter Murray, and Michael A. Saunders. Methods for modifying matrix factorizations. *Mathematics of Computation*, 28:505–535, 1974.

[56] Philip E. Gill, Walter Murray, and Margaret H. Wright. *Practical Optimization*. Academic Press, London, 1981.

[57] Philip E. Gill and Walter Murray. Quasi-Newton methods for unconstrained optimization. *IMA Journal of Applied Mathematics*, 9(1):91–108, 1972.

[58] Ake Björck. *Numerical Methods for Least Squares Problems*. SIAM, Philadelphia, 1996.

[59] Benedetta Morini and Margherita Porcelli. TRESNEI, a Matlab trust-region solver for systems of nonlinear equalities and inequalities. *Computational Optimization and Applications*, 51(1):27–49, 2012.

[60] Farid Alizadeh and Donald Goldfarb. Second-order cone programming. *Mathematical Programming*, 95(1):3–51, 2003.

[61] Miguel Sousa Lobo, Lieven Vandenberghe, Stephen Boyd, and Hervé Lebret. Applications of second-order cone programming. *Linear Algebra and its Applications*, 284(1–3):193–228, 1998.

[62] Zheng-Hai Huang, Defeng Sun, and Gongyun Zhao. A smoothing Newton-type algorithm of stronger convergence for the quadratically constrained convex quadratic programming. *Computational Optimization and Applications*, 35(2):199–237, 2006.

[63] Eckart Zitzler, Lothar Thiele, Marco Laumanns, Carlos M. Fonseca, and Viviane Grunert da Fonseca. Performance assessment of multiobjective optimizers: An analysis and review. *IEEE Transactions on Evolutionary Computation*, 7(2):117–132, 2003.

[64] Ferrante Neri, Carlos Cotta, and Pablo Moscato. *Handbook of Memetic Algorithms*. Studies in Computational Intelligence. Springer, 2011.

[65] Jonq-Hyun Ryu and Sujin Kim. A derivative-free trust-region method for biobjective optimization. *SIAM Journal on Optimization*, 24(1):334–362, 2014.

[66] Saúl Martínez Zapotecas and Carlos A. Coello Coello. A memetic algorithm with non gradient-based local search assisted by a meta-model. In Robert Schaefer, Carlos Cotta, Joanna Kołodziej, and Günter Rudolph, editors, *Parallel Problem Solving from Nature, PPSN XI*, volume 6238 of *Lecture Notes in Computer Science*, pages 576–585. Springer Berlin Heidelberg, 2010.

[67] Lofti Zadeh. Optimality and non-scalar-valued performance criteria. *IEEE Transactions on Automatic Control*, 8(1):59–60, 1963.

[68] Yacov Y. Haimes, Leon S. Lasdon, and David A. Wismer. On a bicriterion formulation of the problems of integrated system identification and system optimization. *IEEE Transactions on Systems, Man and Cybernetics*, SMC-1(3):296–297, 1971.

[69] George Mavrotas. Effective implementation of the ϵ-constraint method in multi-objective mathematical programming problems. *Applied Mathematics and Computation*, 213(2):455–465, 2009.

[70] Juhani Koski. Defectiveness of weighting method in multicriterion optimization of structures. *Communications in Applied Numerical Methods*, 1(6):333–337, 1985.

[71] Timothy W. Athan and Panos Y. Papalambros. A note on weighted criteria methods for compromise solutions in multi-objective optimization. *Engineering Optimization*, 27(2):155–176, 1996.

[72] Achille Messac. Physical Programming: Effective optimization for computational design. *AIAA Journal*, 34(1):149–158, 1996.

[73] Achille Messac and Christopher A. Mattson. Generating well-distributed sets of Pareto points for engineering design using Physical Programming. *Optimization and Engineering*, 3(4):431–450, 2002.

[74] Kai-Tai Fang, Dennis K.J. Lin, Peter Winker, and Yong Zhang. Uniform design: Theory and application. *Technometrics*, 42(3):237–248, 2000.

[75] Kai-Tai Fang and Ling-Yau Chan. Uniform design and its industrial applications. In Hoang Pham, editor, *Springer Handbook of Engineering Statistics*, pages 229–247. Springer London, 2006.

[76] Achille Messac and Christopher A. Mattson. Normal Constraint method with guarantee of even representation of complete Pareto frontier. *AIAA Journal*, 42(10):2101–2111, 2004.

[77] Sergei V. Utyuzhnikov, P. Fantini, and Marin D. Guenov. A method for generating a well-distributed Pareto set in nonlinear multiobjective optimization. *Journal of Computational and Applied Mathematics*, 223(2):820–841, 2009.

[78] Tohid Erfani and Sergei V. Utyuzhnikov. Directed Search Domain: a method for even generation of the Pareto frontier in multiobjective optimization. *Engineering Optimization*, 43(5):467–484, 2011.

[79] Kely D.V. Villacorta, Paulo R. Oliveira, and Antoine Soubeyran. A trust-region method for unconstrained multiobjective problems with applications in satisficing processes. *Journal of Optimization Theory and Applications*, 160(3):865–889, 2014.

[80] Shaojian Qu, Mark Goh, and Bing Liang. Trust region methods for solving multiobjective optimisation. *Optimization Methods and Software*, 28(4):796–811, 2013.

[81] Enrico Miglierina, Elena Molho, and Maria C. Recchioni. Box-constrained multi-objective optimization: A gradient-like method without "a priori" scalarization. *European Journal of Operational Research*, 188(3):662–682, 2008.

[82] John C. Butcher. Numerical methods for ordinary differential equations in the 20th century. *Journal of Computational and Applied Mathematics*, 125(1–2):1–29, 2000.

[83] Bernt Oksendal. Stochastic differential equations. In *Stochastic Differential Equations*, Universitext, pages 65–84. Springer Berlin Heidelberg, 2003.

[84] Xuerong Mao. *Stochastic Differential Equations and Applications*. Elsevier, 2nd edition, 2007.

[85] H. Schwetlick. *Numerische Lösung nichtlinearer Gleichungen*. Oldenbourg Verlag, 1979.

[86] Gene H. Golub and Charles F. Van Loan. *Matrix Computations*. The Johns Hopkins University Press, Baltimore, 4th edition, 2013.

[87] Alberto Lovison. Singular continuation: Generating piecewise linear approximations to Pareto sets via global analysis. *SIAM Journal on Optimization*, 21(2):463–490, 2011.

[88] Alberto Lovison. Global search perspectives for multiobjective optimization. *Journal of Global Optimization*, 57(2):385–398, 2013.

[89] E. Taylor Whittaker and George N. Watson. *A Course in Modern Analysis*, chapter Forms of the Remainder in Taylor's Series, pages 95–96. Cambridge University Press, 4th edition, 1990.

[90] Maik Ringkamp, Sina Ober-Blöbaum, Michael Dellnitz, and Oliver Schütze. Handling high dimensional problems with multi-objective continuation methods via successive approximation of the tangent space. *Engineering Optimization*, 44(6):1117–1146, 2012.

[91] Benjamin Martin, Alexandre Goldsztejn, Laurent Granvilliers, and Christophe Jermann. Certified parallelotope continuation for one-manifolds. *SIAM Journal on Numerical Analysis*, 51(6):3373–3401, 2013.

[92] Joanna Rakowska, Raphael T. Haftka, and Layne T. Watson. An active set algorithm for tracing parametrized optima. *Structural optimization*, 3(1):29–44, 1991.

[93] Bruce N. Lundberg and Aubrey B. Poore. Numerical continuation and singularity detection methods for parametric nonlinear programming. *SIAM Journal on Optimization*, 3(1):134–154, 1993.

[94] Víctor Pereyra. Fast computation of equispaced Pareto manifolds and Pareto fronts for multiobjective optimization problems. *Mathematics and Computers in Simulation*, 79(6):1935–1947, 2009.

[95] Ken Harada, Jun Sakuma, and Shigenobu Kobayashi. Local search for multiobjective function optimization: Pareto Descent method. In *Proceedings of the 8th Annual Conference on Genetic and Evolutionary Computation*, GECCO '06, pages 659–666, New York, NY, USA, 2006. ACM.

[96] Ken Harada, Jun Sakuma, Isao Ono, and Shigenobu Kobayashi. Constraint-handling method for multi-objective function optimization: Pareto Descent Repair Operator. In Shigeru Obayashi, Kalyanmoy Deb, Carlo Poloni, Tomoyuki Hiroyasu, and Tadahiko Murata, editors, *Evolutionary Multi-Criterion Optimization*, volume 4403 of *Lecture Notes in Computer Science*, pages 156–170. Springer Berlin Heidelberg, 2007.

[97] Hassler Whitney. On the abstract properties of linear dependence. *American Journal of Mathematics*, 57(3):509–533, 1935.

[98] Jos F. Sturm. Using SeDuMi 1.02, a Matlab toolbox for optimization over symmetric cones. *Optimization Methods and Software*, 11(1–4):625–653, 1999.

[99] Jos F. Sturm. Implementation of interior point methods for mixed semidefinite and second order cone optimization problems. *Optimization Methods and Software*, 17(6):1105–1154, 2002.

[100] Kim-Chuan Toh, Michael J. Todd, and Reha H. Tütüncü. SDPT3 a Matlab software package for semidefinite programming, version 1.3. *Optimization Methods and Software*, 11(1-4):545–581, 1999.

[101] Reha H. Tütüncü, Kim-Chuan Toh, and Michael J. Todd. Solving semidefinite-quadratic-linear programs using SDPT3. *Mathematical Programming*, 95(2):189–217, 2003.

[102] Gurobi optimization. `http://www.gurobi.com/`, 2014.

[103] Mosek. `http://mosek.com/`, 2014.

[104] Michael Grant and Stephen Boyd. CVX: Matlab software for disciplined convex programming, version 2.1. `http://cvxr.com/cvx`, 2014.

[105] Michael Grant and Stephen Boyd. Graph implementations for nonsmooth convex programs. In V. Blondel, S. Boyd, and H. Kimura, editors, *Recent Advances in Learning and Control*, Lecture Notes in Control and Information Sciences, pages 95–110. Springer-Verlag Limited, 2008.

[106] Eugene L. Allgower and Phillip H. Schmidt. An algorithm for piecewise-linear approximation of an implicitly defined manifold. *SIAM Journal on Numerical Analysis*, 22(2):322–346, 1985.

[107] Werner C. Rheinboldt. On the computation of multi-dimensional solution manifolds of parametrized equations. *Numerische Mathematik*, 53(1-2):165–181, 1988.

[108] Michael E. Henderson. Multiple parameter continuation: Computing implicitly defined k-manifolds. *International Journal of Bifurcation and Chaos*, 12(03):451–476, 2002.

[109] Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein. *Introduction to Algorithms*. The MIT Press, 3rd edition, 2009.

[110] J. Bräuninger. A Quasi-Newton method with Cholesky factorization. *Computing*, 25(2):155–162, 1980.

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

[112] Hui Li and Qingfu Zhang. Multiobjective optimization problems with complicated Pareto sets, MOEA/D and NSGA-II. *IEEE Transactions on Evolutionary Computation*, 13(2):284–302, 2009.

[113] Eckart Zitzler, Kalyanmoy Deb, and Lothar Thiele. Comparison of multiobjective evolutionary algorithms: Empirical results. *Evolutionary Computation*, 8(2):173–195, 2000.

[114] Kalyanmoy Deb, Lothar Thiele, Marco Laumanns, and Eckart Zitzler. Scalable multi-objective optimization test problems. In *Congress on Evolutionary Computation (CEC-2002)*, pages 825–830. IEEE Press, 2002.

[115] Juan J. Durillo, Antonio J. Nebro, Carlos A. Coello Coello, Francisco Luna, and Enrique Alba. A comparative study of the effect of parameter scalability in multi-objective metaheuristics. In *Congress on Evolutionary Computation (CEC-2008)*, pages 1893–1900, Hong Kong, 2008. IEEE Service Center.

[116] Juan J. Durillo, Antonio J. Nebro, Carlos A. Coello Coello, José García-Nieto, Francisco Luna, and Enrique Alba. A study of multiobjective metaheuristics when solving parameter scalable problems. *IEEE Transactions on Evolutionary Computation*, 14(4):618–635, 2010.

[117] Dong C. Liu and Jorge Nocedal. On the limited memory BFGS method for large scale optimization. *Mathematical Programming*, 45(1-3):503–528, 1989.

[118] Stephen G. Nash and Jorge Nocedal. A numerical study of the limited memory BFGS method and the truncated-Newton method for large scale optimization. *SIAM Journal on Optimization*, 1(3):358–372, 1991.

[119] Xingfu Zou, Ionel M. Navon, Marsha Berger, Kang H. Phua, Tamar Schlick, and François-Xavier Le Dimet. Numerical experience with limited-memory Quasi-Newton and truncated Newton methods. *SIAM Journal on Optimization*, 3(3):582–608, 1993.

[120] Richard H. Byrd, Jorge Nocedal, and Robert B. Schnabel. Representations of Quasi-Newton matrices and their use in limited memory methods. *Mathematical Programming*, 63(1-3):129–156, 1994.

[121] John Greenstadt. A Quasi-Newton method with no derivatives. *Mathematics of Computation*, 26(117):145–166, 1972.

[122] John Greenstadt. Revision of a derivative-free Quasi-Newton method. *Mathematics of Computation*, 32(141):201–221, 1978.

[123] Donghui Li, Liqun Qi, and Vera Roshchina. A new class of Quasi-Newton updating formulas. *Optimization Methods and Software*, 23(2):237–249, 2008.

[124] Rudolph Günter, Heike Trautmann, Soumyadip Sengupta, and Oliver Schütze. Evenly spaced Pareto front approximations for tricriteria problems based on triangulation. In Robin C. Purshouse, Peter J. Fleming, Carlos M. Fonseca, Salvatore Greco, and Jane Shaw, editors, *Evolutionary Multi-Criterion Optimization*, volume 7811 of *Lecture Notes in Computer Science*, pages 443–458. Springer Berlin Heidelberg, 2013.

[125] Markus Hartikainen, Kaisa Miettinen, and Margaret M. Wiecek. PAINT: Pareto front interpolation for nonlinear multiobjective optimization. *Computational Optimization and Applications*, 52(3):845–867, 2012.