ROM-Based Multiobjective Optimization with PDE Constraints

Doctoral thesis
for the award of the academic degree
Doctor of Natural Sciences
(Dr.rer.nat.)

at the

Universität
Konstanz

Faculty of Sciences
Department of Mathematics and Statistics

submitted by
Stefan Banholzer

Day of the oral examination: 7th July 2021

Referee: Prof. Dr. Stefan Volkwein, University of Konstanz
Referee: Prof. Dr. Michael Dellnitz, Paderborn University
Abstract

In many optimization problems arising from applications, several objectives are present. Thus, if these objectives are conflicting with each other, there is no canonical solution to these problems. This leads to the notion of multiobjective optimization, in which the goal is to compute the set of all optimal compromises (the Pareto set or Pareto front) between the conflicting objectives. Multiobjective optimization problems with PDE constraints are especially relevant in applications, since many occurring systems can be modeled by a PDE. This thesis is concerned with investigating methods for efficiently solving PDE-constrained multiobjective optimization problems with a potentially arbitrary number of objectives.

Scalarization methods are a popular tool for solving general multiobjective optimization problems. Their underlying idea is to transform the problem into a series of scalar optimization problems, which can then be solved by using well-known techniques from scalar optimization. For two specific methods – the Euclidean reference point method and the Pascoletti-Serafini method – a hierarchical approach of computing the Pareto front based on iteratively solving subproblems is presented and rigorously analyzed. The resulting algorithms are tested by a linear-quadratic multiobjective optimal control problem and a non-convex multiobjective parameter optimization problem.

However, solving these problems implies a high computational effort due to the high-dimensional equation systems arising from a discretization of the PDE. Model-order reduction techniques, which replace the high-dimensional model of the PDE with a low-dimensional reduced-order model, are a popular tool for reducing the computational effort in these cases. One issue in this context is finding a good balance between reducing the computational effort and the resulting approximation error. To this end, it is shown how a-posteriori error estimates and a Trust-Region method can be used to guarantee a desired error tolerance while still gaining a sufficient decrease of the computational effort. Numerical experiments on the two afore-mentioned examples are used to verify these approaches.
Zusammenfassung


Acknowledgements

First of all, I would like to express my sincerest gratitude to Prof. Dr. Stefan Volkwein for introducing me to the exciting topics of PDE-constrained optimization, multiobjective optimization and model-order reduction and for giving me the opportunity to further work on these topics. His support and guidance during my entire time in his group were extremely helpful. I want to thank him for the many fruitful discussions providing me with valuable new input and ideas, but also for giving me the freedom to follow my own ideas.

I am grateful to my former co-worker Dr. Dennis Beermann who guided me during the first time of my research. His counter example for three objective functions was the starting point for the main part of the first chapter.

I would also like to thank Prof. Dr. Michael Dellnitz, Dr. Sebastian Peitz and Bennet Gebken from the Paderborn University for the interesting and productive collaboration on the joint DFG project. My annual visits to Paderborn have been enlightening and enjoyable both on an academic and a personal level.

I gratefully acknowledge the financial support by the Deutsche Forschungsgesellschaft (DFG) Priority Programme 1962 – Non-smooth and Complementary-based Distributed Parameter Systems: Simulation and Hierarchical Optimization throughout my thesis. Moreover, I received financial support by the Landesgraduiertenförderung of Baden-Württemberg for three years.

To Christian Jäkle, Dr. Luca Mechelli, Dr. Georg Müller and Lena Reichle, I would like to express my gratitude for proof-reading parts of my thesis and giving me valuable hints and suggestions. All remaining errors in this thesis are mine.

Many thanks go to all my former and present colleagues at the University of Konstanz for the great atmosphere, the interesting mathematical and personal discussions, the refreshing coffee breaks – to which I was luckily accepted even as a non-coffee drinker – and the evenings spent together.

I am grateful to all the old friends from home and the new ones I have made throughout my studies both in Konstanz and in Uppsala. Thank you for your support, the time spent together and the many memorable moments.

# Contents

## Introduction 1

1 Multiobjective Optimization ........................................... 9
   1.1 Problem Formulation .............................................. 9
   1.2 Pareto Optimality ................................................ 10
      1.2.1 Properties of Pareto Optimality ........................... 15
   1.3 Hierarchical Structure of Pareto Sets ........................ 17
   1.4 General Scalarization Methods .................................. 21
      1.4.1 Solving a Multiobjective Optimization Problem .......... 23
      1.4.2 Quality Criteria for a Numerical Implementation .......... 24
   1.5 The Weighted-Sum Method ....................................... 25
      1.5.1 Problem Formulation ......................................... 26
      1.5.2 The Weighted-Sum Method for Subproblems ................. 31
   1.6 The Euclidean Reference Point Method ........................ 33
      1.6.1 Problem Formulation ......................................... 34
      1.6.2 Pareto Admissible Reference Points ......................... 37
      1.6.3 A Pareto Sufficient Set of Reference Points ............... 41
      1.6.4 The Hierarchical Euclidean Reference Point Method ....... 49
   1.7 The Pascoletti-Serafini Method .................................. 65
      1.7.1 Problem Formulation ......................................... 66
      1.7.2 Literature Review on Solution Strategies .................. 71
      1.7.3 A Pareto Sufficient Set of Reference Points ............... 73
      1.7.4 The Hierarchical Pascoletti-Serafini Method ............... 78

2 Multiobjective PDE-Constrained Optimization ...................... 107
   2.1 A Linear-Quadratic Multiobjective Optimal Control Problem .................. 108
      2.1.1 The State Equation ......................................... 108
      2.1.2 The Optimal Control Problem ............................... 111
      2.1.3 The Multiobjective Optimal Control Problem ............... 115
      2.1.4 Numerical Experiments ...................................... 115
   2.2 A Non-Convex Multiobjective Parameter Optimization Problem ........ 123
      2.2.1 The State Equation ......................................... 123
      2.2.2 The Parameter Optimization Problem ....................... 126
      2.2.3 The Multiobjective Parameter Optimization Problem ........ 129
      2.2.4 Numerical Experiments ...................................... 129
## 3 Model-Order Reduction for PDE-Constrained Multiobjective Optimization

3.1 POD-Based Linear-Quadratic Multiobjective Optimal Control
   3.1.1 Error Analysis for the Weighted-Sum Method
   3.1.2 Error Analysis for the Euclidean Reference Point Method
   3.1.3 Error Analysis for the Pascoletti-Serafini Method
   3.1.4 Certified Error Control
   3.1.5 Numerical Experiments

3.2 RB-Based Multiobjective Parameter Optimization
   3.2.1 A Trust-Region RB Algorithm
   3.2.2 Numerical Experiments

## 4 Conclusion and Outlook

A Model-Order Reduction
   A.1 The Reduced Basis Method
   A.2 Proper Orthogonal Decomposition

B An Augmented Lagrangian Penalty Method

Bibliography
Introduction

There is rarely a situation in decision making where only one objective is of interest. In daily life, one example for this observation is the decision for the best way of transportation in terms of price, safety, travel time, risk and environmental friendliness. A more technical example is the energy efficient heating of a house with the objectives of reaching a desired temperature and using as little energy as possible [FHC06].

Especially in technical applications, problems in decision making can often be modeled by an optimization problem. However, most of the times, the different objectives are conflicting with each other so that there is no single solution which optimizes all objectives simultaneously. One approach to deal with this issue is to set up a scalar optimization problem, which reflects predefined preferences with respect to the different objectives. These preferences might be expressed via different weighting factors, which model the intended trade-offs between the objectives, or via hard constraints on some objectives. However, there is no guarantee in practice that the resulting solution will be satisfactory, since, e.g., it is not easy to translate the preferences into suitable trade-off values before the optimization. In particular, one is limited to one solution without knowing about possibly more preferable alternatives. This drawback leads to the second approach, namely seeing the problem in the context of multiobjective optimization, in which the goal is not to compute only one solution but a set of optimal compromises between the objectives. Subsequently, based on this set, one can decide for the most preferred solution out of all given optimal alternatives.

The notion of an optimal compromise was independently formalized by Francis Ysidro Edgeworth (1845–1926) in 1881 [Edg81] and Vilfredo Federico Pareto (1848–1923) in 1906 [Par06] in the context of optimal allocation of resources of a society. Their definition states that an allocation is optimal if there is no other allocation in which no one is worse off and at least one individual is better off. Nowadays, this optimality notion is still the most commonly used in multiobjective optimization and is denoted by Edgeworth-Pareto optimality, or mostly just Pareto optimality.

The formalization of the concept of an optimal compromise has brought multiobjective optimization into the focus of mathematical research. Especially since the translation of Pareto’s work into English in 1971 [Par71], multiobjective optimization has become a vivid field of research in applied mathematics ([DW04]) with many authors contributing to it both from a theoretical and a practical point of view. Some mentionable works are the discovery of KKT-conditions for multiobjective optimization by Kuhn and Tucker in 1951 [KT51], the early survey on multiobjective optimization by Stadler [Sta79] and the extensive books on both theoretical and practical aspects of multiobjective optimization.
by Sawarigi [SNT85], Steuer [Ste86], Stadler [Sta88] and more recently Miettinen [Mie99], Ehrgott [Ehr05] and Jahn [Jah11].

In general, the set of optimal compromises is of infinite cardinality so that it is not possible to compute this set entirely. In fact, for many applications, even computing a good approximation of this set became only possible with the increase in computational power towards the end of the 20th century. Nowadays, four major approaches for computing such an approximation can be identified in the literature: Evolutionary algorithms [Deb01], set-oriented methods [DSH05, SWOBD13], homotopy methods [Hil01] and scalarization methods [Ehr05, Jah11, Mie99].

The basic idea of scalarization methods is to transform the multiobjective optimization problem (MOP) into a series of scalar optimization problems, which can then be solved by using well-known techniques from scalar optimization. To this end, a family of parameter-dependent scalarization functions, which map from the objective space to the real numbers, and the corresponding parameter set are defined. The goal is to find a suitable parameter set such that solving the scalarized optimization problem to every parameter value results in an optimal compromise and, reversely, every optimal compromise can be obtained by solving a scalarized optimization problem to a certain parameter value [Wie86]. There are many different possibilities for defining appropriate families of parameter-dependent scalarization functions. Popular examples are the weighted-sum scalarization [Zad63], reference point scalarizations with respect to the Euclidean norm [Wie80] or the Chebyshev norm [Mie99], the Pascoletti-Serafini scalarization [PS84], and the normal-boundary intersection (NBI) [DD98] amongst many others.

However, for many of these methods – especially of those whose parameter can be interpreted as a reference point – characterizing the required parameter set turns out to be quite complicated in practice, since it is problem dependent and cannot be determined before the computation. While simple geometrical arguments allow for a characterization of the parameter set in the case of two objective functions, this is not simply generalizable to more than two objective functions. One example is the NBI, for which the required reference points cannot be chosen by only using the individual minimizers of the objective functions and the so-called CHIM simplex for more than two objective functions, see [DD98, Figure 3]. This deficiency can only be overcome with a quite complex procedure, see [Das97]. Thus, for many of these scalarization methods, the resulting algorithms were only developed for MOPs with two objective functions.

In recent years, there were some approaches for tackling this problem. In [Eic08], the author could restrict the required parameter set for the Pascoletti-Serafini method to a hyperplane. By additionally solving some specific scalar optimization problems, the parameter set could be further bounded. In [MAL12, MGGS09], the idea of using subproblems, i.e., MOPs which only consist of a subset of all objective functions, to compute the Pareto front hierarchically by the Pascoletti-Serafini method was introduced. To be precise, the authors showed how the solution of subproblems containing $k - 1$ objective functions can be used to construct reference points for solving the subproblems containing $k$ objective functions. This approach was later also followed and enhanced by [KSd15] and in a similar way also in [DK19]. As a result, these algorithms can in principle solve MOPs with an arbitrary number of objective functions.
One relevant application for multiobjective optimization is the field of optimization with partial differential equation (PDE) constraints. PDEs are an important tool for modeling complex systems occurring in various fields such as, e.g., engineering, natural sciences and economy. One simple example is the so-called heat equation, which describes how the temperature distribution in a given domain evolves over time. In many applications, one is not only interested in modeling these systems, but also wants to influence them in such a way that the outcome is optimal with respect to a given objective. This leads to the notion of a PDE-constrained optimization problem. It is characteristic for these problems that there is a natural splitting of the optimization variable into an input $u$, whose value influences the outcome $y$ of the system, which is often referred to as the state.

Two important special cases of PDE-constrained optimization problems are optimal control problems [Lio71, Trö10] and parameter optimization problems [HPUU09]. In optimal control problems, the input $u$ is given as an external control acting on the system, whereas in parameter optimization problems, $u$ is seen as a parameter of the PDE. Coming back to the example of the heat equation, an optimal control problem might be given by optimizing a heating input such that the temperature distribution reaches or comes at least close to a desired temperature distribution. On the other hand, a parameter optimization problem could consist of optimizing the diffusivity in the room to reach the desired temperature distribution.

In a natural way, PDE-constrained optimization problems can also be seen in the context of multiobjective optimization, since there are often several objectives involved. However, in most publications, they are seen in the framework of scalar optimization [HPUU09, Trö10]. In particular, the focus is on studying properties of scalar PDE-constrained optimization problems and developing algorithms for solving them. Consequently, the literature on multiobjective PDE-constrained optimization problems is not very rich and has only grown recently. The weighted-sum method was applied to a multiobjective optimal control problem (MOCP) of a semilinear parabolic PDE in [ITV16], and to a linear-quadratic MOCP of an elliptic PDE in [IUV17]. The Euclidean reference point method was used in [POBD19] for solving an MOCP for fluid flow and in [Ban17, BBV17, BMV19, Mak18, Spu19] for solving a linear-quadratic MOCP of a parabolic PDE. In [Bee19], the weighted-sum method and the Euclidean reference point method were used to solve both linear-quadratic and non-convex MOCPs of a parabolic PDE. A set-oriented method was applied to solve an MOCP of a semilinear heat equation in [BDPV18]. In [BGD+19], a set-oriented homotopy method, which is based on [SDD05], was used to solve a multiobjective parameter optimization problem (MPOP) of an elliptic PDE.

One reason why the study of multiobjective PDE-constrained optimization problems has only recently come into the focus of researchers is the (numerical) complexity of these problems. Solving a PDE on its own is already computationally challenging, since the arising equation systems, which often stem from a Finite Element (FE) discretization of the PDE, are high-dimensional ($10^3$–$10^6$ degrees of freedom depending on the PDE). When optimizing with PDE constraints, it is inevitable to solve the PDE numerous times for different input parameters. Moreover, independent of the utilized solution method for solving a multiobjective PDE-constrained optimization problem, the computational effort
increases drastically in comparison to a scalar PDE-constrained optimization problem. Even on modern computers, this can often not be done in an acceptable amount of time. One idea to circumvent this problem is the use of model-order reduction (MOR) [BGTQ+21, SvdVR08]. In MOR, the idea is to replace the high-dimensional model of the PDE by a low-dimensional surrogate model. There are various different techniques for MOR such as, e.g., balanced truncation [Ant05, BQOQO00, GA04, MS05], the reduced basis (RB) method [Haa17, HRS16, PR07, QMN16, RHP07] and proper orthogonal decomposition (POD) [HLBR12, KV99, KV01, LC18, RP03, Sir87].

The RB method is especially interesting in the field of parametrized PDEs [Haa17, PR07, RHP07]. The idea is to use solutions of the high-dimensional model of the PDE at a few carefully chosen parameter values to approximate the entire solution ‘manifold’ of the PDE. One way of determining these parameter values is to split the computation into an offline and an online phase. In the costly offline phase, the RB model is generated by using, e.g., a greedy method, which ensures a desired exactness of the RB model for the entire parameter set, see [BMP+12, VPRP03]. In the online phase, the low-dimensional surrogate model can then be evaluated cheaply for different parameter values. Another approach is the adaptive basis enrichment, which is especially suitable in PDE-constrained optimization. Given an initial RB space, it is enriched along the optimization path in order to maintain a desired level of accuracy for the reduced-order objective function and its gradient, see, e.g., [QGVW17, YM13].

The POD method has been mainly used for finding the dominant structures for dynamical systems arising, e.g., from time-dependent PDEs [Lum67, LC18, Sir87]. Given one or several solution trajectories, the idea is to find a POD space of dimension $\ell$ such that a mean-square error between the snapshots and their projection onto the POD space are minimized. This can be lead back to a singular value decomposition of the given snapshots, see [GV17].

The POD method has been mainly used for finding the dominant structures for dynamical systems arising, e.g., from time-dependent PDEs [Lum67, LC18, Sir87]. Given one or several solution trajectories, the idea is to find a POD space of dimension $\ell$ such that a mean-square error between the snapshots and their projection onto the POD space are minimized. This can be lead back to a singular value decomposition of the given snapshots, see [GV17].

The challenge of using MOR for solving PDE-constrained optimization problems is to find a good balance between reducing the computational effort and guaranteeing that the error between the reduced-order and the full-order solutions is small. To this end, in [GNV17, GV17, KTV13, KG14b, TV09], the authors use a-posteriori error estimates, which estimate the error between the reduced-order and the full-order solution only based on the reduced-order solution. In the literature, this is often called ‘certified’, since the a-posteriori error estimate is rigorous, i.e., it is an upper bound on the error. Based on this observation, a-posteriori error estimates can be used together with update schemes for the reduced-order model to ensure that the error is always below a desired error tolerance [GV17, TV09]. Moreover, in the last two decades, adaptations of Trust-Region algorithms ([CGT00]) to the framework of MOR have been proposed [AFS00, Fah00, KMO+20, QGVW17, Rog14, YM13]. In these strategies, a-posteriori error estimates for the objective function and its gradient are used to decide whether the reduced-order model has to be updated or not at the current iterate.

The combination of multiobjective PDE-constrained optimization and MOR has already been investigated in the literature. In [ITV16, IUV17], a POD greedy algorithm ([HO08]) and the RB method, respectively, were used to reduce the computational effort.
[POBD19], the POD method was combined with the Euclidean reference point method for solving an MOCP for fluid flow. An a-posteriori error estimate for the Euclidean reference point method was introduced in [Ban17, BBV17], which laid the basis for using a certified POD basis update scheme in [Ban17, BBV17, BMV19, Mak18, Spu19]. In [Bee19], the author used similar estimates and a POD update scheme for the weighted-sum method and the Euclidean reference point method. It was shown how to use a-posteriori error estimates for the state and adjoint equation in a gradient-free, set-oriented method in [BDPV18]. In [Pei17], a Trust-Region approach was applied to an MOCP. The RB method was combined with a set-oriented homotopy approach in [BDG19], to which the author made substantial contributions. However, this is not within the scope of this thesis. A good survey can also be found in [PD18].

Outline

This thesis focuses on investigating and developing methods for solving general convex and non-convex MOPs with a potentially arbitrary number of objective functions, which are then applied to multiobjective PDE-constrained optimization problems. Due to the resulting computational effort, it is shown how well-known techniques from MOR can be combined with these methods in order to gain a significant decrease of the computational time while maintaining a desired level of accuracy for the solutions.

In Chapter 1, we deal with general MOPs and focus on developing methods for solving MOPs with an arbitrary number of objective functions by the Euclidean reference point method and the Pascoletti-Serafini method. The idea for these algorithms is based on hierarchically solving subproblems of the MOP.

- An appropriate framework and notation for dealing with general MOPs is introduced in Section 1.1.
- In Section 1.2, we introduce different concepts of Pareto optimality and illustrate them with some simple examples. Moreover, we show basic properties such as the equivalence of the different notions of Pareto optimality and sufficient conditions for the existence of Pareto optimal points.
- By introducing the concept of a subproblem, we study the hierarchical structure of Pareto sets in Section 1.3. In particular, we show that a Pareto optimal point of a subproblem is at least weakly Pareto optimal for all problems containing this subproblem. In the further course, we illustrate how this structural property can be used, in principle, to solve the MOP in a hierarchical way by subsequently solving all of its subproblems.
- In Section 1.4, we introduce the general idea of scalarization methods and show that they can compute every Pareto optimal point. In Section 1.4.1, the approach of defining a family of scalarization functions to solve an MOP is described. In particular, some criteria are introduced that a scalarization method should satisfy. In the same way, we discuss three quality criteria for evaluating the numerical implementation of a scalarization method in Section 1.4.2.
• The weighted-sum method is the focus of Section 1.5. By using existing results from the literature, we show that it can successfully deal with convex MOPs from a theoretical point of view. These results are summarized by defining certain solution operators, which will be used for the Euclidean reference point method in Section 1.6. After discussing well-known drawbacks of the weighted-sum method in the numerical implementation, which make the method less attractive in applications, the results from the first part of this section are transferred to the notation of subproblems in Section 1.5.2.

• In Section 1.6, the Euclidean reference point method is studied. By introducing an extended Euclidean reference point problem and using its connection to the weighted-sum problem, we show that the Euclidean reference point method can solve MOPs in a convex framework if the reference points, i.e., the parameters of the method, are chosen appropriately. Subsequently, we are concerned with the question of how such a suitable set of reference points can be determined. To this end, by using the connection to the weighted-sum method once more, we show that this set can be characterized by the solutions to all subproblems of the MOP. By transferring this result to subproblems, a hierarchical algorithm for solving an MOP with arbitrarily many objective functions is developed. For this algorithm, we prove analytically that it computes every Pareto optimal point. Moreover, we show that its numerical implementation guarantees a good approximation of the Pareto front, which is tested and compared to the weighted-sum method in a polynomial example with three objective functions.

• The Pascoletti-Serafini method is studied in Section 1.7. We start by showing that it is suitable for solving quite general, possibly non-convex MOPs. As for the Euclidean reference point method, we proceed with the question of how a suitable set of reference points can be determined, and it turns out that this can also be done via the solutions to subproblems. However, for non-convex problems we cannot completely characterize this set, but obtain that there is a gap between the sufficient and necessary conditions. Nevertheless, following similar steps as for the Euclidean reference point method, the necessary condition allows us to define an algorithm for solving MOPs with an arbitrary number of objective functions by the Pascoletti-Serafini method, similar to the one presented in [KSD15, MAL12, MGGS09]. Its numerical implementation is shown to guarantee a good approximation of the Pareto front. We test this algorithm and compare it to the weighted-sum method and the Euclidean reference point method by conducting numerical experiments on a non-convex example from the literature with four objective functions.

In Chapter 2, the developed numerical algorithms from Chapter 1 for solving MOPs by the weighted-sum method, the Euclidean reference point method and the Pascoletti-Serafini method are tested and compared by applying them to two PDE-constrained optimization problems.

• A linear-quadratic multiobjective optimal control problem (MOCP) stemming from the energy efficient heating, ventilation and air conditioning (HVAC) operation of
a room is introduced in Section 2.1. The governing state equation is a parabolic diffusion-advection equation on which the control acts as a distributed heating input. For this equation, we show that it is uniquely solvable for every control input. As objective functions, we choose weighted sums of terms of tracking type and of terms measuring the control input. We subsequently show that the gradients and the Hessians of these objective functions can be numerically evaluated by using the so-called adjoint equations. In a numerical experiment, we consider an MOCP with three objective functions and compare the performance of the three scalarization methods in terms of the quality criteria introduced in Section 1.4.2 as well as the induced computational costs.

As a second model problem, we introduce a non-convex multiobjective parameter optimization problem (MPOP) of an elliptic PDE in Section 2.2. The governing elliptic PDE is a diffusion-reaction equation, in which the parameters are given by the diffusivity of certain subdomains and the reactivity. For a certain parameter range, we can show that this equation is uniquely solvable. However, the parameters act on the bilinear form of the PDE, so that the parameter-to-state mapping is non-linear. Since the objective functions are again given as weighted sums of terms of tracking type and of terms measuring the parameter input, this implies that the problem is non-convex. We can show that it fits into the non-convex framework from Section 1.7. For the numerical experiment, we use an example with three objective functions and compare the results obtained by using the three scalarization methods by the means of the quality measures introduced in Section 1.4.2 and the induced computational costs.

The application of model-order reduction to the two multiobjective PDE-constrained optimization problems from Chapter 2 is studied in Chapter 3.

1. In Section 3.1, we investigate the application of the POD method to general linear-quadratic MOCPs. We start by reviewing existing a-priori and a-posteriori error estimates for the weighted-sum method and the Euclidean reference point method from the literature. Based on the same principles, we show corresponding results for the Pascoletti-Serafini method. The a-posteriori error estimates together with a simple update scheme for the POD basis allow us to construct an algorithm for controlling the approximation error throughout the multiobjective optimization. In numerical experiments on the MOCP from Section 2.1, we first analyze the a-posteriori error estimates with respect to their overestimation of the true error, before we apply the developed algorithm to the three scalarization methods. These results are analyzed and compared to results obtained by using an a-priori computed POD basis.

2. In Section 3.2, we introduce and describe a Trust-Region (TR-) RB algorithm for solving PDE-constrained parameter optimization problems, which was recently published in [BKM+20]. In particular, under some assumptions on the problem and the involved cost functions, a convergence result is presented. Subsequently, we show that the scalar optimization problems obtained from applying the three
scalarization methods to the MPOP from Section 2.2 satisfy these assumptions. In numerical experiments conducted on the Pascoletti-Serafini method, we verify the convergence result for the TR-RB algorithm and compare the results with those obtained from using the FE method and the typical offline-online splitting of the RB method in combination with a greedy algorithm.

Following this, we draw a conclusion and give an outlook on possible further interesting work.

In Appendix A, the basics of the RB and the POD method are explained and it is shown how they can be applied to scalar PDE-constrained optimization problems. Finally, an augmented Lagrangian penalty algorithm, which is used for solving the Pascoletti-Serafini problems numerically, is introduced in Appendix B.
Multiobjective Optimization

Many optimization problems arising in applications can be formulated using several objective functions. In most cases, it is not possible to achieve the optimal values of all functions simultaneously, since the cost functions are conflicting with each other. This leads to the notion of a multiobjective optimization problem (MOP), in which the goal is to compute the set of optimal compromises between the cost functions.

After briefly introducing the notation for a general MOP in Section 1.1, we define and analyze properties of different notions of Pareto optimality in Section 1.2. Subsequently, we show the hierarchical structure of the set of all Pareto optimal points in Section 1.3. The idea of scalarization methods is presented in Section 1.4. Moreover, a general approach of solving MOPs by scalarization methods is presented and some quality criteria of such a method are introduced. Afterwards, we introduce three different scalarization methods in the Sections 1.5, 1.6 and 1.7. Well-known results for the weighted-sum method for convex MOPs from the literature are presented in Section 1.5 for the purpose of using them later in Section 1.6 for the Euclidean reference point method. In there, we show how the hierarchical structure of the set of all Pareto optimal points can be used to develop a procedure for hierarchically solving convex MOPs by the Euclidean reference point method. For solving non-convex MOPs, the Pascoletti-Serafini method is investigated in Section 1.7. As for the Euclidean reference point method, we show how an algorithm for hierarchically solving an MOP by the Pascoletti-Serafini method can be constructed. For both the Euclidean reference point method and the Pascoletti-Serafini method, these approaches are based on the ideas presented in [MAL12, MGGS09].

Note that the results from the Sections 1.5 and 1.6 have been published in a similar form as a preprint [BV19], to which the author made substantial contributions.

1.1 Problem Formulation

For the rest of this chapter, let \((U, \langle \cdot, \cdot \rangle_U)\) be a real Hilbert space, \(U_{\text{ad}} \subset U\) non-empty, convex and closed, \(k \geq 2\) arbitrary and \(\hat{J}_1, \ldots, \hat{J}_k: U_{\text{ad}} \subset U \rightarrow \mathbb{R}\) be given real-valued functions. To shorten the notation, we write \(\hat{J} := (\hat{J}_1, \ldots, \hat{J}_k)^T: U_{\text{ad}} \rightarrow \mathbb{R}^k\). In the following, we deal with the multiobjective optimization problem

\[
\min_{u \in U_{\text{ad}}} \hat{J}(u) = \min_{u \in U_{\text{ad}}} \begin{pmatrix} \hat{J}_1(u) \\ \vdots \\ \hat{J}_k(u) \end{pmatrix}.
\]
We introduce the following terminologies for the multiobjective optimization problem (MOP).

**Definition 1.1.1.** (i) The functions $\hat{J}_1, \ldots, \hat{J}_k$ are called cost functions or objective functions. Analogously, the vector-valued function $\hat{J}: U_{ad} \to \mathbb{R}^k$ is named the (multiobjective) cost function or (multiobjective) objective function.

(ii) The Hilbert space $U$ is the admissible space, the set $U_{ad}$ is called the admissible set and a vector $u \in U_{ad}$ is called admissible.

(iii) The space $\mathbb{R}^k$ is named the objective space and the image set $\hat{J}(U_{ad})$ is called the objective set. An element $y = \hat{J}(u) \in \hat{J}(U_{ad})$ is called objective point.

### 1.2 Pareto Optimality

The definition of an optimal compromise for (MOP) depends on the chosen optimality concept. The most popular concept is the Pareto optimality or Pareto efficiency, which dates back to works of Vilfredo Pareto [Par06] and Francis Ysidro Edgeworth [Edg81]. Therefore, it is referred to as Pareto-Edgeworth or Pareto optimality in the literature. In this thesis, we will use the more common notion of Pareto optimality. In this section, we introduce different notions of Pareto optimality for (MOP). To visualize these concepts, we provide several examples for different situations which might occur before some properties of the set of all Pareto optimal solutions are shown.

In contrast to a scalar-valued optimization problem, it is a priori not clear how to define a solution of (MOP). The reason for this is that there is no canonical ordering on $\mathbb{R}^k$ for $k \geq 2$.

**Definition 1.2.1** (Partial/Total ordering). Let $\mathcal{X}$ be an arbitrary set. A binary relation $\leq$ is called a partial ordering on $\mathcal{X}$ if

(i) $\forall x \in \mathcal{X}$: $x \leq x$ (reflexivity),

(ii) $\forall x, y \in \mathcal{X}$: $(x \leq y \& y \leq x) \Rightarrow x = y$ (antisymmetry),

(iii) $\forall x, y, z \in \mathcal{X}$: $(x \leq y \& y \leq z) \Rightarrow x \leq z$ (transitivity).

A partial ordering $\leq$ on $\mathcal{X}$ is called a total ordering if for every $x, y \in \mathcal{X}$ it holds $x \leq y$ or $y \leq x$.

From the viewpoint of optimization, it seems to be preferable to choose a total ordering on $\mathbb{R}^k$ (e.g., the lexicographical ordering), since this would make all vectors in $\mathbb{R}^k$ comparable to each other and would allow for a unique solution to the problem. However, in most MOPs, the objective functions are conflicting with each other. Thus, it might actually be not desirable to find only one solution, but rather a set of solutions which form optimal compromises or trade-offs between the objectives. Partial orderings are more suitable for this situation, since they allow vectors to be incomparable, i.e., to form compromises. The most common choice is the product ordering of the natural ordering on $\mathbb{R}$, which compares two points $x, y \in \mathbb{R}^k$ componentwise.
1.2 Pareto Optimality

Figure 1.1: The sets \( y + \mathbb{R}^2_\geq = \{ x \in \mathbb{R}^2 \mid x \geq y \} \) and \( y + \mathbb{R}^2_\leq = \{ x \in \mathbb{R}^2 \mid x \leq y \} \).

**Definition 1.2.2** (Product ordering on \( \mathbb{R}^k \)). On \( \mathbb{R}^k \) we define the partial ordering \( \leq \), which is given by

\[
x \leq y : \iff (\forall i \in \{1, \ldots, k\} : x_i \leq y_i)
\]

for all \( x, y \in \mathbb{R}^k \). Moreover, we define

\[
x < y : \iff (\forall i \in \{1, \ldots, k\} : x_i < y_i).
\]

For convenience, we write

\[
x \preceq y : \iff (x \leq y \land x \neq y)
\]

for all \( x, y \in \mathbb{R}^k \) and define the sets

\[
\mathbb{R}^k_\geq := \{ y \in \mathbb{R}^k \mid y \leq 0 \},
\]

\[
\mathbb{R}^k_\leq := \{ y \in \mathbb{R}^k \mid y \geq 0 \}.
\]

Analogously, the relations \( \geq \), \( > \) and \( \preceq \) as well as the sets \( \mathbb{R}^k_\geq \) and \( \mathbb{R}^k_\leq \) are defined.

From a mathematical point of view, it is possible to study MOPs on an abstract level without specifying the partial ordering by using the connection of partial orderings and cones: On the one hand, any pointed convex cone \( C \subset \mathbb{R}^k \) induces a partial ordering \( \leq_C \) on \( \mathbb{R}^k \) by the relation \( x \leq_C y \iff y - x \in C \). On the other hand, any partial ordering \( \leq_k \) induces the pointed convex cone \( C := \{ x \in \mathbb{R}^k \mid 0 \leq_k x \} \subset \mathbb{R}^k \) (cf. [Ehr05, Theorem 1.17]). This general approach is followed, e.g., in [Ehr05, Eic08, Jah11].

In this thesis, we will use the product ordering on \( \mathbb{R}^k \), which was introduced in Definition 1.2.2. Its induced pointed convex cone \( C \) is given by \( C = \mathbb{R}^k_\geq \). Therefore, for any \( y \in \mathbb{R}^k \) it holds \( \{ x \in \mathbb{R}^k \mid x \geq y \} = y + \mathbb{R}^k_\geq \) and \( \{ x \in \mathbb{R}^k \mid x \leq y \} = y - \mathbb{R}^k_\leq \), see Figure 1.1. Using the product ordering to define the concept of optimal compromises for (MOP) leads to the notion of Pareto optimality.
Definition 1.2.3. (i) An admissible vector \( \bar{u} \in \mathcal{U}_{ad} \) and its corresponding objective point \( \bar{y} := \bar{J}(\bar{u}) \in \bar{J}(\mathcal{U}_{ad}) \) are called (locally) weak Pareto optimal if there is no \( \hat{u} \in \mathcal{U}_{ad} \) (in a neighborhood of \( \bar{u} \)) with \( \bar{J}(\hat{u}) < \bar{J}(\bar{u}) \). The sets

\[
\mathcal{U}_{opt,w} := \{ u \in \mathcal{U}_{ad} \mid u \text{ is weakly Pareto optimal} \} \subset \mathcal{U}_{ad},
\]

\[
\mathcal{U}_{opt,w,loc} := \{ u \in \mathcal{U}_{ad} \mid u \text{ is locally weakly Pareto optimal} \} \subset \mathcal{U}_{ad}
\]

are called the weak Pareto set and the locally weak Pareto set, respectively, and the sets

\[
\mathcal{J}_{opt,w} := \bar{J}(\mathcal{U}_{opt,w}) \subset \mathbb{R}^k,
\]

\[
\mathcal{J}_{opt,w,loc} := \bar{J}(\mathcal{U}_{opt,w,loc}) \subset \mathbb{R}^k,
\]

are called the weak Pareto front and the locally weak Pareto front, respectively.

(ii) An admissible vector \( \bar{u} \in \mathcal{U}_{ad} \) and its corresponding objective point \( \bar{y} := \bar{J}(\bar{u}) \in \bar{J}(\mathcal{U}_{ad}) \) are called (locally) Pareto optimal if there is no \( \hat{u} \in \mathcal{U}_{ad} \) (in a neighborhood of \( \bar{u} \)) with \( \bar{J}(\hat{u}) \not\leq \bar{J}(\bar{u}) \). The sets

\[
\mathcal{U}_{opt} := \{ u \in \mathcal{U}_{ad} \mid u \text{ is Pareto optimal} \} \subset \mathcal{U}_{ad},
\]

\[
\mathcal{U}_{opt,loc} := \{ u \in \mathcal{U}_{ad} \mid u \text{ is locally Pareto optimal} \} \subset \mathcal{U}_{ad}
\]

are called the Pareto set and the local Pareto set, respectively, and the sets

\[
\mathcal{J}_{opt} := \bar{J}(\mathcal{U}_{opt}) \subset \mathbb{R}^k,
\]

\[
\mathcal{J}_{opt,loc} := \bar{J}(\mathcal{U}_{opt,loc}) \subset \mathbb{R}^k
\]

are called the Pareto front and the local Pareto front, respectively.

Remark 1.2.4. (i) The notion of (weak) Pareto optimality can also be defined by using the ordering cones: An admissible vector \( \bar{u} \in \mathcal{U}_{ad} \) and its corresponding objective point \( \bar{y} := \bar{J}(\bar{u}) \in \bar{J}(\mathcal{U}_{ad}) \) are (weakly) Pareto optimal if and only if it holds

\[
(\bar{y} - \mathbb{R}^k_{\geq}) \cap \bar{J}(\mathcal{U}_{ad}) = \{ \bar{y} \} \quad (\bar{y} - \mathbb{R}^k_{\geq}) \cap \bar{J}(\mathcal{U}_{ad}) = \emptyset.
\]

(ii) If we want to talk about the different notions of (local) (weak) Pareto optimality in one sentence, we use the notation \( \mathcal{U}_{opt,(w),(loc)} \) to keep the sentence compact. Analogously, \( \mathcal{U}_{opt,(w,loc), \mathcal{J}_{opt,(w),(loc)} \mathcal{J}_{opt,(w,loc)}} \) etc. are to be understood.

Definition 1.2.5. (i) We define the ideal objective point \( y^id \in \mathbb{R}^k \cup \{-\infty\} \) by

\[
y^id := \inf_{u \in \mathcal{U}_{ad}} \bar{J}_i(u) \quad \text{for all } i \in \{1, \ldots, k\}.
\]

(ii) The (local) (weak) nadir objective point \( y^{nad,(w),(loc)} \in \mathbb{R}^k \cup \{-\infty, \infty\} \) is given by

\[
y^{nad,(w),(loc)} := \sup\{ y_i \mid y \in \mathcal{J}_{opt,(w),(loc)} \} \quad \text{for all } i \in \{1, \ldots, k\}.
\]

Note that we use the convention that the supremum over an empty set is \( -\infty \).
The following examples illustrate the different concepts of Pareto optimality.

**Example 1.2.6.**  
(i) Let \( \mathcal{U} = \mathbb{R}^2 \) and define the admissible set \( \mathcal{U}_{ad} = \mathcal{U} \) as well as the cost functions \( \hat{J}_1, \hat{J}_2 : \mathcal{U}_{ad} \to \mathbb{R} \) by

\[
\hat{J}_1(u) := u_1^2 + u_2^2, \\
\hat{J}_2(u) := (u_1 - 1)^2 + (u_2 - 1)^2
\]

for all \( u \in \mathcal{U}_{ad} \). Since the cost functions measure the Euclidean distance of an admissible vector \( u \in \mathcal{U}_{ad} \) to the points \( (0, 0)^T \) and \( (1, 1)^T \), respectively, it can be concluded that the (local) (weak) Pareto set is given by the connecting line \( V := \{(v, v)^T | v \in [0, 1]\} \) of the two points. Indeed, for any point \( u \not\in V \) it holds \( \hat{J}(P_V(u)) < \hat{J}(u) \), where \( P_V \) is the canonical projection onto \( V \). It is also clear that any \( u \in V \) is Pareto optimal, since for the functions \( \hat{J}_i^p : [0, 1] \to \mathbb{R}, v \mapsto \hat{J}_i((v, v)^T) \) \( (i = 1, 2) \) we have that \( \hat{J}_i^p \) is strictly monotonically increasing, whereas \( \hat{J}_i^p \) is strictly monotonically decreasing. Consequently, it holds

\[
\mathcal{U}_{opt} = \mathcal{U}_{opt, w} = \mathcal{U}_{opt, loc} = \mathcal{U}_{opt, w, loc} = V,
\]

see Figure 1.2(a). This directly implies

\[
\mathcal{J}_{opt} = \mathcal{J}_{opt, w} = \mathcal{J}_{opt, loc} = \mathcal{J}_{opt, w, loc} = \hat{J}(\mathcal{U}_{opt}),
\]

which is depicted in Figure 1.2(b) together with the entire objective set. Note that for two cost functions it is relatively easy to determine the Pareto front by using Remark 1.2.4 in case that the entire objective set is given. This confirms our theoretical considerations that it holds

\[
\mathcal{J}_{opt} = \mathcal{J}_{opt, w} = \mathcal{J}_{opt, loc} = \mathcal{J}_{opt, w, loc} = \hat{J}(V)
\]

for this example.
1. Multiobjective Optimization

Figure 1.3: Visualization of the (weak) Pareto set and the (weak) Pareto front for Example 1.2.6 (ii).

(ii) Let again $\mathcal{U} = \mathbb{R}^2$ and $\mathcal{U}_{ad} = \mathcal{U}$ and define the cost functions by

$$
\tilde{J}_1(u) := \begin{cases} 
  u_1^2 + u_2^2, & \text{if } u_1^2 + u_2^2 \geq 0.1, \\
  0.1, & \text{if } u_1^2 + u_2^2 < 0.1,
\end{cases}
$$

$$
\tilde{J}_2(u) := (u_1 - 1)^2 + (u_2 - 1)^2
$$

for all $u \in \mathcal{U}_{ad}$. Similar arguments as for the first example show that

$$
\mathcal{U}_{opt} = \mathcal{U}_{opt,loc} = \{(v,v)^T \mid v \in [\sqrt{0.05}, 1]\}
$$

and

$$
\mathcal{U}_{opt,w} = \mathcal{U}_{opt,w,loc} = \mathcal{U}_{opt} \cup \{u \in \mathbb{R}^2 \mid \|u\|^2 \leq 0.1\}.
$$

In particular, we have $\mathcal{U}_{opt} \neq \mathcal{U}_{opt,w}$, see Figure 1.3(a). By looking at the objective set $\tilde{J}(\mathcal{U}_{ad})$, which is depicted in Figure 1.3(b), the set $\mathcal{J}_{opt,w} \setminus \mathcal{J}_{opt}$ can be easily identified as the vertical line segment plotted in green: For any point on the line segment it is possible to decrease the second component while the first component remains constant, but it is not possible to decrease both components at the same time.

(iii) For the last example let $\mathcal{U} = \mathbb{R}$ and $\mathcal{U}_{ad} = [0, 1]$. The cost functions are given by

$$
\tilde{J}_1(u) := u,
$$

$$
\tilde{J}_2(u) := -u - 0.2 \cos(4\pi u)
$$

for all $u \in [0, 1]$. Note that the graph of the function $\tilde{J}_2$ coincides with the objective set $\tilde{J}(\mathcal{U}_{ad})$, since the first cost function is the identity. Moreover, the first cost function is strictly monotonically increasing on $\mathcal{U}_{ad}$, which implies that the local (weak) Pareto set is given by all points $u \in \mathcal{U}_{ad}$, which are no local maximizer of $\tilde{J}_2$.
1.2 Pareto Optimality

In this section, we study important properties of Pareto optimality, which increase the understanding of the concept of Pareto optimality and will help us later when developing methods for solving MOPs.

1.2.1 Properties of Pareto Optimality

In this section, we study important properties of Pareto optimality, which increase the understanding of the concept of Pareto optimality and will help us later when developing methods for solving MOPs.

1.2.1.1 Equivalence of Optimality Notions

As for scalar-valued optimization problems, local and global (weak) Pareto optimality coincide if the cost functions are convex.

Lemma 1.2.7. If the cost functions \( \hat{J}_1, \ldots, \hat{J}_k \) are convex, then \( U_{\text{opt.(w),loc}} = U_{\text{opt.(w)}} \).

Proof. This can be proven in the same way as [Bee19, Lemma 3.10].
For strictly convex cost functions, it is even possible to show that weak Pareto optimality is equivalent to Pareto optimality.

**Lemma 1.2.8.** If the cost functions \( \hat{J}_1, \ldots, \hat{J}_k \) are strictly convex, then we have that \( U_{\text{opt}} = U_{\text{opt},w} \).

**Proof.** A proof can be found in [BV19, Lemma 2.4].

**Remark 1.2.9.** The Lemmata 1.2.7 and 1.2.8 provide further insight into Example 1.2.6.

(i) In Example 1.2.6 (i), both cost functions are strictly convex, so that Lemma 1.2.8 implies the equality \( U_{\text{opt}} = U_{\text{opt},w} = U_{\text{opt},\text{loc}} = U_{\text{opt},w,\text{loc}} \).

(ii) Both cost functions of Example 1.2.6 (ii) are convex. Thus, by Lemma 1.2.7, it holds \( U_{\text{opt}} = U_{\text{opt},\text{loc}} \) and \( U_{\text{opt},w} = U_{\text{opt},w,\text{loc}} \).

(iii) The second cost function of Example 1.2.6 (iii) is non-convex, so that neither Lemma 1.2.7 nor Lemma 1.2.8 are applicable. And indeed, we find that the local and global (weak) Pareto sets do not coincide. \( \Box \)

### 1.2.1.2 Existence of Pareto Optimal Points

One central question in multiobjective optimization is whether there are any (locally) (weakly) Pareto optimal points. In this section, we provide some sufficient existence conditions.

We start by showing that any global (local) minimizer of an individual cost function \( \hat{J}_i \) is at least (locally) weakly Pareto optimal for (MOP). In this sense, weak Pareto optimality is an extension of the standard notion of optimality for scalar-valued optimization problems. However, it cannot be guaranteed that any global (local) minimizer is also (locally) Pareto optimal.

**Lemma 1.2.10.** If \( \bar{u} \in U_{\text{ad}} \) is a global (local) minimizer of the cost function \( \hat{J}_i \) for an arbitrary \( i \in \{1, \ldots, k\} \), then it holds \( \bar{u} \in U_{\text{opt},w,(\text{loc})} \). If \( \bar{u} \notin U_{\text{opt},(\text{loc})} \), then there is another global (local) minimizer \( \tilde{u} \) of \( \hat{J}_i \) (in every neighborhood of \( \bar{u} \)) with \( \hat{J}(\bar{u}) \leq \hat{J}(\tilde{u}) \).

**Proof.** Let \( \bar{u} \in U_{\text{ad}} \) be a global (local) minimizer of the cost function \( \hat{J}_i \) for an arbitrary \( i \in \{1, \ldots, k\} \). Then there is no \( u \in U_{\text{ad}} \) (in a neighborhood of \( \bar{u} \)) with \( \hat{J}_i(u) < \hat{J}_i(\bar{u}) \). In particular, this implies that there is no \( u \in U_{\text{ad}} \) (in a neighborhood of \( \bar{u} \)) with \( \hat{J}(u) < \hat{J}(\bar{u}) \). Hence, it holds \( \bar{u} \in U_{\text{opt},w,(\text{loc})} \).

If \( \bar{u} \notin U_{\text{opt},(\text{loc})} \), there is \( \tilde{u} \in U_{\text{ad}} \) (in every neighborhood of \( \bar{u} \)) with \( \hat{J}(\tilde{u}) \leq \hat{J}(\bar{u}) \). In particular, this implies \( \hat{J}(\bar{u}) \leq \hat{J}(\tilde{u}) \), so that \( \tilde{u} \) is also a global (local) minimizer of \( \hat{J}_i \). \( \Box \)

**Remark 1.2.11.** Coming back to Example 1.2.6 (ii), \( \{u \in \mathbb{R}^2 \mid ||u||^2 \leq 0.1\} \subset U_{\text{opt},w} \) follows directly from Lemma 1.2.10, since this is the set of global minimizers of \( \hat{J}_1 \).
Moreover, the second statement of Lemma 1.2.10 applies to this example, since we have that \( \{ u \in \mathbb{R}^2 \mid \|u\|^2 \leq 0.1 \} \not\subset U_{\text{opt}} \). In fact, 
\[ \tilde{u} \not\in U_{\text{opt}} \text{ for any } \tilde{u} \in \{ u \in \mathbb{R}^2 \mid \|u\|^2 \leq 0.1 \} \setminus \{(\sqrt{0.05}, \sqrt{0.05})^T\}, \]
since \((\sqrt{0.05}, \sqrt{0.05})^T\) is also a global minimizer of \( \hat{J}_1 \) and 
\[ \hat{J}_2((\sqrt{0.05}, \sqrt{0.05})^T) < \hat{J}_2(\tilde{u}) \text{ for all } \tilde{u} \in \{ u \in \mathbb{R}^2 \mid \|u\|^2 \leq 0.1 \} \setminus \{(\sqrt{0.05}, \sqrt{0.05})^T\} \]
is satisfied.

More generally, it is possible to show the following existence result, which goes back to [Bor83] and also appears in a similar form in [Har78, SNT85].

**Theorem 1.2.12.** Suppose that there is \( y \in \hat{J}(U_{\text{ad}}) + \mathbb{R}_\geq^k \) such that the set \( (y - \mathbb{R}_\geq^k) \cap (\hat{J}(U_{\text{ad}}) + \mathbb{R}_\geq^k) \) is compact. Then it holds \( J_{\text{opt}} \neq \emptyset \).

**Proof.** This is a slight generalization of [Ehr05, Theorem 2.10] using the argument that adding \( \mathbb{R}_\geq^k \) to the set \( \hat{J}(U_{\text{ad}}) \) does not change the Pareto front \( J_{\text{opt}} \).

Given any \( y = \hat{J}(u) \in \hat{J}(U_{\text{ad}}) \) with \( y \not\in J_{\text{opt}} \), it follows directly from the definition of Pareto optimality that there is \( \bar{y} = \hat{J}(\bar{u}) \in \hat{J}(U_{\text{ad}}) \) with \( \bar{y} \leq y \). However, even if the Pareto front \( J_{\text{opt}} \) is not empty (e.g., since the assumptions of Theorem 1.2.12 are satisfied), it is not clear that there is \( \bar{y} \in J_{\text{opt}} \) with \( \bar{y} \leq y \). If this property holds for all \( y \in \hat{J}(U_{\text{ad}}) \setminus J_{\text{opt}} \), the set \( J_{\text{opt}} \) is said to be externally stable, see, e.g., [Ehr05, SNT85].

**Definition 1.2.13.** The set \( J_{\text{opt}} \) is said to be externally stable if for every \( y \in \hat{J}(U_{\text{ad}}) \) there is \( \bar{y} \in J_{\text{opt}} \) with \( \bar{y} \leq y \). This is equivalent to \( \hat{J}(U_{\text{ad}}) \subset J_{\text{opt}} + \mathbb{R}_\geq^k \).

Especially for the investigation of suitable solution methods for solving (MOP), we are interested in guaranteeing that the Pareto front is externally stable. The next result provides a sufficient condition for this property.

**Theorem 1.2.14.** If for every \( y \in \hat{J}(U_{\text{ad}}) + \mathbb{R}_\geq^k \) the set \( (y - \mathbb{R}_\geq^k) \cap (\hat{J}(U_{\text{ad}}) + \mathbb{R}_\geq^k) \) is compact, then \( J_{\text{opt}} \) is externally stable.

**Proof.** For a proof of a similar version of this theorem, we refer to [Ehr05, Theorem 2.21].

### 1.3 Hierarchical Structure of Pareto Sets

In Lemma 1.2.10, we saw that global (local) minimizers of the individual cost functions \( \hat{J}_1, \ldots, \hat{J}_k \) are at least (locally) weakly Pareto optimal for (MOP). Based on this result, one might ask the more general question if (locally) (weakly) Pareto optimal points for subproblems of (MOP), i.e., problems only considering a subset of all \( k \) cost functions, are also (locally) (weakly) Pareto optimal for (MOP).
Definition 1.3.1. Given $I \subset \{1, \ldots, k\}$, we denote by $J^I$ the multiobjective cost function $(\hat{J}_i)_{i \in I}: U_{ad} \to \mathbb{R}^I$, and call the problem
\[
\min_{u \in U_{ad}} J^I(u)
\] (MOP(I))
a subproblem of (MOP).

To be able to speak about subproblems of (MOP) in a compact form, we need to introduce some notations.

Definition 1.3.2. For $I, K \subset \{1, \ldots, k\}$ with $K \subset I$, we make the following definitions.

(i) For every vector $y \in \mathbb{R}^I$ we denote by $y^K := (y_i)_{i \in K} \in \mathbb{R}^K$ the canonical projection to $\mathbb{R}^K$.

(ii) The set $U_{I_{opt, (w), (loc)}} := \{u \in U_{ad} | u \text{ is (locally) (weakly) Pareto optimal for (MOP(I))}\}$ denotes the (local) (weak) Pareto set and the set $J^I_{opt, (w), (loc)} := J^I(U^I_{opt, (w), (loc)}) \subset \mathbb{R}^I$ denotes the (local) (weak) Pareto front of the subproblem (MOP(I)).

(iii) The (local) (weak) nadir objective point for the subproblem (MOP(I)) is defined by
\[
y^I_{n\text{ad}, I_{opt, (w), (loc)}} := \sup\{y_i | y \in J^I_{opt, (w), (loc)}\}
\] for all $i \in I$.

As a first result, we can show that (locally) weakly Pareto optimal points for a subproblem (MOP(I)) are also (locally) weakly Pareto optimal for (MOP). In general, this is not true for (locally) Pareto optimal points. However, it is possible to prove a weaker result.

Lemma 1.3.3. Let $I \subset \{1, \ldots, k\}$ be arbitrary.

(i) For any $K \subset I$ it holds
\[
U^K_{opt, (loc)} \subset U^K_{opt, (w), (loc)} \subset U^I_{opt, (w), (loc)} \subset U_{opt, (w), (loc)}.
\]

(ii) Let $K \subset I$ be arbitrary. If $\bar{u} \in U^K_{opt, (loc)}$, then it either holds $\bar{u} \in U^I_{opt, (loc)}$, or (in every neighborhood of $\bar{u}$) there is $\tilde{u} \in U^K_{opt, (loc)}$ with $J^I(\tilde{u}) \leq J^I(\bar{u})$ and $J^K(\tilde{u}) = J^K(\bar{u})$.

Proof. Let $I \subset \{1, \ldots, k\}$ be arbitrary.

(i) Let $K \subset I$ be arbitrary. It is clear that $U^K_{opt, (loc)} \subset U^K_{opt, (w), (loc)}$ holds. Let $\bar{u} \in U^K_{opt, (w), (loc)}$ be arbitrary. Thus, there is no $\tilde{u} \in U_{ad}$ (in the corresponding neighborhood of $\bar{u}$) with $J^K(\tilde{u}) < J^K(\bar{u})$. In particular, this implies that there is no $\tilde{u} \in U_{ad}$ (in the corresponding neighborhood of $\bar{u}$) with $J^I(\tilde{u}) < J^I(\bar{u})$, so that $\bar{u} \in U^I_{opt, (loc)}$ holds. The inclusion $U^I_{opt, (loc)} \subset U_{opt, (w), (loc)}$ follows with the same argument.
(ii) Let $K \subset I$ and $\bar{u} \in \mathcal{U}^{K}_{\text{opt,(loc)}}$ be arbitrary. If $\bar{u} \notin \mathcal{U}^{I}_{\text{opt,(loc)}}$, then, by definition, there is $\tilde{u} \in \mathcal{U}_{\text{ad}}$ (in every neighborhood of $\bar{u}$) with $\hat{J}^{I}(\bar{u}) \not\preceq \hat{J}^{I}(\tilde{u})$. However, since $\bar{u} \in \mathcal{U}^{K}_{\text{opt,(loc)}}$ and $\tilde{u}$ is close enough to $\bar{u}$, it cannot hold $\hat{J}^{K}(\tilde{u}) \not\preceq \hat{J}^{K}(\bar{u})$. Therefore, we get $\hat{J}^{K}(\tilde{u}) = \hat{J}^{K}(\bar{u})$, which additionally implies $\tilde{u} \in \mathcal{U}^{K}_{\text{opt,(loc)}}$.

The following result is a rather simple conclusion of Lemma 1.3.3.

**Corollary 1.3.4.** Let $I \subset \{1, \ldots, k\}$ be arbitrary. Then it holds

$$\mathcal{U}^{I}_{\text{opt}} \subset \left( \bigcup_{K \subseteq I} \mathcal{U}^{K}_{\text{opt}} \right) \cup \left( \mathcal{U}^{I}_{\text{opt}} \setminus \bigcup_{K \subseteq I} \mathcal{U}^{K}_{\text{opt}} \right).$$

(1.3.1)

$$\mathcal{U}^{I}_{\text{opt,w}} = \left( \bigcup_{K \subseteq I} \mathcal{U}^{K}_{\text{opt,w}} \right) \cup \left( \mathcal{U}^{I}_{\text{opt,w}} \setminus \bigcup_{K \subseteq I} \mathcal{U}^{K}_{\text{opt,w}} \right).$$

(1.3.2)

**Proof.** For both statements ‘$\subset$’ is clear, while for the second statement the inclusion ‘$\supset$’ follows from Lemma 1.3.3 (i).

Given the convexity of the cost functions, it can be concluded that all weakly Pareto optimal points are actually Pareto optimal for the problem itself or for one of its subproblems. This result was first shown in [LTWW84] for multiobjective maximization problems.

**Lemma 1.3.5.** Let the cost functions $\hat{J}_{1}, \ldots, \hat{J}_{k}$ be convex and $I \subset \{1, \ldots, k\}$ be arbitrary. Then it holds

$$\mathcal{U}^{I}_{\text{opt,w}} = \bigcup_{K \subseteq I} \mathcal{U}^{K}_{\text{opt}}.$$  

(1.3.3)

**Proof.** For a proof see [LTWW84, Corollary 1].

**Corollary 1.3.6.** Let the cost functions $\hat{J}_{1}, \ldots, \hat{J}_{k}$ be convex and $I \subset \{1, \ldots, k\}$ be arbitrary. Then it holds

$$\mathcal{U}^{I}_{\text{opt,w}} \setminus \bigcup_{K \subseteq I} \mathcal{U}^{K}_{\text{opt,w}} = \mathcal{U}^{I}_{\text{opt}} \setminus \bigcup_{K \subseteq I} \mathcal{U}^{K}_{\text{opt}}.$$
for any \( I \subset \{1, \ldots, k\} \). Using this identity twice, we can conclude

\[
\bigcup_{K \subset I} U_{opt,w}^{K} \setminus \bigcup_{K \subset I} U_{opt,w}^{K} = \left( \bigcup_{K \subset I} U_{opt,w}^{K} \right) \setminus \left( \bigcup_{K \subset I} U_{opt,w}^{I} \setminus U_{opt,w}^{K} \right)
\]

which proves the statement.

\[\square\]

**Remark 1.3.7.** In the literature, an MOP for which (1.3.3) is fulfilled is called Pareto reducible, see e.g. [Pop05]. The convexity of the cost functions is only a sufficient condition for Pareto reducibility. In [MB94, Pop05], more general sufficient conditions for a problem to be Pareto reducible were shown. However, for our purposes here it is enough to know that convexity of the cost functions implies that (1.3.3) holds.

The last result of this section is concerned with the (weak) nadir objective point \( y_{nad,I,w}^{\text{nad},I,\{w\}} \). More specifically, we show that the \( i \)-th component of \( y_{nad,I,w}^{\text{nad},I,\{w\}} \) can be computed by only using solutions of the subproblem of (MOP(I)) not containing the cost function \( J_i \). This result will be important in the numerical implementation of algorithms for the Euclidean reference point method (Section 1.6) and the Pascoletti-Serafini method (Section 1.7). A similar statement was already proven in [ETP03, Theorem 4], where the general idea of the proof was taken from.

**Theorem 1.3.8.** Let \( I \subset \{1, \ldots, k\} \) be arbitrary and assume that \( \tilde{g}_{opt}^{K} \) is externally stable for all \( K \subset I \). For the (weak) nadir objective point \( y_{nad,I,w}^{\text{nad},I,\{w\}} \) it holds

\[
y_{i}^{\text{nad},I,\{w\}} = \sup \left\{ \tilde{J}_i(u) \mid u \in U_{opt,(w)}^{I} \cap U_{opt,(w)}^{J_{\{i\}}} \right\}
\]  

(1.3.4)

for all \( i \in I \), i.e., to determine the \( i \)-th component of \( y_{nad,I,w}^{\text{nad},I,\{w\}} \) it is enough to look at solutions to the subproblem of (MOP(I)) not containing \( J_i \).

**Proof.** To prove (1.3.4), let \( i \in I \) be arbitrary and assume that there is \( \varepsilon > 0 \) such that \( y_{i}^{\text{nad},I,\{w\}} - \varepsilon > \tilde{J}_i(u) \) for all \( u \in U_{opt,(w)}^{I} \cap U_{opt,(w)}^{J_{\{i\}}} \). Then, by the definition of \( y_{nad,I,w}^{\text{nad},I,\{w\}} \), there is \( \tilde{u} \in U_{opt,(w)}^{I} \) with \( \tilde{u} \notin U_{opt,(w)}^{J_{\{i\}}} \) and \( \tilde{J}_i(\tilde{u}) > \tilde{J}_i(u) \) for all \( u \in U_{opt,(w)}^{I} \cap U_{opt,(w)}^{J_{\{i\}}} \). Since \( \tilde{u} \notin U_{opt,(w)}^{J_{\{i\}}} \) and \( \tilde{J}_i(\tilde{u}) \) is externally stable by assumption, there is \( \tilde{u} \in U_{opt,(w)}^{J_{\{i\}}} \) with \( \tilde{J}_i(\tilde{u}) \leq \tilde{J}_i^{J_{\{i\}}}(\tilde{u}) \) (or \( \tilde{J}_i^{J_{\{i\}}}(\tilde{u}) < \tilde{J}_i^{J_{\{i\}}}(\tilde{u}) \), respectively). But then \( \tilde{J}_i(\tilde{u}) \leq \tilde{J}_i^{J_{\{i\}}}(\tilde{u}) \) (or \( \tilde{J}_i^{J_{\{i\}}}(\tilde{u}) < \tilde{J}_i(\tilde{u}) \), respectively), which is a contradiction to \( \tilde{u} \in U_{opt,(w)}^{I} \). Thus, we have

\[
y_{i}^{\text{nad},I,\{w\}} = \sup \left\{ \tilde{J}_i(u) \mid u \in U_{opt,(w)}^{I} \right\} = \sup \left\{ \tilde{J}_i(u) \mid u \in U_{opt,(w)}^{I} \cap U_{opt,(w)}^{J_{\{i\}}} \right\}
\]

which is what we had to show. 

\[\square\]
Remark 1.3.9. By Lemma 1.3.3, it holds

\[ U_{\text{opt},w}^I \cap U_{\text{opt},w}^{I\setminus\{i\}} = U_{\text{opt},(w)}^{I\setminus\{i\}} \]

and

\[ U_{\text{opt}}^I \cap U_{\text{opt}}^{I\setminus\{i\}} = \left\{ \bar{u} \in U_{\text{opt}}^{I\setminus\{i\}} \mid \exists \tilde{u} \in U_{\text{opt}}^{I\setminus\{i\}} : \hat{J}_{I\setminus\{i\}}(\tilde{u}) = \hat{J}_{I\setminus\{i\}}(\bar{u}) \land \hat{j}_i(\tilde{u}) < \hat{j}_i(\bar{u}) \right\}. \]

Hence, for the computation of \( y_{i\text{nad},I,(w)} \) only the set \( U_{\text{opt},(w)}^{I\setminus\{i\}} \) is required. ♦

Remark 1.3.10. Corollary 1.3.4 and Lemma 1.3.5 show that the (weak) Pareto front and the (weak) Pareto set of \((MOP)\) are contained in the set of all (weak) Pareto fronts and (weak) Pareto sets of all of its subproblems. This fact can be used to set up a hierarchical procedure for computing the (weak) Pareto front:\ Let a subproblem \((MOP(I))\) for some \( I \subset \{1, \ldots, k\} \) be given and assume that the sets \( U_{\text{opt},(w)}^K \) have already been computed for all \( K \subseteq I \). Then, in order to obtain the set \( U_{\text{opt},(w)}^I \), it is only required to compute \( U_{\text{opt},(w)}^I \setminus \bigcup_{K \subsetneq I} U_{\text{opt},(w)}^K \). The general outline of such an approach looks as follows:

(i) Solve all subproblems with one cost function, i.e., compute the global minimizers of the cost functions \( \hat{J}_1, \ldots, \hat{J}_k \).

(ii) Use this information to solve all \( \binom{k}{2} \) subproblems \((MOP(I))\) with \( I \subset \{1, \ldots, k\} \) and \(|I| = 2\).

(iii) Use the information of all subproblems with less than three cost functions to solve all \( \binom{k}{3} \) subproblems \((MOP(I))\) with \( I \subset \{1, \ldots, k\} \) and \(|I| = 3\).

\[ \vdots \]

(k) Use the information of all subproblems with less than \( k \) cost functions to solve the original problem \((MOP)\).

In the Sections 1.6 and 1.7, we will show how this general approach can be realized for the Euclidean reference point method and the Pascoletti-Serafini method, respectively, to solve MOPs with an arbitrary number of cost functions. Note that the idea of solving subproblems to build up the Pareto front hierarchically is not new. This approach was first applied to the Pascoletti-Serafini method in [MGGS09] and [MAL12], and later adapted and improved in [BKR17, DK19, KSd15]. At this point, we refer to Section 1.7.2, where the existing literature on this topic is reviewed in more detail. Let us additionally mention that such a hierarchical approach is also used in [GPD19] to investigate the hierarchical structure of the Pareto critical set. ♦

1.4 General Scalarization Methods

In the previous sections, we introduced and investigated the notion of Pareto optimality for \((MOP)\). However, it is still unclear how Pareto optimal points can be computed. One of the most used approaches in the literature is the scalarization method, see, e.g.,
1. Multiobjective Optimization

[Ehr05, Jah11, Mie99, Wie86]. In this section, we present the idea of a general scalarization method, before we deal with three concrete methods in the Sections 1.5, 1.6 and 1.7.

The fundamental concept of scalarization methods is to transform (MOP) into a scalar optimization problem, which can then be solved by using well-known techniques from scalar optimization. This idea is realized by first choosing a scalarization function \( g: \mathbb{R}^k \to \mathbb{R} \) and then minimizing the concatenation \( g \circ \hat{J}: \mathcal{U}_{ad} \to \mathbb{R} \). Under certain properties of the function \( g \), one can ensure that any minimizer of the scalarized problem \( g \circ \hat{J} \) is in fact (locally) (weakly) Pareto optimal. On the other hand, it is also possible to show that for any (locally) (weakly) Pareto optimal point there is a scalarization function \( g \), such that this point is a minimizer of \( g \circ \hat{J} \).

**Definition 1.4.1.** Let \( g: \mathbb{R}^k \to \mathbb{R} \) be a scalarization function. Then we define the scalarized function \( \hat{J}^g := g \circ \hat{J} \) and call the scalar-valued minimization problem

\[
\min_{u \in \mathcal{U}_{ad}} \hat{J}^g(u)
\]

(SOP)

a scalarized problem of (MOP).

The property of monotonicity of the scalarization functions \( g \) turns out to be the key to ensure Pareto optimality of any solution of (SOP).

**Definition 1.4.2.** Let \( X \subset \mathbb{R}^k \) be arbitrary. We call a function \( g: \mathbb{R}^k \to \mathbb{R} \)

(i) monotonically increasing on \( X \) if for all \( x, y \in X \) with \( x \leq y \) it holds \( g(x) \leq g(y) \).

(ii) strictly monotonically increasing on \( X \) if for all \( x, y \in X \) with \( x < y \) it holds \( g(x) < g(y) \).

(iii) strongly monotonically increasing on \( X \) if for all \( x, y \in X \) with \( x \preceq y \) it holds \( g(x) < g(y) \).

Now we can give the following sufficient conditions for a solution of a scalarized problem to be (locally) (weakly) Pareto optimal, see, e.g., [Ehr05, Theorem 4.20], [Jah11, Lemmata 5.14 & 5.24].

**Lemma 1.4.3.** Let \( g: \mathbb{R}^k \to \mathbb{R} \) be a scalarization function.

(i) If \( g \) is monotonically increasing on \( \hat{J}(\mathcal{U}_{ad}) \) and \( \bar{u} \in \mathcal{U}_{ad} \) is the unique global (local) solution of (SOP), then \( \bar{u} \) is (locally) Pareto optimal.

(ii) If \( g \) is strictly monotonically increasing on \( \hat{J}(\mathcal{U}_{ad}) \), then any global (local) solution of (SOP) is (locally) weakly Pareto optimal.

(iii) If \( g \) is strongly monotonically increasing on \( \hat{J}(\mathcal{U}_{ad}) \), then any global (local) solution of (SOP) is (locally) Pareto optimal.

**Proof.** For a proof of these statements, we refer to [Bee19, Lemma 3.13].
On the other hand, for any (locally) weakly Pareto optimal point, it is also possible to find a scalarization function $g$ such that this point is a global (local) solution of the scalarized problem (SOP).

**Lemma 1.4.4.** Let $\bar{u} \in \mathcal{U}_{opt,(w),(loc)}$ be arbitrary. Then $\bar{u}$ is a global (local) solution of (SOP) for the scalarization function

$$g: \mathbb{R}^k \to \mathbb{R}, \quad g(x) := \max_{i \in \{1, \ldots, k\}} (x_i - z_i),$$

where $z := \hat{J}(\bar{u}) - (1, \ldots, 1)^T \in \mathbb{R}^k$.

**Proof.** Let $\bar{u} \in \mathcal{U}_{ad}$ be (locally) weakly Pareto optimal, and define the scalarization function $g$ as well as the reference point $z$ as above. If we assume that $\bar{u}$ is no global (local) solution of (SOP), there is $\tilde{u} \in \mathcal{U}_{ad}$ (in every neighborhood of $\bar{u}$) with

$$\max_{i \in \{1, \ldots, k\}} \left( \hat{J}_i(\tilde{u}) - \hat{J}_i(\bar{u}) + 1 \right) < \max_{i \in \{1, \ldots, k\}} \left( \hat{J}_i(\bar{u}) - \hat{J}_i(\bar{u}) + 1 \right) = 1,$$

from which $\max_{i \in \{1, \ldots, k\}} \left( \hat{J}_i(\tilde{u}) - \hat{J}_i(\bar{u}) \right) < 0$ can be concluded. However, this implies $\hat{J}(\tilde{u}) < \hat{J}(\bar{u})$, which is a contradiction to $\bar{u}$ being (locally) weakly Pareto optimal. Thus, $\bar{u}$ is a global (local) solution of (SOP) for the scalarization function $g$. 

**Remark 1.4.5.** In practice, the result of Lemma 1.4.4 is not very useful, since we would have to know the objective point $\hat{J}(\bar{u})$ in order to define the scalarization function $g$. ⊢

### 1.4.1 Solving a Multiobjective Optimization Problem

We saw that by using a scalarization function $g$ with certain monotonicity properties, we can transform (MOP) into a scalar-valued optimization problem (SOP), whose solutions are (locally) weakly Pareto optimal. However, in the end, our goal is to compute the entire (local) (weak) Pareto set or the entire (local) (weak) Pareto front, respectively, and not only individual Pareto optimal points. This means that we want to find an index set $\mathcal{I}$ and a family of scalarization functions $(g_i)_{i \in \mathcal{I}}$ such that the following criteria are met.

(i) **Pareto admissibility:** For every $i \in \mathcal{I}$ the scalarized problem (SOP) with scalarization function $g_i$ has a global (local) solution, and every global (local) solution is a (locally) (weakly) Pareto optimal point of (MOP).

(ii) **Pareto sufficiency:** For every (locally) (weakly) Pareto optimal point there is $i \in \mathcal{I}$ such that this point is a global (local) solution of (SOP) for the scalarization function $g_i$.

(iii) **Efficiency:** The family of scalarization functions $(g_i)_{i \in \mathcal{I}}$ is efficient. In the best case, the mapping $\mathcal{I} \to \mathcal{U}_{opt,(w),(loc)}, \ i \mapsto \bar{u}$ (or the mapping $\mathcal{I} \to \mathcal{J}_{opt,(w),(loc)}, \ i \mapsto \hat{J}(\bar{u})$), where $\bar{u}$ is the global (local) solution of (SOP) with scalarization function $g_i$, is bijective. Depending on the problem, this is not always possible, since (SOP) might not have a unique solution.
In the Sections 1.5, 1.6 and 1.7, we present three different types of scalarization functions – the weighted-sum scalarization and the Euclidean reference point scalarization, which are both applicable to convex MOPs, and the Pascoletti-Serafini scalarization, which can be applied to more general, and in particular non-convex, problems – and show how the index set \( I \) can be chosen to guarantee the above mentioned properties in all three cases. While the choice of the index set \( I \) will turn out to be straight-forward for the weighted-sum method, it is a bit more delicate for the Euclidean reference point method and the Pascoletti-Serafini method, respectively. In fact, in the latter two cases the index set \( I \) is problem dependent and, to the best of the author’s knowledge, it is not possible to determine the entire index set before any calculations (at least if we want to have an efficient index set in the sense of (iii) for the Pascoletti-Serafini method). However, we can show that the index sets can be built up sequentially in a hierarchical way by using subproblems of \( \text{MOP} \) as introduced in Section 1.3.

1.4.2 Quality Criteria for a Numerical Implementation

Before we introduce the three different scalarization methods, let us briefly discuss which quality criteria a numerical implementation of a scalarization method should satisfy. In general, both the (local) (weak) Pareto set/front are of infinite cardinality, so that we can only hope to approximate them numerically. Instead of computing \( U_{\text{opt},(w)\text{,loc}} \), \( J_{\text{opt},(w)\text{,loc}} \), we only obtain finite subsets

\[
U_{\text{num, opt},(w)\text{,loc}} = \{ \bar{u}^1, \ldots, \bar{u}^N \} \quad \text{and} \quad J_{\text{num, opt},(w)\text{,loc}} = \{ \hat{J}(\bar{u}^1), \ldots, \hat{J}(\bar{u}^N) \},
\]

respectively. When using a scalarization method to do so, this can be achieved by choosing a finite index set \( I_{\text{num}} \) and then solving the scalarized problems (SOP) for all scalarization functions \( (g_i)_{i \in I_{\text{num}}} \).

In [Say00], three different quality criteria were introduced.

(i) **Coverage:** All parts of the Pareto set/front should be represented in the finite sets \( U_{\text{num, opt},(w)\text{,loc}} \) and \( J_{\text{num, opt},(w)\text{,loc}} \), respectively. This can be measured by

\[
\text{COV}(J_{\text{opt},(w)\text{,loc}}) := \max_{\bar{y} \in J_{\text{opt},(w)\text{,loc}}} \min_{y \in J_{\text{num, opt},(w)\text{,loc}}} \| \bar{y} - y \|
\]

for any norm \( \| \cdot \| \) on \( \mathbb{R}^k \). Unless stated otherwise, we will use the Euclidean norm to computed this value. However, in many practical applications, the exact (local) (weak) Pareto front \( J_{\text{opt},(w)\text{,loc}} \) is not known. Thus, given a very fine approximation \( J_{\text{num, fine},(w)\text{,loc}} \) of \( J_{\text{opt},(w)\text{,loc}} \), the coverage can be computed by the formula

\[
\text{COV}_{\text{num}}(J_{\text{opt},(w)\text{,loc}}) := \max_{\bar{y} \in J_{\text{num, fine},(w)\text{,loc}}} \min_{y \in J_{\text{num},(w)\text{,loc}}} \| \bar{y} - y \| \quad (1.4.1)
\]

in practice.
1.5 The Weighted-Sum Method

(ii) **Uniformity:** The points on the Pareto set/front should be distributed (almost) equidistantly. This might be quantified by the ratio

\[
\frac{\max_{\bar{y} \in \mathcal{J}_{\text{opt.(w)},(\text{loc})}} \min_{\bar{y} \neq y \in \mathcal{J}_{\text{opt.(w)},(\text{loc})}} \| \bar{y} - y \|}{\min_{\bar{y} \in \mathcal{J}_{\text{opt.(w)},(\text{loc})}} \min_{\bar{y} \neq y \in \mathcal{J}_{\text{opt.(w)},(\text{loc})}} \| \bar{y} - y \|} \geq 1.
\]

In the best case, this ratio is close to 1. In practice, it might happen that two discretization points are very close to each other, whereas all the other points are uniformly distributed. However, this still leads to a large ratio. To this end, we choose the ratio

\[
\text{uni}^{\text{num}}(\mathcal{J}_{\text{opt.(w)},(\text{loc})}) := \frac{\max_{\bar{y} \in \mathcal{J}_{\text{opt.(w)},(\text{loc})}} \min_{\bar{y} \neq y \in \mathcal{J}_{\text{opt.(w)},(\text{loc})}} \| \bar{y} - y \|}{\text{avg}_{\bar{y} \in \mathcal{J}_{\text{opt.(w)},(\text{loc})}} \min_{\bar{y} \neq y \in \mathcal{J}_{\text{opt.(w)},(\text{loc})}} \| \bar{y} - y \|},
\]

where \( \text{avg} \) computes the average, as a measure for uniformity. This value is less sensitive to the behavior of single points.

(iii) **Cardinality:** The approximation should contain a reasonable number of elements (depending on the number and the extent of the cost functions). Of course, this is directly connected to the coverage and uniformity of the approximation as well.

So for a numerical implementation of a scalarization method, the focus lies on the question how to guarantee the above mentioned criteria by choosing the index set \( \mathcal{I}^{\text{num}} \) appropriately. This question will be discussed for the Euclidean reference point method and the Pascoletti-Serafini method. For the weighted-sum method, we will only describe that a naive discretization of the index set \( \mathcal{I} \) does in many cases not lead to a satisfying approximation of the Pareto set/front with respect to the quality criteria from above.

1.5 The Weighted-Sum Method

The best known scalarization method in the literature is the weighted-sum method (WSM). It dates back to the 1960s, where it was introduced in works by Zadeh ([Zad63]) and Geoffrion ([Geo68]). Later on, it was not only applied to various fields of multiobjective optimization, as for example multiobjective truss optimization ([Kos88, KS87]), multiobjective structural optimization ([PSQX01]) and more recently PDE-constrained multiobjective optimization ([IUV17]), but also further studied and enhanced from a mathematical point of view (see, e.g., [Coh04, KdW05, Lin76, MA10]).

A good overview of this method can also be found in [Ehr05, Chapter 3] and [Mie99, pp. 78–85].

In this thesis, we will not give any new insights into the WSM, but only summarize and rephrase existing results from the literature tailored for our purposes. The main goal of this section is to lay the foundation for the Euclidean reference point method in Section 1.6, for which its connection to the WSM will turn out to be crucial.

Note that this section was already published in a similar form in the preprint [BV19], to which the author made substantial contributions.
1. Multiobjective Optimization

1.5.1 Problem Formulation

The fundamental idea of the WSM is to choose a non-negative weight for every objective function and then to minimize the sum of the weighted objective functions.

**Definition 1.5.1.** Let \( \alpha \in \mathbb{R}_+^k \) be arbitrary. Then we define the scalarization function

\[
g_\alpha : \mathbb{R}^k \to \mathbb{R}, \quad x \mapsto \sum_{i=1}^k \alpha_i x_i.
\]

Using this scalarization function in (SOP) leads to the so-called weighted-sum scalarized function \( \hat{J}_{g_\alpha} := \sum_{i=1}^k \alpha_i \hat{J}_i \) and the weighted-sum problem (WSP) with weight \( \alpha \)

\[
\min_{u \in \mathcal{U}_{ad}} \hat{J}_{g_\alpha}(u) = \min_{u \in \mathcal{U}_{ad}} \sum_{i=1}^k \alpha_i \hat{J}_i(u).
\]

(WSP(\( \alpha \)))

We can show that for every weight \( \alpha \in \mathbb{R}_+^k \) any global (local) solution of (WSP(\( \alpha \))) is at least (locally) weakly Pareto optimal.

**Lemma 1.5.2.** Let \( \alpha \in \mathbb{R}_+^k \) be arbitrary and suppose that \( \bar{u} \in \mathcal{U}_{ad} \) is a global (local) solution of (WSP(\( \alpha \))). Then it holds \( \bar{u} \in \mathcal{U}_{\text{opt},w,(\text{loc})} \). If it additionally holds \( \alpha > 0 \) or \( \bar{u} \) is the unique (local) solution (in a neighborhood of \( \bar{u} \)) of (WSP(\( \alpha \))), then \( \bar{u} \in \mathcal{U}_{\text{opt},(\text{loc})} \).

**Proof.** Since the scalarization function \( g_\alpha \) is strictly monotonically increasing for every \( \alpha \in \mathbb{R}_+^k \), and even strongly monotonically increasing if \( \alpha > 0 \) holds, the statement follows directly from Lemma 1.4.3. \( \square \)

However, the reverse statement does not hold true in general (i.e., there are Pareto optimal points which are no solution to any WSP), but only under some additional convexity assumptions. This is one major drawback of the WSM and was already noted by many
authors (see, e.g., [AP96, Kos85, SD93]). The crucial property required to show the reverse statement is the convexity of the set \( \hat{J}(U_{ad}) + \mathbb{R}^k_{\geq} \). The following lemma shows that this set is convex if all cost functions are convex.

**Lemma 1.5.3.** Let the cost functions \( \hat{J}_1, \ldots, \hat{J}_k \) be convex. Then the set \( \hat{J}(U_{ad}) + \mathbb{R}^k_{\geq} \) is convex.

**Proof.** This statement was already shown in [Ban17, Lemma 3.19].

Now we can show that under the assumption that all cost functions are convex, the reverse statement of Lemma 1.5.2 holds true. The proof can be found in various publications in the literature, see, e.g., [Zad63], [Ehr05, Theorem 3.5], [Jah11, Theorem 5.4]. It uses the fact that the set \( \hat{J}(U_{ad}) + \mathbb{R}^k_{\geq} \) is convex (cf. Lemma 1.5.3) together with a hyperplane separation result (see, e.g., [Jah11, Theorem 3.14]).

**Theorem 1.5.4.** Let the cost functions \( \hat{J}_1, \ldots, \hat{J}_k \) be convex. If \( \bar{u} \in U_{opt,w} \), then there is \( \alpha \in \mathbb{R}^k_{\geq} \) such that \( \bar{u} \) is a global solution of \( (WSP(\alpha)) \).

**Proof.** For the proof of this statement, we refer to [Ehr05, Theorem 3.5] or [Jah11, Theorem 5.4].

**Remark 1.5.5.** (i) Since we assume that the cost functions \( \hat{J}_1, \ldots, \hat{J}_k \) are convex in Theorem 1.5.4, Lemma 1.2.7 implies that it holds \( U_{opt,(w),loc} = U_{opt,(w)} \). Thus, we do not need to distinguish between global and local Pareto optimal points in this case.

(ii) Since \( U_{opt} \subset U_{opt,w} \), the statement of Theorem 1.5.4 is in particular true for all Pareto optimal points. However, the reverse statement of Lemma 1.5.2 does in general not hold true in this case: If \( \bar{u} \in U_{opt} \), there does not need to be \( \alpha \in \mathbb{R}^k_{\geq} \) such that \( \bar{u} \) is a global solution of \( (WSP(\alpha)) \).

(iii) By scaling, we can see that every weakly Pareto optimal point solves \( (WSP(\alpha)) \) for a weight

\[
\alpha \in \Delta_k := \{ \beta \in \mathbb{R}^k_{\geq} \mid \sum_{i=1}^k \beta_i = 1 \}.
\]

Thus, in the terminology of Section 1.4.1, we can consider the family of scalarization functions \( (g_\alpha)_{\alpha \in \Delta_k} \). For convenience, we define

\[
\Delta_\geq_k := \{ \beta \in \mathbb{R}^k_{\geq} \mid \sum_{i=1}^k \beta_i = 1 \},
\]

\[
\partial \Delta_k := \{ \beta \in \mathbb{R}^k \mid \exists i \in \{1, \ldots, k\} : \beta_i = 0 \},
\]

and see that \( \partial \Delta_k \) is indeed the boundary of \( \Delta_k \) with respect to the relative topology on \( \{ \beta \in \mathbb{R}^k \mid \sum_{i=1}^k \beta_i = 1 \} \).
The previous results show that the WSM is in general only suitable for MOPs with convex cost functions. Therefore, we will restrict ourselves to this case in the following and introduce the following assumptions, which are common in convex optimization. Note that we already assume in the entire chapter that $\mathcal{U}_{\text{ad}} \subset \mathcal{U}$ is non-empty, convex and closed.

**Assumption 1.1.** The cost functions $\hat{J}_1, \ldots, \hat{J}_k$ are convex, lower semi-continuous and bounded from below. Additionally, in case that $\mathcal{U}_{\text{ad}}$ is unbounded, we assume that

$$\lim_{\|u\| \to \infty} \hat{J}_i(u) = \infty \quad \text{for all } i \in \{1, \ldots, k\}.$$

Under these assumptions, it is possible to show that $(\text{WSP}(\alpha))$ actually has a global solution.

**Theorem 1.5.6.** Let Assumption 1.1 be satisfied and $\alpha \in \Delta_k$ be arbitrary. Then $(\text{WSP}(\alpha))$ has a global solution. If there is $i \in \{1, \ldots, k\}$ such that $\hat{J}_i$ is strictly convex and $\alpha_i > 0$ holds, the solution is unique.

**Proof.** The statement can be shown with standard arguments in convex optimization, see, e.g., [ET99, Proposition II 1.2]. For a detailed proof, we refer to [Bee19, Theorem 3.20].

Under additional differentiability assumptions, we can show the following necessary and sufficient first-order optimality condition for a solution of $(\text{WSP}(\alpha))$.

**Lemma 1.5.7.** Let Assumption 1.1 be satisfied, $\alpha \in \Delta_k$ be arbitrary and assume additionally that all cost functions are Fréchet differentiable. Then a necessary and sufficient first-order optimality condition for a solution $\bar{u} \in \mathcal{U}_{\text{ad}}$ of $(\text{WSP}(\alpha))$ is given by

$$\left\langle \sum_{i=1}^{k} \alpha_i \nabla \hat{J}_i(\bar{u}), u - \bar{u} \right\rangle \geq 0 \quad \text{for all } u \in \mathcal{U}_{\text{ad}}. \quad (1.5.1)$$

**Proof.** Note that it holds

$$\nabla \hat{J}^g(u) = \sum_{i=1}^{k} \alpha_i \nabla \hat{J}_i(u)$$

for all $u \in \mathcal{U}_{\text{ad}}$. Now the statement follows, e.g., from [Trö10, Lemma 2.21].

Moreover, we can show that the Pareto front $\hat{J}_{\text{opt}}$ is externally stable under Assumption 1.1. To this end, we first prove that the set $\hat{J}(\mathcal{U}_{\text{ad}}) + \mathbb{R}^k$ is closed under these assumptions.

**Lemma 1.5.8.** Let Assumption 1.1 be satisfied. Then the set $\hat{J}(\mathcal{U}_{\text{ad}}) + \mathbb{R}^k$ is closed.
Theorem 1.5.9. Let Assumption 1.1 be satisfied. Then the Pareto front $\mathcal{J}_{\text{opt}}$ is externally stable.

Proof. We want to prove this statement by using Theorem 1.2.14. Thus, we need to show that the set $(y - R^k_y) \cap (\hat{\mathcal{J}}(U_{\text{ad}}) + R^k_y)$ is compact for every $y \in \hat{\mathcal{J}}(U_{\text{ad}}) + R^k_y$. Let $y \in \hat{\mathcal{J}}(U_{\text{ad}}) + R^k_y$ be arbitrary. By using Assumption 1.1, one can show that the set $(y - R^k_y) \cap (\hat{\mathcal{J}}(U_{\text{ad}}) + R^k_y)$ is bounded. By Lemma 1.5.8, the set $\hat{\mathcal{J}}(U_{\text{ad}}) + R^k_y$ is closed, and of course, the set $(y - R^k_y)$ is closed as well, so that the closedness of $(y - R^k_y) \cap (\hat{\mathcal{J}}(U_{\text{ad}}) + R^k_y)$ follows directly. Thus, the set $(y - R^k_y) \cap (\hat{\mathcal{J}}(U_{\text{ad}}) + R^k_y)$ is compact for all $y \in \hat{\mathcal{J}}(U_{\text{ad}}) + R^k_y$, which is what we had to show.

By using the external stability of the Pareto front $\mathcal{J}_{\text{opt}}$, it can be directly concluded that for every $\alpha \in \Delta_k$, there is a global, Pareto optimal solution of $\text{(WSP}(\alpha))$.

Corollary 1.5.10. Let Assumption 1.1 be satisfied and $\alpha \in \Delta_k$ be arbitrary. Then there is a solution $\bar{u}$ of $\text{(WSP}(\alpha))$ with $\bar{u} \in U_{\text{opt}}$.

Proof. Let $\alpha \in \Delta_k$ be arbitrary. According to Theorem 1.5.6, there is a solution $\tilde{u}$ of $\text{(WSP}(\alpha))$. In Theorem 1.5.9, we showed that the Pareto front $\mathcal{J}_{\text{opt}}$ is externally stable. Thus, there is $\bar{u} \in U_{\text{opt}}$ with $\hat{\mathcal{J}}(\bar{u}) \leq \hat{\mathcal{J}}(\tilde{u})$. From this, we get directly that $\bar{u}$ is also a solution of $\text{(WSP}(\alpha))$.

The strict convexity of the cost functions implies the uniqueness of the solution of $\text{(WSP}(\alpha))$.

Corollary 1.5.11. Let Assumption 1.1 be satisfied, $\alpha \in \Delta_k$ be arbitrary and assume additionally that all cost functions are strictly convex. Then the unique solution $\bar{u} \in U_{\text{ad}}$ of $\text{(WSP}(\alpha))$ fulfills $\bar{u} \in U_{\text{opt}}$.

Proof. This statement follows directly from Theorem 1.5.6 and Lemma 1.5.2.

We can summarize the results about the WSM by introducing the following (set-valued) solution mappings, which will also be needed in Section 1.6.

Definition 1.5.12. Let Assumption 1.1 be satisfied. Then we can define the (set-valued) solution mappings

\begin{align*}
\mathcal{W}_{\text{opt},w} \colon \Delta_k &\Rightarrow U_{\text{opt},w}, \quad \alpha \mapsto \{ u \in U_{\text{ad}} \mid u \text{ is a solution of } \text{(WSP}(\alpha)) \}, \\
\mathcal{W}_{\text{opt}} \colon \Delta_k &\Rightarrow U_{\text{opt}}, \quad \alpha \mapsto \mathcal{W}_{\text{opt},w}(\alpha) \cap U_{\text{opt}}, \\
\mathcal{F}_{\text{opt},(w)} \colon \Delta_k &\Rightarrow \mathcal{J}_{\text{opt},(w)}, \quad \alpha \mapsto \hat{\mathcal{J}}(\mathcal{W}_{\text{opt},(w)}(\alpha)).
\end{align*}
For these mappings, we can show the following properties.

**Theorem 1.5.13.** If Assumption 1.1 holds, then the mappings $W_{\text{opt}(w)}$ and $F_{\text{opt}(w)}$ are well-defined. For every $\alpha \in \Delta_k$ we have

$$W_{\text{opt}(w)}(\alpha) \neq \emptyset, \quad F_{\text{opt}(w)}(\alpha) \neq \emptyset.$$  

(1.5.2)

Moreover, it holds

$$W_{\text{opt}(w)}(\Delta_k) = U_{\text{opt}(w)}, \quad F_{\text{opt}(w)}(\Delta_k) = J_{\text{opt}(w)},$$

(1.5.3)

and

$$W_{\text{opt}(w)}|_{\Delta^>_{\tilde{k}}} = W_{\text{opt}}|_{\Delta^>_{\tilde{k}}}, \quad F_{\text{opt}(w)}|_{\Delta^>_{\tilde{k}}} = F_{\text{opt}}|_{\Delta^>_{\tilde{k}}}.$$  

(1.5.4)

If all the cost functions $\tilde{J}_1, \ldots, \tilde{J}_k$ are strictly convex, the mappings $W_{\text{opt}(w)} = W_{\text{opt}}$ and $F_{\text{opt}(w)} = F_{\text{opt}}$ are single-valued.

**Proof.** The well-definedness of the mappings follows directly from Lemma 1.5.2. The statements in (1.5.2) are directly implied by Theorem 1.5.6 and Corollary 1.5.10. The identities in (1.5.3) can be concluded from Theorem 1.5.4, and (1.5.4) is a direct consequence of the last statement of Lemma 1.5.2. Finally, Corollary 1.5.11 implies that $W_{\text{opt}(w)} = W_{\text{opt}}$ and $F_{\text{opt}(w)} = F_{\text{opt}}$ are single-valued if the cost functions $\tilde{J}_1, \ldots, \tilde{J}_k$ are additionally strictly convex.

In Theorem 1.5.13, it was shown that the mappings $W_{\text{opt}(w)} = W_{\text{opt}}$ and $F_{\text{opt}(w)} = F_{\text{opt}}$ are single-valued if the cost functions are strictly convex. In this case, the mappings are treated as functions. Under even stronger assumptions, we can then show the continuity of the functions $W_{\text{opt}}$ and $F_{\text{opt}}$.

**Theorem 1.5.14.** Let Assumption 1.1 be satisfied and assume additionally that the cost functions $\tilde{J}_1, \ldots, \tilde{J}_k$ are strongly convex and twice continuously Fréchet differentiable. Then the functions $W_{\text{opt}}$ and $F_{\text{opt}}$ are continuous.

**Proof.** The continuity of $W_{\text{opt}}$ under the given assumptions follows from [Ban17, Theorem 3.22]. The continuity of $F_{\text{opt}}$ can then be concluded from $F_{\text{opt}} = \tilde{J} \circ W_{\text{opt}}$ and the continuity of $W_{\text{opt}}$ and $\tilde{J}$.

**Remark 1.5.15.** From a theoretical point of view, the WSM seems to be the natural choice for solving convex MOPs satisfying Assumption 1.1. Firstly, using the terminology of Section 1.4.1, we have shown that the WSM together with the index set $\Delta_k$ is Pareto admissible and Pareto sufficient. Secondly, it is also easy to understand and to implement, since the set $\Delta_k$ can be discretized before any computations for an arbitrary number of cost functions.

However, as it turns out, the method often fails to produce a satisfying result in the practical implementation. In fact, an equidistant distribution of the weights in $\Delta_k$ does often not
1.5 The Weighted-Sum Method

lead to an even spread of the corresponding Pareto optimal points on the Pareto front, i.e., the coverage of the method is not good. Quite on the contrary, in many cases a clustering behavior can be observed, which implies a poor uniformity of the method as well. In [DD97], the reason for this practical deficiency was investigated in detail.

Several strategies have been proposed to overcome these weaknesses. Most of them are based on an adaptive choice of the weights and try additionally to overcome the drawback of not being able to deal with non-convex MOPs (however, only on a heuristical level without any theoretical guarantees), see, e.g., [KdW05, RKW09]. Here we will not further elaborate on these strategies, but rather present the Euclidean reference point method in Section 1.6, which is also especially suitable for convex MOPs.

1.5.2 The Weighted-Sum Method for Subproblems

The relative importance of cost functions \( \hat{J}_1, \ldots, \hat{J}_k \) in the problem \( \text{WSP}(\alpha) \) is reflected by the weight vector \( \alpha \). In particular, a weight \( \alpha_i \) being zero indicates that the corresponding cost function \( \hat{J}_i \) is neglected in \( \text{WSP}(\alpha) \). Consequently, if not all components of the weight \( \alpha \) are strictly positive, we can interpret \( \text{WSP}(\alpha) \) as a WSP for a subproblem of \( \text{MOP} \). The goal of this section is to introduce a suitable notation for this situation.

**Definition 1.5.16.** Let \( I \subset \{1, \ldots, k\} \) and \( \alpha \in \mathbb{R}^I_\geq \) be arbitrary. Then the weighted-sum problem for \( \text{MOP}(I) \) with weight \( \alpha \) reads

\[
\min_{u \in \mathcal{U}_{ad}} \sum_{i \in I} \alpha_i \hat{J}_i(u). \quad \text{(WSP}(I, \alpha))
\]

The following statement follows directly from the intuition that we gave in the introduction to this section.

**Lemma 1.5.17.** Let \( I, K \subset \{1, \ldots, k\} \) with \( K \subsetneq I \) and \( \alpha \in \mathbb{R}^I_\geq \) with \( \alpha^{I \setminus K} = 0 \) be arbitrary. Then the two problems \( \text{WSP}(K, \alpha^K) \) and \( \text{WSP}(I, \alpha) \) are equivalent.

**Proof.** The statement follows directly from the equality of the weighted-sum cost functions \( \sum_{i \in K} \alpha_i \hat{J}_i = \sum_{i \in I} \alpha_i \hat{J}_i \).

**Definition 1.5.18.** Let \( I \subset \{1, \ldots, k\} \) be arbitrary. Then we define the sets

\[
\Delta_I := \{ \alpha \in \mathbb{R}^I_\geq \mid \sum_{i \in I} \alpha_i = 1 \}, \\
\Delta^>_I := \{ \alpha \in \mathbb{R}^I_\geq \mid \sum_{i \in I} \alpha_i = 1 \}, \\
\partial \Delta_I := \{ \alpha \in \Delta_I \mid \exists i \in I: \alpha_i = 0 \},
\]

where \( \partial \Delta_I \) is the boundary of \( \Delta_I \) in the relative topology on \( \{ \alpha \in \mathbb{R}^I \mid \sum_{i \in I} \alpha_i = 1 \} \).
Using the same arguments as in the previous section, we can also introduce the solution mappings for WSPs for subproblems (MOP(I)).

**Definition 1.5.19.** Let Assumption 1.1 be satisfied and \( I \subset \{1, \ldots, k\} \) be arbitrary. Then we can define the (set-valued) solution mappings

\[
W^I_{\text{opt, w}} : \Delta_I \Rightarrow \mathbb{U}^I_{\text{opt, w}}, \quad \alpha \mapsto \{ u \in \mathbb{U}_{\text{ad}} | u \text{ is a solution of } \text{(WSP}(I, \alpha)) \},
\]

\[
W^I_{\text{opt}} : \Delta_I \Rightarrow \mathbb{U}^I_{\text{opt}}, \quad \alpha \mapsto W^I_{\text{opt, w}}(\alpha) \cap \mathbb{U}^I_{\text{opt}},
\]

\[
F^I_{\text{opt, (w)}} : \Delta_I \Rightarrow \mathbb{J}^I_{\text{opt, (w)}}, \quad \alpha \mapsto \mathbb{J}^I_{\text{opt, (w)}}(W^I_{\text{opt, w}}(\alpha)).
\]

For these mappings, we can show the following properties.

**Theorem 1.5.20.** Let Assumption 1.1 be satisfied and \( I \subset \{1, \ldots, k\} \) be arbitrary. Then the mappings \( W^I_{\text{opt, w}} \) and \( F^I_{\text{opt, (w)}} \) are well-defined. For every \( \alpha \in \Delta_I \) we have

\[
W^I_{\text{opt, w}}(\alpha) \neq \emptyset, \quad F^I_{\text{opt, (w)}}(\alpha) \neq \emptyset.
\]

Moreover, the identities

\[
W^I_{\text{opt, w}}(\Delta_I) = \mathbb{U}^I_{\text{opt, w}}, \quad F^I_{\text{opt, (w)}}(\Delta_I) = \mathbb{J}^I_{\text{opt, (w)}},
\]

and

\[
W^I_{\text{opt, w}}|_{\partial \Delta_I} = W^I_{\text{opt}}|_{\Delta_I} , \quad F^I_{\text{opt, w}}|_{\Delta_I} = F^I_{\text{opt}}|_{\Delta_I}
\]

hold. If all the cost functions \( (\mathbb{J}_i)_{i \in I} \) are additionally strictly convex, the mappings \( W^I_{\text{opt, w}} = W^I_{\text{opt}} \) and \( F^I_{\text{opt, w}} = F^I_{\text{opt}} \) are single-valued.

**Proof.** The proof follows by adapting the proof of Theorem 1.5.13 to the notation of subproblems. \( \square \)

**Corollary 1.5.21.** Let Assumption 1.1 be satisfied and \( I, K \subset \{1, \ldots, k\} \) with \( K \subset I \) be arbitrary. Then we have

\[
W^K_{\text{opt, w}}(\{ \alpha \in \Delta_I | \alpha_i = 0 \text{ for all } i \in I \setminus K \}) = W^K_{\text{opt, w}}(\Delta_K) = \mathbb{U}^K_{\text{opt, w}}. \tag{1.5.5}
\]

In particular, the equality

\[
W^I_{\text{opt, w}}(\partial \Delta_I) = \bigcup_{K \subset I} W^K_{\text{opt, w}}(\Delta_K). \tag{1.5.6}
\]

holds. Moreover, we have the relations

\[
W^I_{\text{opt, w}}(\Delta_I) = W^I_{\text{opt}}(\Delta_I) \supset \mathbb{U}^I_{\text{opt}} \setminus \bigcup_{K \subset I} \mathbb{U}^K_{\text{opt, w}} = \mathbb{U}^I_{\text{opt, w}} \setminus \bigcup_{K \subset I} \mathbb{U}^K_{\text{opt, w}}. \tag{1.5.7}
\]
Proof. The identity (1.5.5) follows directly from Lemma 1.5.17 and Theorem 1.5.20. Moreover, it directly implies (1.5.6).

In (1.5.7), the identities
\[
W^I_{\text{opt}, w}(\Delta^>_I) = W^I_{\text{opt}}(\Delta^>_I) \quad \text{and} \quad U^I_{\text{opt}} \setminus \bigcup_{K \subseteq I} U^K_{\text{opt}} = U^I_{\text{opt}, w} \setminus \bigcup_{K \subseteq I} U^K_{\text{opt}, w}
\]
follow from Theorem 1.5.20 and Corollary 1.3.6, respectively. Now let \( \bar{u} \in U^I_{\text{opt}} \setminus \bigcup_{K \subseteq I} U^K_{\text{opt}} \) be arbitrary. Then, by Theorem 1.5.20, there is \( \alpha \in \Delta_I \) with \( \bar{u} \in W^I_{\text{opt}, w}(\alpha) \). If we assume that \( K := \{ i \in I \mid \alpha_i > 0 \} \subseteq I \), then we have \( \bar{u} \in W^K_{\text{opt}}(\alpha^K) \subseteq U^K_{\text{opt}} \) by Lemma 1.5.17 and Theorem 1.5.20, which is a contradiction to the choice of \( \bar{u} \). Hence, we infer \( \alpha \in \Delta^>_I \), which concludes the proof.

Remark 1.5.22. One of the essences of the previous corollary is that solving \((\text{WSP}(I, \alpha))\) for all boundary weights \( \alpha \in \partial \Delta_I \) corresponds to solving all subproblems \((\text{MOP}(K))\) of \((\text{MOP}(I))\) with \( K \subset I \).

\[ \square \]

1.6 The Euclidean Reference Point Method

Another common type of scalarization methods are the so-called reference point methods. As the name suggests, the parameter in these methods is a reference point representing a desired objective value in the objective space. In most cases, this reference point is not attainable by the multiobjective cost function, i.e., it is no element of the objective set, so that the goal is to reach it as closely as possible. Therefore, the idea of a reference point method is to find an objective point with minimal distance to the reference point. The \( p \)-norms for \( p \in [1, \infty] \) are often used as a measure of this distance (cf. [Mie99, Wie86, Wie98]). In this section, we will introduce a specific reference point method – the so-called Euclidean reference point method (ERPM) – which uses the Euclidean norm as a measure for the distance between the reference point and any objective point (see, e.g., [Wie80, Wie86]). In recent years, the ERPM was used, e.g., in multiobjective energy management ([RBW+09]) and multiobjective optimal control ([BBV17, POBD19]).

The reference point is seen as a parameter in this method, and the goal is to find a suitable set of reference points for computing all Pareto optimal points. It turns out that this is not as straight-forward as choosing the weights for the WSM in Section 1.5. In fact, to the best of the author’s knowledge, it is not possible to determine such a suitable set of reference points before any computations, especially for a large number of cost functions \( k \geq 3 \).

In this section, we show how such a set can be computed hierarchically by subsequently solving all subproblems of \((\text{MOP})\).

First, we introduce the Euclidean reference point problem in Section 1.6.1 and study its properties concerning solvability and Pareto optimality of solutions. For an extended Euclidean reference point problem, the connection to the WSP is shown under the convexity assumptions from Assumption 1.1. This connection is used to define the notions of Pareto admissible and Pareto sufficient sets of reference points in Section 1.6.2, where a
Pareto sufficient set of reference points contains all required reference points for computing all Pareto optimal points. In Section 1.6.3, we show how a specific Pareto sufficient set of reference points can be characterized by using solutions of subproblems. These results are transferred from the entire (MOP) to all of its subproblems in Section 1.6.4. Subsequently, it is shown how these results can be used to develop an algorithm for hierarchically computing a Pareto sufficient set of reference points, and, consequently, for computing all Pareto optimal points.

Note that this section was already published in a similar form in the preprint [BV19], to which the author made substantial contributions. In particular, for some of the proofs in this section, the reader is referred to this preprint.

1.6.1 Problem Formulation

Definition 1.6.1. For a given reference point \( z \in \mathbb{R}^k \), we define the parameter-dependent scalarization function

\[
    g_z : \mathbb{R}^k \to \mathbb{R}, \quad x \mapsto \frac{1}{2} \| x - z \|_2^2 = \frac{1}{2} \sum_{i=1}^{k} (x_i - z_i)^2,
\]

which measures the halved, squared Euclidean distance of the point \( x \) to the reference point \( z \). Define the Euclidean reference point function

\[
    \hat{J}_{g_z}(u) := \frac{1}{2} \sum_{i=1}^{k} (\hat{J}_{i}(u) - z_i)^2.
\]

Then the Euclidean reference point problem (ERPP) to the reference point \( z \) reads

\[
    \min_{u \in \mathcal{U}_{\text{ad}}} \hat{J}_{g_z}(u) = \min_{u \in \mathcal{U}_{\text{ad}}} \frac{1}{2} \sum_{i=1}^{k} (\hat{J}_{i}(u) - z_i)^2. \quad (\text{ERPP}(z))
\]

Under some assumption on the reference point, every solution of (ERPP\( (z) \)) is Pareto optimal.

Lemma 1.6.2. Let \( z \in \mathbb{R}^k \) with \( z \leq y_{\text{id}} \) be arbitrary and suppose that \( \bar{u} \in \mathcal{U}_{\text{ad}} \) is a global (local) solution of (ERPP\( (z) \)). Then it holds \( u \in \mathcal{U}_{\text{opt.}(\text{loc})} \).

Proof. The scalarization function \( g_z \) is strongly monotonically increasing on \( \hat{J}(\mathcal{U}_{\text{ad}}) \) if \( z \leq y_{\text{id}} \) holds. Therefore, the statement follows directly from Lemma 1.4.3. \( \square \)

Moreover, under Assumption 1.1 and the condition \( z \leq y_{\text{id}} \), the existence of a solution of (ERPP\( (z) \)) is guaranteed.

Lemma 1.6.3. Let Assumption 1.1 be satisfied and \( z \in \mathbb{R}^k \) with \( z \leq y_{\text{id}} \) be arbitrary. Then (ERPP\( (z) \)) has a global solution.
Proof. Let $z \in \mathbb{R}^k$ with $z \leq y^{id}$ be arbitrary. One can show that the Euclidean reference point function $J^{y^z}$ is convex, lower semi-continuous, bounded from below and fulfills $\lim_{\|u\|_U \to \infty} J^{y^z}(u) = \infty$ in case that $U_{ad}$ is unbounded, see, e.g., [Bee19, Theorem 3.34]. Then the existence of a global minimizer of $J^{y^z}$ follows with standard arguments in convex optimization, see, e.g., [ET99, Proposition II 1.2].

However, the condition $z \leq y^{id}$ is quite restrictive. One direct consequence from Lemma 1.6.2 is that any $\bar{u} \in U_{opt} \setminus U_{opt}$ can never be a solution of (ERPP($z$)) if $z \leq y^{id}$ holds. Therefore, we want to find a weaker condition on the reference point to guarantee both solvability of (ERPP($z$)) and (weak) Pareto optimality of the solution(s). To this end, we introduce a generalization of (ERPP($z$)) as an auxiliary problem, see also [Bee19, Definition 3.36].

Definition 1.6.4. For a given reference point $z \in \mathbb{R}^k$, we call

$$\min_{y \in \hat{J}(U_{ad}) + \mathbb{R}^k_{\geq}} \frac{1}{2} \|y - z\|^2_2$$

(EERPP($z$))

the extended Euclidean reference point problem (EERPP).

Our first goal is to prove that (EERPP($z$)) has a unique solution for every reference point $z \in \mathbb{R}^k$ if Assumption 1.1 is satisfied.

Theorem 1.6.5. Let Assumption 1.1 be satisfied. Then for all reference points $z \in \mathbb{R}^k$ the problem (EERPP($z$)) has a unique solution $\bar{y} \in \hat{J}(U_{ad}) + \mathbb{R}^k_{\geq}$. A sufficient and necessary optimality condition for the unique solution $\bar{y}$ is given by

$$\langle \bar{y} - z, y - \bar{y} \rangle \geq 0 \quad \text{for all } y \in \hat{J}(U_{ad}) + \mathbb{R}^k_{\geq}. \quad (1.6.1)$$

Proof. According to Lemmata 1.5.3 and 1.5.8, the set $\hat{J}(U_{ad}) + \mathbb{R}^k_{\geq}$ is convex and closed. By applying the Hilbert projection theorem ([Rud08, Theorem 4.10]), we get that (EERPP($z$)) has a unique solution $\bar{y} \in \hat{J}(U_{ad}) + \mathbb{R}^k_{\geq}$ for all reference points $z \in \mathbb{R}^k$. Moreover, the optimality condition (1.6.1) was shown in [Ban17, Definition and Theorem 2.14].

Definition 1.6.6. Let Assumption 1.1 be satisfied. Then we can define the solution mapping

$$P : \mathbb{R}^k \to \hat{J}(U_{ad}) + \mathbb{R}^k_{\geq}, \ z \mapsto \bar{y},$$

which maps a reference point $z \in \mathbb{R}^k$ to the unique solution $\bar{y}$ of (EERPP($z$)).

Since the solution mapping $P$ can be seen as a projection, we can show the following continuity result.

Theorem 1.6.7. Let Assumption 1.1 be satisfied and $z^1, z^2 \in \mathbb{R}^k$ be two arbitrary reference points. Then it holds

$$\|P(z^1) - P(z^2)\|_2 \leq \|z^1 - z^2\|_2,$$

i.e., the mapping $P$ is Lipschitz continuous with Lipschitz constant 1.
Proof. This is a general property of projections onto convex and closed sets. It is shown, e.g., in the last part of [Ban17, Definition and Theorem 2.14].

\textbf{Lemma 1.6.8.} Let Assumption 1.1 be satisfied and \( z \in \mathbb{R}^k \) be arbitrary. Then it holds \( \mathcal{P}(z) \geq z \).

\textit{Proof.} Using the optimality condition (1.6.1) with \( y = \mathcal{P}(z) + d \in \hat{J}(\text{ad}) + \mathbb{R}_k^{\geq} \) for an arbitrary \( d \in \mathbb{R}_k^{\geq} \) yields \( \langle \mathcal{P}(z) - z, d \rangle \geq 0 \). Hence, we can conclude \( \mathcal{P}(z) \geq z \).

We can show the following connection between the EERPP and the WSP. This was already stated in a similar way in [Wie80, Lemma 5] and [Bee19, Lemma 3.27].

\textbf{Theorem 1.6.9.} Let Assumption 1.1 be satisfied.

(i) Let \( z \in (\hat{J}(\text{ad}) + \mathbb{R}_k^{\geq})^c \) be a reference point with projection \( \mathcal{P}(z) = \hat{J}(\bar{u}) + d \in \hat{J}(\text{ad}) + \mathbb{R}_k^{\geq} \) for some \( \bar{u} \in \text{ad} \) and \( d \in \mathbb{R}_k^{\geq} \). For the weight \( \alpha := \frac{\mathcal{P}(z) - z}{\| \mathcal{P}(z) - z \|_1} \in \Delta_k \) we have \( \bar{u} \in W_{\text{opt},w}(\alpha) \) and \( \langle d, \alpha \rangle = 0 \). In particular, if \( d \neq 0 \), there is \( i \in \{1, \ldots, k\} \) with \( \alpha_i = 0 \) so that \( \alpha \in \partial \Delta_k \) in this case.

(ii) If, on the other hand, \( \bar{u} \in W_{\text{opt},w}(\alpha) \) for some weight \( \alpha \in \Delta_k \), then we have \( \mathcal{P}(\hat{J}(\bar{u}) - t\alpha) = \hat{J}(\bar{u}) \) for any \( t \geq 0 \).

\textit{Proof.} (i) Let \( z \in (\hat{J}(\text{ad}) + \mathbb{R}_k^{\geq})^c \) be arbitrary. Then we have \( \mathcal{P}(z) = \hat{J}(\bar{u}) + d \neq z \) and hence, Lemma 1.6.8 implies \( \mathcal{P}(z) \geq z \). Define \( \alpha := \frac{\mathcal{P}(z) - z}{\| \mathcal{P}(z) - z \|_1} \in \Delta_k \).

By choosing \( y = \hat{J}(u) + d \) for an arbitrary \( u \in \text{ad} \) in the optimality condition (1.6.1), we obtain

\[
0 \leq \langle \mathcal{P}(z) - z, \hat{J}(u) + d - \mathcal{P}(z) \rangle = \langle \mathcal{P}(z) - z, \hat{J}(u) + d - \hat{J}(\bar{u}) - d \rangle = \| \mathcal{P}(z) - z \|_1 \langle \alpha, \hat{J}(u) - \hat{J}(\bar{u}) \rangle.
\]

This implies that \( \bar{u} \) is a global solution of \( (\text{WSP}(\alpha)) \), so that \( \bar{u} \in W_{\text{opt},w}(\alpha) \).

To prove that \( d \) and \( \alpha \) are orthogonal to each other, we plug \( y = \hat{J}(\bar{u}) \) into the optimality condition (1.6.1). This yields

\[
0 \leq \langle \mathcal{P}(z) - z, \hat{J}(\bar{u}) - \mathcal{P}(z) \rangle = \langle \mathcal{P}(z) - z, \hat{J}(\bar{u}) - \hat{J}(\bar{u}) - d \rangle = \| \mathcal{P}(z) - z \|_1 \langle \alpha, -d \rangle.
\]

Since \( \alpha, d \geq 0 \) and \( \| \mathcal{P}(z) - z \|_1 > 0 \), this implies \( \langle \alpha, d \rangle = 0 \).
(ii) Now let $\alpha \in \Delta_k$ be arbitrary and $\bar{u} \in W_{\text{opt},w}(\alpha)$. For any $t \geq 0$ define $z := \hat{J}(\bar{u}) - t\alpha$. We show $P(z) = \hat{J}(\bar{u})$ by checking the optimality condition (1.6.1). For any $y \in \hat{J}(\mathcal{U}_{\text{ad}}) + \mathbb{R}_k^k$ we have

$$\langle \hat{J}(\bar{u}) - z, y - \hat{J}(\bar{u}) \rangle = \langle \hat{J}(\bar{u}) - (\hat{J}(\bar{u}) - t\alpha), y - \hat{J}(\bar{u}) \rangle = t\langle \alpha, y - \hat{J}(\bar{u}) \rangle \geq 0,$$

since $\bar{u}$ is a global solution of $(\text{WSP}(\alpha))$ and $y \in \hat{J}(\mathcal{U}_{\text{ad}}) + \mathbb{R}_k^k$. Hence, we can conclude $P(z) = \hat{J}(\bar{u})$.

Based on Theorem 1.6.9, we can also define the following function, which will become important later.

**Definition 1.6.10.** Let Assumption 1.1 be satisfied. Then we define the function $\mathcal{H}$ by

$$\mathcal{H}: (\hat{J}(\mathcal{U}_{\text{ad}}) + \mathbb{R}_k^k)^c \rightarrow \Delta_k, \ z \mapsto \frac{P(z) - z}{\|P(z) - z\|_1}.$$

**Lemma 1.6.11.** Let Assumption 1.1 be satisfied. Then the function $\mathcal{H}$ is continuous.

**Proof.** For any $z \in (\hat{J}(\mathcal{U}_{\text{ad}}) + \mathbb{R}_k^k)^c$ we know that $P(z) \neq z$. From Theorem 1.6.7, we know that the function $P$ is continuous. Now the continuity of $\mathcal{H}$ follows as a concatenation of continuous functions.

### 1.6.2 Pareto Admissible Reference Points

In Theorem 1.6.5, we have seen that $(\text{ERPP}(z))$ has a unique solution for all reference points in $\mathbb{R}^k$. In the following, we want to investigate for which reference points the problems $(\text{ERPP}(z))$ and $(\text{EERPP}(z))$ are equivalent in the sense that

(i) $(\text{ERPP}(z))$ has a global solution,

(ii) the set of global solutions is given by $\{u \in \mathcal{U}_{\text{ad}} \mid \hat{J}(u) = P(z)\}$.

**Definition 1.6.12.** Let Assumption 1.1 be satisfied. A reference point $z \in \mathbb{R}^k$ is called admissible if the global solutions of $(\text{ERPP}(z))$ are given by the set

$$\{u \in \mathcal{U}_{\text{ad}} \mid \hat{J}(u) = P(z)\} \neq \emptyset.$$

The set of all admissible reference points is denoted by $\mathcal{Z}_{\text{ad}}$.

**Definition 1.6.13.** Let Assumption 1.1 be satisfied. We define the solution mapping

$$\mathcal{M}: \mathcal{Z}_{\text{ad}} \mapsto \mathcal{U}_{\text{ad}}, \ z \mapsto \{u \in \mathcal{U}_{\text{ad}} \mid u \text{ is a global solution of } (\text{ERPP}(z))\},$$

which maps an admissible reference point to the set of all global solutions of $(\text{ERPP}(z))$.

**Corollary 1.6.14.** Let Assumption 1.1 be satisfied. For all $z \in \mathcal{Z}_{\text{ad}}$ it holds $\mathcal{M}(z) \neq \emptyset$. Moreover, we have $\hat{J}(\mathcal{M}(z)) = \{P(z)\}$, i.e., $\mathcal{M}(z) = \hat{J}^{-1}(\{P(z)\})$.

**Proof.** Both statements follow directly from Definition 1.6.12.
1. Multiobjective Optimization

(a) Solving (ERPP(z)).

(b) Solving (EERPP(z)).

(c) Solving (ERPP(z)).

(d) Solving (EERPP(z)).

Figure 1.6: (a) & (b): The solutions for (ERPP(z)) and (EERPP(z)) coincide, so that $z \in Z_{ad}$ holds. (c) & (d): The solutions for (ERPP(z)) and (EERPP(z)) do not coincide, so that $z \notin Z_{ad}$ holds.

Remark 1.6.15. Solving (ERPP(z)) is equivalent to solving (EERPP(z)) for every $z \in Z_{ad}$. Therefore, the mapping $P$ is also the solution mapping of the ERPP for reference points in $Z_{ad}$, see Corollary 1.6.14. In particular, the Lipschitz continuity shown in Theorem 1.6.7 also holds for solutions of the ERPPs if the reference points $z^1, z^2$ are taken from the set $Z_{ad}$. This is the reason why the ERPM performs better than the WSM when it comes to approximating the Pareto front. The advantage is that the maximal distance between two Pareto optimal points can be directly controlled by bounding the distance of the corresponding reference points. In this way, it is possible to get a guaranteed coverage of the Pareto front – in contrast to the WSM.

The following theorem gives a first (a posteriori) characterization of admissible reference points.
Theorem 1.6.16. Let Assumption 1.1 be satisfied and \( z \in \mathbb{R}^k \) be arbitrary. Then

\[
z \in \mathcal{Z}_{\text{ad}} \iff \mathcal{P}(z) \in \hat{J}(\mathcal{U}_{\text{ad}}).
\]

Proof. A proof of this statement can be found in [BV19, Theorem 3.14]. □

By using Theorem 1.6.16 in combination with Theorem 1.6.9, we can prove a characterization for admissible reference points in the set \( (\hat{J}(\mathcal{U}_{\text{ad}}) + \mathbb{R}^k_{\geq})^\mathcal{C} \) — if the solutions to all WSPs are known.

Corollary 1.6.17. Let Assumption 1.1 be satisfied and \( z \in (\hat{J}(\mathcal{U}_{\text{ad}}) + \mathbb{R}^k_{\geq})^\mathcal{C} \) be arbitrary. Then the following statements are equivalent:

(i) \( z \in \mathcal{Z}_{\text{ad}} \),
(ii) \( \mathcal{P}(z) \in \hat{J}(\mathcal{U}_{\text{ad}}) \),
(iii) \( \mathcal{P}(z) \in \mathcal{J}_{\text{opt},w} \),
(iv) \( \mathcal{P}(z) \in \mathcal{F}_{\text{opt},w}(\alpha) \subset \mathcal{J}_{\text{opt},w} \) with \( \alpha = \frac{\mathcal{P}(z) - z}{\|\mathcal{P}(z) - z\|_1} \),
(v) \( \mathcal{M}(z) \subset \mathcal{W}_{\text{opt},w}(\alpha) \subset \mathcal{U}_{\text{opt},w} \) with \( \alpha = \frac{\mathcal{P}(z) - z}{\|\mathcal{P}(z) - z\|_1} \),
(vi) \( \exists \alpha \in \Delta_k : \exists u \in \mathcal{W}_{\text{opt},w}(\alpha) : \exists t > 0 \colon z = \hat{J}(u) - t\alpha \).

Proof. (i) \( \iff \) (ii) was shown in Theorem 1.6.16. (iv) \( \Rightarrow \) (iii) \( \Rightarrow \) (ii) is clear, and (ii) \( \Rightarrow \) (iv) follows from Theorem 1.6.9 (i). The equivalence (iv) \( \iff \) (v) is directly implied by Corollary 1.6.14. The implication (iv) \( \Rightarrow \) (vi) is proved by rearranging the equality \( \alpha = (\mathcal{P}(z) - z)/\|\mathcal{P}(z) - z\|_1 \). Finally, we can conclude (vi) \( \Rightarrow \) (iv) from Theorem 1.6.9 (ii). □

Remark 1.6.18. (i) We only consider reference points in \( z \in (\hat{J}(\mathcal{U}_{\text{ad}}) + \mathbb{R}^k_{\geq})^\mathcal{C} \), since for every \( z \in \hat{J}(\mathcal{U}_{\text{ad}}) + \mathbb{R}^k_{\geq} \), it trivially holds \( \mathcal{P}(z) = z \).

(ii) In particular, Corollary 1.6.17 tells us that for every admissible reference point \( z \in \mathcal{Z}_{\text{ad}} \), which additionally fulfills \( z \notin \hat{J}(\mathcal{U}_{\text{ad}}) + \mathbb{R}^k_{\geq} \), there is a global solution of \( (\text{ERPP}(z)) \) and all global solutions are weakly Pareto optimal. Note that this is a weaker assumption on the reference point \( z \) than the one used in Lemmata 1.6.2 and 1.6.3. However, the drawback is that we need to know the solutions of \( (\text{WSP}(\alpha)) \) for all weights \( \alpha \in \Delta_k \) to decide if a reference point is admissible. This issue will be tackled later on in Section 1.6.3. □

The next step is to investigate for which admissible reference points the solutions of \( (\text{ERPP}(z)) \) are Pareto optimal. To this end, the notion of (weakly) Pareto admissible reference points is introduced, cf. Section 1.4.1.
Proof. We only need to show \( \mathcal{M}(z) \subset \U_{\text{opt.}(w)} \), or equivalently \( \mathcal{P}(z) \in \partial_{\text{opt.}(w)} \), holds. The set of all (weakly) Pareto admissible reference points is denoted by \( \mathcal{Z}_{\text{opt.}(w)} \).

**Remark 1.6.20.** By Corollary 1.6.17 and Remark 1.6.18, we have

\[
\mathcal{Z}_{\text{ad}} \cap \left( \hat{J}(\mathcal{U}_{\text{ad}}) + \mathbb{R}^k_+ \right)^c \subset \mathcal{Z}_{\text{opt.}(w)}.
\]

Moreover, the Lemmata 1.6.2 and 1.6.3 imply that

\[
\{ z \in \mathbb{R}^k \mid z \leq y^{\text{jd}} \} \subset \mathcal{Z}_{\text{opt}}
\]

is satisfied.

A first characterization of the set of (weakly) Pareto admissible points is given by the next theorem.

**Theorem 1.6.21.** Let Assumption 1.1 be satisfied. Then we have

\[
\mathcal{Z}_{\text{opt.}(w)} = \left\{ z \in \mathbb{R}^k \mid \exists \alpha \in \Delta_k : \exists u \in \mathcal{W}_{\text{opt.}(w)}(\alpha) : \exists t \geq 0 : z = \hat{J}(u) - t\alpha \right\}. \tag{1.6.2}
\]

**Proof.** '\( \subset \)': Let \( z \in \mathcal{Z}_{\text{opt.}(w)} \) be arbitrary. Then, by definition, we know that \( \mathcal{P}(z) \in \partial_{\text{opt.}(w)} \). If \( z \in \hat{J}(\mathcal{U}_{\text{ad}}) + \mathbb{R}^k_+ \), we have \( z = \mathcal{P}(z) \in \partial_{\text{opt.}(w)} \). Hence, there is \( \alpha \in \Delta_k \) and \( u \in \mathcal{W}_{\text{opt.}(w)}(\alpha) \) with \( z = \mathcal{P}(z) = \hat{J}(u) - 0 \cdot \alpha \). If \( z \notin \hat{J}(\mathcal{U}_{\text{ad}}) + \mathbb{R}^k_+ \), the claim follows from the implication (i) \( \Rightarrow \) (vi) of Corollary 1.6.17, since \( \mathcal{Z}_{\text{opt.}(w)} \subset \mathcal{Z}_{\text{ad}} \).

'\( \supset \)': Assume that there are \( \alpha \in \Delta_k \), \( u \in \mathcal{W}_{\text{opt.}(w)}(\alpha) \) and \( t \geq 0 \) with \( z = \hat{J}(u) - t\alpha \). Then Theorem 1.6.9 implies \( \mathcal{P}(z) = \hat{J}(u) \in \mathcal{F}_{\text{opt.}(w)}(\alpha) \subset \partial_{\text{opt.}(w)} \). Thus, \( z \in \mathcal{Z}_{\text{opt.}(w)} \). \( \square \)

After having defined (weakly) Pareto admissible reference points, we are interested in investigating whether all (weakly) Pareto optimal points are solutions to an ERPP. To this end, we introduce the notion of (weak) Pareto sufficiency, cf. Section 1.4.1.

**Definition 1.6.22.** A set \( Z \subset \mathbb{R}^k \) is called (weakly) Pareto sufficient if \( \mathcal{M}(Z) = \U_{\text{opt.}(w)} \), or equivalently \( \mathcal{P}(Z) = \partial_{\text{opt.}(w)} \), holds.

**Theorem 1.6.23.** Let Assumption 1.1 be satisfied. Then we have

\[
\mathcal{M}(\mathcal{Z}_{\text{opt.}(w)}) = \U_{\text{opt.}(w)} \quad \text{and} \quad \mathcal{P}(\mathcal{Z}_{\text{opt.}(w)}) = \partial_{\text{opt.}(w)}, \tag{1.6.3}
\]

i.e., the set \( \mathcal{Z}_{\text{opt.}(w)} \) is (weakly) Pareto sufficient.

**Proof.** We only need to show \( \mathcal{M}(\mathcal{Z}_{\text{opt.}(w)}) = \U_{\text{opt.}(w)} \), since \( \mathcal{P}(\mathcal{Z}_{\text{opt.}(w)}) = \partial_{\text{opt.}(w)} \) can be directly concluded from Corollary 1.6.14 in this case.

'\( \subset \)': This inclusion follows from the definition of \( \mathcal{Z}_{\text{opt.}(w)} \).
'⊃': Let \( u \in U_{\text{opt},(w)} \) be arbitrary. By Theorem 1.5.13, there is a weight \( \alpha \in \Delta_k \) such that \( u \in W_{\text{opt},(w)}(\alpha) \). Define \( z := \hat{J}(u) - t\alpha \) for an arbitrary \( t \geq 0 \). By (1.6.2), we know that \( z \in Z_{\text{opt},(w)} \) and Theorem 1.6.9 (iii) implies \( P(z) = \hat{J}(u) \). By Corollary 1.6.14, this induces \( u \in M(z) \subseteq M(Z_{\text{opt},(w)}) \), which concludes the proof.

For all reference points in \( Z_{\text{opt},(w)} \) we can show – under additional differentiability assumptions – the following necessary and sufficient first-order optimality condition for a solution of the ERPP.

**Lemma 1.6.24.** Let Assumption 1.1 be satisfied, \( z \in Z_{\text{opt},(w)} \) be arbitrary and assume additionally that all cost functions are Fréchet differentiable. Then a necessary first-order optimality condition for a solution \( \bar{u} \in U_{\text{ad}} \) of \( (\text{ERPP}(z)) \) is given by

\[
\sum_{i=1}^{k} (\hat{J}_i(\bar{u}) - z_i) \nabla \hat{J}_i(\bar{u}) \geq 0 \quad \text{for all} \ u \in U_{\text{ad}}.
\]

(1.6.4)

Moreover, if \( \bar{u} \in U_{\text{ad}} \) satisfies (1.6.4) and \( \hat{J}(\bar{u}) \geq z \), then \( \bar{u} \) is a solution of \( (\text{ERPP}(z)) \).

**Proof.** Note that it holds

\[
\nabla \hat{J}_{\alpha}(u) = \sum_{i=1}^{k} (\hat{J}_i(u) - z_i) \nabla \hat{J}_i(u)
\]

for all \( u \in U_{\text{ad}} \). From \cite[Lemma 2.21]{Trö10}, we immediately get that (1.6.4) is a necessary first-order optimality condition for a solution of \( (\text{ERPP}(z)) \). On the other hand, let \( \bar{u} \in U_{\text{ad}} \) be arbitrary such that (1.6.4) and \( \hat{J}(\bar{u}) \geq z \) are satisfied. By defining \( \alpha := (\hat{J}(\bar{u}) - z)/\|\hat{J}(\bar{u}) - z\|_1 \in \Delta_k \), we see that \( \bar{u} \) fulfills the sufficient first-order optimality condition (1.5.1) for a solution of \( (\text{WSP}(\alpha)) \). Thus, \( \bar{u} \) is a solution of \( (\text{WSP}(\alpha)) \). Since we can write \( z = \hat{J}(\bar{u}) - t\alpha \) for some \( t > 0 \), we infer \( P(z) = \hat{J}(\bar{u}) \) from Theorem 1.6.9. From Corollary 1.6.14, we can now conclude \( \bar{u} \in M(z) \) so that \( \bar{u} \) is a solution of \( (\text{ERPP}(z)) \).

### 1.6.3 A Pareto Sufficient Set of Reference Points

In the previous section, we have seen that the set \( Z_{\text{opt},(w)} \) is (weakly) Pareto sufficient and the biggest (weakly) Pareto admissible subset of \( \mathbb{R}^k \). In particular, it contains all reference points that are relevant for a computation of the (weak) Pareto front by the ERPM.

However, thinking of a numerical implementation of the ERPM, there are still two issues: First, \( Z_{\text{opt},(w)} \) contains many redundant reference points, which would make a numerical algorithm inefficient. In fact, given \( \alpha \in \Delta_k \) and \( y \in F_{\text{opt},(w)}(\alpha) \), Theorem 1.6.9 tells us that \( P(y - t\alpha) = y \) holds for all \( t \geq 0 \). Consequently, a set \( Z \subseteq Z_{\text{opt},(w)} \) is already (weakly) Pareto sufficient if for each \( \alpha \in \Delta_k \) and \( y \in F_{\text{opt},(w)}(\alpha) \), there is exactly one \( t = t(y, \alpha) \geq 0 \) such that \( z = y - t(y, \alpha) \alpha \in Z \). Second, the characterization (1.6.2) of \( Z_{\text{opt},(w)} \) given in Theorem 1.6.21 relies on the solutions of \( (\text{WSP}(\alpha)) \) for all weights \( \alpha \in \Delta_k \). Since we
would have already computed the (weak) Pareto front if we had solved \((WSP(\alpha))\) for all weights \(\alpha \in \Delta_k\), this characterization is useless from a practical point of view.

For the case of two cost functions, there are several approaches in the literature to overcome these issues in a numerical implementation by iteratively computing Pareto admissible points such that the entire Pareto front is approximated. These approaches make use of the fact that the objective space is two-dimensional in two ways: First, it is geometrically evident that a Pareto sufficient set of reference points can be bounded by using information about the minimizers of the two cost functions. Thus, the starting and end point of the iteration can be easily specified after having computed the minimizers of the two cost functions. Second, there is a clear direction (either up or down), in which the next reference point can be chosen. In this regard, it was shown in [Del08] that, given a reference point \(z\) and a corresponding solution \(\tilde{u}\) of \((ERPP(z))\), the vector \(\tilde{J}(\tilde{u}) - z\) is orthogonal to the tangent space of the Pareto front at the point \(\tilde{J}(\tilde{u})\). This result was used to iteratively compute new reference points based on tangent information of the Pareto front. This idea was cited and slightly modified in [RBW+09], where the new reference point is chosen based on a first-order approximation of the slope of the Pareto front, i.e., the tangent information of the Pareto front is only approximated. Later on, these approaches or slight modifications were also used in several other publications for the case of two cost functions, cf. [Ban17, BBV16, BBV17, Bee19, POBD19].

However, these two geometrical arguments can only be used in the case of two cost functions, see, e.g., [Bee19, Figure 3.6]. Thus, it is not straightforward to systematically generalize these approaches to more than two cost functions. In fact, to the best of the author’s knowledge, there are no systematic approaches for solving MOPs by the ERPM in the case of more than two cost functions. However, for other scalarization methods, as, e.g., the Pascoletti-Serafini method (see Section 1.7), such approaches exist. In [MGGS09], the solution to subproblems with \(l - 1\) cost functions is used to construct reference points for subproblems with \(l\) cost functions for the Pascoletti-Serafini method. For further description, discussion and possible modifications of this approach, the reader is referred to Section 1.7.2 at this point.

For the ERPM, we will also follow this idea and use solutions to subproblems to hierarchically compute the Pareto front. However, our incentive is to find a characterization of a Pareto sufficient set of reference points, whereas the idea in [MGGS09] was to directly characterize the boundary of the Pareto front by the solutions to subproblems. In particular, we show that this Pareto sufficient set of reference points is intrinsically characterized by the solutions to subproblems as well and we will use this characterization for developing an algorithm for solving (MOP) with arbitrarily many cost functions in Section 1.6.4.

To this end, in this section, we will show how an efficient (weakly) Pareto sufficient subset of \(\mathcal{Z}_{\text{opt},(w)}\) can be constructed and how it can be characterized by the solutions to \((WSP(\alpha))\) for all \(\alpha \in \partial \Delta_k\), i.e., by solutions to subproblems of (MOP), cf. Remark 1.5.22. This lays the foundation for the hierarchical algorithm in Section 1.6.4.
Let us now show some properties of the mapping $G$. We have

$$y^\text{id} := y^\text{id} - \tilde{d},$$

where the ideal point $y^\text{id}$ was introduced in Definition 1.2.5. Let $D_i \subset \mathbb{R}^k$ be given by

$$D_i := \{ y \in \mathbb{R}^k | y \geq \tilde{y}^\text{id}, y_i = \tilde{y}^\text{id}_i \}$$

for all $i \in \{1, \ldots, k\}$. Then the set $D \subset \mathbb{R}^k$ is defined by $D := \bigcup_{i=1}^k D_i$.

For the remaining part of this chapter, let $\tilde{d} \in \mathbb{R}^k$ be arbitrary but fixed and $\tilde{y}^\text{id}$ and $D$ be given as in Definition 1.6.25. In the following, we want to investigate the set $\mathcal{Z}_{\text{opt.}(w)} \cap D$.

Remark 1.6.26. We use the shifted ideal point $\tilde{y}^\text{id}$ instead of $y^\text{id}$ to enforce that we have $D \subset \left(\hat{J}(\mathcal{U}_{\text{ad}}) + \mathbb{R}^k_{\geq}\right)^c$. Indeed, for an arbitrary $y \in D$ there is $i \in \{1, \ldots, k\}$ with $y \in D_i$, i.e., $y_i = \tilde{y}^\text{id}_i - d_i < y^\text{id}_i$. By definition of $y^\text{id}$, this implies $y \notin \hat{J}(\mathcal{U}_{\text{ad}}) + \mathbb{R}^k_{\geq}$. ♦

Definition 1.6.27. Let Assumption 1.1 be satisfied. Given $D$ as above, we can define the (set-valued) mapping $G_{\text{opt.}(w)}^D$ by

$$G_{\text{opt.}(w)}^D : \Delta_k \ni \alpha \mapsto \left\{ y - t\alpha \bigg| y \in \mathcal{F}_{\text{opt.}(w)}^\alpha, t = \min_{i \in \{1, \ldots, k\}} \frac{y_i - \tilde{y}^\text{id}_i}{\alpha_i} \right\}.$$

Theorem 1.6.28. Let Assumption 1.1 be satisfied. Then the mapping $G_{\text{opt.}(w)}^D$ is well-defined and we have $G_{\text{opt.}(w)}^D(\alpha) \neq \emptyset$ for all $\alpha \in \Delta_k$. If the cost functions $\hat{J}_1, \ldots, \hat{J}_k$ are additionally strictly convex, the mapping $G_{\text{opt.}}^D = G_{\text{opt.}}^D$ is single-valued. If the cost functions are even strongly convex and twice continuously Fréchet differentiable, the function $G_{\text{opt.}}^D$ is continuous.

Proof. The well-definedness of $G_{\text{opt.}(w)}^D$ can be proven by using the definition of $D$, see [BV19, Theorem 4.4] for more details. For any $\alpha \in \Delta_k$, we have $\mathcal{F}_{\text{opt.}(w)}^\alpha(\alpha) \neq \emptyset$ by Theorem 1.5.13. Hence, we can also conclude $G_{\text{opt.}(w)}^D(\alpha) \neq \emptyset$. The other two properties follow directly from Theorem 1.5.13 and Theorem 1.5.14 together with the continuity of the function $(y, \alpha) \mapsto \min_{i \in \{1, \ldots, k\}} (y_i - \tilde{y}^\text{id}_i)/\alpha_i$ at any $(y, \alpha) \in \Delta_k$, respectively. ♦

Remark 1.6.29. Figuratively speaking, the mapping $G_{\text{opt.}(w)}^D$ computes the intersection points of the rays $t \mapsto y - t\alpha$ with the set $D$, see Figure 1.7. One can easily show that this intersection point is unique. In particular, if we have $z = y - \frac{y_i - \tilde{y}^\text{id}_i}{\alpha_i} \alpha \in D_i$ for some $\alpha \in \Delta_k$, $y \in \mathcal{F}_{\text{opt.}(w)}^\alpha(\alpha)$ and $i \in \{1, \ldots, k\}$, we can already conclude $z \in G_{\text{opt.}(w)}^D(\alpha)$. ♦

Let us now show some properties of the mapping $G_{\text{opt.}(w)}^D$.
Figure 1.7: Illustration of the mapping $G^D_{\text{opt}}$. Left: The construction of $G^D_{\text{opt}}(\alpha^{(i)})$ for three different weights $\alpha^{(1)}, \alpha^{(2)}, \alpha^{(3)} \in \Delta_2$. Right: The entire set $G^D_{\text{opt}}(\Delta_2)$.

Lemma 1.6.30. Let Assumption 1.1 be satisfied.

(i) For $\alpha, \beta \in \Delta_k$ with $\alpha \neq \beta$ we have $G^D_{\text{opt},(w)}(\alpha) \cap G^D_{\text{opt},(w)}(\beta) = \emptyset$.

(ii) We have

$$G^D_{\text{opt}}|_{\Delta_k} = G^D_{\text{opt},w}|_{\Delta_k}.$$

In particular, this implies $G^D_{\text{opt}}(\Delta_k^\gamma) = G^D_{\text{opt},w}(\Delta_k^\gamma)$.

(iii) The identities

$$G^D_{\text{opt}}(\Delta_k) \setminus G^D_{\text{opt}}(\partial \Delta_k) = G^D_{\text{opt}}(\Delta_k^\gamma) = G^D_{\text{opt},w}(\Delta_k^\gamma) = G^D_{\text{opt},w}(\Delta_k) \setminus G^D_{\text{opt},w}(\partial \Delta_k)$$

are satisfied.

Proof. These statements are shown in [BV19, Lemma 4.5, Corollaries 4.6 & 4.7].

As explained in the beginning of this section, we want to study the set $Z_{\text{opt},(w)} \cap D$. As it turns out, this set is the image of $\Delta_k$ under the mapping $G^D_{\text{opt},(w)}$.

Theorem 1.6.31. Let Assumption 1.1 be satisfied. Then it holds

$$G^D_{\text{opt},(w)}(\Delta_k) = Z_{\text{opt},(w)} \cap D.$$

Proof. The proof mainly uses the definitions of $G^D_{\text{opt},(w)}$ and $D$ as well as the characterization of $Z_{\text{opt},(w)}$ in Theorem 1.6.21. For a detailed version of the proof, we refer to [BV19, Theorem 4.8].

Using Theorem 1.6.9 (ii), Theorem 1.6.23 and Theorem 1.6.31, we can prove that the set $G^D_{\text{opt},(w)}(\Delta_k)$ is (weakly) Pareto sufficient.
1.6 The Euclidean Reference Point Method

**Theorem 1.6.32.** Let Assumption 1.1 be satisfied. Then it holds
\[
\mathcal{M} \left( \mathcal{G}^D_{\text{opt}(w)}(\alpha) \right) = \mathcal{W}_{\text{opt}(w)}(\alpha) \quad \text{and} \quad \mathcal{P} \left( \mathcal{G}^D_{\text{opt}(w)}(\alpha) \right) = \mathcal{F}_{\text{opt}(w)}(\alpha)
\]
for all \( \alpha \in \Delta_k \). In particular, this implies
\[
\mathcal{M} \left( \mathcal{G}^D_{\text{opt}(w)}(\Delta_k) \right) = \mathcal{U}_{\text{opt}(w)} \quad \text{and} \quad \mathcal{P} \left( \mathcal{G}^D_{\text{opt}(w)}(\Delta_k) \right) = \mathcal{J}_{\text{opt}(w)},
\]
i.e., the set \( \mathcal{G}^D_{\text{opt}(w)}(\Delta_k) \) is (weakly) Pareto sufficient.

**Proof.** For both (1.6.5) and (1.6.6), we only need to prove the identity involving \( \mathcal{P} \), since then Corollary 1.6.14 implies the corresponding identity involving \( \mathcal{M} \). So let \( \alpha \in \Delta_k \) be arbitrary.

'\( \subset \)': To prove \( \mathcal{P}(\mathcal{G}^D_{\text{opt}(w)}(\alpha)) \subset \mathcal{F}_{\text{opt}(w)}(\alpha) \), let \( z \in \mathcal{G}^D_{\text{opt}(w)}(\alpha) \) be arbitrary. Then we can write
\[
z = y - \min_{i \in \{1, \ldots, k\}} \frac{y_i - \hat{y}^\text{id}_i}{\alpha_i} \alpha
\]
for some \( y \in \mathcal{F}_{\text{opt}(w)}(\alpha) \). Now Theorem 1.6.9 (ii) directly implies \( \mathcal{P}(z) = y \in \mathcal{F}_{\text{opt}(w)}(\alpha) \), which is what we had to show.

'\( \supset \)': To show \( \mathcal{F}_{\text{opt}(w)}(\alpha) \subset \mathcal{P}(\mathcal{G}^D_{\text{opt}(w)}(\alpha)) \), let \( y \in \mathcal{F}_{\text{opt}(w)}(\alpha) \) be arbitrary. Again from Theorem 1.6.9 (ii), we infer \( \mathcal{P}(z) = y \) for the reference point
\[
z := y - \min_{i \in \{1, \ldots, k\}} \frac{y_i - \hat{y}^\text{id}_i}{\alpha_i} \alpha \in \mathcal{G}^D_{\text{opt}(w)}(\alpha).
\]
Thus, \( y = \mathcal{P}(z) \in \mathcal{P}(\mathcal{G}^D_{\text{opt}(w)}(\alpha)) \), which was what we had to show.

Finally, (1.6.6) can be concluded from \( \mathcal{J}_{\text{opt}(w)} = \mathcal{F}_{\text{opt}(w)}(\Delta_k) \), cf. Theorem 1.5.13.

From Theorem 1.6.31, we have \( \mathcal{G}^D_{\text{opt}(w)}(\Delta_k) = \mathcal{Z}_{\text{opt}(w)} \cap D \) and Theorem 1.6.32 tells us that this set is (weakly) Pareto sufficient. However, there is still the same problem as before: We need to solve (WSP(\( \alpha \))) for all weights \( \alpha \in \Delta_k \) to obtain the set \( \mathcal{G}^D_{\text{opt}(w)}(\Delta_k) \).
In the following, we will show how to circumvent this problem.

**Definition 1.6.33.** Let Assumption 1.1 be satisfied. Then we define the function
\[
\mathcal{H}^D_{\text{opt}(w)} := \mathcal{H} |_{\mathcal{G}^D_{\text{opt}(w)}(\Delta_k)} : \mathcal{G}^D_{\text{opt}(w)}(\Delta_k) \to \Delta_k,
\]
where the function \( \mathcal{H} \) was defined in Definition 1.6.10.

**Corollary 1.6.34.** Let Assumption 1.1 be satisfied. Then the function \( \mathcal{H}^D_{\text{opt}(w)} \) is continuous.

**Proof.** This follows directly from Lemma 1.6.11, where we showed that \( \mathcal{H} \) is continuous.
We have the following connection between the mapping $G_{\text{opt},(w)}^D$ and the function $H_{\text{opt},(w)}^D$.

**Theorem 1.6.35.** Let Assumption 1.1 be satisfied. For all $\alpha \in \Delta_k$ it holds

$$(H_{\text{opt},(w)}^D)^{-1}(\{\alpha\}) = G_{\text{opt},(w)}^D(\alpha),$$

i.e., $G_{\text{opt},(w)}^D$ is the preimage mapping of $H_{\text{opt},(w)}^D$.

**Proof.** The proof of this statement uses the definitions of $G_{\text{opt},(w)}^D$ and $H_{\text{opt},(w)}^D$ together with Theorem 1.6.9 (ii) and is carried out in detail in [BV19, Theorem 4.12].

**Corollary 1.6.36.** Let Assumption 1.1 be satisfied. If the cost functions $\hat{J}_1, \ldots, \hat{J}_k$ are additionally strictly convex, it holds $G_{\text{opt},(w)}^D = (H_{\text{opt},(w)}^D)^{-1}$.

**Proof.** If the cost functions are additionally strictly convex, it was shown in Theorem 1.6.28 that $G_{\text{opt}}^D = G_{\text{opt},w}^D$ is single-valued. Now the claim follows directly from Theorem 1.6.35.

**Corollary 1.6.37.** Let Assumption 1.1 be satisfied and the cost functions be additionally strongly convex and twice continuously Fréchet differentiable. Then the mapping $G_{\text{opt},(w)}^D$ is a homeomorphism between $\Delta_k$ and $G_{\text{opt},(w)}^D(\Delta_k)$.

**Proof.** We already know that under these assumptions the functions $H_{\text{opt},(w)}^D$ and $G_{\text{opt},(w)}^D$ are continuous (cf. Lemma 1.6.11 and Theorem 1.6.28) and that $G_{\text{opt}}^D = (H_{\text{opt}}^D)^{-1}$ (cf. Corollary 1.6.36). Thus, $G_{\text{opt},(w)}^D$ is a homeomorphism between $\Delta_k$ and $G_{\text{opt},(w)}^D(\Delta_k)$.

The following theorem is the main result of this section. As it turns out, the set $G_{\text{opt},w}(\partial \Delta_k)$ contains enough information to characterize the set $G_{\text{opt},(w)}^D(\Delta_k) = G_{\text{opt},w}^D(\Delta_k)$.

**Theorem 1.6.38.** Let Assumption 1.1 be satisfied and $z \in D$ be arbitrary. Then the following statements are equivalent:

1. $z \notin G_{\text{opt},(w)}^D(\Delta_k^\circ) = G_{\text{opt},w}^D(\Delta_k^\circ)$,
2. $\exists \tilde{z} \in G_{\text{opt},w}^D(\partial \Delta_k) : \exists \beta \in \{H_{\text{opt},w}^D(\tilde{z})\}^\perp \cap \mathbb{R}_+^k : z = \tilde{z} + \beta$.

**Proof.** First of all, we note that the equality $G_{\text{opt}}^D(\Delta_k^\circ) = G_{\text{opt},w}^D(\Delta_k^\circ)$ was shown in Lemma 1.6.30 (ii). Now let $z \in D$ be arbitrary. In particular, there is $i \in \{1, \ldots, k\}$ with $z \in D_i$.

(i) $\Rightarrow$ (ii): Assume that $z \notin G_{\text{opt},(w)}^D(\Delta_k^\circ)$.

If $z \in G_{\text{opt},w}(\partial \Delta_k)$, the condition (ii) is satisfied for $\tilde{z} = z$ and $\beta = 0$.

If $z \notin G_{\text{opt},w}(\partial \Delta_k)$, then we can conclude $z \notin G_{\text{opt},w}^D(\Delta_k)$, since we have

$G_{\text{opt},w}^D(\Delta_k) = G_{\text{opt},w}^D(\Delta_k^\circ) \cup G_{\text{opt},w}^D(\partial \Delta_k)$.
by Lemma 1.6.30 (iii). In Theorem 1.6.31, we have shown $z_{\text{opt},w} \cap D = \mathcal{G}_{\text{opt},w}^D(\Delta_k)$. Thus, we obtain $z \notin z_{\text{opt},w}$. By definition, this directly implies $\mathcal{P}(z) \notin \mathcal{F}_{\text{opt},w}$. From the implication (iii) $\Rightarrow$ (ii) of Corollary 1.6.17, we conclude $\mathcal{P}(z) \notin \hat{J}(U_{\text{ad}})$. Hence, there is $u \in U_{\text{ad}}$ and $d \in \mathbb{R}^k_\perp$ with $\mathcal{P}(z) = \hat{J}(u) + d$. According to Theorem 1.6.9 (i), we have $\hat{J}(u) \in \mathcal{F}_{\text{opt},w}(\alpha)$ for the weight $\alpha := \mathcal{H}(z) = \frac{\mathcal{P}(z) - z}{\|\mathcal{P}(z) - z\|_1} \in \Delta_k$,

as well as $(d, \alpha) = 0$ and $\alpha \in \partial \Delta_k$. In particular, this implies $d \in \{\alpha\}^\perp \cap \mathbb{R}^k_\perp$. Moreover, due to $z \in D_i$, we know that $\mathcal{P}(z)_i > \bar{y}^d_i = z_i$, cf. Remark 1.6.26. Thus, from the definition of $\alpha$, we immediately obtain $\alpha_i > 0$ and hence, $d_i = 0$. Note that we have

$z = \mathcal{P}(z) - (\mathcal{P}(z) - z) = \mathcal{P}(z) - \|\mathcal{P}(z) - z\|_1 \alpha = \mathcal{P}(z) - \frac{(\mathcal{P}(z) - z)_i}{\alpha_i} \alpha$,

since $\alpha = (\mathcal{P}(z) - z)/\|\mathcal{P}(z) - z\|_1$ and $z_i = \bar{y}^d_i$. Next, we define $\tilde{z}$ by

$\tilde{z} := \hat{J}(u) - \frac{\hat{J}(u) - \bar{y}^d_i}{\alpha_i} \alpha = \mathcal{P}(z) - d - \frac{(\mathcal{P}(z))_i - d_i - \bar{y}^d_i}{\alpha_i} \alpha$,

where the last equality holds because of $d_i = 0$, and conclude that $z = \tilde{z} + d$. Moreover, $d_i = 0$ implies $\tilde{z}_i = \bar{y}^d_i$, and for every $j \in \{1, \ldots, k\}$ we have that $\alpha_j = 0$ implies $\tilde{z}_j = \hat{J}(u)_j > \bar{y}^d_j$, and if $\alpha_j > 0$, then $d_j = 0$ so that $\tilde{z}_j = z_j \geq \bar{y}^d_j$ can be concluded from $z \in D_i$. Altogether, this implies $\tilde{z} \in D_i$ so that $\tilde{z} \in \mathcal{G}_{\text{opt},w}^D(\alpha) \subset \mathcal{G}_{\text{opt},w}^D(\partial \Delta_k)$, cf. Remark 1.6.29. Finally, Theorem 1.6.35 yields $\mathcal{H}_{\text{opt},w}^D(\tilde{z}) = \alpha$, which concludes the proof of the implication (i) $\Rightarrow$ (ii).

(ii) $\Rightarrow$ (i): Assume that there are $\tilde{z} \in \mathcal{G}_{\text{opt},w}^D(\partial \Delta_k)$ and $\beta \in \{\mathcal{H}_{\text{opt},w}^D(\tilde{z})\}^\perp \cap \mathbb{R}^k_\perp$ with $z = \tilde{z} + \beta$.

If $z \notin \mathcal{G}_{\text{opt},w}^D(\Delta_k)$, we are done.

If $z \in \mathcal{G}_{\text{opt},w}^D(\Delta_k)$, there are $\alpha \in \Delta_k$ and $y \in \mathcal{F}_{\text{opt},w}(\alpha)$ such that $z = y - t\alpha$ for $t := \min \{1, \ldots, k\} (y_i - \bar{y}^d_i)/\alpha_i$. Note that $\langle \beta, \mathcal{H}_{\text{opt},w}^D(\tilde{z}) \rangle = 0$ implies $\langle \beta, \mathcal{P}(\tilde{z}) - \beta \rangle = 0$.

Altogether, we can conclude

$\langle \mathcal{P}(\tilde{z}) + \beta - z, y - \mathcal{P}(\tilde{z}) - \beta \rangle = \langle \mathcal{P}(\tilde{z}) + \beta - \tilde{z} - \beta, y - \mathcal{P}(\tilde{z}) - \beta \rangle = \langle \mathcal{P}(\tilde{z}) - \tilde{z}, y - \mathcal{P}(\tilde{z}) - \beta \rangle = \langle \mathcal{P}(\tilde{z}) - \tilde{z}, y - \mathcal{P}(\tilde{z}) \rangle \geq 0$,

for all $y \in \hat{J}(U_{\text{ad}}) + \mathbb{R}^k_\perp$, where we used the necessary optimality condition (1.6.1) for $\mathcal{P}(\tilde{z})$ in the last inequality. This shows that $\mathcal{P}(\tilde{z}) + \beta$ satisfies the sufficient optimality condition.
1. Multiobjective Optimization

Figure 1.8: Illustrations of the sets $G^D_{\text{opt}}(\Delta_k)$ and $G^D_{\text{opt}}(\partial \Delta_k)$ for $k = 2$ (left) and $k = 3$ (right).

(1.6.1) for $P(z)$. Thus, we have $P(z) = P(\tilde{z}) + \beta$, so that we get

$$\alpha = H^D_{\text{opt},w}(z) = \frac{P(z) - z}{\|P(z) - z\|_1} = \frac{P(\tilde{z}) - z}{\|P(z) - z\|_1} = H^D_{\text{opt},w}(\tilde{z}) \in \partial \Delta_k,$$

which implies $z \in G^D_{\text{opt},w}(\alpha) \subset G^D_{\text{opt},w}(\partial \Delta_k)$. Since we have $G^D_{\text{opt},w}(\partial \Delta_k) \cap G^D_{\text{opt},w}(\Delta_k^\gamma) = \emptyset$ by Lemma 1.6.30 (i), this yields $z \notin G^D_{\text{opt},w}(\Delta_k^\gamma)$, which is what we had to show.

Remark 1.6.39. The essence of Theorem 1.6.38 is that we have a complete characterization of the (weakly) Pareto sufficient set $G^D_{\text{opt}}(\Delta_k) = \mathcal{Z}_{\text{opt}}(w) \cap D$ by only using the set $G^D_{\text{opt},w}(\partial \Delta_k)$. This result lays the foundation for using a hierarchical algorithm for computing the Pareto front, which is described in the next section.

Theorem 1.6.40. Let Assumption 1.1 be satisfied. Then it holds

$$\partial G^D_{\text{opt},w}(\Delta_k) \cap G^D_{\text{opt},w}(\Delta_k) \subset G^D_{\text{opt},w}(\partial \Delta_k), \quad (1.6.7)$$

$$\partial G^D_{\text{opt}}(\Delta_k) \cap G^D_{\text{opt}}(\Delta_k) = G^D_{\text{opt}}(\partial \Delta_k). \quad (1.6.8)$$

Proof. For a proof of this statement we refer to [BV19, Theorem 4.14].

If we have stronger assumptions on the cost functions, it is possible to show the equality $\partial G^D_{\text{opt}}(\Delta_k) = G^D_{\text{opt}}(\partial \Delta_k)$, cf. Figure 1.8.
Corollary 1.6.41. Assume that Assumption 1.1 holds and let the cost functions be additionally strongly convex and twice continuously Fréchet differentiable. Then we have
\[ \partial G^D_{\text{opt}}(\Delta_k) = G^D_{\text{opt}}(\partial \Delta_k). \]

Proof. In Corollary 1.6.37, we showed that \( G^D_{\text{opt}} \) is a homeomorphism between the sets \( \Delta_k \) and \( G^D_{\text{opt}}(\Delta_k) \) under these assumptions. Thus, the set \( G^D_{\text{opt}}(\Delta_k) \) is closed, so that the statement follows directly from the identity (1.6.8).

### 1.6.4 The Hierarchical Euclidean Reference Point Method

Let us briefly review the results from the previous section: Our object of interest was the set \( Z_{\text{opt},(w)} \cap D \), for which we could show that it is (weakly) Pareto sufficient. Moreover, it is efficient in the sense that for every \( \alpha \in \Delta_k \) and \( y \in F_{\text{opt},(w)}(\alpha) \), there is exactly one \( t = t(y, \alpha) \geq 0 \) such that \( z = y - t(y, \alpha)\alpha \in Z_{\text{opt},(w)} \cap D \). Starting from the characterization \( G^D_{\text{opt},(w)}(\Delta_k) = Z_{\text{opt},(w)} \cap D \) in Theorem 1.6.31, which requires the solutions of (WSP(\( \alpha \))) for all \( \alpha \in \Delta_k \), we were able to show a characterization of \( Z_{\text{opt},(w)} \cap D \) in Theorem 1.6.38, which only uses the solutions of (WSP(\( \alpha \))) for all \( \alpha \in \partial \Delta_k \).

This characterization can now be used to hierarchically compute the (weak) Pareto front in the following way: According to Corollary 1.5.21 and Remark 1.5.22, solving (WSP(\( \alpha \))) for all boundary weights \( \alpha \in \partial \Delta_k \) corresponds to solving all subproblems (MOP(I)) with \( I \subseteq \{1, \ldots, k\} \). In other words, this means that for computing the set of reference points \( G^D_{\text{opt},(w)}(\Delta^>_k) \), we first need to solve all such subproblems. This procedure can be iterated: If we want to solve (MOP(I)) for \( |I| > 1 \), we can carry out the same arguments as in the previous section. Thus, to characterize a Pareto sufficient set of reference points for this subproblem, we need the solutions to (WSP(I,\( \alpha \))) for all \( \alpha \in \partial \Delta_I \). Again by Remark 1.5.22, this is equivalent to solving (MOP(J)) for all \( J \subseteq I \). Finally, if \( |I| = 1 \) holds, i.e., \( I = \{i\} \) for some \( i \in \{1, \ldots, k\} \), solving (MOP(I)) corresponds to computing the minimizers of \( \hat{J}_i \).

In this section, we transfer the results from the previous section to the framework of subproblems. Afterwards, we formalize the procedure described above and present a hierarchical algorithm for computing the Pareto front by the ERPM. For this algorithm, we prove analytically that it solves (MOP). Finally, we show how it can be implemented numerically in an efficient way and also show that this implementation solves (MOP) numerically, which is verified by a simple polynomial example.

#### 1.6.4.1 The Euclidean Reference Point Method for Subproblems

For any \( I \subset \{1, \ldots, k\} \), all the statements from the previous sections can also be shown analogously for the subproblem (MOP(I)). Therefore, we will only introduce the notation for this more general situation and state the most important results without any proofs.

We start by defining the ERPP for a subproblem (MOP(I)).
Definition 1.6.42. Let \( I \subset \{1, \ldots, k\} \) and \( z \in \mathbb{R}^I \) be arbitrary. Then we define the Euclidean reference point problem (ERPP) for the subproblem (MOP(I)) with respect to the reference point \( z \) by
\[
\min_{u \in \overline{U}_{\text{ad}}} \frac{1}{2} \sum_{i \in I} \left( \hat{J}_i(u) - z_i \right)^2.
\] (ERPP\((I, z)\))

Definition 1.6.43. Let \( I \subset \{1, \ldots, k\} \) and \( z \in \mathbb{R}^I \) be arbitrary. Then we call
\[
\min_{y \in \hat{J}^I(\overline{U}_{\text{ad}}) + \mathbb{R}^I_L} \frac{1}{2} \|y - z\|^2_2
\] (EERPP\((I, z)\))

the extended Euclidean reference point problem (EERPP) for the subproblem (MOP(I)) with respect to the reference point \( z \).

Theorem 1.6.44. Let Assumption 1.1 be satisfied and \( I \subset \{1, \ldots, k\} \) be arbitrary. Then for all reference points \( z \in \mathbb{R}^I \) the extended Euclidean reference point problem (EERPP\((I, z)\)) has a unique solution \( \bar{y} \in \hat{J}^I(\overline{U}_{\text{ad}}) + \mathbb{R}^I_L \). A sufficient and necessary optimality condition for the unique solution \( \bar{y} \) is given by
\[
\langle \bar{y} - z, y - \bar{y} \rangle \geq 0 \quad \text{for all} \quad y \in \hat{J}^I(\overline{U}_{\text{ad}}) + \mathbb{R}^I_L.
\] (1.6.9)

Definition 1.6.45. Let Assumption 1.1 be satisfied and \( I \subset \{1, \ldots, k\} \) be arbitrary. Then we can define the solution mapping
\[
P^I : \mathbb{R}^I \to \hat{J}^I(\overline{U}_{\text{ad}}) + \mathbb{R}^I_L, \quad z \mapsto \bar{y},
\]
which maps a reference point \( z \in \mathbb{R}^I \) to the unique solution \( \bar{y} \) of (EERPP\((I, z)\)).

Theorem 1.6.46. Let Assumption 1.1 be satisfied and \( I \subset \{1, \ldots, k\} \) be arbitrary.
(i) Let \( z \in (\hat{J}^I(\overline{U}_{\text{ad}}) + \mathbb{R}^I_L)^c \) be a reference point with projection
\[
P^I(z) = \hat{J}^I(\bar{u}) + d \in \hat{J}^I(\overline{U}_{\text{ad}}) + \mathbb{R}^I_L
\]
for some \( \bar{u} \in \overline{U}_{\text{ad}} \) and \( d \in \mathbb{R}^I_L \). For the weight
\[
\alpha := \frac{P^I(z) - z}{\|P^I(z) - z\|_1} \in \Delta_I
\]
we have \( \bar{u} \in W^I_{\text{opt}, w}(\alpha) \) and \( \langle d, \alpha \rangle = 0 \). In particular, if \( d \neq 0 \), there is \( i \in I \) with \( \alpha_i = 0 \) so that \( \alpha \in \partial \Delta_I \) in this case.
(ii) If, on the other hand, \( \bar{u} \in W^I_{\text{opt}, w}(\alpha) \) for some weight \( \alpha \in \Delta_I \), then we have
\[
P^I(\hat{J}^I(\bar{u}) - ta) = \hat{J}^I(\bar{u})
\]
for any \( t \geq 0 \).
**Definition 1.6.47.** Let Assumption 1.1 be satisfied and \( I \subset \{1, \ldots, k\} \) be arbitrary. A reference point \( z \in \mathbb{R}^I \) is called admissible for (MOP(I)) if the global solutions of (ERPP(I, z)) are given by the set
\[
\{ u \in \mathcal{U}_{ad} \mid \hat{J}^I(u) = \mathcal{P}^I(z) \} \neq \emptyset.
\]
The set of all admissible reference points is denoted by \( \mathcal{Z}_{ad}^I \).

**Definition 1.6.48.** Let Assumption 1.1 be satisfied and \( I \subset \{1, \ldots, k\} \) be arbitrary. We define the solution mapping
\[
\mathcal{M}^I : \mathcal{Z}_{ad}^I \Rightarrow \mathcal{U}_{ad}, \quad z \mapsto \{ u \in \mathcal{U}_{ad} \mid u \text{ is a global solution of } (ERPP(I, z)) \}
\]
which maps an admissible reference point to the set of all global solutions of (ERPP(I, z)).

**Corollary 1.6.49.** Let Assumption 1.1 be satisfied and \( I \subset \{1, \ldots, k\} \) be arbitrary. For all \( z \in \mathcal{Z}_{ad}^I \) it holds \( \mathcal{M}^I(z) \neq \emptyset \). Moreover, we have \( \hat{J}^I(\mathcal{M}^I(z)) = \{ \mathcal{P}^I(z) \} \), i.e., \( \mathcal{M}^I(z) = (\hat{J}^I)^{-1}(\{ \mathcal{P}^I(z) \}) \).

**Definition 1.6.50.** Let \( I \subset \{1, \ldots, k\} \) be arbitrary. A reference point \( z \in \mathbb{R}^I \) is called (weakly) Pareto admissible for (MOP(I)) if it is admissible and additionally
\[
\mathcal{M}^I(z) \subset \mathcal{U}_{opt,(w)}^I \quad \text{(or equivalently } \mathcal{P}^I(z) \in \mathcal{J}_{opt,(w)}^I)\]
holds. The set of all (weakly) Pareto admissible reference points for (MOP(I)) is denoted by \( \mathcal{Z}_{opt,(w)}^I \).

**Definition 1.6.51.** Let \( I \subset \{1, \ldots, k\} \) be arbitrary. A set \( Z \subset \mathbb{R}^I \) is called (weakly) Pareto sufficient for (MOP(I)) if \( \mathcal{M}^I(Z) = \mathcal{U}_{opt,(w)}^I \), or equivalently \( \mathcal{P}^I(Z) = \mathcal{J}_{opt,(w)}^I \), holds.

**Theorem 1.6.52.** Let Assumption 1.1 be satisfied and \( I \subset \{1, \ldots, k\} \) be arbitrary. Then it holds
\[
\mathcal{M}^I(\mathcal{Z}_{opt,(w)}^I) = \mathcal{U}_{opt,(w)}^I \quad \text{and} \quad \mathcal{P}^I(\mathcal{Z}_{opt,(w)}^I) = \mathcal{J}_{opt,(w)}^I
\]
i.e., the set \( \mathcal{Z}_{opt,(w)}^I \) is (weakly) Pareto sufficient for (MOP(I)).

**Definition 1.6.53.** Let \( I \subset \{1, \ldots, k\} \) be arbitrary. Given the vector \( \hat{d} \in \mathbb{R}_+^k \) and the shifted ideal point \( \hat{y}^{id} \in \mathbb{R}^k \), which were both introduced in Definition 1.6.25, let \( \mathcal{D}_i^I \subset \mathbb{R}^I \) be given by
\[
\mathcal{D}_i^I := \{ y \in \mathbb{R}^I \mid y \geq (\hat{y}^{id})^I, \ y_i = \hat{y}_i^{id} \}
\]
for all \( i \in I \). Then the set \( \mathcal{D}^I \subset \mathbb{R}^I \) is defined by \( \mathcal{D}^I := \bigcup_{i \in I} \mathcal{D}_i \).
Definition 1.6.54. Let Assumption 1.1 be satisfied and \( I \subset \{1, \ldots, k\} \) be arbitrary. Given \( D^I \) as above, we can define the (set-valued) mapping \( G_{D^I}^{\text{opt.(w)}} : \Delta_I \rightarrow D^I \) by
\[
G_{D^I}^{\text{opt.(w)}}(\alpha) = \left\{ y - t\alpha \mid y \in F_{\text{opt.(w)}}(\alpha), t = \min_{i \in I} \frac{y_i - \tilde{y}_i}{\alpha_i} \right\}.
\]

Theorem 1.6.55. Let Assumption 1.1 be satisfied and \( I \subset \{1, \ldots, k\} \) be arbitrary. Then it holds
\[
G_{D^I}^{\text{opt.(w)}}(\Delta_I) = Z_{D^I}^{\text{opt.(w)}}(\Delta_I) \cap D^I.
\]

Lemma 1.6.56. Let Assumption 1.1 be satisfied and \( I \subset \{1, \ldots, k\} \) be arbitrary. Then it holds
\[
G_{D^I}^{\text{opt.(w)}}|_{\Delta_I^>} = G_{D^I}^{\text{opt.(w)}}|_{\Delta_I}.\]
In particular, this implies \( G_{D^I}^{\text{opt.(w)}}(\Delta_k^>) = G_{D^I}^{\text{opt.(w)}}(\Delta_k^>).\)

Theorem 1.6.57. Let Assumption 1.1 be satisfied and \( I \subset \{1, \ldots, k\} \) be arbitrary. Then we have
\[
M^I(\varphi_{D^I}^{\text{opt.(w)}}(\alpha)) = W^I_{\text{opt.(w)}}(\alpha) \quad \text{and} \quad P^I(\varphi_{D^I}^{\text{opt.(w)}}(\alpha)) = F^I_{\text{opt.(w)}}(\alpha)\]
for all \( \alpha \in \Delta_I. \) In particular, this implies
\[
M^I(\varphi_{D^I}^{\text{opt.(w)}}(\Delta_I)) = U^I_{\text{opt.(w)}} \cup \bigcup_{K \subset I} U^K_{\text{opt}(w)} \quad \text{and} \quad P^I(\varphi_{D^I}^{\text{opt.(w)}}(\Delta_I)) = F^I_{\text{opt.(w)}}.
\]
i.e., the set \( \varphi_{D^I}^{\text{opt.(w)}}(\Delta_I) \) is (weakly) Pareto sufficient for (MOP(I)).

The next corollary is needed to formally verify the hierarchical algorithm.

Corollary 1.6.58. Let Assumption 1.1 be satisfied and \( I \subset \{1, \ldots, k\} \) be arbitrary. Then
\[
M^I(\varphi_{D^I}^{\text{opt.(w)}}(\Delta_I^>)) = M^I(\varphi_{D^I}^{\text{opt.(w)}}(\Delta_I^>)) \supset U^I_{\text{opt.(w)}} \cup \bigcup_{K \subset I} U^K_{\text{opt}(w)} \cup \bigcup_{K \subset I} U^K_{\text{opt}(w)}.
\]

Proof. By Lemma 1.6.56, we have
\[
M^I(\varphi_{D^I}^{\text{opt.(w)}}(\Delta_I^>)) = M^I(\varphi_{D^I}^{\text{opt.(w)}}(\Delta_I^>).
\]
From Corollary 1.3.6, we infer
\[
U^I_{\text{opt}(w)} \cup \bigcup_{K \subset I} U^K_{\text{opt}(w)} = U^I_{\text{opt}(w)} \cup \bigcup_{K \subset I} U^K_{\text{opt}(w)}.
\]
Moreover, we can use Theorem 1.6.57 and Corollary 1.5.21 to conclude
\[
M^I(\varphi_{D^I}^{\text{opt.(w)}}(\Delta_I^>)) = W^I_{\text{opt}}(\Delta_I^>) \supset U^I_{\text{opt}(w)} \cup \bigcup_{K \subset I} U^K_{\text{opt}(w)}.
\]
Altogether, this proves (1.6.13).
\textbf{Definition 1.6.59.} Let Assumption 1.1 be satisfied and \( I \subset \{1, \ldots, k\} \) be arbitrary. Then we define the function \( \mathcal{H}^I \) by

\[ \mathcal{H}^I : (\tilde{J}(\mathcal{U}_{ad}) + \mathbb{R}_+^I)^c \to \Delta_I, \ z \mapsto \frac{\mathcal{P}^I(z) - z}{\|\mathcal{P}^I(z) - z\|_1}. \]

Moreover, we define the function \( \mathcal{H}_{opt,(w)}^{D^I} : \mathcal{G}_{opt,(w)}^{D^I}(\Delta_I) \to \Delta_I \) by

\[ \mathcal{H}_{opt,(w)}^{D^I} = \mathcal{H}^I |_{\mathcal{G}_{opt,(w)}^{D^I}(\Delta_I)}. \]

\textbf{Theorem 1.6.60.} Let Assumption 1.1 be satisfied and \( I \subset \{1, \ldots, k\} \) with \(|I| > 1\) as well as \( z \in D^I \) be arbitrary. Then the following statements are equivalent:

(i) \( z \notin \mathcal{G}_{opt}(\Delta_I^\gamma) = \mathcal{G}_{opt,w}(\Delta_I^\gamma) \),

(ii) \( \exists z \in \mathcal{G}_{opt,w}(\partial \Delta_I) : \exists \beta \in \mathcal{H}_{opt,w}(z) \cap \mathbb{R}_+^I: z = z + \beta \).

\subsection{1.6.4.2 The Hierarchical Algorithm}

In this section, we will describe and analyze a hierarchical algorithm for solving \((\text{MOP})\) by the ERPM. To this end, we need the following theorem, in which we show that the solutions to all subproblems \((\text{MOP}(K))\) for all \( K \subset I \) can be used to compute the set \( \mathcal{G}_{opt,w}(\partial \Delta_I) \), which is needed in the characterization of the reference points \( \mathcal{G}_{opt}(\Delta_I^\gamma) \) in Corollary 1.6.60. This seems to be trivial at first. However, note that given any \( K \subset I \), it holds \( \mathcal{G}_{opt}^{D^K}(\Delta_K^\gamma) \subset \mathbb{R}^K \), whereas we have \( \mathcal{G}_{opt,w}(\partial \Delta_I) \subset \mathbb{R}^I \). Thus, we need a way to 'lift' reference points from \( \mathbb{R}^K \) to \( \mathbb{R}^I \).

\textbf{Theorem 1.6.61.} Let Assumption 1.1 be satisfied and \( I \subset \{1, \ldots, k\} \) be arbitrary. Assume that \((\text{MOP}(K))\) has been solved by the ERPM for all \( K \subset I \), i.e., in particular, for all \( K \subset I \) the set

\[ \{(u, z) \in \mathcal{U}_{opt,w}^K \times \mathcal{G}_{opt}^{D^K}(\Delta_K^\gamma) \mid u \in \mathcal{M}^K(z)\} \]

has been computed. Then it is possible to compute \( \mathcal{G}_{opt,w}(\partial \Delta_I) \).

\textbf{Proof.} Let \( I \subset \{1, \ldots, k\} \) and \( z \in \mathcal{G}_{opt,w}(\partial \Delta_I) \) be arbitrary. Then by definition, there is \( \alpha \in \partial \Delta_I \) and \( y = \tilde{J}(\mathcal{U}) \in \mathcal{F}_{opt,w}^I(\alpha) \) with \( z = y - t\alpha \), where \( t = \min_{i \in I} (y_i - \tilde{y}_i^{id})/\alpha_i \). Define \( K := \{i \mid \alpha_i > 0\} \subset I \). From Lemma 1.5.17 and Theorem 1.5.20, we infer \( \bar{u} \in \mathcal{W}_{opt}(\alpha^K) \), so that \( y^K = J^K(\bar{u}) \in \mathcal{F}_{opt}(\alpha^K) \). Moreover, we have

\[ t = \min_{i \in I} \frac{y_i - \tilde{y}_i^{id}}{\alpha_i} = \min_{i \in K} \frac{y_i - \tilde{y}_i^{id}}{\alpha_i}, \]

since \( \alpha_i = 0 \) for all \( i \in I \setminus K \). Thus,

\[ \bar{z} := z^K = y^K - t\alpha^K \in \mathcal{G}_{opt}^{D^K}(\alpha^K), \]
so that $\mathcal{P}^K(\tilde{z}) = y^K = \hat{j}^K(\tilde{u})$ by Theorem 1.6.46 (ii), which implies that $\tilde{u} \in \mathcal{M}^K(\tilde{z})$. By assumption, we have computed the tupel $(\tilde{u}, \tilde{z})$ and can also compute $\hat{j}^K(\tilde{u})$. Now $z$ can be constructed by setting

$$z_i = \begin{cases} \tilde{z}_i, & \text{if } i \in K, \\ \hat{J}_i(\tilde{u}), & \text{if } i \in I \setminus K. \end{cases}$$

To conclude, we have just shown that every $z \in G^I(\partial \Delta_I)$ can be constructed if the sets

$$\{(u, z) \in U^K_{\text{opt, w}} \times G^K(\Delta_K) \mid u \in \mathcal{M}^K(z)\}$$

have been computed for all $K \subset I$, which is what we had to prove.

In the proof of the previous theorem, we showed how to 'lift' reference points from $G^K(\Delta_K) \subset \mathbb{R}^K$ to $G^I_{\text{opt, w}}(\partial \Delta_I) \subset \mathbb{R}^I$. In particular, it was proven that this lifting does not change the solution of the corresponding ERPP. This property is summarized in the following corollary.

**Corollary 1.6.62.** Let Assumption 1.1 be satisfied and $K, I \subset \{1, \ldots, k\}$ with $K \subset I$ be arbitrary. Furthermore, let $\alpha \in \Delta_K$, $z \in G^K_{\text{opt}}(\alpha) \subset G^K_{\text{opt}}(\Delta_K)$ and $\tilde{u} \in \mathcal{M}^K(z)$ be arbitrary. Then we have $\tilde{u} \in \mathcal{M}^I(\tilde{z})$ for the reference point $\tilde{z} \in G^I_{\text{opt, w}}(\tilde{\alpha}) \subset G^I_{\text{opt, w}}(\partial \Delta_I)$ defined by

$$\tilde{z}_i = \begin{cases} z_i, & \text{if } i \in K, \\ \hat{J}_i(\tilde{u}), & \text{if } i \in I \setminus K, \end{cases}$$

where the weight $\tilde{\alpha} \in \partial \Delta_I$ is given by

$$\tilde{\alpha}_i = \begin{cases} \alpha_i, & \text{if } i \in K, \\ 0, & \text{if } i \in I \setminus K. \end{cases}$$

In particular, $\mathcal{P}^I(\tilde{z}) = \hat{j}^I(\tilde{u})$ holds.

**Proof.** Let $K, I \subset \{1, \ldots, k\}$ with $K \subset I$, $\alpha \in \Delta_K$, $z \in G^K_{\text{opt}}(\alpha) \subset G^K_{\text{opt}}(\Delta_K)$ and $\tilde{u} \in \mathcal{M}^K(z)$ be arbitrary. From Theorem 1.6.57, we can conclude $\tilde{u} \in W^K_{\text{opt}}(\alpha)$, and Lemma 1.5.17 implies that $\tilde{u} \in W^I_{\text{opt, w}}(\tilde{\alpha})$ if we define $\tilde{\alpha}$ as in (1.6.15). With the same arguments as in the proof of Theorem 1.6.61, we can show that $\tilde{z} \in G^I_{\text{opt, w}}(\tilde{\alpha}) \subset G^I_{\text{opt, w}}(\partial \Delta_I)$ holds if we construct $\tilde{z}$ as in (1.6.14). Now from Theorem 1.6.46 (ii), it follows that $\mathcal{P}^I(\tilde{z}) = \hat{j}^I(\tilde{u})$ and thus, also $\tilde{u} \in \mathcal{M}^I(\tilde{z})$. 

Now we have all the tools to mathematically describe and analyze the hierarchical algorithm for computing the Pareto front by using the ERPM, cf. Algorithm 1.

For this algorithm, we can show that it solves indeed the problem (MOP).
1.6 The Euclidean Reference Point Method

Algorithm 1 Solving (MOP) by the ERPM
1: for $j = 1 : k$ do
2: Set $I := \{j\}$;
3: Compute $U_{opt,w}(I) := \{u | u \text{ minimizes } J_j\}$;
4: Compute $y^i$ and $\hat{y}^i$;
5: Set $U(I) = \{(u, \hat{G}^D(I)) | u \in \hat{U}\}\}
6: end for
7: for $i = 2 : k$ do
8: for all $I \subset \{1, \ldots, k\}$ with $|I| = i$ do
9: Initialize $U_{opt,w}(I) = \bigcup_{K \subseteq I} U_{opt,w}(K)$ and $U(I) = \emptyset$;
10: Compute $G^D_{opt,w}(\Delta I)$ by applying Theorem 1.6.61;
11: Use Theorem 1.6.60 to compute the set $G^D_{opt}(\Delta I)$;
12: for all $z \in G^D_{opt}(\Delta I)$ do
13: Solve (ERPP($I, z$)), i.e., compute $M^I(z)$;
14: Set $U_{opt,w}(I) \leftarrow U_{opt,w}(I) \cup M^I(z)$;
15: Set $U(I) \leftarrow U(I) \cup \{(u, z) | u \in M^I(z)\}$;
16: end for
17: end for
18: end for
19: if computeParetoFront == true then
20: Remove all $\hat{u} \in U_{opt,w}(\{1, \ldots, k\})$ with $\hat{u} \notin U_{opt}$ by a non-dominance test;
21: end if

Theorem 1.6.63. Let Assumption 1.1 be satisfied. Then Algorithm 1 solves (MOP) in the sense that we have

$$U_{opt,w}(I) = U^I_{opt,w},$$

$$U(I) = \{(u, z) \in U_{opt}^I \times G^D_{opt}(\Delta I) | u \in M^I(z)\}$$

for all $I \subset \{1, \ldots, k\}$.

Proof. We prove the statement by induction over the cardinality $i$ of the set $I$.

$i = 1$: For $I = \{j\}$ for any $j \in \{1, \ldots, k\}$, we have

$$\hat{U}_{opt,w} = \{u \in U_{ad} | u \text{ minimizes } J_j\},$$

$$\{(u, z) \in U_{opt}^I \times G^D_{opt}(\Delta I) | u \in M^I(z)\} = \{(u, z) | u \text{ minimizes } J_j, z = G^D_{opt}(1)\}.$$

These sets are computed in lines 3 and 5 of Algorithm 1.

$(1, \ldots, i - 1) \rightarrow i$ (for $2 \leq i \leq k$): Let $I \subset \{1, \ldots, k\}$ with $|I| = i$ be arbitrary. By induction assumption, we have

$$\hat{U}_{opt,w}(K) = U^K_{opt,w},$$

$$U(I) = \{(u, z) \in U_{opt}^I \times G^D_{opt}(\Delta I) | u \in M^I(z)\}$$
for every $K \subseteq I$. Then by Theorem 1.6.61, the set $G_{\text{opt},w}^D(\partial \Delta_I)$ can be computed. Using the characterization from Theorem 1.6.60, we can obtain $G_{\text{opt},w}^D(\Delta_I^>) = G_{\text{opt},w}^D(\Delta_I^>)$ from $G_{\text{opt},w}^D(\partial \Delta_I)$.

By induction assumption, the set $\tilde{U}_{\text{opt},w}(I)$ is initialized by $\tilde{U}_{\text{opt},w}(I) = \bigcup_{K \subseteq I} U_K$ in line 9. From Corollary 1.6.58, we infer

$$M_I \left( G_{\text{opt},w}^D(\Delta_I^>) \right) \supset U_{\text{opt},w} \setminus \bigcup_{K \subseteq I} U_K = U_{\text{opt},w} \setminus \bigcup_{K \subseteq I} U_K.$$

Thus, by computing and adding $M_I(z)$ to $\tilde{U}_{\text{opt},w}(I)$ for all $z \in G_{\text{opt},w}^D(\Delta_I^>)$ in lines 13 and 14, we obtain $\tilde{U}_{\text{opt},w}(I) = U_{\text{opt},w}$.

Moreover, adding the tuples $(u,z)$ to the set $UZ(I)$ for all $z \in G_{\text{opt},w}^D(\Delta_I^>)$ and all $u \in M_I(z)$ in lines 13 and 15 results in

$$UZ(I) = \{ (u,z) \in U_{\text{opt}} \times G_{\text{opt},w}^D(\Delta_I^>) \mid u \in M_I(z) \},$$

which concludes the proof.

**Remark 1.6.64.** Let us summarize the results for the ERPM in the terminology of Section 1.4.1: We choose the family of scalarization functions $(g_z)_{z \in G_{\text{opt},w}^D(\Delta_k)}$ for solving (MOP) by the ERPM. In Algorithm 1, these scalarization functions are obtained in a hierarchical way. Moreover, we have shown the Pareto admissibility and sufficiency of this family of scalarization functions. In general, we do not have a bijection between the set of reference points $G_{\text{opt},w}^D(\Delta_k)$ and the (weak) Pareto front $\delta_{\text{opt},w}$, since it might happen that $P(z_1) = P(z_2)$ holds for some reference points $z_1, z_2 \in G_{\text{opt},w}^D(\Delta_k)$ with $z_1 \neq z_2$. Nevertheless, the method is as efficient as possible, since we cannot know in advance if and for which reference points this behavior occurs.

**1.6.4.3 Numerical Implementation**

In Theorem 1.6.63, we could show that Algorithm 1 is capable of solving (MOP) by the ERPM. In this section, we will focus on the numerical implementation of the algorithm. As described in Section 1.4.2, by solving (MOP) numerically we want to obtain approximations

$$U_{\text{opt},w}^{\text{num}} = \{ \tilde{u}^1, \ldots, \tilde{u}^N \} \quad \text{and} \quad \delta_{\text{opt},w}^{\text{num}} = \{ \tilde{J}(\tilde{u}^1), \ldots, \tilde{J}(\tilde{u}^N) \}$$

of $U_{\text{opt},w}$ and $\delta_{\text{opt},w}$, respectively. Moreover, three quality criteria – coverage, uniformity and cardinality – should be guaranteed by the numerical algorithm. These points will be discussed in the end of this section.

We start with describing a numerical method for generating the reference points $G_{\text{opt},w}^D(\Delta_I^>)$ in line 11, and show that this method works efficiently in practice. To this end, we require a slight reformulation of Theorem 1.6.60.
Corollary 1.6.65. Let Assumption 1.1 be satisfied and \( I \subset \{1, \ldots, k\} \) with \(|I| > 1\) as well as \( z \in D^I \) be arbitrary. Then the following statements are equivalent:

(i) \( z \not\in G^D_{\text{opt}}(\Delta^\gamma_K) = G^D_{\text{opt}}(\Delta^\gamma_K) \),

(ii) \( \exists \alpha \in \partial \Delta_I : \exists \bar{z} \in G^D_{\text{opt}, w}(\alpha) : z^K = \bar{z}^K \& z^{I \setminus K} \geq \bar{z}^{I \setminus K} \, \text{for} \, \bar{K} := \{i \mid \alpha_i > 0\} \),

(iii) \( \exists K \subseteq I : \exists (\bar{u}, \bar{z}) \in \mathcal{UZ}(K) : z^K = \bar{z} \& z^{I \setminus K} \geq \bar{J}^{I \setminus K}(\bar{u}) \).

Proof. The equivalence (i) \( \iff \) (ii) follows directly from Theorem 1.6.60.

(ii) \( \Rightarrow \) (iii): Assume that there is \( \alpha \in \partial \Delta_I \) and \( \bar{z} \in G^D_{\text{opt}, w}(\alpha) \) with \( z^K = \bar{z}^K \) and \( z^{I \setminus K} \geq \bar{z}^{I \setminus K} \) for \( K := \{i \mid \alpha_i > 0\} \). By definition of \( G^D_{\text{opt}, w}(\alpha) \), there is \( \bar{u} \in W^I_{\text{opt}, w}(\alpha) \) with

\[
\bar{z} = \bar{J}^I(\bar{u}) - \min_{i \in I} \frac{\bar{J}_i(\bar{u}) - \bar{g}^d_i}{\alpha_i}.
\]

As in the proof of Theorem 1.6.61, we can show that

\[
\bar{u} \in W_{\text{opt}}(\alpha^K) \quad \text{and} \quad \bar{z} \in G^D_{\text{opt}}(\alpha^K) \subset G^D_{\text{opt}}(\Delta^\gamma_K).
\]

Theorem 1.6.46 (ii) implies that \( \bar{u} \in M^K(\bar{z}^K) \). Therefore, by Theorem 1.6.63, it holds \((\bar{u}, \bar{z}) \in \mathcal{UZ}(K)\). Finally, due to \( z^{I \setminus K} \geq \bar{z}^{I \setminus K} = \bar{J}^{I \setminus K}(\bar{u}) \), we know that (iii) is satisfied.

(iii) \( \Rightarrow \) (ii): Assume that there is \( K \subseteq I \) and \((\bar{u}, \bar{z}) \in \mathcal{UZ}(K)\) such that \( z^K = \bar{z} \) and \( z^{I \setminus K} \geq \bar{J}^{I \setminus K}(\bar{u}) \). By Theorem 1.6.63, it holds \( \bar{z} \in G^D_{\text{opt}}(\alpha) \) for some \( \alpha \in \Delta^\gamma_K \) and \( \bar{u} \in M^K(\bar{z}) \). From Corollary 1.6.62, we can induce that \( \bar{z} \in G^D_{\text{opt}, w}(\hat{\alpha}) \subset G^D_{\text{opt}, w}(\partial \Delta_I) \), where \( \bar{z} \) and \( \hat{\alpha} \) are set as in (1.6.14) and (1.6.15), respectively. In particular, it holds \( K = \{i \mid \hat{\alpha}_i > 0\} \). Moreover, we have

\[
z^K = \bar{z} = \hat{z}^K \quad \text{and} \quad z^{I \setminus K} \geq \hat{J}^{I \setminus K}(\hat{u}) = \hat{z}^{I \setminus K},
\]

which shows that (ii) holds.

By using the nadir objective point \( y^{\text{nad}, I} \), which can be computed by using the solutions of subproblems of \( \text{(MOP(I))} \) as shown in Theorem 1.3.8, we can show that \( G^D_{\text{opt}}(\Delta^\gamma_K) \) is contained in a bounded set, which can be computed by only using information of previously solved subproblems of \( \text{(MOP(I))} \).

Lemma 1.6.66. Let Assumption 1.1 be satisfied and \( I \subset \{1, \ldots, k\} \) be arbitrary. Then we have

\[
G^D_{\text{opt}}(\Delta^\gamma_K) \subset \bigcup_{i \in I} \{z \in D^I_i \mid \forall j \in I \setminus \{i\} : z_j \leq y^{\text{nad}, I}_j\}.
\]

Proof. Let \( I \subset \{1, \ldots, k\} \) and \( z \in G^D_{\text{opt}}(\Delta^\gamma_K) \) be arbitrary. Then there is \( i \in I \) with \( z \in D^I_i \), and \( \alpha \in \Delta^\gamma_K \) as well as \( \bar{u} \in W_{\text{opt}}(\alpha) \subset U_{\text{opt}}^I \) such that

\[
z = \hat{J}^I(\bar{u}) - \min_{i \in I} \frac{\hat{J}_i(\bar{u}) - \hat{g}^d_i}{\alpha_i}.
\]
holds. For any \( j \in I \setminus \{ i \} \), we can easily conclude
\[
    z_j = \hat{J}_j(\bar{u}) - \min_{i \in I} \frac{\hat{J}_i(\bar{u}) - \hat{y}_{id}^j}{\alpha_i} \leq \sup_{u \in \Omega_{\text{opt}}} \hat{J}_j(u) = y_{\text{opt}}^{\text{nad},I}^j,
\]
which is what we had to show. \( \Box \)

As a next step, we introduce a grid on \( D^f \), which covers the possible range of the set \( G_{\text{opt}}(\Delta^f_i) \).

**Definition 1.6.67.** Let \( I \subset \{1, \ldots, k\} \) be arbitrary. For a given grid size \( h > 0 \) and any \( i \in I \), we define
\[
    Z_i^{h,I} := \left\{ z \in D^f \mid \forall j \in I \setminus \{i\} : \left( \exists k \geq 0 : z_j = \hat{y}_{id}^j + \frac{h}{2} + kh \right) \& \left( z_j \leq y_{\text{opt}}^{\text{nad},I}^j \right) \right\}.
\]
Furthermore, we set
\[
    Z_i^{h,1} := \bigcup_{i \in I} Z_i^{h,I}.
\]
If \( I = \{1, \ldots, k\} \), we write \( Z^h := Z_i^{h,1} \).

Now we can formulate the numerical algorithm for solving (MOP) by the ERPM (see Algorithm 2).

The difference to Algorithm 1 is that the reference points are only chosen from the grid \( Z_i^{h,I} \) in each iteration. This leads to subsets \( \Omega_{\text{opt},w}(I) \) and \( \Omega_{\text{num}}(I) \) of \( \Omega_{\text{opt},w}(I) \) and \( \Omega_{\text{opt}}(I) \), respectively. The sets \( \Omega_{\text{num}}(I) \) are then used in line 10 to remove reference points that do not lie in \( G_{\text{opt}}(\Delta^f_i) \). In the following two results, we show that this procedure results in the set \( G_{\text{opt}}(\Delta^f_i) \cap Z_i^{h,I} \), i.e., all relevant reference points of the grid \( Z_i^{h,I} \) are selected.

**Theorem 1.6.68.** Let Assumption 1.1 be satisfied and \( I \subset \{1, \ldots, k\} \) with \( |I| > 1 \) as well as \( z \in Z_i^{h,I} \) be arbitrary. Then the following statements are equivalent:

(i) \( z \notin G_{\text{opt}}(\Delta^f_i) = G_{\text{opt},w}(\Delta^f_i) \),

(ii) \( \exists K \subseteq I : \exists (\bar{u}, \bar{z}) \in \Omega_{\text{num}}(K) : z^K = \bar{z} \& z^{I \setminus K} \geq J^{I \setminus K}(\bar{u}) \).

**Proof.** We prove this statement by induction over the cardinality \( i \) of the set \( I \).

\( i = 2 \): In this case, we can use \( \Omega_{\text{num}}(K) = \Omega_{\text{opt}}(K) \) for all \( K \subseteq I \), since \( |K| = 1 \). Therefore, the equivalence follows directly from Corollary 1.6.65.

\( (2, \ldots, i-1) \rightarrow i \) (for \( 3 \leq i \leq k \)): Let \( I \subset \{1, \ldots, k\} \) with \( |I| = i \) be arbitrary.

(ii) \( \Rightarrow \) (i): In view of Corollary 1.6.65, this can be concluded from \( \Omega_{\text{num}}(K) \subseteq \Omega_{\text{opt}}(K) \) for all \( K \subset \{1, \ldots, k\} \).

(i) \( \Rightarrow \) (ii): Assume that \( z \in Z_i^{h,I} \) but \( z \notin G_{\text{opt}}(\Delta^f_i) = G_{\text{opt},w}(\Delta^f_i) \). By Corollary 1.6.65, there is \( K \subseteq I \) and \( (\bar{u}, \bar{z}) \in \Omega_{\text{opt}}(K) \) with \( z^K = \bar{z} \) and \( z^{I \setminus K} \geq J^{I \setminus K}(\bar{u}) \). We only need
Furthermore, from Section 1.4.2, this follows directly from Theorem 1.6.68.

**Proof.**

Let us now discuss the quality criteria of the numerical implementation of Corollary 1.6.69. Let \( z \) be arbitrary. For the set \( \mathcal{Z}^{\text{num}}(I) \), which is computed in line 10 of Algorithm 2, we have that

\[
\mathcal{Z}^{\text{num}}(I) = \mathcal{Z}^{h,I} \cap \mathcal{G}^{D^I}(\Delta^\gamma_K).
\]

Therefore, the reference point \( \bar{z} \) lies in the set \( \mathcal{Z}^{\text{num}}(K) \) in line 10 of Algorithm 2 inside the iteration for the set \( K \). Consequently, we have \((\bar{u}, \bar{z}) \in \mathcal{U} \mathcal{Z}^{\text{num}}(K)\) which concludes the proof.

**Corollary 1.6.69.** Let Assumption 1.1 be satisfied and \( I \subset \{1, \ldots, k\} \) with \(|I| > 1\) be arbitrary. For the set \( \mathcal{Z}^{\text{num}}(I) \), which is computed in line 10 of Algorithm 2, we have that

\[
\mathcal{Z}^{\text{num}}(I) = \mathcal{Z}^{h,I} \cap \mathcal{G}^{D^I}(\Delta^\gamma_I).
\]

**Proof.** This follows directly from Theorem 1.6.68.

**Remark 1.6.70.** Let us now discuss the quality criteria of the numerical implementation from Section 1.4.2.

(i) **Coverage:** The Lipschitz continuity of the mapping \( \mathcal{P} \), which was shown in Theorem 1.6.7, allows us to lead the coverage of the Pareto front back to the
problems, we use the built-in Matlab-function \( \text{fmincon} \). The first step of solving (MOP) is to compute the minimizers of the cost functions involving three cost functions \( \hat{J}_1, \hat{J}_2, \hat{J}_3 \): \([-1,1]^5 \subset \mathbb{R}^5 \rightarrow \mathbb{R} \). In this case, the admissible space is given by \( \mathcal{U} = \mathbb{R}^5 \) and the admissible set by \( \mathcal{U}_{\text{ad}} = [-1,1]^5 \). The three cost functions are defined by

\[
\hat{J}_1(u) := \frac{1}{5} \left( u_1^2 + 5u_2^2 + 2u_3^2 + (u_4 - 1)^4 + 2(u_5 + 0.2)^4 \right), \\
\hat{J}_2(u) := \frac{1}{100} \left( 3(x_1 - 2)^4 + 2u_2^2 + (u_3 + 3)^4 + u_4^2 + 3(u_5 - 0.5)^2 \right), \\
\hat{J}_3(u) := \frac{1}{100} \left( (u_1 + 5)^2 + 4(u_2 - 2)^4 + u_3^2 + u_4^2 + u_5^4 \right).
\]

It can be easily seen that the cost functions are strictly convex. In particular, it holds \( \mathcal{U}_{\text{opt}} = \mathcal{U}_{\text{opt,w}} \) and \( J_{\text{opt}} = J_{\text{opt,w}} \) by Lemma 1.2.8. We use the shifting vector \( \hat{d} := 0.4 \cdot (1,1,1)^T \in \mathbb{R}^3 \) and the grid size \( h = 0.1 \). For solving the occurring optimization problems, we use the built-in Matlab-function \text{fmincon}.}

The first step of solving (MOP) is to compute the minimizers of the cost functions \( \hat{J}_1, \hat{J}_2, \hat{J}_3 \), cf. line 3, and construct the sets \( U\mathcal{Z}^{\text{num}}(\{1\}), U\mathcal{Z}^{\text{num}}(\{2\}) \) and \( U\mathcal{Z}^{\text{num}}(\{3\}) \) in line 5.

With these information, we can continue to solve all subproblems with two cost functions. We illustrate the procedure for \( I = \{1,2\} \). For computing the new reference points, we construct the grid \( \mathcal{Z}^{h,I} \) on \( D^I \). This situation is shown in Figure 1.9(a). We see that by applying the condition from line 10, no reference point is removed from the grid, i.e., all grid points lie in \( \mathcal{G}^{D^I}_{\text{opt}}(\Delta^I) \) so that \( \mathcal{Z}^{\text{num}}(I) = \mathcal{Z}^{h,I} \). Solving (ERPP(I,z)) for all \( z \in \mathcal{Z}^{h,I} \) results in an approximation \( J_{\text{opt}}^{I,\text{num}} \) of the Pareto front \( J_{\text{opt}}^{I} \), cf. Figure 1.9(b). Whereas
Figure 1.9: (a) & (b): Solving the subproblem (MOP(I)) for \( I = \{1, 2\} \) by Algorithm 2. (c): Results for the iterative method from [BBV16, BBV17, Bee19, POBD19].

The coverage of the Pareto front is good, the distribution of the approximation points is not completely uniform. There are two reasons for this: The first one is problem-specific. Around the point \((0.65, 0.1)\), there seems to be a sudden change in the slope of the Pareto front, i.e., the curvature is quite high at that point. This entails that the solutions of the corresponding ERPPs are closer to each other. The second reason is that the reference points are generated on the set \( D^I \), which consists of two orthogonal shifted coordinate lines. Therefore, although the grid in \( D^I \) was chosen with a constant grid size of \( h = 0.1 \), the distance between neighboring reference points close to the intersection point \( \tilde{y}^{id,I} \) of the coordinate lines is smaller. Moreover, the coordinate lines do not approximate the slope of the Pareto front very well close to \( \tilde{y}^{id,I} \), which also results in a less uniform approximation of the middle part of the Pareto front.

At this point, we refer to the algorithms presented in [BBV17, Del08, Bee19, POBD19, RBW+09], where the reference points for two cost functions were chosen iteratively by taking into account first-order information of the Pareto front. Indeed, using this approach...
yields a more uniform approximation of the Pareto front, see Figure 1.9(c). However, this method cannot be canonically generalized to more than two cost functions. Moreover, it is not possible to combine the two approaches by using the iterative generation of reference points for two cost functions and the generation of reference points used in Algorithm 2 for more than two cost functions. The reason is that for Algorithm 2 to work it is crucial that all reference points are chosen on the grid $Z_{\Delta_3}$. Otherwise, the selection procedure for the new reference points would not work, and, in particular, the result from Corollary 1.6.69 would not hold anymore.

After having solved all subproblems with two cost functions, we come to solving (MOP). For generating the respective reference points in $G_{\text{opt}}^D(\Delta_3^>)$, we use the condition from Theorem 1.6.68 (ii), which is based in the lifting approach introduced in Theorem 1.6.61. This situation is shown in Figure 1.10(a). Note that in this figure, the discretization of the set $G_{\text{opt}}^D(\partial \Delta_3)$ is obtained by lifting the reference points of all subproblems. By using the solutions of all subproblems, the nadir objective point $y_{\text{nad}}$ is computed with the help of Theorem 1.3.8 and the grid $Z_h$ is constructed according to Definition 1.6.67. Then the condition from Theorem 1.6.68 (ii) is used for all $K \subseteq \{1, 2, 3\}$ to subsequently remove all grid points $z \notin G_{\text{opt}}^D(\Delta_3^>)$. This is shown in detail for the set $D_1$ in the Figures 1.10(b)–(e). Note that this only requires componentwise comparisons between the elements of $Z_h$ and the elements of the discretization of the set $G_{\text{opt}}^D(\partial \Delta_3)$. Thus, this procedure is cheap in practice. The resulting set of all reference points $Z_{\text{num}}(\{1, 2, 3\}) = G_{\text{opt}}^D(\Delta_3^>) \cap Z_h$ is shown in Figure 1.10(f).

Finally, solving (ERPP($z$)) for all $z \in Z_{\text{num}}(\{1, 2, 3\})$ results in the approximation $J_{\text{num}}^{\text{opt}}$ of the Pareto front, cf. Figure 1.11(a). In total, the Pareto front is approximated with 665 points. We observe that all parts of the Pareto front are well approximated, which implies a good coverage of the Pareto front and is confirmed by the coverage value of 0.07, cf. Table 1.1. For the computation of this value, we took a fine reference approximation Pareto front computed by the Pascoletti-Serafini method (see Section 1.7) with around 160,000 approximation points. However, as we have seen for the subproblem $I = \{1, 2\}$, the approximation is not entirely uniform for all parts of the Pareto front. Again, the reason is partly due to the shape of the Pareto front, i.e., due to the fact that it has parts with a large curvature, and partly due to the choice of reference points on the union of shifted coordinate planes $D$. Nevertheless, the method still reaches a uniformity value of 1.88, which is quite close to the ideal value of 1. In [Eic08, KSd15, MAL12, MGGS09], the reference points are not chosen on such a set $D$, but on a $k-1$ dimensional hyperplane $H$. It is theoretically possible to repeat the investigations from the Sections 1.6.3 and 1.6.4 using such a hyperplane, which avoids the kink in the set of reference points and leads to a more uniform approximation of the Pareto front. But especially for more than three cost functions, the numerical selection of the reference points becomes much more difficult for this approach, which is why we decided to use the set $D$ instead.

In Table 1.1, we see additionally the values of the coverage, the uniformity and the cardinality of the approximation of the Pareto front for a varying grid size $h$. The coverage is smaller than the grid size for all tested values, which can be explained by the fact that the function $P$ is Lipschitz continuous with Lipschitz constant 1. In particular, the coverage
1.6 The Euclidean Reference Point Method

(a) Computation of the set $G_{D_\text{opt}}(\partial \Delta_3)$.

(b) The sets $G_{D_\text{opt}}(\partial \Delta_3) \cap D_1$ and $Z_h^\text{opt}$.

(c) Using Theorem 1.6.68 (ii) with $K = \{1\}$ for removing reference points.

(d) Using Theorem 1.6.68 (ii) with $K = \{1, 2\}$ for removing reference points.

(e) Using Theorem 1.6.68 (ii) with $K = \{1, 3\}$ for removing reference points.

(f) Resulting reference points $G_{D_\text{opt}}^D(\Delta_3^z) \cap Z_h$.

Figure 1.10: Creating the set of reference points $G_{D_\text{opt}}^D(\Delta_3^z) \cap Z_h$ as described in line 10 of Algorithm 2.
is of the order of the grid size as stated in Remark 1.6.70. Since the reference point set for an MOP with three cost functions is two-dimensional, it is clear that the number of approximation points grows quadratically in the inverse of the grid size.

Finally, we compare the solution we obtained by using the ERPM for the grid size $h = 0.1$ to the solution of the WSM, for which the set $\Delta_3$ was discretized with $n = 100$ points per dimension leading to 5050 equidistantly distributed weights, cf. Figure 1.11(b). Although around eight times more approximation points are used for the WSM, one immediately sees that the ERPM is superior to the WSM in terms of coverage, since especially the upper and the lower part of the Pareto front are not approximated well by the WSM. This is confirmed by the coverage value of 0.29 for the WSM, whereas the ERPM has a value of 0.07. Moreover, the uniformity of the approximation is also much worse than the one of the ERPM (WSM: 16.33, ERPM: 1.88).

In Table 1.2, the coverage, uniformity and cardinality of the approximation by the WSM are displayed in dependence of the number of discretization points per dimension of the set $\Delta_3$. Even for $n = 200$, which leads to 20100 equidistantly distributed weights, the coverage of the WSM is worse than the one of the ERPM for a grid size of $h = 0.2$ and 188 approximation points. Moreover, the uniformity of the approximation strongly deteriorates with increasing $n$. This confirms the deficiencies of the WSM described in the

Table 1.1: Coverage, uniformity and cardinality for the approximation of the Pareto front of $(\hat{J}_1, \hat{J}_2, \hat{J}_3)$ obtained by the ERPM for different grid sizes $h$.

<table>
<thead>
<tr>
<th>$h$</th>
<th>Coverage</th>
<th>Uniformity</th>
<th>Cardinality</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.5</td>
<td>0.34</td>
<td>3.32</td>
<td>42</td>
</tr>
<tr>
<td>0.2</td>
<td>0.14</td>
<td>2.23</td>
<td>188</td>
</tr>
<tr>
<td>0.1</td>
<td>0.07</td>
<td>1.88</td>
<td>662</td>
</tr>
<tr>
<td>0.05</td>
<td>0.035</td>
<td>1.78</td>
<td>2486</td>
</tr>
<tr>
<td>0.02</td>
<td>0.014</td>
<td>1.70</td>
<td>14944</td>
</tr>
</tbody>
</table>

Figure 1.11: Approximations of the Pareto front $\mathcal{J}_{\text{opt}}$ obtained by the ERPM and the WSM.
\begin{table}[h]
\centering
\begin{tabular}{l|c|c|c|c|c}
  & \textit{n = 10} & \textit{n = 20} & \textit{n = 50} & \textit{n = 100} & \textit{n = 200} \\
\hline
Coverage  & 0.61 & 0.37 & 0.31 & 0.29 & 0.22 \\
Uniformity & 4.94 & 5.31 & 10.16 & 16.33 & 25.28 \\
Cardinality & 55 & 210 & 1275 & 5050 & 20100 \\
\end{tabular}
\caption{Coverage, uniformity and cardinality for the approximation of the Pareto front of $(\hat{J}_1, \hat{J}_2, \hat{J}_3)$ obtained by the WSM for different numbers of weights per dimension.}
\end{table}

Both the WSM and the ERPM are only applicable – from a theoretical point of view – to convex MOPs. To be more precise, both methods rely on the condition that the set $\hat{J}(U_{ad}) + \mathbb{R}_+^k$ is convex, cf. Theorem 1.5.4. However, in many applications, the underlying problem is non-convex so that convexity of $\hat{J}(U_{ad}) + \mathbb{R}_+^k$ cannot be guaranteed, and is, in fact, not fulfilled in many cases. For these problems, the so-called Pascoletti-Serafini method (PSM), which was introduced in [PS84], is more suitable, since it works directly with the ordering cone, so that weak Pareto optimality of the solutions can be directly concluded without any assumptions on the cost functions. In its most general formulation, the Pascoletti-Serafini problem (PSP) reads

$$\begin{align*}
\min_{(u, t)} & \quad t \\
\text{s.t.} & \quad u \in U_{ad}, \ t \in \mathbb{R}, \\
& \quad \hat{J}(u) \in z + tr - C,
\end{align*}$$

where the parameter $z$ can be interpreted as a reference point, the parameter $r$ as a target direction (see [Hel90]) and $C$ is a general ordering cone. Graphically, solving the PSP means to move the negative ordering cone $C$ along the ray with support vector $z$ and target direction $r$ starting from $z - \infty \cdot r$ until the intersection of the set $z + tr - C$ with the objective space $\hat{J}(U_{ad})$ is non-empty for the first time, see Figure 1.12. In [PS84], the authors introduced this general formulation, which allows them to apply the method to MOPs with general ordering cones. With the same generality, the PSM was also studied extensively in [Eic08, Eic09], where both the theoretical properties of the method and the numerical implementation via an adaptive parameter choice was discussed. The method and slight modifications of it were also investigated in many other publications, see, e.g., [Hel90, Hel91, SK87].

Here we will be less general and only deal with the case of Pareto optimality, where the ordering cone is given by $C = \mathbb{R}_+^k$. In Section 1.7.1, we will study basic properties of the PSM. In particular, by mainly following results from [Eic08], it is shown that all Pareto optimal points can be computed by the PSM. After a brief discussion of different solution strategies in the literature in Section 1.7.2, the question of which parameters
have to be chosen to guarantee a computation of the entire Pareto front is investigated in Section 1.7.3. For a specific set of parameters, we will show how it can be characterized by solutions of subproblems in Section 1.7.4, following the idea that was presented in [MAL12, MGGS09] and already used for the ERPM in Section 1.6. This is then used to develop a hierarchical algorithm for solving (MOP) by the PSM.

1.7.1 Problem Formulation

Definition 1.7.1. For a given reference point \( z \in \mathbb{R}^k \) and a given target direction \( r \in \mathbb{R}^k_\succ \) the Pascoletti-Serafini problem (PSP) is given by

\[
\begin{align*}
\min_{(u,t)} & \quad t \\
\text{s.t.} & \quad u \in U_{ad}, \ t \in \mathbb{R}, \\
& \quad \hat{J}_i(u) - z_i \leq t r_i, \quad i = 1, \ldots, k.
\end{align*}
\] (PSP\((z,r)\))

It is also possible to define the PSP by introducing a scalarization function \( g_{z,r} \).

Definition 1.7.2. For \( z \in \mathbb{R}^k \) and \( r \in \mathbb{R}^k_\succ \) we define the scalarization function

\[
g_{z,r} : \mathbb{R}^k \to \mathbb{R}, \ x \mapsto \max_{i \in \{1, \ldots, k\}} \frac{1}{r_i} (x_i - z_i),
\]

and the Pascoletti-Serafini scalarized function

\[
\hat{J}^{g_{z,r}} := \max_{i \in \{1, \ldots, k\}} \frac{1}{r_i} (\hat{J}_i - z_i).
\]

Then the reformulated Pascoletti-Serafini problem is given by

\[
\min_{u \in U_{ad}} \hat{J}^{g_{z,r}}(u) = \min_{u \in U_{ad}} \max_{i \in \{1, \ldots, k\}} \frac{1}{r_i} \left( \hat{J}_i(u) - z_i \right). \quad (RPSP(z,r))
\]
It is quite straight-forward to show that the two problems \((\text{PSP}(z, r))\) and \((\text{RPSP}(z, r))\) are equivalent in the following sense:

**Theorem 1.7.3.** Let \(z \in \mathbb{R}^k\) and \(r \in \mathbb{R}_+^k\) be arbitrary. On the one hand, if \((\bar{u}, \bar{t})\) is a global (local) solution of \((\text{PSP}(z, r))\), then \(\bar{u}\) is a global (local) solution of \((\text{RPSP}(z, r))\) with minimal function value \(\bar{t}\). On the other hand, if \(\bar{u}\) is a global (local) solution of \((\text{RPSP}(z, r))\), then \((\bar{u}, \bar{t})\) with \(\bar{t} := \max_{i \in \{1, \ldots, k\}} 1/r_i \left(\hat{J}_i(\bar{u}) - z_i\right)\) is a global (local) solution of \((\text{PSP}(z, r))\).

**Proof.** Let \((\bar{u}, \bar{t})\) be a global (local) solution of \((\text{PSP}(z, r))\). In particular, the inequality \(\hat{J}_i(\bar{u}) - z_i \leq \bar{t} r_i\) is satisfied for all \(i \in \{1, \ldots, k\}\), and there is at least one \(j \in \{1, \ldots, k\}\) such that \(\hat{J}_j(\bar{u}) - z_j = \bar{t} r_j\). Altogether, this implies \(\bar{t} = \max_{i \in \{1, \ldots, k\}} 1/r_i \left(\hat{J}_i(\bar{u}) - z_i\right)\).

Now let \(\bar{u} \in \mathcal{U}_{\text{ad}}\) (in a suitable neighborhood of \(\bar{u}\)) be arbitrary. Due to the global (local) minimality of \((\bar{u}, \bar{t})\), there is no \(\hat{t} \in \mathcal{U}_{\text{ad}}\) with \(\hat{t} < \bar{t}\) and \(\hat{J}_i(\bar{u}) - z_i \leq \hat{t} r_i\) for all \(i \in \{1, \ldots, k\}\). In particular, this implies

\[
\max_{i \in \{1, \ldots, k\}} \frac{1}{r_i} \left(\hat{J}_i(\bar{u}) - z_i\right) \geq \bar{t} = \max_{i \in \{1, \ldots, k\}} \frac{1}{r_i} \left(\hat{J}_i(\bar{u}) - z_i\right).
\]

This shows that \(\bar{u}\) is a global (local) solution of \((\text{RPSP}(z, r))\) with function value \(\hat{t} = \max_{i \in \{1, \ldots, k\}} 1/r_i \left(\hat{J}_i(\bar{u}) - z_i\right)\).

Now let \(\bar{u} \in \mathcal{U}_{\text{ad}}\) be a global (local) solution of \((\text{RPSP}(z, r))\). By defining

\[
\bar{t} := \max_{i \in \{1, \ldots, k\}} \frac{1}{r_i} \left(\hat{J}_i(\bar{u}) - z_i\right),
\]

we clearly have \(\hat{J}(\bar{u}) - z \leq \bar{t} r\), i.e., \((\bar{u}, \bar{t})\) is admissible for \((\text{PSP}(z, r))\). Moreover, for any other \((\bar{u}, \bar{t}) \in \mathcal{U}_{\text{ad}} \times \mathbb{R}\) (where \(\bar{u}\) is supposed to be in a suitable neighborhood of \(\bar{u}\)) fulfilling \(\hat{J}(\bar{u}) - z \leq \hat{t} r\), we have

\[
\bar{t} \geq \max_{i \in \{1, \ldots, k\}} \frac{1}{r_i} \left(\hat{J}_i(\bar{u}) - z_i\right) \geq \max_{i \in \{1, \ldots, k\}} \frac{1}{r_i} \left(\hat{J}_i(\bar{u}) - z_i\right) = \hat{t},
\]

where we used that \(\bar{u}\) is a global (local) solution of \((\text{RPSP}(z, r))\). Thus, \((\bar{u}, \bar{t})\) is a global (local) solution of \((\text{PSP}(z, r))\).

**Remark 1.7.4.** Due to Theorem 1.7.3, the two problems \((\text{PSP}(z, r))\) and \((\text{RPSP}(z, r))\) are equivalent. In the following, we will thus use the formulation that seems to be more suitable for the respective situation. In particular, if we are only interested in or just want to address the minimizing variable \(\bar{u} \in \mathcal{U}_{\text{ad}}\), we use the formulation \((\text{RPSP}(z, r))\). On the other hand, if also the corresponding minimal function value \(\hat{t} = \max_{i \in \{1, \ldots, k\}} 1/r_i \left(\hat{J}_i(\bar{u}) - z_i\right)\) is of interest, we use the formulation \((\text{PSP}(z, r))\).

We can show that every global (local) solution of \((\text{RPSP}(z, r))\) is (locally) weakly Pareto optimal (see [Eic08, Theorem 2.1]).

**Theorem 1.7.5.** Let \(z \in \mathbb{R}^k\) and \(r \in \mathbb{R}_+^k\) be arbitrary. Furthermore, assume that there is a global (local) solution \(\bar{u} \in \mathcal{U}_{\text{ad}}\) of \((\text{RPSP}(z, r))\). Then \(\bar{u} \in \mathcal{U}_{\text{opt}}(w, (\text{loc}))\).
Proof. This follows directly from Lemma 1.4.3, since the function $g_{z,r}$ is strictly monotonically increasing on $\hat{J}(U_{ad})$. 

Note that we did not make any statement about the solvability of the optimization problem (RPSP$(z,r)$). In fact, in general, it cannot be guaranteed that there is a global solution of (RPSP$(z,r)$) for every reference point $z \in \mathbb{R}^k$ and every target direction $r \in \mathbb{R}^k_\geq$. To this end, we introduce the following assumptions on the cost functions.

**Assumption 1.2.** The cost functions $\hat{J}_1, \ldots, \hat{J}_k$ are weakly lower semi-continuous and bounded from below. Additionally, in case that $U_{ad}$ is unbounded, we have

$$\lim_{\|u\|_{U_{ad}} \to \infty} \hat{J}_i(u) = \infty \quad \text{for all } i \in \{1, \ldots, k\}.$$

Supposing that Assumption 1.2 holds, we can show that the problem (RPSP$(z,r)$) has a global solution. For this purpose, we introduce the extended Pascoletti-Serafini problem (see also the EERPP in Definition 1.6.4).

**Definition 1.7.6.** For any $z \in \mathbb{R}^k$ and $r \in \mathbb{R}^k_\geq$ we define the extended Pascoletti-Serafini problem

$$\min_{y \in \hat{J}(U_{ad}) + \mathbb{R}^k_\geq} g_{z,r}(y) = \min_{y \in \hat{J}(U_{ad}) + \mathbb{R}^k_\geq} \max_{i \in \{1, \ldots, k\}} \frac{1}{r_i} (y_i - z_i).$$

(EPSP$(z,r)$)

First, we show that (RPSP$(z,r)$) is equivalent to (EPSP$(z,r)$).

**Lemma 1.7.7.** Let $z \in \mathbb{R}^k$ and $r \in \mathbb{R}^k_\geq$ be arbitrary. If $\bar{u} \in U_{ad}$ is a global solution of (RPSP$(z,r)$), then $\bar{y} := \hat{J}(\bar{u})$ is a global solution of (EPSP$(z,r)$). On the other hand, if $\bar{y} = \hat{J}(\bar{u}) + d \in \hat{J}(U_{ad}) + \mathbb{R}^k_\geq$ is a global solution of (EPSP$(z,r)$), then $\bar{u}$ is a global solution of (RPSP$(z,r)$).

**Proof.** This is an easy conclusion from the monotonicity properties of the function $g_{z,r}$. 

We can show that (EPSP$(z,r)$) has a global solution by using the closedness of $\hat{J}(U_{ad}) + \mathbb{R}^k_\geq$, see Lemma 1.5.8 for the case of convex cost functions.

**Proposition 1.7.8.** Let Assumption 1.2 be satisfied. Then the set $\hat{J}(U_{ad}) + \mathbb{R}^k_\geq$ is closed.

**Proof.** This can be shown by using the exact same arguments as in the proof of Lemma 1.5.8, cf. [Ban17, Theorem 3.35]. In fact, there the convexity of the cost functions is only used to conclude that they are weakly lower semi-continuous, which we directly assume in Assumption 1.2.

The existence of a global solution of (EPSP$(z,r)$) can now be easily concluded.
Theorem 1.7.9. Let Assumption 1.2 be satisfied and \( z \in \mathbb{R}^k \) as well as \( r \in \mathbb{R}^k_\geq \) be arbitrary. Then \((\text{EPSP}(z,r))\) has a global solution \( \bar{y} \in \tilde{J}(U_{\text{ad}}) + \mathbb{R}^k_\geq \).

Proof. Let \( \tilde{y} \in \tilde{J}(U_{\text{ad}}) + \mathbb{R}^k_\geq \) be arbitrary and define \( \tilde{t} := \max_{i \in \{1, \ldots, k\}} 1/r_i (\tilde{y} - z_i) \). Then solving \((\text{EPSP}(z,r))\) is equivalent to solving

\[
\min_{y \in Y} g_{z,r}(y) = \min_{y \in Y} \max_{i \in \{1, \ldots, k\}} \frac{1}{r_i} (y - z_i) \tag{1.7.1}
\]

with \( Y := (\tilde{J}(U_{\text{ad}}) + \mathbb{R}^k_\geq) \cap (\tilde{t} - \mathbb{R}^k_\geq) \). By Proposition 1.7.8, the set \( \tilde{J}(U_{\text{ad}}) + \mathbb{R}^k_\geq \) is closed. Since the cost functions are bounded from below by Assumption 1.2, it is also bounded from below. Moreover, the set \( \tilde{t} - \mathbb{R}^k_\geq \) is closed and bounded from above. Thus, we can conclude that the set \( Y \) is compact. The function \( g_{z,r} \) is continuous and consequently attains its minimum on the compact set \( Y \).

As a corollary of the previous results, we get the existence of a global solution of \((\text{RPSP}(z,r))\).

Corollary 1.7.10. Let Assumption 1.2 be satisfied and \( z \in \mathbb{R}^k \) as well as \( r \in \mathbb{R}^k_\geq \) be arbitrary. Then \((\text{RPSP}(z,r))\) has a global solution \( \bar{u} \in U_{\text{ad}} \).

Proof. The problem \((\text{EPSP}(z,r))\) has a global solution \( \tilde{u} \in \tilde{J}(U_{\text{ad}}) + \mathbb{R}^k_\geq \) by Theorem 1.7.9. By Lemma 1.7.7, we get that \( \bar{u} \) is a global solution of \((\text{RPSP}(z,r))\).

Corollary 1.7.11. Let Assumption 1.2 be satisfied. Then the Pareto front \( J_{\text{opt}} \) is externally stable.

Proof. To apply Theorem 1.2.14, we need to show that the set \( (y - \mathbb{R}^k_\geq) \cap (\tilde{J}(U_{\text{ad}}) + \mathbb{R}^k_\geq) \) is compact for all \( y \in \tilde{J}(U_{\text{ad}}) + \mathbb{R}^k_\geq \). This follows with the same arguments as the compactness of the set \( (\tilde{J}(U_{\text{ad}}) + \mathbb{R}^k_\geq) \cap (\tilde{t} - \mathbb{R}^k_\geq) \) in the proof of Theorem 1.7.9.

Now we can even conclude that \((\text{RPSP}(z,r))\) has a Pareto optimal solution.

Corollary 1.7.12. Let Assumption 1.2 be satisfied and \( z \in \mathbb{R}^k \) as well as \( r \in \mathbb{R}^k_\geq \) be arbitrary. Then \((\text{RPSP}(z,r))\) has a global solution \( \bar{u} \in U_{\text{opt}} \).

Proof. By Corollary 1.7.10, there is a global solution \( \tilde{u} \in U_{\text{ad}} \) of \((\text{RPSP}(z,r))\). Since the Pareto front is externally stable by Corollary 1.7.11, there is \( \bar{u} \in U_{\text{opt}} \) such that \( \tilde{J}(\tilde{u}) \leq \tilde{J}(\bar{u}) \). The monotonicity of the function \( g_{z,r} \) implies that \( \bar{u} \) is also a global solution of \((\text{RPSP}(z,r))\).

Conversely, it is also possible to show that for every (weakly) Pareto optimal point the parameters \( z \) and \( r \) can be chosen in such a way that this point is a solution of \((\text{RPSP}(z,r))\).

A similar statement was shown in [Eic08, Theorem 2.1].
Theorem 1.7.13. Let \( \bar{u} \in U_{\text{opt},w,(\text{loc})} \) be arbitrary. Then for every \( r \in \mathbb{R}_k^+ \) and every \( \bar{t} \in \mathbb{R} \) we have that \( \bar{u} \) is a global (local) solution of \( (\text{RPSP}(z,r)) \) for the reference point \( z := \bar{J}(\bar{u}) - \bar{t}r \). If even \( \bar{u} \in U_{\text{opt}} \), any other global solution \( \tilde{u} \) of \( (\text{RPSP}(z,r)) \) satisfies \( \bar{J}(\bar{u}) = \tilde{J}(\tilde{u}) \).

Proof. We show the first statement by contradiction. Let \( \bar{u} \in U_{\text{opt},w,(\text{loc})}, \ r \in \mathbb{R}_k^+ \) and \( \bar{t} \in \mathbb{R} \) be arbitrary and define \( z := \bar{J}(\bar{u}) - \bar{t}r \). Assume that there is \( \bar{u} \in U_{\text{ad}} \) (in every neighborhood of \( \bar{u} \) with \( \bar{J}^z_r(\bar{u}) < \tilde{J}^z_r(\bar{u}) \)). By plugging in the formula for the reference point \( z \), this implies

\[
\bar{t} + \max_{i \in \{1, \ldots, k\}} \frac{1}{r_i} \left( \hat{J}_i(\bar{u}) - \hat{J}_i(\bar{u}) \right) = \max_{i \in \{1, \ldots, k\}} \frac{1}{r_i} \left( \hat{J}_i(\bar{u}) - z_i \right) < \max_{i \in \{1, \ldots, k\}} \frac{1}{r_i} \left( \hat{J}_i(\bar{u}) - z_i \right) = \bar{t},
\]

so that \( \max_{i \in \{1, \ldots, k\}} 1/r_i (\hat{J}_i(\bar{u}) - \hat{J}_i(\bar{u})) < 0 \). Thus, \( \bar{J}(\bar{u}) < \bar{J}(\bar{u}) \), which is a contradiction to \( \bar{u} \in U_{\text{opt},w,(\text{loc})} \). Therefore, \( \bar{u} \) is a global (local) solution of \( (\text{RPSP}(z,r)) \).

If \( \bar{u} \in U_{\text{opt}} \) and \( \bar{u} \in U_{\text{ad}} \) is another global solution of \( (\text{RPSP}(z,r)) \), with the same arguments as above, we can conclude

\[
\bar{t} + \max_{i \in \{1, \ldots, k\}} \frac{1}{r_i} \left( \hat{J}_i(\bar{u}) - \hat{J}_i(\bar{u}) \right) = \max_{i \in \{1, \ldots, k\}} \frac{1}{r_i} \left( \hat{J}_i(\bar{u}) - z_i \right) = \bar{t}.
\]

Therefore, we must have \( \bar{J}(\bar{u}) \leq \bar{J}(\bar{u}) \). Since \( \bar{u} \) is Pareto optimal, we cannot have \( \bar{J}(\bar{u}) \not\leq \bar{J}(\bar{u}) \), so that we obtain \( \bar{J}(\bar{u}) = \bar{J}(\bar{u}) \).

\[\square\]

Remark 1.7.14. Note that Lemma 1.4.4, in which we showed that for every (locally) (weakly) Pareto optimal point there is a scalarization function such that this point is a solution to the scalarized problem (SOP), is a special case of Theorem 1.7.13.

\[\diamondsuit\]

Under additional assumptions, we can show the following necessary first-order optimality condition for a global solution of \( (\text{PSP}(z,r)) \).

Lemma 1.7.15. Let Assumption 1.2 be satisfied. Assume additionally that \( U_{\text{ad}} \) contains an interior point and that all cost functions are Fréchet differentiable. Let \( z \in \mathbb{R}_k^+ \) as well as \( r \in \mathbb{R}_k^+ \) be arbitrary. If \( (\bar{u}, \bar{t}) \in U_{\text{ad}} \times \mathbb{R} \) is a global solution of \( (\text{PSP}(z,r)) \), then there is a Lagrange multiplier \( \lambda \in \mathbb{R}_k^+ \) such that

\[
\sum_{i=1}^{k} \lambda_i r_i = 1, \quad (1.7.2a)
\]

\[
\langle \sum_{i=1}^{k} \lambda_i \nabla \hat{J}_i(\bar{u}), u - \bar{u} \rangle \geq 0 \quad \text{for all } u \in U_{\text{ad}}, \quad (1.7.2b)
\]

\[
\hat{J}_i(\bar{u}) - z_i \leq \bar{t}r_i \quad \text{for all } i \in \{1, \ldots, k\}, \quad (1.7.2c)
\]

\[
\lambda_i \left( \hat{J}_i(\bar{u}) - z_i - \bar{t}r_i \right) = 0 \quad \text{for all } i \in \{1, \ldots, k\} \quad (1.7.2d)
\]

hold.
Proof. We can write the problem \((\text{PSP}(z,r))\) in the form
\[
\min_{(u,t) \in S} \quad f(u,t)
\]
s.t. \(g_i(u,t) \leq 0, \quad i = 1, \ldots, k.
\]
with \(S := \mathbb{U}_\text{ad} \times \mathbb{R}\),
\[
f : \mathbb{U} \times \mathbb{R} \to \mathbb{R}, \quad f(u,t) := t,
\]
and
\[
g : \mathbb{U} \times \mathbb{R} \to \mathbb{R}^k, \quad g_i(u,t) := \hat{J}_i(u) - z_i - t r_i \quad \text{for all } i \in \{1, \ldots, k\}.
\]
If we can show that there is \((\hat{u}, \hat{t}) \in \text{int}(S) = \text{int}(\mathbb{U}_\text{ad}) \times \mathbb{R}\) with
\[
g(\bar{u}, \bar{t}) + g'(\bar{u}, \bar{t})((\hat{u}, \hat{t}) - (\bar{u}, \bar{t})) < 0,
\]
the claim follows directly from [Jah11, Theorem 7.4]. By assumption, we can choose \(\hat{u} \in \text{int}(\mathbb{U}_\text{ad})\). Because of \(g(\bar{u}, \bar{t}) \leq 0\), we can write
\[
g(\bar{u}, \bar{t}) + g'(\bar{u}, \bar{t})((\hat{u}, \hat{t}) - (\bar{u}, \bar{t})) \leq g'(\bar{u}, \bar{t})((\hat{u}, \hat{t}) - (\bar{u}, \bar{t}))
\]
\[
= \begin{pmatrix} \langle \nabla \hat{J}_1(\bar{u}), \hat{u} - \bar{u} \rangle_U - (\hat{t} - \bar{t}) r_1 \\ \vdots \\ \langle \nabla \hat{J}_k(\bar{u}), \hat{u} - \bar{u} \rangle_U - (\hat{t} - \bar{t}) r_k \end{pmatrix} < 0,
\]
for
\[
\hat{t} > \max_{i \in \{1, \ldots, k\}} \frac{\langle \nabla \hat{J}_i(\bar{u}), \hat{u} - \bar{u} \rangle_U + \bar{t} r_i}{r_i},
\]
which is what we had to show. \(\square\)

1.7.2 Literature Review on Solution Strategies

In the previous section, we have seen that the PSM is in principle capable of computing every (locally) (weakly) Pareto optimal point without any assumptions on the cost functions, cf. Theorem 1.7.13. Based on this result, there are many examples in the literature on how to design algorithms using the PSM to solve (MOP). Here we do not want to discuss all of these approaches, but only describe a few of those that are able to deal with MOPs with arbitrarily many cost functions.

As we have seen, the PSM has two parameters \(z\) and \(r\), which might be varied to obtain different solutions. There are mainly two different approaches in the literature: Keeping \(r\) fixed and varying \(z\) or vice versa.

In this thesis, we will follow the first approach, i.e., we will fix a target direction \(r\) and vary \(z\) to obtain all Pareto optimal points. This was also done in [Eic08, Section 2.3], where it was shown that the reference points \(z\) can be restricted to a subset \(H^\circ\) (or \(\hat{H}\) in the case of \(k > 2\)) of a \((k-1)\)-dimensional hyperplane \(H\). However, the method turns out to be, on the one hand, not very efficient in the sense that \(\hat{H}\) might contain many redundant reference
points if \( k > 2 \). On the other hand, the method might not be applicable numerically for problems with \( k > 2 \), for which the set

\[
\tilde{H} := \{ y \in H \mid \exists t \in \mathbb{R} : \exists u \in \mathcal{U}_{\text{ad}} : y + tr = \hat{J}(u) \}
\]

is not bounded. Note that this might happen even if the Pareto front is bounded.

In [MGGS09], the idea of using the solutions of subproblems to obtain information that can be used to solve (MOP) is described for the first time. The authors show that under some assumptions on the Pareto front, the so-called trade-off limits, i.e., points on the Pareto front, which cannot be improved in at least one component, are given by the solutions to subproblems. Based on this, they develop the idea of first computing these trade-off limits (the 'boundary' of the Pareto front) by solving subproblems and then filling the 'interior' of the Pareto front. To this end, the reference points are computed by solving linear programs involving the previously computed solutions of subproblems. A very similar approach was independently described in [MAL12] with the difference being that they use a Centroidal Voronoi Tessellation (CVT) algorithm for the construction of the reference points. Note that the authors in [MAL12] see their method as a modification of the NBI method. However, this modified method can also be seen as a slight reformulation of the Pascoletti-Serafini method. The approach presented in [MGGS09] is also discussed and enhanced in [KSd15], where some weaknesses and inaccuracies of the method presented in [MGGS09] are shown. The authors propose a more precise definition of the 'boundary' of the Pareto front, which allows the method to perform in more general cases.

Here we will follow the idea presented in [MAL12, MGGS09] of solving (MOP) sequentially by hierarchically solving subproblems of (MOP), see also Section 1.6.4, where this procedure was developed for the ERPM. However, we take a different perspective than the previously mentioned approaches. Our incentive is to find and characterize a set of reference points, which guarantees that all (weakly) Pareto optimal points can be obtained. It turns out that this set – analogously to the results found in [KSd15, MAL12, MGGS09] – can be described by looking at solutions of subproblems. In contrast to this, the approach in [KSd15, MAL12, MGGS09] directly wants to find the 'boundary' of the Pareto front. In fact, in neither of these publications it is formally proven that their method of filling the interior of the Pareto front works. In this sense, our contribution can be seen as an addition and partly also a generalization to the results in [KSd15, MAL12, MGGS09]. Moreover, we will show that our approach allows a computationally fast and efficient way of choosing reference points, which neither needs to solve linear programs, nor to use a CVT algorithm.

For the sake of completeness, let us also note that the latter approach, i.e., keeping \( z \) fixed and varying \( r \), was followed, e.g., in [Kha16, KKK14, DK19]. In [KKK14], the parameter \( z \) is chosen to be the ideal point, i.e., \( z = y^{\text{id}} \), and then the parameter \( r \) is taken from the parameter set \( R := \{ \beta \in \mathbb{R}_+^k \mid ||\beta||_2 = 1 \} \). It is shown that all Pareto optimal points can be obtained by solving (PSP(\( z, r \))) for the reference point \( z \) and all \( r \in R \). Moreover, the set \( R \) does not depend on the MOP and can, in particular, be easily computed for an arbitrary number of cost functions. Therefore, this approach for choosing the parameter values is comparable to the WSM, where the weights are chosen from the set \( \Delta_k \), which is problem-independent as well. However, this method also fails to generate well-distributed
approximations of the Pareto front in some (convex) cases, even if the parameters \( r \in \mathbb{R} \) are evenly distributed (see, e.g., [KSd15, Section 4.3]). The approach from [KKK14] is extended to more general ordering cones in [Kha16]. In [DK19], this method is enhanced by taking into account information provided by solving subproblems of the MOP, building on the idea proposed in [MGGS09]. This leads to a restriction of the parameter set \( R \) and, therefore, to a more efficient algorithm.

### 1.7.3 A Pareto Sufficient Set of Reference Points

In Section 1.7.1, we showed that every global (local) solution of \((\text{RPSP}(z,r))\) is at least (locally) weakly Pareto optimal for any choice of \( z \in \mathbb{R}^k \) and \( r \in \mathbb{R}^k_+ \), and that every (locally) (weakly) Pareto optimal point can be obtained by solving \((\text{RPSP}(z,r))\) for a certain reference point. From now on, let \( r \in \mathbb{R}^k_+ \) be arbitrary, but fixed. In this section, we will investigate which reference points are necessary and sufficient to compute the entire (local) (weak) Pareto front. To summarize the results for the PSM, we introduce the following solution mappings.

**Definition 1.7.16.** We define the solution mappings

\[
Q_{\text{opt},w}: \mathbb{R}^k \to U_{\text{opt},w}, \quad z \mapsto \{ u \in U_{\text{ad}} \mid u \text{ global solution of } (\text{RPSP}(z,r)) \},
\]

\[
Q_{\text{opt},w,\text{loc}}: \mathbb{R}^k \to U_{\text{opt},w,\text{loc}}, \quad z \mapsto \{ u \in U_{\text{ad}} \mid u \text{ local solution of } (\text{RPSP}(z,r)) \},
\]

\[
Q_{\text{opt},(\text{loc})}: \mathbb{R}^k \to U_{\text{opt},(\text{loc})}, \quad z \mapsto Q_{\text{opt},w,\text{loc}}(z) \cap U_{\text{opt},(\text{loc})},
\]

\[
R_{\text{opt},(w),(\text{loc})}: \mathbb{R}^k \to J_{\text{opt},w,(\text{loc})}, \quad z \mapsto \hat{J}(Q_{\text{opt},w,(\text{loc})}(z)).
\]

For these mappings, we can show the following statements.

**Theorem 1.7.17.** The mappings \( Q_{\text{opt},(w),(\text{loc})} \) and \( R_{\text{opt},(w),(\text{loc})} \) are well-defined and we have

\[
Q_{\text{opt},(w),(\text{loc})}(\mathbb{R}^k) = U_{\text{opt},(w),(\text{loc})}, \quad R_{\text{opt},(w),(\text{loc})}(\mathbb{R}^k) = J_{\text{opt},w,(\text{loc})}.
\]  

(1.7.3)

Moreover, if Assumption 1.2 is satisfied, we have

\[
Q_{\text{opt},(w),(\text{loc})}(z) \neq \emptyset, \quad R_{\text{opt},(w),(\text{loc})}(z) \neq \emptyset
\]  

(1.7.4)

for all \( z \in \mathbb{R}^k \).

**Proof.** The well-definedness follows directly from Theorem 1.7.5. The identities (1.7.3) can be concluded from Theorem 1.7.13 and the statements (1.7.4) follow from Corollary 1.7.10 and Corollary 1.7.12.

In contrast to the ERPM, any choice of the reference point in \((\text{RPSP}(z,r))\) leads to at least a (locally) weakly Pareto optimal point. Therefore, we do not need to introduce the notion of a (locally) (weakly) Pareto admissible reference point for the PSM, cf. Definition 1.6.19. However, as for the ERPM, the question is, which set of reference points guarantees the computation of the entire (local) (weak) Pareto front. To this end, we introduce the notion of a (locally) (weakly) Pareto sufficient set for the PSM, cf. Definition 1.6.22.
**Definition 1.7.18.** A set \( Z \subset \mathbb{R}^k \) is called (locally) (weakly) Pareto sufficient if it holds
\[
\mathcal{Q}_{\text{opt},(w),(\text{loc})}(Z) = \mathcal{U}_{\text{opt},(w),(\text{loc})}.
\]

By Theorem 1.7.17, we immediately get the following result.

**Corollary 1.7.19.** The set \( \mathbb{R}^k \) is (locally) (weakly) Pareto sufficient.

The next naturally arising question is whether there is a (locally) (weakly) Pareto sufficient set \( Z \subset \mathbb{R}^k \), and if so, how it can be characterized. A first partial answer is given by the next lemma. By using Theorem 1.7.13, we can show a sufficient conditions for a set \( Z \subset \mathbb{R}^k \) to be (locally) (weakly) Pareto sufficient.

**Lemma 1.7.20.** Let \( Z \subset \mathbb{R}^k \) be arbitrary. If
\[
\forall \bar{u} \in \mathcal{U}_{\text{opt},(w),(\text{loc})}: \exists t \in \mathbb{R}: \hat{J}(\bar{u}) - tr \in Z,
\]
then \( Z \) is (locally) (weakly) Pareto sufficient.

**Proof.** Let \( Z \subset \mathbb{R}^k \) be such that (1.7.5) is satisfied. Let \( \bar{u} \in \mathcal{U}_{\text{opt},(w),(\text{loc})} \) be arbitrary. We need to show that there is \( z \in Z \) with \( \bar{u} \in \mathcal{Q}_{\text{opt},(w),(\text{loc})}(z) \). Indeed, by (1.7.5) there is \( t \in \mathbb{R} \) with \( z := \hat{J}(\bar{u}) - tr \in Z \) and by Theorem 1.7.13 it holds \( \bar{u} \in \mathcal{Q}_{\text{opt},(w),(\text{loc})}(z) \), which proves the claim.

**Remark 1.7.21.** In other words, a set \( Z \subset \mathbb{R}^k \) is (locally) (weakly) Pareto sufficient if for all \( \bar{u} \in \mathcal{U}_{\text{opt},(w),(\text{loc})} \) the ray \( t \mapsto \hat{J}(\bar{u}) - tr \) has an intersection point with \( Z \). This is comparable to the characterization of the set of all (weakly) Pareto admissible reference points \( \mathcal{Z}_{\text{opt},(w)} \) for the ERPM in Theorem 1.6.21. The difference is that for the ERPM the direction of the ray depends on the weight vector \( \alpha \) of the corresponding WSP, while for the PSM the direction is always given by the vector \( r \). Due to this similarity, we can adapt the procedure that we presented for the ERPM to the PSM.

**Definition 1.7.22.** Recall the definition of the set \( D \) in Definition 1.6.25. Then we define the sets
\[
\mathcal{Z}_{\text{opt},(w),(\text{loc})} := \{ z \in D | \exists \bar{u} \in \mathcal{U}_{\text{opt},(w),(\text{loc})}: \exists t \in \mathbb{R}: z = \hat{J}(\bar{u}) - tr \}.
\]

The following functions compute the intersection point of the ray \( t \mapsto \hat{J}(\bar{u}) - tr \) with the set \( D \).

**Definition 1.7.23.** For any \( y \in \mathbb{R}^k \) we set
\[
t^D(y) := \min_{i \in \{1, \ldots, k\}} \frac{y_i - \tilde{y}^i}{r_i} \in \mathbb{R}.
\]

Then we define the functions
\[
\mathcal{N}^D : \mathcal{J}_{\text{opt},w,\text{loc}} \to D, \quad \mathcal{N}^D(y) := y - t^D(y) r, \\
\mathcal{L}^D : \mathcal{U}_{\text{opt},w,\text{loc}} \to D, \quad \mathcal{L}^D(u) := \mathcal{N}^D(\hat{J}(u)).
\]
1.7 The Pascoletti-Serafini Method

The Pascoletti-Serafini Method

(a) The mapping $\mathcal{N}^D$

(b) The set $\mathcal{Z}^D_{\text{opt}(w)}$

Figure 1.13: Left: Illustrations of the mapping $\mathcal{N}^D$. For any $y \in \mathcal{J}_{\text{opt}(w)}$, the point $\mathcal{N}^D(y)$ is given by the intersection point of the ray $t \mapsto y - tr$ with the set $D$. Right: Illustration of the set $\mathcal{Z}^D_{\text{opt}(w)} = \mathcal{N}^D(\mathcal{J}_{\text{opt}(w)})$, cf. (1.7.6).

Lemma 1.7.24. (i) The functions $\mathcal{N}^D$ and $\mathcal{L}^D$ are well-defined.

(ii) We have

$$\mathcal{Z}^D_{\text{opt}(w),(\text{loc})} = \mathcal{N}^D(\mathcal{J}_{\text{opt}(w),(\text{loc})}) = \mathcal{L}^D(\mathcal{U}_{\text{opt}(w),(\text{loc})})$$

$$= \left\{ \bar{J}(\bar{u}) - t^D(\bar{J}(\bar{u})) r \mid \bar{u} \in \mathcal{U}_{\text{opt}(w),(\text{loc})} \right\}. \quad (1.7.6)$$

Proof. (i) Let $\bar{y} = \bar{J}(\bar{u}) \in \mathcal{J}_{\text{opt},w,(\text{loc})}$ be arbitrary and select $j \in \{1, \ldots, k\}$ with

$$j \in \arg \min_{i \in \{1, \ldots, k\}} \frac{\bar{y}_i - \bar{y}^i_{id}}{r_i}.$$  

Then it is straightforward to show that $\mathcal{N}^D(\bar{y}) \in D_j \subset D$ holds, so that $\mathcal{N}^D$ is well-defined. Moreover, the well-definedness of $\mathcal{L}^D$ follows directly from the well-definedness of $\mathcal{N}^D$.

(ii) We only prove the identity $\mathcal{Z}^D_{\text{opt}(w),(\text{loc})} = \mathcal{L}^D(\mathcal{U}_{\text{opt}(w),(\text{loc})})$, since the rest follows directly by definition.

'\subset': Let $z \in \mathcal{Z}^D_{\text{opt}(w),(\text{loc})}$ be arbitrary. Then by definition, there is $\bar{u} \in \mathcal{U}_{\text{opt}(w),(\text{loc})}$ and $t \in \mathbb{R}$ such that $z = \bar{J}(\bar{u}) - tr \in D$ holds. If we can show that $t = t^D(\bar{J}(\bar{u}))$, we are done. Since $z \in D$, there is $i \in \{1, \ldots, k\}$ with $z \in D_i$. By definition, this implies $z \geq \bar{y}^i_{id}$ and $z_i = \bar{y}^i_{id}$. From this, we get $t \leq (\bar{J}(\bar{u}) - \bar{y}^i_{id})/r_j$ for all $j \in \{1, \ldots, k\}$, and $z_i = \bar{y}^i_{id}$ can be used to conclude $t = (\bar{J}(\bar{u}) - \bar{y}^i_{id})/r_i$. Altogether, this implies $t = t^D(\bar{J}(\bar{u}))$, which finishes the proof of this inclusion.

'\supset': This follows from the well-definedness of the function $\mathcal{L}^D: \mathcal{J}_{\text{opt},w,(\text{loc})} \to D$. 

\qed
Corollary 1.7.25. The set $\mathcal{Z}^D_{\text{opt,(w),(loc)}}$ is (locally) (weakly) Pareto sufficient.

Proof. This follows directly from the definition of the set $\mathcal{Z}^D_{\text{opt,(w),(loc)}}$, Lemma 1.7.20 and (1.7.6).

Moreover, we can show the following connections between $\mathcal{R}_{\text{opt,(w),(loc)}}$ and $\mathcal{N}^D$ as well as $\mathcal{Q}_{\text{opt,(w),(loc)}}$ and $\mathcal{L}^D$.

Theorem 1.7.26. It holds

$$z \in \mathcal{N}^D(\mathcal{R}_{\text{opt,(w),(loc)}}(z)) \quad \text{for all } z \in \mathcal{Z}^D_{\text{opt,(w),(loc)}},$$

(1.7.7)

$$\tilde{y} \in \mathcal{R}_{\text{opt,(w),(loc)}}(\mathcal{N}^D(\tilde{y})) \quad \text{for all } \tilde{y} \in \mathcal{J}_{\text{opt,(w),(loc)}},$$

(1.7.8)

Furthermore, we have

$$\mathcal{N}^D(\mathcal{R}_{\text{opt}}(z)) = \{z\} \quad \text{for all } z \in \mathcal{Z}^D_{\text{opt}},$$

(1.7.9)

$$\mathcal{R}_{\text{opt}}(\mathcal{N}^D(\tilde{y})) = \{\tilde{y}\} \quad \text{for all } \tilde{y} \in \mathcal{J}_{\text{opt}}.$$  

(1.7.10)

In particular, $\mathcal{R}_{\text{opt}|_{\mathcal{Z}^D_{\text{opt}}}}$ is single-valued, and $\mathcal{N}^D|_{\mathcal{J}_{\text{opt}}}$ and $\mathcal{R}_{\text{opt}}|_{\mathcal{Z}^D_{\text{opt}}}$ are bijective with $(\mathcal{N}^D|_{\mathcal{J}_{\text{opt}}})^{-1} = \mathcal{R}_{\text{opt}|_{\mathcal{Z}^D_{\text{opt}}}}$.

Proof. We start by showing (1.7.7). Let $z \in \mathcal{Z}^D_{\text{opt,(w),(loc)}}$ be arbitrary. From (1.7.6), we infer that there is $\tilde{u} \in \mathcal{U}_{\text{opt,(w),(loc)}}$ with $z = \tilde{J}(\tilde{u}) - t^D(\tilde{J}(\tilde{u})) r$. By Theorem 1.7.13, $\tilde{u}$ is a global (local) solution of $(\text{RPSP}(z,r))$, so that $\tilde{J}(\tilde{u}) \in \mathcal{R}_{\text{opt,(w),(loc)}}(z)$. Now by definition, it follows $z = \mathcal{N}^D(\tilde{J}(\tilde{u})) \in \mathcal{N}^D(\mathcal{R}_{\text{opt,(w),(loc)}}(z))$.

For the proof of (1.7.8), let $\tilde{y} = \tilde{J}(\tilde{u}) \in \mathcal{J}_{\text{opt,(w),(loc)}}$ be arbitrary. For $z := \mathcal{N}^D(\tilde{y})$, it holds $z = \tilde{y} - t^D(\tilde{J}(\tilde{u})) r$ and by Theorem 1.7.13, $\tilde{u}$ is a global (local) solution of $(\text{RPSP}(z,r))$ to the reference point $z$. This implies $\tilde{y} \in \mathcal{R}_{\text{opt,(w),(loc)}}(\mathcal{N}^D(\tilde{y}))$ and concludes the proof.

To show (1.7.9), let $z \in \mathcal{Z}^D_{\text{opt}}$ be arbitrary. Note that ‘⊂’ directly follows from (1.7.7).

‘⊃’: There is $\tilde{u} \in \mathcal{U}_{\text{opt}}$ with $z = \tilde{J}(\tilde{u}) - t^D(\tilde{J}(\tilde{u})) r$. By Theorem 1.7.13, $\tilde{u}$ is a global solution to $(\text{RPSP}(z,r))$, and for any other global solution $\tilde{u}$ of $(\text{RPSP}(z,r))$ we have $\tilde{J}(\tilde{u}) = \tilde{J}(\tilde{u})$. Thus, $\mathcal{R}_{\text{opt}}(z) = \{\tilde{J}(\tilde{u})\}$ and by definition we have $z = \tilde{J}(\tilde{u}) - t^D(\tilde{J}(\tilde{u})) r = \mathcal{N}^D(\tilde{J}(\tilde{u}))$, which concludes the proof of (1.7.9).

In order to show (1.7.10), let $\tilde{y} = \tilde{J}(\tilde{u}) \in \mathcal{J}_{\text{opt}}$ be arbitrary. Note that ‘⊃’ follows from (1.7.8).

‘⊂’: It holds $z := \mathcal{N}^D(\tilde{y}) = \tilde{y} - t^D(\tilde{J}(\tilde{u})) r$. By Theorem 1.7.13, $\tilde{u}$ is a global solution to $(\text{RPSP}(z,r))$, and for any other global solution $\tilde{u}$ of $(\text{RPSP}(z,r))$ it holds $\tilde{J}(\tilde{u}) = \tilde{y}$. Thus, $\mathcal{R}_{\text{opt}}(z) = \{\tilde{y}\}$, which concludes the proof. □
Theorem 1.7.27. It holds
\[ z \in \mathcal{L}^D(\mathcal{Q}_{\text{opt}}(w),(\text{loc}))(z) \quad \text{for all } z \in \mathcal{Z}_{\text{opt}}^D(\mathcal{Q}_{\text{opt}}(w),(\text{loc})), \]
\[ \bar{u} \in \mathcal{Q}_{\text{opt}}(w),(\text{loc})(\mathcal{L}^D(\bar{u})) \quad \text{for all } \bar{u} \in \mathcal{U}_{\text{opt}}(w),(\text{loc}). \]
Moreover, we have
\[ \{z\} = \mathcal{L}^D(\mathcal{Q}_{\text{opt}}(z)) \quad \text{for all } z \in \mathcal{Z}_{\text{opt}}^D. \]

Proof. All statements follow either directly from Theorem 1.7.26 or can be shown with the same arguments as in the proof thereof. Note that we can only show \( \bar{u} \in \mathcal{Q}_{\text{opt}}(\mathcal{L}^D(\bar{u})) \) for all \( \bar{u} \in \mathcal{U}_{\text{opt}} \), but not the other inclusion, although it holds \( \mathcal{R}_{\text{opt}}(\mathcal{N}^D(\bar{y})) = \bar{y} \) for all \( \bar{y} \in \mathcal{J}_{\text{opt}} \). The reason for this is that the preimage \( \mathcal{J}^{-1}(\{\bar{y}\}) \) might contain more than one element. \(\square\)

In Theorem 1.7.26, we proved that \( \mathcal{R}_{\text{opt}}|_{\mathcal{Z}_{\text{opt}}^D} \) is single-valued and can thus be seen as a function. In the following, we show that this function is even Lipschitz continuous. A similar result was shown in [Wie86, Theorem 11].

Theorem 1.7.28. The function \( \mathcal{R}_{\text{opt}}|_{\mathcal{Z}_{\text{opt}}^D} : (\mathcal{Z}_{\text{opt}}^D; \|\cdot\|_\infty) \rightarrow (\mathcal{J}_{\text{opt}}; \|\cdot\|_\infty) \) is Lipschitz continuous with Lipschitz constant
\[ L_{\mathcal{R},\infty} := 1 + \max_{i,j \in \{1,\ldots,k\}} \frac{r_i}{r_j}. \]

Proof. Let \( z^1, z^2 \in \mathcal{Z}_{\text{opt}}^D \) be arbitrary. According to (1.7.6) and Theorem 1.7.13, there are global solutions \( \bar{u}^1, \bar{u}^2 \in \mathcal{U}_{\text{opt}} \) of (RPSP(\( z^1, r \))) and (RPSP(\( z^2, r \))), respectively, which fulfill \( \bar{J}(\bar{u}^j) - z^j = \bar{v}^j r \), where \( \bar{v}^j := t^D(\bar{J}(\bar{u}^j)) \) for \( j = 1, 2 \). Now we can write
\[ \|\mathcal{R}_{\text{opt}}(z^1) - \mathcal{R}_{\text{opt}}(z^2)\|_\infty = \max_{i \in \{1,\ldots,k\}} \left| \bar{J}_i(\bar{u}^1) - \bar{J}_i(\bar{u}^2) \right| \]
\[ = \max_{i \in \{1,\ldots,k\}} \left| t^1 r_i + z^1_i - (t^2 r_i + z^2_i) \right| \]
\[ \leq \max_{i \in \{1,\ldots,k\}} \left( |(t^1 - t^2) r_i| + \|z^1 - z^2\|_\infty \right). \]

Furthermore, for all \( i \in \{1,\ldots,k\} \) we have
\[ \bar{J}_i(\bar{u}^1) - z^2_i = \bar{J}_i(\bar{u}^1) - z^1_i + z^1_i - z^2_i = (t^1 + \frac{z^1_i - z^2_i}{r_i}) r_i, \]
so that
\[ t^2 \leq t^1 + \max_{i \in \{1,\ldots,k\}} \frac{z^1_i - z^2_i}{r_i} \leq t^1 + \frac{\max_{i \in \{1,\ldots,k\}} z^1_i - z^2_i}{\min_{i \in \{1,\ldots,k\}} r_i} \]
can be concluded. With the same arguments, we obtain
\[ t^1 \leq t^2 + \frac{\max_{i \in \{1,\ldots,k\}} z^2_i - z^1_i}{\min_{i \in \{1,\ldots,k\}} r_i}. \]
Thus, we have
\[ |\bar{t}^1 - \bar{t}^2| \leq \frac{\|z^1 - z^2\|_\infty}{\min_{i \in \{1, \ldots, k\}} r_i}. \]
Altogether, this yields
\[
\|R_{\text{opt}}(z^1) - R_{\text{opt}}(z^2)\|_\infty \leq \max_{i \in \{1, \ldots, k\}} |(\bar{t}^1 - \bar{t}^2) r_i| + \|z^1 - z^2\|_\infty
\]
\[
\leq \max_{i \in \{1, \ldots, k\}} \left( \frac{\|z^1 - z^2\|_\infty}{\min_{j \in \{1, \ldots, k\}} r_j} \right) r_i + \|z^1 - z^2\|_\infty
\]
\[
= \left( 1 + \max_{i,j \in \{1, \ldots, k\}} \frac{r_i}{r_j} \right) \|z^1 - z^2\|_\infty,
\]
which concludes the proof.

Remark 1.7.29. Due to the equivalence of the norms \(\|\cdot\|_2\) and \(\|\cdot\|_\infty\), we can conclude from Theorem 1.7.28 that the function \(R_{\text{opt}}|_{\mathcal{Z}^D_{\text{opt}}} : (\mathcal{Z}^D_{\text{opt}}, \|\cdot\|_2) \rightarrow (\mathcal{J}_{\text{opt}}, \|\cdot\|_2)\) is Lipschitz continuous with Lipschitz constant \(L_{R,2} := \sqrt{k} L_{R,\infty}\).

Remark 1.7.30. Not only is the set \(\mathcal{Z}^D_{\text{opt}}\) Pareto sufficient, but, in particular, the Theorems 1.7.26 and 1.7.28 tell us that \(R_{\text{opt}}|_{\mathcal{Z}^D_{\text{opt}}}\) is a Lipschitz continuous bijection between \(\mathcal{Z}^D_{\text{opt}}\) and the Pareto front \(\mathcal{J}_{\text{opt}}\). Therefore, the set \(\mathcal{Z}^D_{\text{opt}}\) is an 'optimal' Pareto sufficient set in the sense that we cannot remove any reference point from \(\mathcal{Z}^D_{\text{opt}}\) without losing the Pareto sufficiency of the set. Although there is no bijection between \(\mathcal{Z}^D_{\text{opt},(w),(\text{loc})}\) and \(\mathcal{J}_{\text{opt},(w),(\text{loc})}\), the set \(\mathcal{Z}^D_{\text{opt},(w),(\text{loc})}\) is still (locally) (weakly) Pareto sufficient and can thus be used for the computation of \(\mathcal{J}_{\text{opt},(w),(\text{loc})}\).

1.7.4 The Hierarchical Pascoletti-Serafini Method

Unfortunately, so far the set \(\mathcal{Z}^D_{\text{opt},(w),(\text{loc})}\) can only be characterized by the set \(U_{\text{opt},(w),(\text{loc})}\), i.e., after the computation of the (local) (weak) Pareto set. So for a numerical implementation of the method, the characterization (1.7.6) is of no use. This is comparable to the characterization of the set \(\mathcal{Z}_{\text{opt},(w)} \cap D\) for the ERPM in Theorem 1.6.31, for which the solutions to all WSPs are needed. As for the ERPM, to the best of the author’s knowledge, it is not possible to characterize the sets \(\mathcal{Z}^D_{\text{opt},(w),(\text{loc})}\) before any computations. However, there is a hierarchical way of obtaining a superset of \(\mathcal{Z}^D_{\text{opt},(w),(\text{loc})}\) without having to compute the entire (local) (weak) Pareto set first. This is explained in this section.

1.7.4.1 The Pascoletti-Serafini Method for Subproblems

In this section, we introduce the PSM for subproblems of (MOP). The notation is based on Section 1.3. Note that all results from the previous sections about the PSM can be transferred to this situation. For the purpose of brevity, we will only summarize the most important statements here. We still use the same fixed target direction \(r \in \mathbb{R}^k\) as in the previous sections.
**Theorem 1.7.32.** Let $I \subset \{1, \ldots, k\}$ be arbitrary. For a given reference point $z \in \mathbb{R}^I$, we define the Pascoletti-Serafini problem for $(\text{MOP}(I))$ by

$$\min_{(u,t)} \ t \quad \text{s.t.} \quad u \in \mathbb{U}_{\text{ad}}, \ t \in \mathbb{R}, \quad (\text{PSP}(I,z,r))$$

$$j^I(u) - z \leq t r^I.$$ 

Again, it is possible to show that $(\text{PSP}(I,z,r))$ is equivalent (in the sense of Theorem 1.7.3) to the problem

$$\min_{u \in \mathbb{U}_{\text{ad}}} \max_{i \in I} \frac{1}{r_i} \left( \hat{j}(u) - z_i \right). \quad (\text{RPSP}(I,z,r))$$

**Theorem 1.7.33.** Let $I \subset \{1, \ldots, k\}$ and $z \in \mathbb{R}^I$ be arbitrary. Furthermore, assume that there is a global (local) solution $\bar{u}$ of $(\text{RPSP}(I,z,r))$. Then it holds $\bar{u} \in \mathbb{U}^I_{\text{opt},w,(\text{loc})}$.

**Corollary 1.7.34.** Let Assumption 1.2 be satisfied and $I \subset \{1, \ldots, k\}$ as well as $z \in \mathbb{R}^I$ be arbitrary. Then $(\text{RPSP}(I,z,r))$ has a global solution $\bar{u} \in \mathbb{U}^I_{\text{opt}}$.

**Theorem 1.7.35.** Let $I \subset \{1, \ldots, k\}$ and $\bar{u} \in \mathbb{U}^I_{\text{opt},w,(\text{loc})}$ be arbitrary. Then for every $\bar{t} \in \mathbb{R}$ we have that $\bar{u}$ is a global (local) solution of $(\text{RPSP}(I,z,r))$ for the reference point $z := \hat{j}^I(\bar{u}) - \bar{t} r^I$. If even $\bar{u} \in \mathbb{U}^I_{\text{opt}}$, any other global solution $\bar{u}$ of $(\text{RPSP}(I,z,r))$ satisfies $\hat{j}^I(\bar{u}) = \hat{j}^I(\bar{u})$.

**Definition 1.7.36.** We define the solution mappings

$$Q^I_{\text{opt},w}: \mathbb{R}^I \to \mathbb{U}^I_{\text{opt},w}, \quad z \mapsto \{u \in \mathbb{U}_{\text{ad}} \mid u \text{ global sol. of } (\text{RPSP}(I,z,r))\},$$

$$Q^I_{\text{opt},w,(\text{loc})}: \mathbb{R}^I \to \mathbb{U}^I_{\text{opt},w,(\text{loc})}, \quad z \mapsto \{u \in \mathbb{U}_{\text{ad}} \mid u \text{ local sol. of } (\text{RPSP}(I,z,r))\},$$

$$Q^I_{\text{opt},(\text{loc})}: \mathbb{R}^I \to \mathbb{U}^I_{\text{opt},(\text{loc})}, \quad z \mapsto Q^I_{\text{opt},w,(\text{loc})}(z) \cap \mathbb{U}^I_{\text{opt},(\text{loc})},$$

$$R^I_{\text{opt},w,(\text{loc})}: \mathbb{R}^I \to \delta^I_{\text{opt},w,(\text{loc})}, \quad z \mapsto \hat{j}^I(Q^I_{\text{opt},w,(\text{loc})}(z)).$$

**Theorem 1.7.36.** Let $I \subset \{1, \ldots, k\}$ be arbitrary. Then the mappings $Q^I_{\text{opt},w,(\text{loc})}$ and $R^I_{\text{opt},w,(\text{loc})}$ are well-defined and it holds

$$Q^I_{\text{opt},w,(\text{loc})}(\mathbb{R}^I) = \mathbb{U}^I_{\text{opt},w,(\text{loc})}, \quad R^I_{\text{opt},w,(\text{loc})}(\mathbb{R}^I) = \delta^I_{\text{opt},w,(\text{loc})}.$$ 

Moreover, if Assumption 1.2 is satisfied, we have

$$Q^I_{\text{opt},w,(\text{loc})}(z) \neq \emptyset, \quad R^I_{\text{opt},w,(\text{loc})}(z) \neq \emptyset$$

for all $z \in \mathbb{R}^I$.

**Definition 1.7.37.** Let $I \subset \{1, \ldots, k\}$ be arbitrary. A set $Z \subset \mathbb{R}^I$ is called (locally) (weakly) Pareto sufficient for $(\text{MOP}(I))$ if it holds

$$Q^I_{\text{opt},w,(\text{loc})}(Z) \supseteq \mathbb{U}^I_{\text{opt},w,(\text{loc})}.$$
Definition 1.7.38. Let \( I \subset \{1, \ldots, k\} \) be arbitrary and recall the definition of the set \( D^I \) in Definition 1.6.53. Then for all \( K \subset \{1, \ldots, k\} \) we define the sets

\[
Z_{\text{opt.}(w), (\text{loc})}^{D^I, K} := \{ z \in D^I \mid \exists \bar{u} \in U_{\text{opt.}(w), (\text{loc})}^K : \exists t \in \mathbb{R} : z = \hat{J}^I(\bar{u}) - tr^I \}.
\]

To ease the notation, we write

\[
Z_{\text{opt.}(w), (\text{loc})}^{D^I} := Z_{\text{opt.}(w), (\text{loc})}^{D^I, I}.
\]

If \( I = \{1, \ldots, k\} \) we set

\[
Z_{\text{opt.}(w), (\text{loc})}^{D, K} := Z_{\text{opt.}(w), (\text{loc})}^{D^I, K} \quad \text{and} \quad Z_{\text{opt.}(w), (\text{loc})}^{D} := Z_{\text{opt.}(w), (\text{loc})}^{D^I, I}.
\]

Remark 1.7.39. Definition 1.7.38 does not only transfer the definition of the set \( Z_{\text{opt.}(w), (\text{loc})}^{D} \) to the framework of a subproblem \((\text{MOP}(I))\), but also refines it: The superindex \( K \) indicates that the set \( Z_{\text{opt.}(w), (\text{loc})}^{D^I, K} \) only contains the intersection points of the ray \( t \mapsto \hat{J}^I(\bar{u}) - tr^I \) with the set \( D^I \) for all \( \bar{u} \in U_{\text{opt.}(w), (\text{loc})}^K \). \( \blacklozenge \)

Definition 1.7.40. Let \( I \subset \{1, \ldots, k\} \) be arbitrary. For any \( y \in \mathbb{R}^I \) we set

\[
t^{D^I}(y) := \min_{i \in I} \frac{y_i - \bar{y}^{id}_i}{r_i} \in \mathbb{R}.
\]

Then we define the functions

\[
N^{D^I} : \hat{J}^{I}_{\text{opt.}(w), \text{loc}} \to D^I, \quad N^{D^I}(y) := y - t^{D^I}(y)r^I,
\]

\[
L^{D^I} : U^{I}_{\text{opt.}(w), \text{loc}} \to D^I, \quad L^{D^I}(u) := N^{D^I}(\hat{J}^I(u)).
\]

Lemma 1.7.41. For any \( K \subset I \) we have

\[
Z_{\text{opt.}(w), (\text{loc})}^{D^I, K} = N^{D^I}(\hat{J}^I(U_{\text{opt.}(w), (\text{loc})}^K)) = L^{D^I}(U_{\text{opt.}(w), (\text{loc})}^K) = \left\{ \hat{J}^I(\bar{u}) - t^{D^I}(\hat{J}^I(\bar{u}))r^I \mid \bar{u} \in U_{\text{opt.}(w), (\text{loc})}^K \right\}.
\]

Corollary 1.7.42. Let \( I \subset \{1, \ldots, k\} \) be arbitrary. Then the set \( Z_{\text{opt.}(w), (\text{loc})}^{D^I} \) is (locally) (weakly) Pareto sufficient for \((\text{MOP}(I))\).

Theorem 1.7.43. Let \( I \subset \{1, \ldots, k\} \) be arbitrary. The function

\[
\mathcal{R}^I_{\text{opt}} : (Z_{\text{opt.}(w), (\text{loc})}^{D^I}, \|\cdot\|_\infty) \to (\hat{J}^{I}_{\text{opt.}}, \|\cdot\|_\infty)
\]

is Lipschitz continuous with Lipschitz constant

\[
L_{\mathcal{R}^I_{\text{opt}}, \infty} := 1 + \max_{i,j \in I} \frac{r_i}{r_j}.
\]

Consequently, the function

\[
\mathcal{R}^I_{\text{opt}} : (Z_{\text{opt.}(w), (\text{loc})}^{D^I}, \|\cdot\|_\infty) \to (\hat{J}^{I}_{\text{opt.}}, \|\cdot\|_2)
\]

is Lipschitz continuous with Lipschitz constant \( L_{\mathcal{R}^I_{\text{opt}}, 2} = \sqrt{\|I\|} L_{\mathcal{R}^I_{\text{opt}}, \infty} \).
1.3.4.2 Hierarchical Characterization of $Z_{\text{opt.(w),(loc)}}^{D^I}$

Now we show how the (locally) (weakly) Pareto sufficient set of reference points $Z_{\text{opt.(w),(loc)}}^{D^I}$ can be characterized by only using the sets $U_{\text{opt.(w),(loc)}}^{K}$ for all $K \subseteq I$.

**Lemma 1.7.44.** Let $I \subset \{1, \ldots, k\}$ be arbitrary. Then we have

\[
Q_{I \text{opt.(w),(loc)}}^{I} \left( \bigcup_{K \subseteq I} Z_{\text{opt.(w),(loc)}}^{D^I,K} \right) \supset U_{I \text{opt.(w),(loc)}}^{I} \left( \bigcup_{K \subseteq I} U_{\text{opt.(w),(loc)}}^{K} \right), \tag{1.7.12}
\]

\[
Q_{I \text{opt.(w),(loc)}}^{I} \left( \bigcup_{K \subseteq I} Z_{\text{opt.(w),(loc)}}^{D^I,K} \right) = U_{I \text{opt.(w),(loc)}}^{I} \left( \bigcup_{K \subseteq I} U_{\text{opt.(w),(loc)}}^{K} \right). \tag{1.7.13}
\]

**Proof.** To show (1.7.12) (and ‘⊃’ of (1.7.13)), let

\[
\bar{u} \in U_{I \text{opt.(w),(loc)}}^{I} \setminus \bigcup_{K \subseteq I} U_{\text{opt.(w),(loc)}}^{K}
\]

be arbitrary. Define $z := L^{D^I}(\bar{u}) = J^{I}(\bar{u}) - t^{D^I}(\hat{J}^{I}(\bar{u})) r^{I}$, which implies

\[
z \in Z_{\text{opt.(w),(loc)}}^{D^I} \setminus \bigcup_{K \subseteq I} Z_{\text{opt.(w),(loc)}}^{D^I,K}
\]

according to Lemma 1.7.41. Moreover, Theorem 1.7.34 yields that $\bar{u}$ is a global (local) solution of $(\text{RPSP}(I, z, r))$, so that $\bar{u} \in Q_{I \text{opt.(w),(loc)}}^{I}(z)$ holds.

We continue by proving ’⊂’ of (1.7.13). To this end, let

\[
z \in Z_{\text{opt.(w),(loc)}}^{D^I} \setminus \bigcup_{K \subseteq I} Z_{\text{opt.(w),(loc)}}^{D^I,K}
\]

be arbitrary. By Lemma 1.7.41 (ii), there is $\bar{u} \in U_{I \text{opt}}$ with $z = J^{I}(\bar{u}) - t^{D^I}(\hat{J}^{I}(\bar{u})) r^{I}$. Due to

\[
z \notin \bigcup_{K \subseteq I} Z_{\text{opt.(w),(loc)}}^{D^I,K}
\]

it must hold $\bar{u} \notin \bigcup_{K \subseteq I} U_{\text{opt}}^{K}$. From Theorem 1.7.34, we can conclude that $\bar{u}$ is a global solution of $(\text{RPSP}(I, z, r))$ and that for any other global solution $\tilde{u}$ it holds $\hat{J}^{I}(\bar{u}) = \hat{J}^{I}(\tilde{u})$, so that $\bar{u} \in U_{I \text{opt}} \setminus \bigcup_{K \subseteq I} U_{\text{opt}}^{K}$ holds as well. In total, this implies

\[
Q_{I \text{opt.(w),(loc)}}^{I}(z) \subset U_{I \text{opt.(w),(loc)}}^{I} \setminus \bigcup_{K \subseteq I} U_{\text{opt.(w),(loc)}}^{K},
\]

which concludes the proof. \qed
Remark 1.7.45. Let \( I \subseteq \{1, \ldots, k\} \) be arbitrary. If we assume that the sets \( U^K_{\text{opt}(w),(\text{loc})} \) are already known for all \( K \subseteq I \), then according to Corollary 1.3.4 we only need to compute the sets

\[
U^I_{\text{opt}(w),(\text{loc})} \setminus \bigcup_{K \subseteq I} U^K_{\text{opt}(w),(\text{loc})}
\]

to obtain \( U^I_{\text{opt}(w),(\text{loc})} \). For this, we only require the sets of reference points

\[
Z^I_{\text{opt}(w),(\text{loc})} \setminus \bigcup_{K \subseteq I} Z^I_{K, \text{opt}(w),(\text{loc})}
\]

as shown in Lemma 1.7.44. In the following, our goal is to derive characterizations of these sets, which only use information from \( U^K_{\text{opt}(w),(\text{loc})} \) for all \( K \subseteq I \).

The next result is the key to get the characterization described in Remark 1.7.45. Given a set \( K \subseteq \{1, \ldots, k\} \), a reference point \( z \in \mathbb{R}^K \), and a global (local) solution \( \bar{u} \) of \((\text{RPSP}(K, z, r))\), it shows for any \( I \subseteq \{1, \ldots, k\} \) with \( I \supseteq K \) how reference points \( \bar{z} \in \mathbb{R}^I \) can be constructed, such that \( \bar{u} \) is also a global (local) solution of \((\text{RPSP}(I, \bar{z}, r))\). Note that this is similar to the ‘lifting’ of the reference points for the ERPM in Theorem 1.6.61. Furthermore, the proof essentially uses the same arguments as the authors in [MGGS09] do for developing the idea of trade-off limits, which they call the boundary of the Pareto front.

**Theorem 1.7.46.** Let \( K \subseteq \{1, \ldots, k\} \) and \( z \in \mathbb{R}^K \) be arbitrary. If \((\bar{u}, \bar{t})\) is a global (local) solution of \((\text{PSP}(K, z, r))\), then for any \( I \subseteq \{1, \ldots, k\} \) with \( I \supseteq K \) the tupel \((\bar{u}, \bar{t})\) is also a global (local) solution of \((\text{PSP}(I, \bar{z}, r))\) with

\[
\bar{z}_i = \begin{cases} 
  z_i, & i \in K, \\
  s_i, & i \in I \setminus K,
\end{cases}
\]

for any \( s \in \mathbb{R}^{I \setminus K} \) with \( s \geq \bar{J}^{I \setminus K}(\bar{u}) - \bar{t} r^{I \setminus K} \).

**Proof.** Let \( K \subseteq \{1, \ldots, k\} \), \( I \subseteq \{1, \ldots, k\} \) with \( I \supseteq K \) and \( z \in \mathbb{R}^K \) be arbitrary. Furthermore, let \((\bar{u}, \bar{t})\) be a global (local) solution of \((\text{PSP}(K, z, r))\). For an arbitrary \( s \in \mathbb{R}^{I \setminus K} \) with \( s \geq \bar{J}^{I \setminus K}(\bar{u}) - \bar{t} r^{I \setminus K} \) define \( \bar{z} \in \mathbb{R}^I \) by

\[
\bar{z}_i = \begin{cases} 
  z_i, & i \in K, \\
  s_i, & i \in I \setminus K.
\end{cases}
\]

Since \((\bar{u}, \bar{t})\) is a global (local) solution of \((\text{PSP}(K, z, r))\), we can conclude, in particular, that \( \bar{J}^K(\bar{u}) - \bar{z}^K \leq \bar{t} r^K \). Moreover, the definition of the reference point \( \bar{z} \) implies

\[
\bar{J}_i(\bar{u}) - \bar{z}_i = \bar{J}_i(\bar{u}) - s_i \leq \bar{J}_i(\bar{u}) - \bar{J}_i(\bar{u}) + \bar{t} r_i = \bar{t} r_i \quad \text{for all } i \in I \setminus K,
\]

so that, in total, \( \bar{J}^I(\bar{u}) - \bar{z} \leq \bar{t} r^I \) is satisfied. In particular, the tupel \((\bar{u}, \bar{t})\) is admissible for \((\text{PSP}(I, \bar{z}, r))\). Now assume that \((\bar{u}, \bar{t})\) is no global (local) solution of \((\text{PSP}(I, \bar{z}, r))\). Then there is \((\bar{u}, \bar{t})\) (in every neighborhood of \((\bar{u}, \bar{t})\)) with \( i < \bar{t} \) such that \( \bar{J}_i(\bar{u}) - \bar{z}_i \leq \bar{t} r_i \) holds for all \( i \in I \). In particular, this is valid for all \( i \in K \), which is a contradiction to \((\bar{u}, \bar{t})\) being a global (local) solution of \((\text{PSP}(K, z, r))\). Thus, \((\bar{u}, \bar{t})\) is also a global (local) solution of \((\text{PSP}(I, \bar{z}, r))\). \(\square\)
From Theorem 1.7.46, we can conclude the following sufficient conditions for a reference point not lying in $Z_{opt}^{D_I} \cup K \subseteq I$, $Z_{opt,w}^{D_I,K}$ and $Z_{opt,w}^{D_I} \cup K \subseteq I$, respectively.

**Theorem 1.7.47.** Let $I \subseteq \{1, \ldots, k\}$, $K \subseteq I$ and $z \in \mathbb{R}^I$ be arbitrary. Assume that $(\bar{u}, \bar{t})$ is a global solution of $(PSP(K, z, r))$. Furthermore, let $\tilde{z} \in D_I$ be given by

$$\tilde{z} = \begin{cases} z_j, & j \in K, \\ s_j, & j \in I \setminus K \end{cases}$$

for any $s \in \mathbb{R}^{I \setminus K}$ with $s \geq \hat{j}^{I \setminus K}(\bar{u}) - \bar{t}^{I \setminus K}$. Then we have

$$\tilde{z} \notin Z_{opt,w}^{D_I} \cup Z_{opt,w}^{D_I,K} \cup \bigcup_{L \subseteq I} Z_{opt,w}^{D_I,L}.$$ 

Additionally, if $\bar{u} \in U_{opt}^I$, then

$$\tilde{z} \notin Z_{opt,w}^{D_I} \cup Z_{opt,w}^{D_I,K} \cup \bigcup_{L \subseteq I} Z_{opt,w}^{D_I,L}$$

is satisfied.

**Proof.** Let $z \in \mathbb{R}^I$ be arbitrary and $(\bar{u}, \bar{t})$ be a global solution of $(PSP(K, z, r))$. Assume that for $\tilde{z} \in D_I$ there is $s \in \mathbb{R}^{I \setminus K}$ with $s \geq \hat{j}^{I \setminus K}(\bar{u}) - \bar{t}^{I \setminus K}$ and

$$\tilde{z} = \begin{cases} z_j, & j \in K, \\ s_j, & j \in I \setminus K \end{cases}$$

From Theorem 1.7.46, we know that $(\bar{u}, \bar{t})$ is a global solution of $(PSP(I, \tilde{z}, r))$.

We show $\tilde{z} \notin Z_{opt,w}^{D_I} \cup Z_{opt,w}^{D_I,K}$ by proving that $\tilde{z} \in Z_{opt,w}^{D_I,K}$ implies $\tilde{z} \in Z_{opt,w}^{D_I,K}$. So assume that $\tilde{z} \in Z_{opt,w}^{D_I,K}$ holds. Then there is $\bar{u} \in U_{opt,w}$ with

$$\tilde{z} = \hat{j}^I(\bar{u}) - t^I(\hat{j}^I(\bar{u})) r^I \in D_I.$$

Theorem 1.7.34 implies that $(\bar{u}, t^I(\hat{j}^I(\bar{u})))$ is a global solution of $(PSP(I, \tilde{z}, r))$. Thus, we have $\bar{t} = t^I(\hat{j}^I(\bar{u}))$ and, in particular,

$$\hat{j}^K(\bar{u}) - z = \hat{j}^K(\bar{u}) - \tilde{z}^K = t^I(\hat{j}^I(\bar{u})) r^K = \bar{t} r^K,$$

so that $(\bar{u}, t^I(\hat{j}^I(\bar{u})))$ is a global solution of $(PSP(K, z, r))$ as well. Now Theorem 1.7.32 yields that $\bar{u} \in U_{opt,w}^I$. Hence, we have $\tilde{z} \in Z_{opt,w}^{D_I,K}$ as well. Thus, we have

$$\tilde{z} \notin Z_{opt,w}^{D_I} \cup Z_{opt,w}^{D_I,K} \cup \bigcup_{L \subseteq I} Z_{opt,w}^{D_I,L}.$$
Now assume that \( \bar{u} \in \mathcal{U}^K \) holds. Again, we show \( \bar{z} \notin Z_{\text{opt}}^{D^I} \setminus Z_{\text{opt}}^{D^I,K} \) by proving that \( \bar{z} \in Z_{\text{opt}}^{D^I} \) implies \( \bar{z} \in Z_{\text{opt}}^{D^I,K} \). Assuming that \( \bar{z} \in Z_{\text{opt}}^{D^I} \), there is \( \bar{u} \in \mathcal{U}_{\text{opt}} \) with
\[
\bar{z} = \bar{J}^I(\bar{u}) - t^{D^I}(\bar{J}^I(\bar{u})) r^I \in D^I.
\]

Theorem 1.7.34 implies that \((\bar{u}, t^{D^I}(\bar{J}^I(\bar{u})))\) is a global solution of \((\text{PSP}(I, \bar{z}, r))\) and that for any other global solution \((\tilde{u}, \tilde{t})\), we have \(\bar{J}^I(\bar{u}) = \tilde{J}^I(\tilde{u})\). Thus, we obtain \(\bar{J}^I(\bar{u}) = \tilde{J}^I(\tilde{u})\), since \((\bar{u}, \tilde{t})\) is a global solution of \((\text{PSP}(I, \bar{z}, r))\). In particular, this implies \(\bar{J}^K(\bar{u}) = \tilde{J}^K(\tilde{u})\). From \(\bar{u} \in \mathcal{U}^K\), we can now infer that also \(\bar{u} \in \mathcal{U}^K\). Therefore, we have \(\bar{z} \in Z_{\text{opt}}^{D^I,K} \), which is what we had to show. Therefore, we get that
\[
\bar{z} \notin Z_{\text{opt}}^{D^I} \setminus Z_{\text{opt}}^{D^I,K} \supset Z_{\text{opt}}^{D^I} \setminus \bigcup_{L \subseteq I} Z_{\text{opt}}^{D^I,L}
\]
is satisfied. \( \square \)

The next corollary shows how Theorem 1.7.47 can be used to obtain a necessary condition for a reference point to lie in \(Z_{\text{opt.(w)}}^{D^I} \setminus \bigcup_{K \subseteq I} Z_{\text{opt.(w)}}^{D^I,K} \) by only using information of \(\bigcup_{K \subseteq I} \mathcal{U}_{\text{opt.(w)}}^K\).

**Corollary 1.7.48.** Let \( I \subseteq \{1, \ldots, k\} \) as well as \( z \in D^I \) be arbitrary. If
\[
\exists K \subseteq I : \exists \bar{u} \in \mathcal{U}_{\text{opt.(w)}}^K : \exists \beta \in \{ \gamma \in \mathbb{R}_+^I : |\gamma^K| = 0 \} : z = \mathcal{L}^{D^I}(\bar{u}) + \beta, \tag{1.7.14}
\]
then \( z \notin Z_{\text{opt.(w)}}^{D^I} \setminus Z_{\text{opt.(w)}}^{D^I,K} \supset Z_{\text{opt.(w)}}^{D^I} \setminus \bigcup_{L \subseteq I} Z_{\text{opt.(w)}}^{D^I,L} \).

**Proof.** Let \( z \in D^I \) be arbitrary and suppose that (1.7.14) is satisfied. We want to show that \( z \) can be written in the form of Theorem 1.7.47. Set \( \bar{z} := \mathcal{L}^{D^I}(\bar{u}) - t^{D^I}(\bar{J}^I(\bar{u})) r^I \). By Theorem 1.7.34, the tuple \((\bar{u}, t^{D^I}(\bar{J}^I(\bar{u})))\) is a global solution of \((\text{PSP}(K, \bar{z}^K, r))\). Furthermore, due to the condition on \( \beta \), we have \( \beta^K = 0 \), and thus
\[
z = \begin{cases} 
\bar{z}_j, & j \in K, \\
\bar{J}_j(\bar{u}) - t^{D^I}(\bar{J}^I(\bar{u})) r_j + \beta_j, & j \in I \setminus K.
\end{cases}
\]
Since \( \beta^K \geq 0 \) and \( \bar{u} \in \mathcal{U}_{\text{opt.(w)}}^K \), we can apply Theorem 1.7.47 and get
\[
z \notin Z_{\text{opt.(w)}}^{D^I} \setminus Z_{\text{opt.(w)}}^{D^I,K} \supset Z_{\text{opt.(w)}}^{D^I} \setminus \bigcup_{L \subseteq I} Z_{\text{opt.(w)}}^{D^I,L}. \quad \square
\]

**Example 1.7.49.** The reverse statement of Corollary 1.7.48 does not hold true in general. To see this, look at the example presented in Figure 1.14(a). In there, \( \bar{u}^1 \) is the minimizer of \( J_1 \) and \( \bar{u}^2 \) is the minimizer of \( J_2 \). Based on these minimizers, we can compute \( \mathcal{L}^D(\bar{u}^1) \) and \( \mathcal{L}^D(\bar{u}^2) \), respectively, and determine all reference points \( z \in D \), for which \((1.7.14)\) is satisfied. This is indicated by the pink lines in Figure 1.14(a). Now for the specific
The Pascoletti-Serafini Method

1.7

(a) The reverse statement of Corollary 1.7.48 does not hold.

(b) Corollary 1.7.48 does not hold locally.

Figure 1.14: Counter examples for modifications of Corollary 1.7.48 explained in Example 1.7.49.

Reference point \( z \in D \) in this figure, it is obvious that it holds \( z \notin \mathcal{Z}^D_{\text{opt}} \setminus \bigcup_{K \subseteq \{1,2\}} \mathcal{Z}^{D,K}_{\text{opt}} \), since \( R_{\text{opt}}(z) \) does not lie on the ray \( t \mapsto z + tr \). But we can neither write \( z \) in the form \( z = \mathcal{L}^D(\bar{u}^1) + \beta \) with \( \beta_1 = 0 \) and \( \beta_2 \geq 0 \) nor as \( z = \mathcal{L}^D(\bar{u}^2) + \beta \) with \( \beta_2 = 0 \) and \( \beta_1 \geq 0 \). Therefore, (1.7.14) does not hold.

Moreover, by looking at the same example, we also see that the statement from Corollary 1.7.48 does not hold locally. Indeed, in the situation of Figure 1.14(b), we can write \( z = \mathcal{L}^D(\bar{u}^2) + \beta \) with \( \beta_2 = 0 \) and \( \beta_1 > 0 \), so that (1.7.14) holds. However, we see that the local solution \( R_{\text{opt,loc}}(z) \) lies on the ray \( t \mapsto z + tr \) and is no local solution to any of the subproblems \( K = \{1\} \) or \( K = \{2\} \). Therefore, \( z \in \mathcal{Z}^D_{\text{opt,loc}} \setminus \bigcup_{K \subseteq \{1,2\}} \mathcal{Z}^{D,K}_{\text{opt,loc}} \).

Thus, the local version of Corollary 1.7.48 does not hold.

Remark 1.7.50. Since the corresponding local statement of Corollary 1.7.48 does not hold true in general, the reference point generation for the local (weak) Pareto front is not possible with this argument. Hence, we will focus on computing the (weak) Pareto front in the following.

In Example 1.7.49, we have seen that, in general, Corollary 1.7.48 only provides a necessary condition for reference points to lie in \( \mathcal{Z}^{D_I}_{\text{opt,(w)}} \setminus \bigcup_{K \subseteq I} \mathcal{Z}^{D_I,K}_{\text{opt,(w)}} \). Under some additional assumptions, it is possible to show a sufficient condition as well. To this end, we start with the following theorem, which is a generalization of Theorem 1.7.32 under the assumption of upper semi-continuity of the cost functions.

Assumption 1.3. The cost functions \( \hat{J}_1, \ldots, \hat{J}_k \) are upper semi-continuous.

Theorem 1.7.51. Let Assumption 1.3 be satisfied and \( I \subset \{1, \ldots, k\} \) be arbitrary. Let \( z \in \mathbb{R}^I \) be arbitrary and assume that the problem (PSP\((I, z, r)\)) has a local solution \((\bar{u}, \bar{t})\). Then it holds \( \bar{u} \in \mathcal{U}^E_{\text{opt,w,loc}} \) with \( I^\dagger := \{ i \in I \mid \hat{J}_i(\bar{u}) - z_i = \bar{t}_r \} \).
Proof. Let \( z \in \mathbb{R}^I \) be arbitrary and assume that the problem \((PSP(I, z, r))\) has a local solution \((\bar{u}, \bar{t})\). Define
\[
I^= := \{ i \in I \mid \hat{J}_i(\bar{u}) - z_i = \bar{tr}_i \} \neq \emptyset,
\]
\[
I^< := \{ i \in I \mid \hat{J}_i(\bar{u}) - z_i < \bar{tr}_i \}.
\]

If \( I^= = I \) holds, we obtain \( \bar{u} \in \mathcal{U}^I_{\text{opt}, w, \text{loc}} \) by Theorem 1.7.32. If we have \( I^= \neq I \), we prove the statement by contradiction. Assume that \( \bar{u} \notin \mathcal{U}^I_{\text{opt}, w, \text{loc}} \). Then there is a sequence \( (u_n)_{n \in \mathbb{N}} \subset \mathcal{U}_{\text{ad}} \) with \( u_n \to \bar{u} \) as \( n \to \infty \) and \( \hat{J}^= (u_n) < \hat{J}^= (\bar{u}) \) for all \( n \in \mathbb{N} \). Define
\[
\bar{\varepsilon} := \min_{i \in I^<} \left[ \bar{tr}_i - (\hat{J}_i(\bar{u}) - z_i) \right] > 0.
\]

Since the cost functions are upper semi-continuous, there is a neighborhood \( \tilde{U} \) of \( \bar{u} \) such that
\[
\hat{J}_i(u) - z_i < \hat{J}_i(\bar{u}) - z_i + \frac{\bar{\varepsilon}}{2} < \bar{tr}_i \quad \text{for all } u \in \tilde{U} \text{ and } i \in I^<.
\]
Moreover, we have \( u_n \to \bar{u} \) as \( n \to \infty \) so that there is \( N \in \mathbb{N} \) such that \( u_n \in \tilde{U} \) for all \( n \geq N \). In total, we can therefore conclude
\[
\hat{J}_i(u_n) - z_i < \hat{J}_i(\bar{u}) - z_i + \frac{\bar{\varepsilon}}{2} < \bar{tr}_i, \quad \text{for } i \in I^<,
\]
\[
\hat{J}_i(u_n) - z_i < \hat{J}_i(\bar{u}) - z_i - \bar{tr}_i, \quad \text{for } i \in I^=,
\]
for all \( n \geq N \). However, this implies that \((\bar{u}, \bar{t})\) is not a local minimizer of \((PSP(I, z, r))\), which is a contradiction to the assumption. Hence, \( \bar{u} \in \mathcal{U}^I_{\text{opt}, w, \text{loc}} \).

In the following, we need a technical assumption to prove a sufficient condition for reference points to lie in \( \mathcal{Z}^{D^I}_{\text{opt}, (w)} \setminus \bigcup_{K \subsetneq I} \mathcal{Z}^{D^I, K}_{\text{opt}, (w)} \). It ensures that given \( K \subsetneq I \) and \( u \in \mathcal{U}^K_{\text{opt}, w, \text{loc}} \), the intersection point of the ray \( t \mapsto \hat{J}(u) - t\alpha \) with the set \( D^I \) lies in the set \( D^I_i \) for some \( i \in K \). Note that this assumption was not required for the ERPM, since in there we considered the intersection points of the ray \( t \mapsto \hat{J}(u) - t\alpha \) with the set \( D^I_\alpha \), where \( \alpha \in \Delta_I \) was the weight, for which \( u \in \mathcal{W}^I_{\text{opt}, (w)}(\alpha) \). Thus, \( u \in \mathcal{U}^K_{\text{opt}, w} \) directly implied \( \alpha \cap \Delta_I = \emptyset \), so that the intersection point must lie in the set \( D^I_i \) for some \( i \in K \).

**Assumption 1.4.** Let \( I \subset \{1, \ldots, k\} \) and \( K \subsetneq I \) be arbitrary. Then we assume that for all \( u \in \mathcal{U}^K_{\text{opt}, w, \text{loc}} \) it holds \( \mathcal{L}^{D^I_i}(\bar{u}) \in D^I_i \) for some \( i \in K \).

**Theorem 1.7.52.** Let the Assumptions 1.2–1.4 be satisfied. Let \( I \subset \{1, \ldots, k\} \) and \( z \in D^I \) be arbitrary. If \( z \notin \mathcal{Z}^{D^I}_{\text{opt}, (w), (\text{loc})} \setminus \bigcup_{K \subsetneq I} \mathcal{Z}^{D^I, K}_{\text{opt}, (w), (\text{loc})} \), then
\[
\exists K \subsetneq I : \exists \bar{u} \in \mathcal{U}^K_{\text{opt}, w, \text{loc}} : \exists \beta \in \{ \gamma \in \mathbb{R}_{\geq}^I \mid \gamma^K = 0 \} : z = \mathcal{L}^{D^I}(\bar{u}) + \beta. \quad (1.7.15)
\]
1.7 The Pascoletti-Serafini Method 87

Proof. Let $I \subset \{1, \ldots, k\}$ and $z \in D^I$ be arbitrary and assume that

$$z \notin \mathcal{Z}_{\text{opt},(w),\text{loc}}^{D^I} \setminus \bigcup_{K \subseteq I} \mathcal{Z}_{\text{opt},(w),\text{loc}}^{D^I,K}.$$  

If $z \in \bigcup_{K \subseteq I} \mathcal{Z}_{\text{opt},(w),\text{loc}}^{D^I,K}$, there is $K \subseteq I$ and $\bar{u} \in \mathcal{U}_{\text{opt},(w),\text{loc}}^{K} \subset \mathcal{U}_{\text{opt},\text{loc}}^{K}$ with $z = \mathcal{L}^{D^I}(\bar{u})$, so that (1.7.15) is satisfied with $\beta = 0 \in \{\gamma \in \mathbb{R}^I_+ | \gamma^K = 0\}$.

If $z \notin \bigcup_{K \subseteq I} \mathcal{Z}_{\text{opt},(w),\text{loc}}^{D^I,K}$, then we also have $z \notin \mathcal{Z}_{\text{opt},(w),\text{loc}}^{D^I}$. By Corollary 1.7.33, the problem (PSP$(I, z, r)$) has a global solution $(\bar{u}, \bar{r})$ with $\bar{u} \in \mathcal{U}_{\text{opt}}^{I}$. Due to $z \notin \mathcal{Z}_{\text{opt},(w),\text{loc}}^{D^I}$, we cannot have $z = \bar{J}^{I}(\bar{u}) - \bar{r}I$ so that $\bar{J}^{I}(\bar{u}) - z \notin \bar{r}I$. From Theorem 1.7.51, we know that $\bar{u} \in \mathcal{U}_{\text{opt},\text{loc}}^{I}$ for

$$\emptyset \neq I^\circ := \{i \in I \mid J_i(\bar{u}) - z_i = \bar{r}_i \} \subseteq I.$$  

Now we can write $z = \bar{J}^{I}(\bar{u}) - \bar{r}I + \beta$ with $\beta := z - \bar{J}^{I}(\bar{u}) + \bar{r}I \geq 0$. In particular, $\beta_j = 0$ for all $j \in I^\circ$. We are done if we can show that $\mathcal{L}^{D^I}(\bar{u}) = \bar{J}^{I}(\bar{u}) - \bar{r}I$ holds. To this end, define $\bar{z} := \mathcal{L}^{D^I}(\bar{u}) = \bar{J}^{I}(\bar{u}) - t^{D^I}(\bar{J}^{I}(\bar{u}))r^I \in D^I$. From $z \in D^I$, we infer that there is $i \in I$ with $z \in D^I_i$. Therefore, we have

$$\bar{J}_i(\bar{u}) - \bar{r}_i + \beta_i = z_i = \bar{y}_i \leq \bar{z}_i = \bar{J}_i(\bar{u}) - t^{D^I}(\bar{J}^{I}(\bar{u}))r_i,$$  

so that $\bar{r} \geq t^{D^I}(\bar{J}^{I}(\bar{u}))$. On the other hand, there is $l \in I$ with $\bar{z}_l \in D^I_l$. Due to $\bar{u} \in \mathcal{U}_{\text{opt},\text{loc}}^{I}$, we have $l \in I^\circ$ by Assumption 1.4. In particular, $\beta_l = 0$ can be concluded. Altogether, this implies

$$\bar{J}_i(\bar{u}) - \bar{r}_l = z_l = \bar{y}_l \leq \bar{z}_l = \bar{J}_i(\bar{u}) - t^{D^I}(\bar{J}^{I}(\bar{u}))r_l,$$  

from which $\bar{r} \leq t^{D^I}(\bar{J}^{I}(\bar{u}))$ can be concluded. In total, this leads to $\bar{r} = t^{D^I}(\bar{J}^{I}(\bar{u}))$, so that $\mathcal{L}^{D^I}(\bar{u}) = \bar{J}^{I}(\bar{u}) - \bar{r}I$ holds. Consequently, we have just shown that (1.7.15) is satisfied for $K = I^\circ$, $\bar{u} \in \mathcal{U}_{\text{opt},\text{loc}}^{I}$ and $\beta = z - \bar{J}^{I}(\bar{u}) + \bar{r}I \in \{\gamma \in \mathbb{R}^I_+ | \gamma^K = 0\}$, which concludes the proof. \qed

In the next corollary, we summarize the necessary and sufficient conditions, which we have shown so far.

**Corollary 1.7.53.** Let the Assumptions 1.2–1.4 be satisfied. For all $I \subset \{1, \ldots, k\}$ we have that

$$\{z \in D^I \mid \neg(1.7.15)\} \subset \mathcal{Z}_{\text{opt},(w)}^{D^I} \setminus \bigcup_{K \subseteq I} \mathcal{Z}_{\text{opt},(w)}^{D^I,K} \subset \{z \in D^I \mid \neg(1.7.14)\},$$  

(1.16)

$$\{z \in D^I \mid \neg(1.7.15)\} \subset \mathcal{Z}_{\text{opt},(w),\text{loc}}^{D^I} \setminus \bigcup_{K \subseteq I} \mathcal{Z}_{\text{opt},(w),\text{loc}}^{D^I,K}.$$  

(1.17)

**Remark 1.7.54.** In general, the subset relations in Corollary 1.7.53 are strict, i.e., there is a gap between the sufficient and the necessary condition, see Figure 1.15. The reason
for this is that, in general, it holds \( U_{opt,w,loc}^K \subseteq U_{opt,w,loc}^I \). Therefore, if we want to ensure that we obtain all reference points in \( Z_{opt,(w)}^D \setminus \bigcup_{K \subseteq I} Z_{opt,(w)}^{D,I,K} \), we can only use the necessary condition \( (1.7.14) \). Note that this only influences the computational efficiency, but not the validity of the procedure itself. Indeed, we have

\[
Q_{opt,(w)} \left( \{z \in D^I \mid \neg(1.7.14)\} \right) \subset U_{opt,(w)}^I,
\]

so that solving \((PSP(I,z,r))\) for all reference points in \( \{z \in D^I \mid \neg(1.7.14)\} \) still provides us with (weakly) Pareto optimal points.

\[\text{(a) Sufficient condition } \{z \in D \mid \neg(1.7.15)\}.\]

\[\text{(b) The set } Z_{opt}^D \setminus \bigcup_{K \subseteq I} Z_{opt}^{D,I,K}.\]

\[\text{(c) Necessary condition } \{z \in D \mid \neg(1.7.14)\}.\]

**Figure 1.15:** Illustration of sufficient and necessary conditions for the set of reference points \( Z_{opt}^D \setminus \bigcup_{K \subseteq I} Z_{opt}^{D,I,K} \).

The next theorem will help us to improve the computational efficiency by detecting reference points, which turn out to lie in the difference of the sets \( \{z \in D^I \mid \neg(1.7.14)\} \) and \( Z_{opt,(w)}^D \setminus \bigcup_{K \subseteq I} Z_{opt,(w)}^{D,I,K} \) and are consequently redundant for the computation of the (weak) Pareto front. A similar result was used in [DK19] to detect redundant directions \( r \) from the parameter set of target directions.
Theorem 1.7.55. Let $I \subset \{1, \ldots, k\}$ and $z \in \mathbb{R}^I$ be arbitrary. If there is a global (local) solution $(\tilde{u}, \tilde{I})$ of (PSP($I, z, r))$ with $\tilde{j}^I(\tilde{u}) - z \leq \tilde{r}^I$, then $(\tilde{u}, \tilde{I})$ is also a global (local) solution of (PSP($I, z, r))$ for every reference point $\tilde{z} \in [z - (\tilde{r}^I - (\tilde{j}^I(\tilde{u}) - z)), z]$. Moreover, if $(\tilde{u}, \tilde{I})$ is a global solution of (PSP($I, z, r))$ we have

$$Q_{opt,w}^I(\tilde{z}) \subset Q_{opt,w}^I(z)$$

for all $\tilde{z} \in [z - (\tilde{r}^I - (\tilde{j}^I(\tilde{u}) - z)), z]$.

Proof. Let $I \subset \{1, \ldots, k\}$ and $z \in \mathbb{R}^I$ be arbitrary, and assume that there is a global (local) solution $(\tilde{u}, \tilde{I})$ of (PSP($I, z, r))$ with $\tilde{j}^I(\tilde{u}) - z \leq \tilde{r}^I$. Let $\tilde{z} \in [z - (\tilde{r}^I - (\tilde{j}^I(\tilde{u}) - z)), z]$ be arbitrary. Note that it holds

$$\tilde{j}^I(\tilde{u}) - \tilde{z} \leq \tilde{j}^I(\tilde{u}) - (z - (\tilde{r}^I - (\tilde{j}^I(\tilde{u}) - z))) = \tilde{r}^I,$$

so that $(\tilde{u}, \tilde{I})$ is admissible for (PSP($I, \tilde{z}, r))$. Assume that there is $(\hat{u}, \hat{I})$ (in every neighborhood of $(\tilde{u}, \tilde{I})$) with $\hat{I} < \tilde{I}$ and $\hat{j}^I(u') - \hat{z} \leq \hat{r}^I$. Then we have

$$\hat{j}^I(\hat{u}) - z \leq \hat{j}^I(u') - \hat{z} \leq \hat{r}^I,$$

which is a contradiction to $(\tilde{u}, \tilde{I})$ being a global (local) solution of (PSP($I, z, r))$. Thus, $(\tilde{u}, \tilde{I})$ is a global (local) solution of (PSP($I, \tilde{z}, r))$.

To show the second part, assume that $(\tilde{u}, \tilde{I})$ is a global solution of (PSP($I, z, r))$. In the first part, we have seen that $(\tilde{u}, \tilde{I})$ is also a global solution of (PSP($I, \tilde{z}, r))$, i.e., it holds $\tilde{u} \in Q_{opt,w}^I(\tilde{z})$. Now let $\hat{u} \in Q_{opt,w}^I(\tilde{z})$ be arbitrary. Then we have

$$\hat{I} = \hat{I} := \max_{i \in I} \frac{1}{r_i}(\hat{j}_i(\hat{u}) - \hat{z}_i),$$

so that

$$\hat{j}^I(\hat{u}) - z \leq \hat{j}^I(u') - \hat{z} \leq \hat{r}^I = \tilde{r}^I.$$

This implies that $(\hat{u}, \hat{I})$ is a global solution of (PSP($I, z, r))$ as well. Therefore, we have $\hat{u} \in Q_{opt,w}^I(\tilde{z})$, which concludes the proof. $\square$

Remark 1.7.56. Theorem 1.7.55 can be applied to detect and remove reference points in the following way: If we have solved (PSP($I, z, r))$ for some reference point $z \in \mathbb{R}^I$, and there is a global solution $(\tilde{u}, \tilde{I})$ satisfying $\tilde{j}^I(\tilde{u}) - z \leq \tilde{r}^I$, then we do not have to solve the problems (PSP($I, \tilde{z}, r))$ for any $\tilde{z} \in [z - (\tilde{r}^I - (\tilde{j}^I(\tilde{u}) - z)), z]$, since $Q_{opt,w}^I(\tilde{z}) \subset Q_{opt,w}^I(z)$, see also Figure 1.16. $\Diamond$

1.7.4.3 Special Case: Convex Problems

In this section, we will show that the gap between the sufficient and necessary conditions presented in Corollary 1.7.53 is closed in the setting of convex MOPs, i.e., if Assumption 1.1 is satisfied. Note that we have $U_{opt,loc}^I = U_{opt}^I$ and $U_{opt,w,loc}^I = U_{opt,w}^I$ for all $I \subset \{1, \ldots, k\}$ in this case. Moreover, in this situation, it is possible to show the following identity, cf. Lemma 1.6.30 (iii).
Figure 1.16: Detecting redundant reference points by using Theorem 1.7.55. Left: For the solution \((\bar{u}, \bar{t})\) of \((\text{PSP}(z,r))\) it holds \(\hat{J}(\bar{u}) - z \not\leq \bar{t}r\). Right: By Theorem 1.7.55, \((\bar{u}, \bar{t})\) is also a solution of \((\text{PSP}(\tilde{z},r))\) for the reference point \(\tilde{z} = z - (\bar{t}r - (\hat{J}(\bar{u}) - z))\). The same holds true for every reference point in the set \([z - (\bar{t}r - (\hat{J}(\bar{u}) - z)), z]\), which is colored in light blue. Thus, these reference points are redundant.

**Theorem 1.7.57.** Assume that Assumption 1.1 holds and let \(I \subset \{1, \ldots, k\}\) be arbitrary. Then

\[
\mathcal{Z}_{\text{opt}}^{D,I} \setminus \bigcup_{K \subseteq I} \mathcal{Z}_{\text{opt},w}^{D,I,K} = \mathcal{Z}_{\text{opt}}^{D,I} \setminus \bigcup_{K \subseteq I} \mathcal{Z}_{\text{opt},w}^{D,I,K}.
\]

**Proof.** From Lemma 1.7.41 (ii), we induce that we only need to show

\[
\mathcal{U}_{\text{opt},w}^{I} \setminus \bigcup_{K \subseteq I} \mathcal{U}_{\text{opt},w}^{K} = \mathcal{U}_{\text{opt}}^{I} \setminus \bigcup_{K \subseteq I} \mathcal{U}_{\text{opt}}^{K}.
\]

However, this is exactly the statement of Corollary 1.3.6, which concludes the proof. \(\square\)

Under Assumption 1.1, we can also prove that it is equivalent to use \(\mathcal{U}_{\text{opt}}^{K}\) and \(\mathcal{U}_{\text{opt},w}^{K}\) in the condition (1.7.14).

**Corollary 1.7.58.** Let Assumption 1.1 be satisfied and \(I \subset \{1, \ldots, k\}\) be arbitrary. Then it is equivalent to use \(\mathcal{U}_{\text{opt}}^{K}\) and \(\mathcal{U}_{\text{opt},w}^{K}\) in the condition (1.7.14).

**Proof.** Let \(I \subset \{1, \ldots, k\}\) be arbitrary. We only need to show that

\[
\bigcup_{K \subseteq I} \mathcal{U}_{\text{opt}}^{K} = \bigcup_{K \subseteq I} \mathcal{U}_{\text{opt},w}^{K}.
\]

Again, this can be concluded by repeatedly using the identity (1.3.3) from Lemma 1.3.5. \(\square\)
So far, we have shown in Theorem 1.7.57 and Corollary 1.7.58 that
\[ \{ z \in D^I \mid -(1.7.14) \} \subset Z^{D^I}_{\text{opt}} \cup \bigcup_{K \subseteq I} Z^{D^I_{opt,K}} = Z^{D^I_{opt,w}} \cup \bigcup_{K \subseteq I} Z^{D^I_{opt,w,K}}. \]

Our goal is to prove that even equality holds true. To this end, we intend to show that the statement of Theorem 1.7.52 is satisfied if Assumptions 1.1 and 1.4 hold. To this end, we first prove a modified version of Theorem 1.7.51. In the proof we use that the PSP can be connected to the WSP together with the statement of the following corollary, cf. Corollary 1.7.33.

**Corollary 1.7.59.** Let Assumption 1.1 be satisfied and \( I \subset \{1, \ldots, k\} \) as well as \( z \in \mathbb{R}^I \) be arbitrary. Then \((RPSP(I, z, r))\) has a global solution \( \bar{u} \in U^I_{\text{opt}} \).

**Proof.** This follows directly from Corollary 1.7.33 and the fact that Assumption 1.1 is stronger than Assumption 1.2.

**Theorem 1.7.60.** Assume that Assumption 1.1 is satisfied and let \( I \subset \{1, \ldots, k\} \) be arbitrary. Let \( z \in \mathbb{R}^I \) be arbitrary and denote by \((\bar{u}, \bar{t})\) a global solution of \((PSP(I, z, r))\).

Then we have \( \bar{u} \in U^I_{\text{opt}, w} \) with \( I^w := \{ i \in I \mid \hat{J}_i(\bar{u}) - z_i = \bar{tr}_i \} \).

**Proof.** Let \( z \in \mathbb{R}^I \) be arbitrary and denote by \((\bar{u}, \bar{t})\) a global solution of \((PSP(I, z, r))\). Define
\[
I^w := \{ i \in I \mid \hat{J}_i(\bar{u}) - z_i = \bar{tr}_i \} \neq \emptyset,
\]
\[
I^< := \{ i \in I \mid \hat{J}_i(\bar{u}) - z_i < \bar{tr}_i \}.
\]

Choose \( \beta \in \mathbb{R}^I_{\geq} \) with \( \hat{J}^I(\bar{u}) - z + \beta = \bar{r}^I \). By definition, it holds \( \beta_i = 0 \) if and only if \( i \in I^w \).

In the next step, we show that \( \bar{u} \) is a solution to a WSP for some weight \( \gamma \in \Delta_I \). To see this, we note that it holds
\[
\left( \hat{J}^I(\bar{u}) - z + \beta \right) \cap \mathbb{R}^I_< = \emptyset,
\]

since otherwise there would be \( u \in U_{\text{ad}} \) with \( \hat{J}^I(u) - \hat{J}^I(\bar{u}) - \beta < 0 \), implying that
\[
\hat{J}^I(u) - z < \hat{J}^I(\bar{u}) + \beta - z = \bar{r}^I,
\]

which is a contradiction to \((\bar{u}, \bar{t})\) being a global solution of \((PSP(I, z, r))\). Since the set \( \hat{J}^I(\bar{u}) - z + \beta \) is convex (see Lemma 1.5.3), and the set \( \mathbb{R}^I_< \) is convex and open, we can use a hyperplane separation result (see, e.g., [Jah11, Theorem 3.14]) to show that there is a weight \( \gamma \in \Delta_k \) such that
\[
\langle y - \hat{J}^I(\bar{u}) - \beta + x, \gamma \rangle \geq 0 \quad \text{for all } y \in \hat{J}^I(U_{\text{ad}}) \text{ and } x \in \mathbb{R}^I_>.
\]
Proof. Let \( \hat{y} = \hat{J}^I(\hat{u}) \) and \( x = 0 \) into (1.7.18), we get \(-\beta, \gamma \geq 0\). Since \( \beta, \gamma \geq 0 \), this implies \( \langle \beta, \gamma \rangle = 0 \) so that \( \gamma_i = 0 \) for all \( i \in I^K \). Moreover, this implies that
\[ \langle y, \gamma \rangle \geq \langle \hat{J}^I(\hat{u}), \gamma \rangle \quad \text{for all} \quad y \in \hat{J}^I(\mathcal{U}_{\text{ad}}). \]

Hence, \( \hat{u} \) is a solution to (WSP(\( I, \gamma \))). Due to \( \{ i \mid \gamma_i > 0 \} \subset I^= \), with the help of Corollary 1.5.21, we can conclude \( u \in \mathcal{U}_{\text{opt}, w}^I \), which is what we had to show. \( \square \)

Theorem 1.7.61. Let the Assumptions 1.1 and 1.4 be satisfied and \( I \subset \{1, \ldots, k\} \) be arbitrary. For any \( z \in D^I \) with
\[ z \notin \bigcup_{K \subseteq I} Z_{\text{opt}}^D \bigcup_{K \subseteq I} Z_{\text{opt}, w}^{D^I, K} = Z_{\text{opt}, w}^D \bigcup_{K \subseteq I} Z_{\text{opt}, w}^{D^I, K} \]
the condition
\[ \exists K \subseteq I : \exists \hat{u} \in \mathcal{U}_{\text{opt}}^K : \exists \beta \in \{ \gamma \in \mathbb{R}_+^I \mid \gamma^K = 0 \} : z = \mathcal{L}^{D^I}(\hat{u}) + \beta \quad (1.7.19) \]
is satisfied.

Proof. Let \( I \subset \{1, \ldots, k\} \) and \( z \in D^I \) be arbitrary. Assume that
\[ z \notin \bigcup_{K \subseteq I} Z_{\text{opt}}^D \bigcup_{K \subseteq I} Z_{\text{opt}, w}^{D^I, K} = Z_{\text{opt}, w}^D \bigcup_{K \subseteq I} Z_{\text{opt}, w}^{D^I, K} \]
and denote by \( (\hat{u}, \hat{t}) \) a global solution of (PSP(\( I, z, r \))) with \( \hat{u} \in \mathcal{U}_{\text{opt}}^I \), which exists by Corollary 1.7.59.

Case 1: \( \hat{J}^I(\hat{u}) - z = \hat{t}r^I \)

From \( z \notin \bigcup_{K \subseteq I} Z_{\text{opt}}^D \bigcup_{K \subseteq I} Z_{\text{opt}, w}^{D^I, K} \) and \( z = \mathcal{L}^{D^I}(\hat{u}) \), we can conclude that there is \( K \subseteq I \) with \( \hat{u} \in \mathcal{U}_{\text{opt}}^K \). But then we have \( z = \mathcal{L}^{D^I}(\hat{u}) + \beta \) with \( \beta = 0 \) and we are done.

Case 2: \( \hat{J}^I(\hat{u}) - z \leq \hat{t}r^I \)

By Theorem 1.7.60, we have \( \hat{u} \in \mathcal{U}_{\text{opt}, w}^I \) with \( I^= := \{ j \in I \mid \hat{J}_j(\hat{u}) - z_j = \hat{t}r_j \} \). By the identity (1.3.3), we can conclude that there is \( K \subset I^= \subseteq I \) with \( \hat{u} \in \mathcal{U}_{\text{opt}}^K \). We can write \( z = \hat{J}^I(\hat{u}) \leq \hat{t}r^I + \beta \) with \( \beta := z - \hat{J}^I(\hat{u}) \hat{t}r^I \geq 0 \). By definition, it holds \( \beta_j = 0 \) for all \( j \in I^= \), so that, in particular, \( \beta_j = 0 \) for all \( j \in K \) can be concluded. If we can prove that \( \mathcal{L}^{D^I}(\hat{u}) = \hat{J}^I(\hat{u}) - \hat{t}r^I \) holds, we are done. This can be shown in the same way as in the proof of Theorem 1.7.52. \( \square \)

Now we can formulate the following result, which states that the necessary and sufficient condition for the set \( Z_{\text{opt}}^D \bigcup_{K \subseteq I} Z_{\text{opt}, w}^{D^I, K} \) are equivalent.

Corollary 1.7.62. Let the Assumptions 1.1 and 1.4 be satisfied and \( I \subset \{1, \ldots, k\} \) be arbitrary. Then we have
\[ \{ z \in D^I \mid -(1.7.19) \} = Z_{\text{opt}}^D \bigcup_{K \subseteq I} Z_{\text{opt}, w}^{D^I, K} = Z_{\text{opt}, w}^D \bigcup_{K \subseteq I} Z_{\text{opt}, w}^{D^I, K}. \]
1.7.4.4 The Hierarchical Algorithm

In this section, we use the results from the previous sections to formulate the hierarchical Algorithm 3 for solving (MOP) by the PSM. The general outline of the algorithm follows Remark 1.3.10 and is the same as for Algorithm 1.

Algorithm 3 Solving (MOP) by the PSM

1: \textbf{for} \( j = 1 : k \) \textbf{do}
2: \hspace{1em} Set \( I := \{j\} \);
3: \hspace{1em} Compute \( U_{\text{opt},w}(I) = \{u \mid u \text{ minimizes } J_j\} \);
4: \hspace{1em} Compute \( y_j^{\text{id}} \) and \( \tilde{y}_j^{\text{id}} \);
5: \hspace{1em} Set \( UTZ(I) = \{(u, d_j, L^{D_j}(u)) \mid u \in \hat{U}_{\text{opt},w}(I)\} = \{(u, d_j, \tilde{y}_j^{\text{id}}) \mid u \in \hat{U}_{\text{opt},w}(I)\} \);
6: \textbf{end for}
7: \textbf{for} \( i = 2 : k \) \textbf{do}
8: \hspace{1em} \textbf{for all} \( I \subset \{1, \ldots, k\} \) \textbf{with} \( |I| = i \) \textbf{do}
9: \hspace{2em} Initialize \( U_{\text{opt},w}(I) = \bigcup_{K \subseteq I} U_{\text{opt},w}(K) \) and \( UTZ(I) = \emptyset \);
10: \hspace{2em} Compute the reference points \( Z(I) := \{z \in D^I \mid \neg(1.7.20)\} \);
11: \hspace{2em} \textbf{while} \( Z(I) \neq \emptyset \) \textbf{do}
12: \hspace{3em} Choose \( z \in Z(I) \) and remove \( z \) from \( Z(I) \);
13: \hspace{3em} Solve \((\text{PSP}(I, z, r))/\text{RPSP}(I, z, r)\);
14: \hspace{3em} Set \( \hat{U}_{\text{opt},w}(I) \leftarrow U_{\text{opt},w}(I) \cup Q^I_{\text{opt},w}(z) \);
15: \hspace{3em} Set \( UTZ(I) \leftarrow UTZ(I) \cup \{(\bar{u}, \bar{t}, z) \mid (\bar{u}, \bar{t}) \text{ gl. sol. of } (\text{PSP}(I, z, r))\} \);
16: \hspace{3em} Add solutions of PSPs with respect to redundant reference points (cf. Remark 1.7.56): Set \( UTZ(I) \leftarrow UTZ(I) \cup \{(\bar{u}, \bar{t}, z) \mid (\bar{u}, \bar{t}) \text{ gl. sol. of } (\text{PSP}(I, z, r))\} \), \( z \in Z(I) \cap [z - (\bar{t}r^I - (J^I(\bar{u}) - z))] \);
17: \hspace{3em} Remove redundant reference points (cf. Remark 1.7.56): Set \( Z(I) \leftarrow Z(I) \setminus [z - (\bar{t}r^I - (J^I(\bar{u}) - z))] \) for all \( \bar{u} \in Q^I_{\text{opt},w}(z) \);
18: \hspace{2em} \textbf{end while}
19: \hspace{1em} \textbf{end for}
20: \textbf{end for}
21: \textbf{if} \( \text{computeParetoFront == true} \) \textbf{then}
22: \hspace{1em} Remove all \( u \in \hat{U}_{\text{opt},w}(\{1, \ldots, k\}) \) with \( u \notin U_{\text{opt}} \) by a non-dominance test;
23: \textbf{end if}

In Algorithm 3, we do not use the condition (1.7.14) to exclude reference points in line 10, but
\[
\exists K \subset I: \exists (\bar{u}, \bar{t}, z) \in UTZ(K): z^K = z \land z^{J \setminus K} \geq J^{I \setminus K}(\bar{u}) - \bar{t}r^{I \setminus K}. \quad (1.7.20)
\]
To this end, note that we have that
\[
z \text{ satisfies } (1.7.20) \Rightarrow z \notin Z_{\text{opt},w}^{D^I} \setminus \bigcup_{K \subseteq I} Z_{\text{opt},w}^{D^I,K}
\]
by Theorem 1.7.47, since the set \( UTZ(K) \) only contains tuples \((\bar{u}, \bar{t}, z)\), for which \((\bar{u}, \bar{t})\) is a global solution of \((\text{PSP}(K, z, r))\), see the lines 15 and 16 of Algorithm 3.
The reason for not using (1.7.14) is that the gap between \( Z_{opt(w)}^I \setminus \bigcup_{K \subseteq I} Z_{opt(w)}^{D^I,K} \) and the set \( \{ z \in D^I \mid - (1.7.14) \} \) is possibly large in practice, see, e.g., Figure 1.19(a),(c). This is due to the fact that condition (1.7.14) is a special case of the statement of Theorem 1.7.47. Indeed, the condition (1.7.14) only uses the specific reference point \( \bar{z} = \mathcal{L}^{D^I}(\bar{u}) \) for each given \( \bar{u} \in U_{K \subseteq I}^{opt(w)} \), whereas the statement of Theorem 1.7.47 is valid for any tupel \( (\bar{u}, \bar{t}, \bar{z}) \), for which \( \bar{z} \in \mathbb{R}^K \) and \( (\bar{u}, \bar{t}) \) is a global solution of (PSP(\( K, \bar{z}, r \))). This leads to the idea of the sets \( \mathcal{UTZ}(K) \) in Algorithm 3, in which these tupels are saved for each reference point. In practice, this allows for a more efficient removal of reference points, cf. Figure 1.19(b),(d).

With this clarification, it is now possible to prove analytically that Algorithm 3 solves (MOP).

**Theorem 1.7.63.** Algorithm 3 solves (MOP) in the sense that it holds

\[
\tilde{U}_{opt,w}^I(I) = U_{opt,w}^I \tag{1.7.21}
\]

for all \( I \subset \{1, \ldots, k\} \). Moreover, we have

\[
\forall \bar{u} \in U_{opt,w}^I \setminus \bigcup_{K \subseteq I} U_{opt,w}^K : (\bar{u}, t^{D^I}(J^I(\bar{u})), \mathcal{L}^{D^I}(\bar{u})) \in \mathcal{UTZ}(I)
\]

for all \( I \subset \{1, \ldots, k\} \).

**Proof.** We prove (1.7.21) by induction over the cardinality \( i \) of the set \( I \).

\( i = 1 \): For \( I = \{j\} \) for any \( j \in \{1, \ldots, k\} \), we have

\[
U_{opt,w}^I = \{ u \mid u \text{ minimizes } J_j \}.
\]

This set is computed in line 3.

\( (1, \ldots, i - 1) \to i \) \( (2 \leq i \leq k) \): Let \( I \subset \{1, \ldots, k\} \) with \( |I| \geq 2 \) be arbitrary. By induction assumption, we have

\[
\tilde{U}_{opt,w}^K(K) = U_{opt,w}^K \quad \text{for all } K \subseteq I.
\]

Thus, the set \( \tilde{U}_{opt,w}^I(I) \) is initialized by \( \tilde{U}_{opt,w}^I(I) = \bigcup_{K \subseteq I} U_{opt,w}^K \) in line 9. Moreover, in Lemma 1.7.44, we saw that we only require the sets of reference points \( Z_{opt,w}^{D^I} \setminus \bigcup_{K \subseteq I} Z_{opt,w}^{D^I,K} \) to compute \( U_{opt,w}^I \setminus \bigcup_{K \subseteq I} U_{opt,w}^K \) by solving the respective PSPs. By Theorem 1.7.47, it holds

\[
Z_{opt,w}^{D^I} \setminus \bigcup_{K \subseteq I} Z_{opt,w}^{D^I,K} \subset \{ z \in D^I \mid -(1.7.20) \} =: Z(I).
\]

Therefore, in the lines 13 and 14 we ensure that

\[
U_{opt,w}^I \setminus \bigcup_{K \subseteq I} U_{opt,w}^K \subset \tilde{U}_{opt,w}(I)
\]
1.7 The Pascoletti-Serafini Method

holds as well. Altogether, this proves \( \tilde{u}_{\text{opt},w}(I) = \mathcal{U}_{\text{opt},w}^I \), which is what we had to show. Note at this point that the lines 16 and 17 only increase the efficiency by using the result from Theorem 1.7.55 to detect and remove redundant reference points from \( Z(I) \).

From \( Z_{\text{opt},w}^D \setminus \bigcup_{K \subseteq I} Z_{\text{opt},w}^{D,K} \subset Z(I) \), together with Theorem 1.7.34 it is also clear that

\[
\forall \bar{u} \in \mathcal{U}_{\text{opt},w}^I \setminus \bigcup_{K \subseteq I} \mathcal{U}_{\text{opt},w}^K: (\bar{u}, t^D(I^I(\bar{u})), \mathcal{L}^D(\bar{u})) \in \mathcal{UTZ}(I)
\]

is satisfied for all \( I \subset \{1, \ldots, k\} \).

To conclude this section, we show analytically that using the condition (1.7.20) in line 10 of Algorithm 3 to remove reference points is indeed superior to using the condition (1.7.14).

**Theorem 1.7.64.** Let \( I \subset \{1, \ldots, k\} \) be arbitrary. Then it holds

\[
Z_{\text{opt},w}^D \setminus \bigcup_{K \subseteq I} Z_{\text{opt},w}^{D,K} \subset \{ z \in D^I \mid \neg(1.7.20) \} \subset \{ z \in D^I \mid \neg(1.7.14) \}.
\]

**Proof.** This follows directly from

\[
\forall \bar{u} \in \mathcal{U}_{\text{opt},w}^I \setminus \bigcup_{K \subseteq I} \mathcal{U}_{\text{opt},w}^K: (\bar{u}, t^D(I^I(\bar{u})), \mathcal{L}^D(\bar{u})) \in \mathcal{UTZ}(I),
\]

which was shown in Theorem 1.7.63.

**Remark 1.7.65.** Let us summarize the results for the PSM in the terminology of Section 1.4.1: We choose the families of scalarization functions \((g_{z,r}^I)_{z \in Z(I)}\) for solving (MOP(I)) for all \( I \subset \{1, \ldots, k\} \) by the PSM. Moreover, the set of reference points \( Z(I) \) can be obtained in a hierarchical way by using solutions of subproblems, cf. Algorithm 3. Furthermore, we have shown the Pareto admissibility and sufficiency of these families of scalarization functions. In general, the mapping \( R_{\text{opt},w}^I \) is no bijection between the set of reference points \( Z(I) \) and the (weak) Pareto front \( J_{\text{opt},w}^I \), since in most cases \( Z_{\text{opt},w}^D \subseteq Z(I) \). However, for convex problems, we could show that it holds \( Z_{\text{opt},w}^D = Z(I) \), since the sufficient and necessary conditions for choosing reference points coincide, see Corollary 1.7.62. In this case, the function \( R_{\text{opt}}^I \) is indeed a bijection between \( Z(I) \) and the Pareto front \( J_{\text{opt}}^I \), cf. Theorem 1.7.26.

1.7.4.5 Numerical Implementation

In this section, we discuss how Algorithm 3 can be implemented numerically. The goal is to obtain finite approximations

\[
\mathcal{U}_{\text{opt},w}^{\text{num}} = \{ \bar{u}^1, \ldots, \bar{u}^N \} \quad \text{and} \quad \mathcal{J}_{\text{opt},w}^{\text{num}} = \{ \hat{J}(\bar{u}^1), \ldots, \hat{J}(\bar{u}^N) \}
\]

of \( \mathcal{U}_{\text{opt},w} \) and \( \mathcal{J}_{\text{opt},w} \), respectively, which fulfill certain quality criteria, cf. Section 1.4.2.
The procedure strongly follows Section 1.6.4.3, where the numerical implementation of the ERPM was discussed. We start with a result that restricts the possible domain of the reference points in $Z_{\text{opt},w}^{D_I}$, cf. Lemma 1.6.66.

**Lemma 1.7.66.** Let $I \subset \{1,\ldots,k\}$ be arbitrary. If we set $\tilde{\bar{t}}_i := \frac{\tilde{d}}{r_i}$ for all $i \in I$, where $\tilde{d} = \tilde{y}^{id} - \tilde{y}^{id}$ is the shifting vector (see Definition 1.6.25), then

$$Z_{\text{opt},w}^{D_I} \subset \bigcup_{i \in I} \left\{ z \in D_i^I \mid \forall j \in I \setminus \{i\} : z_j \leq y_j^{\text{nad},I,w} - \tilde{\bar{t}}_j \right\} . \quad (1.7.22)$$

**Proof.** Let $I \subset \{1,\ldots,k\}$ be arbitrary. Furthermore, let $z \in Z_{\text{opt},w}^{D_I}$ be arbitrary, so that there is $i \in I$ with $z \in D_i^I$. Moreover, there is $u \in U_{\text{opt},w}^I$ such that $z = \hat{J}_I(u) - \bar{r}_I$ for $t := (\hat{J}_i(u) - \tilde{y}^{id}_i)/r_i \geq (y^{id}_i - \tilde{y}^{id}_i)/r_i = \tilde{\bar{t}}_i$. But then

$$z_j = \hat{J}_j(u) - \bar{r}_j \leq y_j^{\text{nad},I,w} - \bar{r}_j \leq y_j^{\text{nad},w} - \tilde{\bar{t}}_j$$

for all $j \in I \setminus \{i\}$, which concludes the proof.

As a next step, we introduce a grid on $D_I^I$, which covers the possible domain of the set $Z_{\text{opt},w}^{D_I}$, cf. Definition 1.6.67.

**Definition 1.7.67.** Let $I \subset \{1,\ldots,k\}$ be arbitrary. For a given grid size $h > 0$ and any $i \in I$, we define

$$Z_i^{h,I} := \left\{ z \in D_i^I \mid \forall j \in I \setminus \{i\} : \left( \exists k \geq 0 : z_j = y_{id}^j + \frac{h}{2} + k h \right) \& \left( z_j \leq y_j^{\text{nad},I,w} - \tilde{\bar{t}}_j \right) \right\} .$$

Furthermore, we set

$$Z^{h,I} := \bigcup_{i \in I} Z_i^{h,I} .$$

If $I = \{1,\ldots,k\}$, we write $Z^h := Z^{h,I}$.

The idea is to only choose reference points that lie on the grid $Z^{h,I}$. Therefore, we replace the condition (1.7.20) in line 10 of Algorithm 3 by

$$\exists K \subset I : \exists (\bar{u}, \bar{r}, z) \in UTZ_{\text{num}}^I(K) : z^K = z^I_K \& z^{I \setminus K} \geq \hat{J}^{I \setminus K}(\bar{u}) - \bar{r}^{I \setminus K} , \quad (1.7.23)$$

where $UTZ_{\text{num}}^I(K)$ is a numerical approximation of $UTZ(K)$. Now the numerical algorithm for solving (MOP) by the PSM can be formulated (see Algorithm 4).

It is possible to show that the two conditions (1.7.20) and (1.7.23) are equivalent if we restrict ourselves to reference points on the grid $Z^{h,I}$. This verifies that all relevant reference points on the grid $Z^{h,I}$ are selected in line 10 of Algorithm 4.

**Theorem 1.7.68.** Let $I \subset \{1,\ldots,k\}$ with $|I| \geq 2$ be arbitrary. Then it holds

$$Z(I) \cap Z^{h,I} = Z^{\text{num}}(I) .$$
Algorithm 4 Solving (MOP) numerically by the PSM

1: for \( j = 1 : k \) do
2: Set \( I := \{ j \} \);
3: Compute \( U_{\text{opt},w}^\text{num}(I) = \{ u \mid u \text{ minimizes } J_j \} \);
4: Compute \( \tilde{y}_j^i \text{ and } \hat{y}_j^i \);
5: Set \( UT\overline{Z}^\text{num}(I) = \{(u, d_j, \mathcal{L}^{D_j}(u)) \mid u \in U_{\text{opt},w}^\text{num}(I) \} = \{(u, d_j, \tilde{y}_j^i) \mid u \in U_{\text{opt},w}^\text{num}(I) \} \);
6: end for
7: for \( i = 2 : k \) do
8: for all \( I \subset \{1, \ldots, k\} \) with \( |I| = i \) do
9: Initialize \( U_{\text{opt},w}^\text{num}(I) = \bigcup_{K \subseteq I} U_{\text{opt},w}^\text{num}(K) \) and \( UT\overline{Z}^\text{num}(I) = \emptyset \);
10: Compute the reference points \( Z_{\text{num}}(I) = \{ z \in \mathcal{Z}^{h,I} \mid \neg (1.7.23) \} \);
11: while \( Z_{\text{num}}(I) \neq \emptyset \) do
12: Choose \( z \in Z_{\text{num}}(I) \) and remove \( z \) from \( Z_{\text{num}}(I) \);
13: Solve \( \text{(PSP}(I, z, r)) \) (\( \text{RPSP}(I, z, r) \));
14: Set \( U_{\text{opt},w}^\text{num}(I) \leftarrow U_{\text{opt},w}^\text{num}(I) \cup Q_{\text{opt},w}(z) \);
15: Set \( UT\overline{Z}^\text{num}(I) \leftarrow UT\overline{Z}^\text{num}(I) \cup \{ \bar{u}, \bar{t}, \bar{z} \} \mid (\bar{u}, \bar{t}) \text{ gl. sol. of } (\text{PSP}(I, z, r)) \} \);
16: Add solutions of PSPs with respect to redundant reference points (cf. Remark 1.7.56): Set \( UT\overline{Z}^\text{num}(I) \leftarrow UT\overline{Z}^\text{num}(I) \cup \{ (\bar{u}, \bar{t}, \bar{z}) \mid (\bar{u}, \bar{t}) \text{ gl. sol. of } (\text{PSP}(I, z, r)) \} \);
17: Remove redundant reference points (cf. Remark 1.7.56): Set \( Z_{\text{num}}(I) \leftarrow Z_{\text{num}}(I) \setminus \{ z = (\bar{r}^I - (\bar{J}^I(\bar{u}) - \bar{z})) \} \text{ for all } \bar{u} \in Q_{\text{opt},w}(z) \} \); end while
18: end for
19: end for
20: if \( \text{computeParetoFront } = \text{ true} \) then
21: Remove all \( u \in U_{\text{opt},w}^\text{num}(\{1, \ldots, k\}) \) with \( u \notin U_{\text{opt}} \) by a non-dominance test;
22: end if

Proof. We show this statement by induction over the size \( i \) of the set \( I \).

\( i = 2 \): It clearly holds \( UT\overline{Z}(K) = UT\overline{Z}^\text{num}(K) \) for all \( K \subset \{1, \ldots, k\} \) with \( |K| = 1 \). Therefore, for any \( I \subset \{1, \ldots, k\} \) with \( |I| = 2 \) and any \( z \in \mathcal{Z}^{h,I} \) we can conclude that (1.7.20) \( \iff \) (1.7.23). This shows \( Z(I) \cap \mathcal{Z}^{h,I} = Z_{\text{num}}(I) \).

\((2, \ldots, i - 1) \rightarrow i \text{ } (3 \leq i \leq k)\): Let \( I \subset \{1, \ldots, k\} \) with \( |I| = i \) be arbitrary and assume that \( Z(K) \cap \mathcal{Z}^{h,I} = Z_{\text{num}}(K) \) holds for all \( K \subset \{1, \ldots, k\} \) with \( |K| < i \). Again, let \( z \in \mathcal{Z}^{h,I} \) be arbitrary. If we can show that (1.7.20) \( \iff \) (1.7.23) also holds in this case, we are done.

(1.7.20) \( \Rightarrow \) (1.7.23): Assume that there is \( K \subsetneq I \) and \( (\bar{u}, \bar{t}, \bar{z}) \in UT\overline{Z}(K) \) such that \( z^K = \bar{z}^K \) and \( z^K \geq \bar{J}^\Lambda(K)(\bar{u}) - \bar{r}^\Lambda(K) \) hold. By definition of \( UT\overline{Z}(K) \), it holds, in particular, \( \bar{z} \in Z(K) \). By assumption, this implies \( \bar{z} \in Z_{\text{num}}(K) \), so that (1.7.23) is satisfied.
Remark 1.7.69. To conclude, let us briefly discuss the quality criteria of the numerical implementation that were introduced in Section 1.4.2.

(i) **Coverage:** In Theorem 1.7.28, we showed that the mapping \( r_{\text{opt}} |_{\mathcal{Z}^D_{\text{opt}}} \) is Lipschitz continuous. Therefore, the coverage of the Pareto front is directly connected to the coverage of the set of reference points. By Theorem 1.7.68, we know that all relevant reference points on the grid \( \mathcal{Z}_{h,I} \) are selected by the numerical algorithm. Therefore, we can conclude

\[
\max_{\bar{y} \in \mathcal{J}^{\text{opt}(w)}} \min_{\bar{y} \in \mathcal{J}^{\text{num}(w)}} \| \bar{y} - y \| = O(h).
\]

(ii) **Uniformity:** For reference points \( z^1, z^2 \in \mathcal{Z}^D_{\text{opt}} \) it holds \( r_{\text{opt}}(z^i) - z^i = \bar{t}^i r \) for \( i = 1, 2 \). In particular, we have \( r_{\text{opt}}(z^i) \in \{z^1 + t r \mid t \in \mathbb{R} \} \) for \( i = 1, 2 \). Therefore, we can conclude the estimate

\[
\| r_{\text{opt}}(z^1) - r_{\text{opt}}(z^2) \|_\infty \geq \min_{s,t \in \mathbb{R}} \| z^1 + tr - z^2 - sr \|_\infty,
\]

i.e., the distance between \( r_{\text{opt}}(z^1) \) and \( r_{\text{opt}}(z^2) \) is bounded from below by the distance of the rays with base point \( z^1 \) and \( z^2 \) and direction \( r \), respectively.

(iii) **Cardinality:** Even by fixing the shifting vector \( \tilde{d} \) and the grid size \( h \), the number of elements of the approximation cannot be determined a-priori to any computations. However, it can be bounded from above as soon as the nadir objective point \( y^{\text{nad}(w)} \) is known, see Lemma 1.7.66. Note that the grid size \( h \) can be increased but not decreased throughout the hierarchical procedure. To be more precise, if for some \( I \subset \{1, \ldots, k\} \) the solutions of the subproblem (MOP(K)) for all \( K \subset I \) are computed with grid size \( h \), the problem (MOP(I)) can be solved with grid size \( h, 2h, 3h, \ldots \). Therefore, the approximation can be made coarser but not finer during the algorithm.

Example 1.7.70. To illustrate the numerical implementation of the PSM, we look at an example from rocket injector design (cf. [BKR17, Section 6.2]) involving four cost functions \( \tilde{J}_1, \ldots, \tilde{J}_4 : [0,1]^4 \to \mathbb{R} \), which are all polynomials in four variables of at most degree three. For the exact definition of the cost functions, we refer to [BKR17, Section 6.2]. In our notation, it holds \( \mathcal{U} = \mathbb{R}^4 \) and \( \mathcal{U}_\text{ad} = [0,1]^4 \). In the following, we select the target direction \( r = (1,1,1,1)^T \in \mathbb{R}^4 \).

In a first step, we will describe the process of solving the subproblem \( \tilde{J}_2, \tilde{J}_3, \tilde{J}_4 \). Afterwards, we show projections of approximations the Pareto front of the entire problem \( \tilde{J}_1, \tilde{J}_2, \tilde{J}_3, \tilde{J}_4 \) onto three-dimensional spaces and compare them to the results obtained in [BKR17, Section 6.2].

After computing all individual global minimizers \( \bar{u}^1, \ldots, \bar{u}^4 \) of the cost functions by minimizing the cost functions \( \tilde{J}_1, \ldots, \tilde{J}_4 \) with the Matlab-function \texttt{fmincon}, the next step is to solve the subproblems with two cost functions. For computing the Pareto front
of the subproblem \((\hat{j}_2, \hat{j}_3, \hat{j}_4)\), we need to solve the subproblems \((\hat{j}_2, \hat{j}_3)\), \((\hat{j}_2, \hat{j}_4)\) and \((\hat{j}_3, \hat{j}_4)\). Given the subproblem \((\hat{j}_2, \hat{j}_3)\), i.e., \(I = \{2, 4\}\), the procedure is displayed in Figure 1.17 for the shifting vector \(d = 0.1 \cdot (1, 1, 1, 1)^T\) and the grid size \(h = 0.04\). First, we compute \(\mathcal{L}^{D^I(\bar{u}^2)}\) and \(\mathcal{L}^{D^I(\bar{u}^4)}\) and generate the reference points \(Z^{num}(I) = Z(I) \cap Z^{h^I}\), cf. Theorem 1.7.68 and Figure 1.17(a). As a solver for the occurring PSPs, we use the built-in Matlab-function \textit{fminicon}, which can solve min-max problems. However, since the MOP is non-convex and the solver is local, we have to deal with the problem of ensuring the computation of the global minimizers of the PSPs. To this end, we solve \((\text{RPSP}(I, z, r))\) twice for every reference point \(z \in Z^{num}(I)\): In the first iteration, we start at the reference point \(z_I^{(1)} \in Z^{num}(I)\) being closest to \(\mathcal{L}^{D^I(\bar{u}^2)}\). In particular, for solving \((\text{RPSP}(I, z_I^{(1)}, r))\) we use \(u_I^{(1)} = \bar{u}^2\) as initial value. Assume that we have solved the PSPs for the first \(n\) reference points. Denote by \(\bar{u}_I^{(1)}, \ldots, \bar{u}_I^{(n)}\) the minimizers. Then we set the initial value \(u_I^{(n+1)}\) for solving \((\text{RPSP}(I, z_I^{(n+1)}, r))\) to \(u_I^{(n+1)} = 2\bar{u}_I^{(n)} - \bar{u}_I^{(n-1)}\).

The resulting solutions can be seen in Figure 1.17(b). Due to the local structure of the optimizer, too many points in the top left part of the Pareto front are computed, before the solver jumps to the tip point of the lower right part. This jump occurs for the reference point just above the set of orange colored reference points. Since it holds \(\bar{J}(\bar{u}) - z \neq \hat{t}\) for the corresponding solution, line 17 of Algorithm 4 marks all orange points to have the same minimizer, so that these reference points are redundant and the corresponding PSPs do not need to be solved. For the lower right part of the Pareto front, we can again observe the local structure of the solver. By comparing it to the Pareto front in Figure 1.17(d), it can be seen that the first few points coincide, before the points in Figure 1.17(b) start to deviate from the ones in Figure 1.17(d). Again, this is due to the fact that the solver can only compute local minimizers, so that it cannot detect the global minimizers of the corresponding PSPs.

The first reference point \(z_I^{(1)}\) in the second iteration is the one closest to \(\mathcal{L}^{D^I(\bar{u}^4)}\). From this point on, we proceed towards the top left of the Pareto front in the same way as in the first iteration. In particular, the initial value for the optimization of every PSP is chosen differently from the first iteration. The approximations obtained by this iteration is shown in Figure 1.17(c). While the lower right part of the Pareto front is approximated nicely, we get stuck at the tip point, which is a local minimizer for all reference points colored in orange. Note that in this case, line 17 of Algorithm 4 does not remove any reference points, since the only marked reference points are the ones, for which we have already solved the corresponding PSP.

Having solved \((\text{RPSP}(I, z_I^{(n)}, r))\) for a given reference point \(z_I^{(n)} \in Z^{num}(I)\) for the second time with solution \(\bar{u}_I^{(n)}\), we save \(\bar{u}_I^{(n)}\) as ‘global’ minimizer of \((\text{RPSP}(I, z_I^{(n)}, r))\) if it holds

\[
\max_{i \in I} \bar{J}_i(\bar{u}_I^{(n)}) - (z_I^{(n)})_i \leq \max_{i \in I} \bar{J}_i(\bar{u}_I^{(n)}) - (z_I^{(n)})_i
\]

and select \(\bar{u}_I^{(n)}\) otherwise. The idea of this procedure is to increase the chance that the global minimizer is computed. Of course, this is only of heuristic nature and the use of a global optimizer would be advisable. However, we did not further proceed in this direction. Finally, by additionally applying a non-dominance test, we obtain the approximation of the Pareto front of the subproblem \((\hat{j}_2, \hat{j}_4)\) shown in Figure 1.17(d). The reference points
colored in orange in this figure indicate the redundant reference points in the sense that they would not have been needed for obtaining an approximation of the entire Pareto front. In fact, for each of these reference points it holds $\hat{J}(\bar{u}) - z \neq \bar{t}$, where $\bar{u}$ is the minimizer of the corresponding PSP. However, these points cannot be detected before the computations, since they are contained in the gap between the sufficient and necessary conditions for computing reference points, cf. Corollary 1.7.53.

To get a finer approximation of the Pareto fronts in the following figures, we use a grid size of $h = 0.02$ from now on while the shifting vector $\vec{d} = 0.1 \cdot (1, 1, 1, 1)^T$ remains the same. The Pareto fronts to all subproblems of $(\hat{J}_2, \hat{J}_3, \hat{J}_4)$ with two cost functions can be seen in Figure 1.18(g)–(i). Note that all of the Pareto fronts have non-convex features, i.e., they are all disconnected or contain ‘non-convexly shaped’ parts. As a comparison, we also show the approximations of these Pareto fronts obtained by the WSM and the ERPM using more than 1000 approximation points, see Figures 1.18(a) – (c) for the WSM and Figures 1.18(d) – (f) for the ERPM. As we can see, the WSM fails to approximate

Figure 1.17: Solving the subproblem $(\hat{J}_2, \hat{J}_4)$ by the PSM.
any of the Pareto fronts appropriately. In contrast to that, it is still possible to obtain a good approximation of large parts the Pareto fronts by the ERPM. However, it is also visible that some small parts are missing for the subproblems \((\hat{J}_2, \hat{J}_3)\) and \((\hat{J}_2, \hat{J}_4)\) due to the non-convex shape of the Pareto front in these parts. In Section 2.2.4, we will discuss this in more detail, see also Figure 2.7.

Now let us continue with the PSM again. Having solved the subproblems with two cost functions, we continue with generating the reference points \(Z^{num}(I)\) for the subproblem \((\hat{J}_2, \hat{J}_3, \hat{J}_4)\), cf. Figure 1.19. In the Figures 1.19(a),(c), we show the reference points that we would obtain by using the necessary condition from (1.7.16), which is based on (1.7.14). In this approach, the sets \(L^{D^I}(\hat{u}^{K_{opt}})\) are computed for all \(K \subseteq I\). Based on these, the reference points for which (1.7.14) holds are removed. As we can see, there is a large gap in the plot of \(L^{D^I}(\hat{u}^{(2,4)}_{opt})\), which is due to the jump in the Pareto front of the subproblem
Data from subproblems without redundant reference points.

Data from subproblems with redundant reference points.

Reference points generated by (1.7.14).

Reference points generated by (1.7.23).

Figure 1.19: Difference in the generation of reference points for the subproblem \((\hat{J}_2, \hat{J}_3, \hat{J}_4)\) for the PSM: Using (1.7.14) (left) and using (1.7.23) (right).

\((\hat{J}_2, \hat{J}_3)\) and corresponds to the redundant reference points from Figure 1.18(a). This results in many reference points not being removed, although they are not needed for computing the Pareto front of \((\hat{J}_2, \hat{J}_3, \hat{J}_4)\). In contrast to this, we see the advantage of using the condition (1.7.23) in a numerical implementation. In here, all reference points, for which the corresponding PSP has been previously solved, are taken into consideration by using Theorem 1.7.47. This results in the additional red colored points in Figure 1.19(b). Consequently, much more reference points are removed compared to the first approach using condition (1.7.14).

By solving the PSPs corresponding to these reference points, we finally obtain the Pareto front of the problem \((\hat{J}_2, \hat{J}_3, \hat{J}_4)\) in Figure 1.20(c). In this case, it is not as straightforward to choose an initial value for every PSP as it was for two cost functions. Rather, at the \((n+1)\)-st PSP (R PSP\((I, z_I^{(n+1)}, r))\), we look at the previously solved \(n\) optimization
Figure 1.20: Approximations of the Pareto front of the subproblem \((\hat{J}_2, \hat{J}_3, \hat{J}_4)\) obtained by the WSM, the ERPM and the PSM.

problems with minimizers \(\bar{u}_{I}^{(1)}, \ldots, \bar{u}_{I}^{(n)}\). Let the index \(j\) be determined by

\[ j = \arg \min_{i \in \{1, \ldots, n\}} \left( \max_{l \in I} \hat{J}_l(\bar{u}_{I}^{(i)}) - (z_{I}^{(n+1)})_l \right). \]

Then we set \(u_{0}^{(n+1)} = \bar{u}_{I}^{(j)}\). Note that we solve again the PSP to every reference point twice by using different initial values. As for the case of two cost functions, this is done by iterating differently through the set of reference points. The resulting approximation of the Pareto front covers the entire Pareto front and appears very uniform. For this example, around 1400 Pareto optimal points have been computed, for which around 2800 PSPs were solved in total. By looking at Table 1.3, we see that the values for the coverage and the uniformity are 0.038 and 1.64, respectively, which confirms the visual impressions. Additionally, we infer from Table 1.3 that the coverage is of the order of the grid size \(h\). In fact, the coverage values, which are computed w.r.t. the \(\| \cdot \|_2\)-norm, is in line with the fact that the mapping \(R_{\text{opt}} |_{Z_{\text{opt}}} \) is Lipschitz continuous with
Lipschitz constant $L_{R,2} = 2\sqrt{3}$ in this case, cf. Remark 1.7.29. However, we observe that there is a problem with finding the global minimizer of the PSP for some points, which leads to some small inaccuracies in the approximation. In fact, by looking at Figure 1.20(d), which shows the reference Pareto front that we used for computing the coverage values, we see that some parts close to the boundary of the top right part of the Pareto front are better approximated than in Figure 1.20(c). Despite the complicated non-convex shape of the Pareto front, the uniformity values are all in a range of $1.5 - 2.2$, which indicates a uniform approximation of the Pareto front. They might be even better if we were able to compute the exact set of reference points and not only a superset of it.

Again, we compare these results to those obtained by the WSM and the ERPM. In the Figures 1.20(a), (b), the resulting Pareto fronts are shown. As for the Pareto fronts of the subproblems with two cost functions, the WSM is incapable of approximating the majority of the Pareto front. In contrast to this, the ERPM performs well, since there are only small parts missing. This is confirmed by the coverage value of 0.0712 for the grid size of $h = 0.025$, cf. Table 1.4. For a quite large grid size, the coverage is comparable to that of the PSM. Only when the grid size gets smaller, we see that the missing parts of the Pareto front lead to the coverage value stagnating around a value of 0.05. At the same time, the uniformity value deteriorates, since the computed Pareto optimal points cluster at the boundary, which is already visible in Figure 1.20(b).

Finally, we verify our results for the PSM by comparing them to the ones reported in [BKR17, Section 6.2]. To this end, we compute the Pareto front of the entire problem $(\hat{J}_1, \ldots, \hat{J}_4)$ by Algorithm 4 and project it onto the $\hat{J}_1\hat{J}_3\hat{J}_2$- and $\hat{J}_1\hat{J}_3\hat{J}_2$-space, respectively, cf. Figure 1.21. We tried to choose approximately the same angle as in [BKR17, Figure 7]. One can clearly see that the shapes of the projected Pareto fronts coincide with each other. Moreover, the coverage of the Pareto front produced by Algorithm 4 seems to be better in comparison to the results in [BKR17, Figure 7], especially in the regions close to the boundary of the Pareto front. Note that we computed a total of around 30,000 Pareto optimal points, whereas around 70,000 Pareto optimal points were computed in [BKR17].

**Table 1.3:** Coverage, uniformity and cardinality for the approximation of the Pareto front of the subproblem $(\hat{J}_2, \hat{J}_3, \hat{J}_4)$ obtained by the PSM for different grid sizes $h$.

<table>
<thead>
<tr>
<th>$h$</th>
<th>Coverage</th>
<th>Uniformity</th>
<th>Cardinality</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.08</td>
<td>0.126</td>
<td>2.21</td>
<td>132</td>
</tr>
<tr>
<td>0.04</td>
<td>0.068</td>
<td>1.88</td>
<td>413</td>
</tr>
<tr>
<td>0.02</td>
<td>0.038</td>
<td>1.64</td>
<td>1394</td>
</tr>
<tr>
<td>0.01</td>
<td>0.0213</td>
<td>1.56</td>
<td>5167</td>
</tr>
<tr>
<td>0.005</td>
<td>0.0111</td>
<td>1.51</td>
<td>19599</td>
</tr>
</tbody>
</table>

**Table 1.4:** Coverage, uniformity and cardinality for the approximation of the Pareto front of the subproblem $(\hat{J}_2, \hat{J}_3, \hat{J}_4)$ obtained by the ERPM for different grid sizes $h$.

<table>
<thead>
<tr>
<th>$h$</th>
<th>Coverage</th>
<th>Uniformity</th>
<th>Cardinality</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.1</td>
<td>0.1223</td>
<td>3.72</td>
<td>135</td>
</tr>
<tr>
<td>0.05</td>
<td>0.0895</td>
<td>3.81</td>
<td>453</td>
</tr>
<tr>
<td>0.025</td>
<td>0.0712</td>
<td>4.54</td>
<td>1618</td>
</tr>
<tr>
<td>0.0125</td>
<td>0.0554</td>
<td>6.05</td>
<td>6052</td>
</tr>
<tr>
<td>0.00625</td>
<td>0.0503</td>
<td>11.49</td>
<td>23302</td>
</tr>
</tbody>
</table>
This suggests that Algorithm 4 also produces a more uniform approximation of the Pareto front.
Chapter 2

Multiobjective PDE-Constrained Optimization

Being a special case of constrained optimization, PDE-constrained optimization arises in many applications in, e.g., physics, industry, economy or medicine, where the underlying processes can be modeled by a PDE. Its special structure allows for a natural splitting of the optimization variable in a parameter or control $u$ and a state $y$. In most cases, the variable $u$ is seen as an input parameter of the PDE, whereas the state $y$ denotes the solution of this PDE. The general form of a PDE-constrained optimization problem with additional control constraints is

$$\min_{(y,u) \in Y \times U} J(y, u) \quad \text{s.t. } e(y, u) = 0, \ u \in U_{ad},$$

(PDE-C-Opt)

where $Y$ is the state space, $U$ is the control space and $J: Y \times U \to \mathbb{R}$ is the cost function. The equality constraint $e(y, u) = 0$ with $e: Y \times U \to Z$ for some Banach space $Z$ encodes the PDE in a functional analytical setting. Finally, the constraints on the control $u$ are expressed via the admissible set $U_{ad}$.

In this chapter, we present two PDE-constrained optimization problems, which will then be used for numerically testing the three scalarization methods from Chapter 1.

In Section 2.1, we introduce a multiobjective optimal control problem (MOCP) of a linear parabolic advection-diffusion equation. We start by analyzing the underlying state equation, before a general version of the cost functions of the problem is presented. For these, it is shown how gradient and Hessian information can be obtained by solving the so-called adjoint equation. Lastly, this is transformed to the multiobjective setting and numerical experiments are conducted for a problem with three cost functions.

In Section 2.2, a non-convex multiobjective parameter optimization problem (MPOP) of an elliptic diffusion-reaction equation is presented. Here the parameters are given by the diffusion and the reaction parameter. Again, after analyzing the state equation, general cost functions are introduced, for which it is shown, on the one hand, that the problem has a global solution and, on the other hand, how gradient and Hessian information can be computed. To this end, the adjoint equation as well as derivatives of the state and adjoint equations are needed. In the last part of this section, we conduct numerical experiments on a problem with three cost functions.
2. A Linear-Quadratic Multiobjective Optimal Control Problem

Our first model problem is a linear-quadratic MOCP with bilateral control constraints, in which the governing PDE is a linear parabolic advection-diffusion equation. This problem can be interpreted as the energy efficient heating, ventilation and air conditioning (HVAC) operation of a room (cf. [FHC06]). Linear-quadratic optimal control is a popular research field and structurally similar problems are studied in, e.g., [GV17, Trö10] in the context of scalar optimal control and in, e.g., [Ban17, BBV17] in the context of multiobjective optimal control.

2.1.1 The State Equation

Let $\Omega \subset \mathbb{R}^d$ for $d \in \{2, 3\}$ be a bounded domain with Lipschitz continuous boundary $\Gamma = \partial \Omega$, which is supposed to be split into disjoint subsets $\Gamma_1, \ldots, \Gamma_r$. Furthermore, let $\Omega_1, \ldots, \Omega_m$ be non-empty, pairwise disjoint subsets of the domain $\Omega$. Let $T > 0$ be a fixed final time. Then we set $Q := (0, T) \times \Omega$, $\Sigma := (0, T) \times \Gamma$ and $\Sigma_i := (0, T) \times \Gamma_i$ for all $i \in \{1, \ldots, r\}$.

Our state equation is given by the following advection-diffusion equation with Robin boundary condition:

$$
y_t(t, x) - \kappa \Delta y(t, x) + b(t, x) \cdot \nabla y(t, x) = \sum_{i=1}^{m} u_i(t) \chi_i(x) \quad \text{a.e. in } Q,$$

$$\kappa \frac{\partial y}{\partial n}(t, s) + \alpha_i y(t, s) = \alpha_i y_a(t, s) \quad \text{a.e. on } \Sigma_i, \quad (2.1.1)$$

$$y(0, x) = y_0(x) \quad \text{a.e. in } \Omega.$$

The constant $\kappa > 0$ is the diffusion coefficient and the time-dependent advection field is given by $b \in L^\infty(Q; \mathbb{R}^d)$. The functions $\chi_1, \ldots, \chi_m \in L^2(\Omega)$ are given by the characteristic functions of the sets $\Omega_1, \ldots, \Omega_m$. For the control variable $u = (u_1, \ldots, u_m)$ we suppose $u \in \mathcal{U} := L^2(0, T; \mathbb{R}^m)$. In the Robin boundary condition, the constants $\alpha_1, \ldots, \alpha_r \geq 0$ model the heat flux between the walls $\Gamma_1, \ldots, \Gamma_r$ and the outside world with temperature $y_a \in L^2(\Sigma)$. Lastly, $y_0 \in L^2(\Omega)$ is a given initial temperature distribution.

Let us now introduce the appropriate framework for treating the state equation analytically. We define $H := L^2(\Omega)$ and $V := H^1(\Omega)$ and endow them with their canonical inner products

$$\langle \varphi, \psi \rangle_H := \int_{\Omega} \varphi(x)\psi(x) \, dx \quad (\varphi, \psi \in H),$$

$$\langle \varphi, \psi \rangle_V := \int_{\Omega} \varphi(x)\psi(x) + \nabla \varphi(x) \cdot \nabla \psi(x) \, dx \quad (\varphi, \psi \in V),$$

and the induced norms $\|\cdot\|_H := \sqrt{\langle \cdot, \cdot \rangle_H}$ and $\|\cdot\|_V := \sqrt{\langle \cdot, \cdot \rangle_V}$, respectively. Then $(H, \|\cdot\|_H)$ and $(V, \|\cdot\|_V)$ are Hilbert spaces and the triple

$$V \hookrightarrow H \hookrightarrow V',$$
where \( V' \) is the dual of \( V \), is a Gelfand triple, i.e., both the canonical embedding \( V \hookrightarrow H \) and the embedding \( H \hookrightarrow V' \), \( h \mapsto \langle h, \cdot \rangle_H \) are continuous and dense (cf. [Wlo87, Definition 17.1 and Theorem 17.4]). Moreover, the embedding \( V \hookrightarrow H \) is compact (cf. [Wlo87, Theorem 7.2]). For the space \( V \), it is possible to define the linear and continuous trace operator \( T: V \to L^2(\partial \Omega) \), which is defined by

\[
T \varphi := \varphi|_{\partial \Omega} \quad \text{for any } \varphi \in C(\overline{\Omega}) \cap V,
\]

and can be uniquely extended to \( V \) (cf. [Wlo87, Theorem 8.7]). In this sense, it is possible to ’evaluate’ any \( \varphi \in V \) at the boundary \( \partial \Omega \). Note that we will identify the space \( L^2(Q) \) with \( L^2(0, T; H) \) in the following. The space

\[
W(0, T) := \{ \varphi \in L^2(0, T; V) \mid \varphi_t \in L^2(0, T; V') \}
\]

endowed with the inner product

\[
\langle \varphi, \psi \rangle_{W(0, T)} := \int_0^T \langle \varphi(t), \psi(t) \rangle_V + \langle \varphi_t(t), \psi_t(t) \rangle_{V'} \, dt
\]

and the induced norm \( ||\cdot||_{W(0, T)} := \sqrt{\langle \cdot, \cdot \rangle_{W(0, T)}} \) is a Hilbert space (cf. [Wlo87, Theorem 25.4]), which is compactly embedded into the space \( L^2(0, T; H) \) by the Aubin-Lions Lemma (cf. [Lio69, Théorème 1]) and continuously embedded into \( C([0, T]; H) \) (cf. [Wlo87, Theorem 25.5]). In particular, it is meaningful to evaluate a function \( y \in W(0, T) \) at any time point \( t \in [0, T] \).

Based on the results from above, we define the following linear operators.

**Definition 2.1.1.** We define the canonical embedding

\[
\Phi_Q: W(0, T) \hookrightarrow H, \quad \Phi_Q y := \varphi,
\]

and the linear and continuous operator

\[
\Phi_T: W(0, T) \to H, \quad \Phi_T y := y(T),
\]

which evaluates any function \( y \in W(0, T) \) at the end-time point \( T \).

Using standard arguments, we can derive the weak formulation of the state equation (2.1.1): Find \( y \in W(0, T) \) such that

\[
\langle y_t(t), \varphi \rangle_{V', V} + a(t; y(t), \varphi) = \langle F(t) + B(u(t)), \varphi \rangle_{V', V} \quad \text{f.a.e. in } (0, T),
\]

\[
y(0) = y_0 \quad \text{in } H.
\]

(2.1.2)

In this weak formulation, \( a(t; \cdot, \cdot): V \times V \to \mathbb{R} \) is a (non-symmetric) bilinear form for almost all \( t \in (0, T) \), which is defined by

\[
a(t; \varphi, \psi) := \kappa \int_{\Omega} \nabla \varphi(x) \cdot \nabla \psi(x) \, dx + \int_{\Omega} b(t, x) \cdot \nabla \varphi(x) \psi(x) \, dx
\]

\[
+ \sum_{i=1}^r \alpha_i \int_{\Gamma_i} \varphi(s) \psi(s) \, ds
\]
for all $\varphi, \psi \in V$. The time-dependent linear functional $F(t) \in V'$ is given by

$$\langle F(t), \varphi \rangle_{V', V} := \sum_{i=1}^{r} \alpha_i \int_{\Gamma_i} y_a(t, s) \varphi(s) \, ds$$

for all $\varphi \in V$ and almost all $t \in (0, T)$. Finally, the linear functional $B: \mathbb{R}^m \to V'$ is defined by

$$\langle Bu, \varphi \rangle_{V', V} := \sum_{i=1}^{m} u_i \int_{\Omega} \chi_i(x) \varphi(x) \, dx$$

for all $u \in \mathbb{R}^m$ and all $\varphi \in V$.

**Lemma 2.1.2.**  (i) For almost all $t \in (0, T)$ it holds

$$|a(t; \varphi, \psi)| \leq \alpha \|\varphi\|_V \|\psi\|_V$$

for all $\varphi, \psi \in V$,

$$a(t; \varphi, \varphi) \geq \alpha_1 \|\varphi\|_V^2 - \alpha_2 \|\varphi\|_H^2$$

for all $\varphi \in V$ for some time-independent constants $\alpha, \alpha_1 > 0$ and $\alpha_2 \geq 0$.

(ii) It holds $F \in L^2(0, T; V')$.

(iii) For all $u \in U$ it holds $(t \mapsto B(u(t))) \in L^2(0, T; V')$.

**Proof.** For very similar operators, the statements were shown, e.g., in [Mec19, Lemma 1.4].

**Remark 2.1.3.** In the following, we will sometimes write $Bu$ for a given $u \in U$. In this case, we see this expression as the function $(t \mapsto B(u(t))) \in L^2(0, T; V')$.

Now we can show the well-posedness of the weak formulation (2.1.2).

**Theorem 2.1.4.** For any $u \in U$, there is a unique solution $y = y(u) \in W(0, T)$ of (2.1.2). Moreover, there is a constant $C > 0$, which is independent of $u$, $F$ and $y_0$, such that the estimate

$$\|y\|_{W(0, T)} \leq C \left( \|u\|_U + \|F\|_{L^2(0, T; V')} + \|y_0\|_H \right)$$

(2.1.3)

is satisfied.

**Proof.** The proof of the unique solvability of (2.1.2) is based on the results of Lemma 2.1.2 and can for example be found in [DL00, p. 512–520]. The a-priori estimate can be shown similarly to [Trö10, Theorem 3.13].

In the following, we want to decompose the solution $y$ of (2.1.2) into a part that is independent of the control $u$, and a part that only depends on $u$, but is independent of $F$ and $y_0$. 

Definition 2.1.5. By $\hat{y} \in W(0, T)$ we denote the unique solution of (2.1.2) for $u = 0$. Moreover, we define the solution operator

$$S \colon \mathcal{U} \to W(0, T)$$

of the equation

$$\langle y_t(t), \varphi \rangle_{V'} + a(t; y(t), \varphi) = \langle B(u(t)), \varphi \rangle_{V'} \quad f.a. \quad \varphi \in V \text{ a.e. in } (0, T),$$

$$y(0) = 0 \quad \text{in } H,$$

which maps any control $u \in \mathcal{U}$ to the unique solution $y = S(u)$ of (2.1.4).

Remark 2.1.6. By the linearity of (2.1.2), we can conclude that $y = \hat{y} + Su$ is the unique solution of (2.1.2) for any $u \in \mathcal{U}$. ♦

Definition 2.1.7. Based on the solution operator $S$, we define the two linear operators

$$S_Q \colon \mathcal{U} \to L^2(0, T; H), \quad S_Q u := \Phi_Q Su,$$

$$S_T \colon \mathcal{U} \to H, \quad S_T u := \Phi_T Su,$$

where the operators $\Phi_Q$ and $\Phi_T$ have been defined in Definition 2.1.1.

Lemma 2.1.8. The operators $S$, $S_Q$ and $S_T$ are linear and continuous.

Proof. The linearity of $S$ follows directly from the linearity of the equation (2.1.2). Its continuity can be concluded from the a-priori estimate (2.1.3). Now the linearity and continuity of $S_Q$ and $S_T$ follow from the fact that they are concatenations of linear and continuous operators. □

2.1.2 The Optimal Control Problem

Let us now come back to the general PDE-constrained optimization problem (PDE-C-Opt). In view of the analysis of the state equation in the previous section, we define the state space $Y := W(0, T)$. Then for our model problem we choose the quadratic cost function $J : Y \times \mathcal{U} \to \mathbb{R}$ defined by

$$J(y, u) := \frac{\sigma_Q}{2} \|\Phi_Q y - y_Q\|^2_{L^2(0, T; H)} + \frac{\sigma_T}{2} \|\Phi_T y - y_T\|^2_H + \frac{\sigma_u}{2} \|u\|^2_{L^2(0, T; \mathbb{R}^m)}$$

for non-negative parameters $\sigma_Q, \sigma_T, \sigma_u \geq 0$, where at least one parameter is strictly positive. The functions $y_Q \in L^2(0, T; H)$ and $y_T \in H$ are a given desired state and target, respectively. Now the optimal control problem with bilateral control constraints reads

$$\min_{(y, u) \in Y \times \mathcal{U}} J(y, u)$$

s.t. $y = y(u)$ solves (2.1.2) with control input $u$,

$$u \in \mathcal{U}_{\text{ad}} := \{u \in \mathcal{U} \mid u_a(t) \leq u(t) \leq u_b(t) \text{ a.e. in } (0, T)\}.$$

Here, $u_a, u_b : (0, T) \to \mathbb{R}^m \cup \{-\infty, \infty\}$ with $u_a(t) \leq u_b(t)$ for all $t \in (0, T)$ are given lower and upper bounds on the control $u$, which are allowed to take the values $-\infty$ and $\infty$, respectively.
**Definition 2.1.9.** By Theorem 2.1.4 and Remark 2.1.6, there is a unique solution \( y = y(u) \) of (2.1.2) for any \( u \in U \), which is given by \( y = S(u) + \hat{y} \). Therefore, by setting \( \hat{y}_Q := y_Q - \Phi_Q \hat{y} \) and \( \hat{y}_T := y_T - \Phi_T \hat{y} \), we can define the essential cost function \( \hat{J} : U \rightarrow \mathbb{R} \) by

\[
\hat{J}(u) := \frac{\sigma_Q}{2} \| S_Q u - \hat{y}_Q \|_{L^2(0,T;H)}^2 + \frac{\sigma_T}{2} \| S_T u - \hat{y}_T \|_H^2 + \frac{\sigma_U}{2} \| u \|_{L^2(0,T;\mathbb{R}^m)}^2.
\]

This allows us to introduce the reduced optimal control problem

\[
\min_{u \in U} \hat{J}(u) \quad \text{s.t.} \quad u \in U_{ad}.
\]  

(2.1.6)

Under some assumption on the parameter \( \sigma_U \), we can show that the cost function \( \hat{J} \) fits into the framework of convex multiobjective optimization, cf. Assumption 1.1.

**Assumption 2.1.** If \( U_{ad} \) is unbounded, assume that \( \sigma_U > 0 \) holds.

**Lemma 2.1.10.** If Assumption 2.1 holds, then the function \( \hat{J} \) is convex, continuous (and thus, in particular, lower semi-continuous) and bounded from below. Moreover, if \( U_{ad} \) is unbounded, it holds \( \lim_{\| u \| \to \infty} \hat{J}(u) = \infty \). Thus, \( \hat{J} \) fulfills Assumption 1.1.

**Proof.** As a concatenation of a quadratic function and a linear operator, every summand of the function \( \hat{J} \) is convex. Therefore, \( \hat{J} \) is convex as well. The continuity of \( \hat{J} \) follows directly from the continuity of the operators \( S_Q \) and \( S_T \), which was shown in Lemma 2.1.8. Moreover, \( \hat{J} \) is clearly bounded from below by 0. Finally, in the case that \( U_{ad} \) is unbounded, we have \( \sigma_U > 0 \) by Assumption 2.1. Hence, \( \lim_{\| u \| \to \infty} \hat{J}(u) = \infty \) follows immediately.

**Corollary 2.1.11.** Let Assumption 2.1 be satisfied. Then there is a global solution \( \bar{u} \in U_{ad} \) of (2.1.6). If \( \sigma_U > 0 \) holds, the solution is unique.

**Proof.** In Lemma 2.1.10, we showed that \( \hat{J} \) fulfills Assumption 1.1. Therefore, the existence of a global solution of (2.1.6) follows from Theorem 1.5.6, where we showed that (WSP(\( \alpha \))) has a solution for any \( \alpha \in \Delta_k \). Here, it holds \( k = 1 \) and thus for \( \alpha = 1 \) the problem (WSP(\( \alpha \))) is equivalent to (2.1.6).

If we assume additionally \( \sigma_U > 0 \), the cost function \( \hat{J} \) is strictly convex, which directly implies the uniqueness of the solution.

**2.1.2.1 Computation of Derivatives**

In this section, we derive representations of the gradient and the Hessian of the essential cost function \( \hat{J} \), which are numerically evaluable. Let us start with deriving the 'standard' representation of the gradient and the Hessian, respectively.
Lemma 2.1.12. The essential cost functional $\hat{J}$ is quadratic. For any $u \in U$ its gradient and Hessian are given by

\[
\nabla \hat{J}(u) = \sigma_Q S_Q^* (S_Q u - \tilde{y}_Q) + \sigma_T S_T^* (S_T u - \tilde{y}_T) + \sigma_{\mathbb{U}} u \quad \in \mathbb{U},
\]
\[
\nabla^2 \hat{J}(u) = \sigma_Q S_Q^* S_Q + \sigma_T S_T^* S_T + \sigma_{\mathbb{U}} \text{id}_U \quad \in L(U, U').
\]

Proof. This follows directly by applying the chain rule and the definition of the adjoint operator.

Now the question is how the adjoint operators $S_Q^*$ and $S_T^*$ can be evaluated in practice. It turns out that this is possible by introducing the so-called adjoint equation.

Definition 2.1.13. Let $z_Q \in L^2(0, T; H)$ and $z_T \in H$ be arbitrary. Then the equation

\[
- \langle p_t(t), \varphi \rangle_{V', V} + a(t; \varphi, p(t)) = \langle z_Q(t), \varphi \rangle_H \quad f.a. \ \varphi \in V \ a.e. \ in \ (0, T),
\]
\[
p(T) = z_T \quad in \ H
\]

is called the adjoint equation of (2.1.2).

Theorem 2.1.14. For any $z_Q \in L^2(0, T; H)$ and $z_T \in H$, the adjoint equation (2.1.7) has a unique solution $p = p(z_Q, z_T) \in W(0, T)$. Moreover, there is a constant $C > 0$, which is independent of $z_Q$ and $z_T$, such that the estimate

\[
\|p\|_{W(0, T)} \leq C \left( \|z_Q\|_{L^2(0, T; H)} + \|z_T\|_H \right)
\]

is satisfied.

Proof. This can be concluded with the same arguments as the proof of Theorem 2.1.4.

The reason to introduce the adjoint equation is the next result, which tells us that it can be used to evaluate the adjoint operators $S_Q^*$ and $S_T^*$, respectively.

Lemma 2.1.15. Let $z_Q \in L^2(0, T; H)$ and $z_T \in H$ be arbitrary. Let

\[
p(z_Q, z_T), p(z_Q, 0), p(0, z_T) \in W(0, T)
\]

be the unique solutions of (2.1.7) for the tupels

\[
(z_Q, z_T), (z_Q, 0), (0, z_T) \in L^2(0, T; H) \times H,
\]

respectively. Then it holds

\[
B^* p(z_Q, 0) = S_Q^* z_Q,
\]
\[
B^* p(0, z_T) = S_T^* z_T.
\]

Due to the linearity of the adjoint equation (2.1.7) and the adjoint operator $B^*$, we have

\[
B^* p(z_Q, z_T) = S_Q^* z_Q + S_T^* z_T.
\]

Proof. This statement is a special case of [Bee19, Lemma 2.10].
Definition 2.1.16. Denote by \( \hat{p} \in W(0,T) \) the unique solution of
\[
-\langle \hat{p}_t(t), \varphi \rangle_{V',V} + a(t; \varphi, \hat{p}(t)) = -\sigma_Q \langle \tilde{y}_Q(t), \varphi \rangle_H \quad \text{f.a. } \varphi \in V \text{ a.e. in } (0,T),
\]
\[
\hat{p}(T) = -\sigma_T \tilde{y}_T \quad \text{in } H.
\]
(2.1.9)

Furthermore, define the linear solution operator \( A : U \rightarrow W(0,T) \), which maps any control \( u \in U \) to the unique solution \( p = Au \) of
\[
-\langle p_t(t), \varphi \rangle_{V',V} + a(t; \varphi, p(t)) = \sigma_Q \langle (S_Q u)(t), \varphi \rangle_H \quad \text{f.a. } \varphi \in V \text{ a.e. in } (0,T),
\]
\[
p(T) = \sigma_T S_T u \quad \text{in } H.
\]
(2.1.10)

Corollary 2.1.17. The solution operator \( A \) is continuous.

Proof. This follows directly from the continuity of the operators \( S_Q \) and \( S_T \), which was shown in Lemma 2.1.8, and the a-priori estimate (2.1.8).

Remark 2.1.18. Due to the linearity of the adjoint equation, the function \( p = Au + \hat{p} \) is the unique solution of
\[
-\langle p_t(t), \varphi \rangle_{V',V} + a(t; \varphi, p(t)) = \sigma_Q \langle (S_Q u)(t) - \tilde{y}_Q, \varphi \rangle_H \quad \text{f.a. } \varphi \in V \text{ a.e. in } (0,T),
\]
\[
p(T) = \sigma_T (S_T u - \tilde{y}_T) \quad \text{in } H
\]
(2.1.11)
for any \( u \in U \).

Now we can finally show the representation of the gradient and the Hessian of \( \hat{J} \) via the adjoint equation.

Corollary 2.1.19. Let \( u \in U \) be arbitrary. Then it holds
\[
\nabla \hat{J}(u) = B^*(Au + \hat{p}) + \sigma_u u \quad \in U,
\]
\[
\nabla^2 \hat{J}(u) = B^* A + \sigma \text{id}_U \quad \in L(U,U').
\]

Proof. This follows directly from Lemma 2.1.12, Lemma 2.1.15 and Definition 2.1.16.

Remark 2.1.20. In our case, the adjoint operator \( B^* : V \rightarrow \mathbb{R}^m \) is given by
\[
B^* \varphi = \begin{pmatrix} \int_{\Omega} \chi_1(x)\varphi(x) \, dx \\ \vdots \\ \int_{\Omega} \chi_m(x)\varphi(x) \, dx \end{pmatrix}
\]
for any \( \varphi \in V \).
2.1.3 The Multiobjective Optimal Control Problem

Based on the linear-quadratic optimal control problem that was introduced and investigated in the previous section, we fix the notation for seeing the problem in a multiobjective context. To this end, let $k \in \mathbb{N}$ and the parameters
\[
\sigma_Q^{(1)}, \ldots, \sigma_Q^{(k)} \geq 0, \quad \sigma_T^{(1)}, \ldots, \sigma_T^{(k)} \geq 0, \quad \sigma_U^{(1)}, \ldots, \sigma_U^{(k)} \geq 0
\]
be given. Moreover, let
\[
y_Q^{(1)}, \ldots, y_Q^{(k)} \in L^2(0, T; H) \quad \text{and} \quad y_T^{(1)}, \ldots, y_T^{(k)} \in H
\]
be the desired states and targets, respectively. By defining $\tilde{y}_Q^{(i)} := y_Q^{(i)} - \Phi_Q \hat{y}$ and $\tilde{y}_T^{(i)} := y_T^{(i)} - \Phi_T \hat{y}$ for all $i \in \{1, \ldots, k\}$, we can introduce the multiobjective essential cost functions $\hat{J}_1, \ldots, \hat{J}_k : U \to \mathbb{R}$ by
\[
\hat{J}_i(u) := \frac{\sigma_Q^{(i)}}{2} \left\| S_Q u - \tilde{y}_Q^{(i)} \right\|_{L^2(0, T; H)}^2 + \frac{\sigma_T^{(i)}}{2} \left\| S_T u - \tilde{y}_T^{(i)} \right\|_H^2 + \frac{\sigma_U^{(i)}}{2} \| u \|_{L^2(0, T; \mathbb{R}^m)}^2
\]
for all $u \in U$ and any $i \in \{1, \ldots, k\}$.

Then the linear-quadratic MOCP reads
\[
\min_{u \in U_{ad}} \hat{J}(u) = \min_{u \in U_{ad}} \left( \begin{array}{c} \hat{J}_1(u) \\ \vdots \\ \hat{J}_k(u) \end{array} \right). \quad (\text{MOCP})
\]

For this problem, we can show that it fits into the framework of convex multiobjective optimization specified in Assumption 1.1.

Lemma 2.1.21. Assume that Assumption 2.1 holds for all cost functions $\hat{J}_1, \ldots, \hat{J}_k$. Then Assumption 1.1 is satisfied.

Proof. This is a direct consequence from Lemma 2.1.10. \hfill \square

2.1.4 Numerical Experiments

All numerical experiments in this thesis were performed on a Notebook Lenovo ThinkPad T460s with Intel Core i7-6600U CPU @ 2.60GHz and 12GB RAM. The codes are written in Matlab.

Let the domain $\Omega$ be given by the two-dimensional unit square, i.e., $\Omega = (0, 1)^2$. For the state equation (2.1.1), we choose the following data:

- The final time is $T = 1$.
- The diffusion parameter is given by $\kappa = 0.5$. 
For (MOCP), we define the three cost functions

\[ J_1(u) := \frac{1}{2} \| u \|_{L^2(0,T;\mathbb{R}^4)}^2, \]

\[ J_2(u) := \frac{1}{2} \left\| S_Q u - \tilde{y}_Q^{(2)} \right\|_{L^2(0,T;H)}^2 + \frac{0.002}{2} \| u \|_{L^2(0,T;\mathbb{R}^4)}^2, \]

\[ J_3(u) := \frac{1}{2} \left\| S_Q u - \tilde{y}_Q^{(3)} \right\|_{L^2(0,T;H)}^2 + \frac{0.001}{2} \| u \|_{L^2(0,T;\mathbb{R}^4)}^2, \]

where we set \( \tilde{y}_Q^{(2)} := y_Q^{(2)} - \Phi_Q \tilde{y} \) and \( \tilde{y}_Q^{(3)} := y_Q^{(3)} - \Phi_Q \tilde{y} \) with the desired temperatures

\[
\begin{align*}
    y_Q^{(2)}(t,x) := & \begin{cases} 
    14, & \text{if } t \in [0,0.3] \cup [0.7,1], \\
    16, & \text{if } t \in (0.3,0.7),
    \end{cases} \\
    y_Q^{(3)}(t,x) := & 16 + 2t.
\end{align*}
\]

Moreover, the lower and upper bounds for the control constraints are given by

\[ u_a(t) = (-5,-5,-5,-5)^T \quad \text{and} \quad u_b(t) = (5,5,5,5)^T \]

for almost all \( t \in (0,1) \), respectively.

For the numerical optimization, we use a Discretize-Before-Optimize (DBO) approach, cf. [BC05]. In this approach, all involved quantities are first discretized which results in a finite-dimensional optimization problem, for which then the representations of the gradient and the Hessian can be derived. In contrast to an Optimize-Before-Discretize (OBD) approach, in which the representations are first derived on an infinite-dimensional level and then discretized, this avoids numerical errors in the gradients, which would lead to an inexact optimization. It is important to note that the resulting discretizations of the DBO approach are not in line with the theoretical representations presented in the
2.1 A Linear-Quadratic MOCP

previous sections: Indeed, using an implicit Euler scheme for the state equation does not yield the same implicit Euler scheme for the adjoint equation when using a DBO approach. In the same way, also the obtained representation for computing the gradient in the DBO scheme deviates from the one, which would result from discretizing the representations from Corollary 2.1.19.

As optimization algorithm for the arising WSPs and ERPPs, we choose a projected Newton-CG method, see, e.g., [Kel99, Section 5.5.2]. As termination criteria, we choose the typical first-order condition

$$
\left\| u^{(i)} - P_{\text{ad}}(u^{(i)} - \nabla J(u^{(i)})) \right\|_{\mathcal{U}} \leq \tau_{\text{FOC}}
$$

with $\tau_{\text{FOC}} = 1e^{-7}$, where $P_{\text{ad}}: \mathcal{U} \rightarrow \mathcal{U}_{\text{ad}}$ is the projection onto the admissible set $\mathcal{U}_{\text{ad}}$, the relative reduction in the cost function

$$
\frac{J(u^{(i)}) - J(u^{(i-1)})}{J(u^{(i-1)})} \leq \tau_{\text{rel}}
$$
with $\tau_{\text{rel}} = 10^{-14}$, the minimal backtracking step size $\tau_{\text{arm, min}} = 10^{-8}$ of the Armijo condition and the maximal number of iterations $i_{\text{max}} = 50$ for the Newton-CG method.

For the PSM, we choose the target direction $r = (1, 1, 1)^T$. For solving the PSPs, we use an augmented Lagrangian penalty method which is based on [NW06, Chapter 17.4] and described in Appendix B. For the parameters used in Algorithm 11, we make the following choices:

- Initial penalty parameter $\mu^{(0)} = 10$ and increasing factor for the penalty parameter $\mu_{\text{incr}} = 10$.
- Initial tolerance for equality constraint $\tau^{(0)}_{\text{EC, sub}} = 1/\sqrt{10}$.
- Initial termination tolerance $\tau^{(0)}_{\text{FOC, sub}} = 0.1$ for the subproblem (B.1.3).
- Termination tolerances $\tau_{\text{EC}} = 10^{-8}$ and $\tau_{\text{FOC}} = 10^{-6}$ for the equality constraint and the first-order condition, respectively.

**Choice of initial value.** For solving the arising WSPs, ERPPs and PSPs by the respective numerical optimization methods, we perform a warm start by using solutions of the previously solved optimization problems. For two cost functions, there is a clear 'direction' when computing the Pareto front, either from top to bottom or from bottom to top. Consequently, a good warm start for the $(n+1)$-st optimization problem is given by

$$u_{n+1} = u^n + (\bar{u}^n - \bar{u}^{n-1}),$$

i.e., we use a first-order approximation of the Pareto set at $\bar{u}^n$. For three cost functions, both the weights and the reference points do not evolve in only one direction. Thus, we cannot use the same approach as for two cost functions. Rather, at the $(n+1)$-st optimization problem, we look at the previously solved $n$ optimization problems with optimal controls $\bar{u}_1, \ldots, \bar{u}_n$. Each of these is then plugged into the cost function of the $(n+1)$-st optimization problem. Let $j \in \{1, \ldots, n\}$ denote the index with the smallest function value. Then we set $u_{n+1} = \bar{u}_j$. For the PSM, we additionally set the initial Lagrange multiplier of the new problem to $\lambda_{n+1} = \lambda_j$.

**Discretization of $\Delta_k$.** For the WSM, we choose 25 uniformly distributed weights per dimension, i.e., we use the discrete sets of weights

$$\Delta_d^2 := \bigcup_{i=0}^{24} \left\{ \left( \frac{i}{24}, 1 - \frac{i}{24} \right) \right\} \quad \text{and} \quad \Delta_d^3 := \bigcup_{i=0}^{24} \bigcup_{j=0}^{24-i} \left\{ \left( \frac{i}{24}, \frac{j}{24}, 1 - \frac{i}{24} - \frac{j}{24} \right) \right\}.$$

**Choice of reference points for the ERPM and the PSM.** For the ERPM and the PSM, we choose the shifting vectors $\tilde{d}_{\text{ERPM}} = (1, 1, 1)^T$ and $\tilde{d}_{\text{PSM}} = 0.5 \cdot (1, 1, 1)^T$, as well as the grid sizes $h_{\text{ERPM}} = 0.3$ and $h_{\text{PSM}} = 0.2$ for the reference point grid, respectively. At this point, we omit an extensive discussion of different choices for these two parameters. In [Spu19, p. 70 ff.], the influence of the shifting vector $\tilde{d}$ both on the approximation fineness of the Pareto front and the convergence of the projected Newton-CG method is investigated for the ERPM.
2.1.4.1 Numerical results

According to the hierarchical computation of the Pareto front, at first the three individual cost functions $\hat{J}_1, \hat{J}_2, \hat{J}_3$ are minimized by using the projected Newton-CG method. Based on these minimizers, the reference points for solving the subproblems with two cost functions $(\hat{J}_1, \hat{J}_2)$, $(\hat{J}_1, \hat{J}_3)$ and $(\hat{J}_2, \hat{J}_3)$ by the ERPM and the PSM are generated by using the schemes from the Sections 1.6.4 and 1.7.4, respectively.

The resulting approximations of the Pareto front of the subproblem $(\hat{J}_1, \hat{J}_2)$ are depicted in Figure 2.2 for the WSM, the ERPM and the PSM. For the WSM, we obtain a non-uniform approximation of the Pareto front as it is described in Remark 1.5.15. In contrast to this, the approximation obtained by the ERPM looks more uniform except for the top left part of the Pareto front, where a small clustering of points can be observed. The best result in terms of approximation uniformity is achieved by the PSM, for which the Pareto optimal points are almost equally distributed along the Pareto front.

![Figure 2.2: Approximations of the Pareto fronts for the subproblem $(\hat{J}_1, \hat{J}_2)$ obtained by the WSM, the ERPM and the PSM.](image)

Having computed the approximations of the Pareto fronts for all three subproblems $(\hat{J}_1, \hat{J}_2)$, $(\hat{J}_1, \hat{J}_3)$ and $(\hat{J}_2, \hat{J}_3)$, we can generate the reference points for solving the problem $(\hat{J}_1, \hat{J}_2, \hat{J}_3)$ by the ERPM and the PSM as described in the Sections 1.6.4 and 1.7.4, cf. Figure 2.3.

Note that in this example, Assumption 1.4 for the PSM is not satisfied, since it holds $\mathcal{L}^D(U^{(1,3)}_{\text{opt}}) \cap D_2 \neq \emptyset$, see the pink crosses on the $\hat{J}_1 \hat{J}_3$-coordinate plane in Figure 2.3(b). Nevertheless, we see that the procedure for computing reference points also works in this case. The only drawback is that some redundant reference points are chosen as well. These points are indicated in red in Figure 2.3(b). Although it is quite easy to see that these reference points are not necessary for computing the Pareto front, we did not show this property analytically. And also numerically, it is hard to detect these points in general, since the discrete intersection $\mathcal{L}^D(U^{\{1,3\}}_{\text{opt}}) \cap D_2$ does not result in the same grid structure as the reference point grid $Z^h_2$ on $D_2$. Thus, it is not possible to exclude the redundant reference points by only componentwise comparisons. Note that we showed a criteria to remove such reference points in Theorem 1.7.55 and Remark 1.7.56. However, the number of reference points which can be removed by this criteria depends on the order in which...
the PSPs are solved. In particular, it is in general not possible to remove every redundant reference point in practice. In this example, five out of the ten redundant reference points could be removed.

Finally, the respective approximations of the Pareto front of the problem \((\hat{J}_1, \hat{J}_2, \hat{J}_3)\) can be seen in Figure 2.4. Again, the WSM reveals its drawbacks by producing a clustering of points in the top right part as well as a rather badly approximated left part of the Pareto front. For the ERPM, we can clearly see the guaranteed coverage of the Pareto front, cf. Theorem 1.6.7 and Remark 1.6.70. However, there is also a clustering of points in the middle of the approximation of the Pareto front mainly due to the structure of the reference points, which are chosen on the three shifted coordinate planes. Again, the best results in terms of approximation uniformity is obtained by the PSM. The interior of the Pareto front is almost approximated equidistantly by the computed Pareto optimal points. Only at the boundary, there is a small clustering of points. This can be partly explained by the redundantly chosen reference points due to Assumption 1.4 not being satisfied.

Our observations are confirmed by the quality measures of coverage and uniformity in Table 2.1. As predicted by the theory, the coverage for the ERPM and the PSM is of the order of the grid sizes \(h_{\text{ERPM}} = 0.3\) and \(h_{\text{PSM}} = 0.2\), respectively. In contrast to this, the coverage of the Pareto front for the WSM is three times as high as for the other two methods. Moreover, as can be seen in Table 2.2, we would need to use a discretization of the set \(\Delta_k\) with 85 points per dimension to reach the same coverage as for the ERPM and the PSM.

The clustering of approximation points for the WSM and the ERPM leads to uniformity values of 2.12 and 3.00, respectively, which are still quite good values. Nevertheless, the approximation obtained by the PSM has the best uniformity with a value of 1.52. Thus, in terms of approximation quality of the Pareto front, the PSM gives us the best coverage and uniformity while using the fewest approximation points. However, by comparing
the computational times of the three methods, the major drawback of the PSM becomes obvious. In comparison to the WSM, the computational time per Pareto optimal points is around 22 times higher. This is due to the structure of the PSP, which is actually a reformulated non-smooth optimization problem. In the numerical implementation, this difficulty is manifested in the fact that we have to use an augmented Lagrangian penalty method to solve the PSPs. This method subsequently solves optimization problems including a penalty term for the inequality constraint \( \hat{J}(u) - z \leq t \), cf. Appendix B. Consequently, the numerical effort for solving a PSP is much higher than for solving a WSP or an ERPP, which can be clearly seen by the average number of iterations and the number of required PDE solves. Comparing the runtimes for the WSM and the ERPM, we observe that the WSM is faster by a factor of around 2. This is due to the fact that the WSPs are actually linear-quadratic optimization problems, while the ERPPs are of quartic nature. In particular, if the optimal control does not have any active point, solving a WSP corresponds to solving one linear equation system, i.e., only one iteration is needed.

Figure 2.4: Approximations of the Pareto fronts for the problem \((\hat{J}_1, \hat{J}_2, \hat{J}_3)\) obtained by the WSM, the ERPM and the PSM.
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>WSM</td>
<td>0.43</td>
<td>2.12</td>
<td>325</td>
<td>2741</td>
<td>8.4</td>
<td>2.1</td>
<td>12817</td>
</tr>
<tr>
<td>ERPM</td>
<td>0.13</td>
<td>3.00</td>
<td>330</td>
<td>5381</td>
<td>16.3</td>
<td>4.1</td>
<td>24740</td>
</tr>
<tr>
<td>PSM</td>
<td>0.12</td>
<td>1.52</td>
<td>210</td>
<td>39308</td>
<td>187.2</td>
<td>27.6</td>
<td>179363</td>
</tr>
</tbody>
</table>

Table 2.1: Performance results for the different scalarization methods. The value of coverage was computed w.r.t. a reference solution consisting of 2637 approximation points computed by the ERPM.

in the Newton-CG algorithm. For the ERPP, the representations of the gradient and the Hessian of the cost function contain the difference $\hat{J}(u) - z$ as a factor, which is known to be positive at the optimal control $\bar{u}$ by Lemma 1.6.8. However, especially in parts of the Pareto front where one or more components of the difference $\hat{J}(\bar{u}) - z$ are close to zero, it might happen during the optimization algorithm that one or several terms $\hat{J}_i(u) - z_i$ become negative. This might result in the Hessian matrix not being positive definite, which might in turn prevent the CG algorithm from computing a Newton direction. In these cases, we take the negative gradient as a descent direction instead, which slows down the convergence of the numerical algorithm. This can also be seen by looking at average number of iterations and the number of PDE solves, which are both around 2 times higher for the ERPM than for the WSM.

<table>
<thead>
<tr>
<th></th>
<th>Coverage</th>
<th>Uniformity</th>
<th>Cardinality</th>
<th>Runtime [s]</th>
</tr>
</thead>
<tbody>
<tr>
<td>$n = 25$</td>
<td>0.43</td>
<td>2.1</td>
<td>325</td>
<td>2741</td>
</tr>
<tr>
<td>$n = 35$</td>
<td>0.31</td>
<td>2.2</td>
<td>630</td>
<td>5494</td>
</tr>
<tr>
<td>$n = 45$</td>
<td>0.24</td>
<td>2.2</td>
<td>1035</td>
<td>8710</td>
</tr>
<tr>
<td>$n = 85$</td>
<td>0.13</td>
<td>2.2</td>
<td>3655</td>
<td>29410</td>
</tr>
</tbody>
</table>

Table 2.2: Performance for different numbers of weights per dimension $n$ for the WSM.

Nevertheless, if we want to achieve the same level of coverage for all three methods, we see that the runtime for the WSM is almost six times as high as for the ERPM. The reason is that a total of around 3600 WSPs have to be solved to reach the same coverage as the ERPM and the PSM do with 330 and 210 approximation points, respectively. In conclusion, this comparison clearly indicates that the ERPM is favorable over the WSM and the PSM for this problem, since it can guarantee a good coverage of the Pareto front while having a moderate runtime.

For all three methods, we observe that more than 95% of the computational time was spent solving the state or adjoint equation. In Chapter 3, we show how the use of model-order reduction can reduce the computational time noticeably by replacing the FE discretization of the PDE with a low-dimensional surrogate model.
2.2 A Non-Convex Multiobjective Parameter Optimization Problem

Let us now introduce a non-convex MPOP of an elliptic PDE, where the governing PDE is a linear diffusion-reaction equation. However, we consider also the bilinear form to be parameter-dependent, which makes the solution mapping non-linear and the problem non-convex. Considering parameter-dependent bilinear forms of elliptic PDEs is fairly standard in the literature, see, e.g., [HPUU09, KG14b, RHP07]. Consequently, this framework has already been used in the context of scalar parameter optimization problems, cf., e.g., [KMO+20, QGVW17].

2.2.1 The State Equation

Analogously to Section 2.1.1, let $\Omega \subset \mathbb{R}^d$ for $d \in \{2, 3\}$ be a bounded domain with Lipschitz continuous boundary $\Gamma = \partial \Omega$. Furthermore, let $\Omega_1, \ldots, \Omega_m$ be a pairwise disjoint decomposition of the domain $\Omega$ and set $\Gamma_i := \partial \Omega_i \cap \partial \Omega$ for all $i = 1, \ldots, m$. Then the state equation is given by the following elliptic diffusion-reaction equation with Robin boundary condition:

$$
-\nabla \cdot \left( \sum_{i=1}^{m} u_i^\kappa \chi_{\Omega_i}(\mathbf{x}) \nabla y(\mathbf{x}) \right) + u^r r(\mathbf{x}) y(\mathbf{x}) = f(\mathbf{x}) \quad \text{a.e. in } \Omega,
$$

$$
\frac{u_i^\kappa}{\partial n}(s) + \alpha y(s) = \alpha y_a(s) \quad \text{a.e. on } \Gamma_i.
$$

(2.2.1)

For every $i \in \{1, \ldots, m\}$, the parameter $u_i^\kappa > 0$ represents the diffusion coefficient on the subdomain $\Omega_i$. By $r \in L^\infty(\Omega)$, we denote the reaction function, which is supposed to satisfy $r > 0$ a.e. in $\Omega$ and is controlled by the parameter $u^r > 0$. On the right-hand side of the first equation, we have the source term $f \in L^2(\Omega)$. The constant $\alpha > 0$ in the Robin boundary condition models the heat flux with the outside world, where a temperature of $y_a \in L^2(\Gamma)$ is assumed. In total, the parameter space is given by $\mathcal{U} = \mathbb{R}^m \times \mathbb{R}$ and any parameter $u \in \mathcal{U}$ can be written as $u = (u^\kappa, u^r)^T$ with $u^\kappa = (u_1^\kappa, \ldots, u_m^\kappa)^T \in \mathbb{R}^m$.

Recall that we defined the spaces $H := L^2(\Omega)$ and $V := H^1(\Omega)$ in Section 2.1.1. Then for any $u \in \mathcal{U}$ the weak formulation of (2.2.1) reads: Find $y \in V$ such that

$$
a(u; y, \varphi) = \mathcal{F}(\varphi) \quad \text{for all } \varphi \in V.
$$

(2.2.2)

Here the parameter-dependent symmetric bilinear form $a(u; \cdot, \cdot): V \times V \rightarrow \mathbb{R}$ is given by

$$
a(u; \varphi, \psi) := \sum_{i=1}^{m} u_i^\kappa \int_{\Omega_i} \nabla \varphi(\mathbf{x}) \nabla \psi(\mathbf{x}) \, d\mathbf{x} + u^r \int_{\Omega} r(\mathbf{x}) \varphi(\mathbf{x}) \psi(\mathbf{x}) \, d\mathbf{x}
$$

$$
+ \alpha \int_{\Gamma} \varphi(s) \psi(s) \, ds
$$

for all $\varphi, \psi \in V$. The linear functional $\mathcal{F} \in V'$ is defined by

$$
\mathcal{F}(\varphi) := \int_{\Omega} f(\mathbf{x}) \varphi(\mathbf{x}) \, d\mathbf{x} + \alpha \int_{\Gamma} y_a(s) \varphi(s) \, ds \quad \text{for all } \varphi \in V.
$$
Lemma 2.2.1. (i) For all \( u \in \mathcal{U} \) it holds
\[
\|a(u; \cdot, \cdot)\|_{L(V, V')} \leq C \|u\|_{\mathcal{U}}
\]
with a constant \( C > 0 \), which does not depend on \( u \).

(ii) For all \( u \in \mathcal{U} \) with \( u^\kappa > 0 \) and \( u^r > 0 \), it holds
\[
a(u; \varphi, \varphi) \geq \min(u_{1}^\kappa, \ldots, u_{m}^\kappa, u^r) \|\varphi\|_V^2 \text{ for all } \varphi \in V.
\]

(iii) The mapping \( F \in V' \) is well-defined.

Proof. All statements follow with similar arguments as in the parabolic case. Thus, we refer again to [Mec19, Lemma 1.4], where similar operators were considered.

Theorem 2.2.2. Let \( u \in \mathcal{U} \) with \( u > 0 \) be arbitrary. Then there is a unique solution \( y = y(u) \in V \) of (2.2.1). Moreover, the estimate
\[
\|y\|_V \leq C \left( \|f\|_{L^2(\Omega)} + \|y_a\|_{L^2(\Gamma)} \right)
\]
holds with a constant \( C > 0 \), which depends continuously on \( u \), but is independent of \( f \) and \( y_a \).

Proof. Using Lemma 2.2.1, this can be concluded from the Lax-Milgram Theorem, see, e.g., [Eva10, Section 6.2.1, Theorem 1].

Definition 2.2.3. Let \( u_{\min}^\kappa \in (0, \infty)^m \) and \( u_{\min}^r > 0 \) be arbitrary. Then we define the set
\[
\mathcal{U}_{eq} := \{ u \in \mathcal{U} | u^\kappa > u_{\min}^\kappa, u^r > u_{\min}^r \}.
\]

In view of Theorem 2.2.2, it is possible to define the solution operator \( S : \mathcal{U}_{eq} \to V \), which maps any parameter \( u \in \mathcal{U}_{eq} \) to the unique solution \( y = S(u) \in V \) of (2.2.2).

Remark 2.2.4. Due to the results shown in Lemma 2.2.1, we can conclude that
\[
a(u; \varphi, \varphi) \geq \alpha_{\min} \|\varphi\|_V^2 \text{ for all } \varphi \in V
\]
holds for all \( u \in \mathcal{U}_{eq} \), where \( \alpha_{\min} := \min((u_{\min, 1}^\kappa), \ldots, (u_{\min, m}^\kappa), u_{\min}^r) > 0 \). In particular, the constant \( C \) in the a-priori estimate (2.2.3) can be chosen independently of \( u \) if we restrict ourselves to parameters \( u \in \mathcal{U}_{eq} \).

Theorem 2.2.5. The solution operator \( S \) is continuous.

Proof. Let \( (u_n)_{n \in \mathbb{N}} \subset \mathcal{U}_{eq} \) be a sequence and \( u \in \mathcal{U}_{eq} \) such that \( u_n \to u \) as \( n \to \infty \). Furthermore, define \( y_n := S(u_n) \) for all \( n \in \mathbb{N} \) and \( y := S(u) \). With these definitions, we can conclude
\[
a(u; y - y_n, \varphi) = F(\varphi) - a(u; y_n, \varphi) = -a(u - u_n; y_n, \varphi)
\]
for all \( \varphi \in V \). In particular, we can plug in \( \varphi = y - y_n \in V \), which implies
\[
\alpha_{\text{min}} \|y - y_n\|^2 \leq a(u; y - y_n, y - y_n) \leq C \|u - u_n\|_U \|y_n\|_V \|y - y_n\|_V.
\]
By rearranging the terms, we obtain
\[
\alpha_{\text{min}} \|y - y_n\|_V \leq C \|u - u_n\|_U \|y_n\|_V.
\]
Since the sequence \((u_n)_{n \in \mathbb{N}}\) is convergent, it is, in particular, bounded. Hence, by the a-priori estimate (2.2.3), the sequence \((y_n)_{n \in \mathbb{N}} \subset V\) is bounded as well (note that the constant \(C\) in the estimate is independent of the parameter, cf. Remark 2.2.4). Altogether, this implies \(\|y - y_n\|_V \to 0\) as \(n \to \infty\), which shows the continuity of \(S\).

Next, we want to show that the solution operator \(S\) is twice continuously Fréchet differentiable. To this end, we need the following remark on the differentiability of the operator \(a\) w.r.t. the parameter \(u\).

**Remark 2.2.6.** By \(\partial_u a\) we denote the partial derivative of \(a\) w.r.t. the parameter \(u\). Since \(a\) is linear in \(u\), it holds
\[
\partial_u a(u; \varphi, \psi)h = a(h; \varphi, \psi),
\]
\[
\partial^2_u a(u; \varphi, \psi) = 0 \in L(U, U')
\]
for all \(u, h \in U\) and all \(\varphi, \psi \in V\). In particular, we can identify \(\partial_u a(u; \varphi, \psi) \in U'\) by
\[
\partial_u a(u; \varphi, \psi) = \begin{pmatrix}
\int_{\Omega_1} \nabla \varphi(x) \nabla \psi(x) \, dx \\
\vdots \\
\int_{\Omega_n} r(x) \varphi(x) \, dx \\
\int_{\Omega} r(x) \varphi(x) \psi(x) \, dx
\end{pmatrix} \in U'
\]
by using the Riesz representation theorem.

**Lemma 2.2.7.** The solution operator \(S\): \(U_{\text{eq}} \to V\) is twice continuously Fréchet differentiable. For the first derivative \(S'\): \(U_{\text{eq}} \to L(U, V)\), we have that for any \(u \in U_{\text{eq}}\) and \(h \in U\) the function \(y^h := S'(u)h \in V\) solves the equation
\[
a(u; y^h, \varphi) = -\partial_u a(u; S(u), \varphi)h \quad \text{for all } \varphi \in V. \tag{2.2.4}
\]
The second derivative \(S''\): \(U_{\text{eq}} \to L(U, L(U, V))\) is given as follows: For any \(u \in U_{\text{eq}}\) and \(h_1, h_2 \in U,\) the function \(y^{h_1, h_2} := S''(u)(h_1, h_2)\) solves the equation
\[
a(u; y^{h_1, h_2}, \varphi) = -\partial_u a(u; S'(u)h_1, \varphi)h_2 - \partial_u a(u; S'(u)h_2, \varphi)h_1 \quad \text{for all } \varphi \in V. \tag{2.2.5}
\]

**Proof.** This can be shown by rewriting (2.2.2) in the form \(e(y, u) = 0\) with \(e: V \times U \to V'\) being defined by
\[
\langle e(y, u), \varphi \rangle_{V', V} := a(u; y, \varphi) - F(\varphi)
\]
for any \(y \in Y, u \in U\) and \(\varphi \in V\), and then using the implicit function theorem, cf. [HPUU09, Section 1.6]. A detailed proof of a similar statement was also carried out in [Bee19, Section 2.1] for a parabolic advection-diffusion equation with controlled advection term.
2.2.2 The Parameter Optimization Problem

Now we introduce the elliptic parameter optimization problem, which will serve as our second model problem of PDE-constrained optimization.

Let the state space be given by \( \mathcal{Y} := \mathcal{V} \). Analogously to Section 2.1.2, we introduce the quadratic cost function \( J: \mathcal{Y} \times \mathcal{U} \rightarrow \mathbb{R} \) defined by

\[
J(y,u) := \sigma_\Omega \| y - y_\Omega \|_H^2 + \sigma_\mathcal{U} \| u - u_d \|_{\mathcal{U}}^2
\]

with non-negative parameters \( \sigma_\Omega, \sigma_\mathcal{U} \geq 0 \), the desired state \( y_\Omega \in H \), and the desired parameter \( u_d \in \mathcal{U} \). Then the parameter optimization problem reads

\[
\min_{(y,u) \in \mathcal{Y} \times \mathcal{U}} J(y,u) \quad \text{s.t.} \quad y = y(u) \text{ solves (2.2.2) for the parameter } u,
\]

\[
\quad u \in \mathcal{U}_{\text{ad}} := \{ u \in \mathcal{U} \mid u_a \leq u \leq u_b \},
\]

where \( u_a, u_b \) with \( u_a \leq u_b \) are lower and upper bounds on the parameter \( u \), which might also take the values \(-\infty\) and \( \infty \), respectively.

By Theorem 2.2.2, we can only conclude that there is a unique solution \( y = y(u) \) of (2.2.2) if \( u \in \mathcal{U}_{\text{eq}} \) holds. Therefore, we will assume that \( \mathcal{U}_{\text{ad}} \subset \mathcal{U}_{\text{eq}} \) holds in the following.

**Assumption 2.2.** We assume that \( \mathcal{U}_{\text{ad}} \subset \mathcal{U}_{\text{eq}} \) holds.

**Definition 2.2.8.** Let Assumption 2.2 be satisfied. Then we can define the essential cost function \( \hat{J}: \mathcal{U}_{\text{eq}} \rightarrow \mathbb{R} \) by

\[
\hat{J}(u) := J(S(u), u).
\]

This allows us to introduce the reduced parameter optimization problem

\[
\min_{u \in \mathcal{U}_{\text{eq}}} \hat{J}(u) \quad \text{s.t.} \quad u \in \mathcal{U}_{\text{ad}}. \tag{2.2.7}
\]

Under appropriate assumptions, we can show that \( \hat{J} \) fulfills the assumptions for non-convex multiobjective optimization, cf. Assumptions 1.2 and 1.3.

**Assumption 2.3.** If \( \mathcal{U}_{\text{ad}} \) is unbounded assume that \( \sigma_\mathcal{U} > 0 \) holds.

**Lemma 2.2.9.** If Assumptions 2.2 and 2.3 hold, then the function \( \hat{J} \) is continuous and bounded from below. Moreover, if \( \mathcal{U}_{\text{ad}} \) is unbounded, it holds \( \lim_{\| u \|_{\mathcal{U}} \rightarrow \infty} \hat{J}(u) = \infty \). In particular, \( \hat{J} \) satisfies the Assumptions 1.2 and 1.3.

**Proof.** In Theorem 2.2.5, we showed that the solution operator \( S \) is continuous. Therefore, \( \hat{J} \) is continuous as a concatenation of continuous functions, which directly implies that \( \hat{J} \) is upper semi-continuous. Moreover, \( \hat{J} \) is also weakly lower semi-continuous, since \( \mathcal{U} \) is
finite-dimensional so that weak and strong convergence coincide. Clearly, \( \hat{J} \) is bounded from below by 0. Finally, if \( \mathcal{U}_{ad} \) is unbounded, Assumption 2.3 guarantees that \( \sigma_\mathcal{U} > 0 \) holds. Now \( \lim_{\|u\|_\mathcal{U} \to \infty} \hat{J}(u) = \infty \) follows immediately.

The previous lemma implies, in particular, that the problem (2.2.7) has a global solution.

**Corollary 2.2.10.** Let Assumption 2.2 and 2.3 be satisfied. Then the problem (2.2.7) has a global solution \( \bar{u} \in \mathcal{U}_{ad} \).

**Proof.** By Lemma 2.2.9, the function \( \hat{J} \) satisfies Assumption 1.2. Thus, Corollary 1.7.10 applied to \( z := 0 \) and \( r := 1 \) tells us that \( (RPSP(z,r)) \) has a global solution \( \bar{u} \in \mathcal{U}_{ad} \). It is easy to see that this is also a global solution of (2.2.7).

### 2.2.2.1 Computation of Derivatives

Similarly to Section 2.1.2.1, we want to derive representations of the gradient and the Hessian of the essential cost function \( \hat{J} \), which can be evaluated in practice. To this end, we start again with the 'standard' representation of the gradient and the Hessian.

**Lemma 2.2.11.** The function \( \hat{J} \) is twice continuously Fréchet differentiable. For any \( u \in \mathcal{U}_{eq} \) and any direction \( h \in \mathcal{U} \), it holds

\[
\nabla \hat{J}(u) = \sigma_\Omega S'(u)^*(S(u) - y_\Omega) + \sigma_\mathcal{U}(u - u_d) \quad \in \mathcal{U},
\]

\[
\nabla^2 \hat{J}(u)h = \sigma_\Omega \left[ (S''(u)h)^*(S(u) - y_\Omega) + S'(u)^* S'(u)h \right] + \sigma_\mathcal{U}h \quad \in \mathcal{U}.
\]

**Proof.** This follows directly by applying the chain rule and the definition of the adjoint operator.

Again, the question is how the adjoint operators \( S'(u)^* \) and \( (S''(u)h)^* \) can be evaluated in practice. As for the parabolic case, this can be done by introducing the adjoint equation.

**Definition 2.2.12.** Let \( u \in \mathcal{U} \) and \( z_\Omega \in \mathcal{H} \) be arbitrary. Then the equation

\[
a(u; \varphi, p) = \langle z_\Omega, \varphi \rangle_\mathcal{H} \quad \text{for all } \varphi \in V.
\]

is called the adjoint equation of (2.2.2).

**Theorem 2.2.13.** For any \( u \in \mathcal{U}_{eq} \) and any \( z_\Omega \in \mathcal{H} \), the adjoint equation (2.2.8) has a unique solution \( p = p(z_\Omega) \in V \). Moreover, there is a constant \( C > 0 \), which can be chosen independently of \( u \) and \( z_\Omega \), such that the estimate

\[
\|p\|_V \leq C \|z_\Omega\|_H
\]

is satisfied.

**Proof.** This can be concluded with the same arguments as the proof of Theorem 2.2.2.
Lemma 2.2.14. Let $u \in \mathcal{U}_{eq}$ and $z_\Omega \in H$ be arbitrary. Let $p = p(z_\Omega) \in V$ be the unique solution of (2.2.8). Then it holds
\[
-\partial_u a(u; S(u), p)h = \langle S'(u)^* z_\Omega, h \rangle_{\mathcal{U}}
\]
for all $h \in \mathcal{U}$.

Proof. Let $h \in \mathcal{U}$ be arbitrary. Since $p$ solves (2.2.8) and we have $S'(u)h \in V$, it holds
\[
\langle S'(u)^* z_\Omega, h \rangle_{\mathcal{U}} = \langle S'(u)h, z_\Omega \rangle_{H} = a(u; S'(u)h, p).
\]
By Lemma 2.2.7, the function $S'(u)h$ solves (2.2.4), so that we can further conclude
\[
\langle S'(u)^* z_\Omega, h \rangle_{\mathcal{U}} = a(u; S'(u)h, p),
\]
which is what we had to show. \qed

Definition 2.2.15. We define the solution operator of the adjoint equation $\mathcal{A} : \mathcal{U}_{eq} \to V$, where for any given $u \in \mathcal{U}_{eq}$, $p := \mathcal{A}(u)$ solves
\[
a(u; \varphi, p) = \langle \sigma_\Omega(S(u) - y_{eq}), \varphi \rangle_{H} \quad \text{for all} \ \varphi \in V.
\]

Corollary 2.2.16. Let Assumption 2.2 be satisfied and $u \in \mathcal{U}_{ad}$ be arbitrary. Then it holds
\[
\nabla \tilde{J}(u) = -\partial_u a(u; S(u), \mathcal{A}(u)) + \sigma_\Omega(u - u_d) \in \mathcal{U},
\]
where we use the representation of $\partial_u a(u; S(u), \mathcal{A}(u)) \in \mathcal{U}'$ in $\mathcal{U}$, cf. Remark 2.2.6.

Proof. This follows from Lemma 2.2.11, Lemma 2.2.14 and Definition 2.2.15. \qed

To derive a representation of the Hessian of $\tilde{J}$ using the adjoint equation, we need to show that the solution operator $\mathcal{A}$ is Fréchet differentiable and find a representation for it. This is done in the next lemma.

Lemma 2.2.17. The solution operator $\mathcal{A} : \mathcal{U}_{eq} \to V$ is continuously Fréchet differentiable. For the first derivative $\mathcal{A}' : \mathcal{U}_{eq} \to L(\mathcal{U}, V)$, we have that for any $u \in \mathcal{U}_{eq}$ and $h \in \mathcal{U}$ the function $p^h := \mathcal{A}'(u)h \in V$ solves the equation
\[
a(u; \varphi, p^h) = -\partial_u a(u; \varphi, \mathcal{A}(u))h + \sigma_\Omega(S'(u)h, \varphi)_{V', V} \quad \text{for all} \ \varphi \in V.
\]

Proof. This can be concluded with the same arguments as the proof of Lemma 2.2.7. \qed

From Corollary 2.2.16 and Lemma 2.2.17, we can directly conclude the desired representation of the Hessian of $\tilde{J}$.

Corollary 2.2.18. Let $u \in \mathcal{U}_{ad}$ and $h \in \mathcal{U}$ be arbitrary. Then it holds
\[
\nabla^2 \tilde{J}(u)h = -\partial_u a(u; S'(u)h, \mathcal{A}(u)) - \partial_u a(u; S(u), \mathcal{A}'(u)h) + \sigma_\Omega h \in \mathcal{U}.
\]

Proof. This follows from Corollary 2.2.16 and Lemma 2.2.17 by applying the chain rule. \qed
2.2.3 The Multiobjective Parameter Optimization Problem

Similarly to Section 2.1.3, we shortly introduce the notation for a MPOP, which is based on the parameter optimization problem introduced in Section 2.2.2. So let \( k \in \mathbb{N} \) be fixed and \( \sigma^{(1)}, \ldots, \sigma^{(k)} \geq 0 \) as well as \( \sigma^{(1)}, \ldots, \sigma^{(k)} \geq 0 \) be non-negative parameters. Furthermore, denote by \( y^{(1)}_\Omega, \ldots, y^{(k)}_\Omega \in H \) the desired states and by \( u^{(1)}_d, \ldots, u^{(k)}_d \in U \) the desired parameters. Then we can define the multiobjective essential cost functions \( \hat{J}_1, \ldots, \hat{J}_k : U_{eq} \to \mathbb{R} \) by

\[
\hat{J}_i(u) := \frac{\sigma^{(i)}_\Omega}{2} \| S(u) - y^{(i)}_\Omega \|_H^2 + \frac{\sigma^{(i)}_U}{2} \| u - u^{(i)}_d \|_U^2
\]

for all \( u \in U_{eq} \) and any \( i \in \{1, \ldots, k\} \). Supposing that Assumption 2.2 is satisfied, we can define the multiobjective parameter optimization problem

\[
\min_{u \in U_{ad}} \hat{J}(u) = \min_{u \in U_{ad}} \begin{pmatrix}
\hat{J}_1(u) \\
\vdots \\
\hat{J}_k(u)
\end{pmatrix}.
\]

This problem fits into the framework of non-convex multiobjective optimization as described in Assumptions 1.2 and 1.3.

**Lemma 2.2.19.** Assume that Assumption 2.2 holds and that Assumption 2.3 is satisfied for all cost functions \( \hat{J}_1, \ldots, \hat{J}_k \). Then the Assumptions 1.2 and 1.3 are satisfied.

**Proof.** This follows directly from Lemma 2.2.9.

2.2.4 Numerical Experiments

We conduct numerical experiments in this section by applying the three scalarization methods from Chapter 1 to the model problem (MPOP). The obtained results are then compared with respect to the approximation quality of the Pareto front as well as the required runtimes.

In the numerical test, we choose the domain \( \Omega \) to be the two-dimensional unit square, i.e., \( \Omega = (0,1)^2 \). For the elliptic state equation (2.2.1), we make the following parameter choices:

- There are four diffusion parameters \( u^{(1)}_\kappa, \ldots, u^{(4)}_\kappa \) for the subdomains \( \Omega_1 = (0,0.5) \times (0,0.5), \Omega_2 = (0,0.5) \times (0.5,1), \Omega_3 = (0.5,1) \times (0,0.5), \Omega_4 = (0.5,1) \times (0.5,1) \).
- The reaction function is chosen as \( r(x) = 1 \) for all \( x \in \Omega \).
- The Robin coefficient on the boundary \( \Gamma \) is set to \( \alpha = 0 \), i.e., we impose homogeneous Neumann boundary conditions.
The source term \( f \) is of the form \( f(x) = \sum_{i=1}^{4} c_i \chi_{\Omega_i}(x) \) with \( c_1 \approx 2.76, c_2 \approx -0.96, c_3 \approx 0.51 \) and \( c_4 \approx -1.66 \). These parameter values were chosen randomly with the goal of obtaining a numerical example with a non-convex Pareto front.

For the spatial discretization of the state equation, we apply the Finite Element (FE) method with 1340 nodes and piecewise linear basis functions.

In this framework, we consider the problem (MPOP) with the three cost functions

\[
\hat{J}_1(u) := \frac{1}{2} \| S(u) - y_{\Omega}^{(1)} \|_H^2 + \frac{0.002}{2} \| u - u_d^{(1)} \|_u^2,
\]

\[
\hat{J}_2(u) := \frac{1}{2} \| S(u) - y_{\Omega}^{(2)} \|_H^2 + \frac{0.002}{2} \| u - u_d^{(2)} \|_u^2,
\]

\[
\hat{J}_3(u) := \frac{0.05}{2} \| u - u_d^{(3)} \|_u^2,
\]

with the desired states

\[
y_{\Omega}^{(1)}(x) := \chi_{(0, 0.5) \times (0, 1)}(x),
\]

\[
y_{\Omega}^{(2)}(x) := \chi_{(0.5, 1) \times (0, 1)}(x),
\]

and the desired parameter values

\[
u_d^{(1)} = u_d^{(2)} := (2, 0, 0, 0.3)^T,
\]

\[
u_d^{(3)} := (2, 1, 1, 0.3)^T.
\]

The lower and upper parameter bounds are given by

\[
u_a = (2, 0.1, 0.1, 0.1, 0.3)^T \quad \text{and} \quad \nu_b = (2, 4, 4, 4, 0.3)^T,
\]

respectively. This implies that \( u_a^\ell = 2 \) and \( u_r = 0.3 \) are seen as constants and we only optimize over the three parameters \( u_2^\ell, u_3^\ell \) and \( u_4^\ell \). Note furthermore, that the desired parameters \( u_d^{(1)} = u_d^{(2)} \) are not admissible. In fact, as the parameters of the source term, they were chosen such that the resulting Pareto front is non-convex.

Although it cannot be guaranteed from a theoretical point of view that the entire Pareto front can be computed by the WSM and the ERPM, we still want to compare the results for these methods with those for the PSM. Note that we choose again the target direction \( r = (1, 1, 1)^T \) for the PSM.

As for the linear-quadratic MOCP in Section 2.1.4, the arising WSPs and ERPPs are solved by a projected Newton-CG method ([Kel99, Section 5.5.2]), for which the termination tolerances for the first-order critical condition, the relative cost function decrease and the Armijo step length are chosen as for the MOCP. The maximal number of iterations is set to \( i_{\text{max}} = 600 \) for the Newton-CG method, since the occurring problems are harder to solve than for the linear-quadratic MOCP.

For solving the PSPs, we use again the augmented Lagrangian penalty method from Appendix B with the same parameter choices as for the linear-quadratic MOCP.
Global vs. local minimization. Since the arising scalarized optimization problems are non-convex, there might be several local minimizers. Our utilized optimization routines can in general only guarantee the convergence to local minimizers. To increase the chances of computing the global minimizer of the respective optimization problem for the subproblems of (MPOP) with two cost functions, we solve the optimization problem twice for every weight or reference point with two different initial values. This is done by computing once the Pareto front from top to bottom and once from bottom to top. For the case of three cost functions, we only minimize the optimization problems once for every weight or reference point, since the results did not change when solving every optimization problem twice.

Of course, this procedure is only of heuristic nature. In fact, the use of global optimization algorithms would also be possible and even advisable in this situation. However, in this thesis, we did not proceed in this direction.

Moreover, after solving a subproblem of (MPOP), we remove non-Pareto optimal points by a non-dominance test.

Choice of initial value. We use the same procedure as for the linear-quadratic MOCP to determine the initial value for the WSM and the ERPM, cf. Section 2.1.4. For the PSM, this choice does not lead to satisfactory results due to the effects of global vs. local minimization, see Figure 2.9(d). Thus, we use a different scheme for the choice of the initial value for PSPs corresponding to reference points for the entire problem ($\hat{J}_1$, $\hat{J}_2$, $\hat{J}_3$):

Let $\bar{u}_i$ be the minimizer of $\hat{J}_i$ for $i = 1, 2, 3$. Then, if $z \in D_i$, we choose $\bar{u}_i$ as the initial value for solving (PSP($z, r$)).

Discretization of $\Delta_k$. For the WSM, we choose 60 uniformly distributed weights per dimension, i.e., we use the discrete sets of weights

$$\Delta_2^d := \bigcup_{i=0}^{59} \left\{ \left( \frac{i}{59}, 1 - \frac{i}{59} \right) \right\} \quad \text{and} \quad \Delta_3^d := \bigcup_{i=0}^{59} \bigcup_{j=0}^{59-i} \left\{ \left( \frac{i}{59}, \frac{j}{59}, 1 - \frac{i}{59} - \frac{j}{59} \right) \right\}.
$$

Choice of reference points for ERPM and PSM. For the ERPM and the PSM, we choose the shifting vectors $\tilde{d}_{\text{ERPM}} = 0.01 \cdot (1, 1, 1)^T$ and $\tilde{d}_{\text{PSM}} = 0.001 \cdot (1, 1, 1)^T$, while the grid size $h$ for the reference point grid is set to $h_{\text{ERPM}} = h_{\text{PSM}} = 0.003$.

2.2.4.1 Numerical results

According to the hierarchical computation of the Pareto front, at first the three individual cost functions $\hat{J}_1$, $\hat{J}_2$, $\hat{J}_3$ are minimized by using the projected Newton-CG method. Here we already observe that the initial value of the optimization has to be chosen carefully. For minimizing the cost function $\hat{J}_2$, an initial value of $u_o = (2, 0.1, 3, 3, 0.3)$ yields the global minimizer $\bar{u}_2 \approx (2, 0.1, 2.70, 1.8, 0.3)$ with the function value $\hat{J}_2(\bar{u}_2) \approx 0.145$, whereas an initial value of $u_o = (2, 1.5, 1, 1, 0.3)$ results in the computation of the local minimizer $\tilde{u}_2 \approx (2, 1.56, 1.88, 1.58, 0.3)$ with a function value of $\hat{J}_2(\tilde{u}_2) \approx 0.151$ (see also Figure 2.2(c), where both the global and the local minimum of $\hat{J}_2$ can be seen). Again, the use of a global minimization algorithm would be beneficial in this situation to not be dependent on the correct (heuristic) choice of the initial value.
Figure 2.5: Approximations of the Pareto fronts for the subproblems with two cost functions for the WSM, the ERPM and the PSM.

Based on these minimizers, the reference points for solving the subproblems with two cost functions \((\hat{J}_1, \hat{J}_2)\), \((\hat{J}_1, \hat{J}_3)\) and \((\hat{J}_2, \hat{J}_3)\) can be generated for the ERPM and the PSM. The resulting approximations of the Pareto fronts are depicted in Figure 2.5 for the WSM, the ERPM and the PSM. These figures nicely show the different properties that a Pareto front might have in the non-convex case:

The Pareto front of the subproblem \((\hat{J}_1, \hat{J}_2)\) is disconnected and consists of two different connected parts, both of which have a 'convex' shape, cf. Figure 2.5(g). The reason for the disconnectedness is the local minimizer of \(\hat{J}_2\), which can be seen at the point \((0.118, 0.151)\).

For the second subproblem \((\hat{J}_1, \hat{J}_3)\), the Pareto front is connected, but its shape is slightly non-convex, as can be seen by looking at the approximation of the Pareto front obtained from using the PSM in Figure 2.5(h).
Lastly, the Pareto front of the subproblem \((\hat{J}_2, \hat{J}_3)\) has a kink at around \((0.151, 0.0622)\), where the two convex parts of the Pareto front meet, as can be seen in Figure 2.5(i). This kink is again due to the local minimizer of the function \(\hat{J}_2\).

Since the Pareto fronts for the subproblems \((\hat{J}_1, \hat{J}_2)\) and \((\hat{J}_2, \hat{J}_3)\) consist of two convex parts and we solve each optimization problem twice with different initial parameter values (once following the Pareto front from top to bottom, once from bottom to top, see the explanation above), the WSM is still capable of approximating large parts of the Pareto fronts, cf. Figures 2.5(a),(c). Note that the missing parts close to the disconnectedness and the kink are due to the fact that the WSM is bad at approximating almost linear parts of the Pareto front (cf. [DD97]), and not due to the shapes of the Pareto fronts. In this case, it is even beneficial that the optimization algorithm is in general only able to compute the local minimizer of the optimization problem. If we were using a global optimization algorithm, parts around the disconnectedness and the kink would be missed by the WSM. However, especially for the subproblem \((\hat{J}_2, \hat{J}_3)\), the drawbacks of the WSM become visible once again, since the Pareto optimal points cluster in the lower right part of the Pareto front, while there is a large distance between neighboring Pareto optimal points in the upper left part of the Pareto front. The slight non-convexity of the Pareto front of the subproblem \((\hat{J}_1, \hat{J}_3)\) results in the WSM not being able to approximate large parts of the Pareto front, cf. Figure 2.5(b). To be precise, all Pareto optimal points between \((0.091, 0.053)\) and \((0.103, 0.033)\) are neither global nor local solutions of WSPs, so that all these points cannot be computed by the WSM.

For the ERPM, we observe that the quality of the approximation strongly depends on the shifting vector \(\tilde{d}_{\text{ERPM}}\). For the choice \(\tilde{d}_{\text{ERPM}} = 0.01 \cdot (1, 1, 1)^T\), the ERPM is able to approximate the entire Pareto fronts of all subproblems, except for a small neighborhood of the kink of the Pareto front of the subproblem \((\hat{J}_2, \hat{J}_3)\), cf. Figures 2.5(d)–(f). However, if we choose \(\tilde{d}_{\text{ERPM}} = 0.1 \cdot (1, 1, 1)^T\), we can see that some parts of the Pareto fronts cannot be approximated anymore, cf. Figure 2.6(a)–(c). This is due to the circular shape of the contour lines of the scalarization function \(g_z\) of the ERPM: If the reference point is close enough to the Pareto front, the optimal function value of the Euclidean reference point function \(\hat{J}^g_z\) is small. This implies that the corresponding contour line of the function \(g_z\) is a circle with a quite small radius, which can ‘fit’ into slight non-convexities of the Pareto front. For a larger distance of the reference point to the Pareto front, the corresponding circular contour line has a larger radius, so that the ERPM misses points in the non-convex part of the Pareto front. This is illustrated in Figure 2.7, where it can be seen that a closer reference points enables a better approximation of the area around the kink of the Pareto front of the subproblem \((\hat{J}_1, \hat{J}_3)\). However, we also observe that if the shifting vector is chosen too small (e.g., \(\tilde{d}_{\text{ERPM}} = 0.001 \cdot (1, 1, 1)^T\)), the Newton-CG algorithm needs in average more iterations to converge. In particular, the maximal number of iterations is reached for significantly more points than for a value of \(\tilde{d}_{\text{ERPM}} = 0.01 \cdot (1, 1, 1)^T\) leading to some inaccuracies. This can be explained by the fact that the term \(\hat{J}(u) - z\) gets negative more often if the reference point is chosen closer to the Pareto front, which slows down the convergence of the Newton-CG algorithm noticeably.
Figure 2.6: Approximations of the Pareto fronts for the subproblems \((\hat{J}_1, \hat{J}_2)\), \((\hat{J}_1, \hat{J}_3)\) and \((\hat{J}_2, \hat{J}_3)\) obtained by the ERPM and the PSM for the choices \(\hat{d}_{\text{ERPM}} = \hat{d}_{\text{PSM}} = 0.1 \cdot (1, 1, 1)^T\).  

Figure 2.7: Influence of the shifting vector \(\hat{d}_{\text{ERPM}}\) for the ERPM and the subproblem \((\hat{J}_2, \hat{J}_3)\). The closer the reference point to the Pareto front, the smaller the resulting circular shaped contour line of the Euclidean reference point cost function at the solution of the ERPP so that non-convexities of the Pareto front can be better approximated.
2.2 A Non-Convex MPOP

(a) ERPM

(b) PSM

(c) PSM – View onto $D_3$

**Figure 2.8:** Reference points for solving the entire problem ($\hat{J}_1, \hat{J}_2, \hat{J}_3$) for the ERPM and the PSM. Note that Assumption 1.4 is not satisfied for the PSM, which results in the choice of redundant reference points. These are indicated in red in (c).

In contrast to this, the PSM produces very uniform approximations of the Pareto fronts for all three subproblems independently of the choice of the shifting vector $\tilde{d}_\text{PSM}$, cf. Figure 2.5(g)–(i) and Figure 2.6(d)–(f). This confirms our analytical results, where we showed that the PSM is a suitable method for approximating Pareto fronts of general non-convex MOPs.

For generating the reference points for solving the problem ($\hat{J}_1, \hat{J}_2, \hat{J}_3$) by the ERPM and the PSM, we use again the procedure described in Sections 1.6.4 and 1.7.4. The resulting reference points can be seen in Figure 2.8. Although the test problem is non-convex, we see that all relevant reference points can still be generated for the ERPM. However, this requires a good approximation of the Pareto fronts of all subproblems with two cost functions, which cannot be guaranteed for the ERPM and strongly depends on the chosen shifting vector $\tilde{d}_\text{ERPM}$ in our case.
Figure 2.9: Approximations of the Pareto fronts for the non-convex problem \((\hat{J}_1, \hat{J}_2, \hat{J}_3)\) obtained by the WSM, the ERPM and the PSM. For the PSM, the quality of the approximation depends on the choice of the initial values for the optimization problems, see (c) & (d).

As for the linear-quadratic MOCP in Section 2.1.4, we observe that Assumption 1.4 is not satisfied, since it holds \(L^D(U_{\text{opt}}^{(1,2)}) \cap D_3 \neq \emptyset\). The resulting redundant reference points can be partly filtered out by using Theorem 1.7.55 and Remark 1.7.56, which is indicated by the points colored in red in Figure 2.8(b),(c).

Finally, the approximations of the Pareto fronts for the entire problem \((\hat{J}_1, \hat{J}_2, \hat{J}_3)\) are displayed in Figure 2.9. While the lower right part of the Pareto front can be approximated by all three methods, only the ERPM and the PSM are capable of approximating the upper left part appropriately, where the Pareto front is slightly non-convex; see also the non-convexity of the Pareto front of the subproblem \((\hat{J}_1, \hat{J}_3)\) in Figure 2.5(h). This is another confirmation that the WSM cannot deal with Pareto fronts with a non-convex shape. Although this can also not be guaranteed for the ERPM, it produces satisfying results in this example. Note, however, that the quality of the approximation depends
Table 2.3: Performance values of the different scalarization methods.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>WSM</td>
<td>0.0290</td>
<td>45.91</td>
<td>1993</td>
<td>846</td>
<td>0.4</td>
<td>9.7</td>
<td>307 144</td>
</tr>
<tr>
<td>ERPM</td>
<td>0.0025</td>
<td>2.19</td>
<td>834</td>
<td>996</td>
<td>1.2</td>
<td>29.0</td>
<td>341 911</td>
</tr>
<tr>
<td>PSM</td>
<td>0.0038</td>
<td>1.39</td>
<td>514</td>
<td>1357</td>
<td>2.6</td>
<td>19.5</td>
<td>435 853</td>
</tr>
</tbody>
</table>

strongly on the chosen value for the shifting vector $\tilde{d}_{ERPM}$, see Figure 2.6(a)–(c). In Figure 2.9(c),(d), we observe that the quality of the approximation by the PSM depends on the chosen initial values for the PSPs. If the initial values are chosen with the same scheme as for the WSM and the ERPM, there are some points missing in the top left part of the Pareto front. The reason is that the PSPs corresponding to the missing points actually converged to the tip point of the lower right part of the Pareto front, which is a local minimizer for these problems. Only by setting the initial values of the PSPs as the upper left tip point of the upper left part, i.e., as the minimizer of $\tilde{J}_2$, we obtain an approximation of the entire Pareto front, as is shown in Figure 2.9(c). However, even in this case, there is a small hole in the Pareto front due to the same problem concerning local vs. global optimization. All reference points, for which this happens, are colored in pink in Figure 2.8(b). This underlines the sensitivity of finding the global minimizers of the PSPs in practice. Moreover, the gap between the sufficient and necessary conditions for computing the reference points for the PSM (cf. Corollary 1.7.53) leads to the fact that solving the PSPs for some reference points does not give us new Pareto optimal points. These reference points are shown in yellow in Figure 2.8(b).

The quality criteria from Section 1.4.2 confirm our observations, see Table 2.3. On the one hand, in terms of coverage, the ERPM is slightly better than the PSM due to the small hole in the approximation of the Pareto front. However, both values are in the order of the grid size. On the other hand, the PSM produces a more uniform approximation. Since the WSM cannot approximate the 'interior' of the upper left part of the Pareto front, the coverage is around ten times worse than the ones for the ERPM and the PSM. This is also the reason for the bad uniformity, since the computed solutions to many WSPs cluster in the lower right part close to the tip point, instead of the interior of the upper left part. For both the WSM and the ERPM, we observe that the maximal number of iterations was reached for quite a lot of WSPs and ERPPs. Consequently, the average iteration number is much higher than for the linear-quadratic MOCP from Section 2.1.4. In contrast to this, the average iteration number for the PSM actually decreases slightly in comparison to the one for the MOCP. One reason for this is that the augmented Lagrangian penalty algorithm does not exploit the linear-quadratic structure of the MOCP, since the subproblems of the augmented Lagrangian penalty algorithm are in general non-convex even in this case. Moreover, more approximation points are chosen in this example here, so that the initial value of the optimization problem is closer to the minimizer in comparison to the linear-quadratic MOCP. Consequently, while the WSM is still the fastest in terms of computational time, the difference between the PSM and the WSM is by far not as high as for the linear-quadratic problem.
Similar to the linear-quadratic MOCP, most of the computational time (around 67\%) was spent solving the state and adjoint equations as well as their derivatives. However, since the computational effort for solving elliptic equations is not as high as for solving parabolic equations, other parts of the implementation contribute noticeably to the computational time as well. In particular, the code has not been optimized with respect to performance, in order to make it easier to handle and adapt. Nevertheless, in Chapter 3, we show that model-order reduction can also be applied to this problem in order to reduce the computational costs.
Model-Order Reduction for PDE-Constrained Multiobjective Optimization

The numerical experiments in Chapter 2 show that the computational costs for solving PDE-constrained MOPs numerically are high for two reasons: First, the number of scalar PDE-constrained optimization problems that have to be solved in each of the scalarization methods is high and increases exponentially with the number of cost functions. Second, for solving any of the occurring optimization problems, the state and adjoint equations have to be solved numerous times. This involves the solution of large-scale linear equation systems, which depend on the dimension of the FE discretization of the PDEs.

While the number of optimization problems can in general not be decreased, the solution of the state and adjoint equations can be sped up by using model-order reduction [BGTQ+21, SvdVR08]. The idea of model-order reduction is to construct a low-dimensional surrogate model of the PDE, which is based on previously computed data from the high-dimensional FE model of the PDE. To this end, a so-called reduced-order space is computed, which is supposed to contain the most prominent characteristics of solutions of the PDE. Solving the PDE in the reduced-order space means solving equation systems of a much smaller dimension compared to the FE model. This allows for a speed-up of several orders of magnitude.

Two well-known model-order reduction techniques are the proper orthogonal decomposition (POD) method (cf. [HLBR12, KV99, RP03, Sir87]) and the reduced basis (RB) method (cf. [BMP+12, HSW13, HRS16, QMN16, RHP07]). Both methods are introduced in Appendix A.

In this chapter, we show how these two methods can be applied to the PDE-constrained MOPs from the Sections 2.1 and 2.2, respectively. Since using model-order reduction introduces an approximation error between the full-order and the reduced-order solutions, one is always confronted with finding a good balance between keeping the approximation error reasonably small and still gaining a sufficient computational speed-up.

To this end, we show in Section 3.1 how a-posteriori error estimates in combination with adapting the POD basis can be used to guarantee a desired accuracy of the solutions of the reduced-order problems in the case of linear-quadratic MOCPs. To show the efficiency of the resulting algorithm in a practical example, it is applied to the MOCP from Section 2.1.
The generalization of the a-posteriori error estimates to non-convex problems is possible in many cases, but implies additional numerical effort, cf. [DH15, ITV16, KTV13, Tre17]. Instead of pursuing this approach, we introduce and discuss a Trust-Region RB algorithm in Section 3.2, which was recently published in [BKM+20] and to which the author made substantial contributions. For this algorithm, we can show the convergence of the iteration sequence to a first-order critical point of the full-order optimization problem. Numerical experiments for the MPOP from Section 2.2 verify this theoretical result.

3.1 POD-Based Linear-Quadratic Multiobjective Optimal Control

The application of the POD method to linear-quadratic optimal control problems has already been studied extensively in the literature, cf., e.g., [GNV17, HV08, KG14b, TV09]. In [TV09], the authors develop both a-priori and a-posteriori error estimates for the POD method applied to the weighted-sum of two objective functions. A-posteriori error estimates, with which the approximation error can be estimated after having computed the reduced-order solution, are especially interesting in this context, since, in general, it is not possible to obtain any a-priori convergence rates for the approximation error. Consequently, they are often used to guarantee that the error between the reduced-order and the full-order solution is sufficiently small in a practical algorithm (cf., e.g., [BBV17, GV17, KG14b, TV09]). For instance, in [TV09], the a-posteriori error estimate is used to develop a simple algorithm extending the POD basis if a given error tolerance is exceeded, cf. [TV09, Algorithm 1]. Similar approaches are used in, e.g., [GNV17, GV17, ITV16, IUV17, KG14b]. Although all of these results were only obtained in the viewpoint of scalar optimization problems, i.e., for one fixed weighted-sum of the objective functions, they can be easily transferred to multiobjective optimization by the WSM.

In [Ban17, BBV17, Bee19, Mak18, Spu19], the application of POD to linear-quadratic MOCPs by the ERPM is analyzed. In particular, similar a-priori results and a-posteriori error estimates as the ones in [TV09] are shown and allow for an error control in this case as well. In [BMV19], these were then combined with different POD update schemes to ensure a desired error tolerance. However, for the PSM there are no such results available in the literature to the best of the author’s knowledge.

Thus, in this section, we will first summarize existing a-posteriori and a-priori estimates for the WSP and the ERPP before we show that both an a-priori and an a-posteriori error estimate can also be developed for the PSP. For all three scalarization methods, these a-posteriori error estimates lay the basis for a numerical algorithm guaranteeing a desired exactness of the reduced-order solutions, cf. Algorithm 7. By conducting numerical experiments on the MOCP from Section 2.1.1, we verify that Algorithm 7 allows for a precise error control while still sufficiently reducing the computational time.
3.1 POD-Based Linear-Quadratic Multiobjective Optimal Control

At this point, we refer the reader to Appendix A.2, where the POD method as well as its application to linear evolution equations and linear-quadratic optimal control problems is introduced.

Now recall the situation of Section 2.1.3 and assume that we are given a POD basis \( \{ \psi_i \}_{i=1}^\ell \) of rank \( \ell \). With this basis, we can obtain reduced-order approximations \( \hat{J}_1^\ell, \ldots, \hat{J}_k^\ell \) of the cost functions \( J_1, \ldots, J_k \), see Appendix A.2.3. Then the reduced-order linear-quadratic MOCP reads

\[
\min_{u \in U_{ad}} \hat{J}^\ell(u) = \min_{u \in U_{ad}} \left( \begin{array}{c} \hat{J}_1^\ell(u) \\ \vdots \\ \hat{J}_k^\ell(u) \end{array} \right),
\] (POD-MOCP)

For the cost functions of the model problem from Section 2.1, we can show the following result, which is essential for the proofs of the following error estimates.

**Lemma 3.1.1.** The cost functions \( J_1, \ldots, J_k \) and their reduced-order approximations \( \hat{J}_1^\ell, \ldots, \hat{J}_k^\ell \) are twice continuously Fréchet differentiable, quadratic and it holds

\[
\langle \nabla^2 J_i(u) h, h \rangle_U \geq \sigma_i^{(i)} \| h \|_U^2,
\] (3.1.1)

\[
\langle \nabla^2 \hat{J}_i^\ell(u) h, h \rangle_U \geq \sigma_i^{(i)} \| h \|_U^2
\] (3.1.2)

for all \( i \in \{1, \ldots, k\} \) and all \( u, h \in U \).

**Proof.** The statements for the full-order cost functions follow from Lemma 2.1.12 and Corollary 2.1.19. For the reduced-order cost functions, the statements can be concluded from Lemma A.2.15. \(\square\)

### 3.1.1 Error Analysis for the Weighted-Sum Method

Let us start by stating the definition of the reduced-order WSP with respect to a weight \( \alpha \in \Delta_k \):

\[
\min_{u \in U_{ad}} (\hat{J}^\ell)^\alpha(u) := \min_{u \in U_{ad}} \sum_{i=1}^k \alpha_i \hat{J}_i^\ell(u).
\] (WSP\(^\ell(\alpha)\))

In Lemma 1.5.7, we have shown a first-order optimality condition for the full-order problem (WSP\((\alpha)\)). This condition can be analogously formulated for the reduced-order problem (WSP\(^\ell(\alpha)\)).

**Lemma 3.1.2.** Let \( \alpha \in \Delta_k \) be arbitrary. Given a POD subspace \( V^\ell \), a necessary and sufficient first-order optimality condition for a solution \( \bar{u}^\ell \in U_{ad} \) of (WSP\(^\ell(\alpha)\)) reads

\[
\sum_{i=1}^k \alpha_i \nabla \hat{J}_i^\ell(\bar{u}^\ell), u - \bar{u}^\ell \rangle_U \geq 0 \quad \text{for all } u \in U_{ad}.
\] (3.1.3)

**Proof.** This follows with the same arguments as Lemma 1.5.7. \(\square\)
3.1.1.1 A-Priori Error Analysis

In this section, we present an a-priori error analysis for the WSP. We start with the following theorem, which is a slight generalization of [HV08, Theorem 4.7].

**Theorem 3.1.3.** Let $V^\ell$ be a given POD subspace. Furthermore, let $\alpha \in \Delta_k$ be arbitrary such that $\sum_{i=1}^k \alpha_i \sigma_i(\bar{u}) > 0$ holds. Denote by $\bar{u}$ and $\bar{u}^\ell$ solutions of (WSP($\alpha$)) and (WSP$^\ell$($\alpha$)), respectively. Then we have

$$\|\bar{u} - \bar{u}^\ell\|_U \leq \left(\sum_{i=1}^k \alpha_i \sigma_i(\bar{u})\right)^{-1} \sum_{i=1}^k \alpha_i \left\|\nabla \hat{J}_i(\bar{u}) - \nabla \hat{J}_i(\bar{u}^\ell)\right\|_U =: \nu_{WSM}(\bar{u}).$$

(3.1.4)

**Proof.** The proof can be obtained by using the optimality conditions (1.5.1) and (3.1.3) as well as the properties shown in Lemma 3.1.1. It is only a slight generalization of the proof of [HV08, Theorem 4.7], to which we refer here, by allowing $k$ cost functions instead of 2, and by working directly with the quadratic nature of the cost functions. □

**Remark 3.1.4.** In Theorem 3.1.3, we can switch the role of $\hat{J}$ and $\bar{u}$ with $\hat{J}^\ell$ and $\bar{u}^\ell$ and obtain the estimate

$$\|\bar{u} - \bar{u}^\ell\|_U \leq \left(\sum_{i=1}^k \alpha_i \sigma_i(\bar{u}^\ell)\right)^{-1} \sum_{i=1}^k \alpha_i \left\|\nabla \hat{J}_i(\bar{u}^\ell) - \nabla \hat{J}_i(\bar{u})\right\|_U =: \nu_{WSM}(\bar{u}^\ell).$$

(3.1.5)

Note that the right-hand side of (3.1.5) only depends on $\bar{u}^\ell$ and not on $\bar{u}$. This estimate is later used in Section 3.1.4.2 in the context of determining the number of required basis functions. □

Similarly to [TV09, Theorem 4.9] and [GV17, Theorem 4.17], we can now show the convergence of the sequence of reduced-order solutions $(\bar{u}^\ell)_{\ell \in \mathbb{N}}$ to the full-order solution $\bar{u}$.

**Corollary 3.1.5.** Let $(V^\ell)_{\ell \in \mathbb{N}}$ be a family of POD subspaces. Furthermore, let $\alpha \in \Delta_k$ be arbitrary such that $\sum_{i=1}^k \alpha_i \sigma_i(\bar{u}) > 0$ holds. Denote by $\bar{u}$ and $\bar{u}^\ell$ solutions of (WSP($\alpha$)) and (WSP$^\ell$($\alpha$)) for all $\ell \in \mathbb{N}$, respectively. If $S\bar{u}, A\bar{u}, \bar{p} \in H^1(0,T;V)$, then it holds

$$\lim_{\ell \to \infty} \|\bar{u}^\ell - \bar{u}\|_U = 0.$$  

(3.1.6)

Moreover, in this case, we have

$$\lim_{\ell \to \infty} \hat{J}_i(\bar{u}^\ell) = \hat{J}_i(\bar{u})$$

(3.1.7)

and

$$\lim_{\ell \to \infty} \nabla \hat{J}_i(\bar{u}^\ell) = \nabla \hat{J}_i(\bar{u})$$

(3.1.8)

for all $i \in \{1, \ldots, k\}$.

**Proof.** The convergence result (3.1.6) follows from Theorem 3.1.3 and Corollary A.2.16. Now (3.1.7) and (3.1.8) can be concluded similarly to [Ban17, Corollary 5.42]. □
### 3.1 POD-Based Linear-Quadratic Multiobjective Optimal Control

#### 3.1.1.2 A-Posteriori Error Analysis

In the previous section, we have seen that the solution $\bar{u}^\ell$ of $(WSP^\ell(\alpha))$ converges to the solution $\bar{u}$ of $(WSP(\alpha))$. However, for a general family of POD subspaces $(V^\ell)_{\ell \in \mathbb{N}}$, it is not possible to prove any convergence rates so that we cannot quantify the error $\|\bar{u} - \bar{u}^\ell\|_U$ a-priori. This is the reason why a-posteriori error estimates are a popular tool when using model-order reduction. Based on the a-posteriori error estimate presented in [TV09, Section 4.7] and [GV17, Section 4.5], which uses a perturbation method (cf. [DHPY95]), we can show the following result, which is only a slight modification of these results.

**Theorem 3.1.6.** Let $V^\ell$ be a POD subspace and $\alpha \in \Delta_k$ be arbitrary such that $\sum_{i=1}^k \alpha_i \sigma_i(\bar{u}) > 0$ holds. Denote by $\bar{u}$ solution of $(WSP(\alpha))$ and let $\bar{u}^\ell \in U_{ad}$ be arbitrary. Then we have

$$\|\bar{u} - \bar{u}^\ell\|_U \leq \left( \sum_{i=1}^k \alpha_i \sigma_i(\bar{u}) \right)^{-1} \|\xi_{WSM}^\ell\|_U : = \eta_{WSM}^\ell(\bar{u}^\ell),$$

(3.1.9)

where the perturbation $\xi_{WSM}^\ell = \xi_{WSM}^\ell(\bar{u}^\ell) \in U$ is given by

$$(\xi_{WSM}^\ell)_{i}(t) = \begin{cases} - \min(0, (\nabla \hat{J}^{\alpha}(\bar{u}))_{i}(t)), & \text{a.e. in } \{ s \in (0, T) | \bar{u}^\ell(s) = (u_a)_i(s) \}, \\ - \max(0, (\nabla \hat{J}^{\alpha}(\bar{u}))_{i}(t)), & \text{a.e. in } \{ s \in (0, T) | \bar{u}^\ell(s) = (u_b)_i(s) \}, \\ - (\nabla \hat{J}^{\alpha}(\bar{u}))_{i}(t), & \text{else} \end{cases}$$

for almost all $t \in (0, T)$ and all $i \in \{1, \ldots, m\}$. Moreover, recall that we have $\nabla \hat{J}^{\alpha}(\bar{u}^\ell) = \sum_{i=1}^k \alpha_i \nabla \hat{J}(\bar{u}^\ell)$.

**Proof.** The proof can be obtained by slightly modifying the one of [TV09, Theorem 4.11] or [GV17, Theorem 4.19]. Therefore, we omit a detailed proof here. ☐

**Remark 3.1.7.** In practice, the a-posteriori error estimate (3.1.9) is often evaluated at the solution $\bar{u}^\ell$ of $(WSP^\ell(\alpha))$. Moreover, note that the evaluation involves the computation of the values of the full-order gradients at $\bar{u}^\ell$.

#### 3.1.2 Error Analysis for the Euclidean Reference Point Method

In this section, we carry out the POD error analysis for the ERPM. The results are similar to the ones presented in [Ban17, BBV17, Mak18] and shall only serve as a summary.

We start by introducing the reduced-order ERPP with respect to the reference point $z$, which is given by

$$\min_{u \in U_{ad}} (\hat{J}^\ell)^{ge}(u) := \min_{u \in U_{ad}} \frac{1}{2} \sum_{i=1}^k \left( \hat{J}_i^\ell(u) - z_i \right)^2.$$  

(ERPP$^\ell(z)$)

For the problem $(ERPP^\ell(z))$, we can show the following first-order optimality condition, cf. Lemma 1.6.24. Note that we denote by $Z_{opt,(w)}^\ell$ the set of (weakly) Pareto admissible reference points for (POD-MOCP).
Lemma 3.1.8. Given a POD subspace $V^\ell$ and a reference point $z \in Z_\text{opt,w}^\ell$, a necessary first-order optimality condition for a solution $\bar{u}^\ell \in U_{\text{ad}}$ of $(\text{ERPP}(z))$ reads

$$\left(\sum_{i=1}^{k}(J_i^\ell(\bar{u}^\ell) - z_i)\nabla J_i^\ell(\bar{u}^\ell), u - \bar{u}^\ell\right)_U \geq 0 \quad \text{for all } u \in U_{\text{ad}}. \quad (3.1.10)$$

This condition is sufficient if $\tilde{J}^\ell(\bar{u}^\ell) \geq z$ is satisfied.

Proof. This follows with the same arguments as Lemma 1.6.24.

3.1.2.1 A-Priori Error Analysis

By using the optimality conditions (1.6.4) and (3.1.10) for $(\text{ERPP}(z))$ and $(\text{ERPP}^\ell(z))$, respectively, we can show the following a-priori error estimate. Note that a similar estimate was already proven in [Ban17, Theorem 5.41] and [Mak18, Theorem 5.53]. Since the proof of the estimate here does not contain any new essential features, we omit it at this point and directly refer to the ones of [Ban17, Theorem 5.41] and [Mak18, Theorem 5.53].

Theorem 3.1.9. Let $V^\ell$ be a POD subspace and $z \in Z_\text{opt,w}^\ell \cap Z_\text{opt,w}^\ell$ be arbitrary. Denote by $\bar{u}$ and $\bar{u}^\ell$ solutions of $(\text{ERPP}(z))$ and $(\text{ERPP}^\ell(z))$, respectively, and define

$$a(\bar{u}) := \sum_{i=1}^{k}\frac{\tilde{J}_i(\bar{u}) - z_i}{2}\sigma^{(i)}_{\bar{u}}; \quad b(\bar{u}) := \sum_{i=1}^{k}\left|\tilde{J}_i(\bar{u}) - z_i\right|\left\|\nabla \tilde{J}_i(\bar{u}) - \nabla \tilde{J}_i^{\ell}(\bar{u})\right\|_U,$$

$$c(\bar{u}) := \frac{1}{2}\left\|\tilde{J}^{\ell}(\bar{u}) - \tilde{J}(\bar{u})\right\|^2_2.$$

If $a(\bar{u}) > 0$, then we have

$$\left\|\bar{u} - \bar{u}^\ell\right\|_U \leq \frac{b(\bar{u}) + \sqrt{b(\bar{u})^2 + 4a(\bar{u})c(\bar{u})}}{2a(\bar{u})} = \nu_{\text{ERPM}}(\bar{u}). \quad (3.1.11)$$

Remark 3.1.10. Similarly to Remark 3.1.4, we can switch the role of $\tilde{J}$ and $\bar{u}$ with $\tilde{J}^{\ell}$ and $\bar{u}^\ell$ in Theorem 3.1.9, respectively, to obtain the estimate

$$\left\|\bar{u} - \bar{u}^\ell\right\|_U \leq \frac{b(\bar{u}^\ell) + \sqrt{b(\bar{u}^\ell)^2 + 4a(\bar{u}^\ell)c(\bar{u}^\ell)}}{2a(\bar{u}^\ell)} = \nu_{\text{ERPM}}(\bar{u}^\ell). \quad (3.1.12)$$

Note that $a(\bar{u}^\ell), b(\bar{u}^\ell)$ and $c(\bar{u}^\ell)$ do not depend on $\bar{u}$, but only on $\bar{u}^\ell$.

As for the WSP, we can now show the following a-priori convergence result.

Corollary 3.1.11. Let $(V^\ell)_{\ell \in \mathbb{N}}$ be a family of POD subspaces and $z \in Z_\text{opt,w} \cap \bigcap_{\ell \in \mathbb{N}} Z_\text{opt,w}^\ell$ be arbitrary. Denote by $\bar{u}$ and $\bar{u}^\ell$ solutions of $(\text{ERPP}(z))$ and $(\text{ERPP}^\ell(z))$, respectively. If $S\bar{u}, A\bar{u}, \bar{p} \in H^1(0,T;V)$ and $a(\bar{u}) > 0$, then

$$\lim_{\ell \to \infty} \left\|\bar{u}^\ell - \bar{u}\right\|_U = 0. \quad (3.1.13)$$
Moreover, in this case, we have

\[
\lim_{\ell \to \infty} \hat{J}_i^\ell(\bar{u}^\ell) = \lim_{\ell \to \infty} \hat{J}_i(\bar{u}) = J_i(\bar{u}),
\]

(3.1.14)

\[
\lim_{\ell \to \infty} \nabla \hat{J}_i^\ell(\bar{u}^\ell) = \lim_{\ell \to \infty} \nabla \hat{J}_i(\bar{u}) = \nabla J_i(\bar{u}),
\]

(3.1.15)

for all \( i \in \{1, \ldots, k\} \).

Proof. The convergence result (3.1.13) follows from Theorem 3.1.9, Theorem A.2.10 and Corollary A.2.16. Now (3.1.14) and (3.1.15) can be concluded similarly to [Ban17, Corollary 5.42].

3.1.2.2 A-Posteriori Error Analysis

As for the WSP, the a-posteriori error analysis for the ERPP also makes use of a perturbation approach presented in [DHPY95]. The following result can in a similar form be found in [Ban17, Theorem 3.51].

**Theorem 3.1.12.** Let \( V^\ell \) be a POD subspace and \( z \in Z_{opt,w} \cap Z_{opt,w}^\ell \) be arbitrary. Denote by \( \bar{u} \) a solution of (ERPP(\( z \))) and let \( \bar{u}^\ell \in U_{ad} \) be arbitrary. Then we have

\[
\|\bar{u} - \bar{u}^\ell\|_U \leq \left( \sum_{i=1}^k \frac{\hat{J}_i(\bar{u}) - z_i}{\sigma_{\ell}^{(i)}} \right)^{-1} \left\| \xi_{\text{ERPM}}^{\ell}(\bar{u}^\ell) \right\|_U =: \eta_{\text{ERPM}}(\bar{u}^\ell),
\]

(3.1.16)

where the perturbation \( \xi_{\text{ERPM}}^{\ell}(\bar{u}^\ell) \in U \) is given by

\[
(\xi_{\text{ERPM}}^{\ell}(t))_i = \begin{cases} 
- \min(0, (\nabla J_{g_{z_i}}(\bar{u}))_i(t)), & \text{a.e. in } \{s \in (0,T) \mid \bar{u}_i(s) = (u_a)_i(s)\}, \\
- \max(0, (\nabla J_{g_{z_i}}(\bar{u}))_i(t)), & \text{a.e. in } \{s \in (0,T) \mid \bar{u}_i(s) = (u_b)_i(s)\}, \\
-(\nabla J_{g_{z_i}}(\bar{u}))_i(t), & \text{else}
\end{cases}
\]

for almost all \( t \in (0,T) \) and all \( i \in \{1, \ldots, m\} \). Moreover, recall that we have

\[
\nabla J_{g_{z_i}}(\bar{u}) = \sum_{i=1}^k (\hat{J}_i(\bar{u}) - z_i) \nabla \hat{J}_i(\bar{u}).
\]

Proof. For a proof we refer to [Ban17, Theorem 3.51].

**Remark 3.1.13.** In practice, the a-posteriori error estimate (3.1.16) is often evaluated at the solution \( \bar{u}^\ell \) of (ERPP(\( z \))). Moreover, note that the evaluation requires the values of the full-order cost functions and their gradients at \( \bar{u}^\ell \).

3.1.3 Error Analysis for the Pascoletti-Serafini Method

Now we tend to the POD error analysis for the PSM. As for the numerical examples in the Sections 2.1.4 and 2.2.4, we choose the target direction \( r = (1, \ldots, 1)^T \in \mathbb{R}^k \). Then the reduced-order PSP for an arbitrary reference point \( z \in \mathbb{R}^k \) reads

\[
\min_{(u,t)} \begin{array}{l}
t \\
\text{s.t. } u \in U_{ad}, \ t \in \mathbb{R},
\end{array} \\
\hat{J}_i^\ell(u) - z_i \leq t, \quad i = 1, \ldots, k.
\]

(PSP(\( z \), \( r \)))
As in the two previous sections, we will make extensive use of the first-order optimality condition for \((\text{PSP}(z, r))\) (cf. Lemma 1.7.15) and \((\text{PSP}^\ell(z, r))\).

**Lemma 3.1.14.** Assume that \(U_{ad}\) contains an interior point and let \(z \in \mathbb{R}^k\) be arbitrary. If \(\left(\bar{u}^\ell, \bar{t}^\ell\right) \in U_{ad} \times \mathbb{R}\) is a global solution of \((\text{PSP}^\ell(z, r))\) for a given POD subspace \(V^\ell\), then there is a Lagrange multiplier \(\lambda^\ell \in \mathbb{R}^k\) such that

\[
\sum_{i=1}^k \lambda_i^\ell = 1, \tag{3.1.17a}
\]

\[
\langle \sum_{i=1}^k \lambda_i^\ell \nabla \hat{J}_i^\ell(\bar{u}^\ell), u - \bar{u}^\ell \rangle \geq 0, \quad \text{for all } u \in U_{ad}, \tag{3.1.17b}
\]

\[
\hat{J}_i^\ell(\bar{u}^\ell) - z_i \leq \bar{t}^\ell, \quad \text{for all } i \in \{1, \ldots, k\}, \tag{3.1.17c}
\]

\[
\lambda_i^\ell \left( \hat{J}_i^\ell(\bar{u}^\ell) - z_i - \bar{t}^\ell \right) = 0, \quad \text{for all } i \in \{1, \ldots, k\} \tag{3.1.17d}
\]

is satisfied.

**Proof.** This follows with the same arguments as Lemma 1.7.15. \(\square\)

**Remark 3.1.15.** Note that in the Lemmata 1.7.15 and 3.1.14, the set \(U_{ad}\) has to contain an interior point. Therefore, the optimality conditions (1.7.2) and (3.1.17) are in principle not applicable to the linear-quadratic MOCP from Section 2.1, since any admissible set of the form \(U_{ad} = [u_a, u_b] \subseteq L^2(0, T; \mathbb{R}^m) = U\) with \(u_a, u_b \in L^\infty(0, T; \mathbb{R}^m)\) does not contain any interior point. However, in our numerical implementation of this problem, the control space \(U = L^2(0, T; \mathbb{R}^m)\) is approximated by a finite-dimensional subspace. In that case, the discretized set \(U_{ad}^d = [u_a^d, u_b^d]\) is finite-dimensional as well and, thus, contains an interior point as long as \(u_a^d < u_b^d\) is satisfied. \(\diamond\)

### 3.1.3.1 A-Priori Error Analysis

Similar to the WSP and the ERPP, it is also possible to prove an a-priori estimate for the error \(\|\bar{u}^\ell - \bar{u}\|_U\) for the PSP. Note that the structure of the PSP is reflected in the a-priori error estimate, since it contains the maximal approximation errors of the cost functions and their gradients.

**Theorem 3.1.16.** Assume that \(U_{ad}\) contains an interior point and let \(V^\ell\) be a POD subspace. Let \(z \in \mathbb{R}^k\) be arbitrary and denote by \((\bar{u}, \bar{t})\) and \((\bar{u}^\ell, \bar{t}^\ell)\) global solutions of \((\text{PSP}(z, r))\) and \((\text{PSP}^\ell(z, r))\) with corresponding Lagrange multipliers \(\lambda\) and \(\lambda^\ell\), respectively. Define

\[
a := a(\lambda) := \sum_{i=1}^k \frac{\lambda_i}{2} \sigma^{(i)}_{U}, \quad b := b(\bar{u}) := \max_{i \in \{1, \ldots, k\}} \left\| \nabla \hat{J}_i(\bar{u}) - \nabla \hat{J}_i^\ell(\bar{u}) \right\|_U, \quad c := c(\bar{u}) := 2 \max_{i \in \{1, \ldots, k\}} \left| \hat{J}_i(\bar{u}) - \hat{J}_i^\ell(\bar{u}) \right|.
\]

If \(a > 0\), then we have

\[
\|\bar{u}^\ell - \bar{u}\|_U \leq \frac{b + \sqrt{b^2 + 4ac}}{2a} =: \nu_{PSM}(\bar{u}). \tag{3.1.18}
\]
Proof. Plugging \( u = \bar{u}^\ell \) in (1.7.2b) and \( u = \bar{u} \) in (3.1.17b), adding the two inequalities and using the complementary conditions (1.7.2d) and (3.1.17d) yields

\[
0 \leq \sum_{i=1}^{k} \bar{\lambda}_i \langle \nabla \hat{J}_i(\bar{u}), \bar{u}^\ell - \bar{u} \rangle_\mathcal{U} + \sum_{i=1}^{k} \bar{\lambda}_i^\ell \langle \nabla \hat{J}_i^\ell(\bar{u}^\ell), \bar{u} - \bar{u}^\ell \rangle_\mathcal{U}
\]

\[
= \sum_{i=1}^{k} \bar{\lambda}_i \left( \hat{J}_i(\bar{u}) + \langle \nabla \hat{J}_i(\bar{u}), \bar{u}^\ell - \bar{u} \rangle_\mathcal{U} - z_i - \ell \right)
\]

\[
+ \sum_{i=1}^{k} \bar{\lambda}_i^\ell \left( \hat{J}_i^\ell(\bar{u}^\ell) + \langle \nabla \hat{J}_i^\ell(\bar{u}^\ell), \bar{u} - \bar{u}^\ell \rangle_\mathcal{U} - z_i - \ell^\ell \right). \tag{3.1.19}
\]

Since the cost functions are quadratic, we have

\[
\hat{J}_i(\bar{u}) + \langle \nabla \hat{J}_i(\bar{u}), \bar{u}^\ell - \bar{u} \rangle_\mathcal{U} = \hat{J}_i(\bar{u}^\ell) - \frac{1}{2} \langle \nabla^2 \hat{J}_i(\bar{u})(\bar{u}^\ell - \bar{u}), \bar{u}^\ell - \bar{u} \rangle_\mathcal{U}, \tag{3.1.20}
\]

\[
\hat{J}_i^\ell(\bar{u}^\ell) + \langle \nabla \hat{J}_i^\ell(\bar{u}^\ell), \bar{u} - \bar{u}^\ell \rangle_\mathcal{U} = \hat{J}_i^\ell(\bar{u}) - \frac{1}{2} \langle \nabla^2 \hat{J}_i^\ell(\bar{u}^\ell)(\bar{u} - \bar{u}^\ell), \bar{u} - \bar{u}^\ell \rangle_\mathcal{U} \tag{3.1.21}
\]

for all \( i \in \{1, \ldots, k\} \). Plugging (3.1.20) and (3.1.21) into (3.1.19) and artificially adding and subtracting \( \bar{\lambda}_i(\hat{J}_i^\ell(\bar{u}^\ell) - \ell^\ell) \) and \( \bar{\lambda}_i^\ell(\hat{J}_i(\bar{u}) - \ell) \) for all \( i \in \{1, \ldots, k\} \), we can conclude

\[
0 \leq \sum_{i=1}^{k} \bar{\lambda}_i \left( \hat{J}_i^\ell(\bar{u}^\ell) - z_i - \ell^\ell - \frac{1}{2} \langle \nabla^2 \hat{J}_i(\bar{u})(\bar{u}^\ell - \bar{u}), \bar{u}^\ell - \bar{u} \rangle_\mathcal{U} + \hat{J}_i(\bar{u}^\ell) - \hat{J}_i^\ell(\bar{u}^\ell) \right)
\]

\[
+ \sum_{i=1}^{k} \bar{\lambda}_i^\ell \left( \hat{J}_i(\bar{u}) - z_i - \ell - \frac{1}{2} \langle \nabla^2 \hat{J}_i^\ell(\bar{u}^\ell)(\bar{u} - \bar{u}^\ell), \bar{u} - \bar{u}^\ell \rangle_\mathcal{U} + \hat{J}_i^\ell(\bar{u}) - \hat{J}_i(\bar{u}) \right)
\]

\[
+ (\ell^\ell - \ell) \sum_{i=1}^{k} \left( \bar{\lambda}_i - \bar{\lambda}_i^\ell \right). \tag{3.1.22}
\]

Due to (1.7.2a) and (3.1.17a), it holds \((\ell^\ell - \ell) \sum_{i=1}^{k} (\bar{\lambda}_i - \bar{\lambda}_i^\ell) = 0\). Hence, by using the inequality constraints (1.7.2c) and (3.1.17c), we can further estimate (3.1.22) and obtain

\[
0 \leq \sum_{i=1}^{k} \bar{\lambda}_i \left( \hat{J}_i(\bar{u}^\ell) - \hat{J}_i^\ell(\bar{u}^\ell) - \frac{1}{2} \langle \nabla^2 \hat{J}_i(\bar{u})(\bar{u}^\ell - \bar{u}), \bar{u}^\ell - \bar{u} \rangle_\mathcal{U} \right)
\]

\[
+ \sum_{i=1}^{k} \bar{\lambda}_i^\ell \left( \hat{J}_i^\ell(\bar{u}^\ell) - \hat{J}_i(\bar{u}) - \frac{1}{2} \langle \nabla^2 \hat{J}_i^\ell(\bar{u}^\ell)(\bar{u} - \bar{u}^\ell), \bar{u} - \bar{u}^\ell \rangle_\mathcal{U} \right). \tag{3.1.23}
\]

Using the fact that the cost functions are quadratic once more, a short calculation shows

\[
\hat{J}_i(\bar{u}^\ell) - \hat{J}_i^\ell(\bar{u}^\ell) = \hat{J}_i(\bar{u}) - \hat{J}_i^\ell(\bar{u}) + \langle \nabla \hat{J}_i(\bar{u}) - \hat{J}_i^\ell(\bar{u}), \bar{u}^\ell - \bar{u} \rangle_\mathcal{U}
\]

\[
+ \frac{1}{2} \langle \nabla^2 \hat{J}_i(\bar{u})(\bar{u}^\ell - \bar{u}), \bar{u}^\ell - \bar{u} \rangle_\mathcal{U} - \frac{1}{2} \langle \nabla^2 \hat{J}_i^\ell(\bar{u}^\ell)(\bar{u}^\ell - \bar{u}), \bar{u}^\ell - \bar{u}^\ell \rangle_\mathcal{U}. \tag{3.1.24}
\]
respectively, to obtain the estimate

$$0 \leq \sum_{i=1}^{k} \lambda_i \left( \tilde{J}_i(\bar{u}) - \tilde{J}^f_i(\bar{u}) + \langle \nabla \tilde{J}_i(\bar{u}) - \tilde{J}^f_i(\bar{u}), \bar{u}^{\ell} - \bar{u} \rangle_U - \frac{1}{2} \langle \nabla^2 \tilde{J}_i(\bar{u})(\bar{u}^{\ell} - \bar{u}), \bar{u}^{\ell} - \bar{u} \rangle_U \right)$$

$$+ \sum_{i=1}^{k} \lambda_i \left( \tilde{J}^f_i(\bar{u}) - \tilde{J}_i(\bar{u}) - \frac{1}{2} \langle \nabla^2 \tilde{J}^f_i(\bar{u}^{\ell})(\bar{u} - \bar{u}^{\ell}), \bar{u} - \bar{u}^{\ell} \rangle_U \right).$$

By rearranging and estimating the terms, we end up with

$$\sum_{i=1}^{k} \lambda_i \left( \tilde{J}_i(\bar{u}) - \tilde{J}^f_i(\bar{u}) + \langle \nabla \tilde{J}_i(\bar{u}) - \tilde{J}^f_i(\bar{u}), \bar{u}^{\ell} - \bar{u} \rangle_U \right) + \sum_{i=1}^{k} \lambda_i \left( \tilde{J}^f_i(\bar{u}) - \tilde{J}_i(\bar{u}) \right)$$

$$\leq \sum_{i=1}^{k} \lambda_i (\tilde{J}_i(\bar{u}) - \tilde{J}^f_i(\bar{u})) + \sum_{i=1}^{k} \lambda_i (\nabla \tilde{J}_i(\bar{u}) - \tilde{J}^f_i(\bar{u}), \bar{u}^{\ell} - \bar{u})_U$$

$$\leq \max_{i \in \{1, \ldots, k\}} |\tilde{J}_i(\bar{u}) - \tilde{J}^f_i(\bar{u})| \sum_{i=1}^{k} |\lambda_i - \lambda_i^f| + \max_{i \in \{1, \ldots, k\}} \|\nabla \tilde{J}_i(\bar{u}) - \nabla \tilde{J}^f_i(\bar{u})\|_U \|\bar{u}^{\ell} - \bar{u}\|_U$$

$$\leq 2 \max_{i \in \{1, \ldots, k\}} |\tilde{J}_i(\bar{u}) - \tilde{J}^f_i(\bar{u})| + \max_{i \in \{1, \ldots, k\}} \|\nabla \tilde{J}_i(\bar{u}) - \nabla \tilde{J}^f_i(\bar{u})\|_U \|\bar{u}^{\ell} - \bar{u}\|_U.$$ 

Now we can use the estimate (3.1.1) together with $\lambda_i^f \geq 0$ to obtain

$$\sum_{i=1}^{k} \lambda_i (\tilde{J}_i(\bar{u}) - \tilde{J}^f_i(\bar{u}))$$

$$\leq 2 \max_{i \in \{1, \ldots, k\}} |\tilde{J}_i(\bar{u}) - \tilde{J}^f_i(\bar{u})|$$

$$+ \max_{i \in \{1, \ldots, k\}} \|\nabla \tilde{J}_i(\bar{u}) - \nabla \tilde{J}^f_i(\bar{u})\|_U \|\bar{u}^{\ell} - \bar{u}\|_U.$$ 

Finally, we arrive at the a-priori estimate (3.1.18) by solving this quadratic inequality. \(\square\)

**Remark 3.1.17.** We can again switch the role of $\tilde{J}$ and $\bar{u}$ with $\tilde{J}^f$ and $\bar{u}^f$ in Theorem 3.1.9, respectively, to obtain the estimate

$$\|\bar{u} - \bar{u}^f\|_U \leq \frac{b + \sqrt{b^2 + 4ac}}{2a} = \nu_{PSM}(\bar{u}^f),$$

(3.1.25)

with $a := a(\lambda)$, $b := b(\bar{u})$, and $c := c(\bar{u})$ being independent of $\bar{u}$. \(\diamond\)

**Corollary 3.1.18.** Assume that $\mathcal{U}_{ad}$ contains an interior point and let $(V^f)_{\ell \in \mathbb{N}}$ be a family of POD subspaces. Let $z \in \mathbb{R}^k$ be arbitrary and denote by $(\bar{u}, \ell)$ and $(\bar{u}^f, \ell^f)$ global solutions of (PSP$(z, r)$) and (PSP$^f(z, r)$), respectively. If $a(\lambda) > 0$ and $S\bar{u}, A\bar{u}, p \in H^1(0, T; V)$, then we have

$$\lim_{\ell \to \infty} \|\bar{u}^f - \bar{u}\|_U = 0.$$ 

(3.1.26)
Moreover, in this case, we can conclude
\[
\lim_{\ell \to \infty} J_i^\ell(\bar{u}^\ell) = \lim_{\ell \to \infty} \hat{J}_i(\bar{u}) = \hat{J}_i(\bar{u}), \tag{3.1.27}
\]
\[
\lim_{\ell \to \infty} \nabla J_i^\ell(\bar{u}^\ell) = \lim_{\ell \to \infty} \nabla \hat{J}_i(\bar{u}) = \nabla \hat{J}_i(\bar{u}) \tag{3.1.28}
\]
for all \( i \in \{1, \ldots, k\} \).

Proof. The convergence result (3.1.26) follows from Theorem 3.1.16, Theorem A.2.10 and Corollary A.2.16. Now (3.1.27) and (3.1.28) can be concluded similarly to [Ban17, Corollary 5.42].

### 3.1.3.2 A-Posteriori Error Analysis

As it turns out, the perturbation approach from [DHPY95] can also be used to obtain an a-posteriori error estimate for the PSP.

**Theorem 3.1.19.** Assume that \( U_{ad} \) contains an interior point and let \( V^\ell \) be a POD subspace. Let \( z \in \mathbb{R}^k \) be arbitrary and denote by \((\bar{u}, \bar{t})\) a global solution of \((\text{PSP}(z, r))\) with associated Lagrange multiplier \( \bar{\lambda} \). For any \((\bar{u}^\ell, \bar{\lambda}^\ell) \in U_{ad} \times \Delta_k\) let \( \xi_{PSM}^\ell = \xi_{PSM}^\ell(\bar{u}^\ell) \in U \) be given by

\[
(\xi_{PSM}^\ell)_i(t) = \begin{cases} 
-\min(0, \langle \zeta(\bar{u}^\ell, \bar{\lambda}^\ell) \rangle_i(t)), & \text{a.e. in } \{ s \in (0, T) : \bar{u}^\ell_i(s) = (u_a)_i(s) \}, \\
-\max(0, \langle \zeta(\bar{u}^\ell, \bar{\lambda}^\ell) \rangle_i(t)), & \text{a.e. in } \{ s \in (0, T) : \bar{u}^\ell_i(s) = (u_b)_i(s) \}, \\
-(\zeta(\bar{u}^\ell, \bar{\lambda}^\ell))_i(t), & \text{else}
\end{cases}
\]

for almost all \( t \in (0, T) \) and all \( i \in \{1, \ldots, m\} \), where \( \zeta(\bar{u}^\ell, \bar{\lambda}^\ell) := \sum_{i=1}^{k} \bar{\lambda}_i^\ell \nabla \bar{J}_i(\bar{u}^\ell) \). Then it holds

\[
\left\| \bar{u}^\ell - \bar{u} \right\|_U \leq \eta_{PSM}^\ell(\bar{u}^\ell) := \frac{b + \sqrt{b^2 + 4ac}}{2a} \tag{3.1.29}
\]

with

\[
a := \sum_{i=1}^{k} \frac{\bar{\lambda}_i}{2} \sigma^{(i)}_U, \quad b := \left\| \xi_{PSM}^\ell \right\|_U, \quad c := \max_{i \in \{1, \ldots, k\}} \left( \hat{J}_i(\bar{u}) - z_i \right) - \sum_{i=1}^{k} \bar{\lambda}_i^\ell \left( \hat{J}_i(\bar{u}^\ell) - z_i \right).
\]

Proof. By Lemma 1.7.15, we know that the tupel \((\bar{u}, \bar{t})\) satisfies the optimality condition (1.7.2). From the choice of \( \xi_{PSM}^\ell \), it can be immediately concluded that

\[
\sum_{i=1}^{k} \langle \bar{\lambda}_i \nabla \hat{J}_i(\bar{u}) + \xi_{PSM}^\ell, u - \bar{u}^\ell \rangle_U \geq 0 \tag{3.1.30}
\]

holds for all \( u \in U_{ad} \). Choosing \( u = \bar{u}^\ell \) in (1.7.2b) and \( u = \bar{u} \) in (3.1.30) and adding the two inequalities yields

\[
0 \leq \sum_{i=1}^{k} \bar{\lambda}_i \langle \nabla \hat{J}_i(\bar{u}), \bar{u}^\ell - \bar{u} \rangle_U + \sum_{i=1}^{k} \bar{\lambda}_i^\ell \langle \nabla \hat{J}_i(\bar{u}^\ell), \bar{u} - \bar{u}^\ell \rangle_U + \langle \xi_{PSM}^\ell, \bar{u} - \bar{u}^\ell \rangle_U.
\]
Using the complementary condition (1.7.2d) and artificially adding and subtracting $\bar{\lambda}_i^\ell (\hat{J}_i(\bar{u}^\ell) - z_i)$ for all $i \in \{1, \ldots, k\}$, we get

\[
0 \leq \sum_{i=1}^{k} \bar{\lambda}_i \left( \hat{J}_i(\bar{u}) + \langle \nabla \hat{J}_i(\bar{u}), \bar{u}^\ell - \bar{u} \rangle_{\mathbb{U}} - z_i - \bar{t} \right) + \sum_{i=1}^{k} \bar{\lambda}_i^\ell \left( \hat{J}_i(\bar{u}^\ell) + \langle \nabla \hat{J}_i(\bar{u}^\ell), \bar{u}^\ell - \bar{u} \rangle_{\mathbb{U}} - z_i \right) - \sum_{i=1}^{k} \bar{\lambda}_i \left( \hat{J}_i(\bar{u}) - z_i \right) - \sum_{i=1}^{k} \bar{\lambda}_i^\ell \left( \hat{J}_i(\bar{u}^\ell) - z_i \right) + \langle \xi_{SM}^\ell, \bar{u} - \bar{u}^\ell \rangle_{\mathbb{U}}.
\] (3.1.31)

By plugging (3.1.20) twice — once in the original version and once with switched roles of $\bar{u}$ and $\bar{u}^\ell$ — into (3.1.31), rearranging the terms and using the Cauchy-Schwarz inequality, we obtain

\[
\sum_{i=1}^{k} \frac{\bar{\lambda}_i}{2} \langle \nabla^2 \hat{J}_i(\bar{u}^\ell)(\bar{u}^\ell - \bar{u}), \bar{u}^\ell - \bar{u} \rangle_{\mathbb{U}} + \sum_{i=1}^{k} \frac{\bar{\lambda}_i^\ell}{2} \langle \nabla^2 \hat{J}_i(\bar{u}^\ell)(\bar{u} - \bar{u}^\ell), \bar{u} - \bar{u}^\ell \rangle_{\mathbb{U}} 
\leq \sum_{i=1}^{k} \bar{\lambda}_i \left( \hat{J}_i(\bar{u}^\ell) - z_i - \bar{t} \right) + \sum_{i=1}^{k} \bar{\lambda}_i^\ell \left( \hat{J}_i(\bar{u}) - z_i \right) - \sum_{i=1}^{k} \bar{\lambda}_i \left( \hat{J}_i(\bar{u}) - z_i \right) - \sum_{i=1}^{k} \bar{\lambda}_i^\ell \left( \hat{J}_i(\bar{u}^\ell) - z_i \right) + \left\| \xi_{SM}^\ell \right\|_{\mathbb{U}} \left\| \bar{u} - \bar{u}^\ell \right\|_{\mathbb{U}}.
\] (3.1.32)

Due to (1.7.2a) and $\bar{\lambda}_i^\ell \in \Delta_k$, we have $\bar{t} \sum_{i=1}^{k} (\bar{\lambda}_i^\ell - \bar{\lambda}_i) = 0$. Thus, for the first line of the right-hand side of the inequality (3.1.32), we can show

\[
\sum_{i=1}^{k} \bar{\lambda}_i \left( \hat{J}_i(\bar{u}^\ell) - z_i - \bar{t} \right) + \sum_{i=1}^{k} \bar{\lambda}_i^\ell \left( \hat{J}_i(\bar{u}) - z_i \right) - \sum_{i=1}^{k} \bar{\lambda}_i \left( \hat{J}_i(\bar{u}) - z_i \right) - \sum_{i=1}^{k} \bar{\lambda}_i^\ell \left( \hat{J}_i(\bar{u}^\ell) - z_i \right) = \sum_{i=1}^{k} \bar{\lambda}_i \left( \hat{J}_i(\bar{u}^\ell) - z_i - \bar{t} \right) + \sum_{i=1}^{k} \bar{\lambda}_i^\ell \left( \hat{J}_i(\bar{u}) - z_i - \bar{t} \right) - \sum_{i=1}^{k} \bar{\lambda}_i \left( \hat{J}_i(\bar{u}) - z_i - \bar{t} \right) - \sum_{i=1}^{k} \bar{\lambda}_i^\ell \left( \hat{J}_i(\bar{u}^\ell) - z_i - \bar{t} \right) + \bar{t} \sum_{i=1}^{k} (\bar{\lambda}_i^\ell - \bar{\lambda}_i).
\]

\[
= \sum_{i=1}^{k} \bar{\lambda}_i \left( \hat{J}_i(\bar{u}^\ell) - z_i \right) + \sum_{i=1}^{k} \bar{\lambda}_i^\ell \left( \hat{J}_i(\bar{u}) - z_i - \bar{t} \right) - \sum_{i=1}^{k} \bar{\lambda}_i^\ell \left( \hat{J}_i(\bar{u}^\ell) - z_i \right).
\] (3.1.33)
Furthermore, by (1.7.2c), we have $\tilde{J}_i(\tilde{u}) - z_i - \tilde{t} \leq 0$ for all $i \in \{1, \ldots, k\}$. Using this in (3.1.33) yields

$$
\sum_{i=1}^{k} \bar{\lambda}_i \left( J_i(\tilde{u}^\ell) - z_i \right) + \sum_{i=1}^{k} \bar{\lambda}_i \left( J_i(\tilde{u}) - z_i \right) - \sum_{i=1}^{k} \bar{\lambda}_i \left( J_i(\tilde{u}_\ell^\ell) - z_i \right) 
\leq \sum_{i=1}^{k} \bar{\lambda}_i \left( J_i(\tilde{u}^\ell) - z_i \right) - \sum_{i=1}^{k} \bar{\lambda}_i \left( J_i(\tilde{u}_\ell^\ell) - z_i \right)
$$

(3.1.34)

$$
\leq \max_{i \in \{1, \ldots, k\}} \left( J_i(\tilde{u}^\ell) - z_i \right) \sum_{i=1}^{k} \bar{\lambda}_i - \sum_{i=1}^{k} \bar{\lambda}_i \left( J_i(\tilde{u}^\ell) - z_i \right)
= \max_{i \in \{1, \ldots, k\}} \left( J_i(\tilde{u}^\ell) - z_i \right) - \sum_{i=1}^{k} \bar{\lambda}_i \left( J_i(\tilde{u}^\ell) - z_i \right).
$$

(3.1.35)

By combining (3.1.32) and (3.1.35) as well as using the estimate (3.1.1) and $\bar{\lambda} \geq 0$, we can conclude

$$
\sum_{i=1}^{k} \frac{\bar{\lambda}_i}{2} \sigma_{i1}^{(i)} \left\| \tilde{u} - \tilde{u}_\ell^\ell \right\|_{\tilde{u}}^2 \leq \max_{i \in \{1, \ldots, k\}} \left( J_i(\tilde{u}^\ell) - z_i \right) - \sum_{i=1}^{k} \bar{\lambda}_i \left( J_i(\tilde{u}_\ell^\ell) - z_i \right) + \left\| \tilde{\xi}_{PSM}^\ell \right\|_{\tilde{u}} \left\| \tilde{u} - \tilde{u}_\ell^\ell \right\|_{\tilde{u}}.
$$

Finally, we obtain the estimate (3.1.29) by solving this quadratic inequality. \qed

**Remark 3.1.20.**

(i) In the situation of Theorem 3.1.19, one usually chooses a solution $(\tilde{u}_\ell^\ell, \tilde{t}_\ell^\ell)$ of the reduced-order problem (PSP($z, r$)) together with its corresponding Lagrange multiplier $\bar{\lambda}^\ell$. However, the choice of $\bar{\lambda}^\ell$ is not coupled to the choice of $\tilde{u}_\ell^\ell$ in Theorem 3.1.19, so that one might choose $\bar{\lambda}^\ell \in \Delta_k$ in such a way that the estimate (3.1.29) is as small as possible.

(ii) For the computation of the a-posteriori error estimate (3.1.29), the full-order cost functions and their gradients need to be evaluated at $\tilde{u}_\ell^\ell$.

(iii) If we choose a solution $(\tilde{u}_\ell^\ell, \tilde{t}_\ell^\ell)$ of the reduced-order problem (PSP($z, r$)) with the corresponding Lagrange multiplier $\bar{\lambda}^\ell$ for evaluating (3.1.29), we can trace back the term $c$ in the a-posteriori error estimate to the errors in the cost functions $J_i(\tilde{u}_\ell^\ell) - J_i(\tilde{u})$ ($i \in \{1, \ldots, k\}$). Indeed, starting from (3.1.34), we can proceed as follows

$$
\sum_{i=1}^{k} \bar{\lambda}_i \left( J_i(\tilde{u}_\ell^\ell) - z_i \right) - \sum_{i=1}^{k} \bar{\lambda}_i \left( J_i(\tilde{u}_\ell^\ell) - z_i \right)
= \sum_{i=1}^{k} \bar{\lambda}_i \left( J_i(\tilde{u}_\ell^\ell) - z_i - \tilde{t}_\ell^\ell \right) - \sum_{i=1}^{k} \bar{\lambda}_i \left( J_i(\tilde{u}_\ell^\ell) - z_i - \tilde{t}_\ell^\ell \right),
$$
since \( \bar{\lambda}, \bar{\lambda}^\ell \in \Delta_k \). Moreover, we can use (3.1.17c) and (3.1.17d) to conclude

\[
\sum_{i=1}^k \bar{\lambda}_i \left( j_i(\bar{u}^\ell) - z_i - \bar{t}^\ell \right) - \sum_{i=1}^k \bar{\lambda}_i^\ell \left( j_i(\bar{u}^\ell) - z_i - \bar{t}^\ell \right) \\
= \sum_{i=1}^k \bar{\lambda}_i \left( j^\ell_i(\bar{u}^\ell) - z_i - \bar{t}^\ell + j_i(\bar{u}^\ell) - j^\ell_i(\bar{u}^\ell) \right) \\
- \sum_{i=1}^k \bar{\lambda}_i^\ell \left( j^\ell_i(\bar{u}^\ell) - z_i - \bar{t}^\ell + j_i(\bar{u}^\ell) - j^\ell_i(\bar{u}^\ell) \right) \\
\leq \sum_{i=1}^k \bar{\lambda}_i \left( j_i(\bar{u}^\ell) - j^\ell_i(\bar{u}^\ell) \right) - \sum_{i=1}^k \bar{\lambda}_i^\ell \left( j_i(\bar{u}^\ell) - j^\ell_i(\bar{u}^\ell) \right) \\
\leq \max_{i \in \{1, \ldots, k\}} \left( j_i(\bar{u}^\ell) - j^\ell_i(\bar{u}^\ell) \right) - \sum_{i=1}^k \bar{\lambda}_i \left( j_i(\bar{u}^\ell) - j^\ell_i(\bar{u}^\ell) \right) =: c, \tag{3.1.36}
\]

which we can use for the term \( c \) in the a-posteriori error estimate (3.1.29) in this case.

3.1.4 Certified Error Control

In the previous sections, we showed that the approximation error \( \| \bar{u}^\ell - \bar{u} \|_U \) can be quantified a-posteriorily for the WSP, the ERPP and the PSP. When solving (POD-MOCP) numerically by one of the scalarization methods, the reduced-order scalarized optimization problems have to be solved iteratively for different weights or reference points, respectively. The aim is that the error \( \| \bar{u}^\ell - \bar{u} \|_U \) stays below a predefined error tolerance \( \varepsilon_U \) for each of the weights or reference points, respectively. In general, it cannot be expected to reach this goal by using a single POD subspace \( V^\ell \) with a reasonably small dimension \( \ell \), which is generated in the beginning of the multiobjective optimization. Therefore, we need to include the possibility of adapting the POD basis when needed. In this section, we describe an approach for deciding when and how to adapt the POD basis which uses the a-posteriori estimates from the previous sections. Similar strategies were already used for a scalar-valued linear-quadratic optimal control problem consisting of the weighted-sum of two objective functions in [GV17, TV09], and for applying the ERPM to a linear-quadratic MOCP in [BBV17, BMV19, Mak18, Spu19].

3.1.4.1 A-Posteriori Update of the Reduced-Order Model

The approach makes use of the three a-posteriori error estimates (3.1.9), (3.1.16) and (3.1.29). The idea is to first solve the reduced-order scalarized optimization problems (WSP\(^\ell(\alpha)\)), (ERPP\(^\ell(z)\)) or (PSP\(^\ell(z,r)\)) and then evaluate the a-posteriori error estimates \( \eta_{WSP}^\ell(\bar{u}^\ell) \), \( \eta_{ERPM}^\ell(\bar{u}^\ell) \) and \( \eta_{PSM}^\ell(\bar{u}^\ell) \), respectively. If these estimates are larger than the predefined error tolerance \( \varepsilon_U \), we know that the reduced-order model was not accurate enough, so that it needs to be updated. This simple scheme is summarized in Algorithm 5.
Algorithm 5 A-posteriori update of the reduced-order model

Require: Error tolerance $\varepsilon_U$, current weight or reference point $\alpha$ or $z$
1: Solve (WSP$^\ell(\alpha)$), (ERPP$^\ell(z)$) or (PSP$^\ell(z, r)$);
2: Compute a-posteriori error estimate $\eta_{WSM}^\ell(\bar{u}^\ell)$, $\eta_{ERPM}^\ell(\bar{u}^\ell)$ or $\eta_{PSM}^\ell(\bar{u}^\ell)$ by (3.1.9), (3.1.16) or (3.1.29) and denote it by $\eta^\ell(\bar{u}^\ell)$;
3: if $\eta^\ell(\bar{u}^\ell) > \varepsilon_U$ then
4: Update the POD basis;
5: Go to line 1;
6: end if

By showing that the a-posteriori error estimates $\eta_{WSM}^\ell(\bar{u}^\ell)$, $\eta_{ERPM}^\ell(\bar{u}^\ell)$ and $\eta_{PSM}^\ell(\bar{u}^\ell)$ converge to 0 if the dimension $\ell$ of the POD basis tends to infinity, we can prove that Algorithm 5 terminates after finitely many iterations for each of the three scalarization methods if the number of POD basis functions is always increased in line 4 of Algorithm 5. For the WSP and the ERPP, this result can be found in [GV17, Theorem 4.19] and [Ban17, Theorems 5.45 & 5.46], respectively.

Theorem 3.1.21. Let the assumptions of Corollary 3.1.5 be satisfied. For the a-posteriori error estimate (3.1.9), it holds
\[
\| \bar{u} - \bar{u}^\ell \|_u \leq \eta_{WSM}^\ell(\bar{u}^\ell) = \left( \sum_{i=1}^k \alpha_i \sigma_u^{(i)} \right)^{-1} \| \xi_{WSM}^\ell \|_u \to 0 \quad \text{as } \ell \to \infty.
\]

Proof. We refer to [GV17, Theorem 4.19], where an almost identical statement is shown. The proof mainly uses the convergence results (3.1.8) to show that $\| \xi_{WSM}^\ell \|_u \to 0$ as $\ell \to \infty$.

Theorem 3.1.22. Let the assumptions of Corollary 3.1.11 be satisfied. For the a-posteriori error estimate (3.1.16), it holds
\[
\| \bar{u} - \bar{u}^\ell \|_u \leq \eta_{ERPM}^\ell(\bar{u}^\ell) = \left( \sum_{i=1}^k \frac{\dot{J}_i(\bar{u}^\ell) - z_i}{2} \sigma_u^{(i)} \right)^{-1} \| \xi_{ERPM}^\ell \|_u \to 0 \quad \text{as } \ell \to \infty.
\]

Proof. A proof for this statement can be found in [Ban17, Theorems 5.45 & 5.46]. It uses the convergence results (3.1.14) and (3.1.15) to show that
\[
\sum_{i=1}^k \frac{\sigma_u^{(i)} \dot{J}_i(\bar{u}^\ell) - z_i}{2} \to \sum_{i=1}^k \frac{\sigma_u^{(i)} \dot{J}_i(\bar{u}) - z_i}{2} > 0 \quad \text{as } \ell \to \infty,
\]
and $\| \xi_{ERPM}^\ell \|_u \to 0$ as $\ell \to \infty$.

In order to show the convergence of the a-posteriori error estimate for the PSP, we have to additionally assume that the Lagrange multipliers $\tilde{\lambda}^\ell$ corresponding to the solution of (PSP$^\ell(z, r)$) converge to the Lagrange multiplier $\tilde{\lambda}$ corresponding to the solution of (PSP$(z, r)$).
Theorem 3.1.23. Let the assumptions of Corollary 3.1.18 be satisfied and assume additionally that the Lagrange multipliers satisfy $\bar{\lambda}^\ell \to \bar{\lambda}$ as $\ell \to \infty$. Then, for the a-posteriori error estimate (3.1.29), it holds that
\[
\|\bar{u}^\ell - \bar{u}\| \leq \eta_{PSM}(\bar{u}^\ell) = \frac{b + \sqrt{b^2 + 4ac}}{2a} \to 0 \quad \text{as } \ell \to \infty,
\]
where $a = a(\bar{\lambda}^\ell)$ and $b = b(\bar{u}^\ell, \bar{\lambda}^\ell)$ are chosen as in Theorem 3.1.19 and $c = c(\bar{u}^\ell, \bar{\lambda}^\ell)$ is given as in (3.1.36).

Proof. By using (3.1.28) together with $\bar{\lambda}^\ell \to \bar{\lambda}$ (as $\ell \to \infty$), one can show
\[
b(\bar{\lambda}^\ell, \bar{u}^\ell) = \|\xi_{PSM}^\ell\|_U \to 0 \quad \text{as } \ell \to \infty
\]
with the same arguments as it is done for the ERPP in [Ban17, Theorems 5.45]. Moreover, (3.1.27) directly implies that
\[
c(\bar{u}^\ell, \bar{\lambda}^\ell) = \max_{i \in \{1, \ldots, k\}} \left( \hat{J}_i(\bar{u}^\ell) - \hat{J}_i^\ell(\bar{u}^\ell) \right) - \sum_{i=1}^k \bar{\lambda}^\ell_i \left( \hat{J}_i(\bar{u}^\ell) - \hat{J}_i^\ell(\bar{u}^\ell) \right) \to 0 \quad \text{as } \ell \to \infty.
\]
Lastly, by using the convergence $\bar{\lambda}^\ell \to \bar{\lambda}$ as $\ell \to \infty$, we can conclude that
\[
a = a(\bar{\lambda}^\ell) = \sum_{i=1}^k \frac{\bar{\lambda}^\ell_i}{2} a_{ui}^{(i)} \to \sum_{i=1}^k \frac{\bar{\lambda}_i}{2} a_{ui}^{(i)} \neq 0 \quad \text{as } \ell \to \infty.
\]
Altogether, this implies $\eta_{PSM}^\ell(\bar{u}^\ell) \to 0$ as $\ell \to \infty$, which concludes the proof.

3.1.4.2 Adapting the POD basis

One remaining question is how to adapt the POD basis in line 4 of Algorithm 5. The simplest idea is to enlarge the current POD basis by a predefined number of basis functions, as proposed, e.g., in [GV17, TV09]. In view of Theorems 3.1.21 – 3.1.23, this strategy will eventually lead to the a-posteriori error estimate converging to 0, implying that Algorithm 5 terminates in finitely many steps. However, in some situations, it might be beneficial to include new snapshot information into the basis, since the dynamics of the PDE have changed due to the variation in the control $u$. Especially in the situation of MOCPs, this situation occurs easily, since the Pareto set might form a large subset of the admissible set. Thus, one existing idea is to solve the FE state and adjoint equations at the current iterate $\bar{u}^\ell$ and extend the old POD basis with these snapshots, see, e.g., [AH01, Rav02]. To this end, we use a strategy which was the basis for the POD greedy algorithm in [HO08]: The new snapshots are first projected onto the orthogonal complement of the current POD basis. Then a new POD basis of these projected snapshots is computed and its most important basis functions (in terms of the size of the corresponding eigenvalues, cf. Theorem A.2.2) are added to the existing POD basis. Additionally, in some situations, it might be beneficial to prevent the POD basis from growing too much. In these cases, the new snapshots can be used to decide which of
the old basis functions should be removed. This is done by computing the projections of the POD basis onto the new snapshots. In this way, we get an idea about which POD basis functions are not relevant for approximating the new snapshots, and hence, for approximating the current dynamics. Consequently, these basis functions can be removed. The entire resulting procedure is shown in Algorithm 6.

Algorithm 6 Extending the POD basis

<table>
<thead>
<tr>
<th>Require:</th>
<th>New snapshots ( y^1, \ldots, y^p \in L^2(0,T;V) ), current POD basis ( \Psi = { \Psi_1, \ldots, \Psi_\ell } ), tolerance ( \varepsilon_{\text{rem}} \in [0,1] ) for removing old basis functions</th>
</tr>
</thead>
<tbody>
<tr>
<td>1:</td>
<td>Compute the projections ( \mathbf{p}_i^\ell(t) := \langle \Psi_i, y^i(t) \rangle ) for all ( i \in {1, \ldots, \ell} ), all ( j \in {1, \ldots, p} ) and almost all ( t \in (0,T) );</td>
</tr>
<tr>
<td>2:</td>
<td>for ( i = 1 : \ell ) do</td>
</tr>
<tr>
<td>3:</td>
<td>if ( \left( \sum_{j=1}^p \int_0^T</td>
</tr>
<tr>
<td>4:</td>
<td>Remove the basis function ( \Psi_i ) from the POD basis;</td>
</tr>
<tr>
<td>5:</td>
<td>end if</td>
</tr>
<tr>
<td>6:</td>
<td>end for</td>
</tr>
<tr>
<td>7:</td>
<td>Set ( \ell' ) as the new number of POD basis functions;</td>
</tr>
<tr>
<td>8:</td>
<td>Compute the orthogonal complement ( y_\perp^i(t) = y^i(t) - \sum_{i'=1}^\ell \langle \Psi_i, y^i(t) \rangle \Psi_i ) for almost all ( t \in (0,T) ) and all ( j \in {1, \ldots, p} );</td>
</tr>
<tr>
<td>9:</td>
<td>Compute a POD basis ( \tilde{\Psi} = { \tilde{\Psi}<em>1, \ldots, \tilde{\Psi}</em>\ell' } ) for the snapshots ( y_\perp^1, \ldots, y_\perp^p ) by using Theorem A.2.2 and Remark A.2.3;</td>
</tr>
<tr>
<td>10:</td>
<td>Define the new POD basis ( \Psi_{\text{new}} := { \Psi_1, \ldots, \Psi_\ell, \tilde{\Psi}<em>1, \ldots, \tilde{\Psi}</em>\ell' } );</td>
</tr>
</tbody>
</table>

But how many basis functions \( \ell' \) should we add in Algorithm 6? If we add too few basis functions, the error \( \| \tilde{\mathbf{u}}^\ell - \bar{\mathbf{u}}^\ell \|_U \) might still be too large in the next iteration, so that we have to repeat the procedure, which is quite time consuming. On the other hand, adding too many basis functions leads to an unnecessarily large POD basis which results in losing some of the speed-up. We found that a good indicator for the number of basis functions having to be added to the current POD basis is given by the estimates \( \nu_{\text{WSM}}(\bar{\mathbf{u}}^\ell) \), \( \nu_{\text{ERPM}}(\bar{\mathbf{u}}^\ell) \) and \( \nu_{\text{PSM}}(\bar{\mathbf{u}}^\ell) \), cf. (3.1.5), (3.1.12) and (3.1.25), respectively. The idea is the following: If it holds \( \eta(\bar{\mathbf{u}}^\ell) > \varepsilon_U \), the POD basis has to be updated and the optimization problem has to be solved again by using the new basis. For this optimization problem, we can canonically use the initial control \( u_0 = \bar{\mathbf{u}}^\ell \) as a warm start, since we assume that \( \| \bar{\mathbf{u}}^\ell - \bar{\mathbf{u}}^{\text{new}} \|_U \) will be small. In the same way, we expect that the a-priori estimates (3.1.5), (3.1.12) and (3.1.25) evaluated at \( \bar{\mathbf{u}}^\ell \) and \( \bar{\mathbf{u}}^{\text{new}} \), respectively, will also be close to each other, i.e., that \( |\nu(\bar{\mathbf{u}}^\ell) - \nu(\bar{\mathbf{u}}^{\text{new}})\rangle \) is small as well. Thus, if we choose the new POD basis such that \( \nu(\bar{\mathbf{u}}^{\text{new}}) < \kappa_1 \varepsilon_U \) holds for some reasonably chosen \( \kappa_1 \in (0,1) \), the heuristic idea is that then also

\[
\nu(\bar{\mathbf{u}}^{\text{new}}) \leq \nu_{\text{WSM}}(\bar{\mathbf{u}}^\ell) + \nu_{\text{ERPM}}(\bar{\mathbf{u}}^\ell) + \nu_{\text{PSM}}(\bar{\mathbf{u}}^\ell) < \kappa_1 \varepsilon_U + \nu(\bar{\mathbf{u}}^\ell) - \nu(\bar{\mathbf{u}}^{\text{new}}) \leq \varepsilon_U
\]

will hold, which would imply

\[
\| \bar{\mathbf{u}}^{\text{new}} - \bar{\mathbf{u}}^\ell \|_U \leq \nu(\bar{\mathbf{u}}^{\text{new}}) < \varepsilon_U,
\]

see Remarks 3.1.4, 3.1.10 and 3.1.17.
Similarly, this idea can also be used in the beginning of each optimization routine to predict the number of needed POD basis functions, even if the POD basis was not updated in the previous step: Given the initial control $u_0$, the number of POD basis functions is chosen such that $\kappa_2\varepsilon_U < \nu(u_0) < \kappa_1\varepsilon_U$ holds for some $\kappa_1, \kappa_2 \in (0, 1)$ with $\kappa_2 < \kappa_1$. The condition $\kappa_2\varepsilon_U < \nu(u_0)$ is supposed to guarantee that not too many basis functions are chosen, while the condition $\nu(u_0) < \kappa_1\varepsilon_U$ shall heuristically ensure that sufficiently many basis functions are chosen such that $\eta(\bar{u}^f) < \varepsilon_U$ will be satisfied. This procedure is summarized in Algorithm 7.

**Algorithm 7** Certified error control

**Require:** Set of weights $\Delta_k$ or set of reference point or $Z$, error tolerance $\varepsilon_U$, factors for adapting the number of POD basis functions $\kappa_1, \kappa_2$

1. Choose initial control $u_0$;
2. Compute initial POD basis;
3. for all $\alpha \in \Delta_k$ or $z \in Z$ do
4. Choose the number of basis functions such that $\kappa_2\varepsilon_U < \nu(u_0) < \kappa_1\varepsilon_U$ holds;
5. Compute the solution $\bar{u}^f$ of $(WSP^f(\alpha))$, $(ERPP^f(z))$ or $(PSP^f(z, r))$;
6. Set $u_0 = \bar{u}^f$;
7. Evaluate the a-posteriori error estimate $\eta^f(\bar{u}^f)$;
8. if $\eta^f(\bar{u}^f) > \varepsilon_U$ then
9. Update the POD basis according to Algorithm 6 such that $\kappa_2\varepsilon_U < \nu(u_0) < \kappa_1\varepsilon_U$ holds;
10. Go to line 5;
11. end if
12. end for

**Remark 3.1.24.** Note that there are several different approaches for adapting the POD basis in these situations. In [BMV19, Mak18, Spu19], in the context of solving an MOCP by the ERPM, the POD basis is first extended by a fixed number of old POD basis functions. Only if a maximal number of POD basis functions is reached, new snapshot information is taken into account. In [Mec19], the new POD basis is computed by using the new snapshots and some of the old POD basis functions, which are chosen based on how they are aligned to the new snapshots. Another more sophisticated approach might be the use of optimality-system POD (OS-POD) [KV08], where the POD basis is computed with respect to the optimization problem. In [GGV15], OS-POD was used in combination with an a-posteriori error estimate for scalar linear-quadratic optimal control problems. Here it could be used, e.g., by applying a few projected gradient steps on the OS-POD system starting from the current iterate $\bar{u}^f$ and the current POD basis. However, for the linear-quadratic MOCPs considered in this thesis, our approach turns out to produce satisfactory results, so that we did not proceed in this direction.

\diamond
3.1 POD-Based Linear-Quadratic Multiobjective Optimal Control

3.1.5 Numerical Experiments

Let us now revisit the linear-quadratic MOCP from Section 2.1.4 and show how Algorithm 7 performs in practice. In particular, the goal is to gain, on the one hand, a computational speed-up in comparison to solving the full-order problem (cf. Table 2.1), and to maintain, on the other hand, a desired level of accuracy.

The main ingredients of Algorithm 7 are the a-posteriori error estimates for the WSP, the ERPP and the PSP. For a valid and efficient algorithm, it is crucial that these estimates are

1. upper bounds on the error \( \| \bar{u}^\ell - \bar{u} \|_U \) in the numerical realization.

2. tight in the sense that the overestimation of the error \( \| \bar{u}^\ell - \bar{u} \|_U \) is not too large.

For the WSP and the ERPP, these two properties have already been studied in the literature with the result that for both methods the a-posteriori error estimates are indeed quite tight upper bounds of the error \( \| \bar{u}^\ell - \bar{u} \|_U \) (see, e.g., [KTGV18, TV09] for the WSP, and [Ban17, BBV17, Mak18, Spu19] for the ERPP). For the PSP, there are no such results available. In the following section, we will study the overestimation of the a-posteriori error estimates both qualitatively and quantitatively and specifically focus on the differences between the estimates and their influence on the overestimation.

3.1.5.1 Analyzing the A-Posteriori Error Estimates

To begin with, we compare the three a-posteriori error estimates qualitatively and investigate, in particular, how the overestimation of the exact error \( \| \bar{u} - \bar{u}^\ell \|_U \), i.e., the ratio between the a-posteriori error estimate and the exact error, originates for each of them.

By looking at the proof for the a-posteriori error estimate for the WSP in [GV17, Theorem 4.19] and transferring the notation to our framework, we see that the overestimation can only stem from the following three estimates that were used in the proof:

\[
\begin{align*}
\langle \sum_{i=1}^{k} \alpha_i \nabla \hat{J}_i(\bar{u}), \bar{u}^\ell - \bar{u} \rangle_U & \geq 0, \\
\langle \xi^\ell, \bar{u} - \bar{u}^\ell \rangle_U & \leq \| \xi^\ell \|_U \| \bar{u} - \bar{u}^\ell \|_U, \\
\sigma^{(i)}_U \| h \|_U^2 & \leq \langle \nabla^2 \hat{J}_i(u)h, h \rangle_U \quad \text{for all } u \in U.
\end{align*}
\]

Note that it actually holds \( \sum_{i=1}^{k} \alpha_i \nabla \hat{J}_i(\bar{u}) = 0 \) in (3.1.37) if the optimal control \( \bar{u} \) does not have any active point. Thus, only if the active set of the optimal control \( \bar{u} \) is non-empty, the estimate (3.1.37) can possibly yield an overestimation. The overestimation by using the Cauchy-Schwarz inequality in (3.1.38) cannot be quantified in general, but depends on how the vectors \( \xi^\ell \) and \( \bar{u} - \bar{u}^\ell \) are aligned. In practice, it turns out that the estimate (3.1.39) of the coercivity constant has the biggest impact on the overestimation of the error, cf. Figure 3.2(a) and the corresponding discussion.
For the ERPP, the estimates (3.1.38) and (3.1.39) as well as
\[
\sum_{i=1}^{k} (\tilde{J}_i(\bar{u}) - z_i) \nabla \tilde{J}_i(\bar{u}), \bar{u}^\ell - \bar{u})_U \geq 0,
\] (3.1.40)
and the two additional estimates
\[
\sum_{i=1}^{k} \frac{\tilde{J}_i(\bar{u}) - z_i}{2} \sigma_{U}^{(i)} \| \bar{u}^\ell - \bar{u} \|^2_U \geq \sum_{i=1}^{k} \frac{\tilde{J}_i(\bar{u}^\ell) - z_i}{2} \sigma_{U}^{(i)} \| \bar{u}^\ell - \bar{u} \|^2_U,
\] (3.1.41)
are used in the proof of the a-posteriori error estimate, see [Ban17, Theorem 3.51]. Again, the optimality condition (3.1.40) can only cause an overestimation if \( \bar{u} \) has an active point. By using the estimate (3.1.41), we underestimate the coercivity constant by a factor of around two. In practice, this is manifested in the fact that the overestimation of the true error has at least a value of around two, cf. Figures 3.2(c),(d). Apart from that, as for the WSP, the estimate (3.1.39) has also a big influence on the overestimation of the error, cf. Figure 3.2(c) and its discussion. From (3.1.42), we see that we can influence the tightness of the a-posteriori error estimate for \( \| \bar{u} - \bar{u}^\ell \|_U \) by changing the distance between the reference point and the Pareto front, i.e., by changing the value of the shifting vector \( \tilde{d}_{ERPM} \). If the factors \( \tilde{J}_i(\bar{u}^\ell) - z_i \) get larger, the estimate (3.1.42) gets tighter, since the term \( \| \tilde{J}(\bar{u}) - \tilde{J}(\bar{u}^\ell) \|^2_2 \) does not scale with \( \tilde{J}(\bar{u}^\ell) - z \). Thus, the value of the shifting vector \( \tilde{d}_{ERPM} \) should not be chosen too small. Recall that we set \( \tilde{d}_{ERPM} = (1, 1, 1)^T \) in our experiment for the FE method, which we will also use for the experiments here.

Finally, for the PSP, we also use the estimates (3.1.38) and (3.1.39) together with
\[
\sum_{i=1}^{k} (\bar{\lambda}_i \nabla \tilde{J}_i(\bar{u}), \bar{u}^\ell - \bar{u})_U \geq 0,
\] (3.1.43)
\[
\sum_{i=1}^{k} \frac{\bar{\lambda}_i}{2} \nabla^2 \tilde{J}_i(\bar{u}^\ell)(\bar{u} - \bar{u}^\ell), \bar{u} - \bar{u}^\ell)_U \geq 0,
\] (3.1.44)
\[
\sum_{i=1}^{k} (\bar{\lambda}_i - \bar{\lambda}_i^\ell) \left( \tilde{J}_i(\bar{u}^\ell) - z_i \right) \leq \max_{i \in \{1, ..., k\}} \left( \tilde{J}_i(\bar{u}^\ell) - z_i \right) - \sum_{i=1}^{k} \bar{\lambda}_i \left( \tilde{J}_i(\bar{u}^\ell) - z_i \right).
\] (3.1.45)
As for the WSP and the ERPP, the optimality condition (3.1.43) is known to be exactly 0 if the \( \bar{u} \) does not have an active point. As for the ERPP, by using the estimate (3.1.44), we underestimate the coercivity constant by a factor of around 2. Moreover, the estimate (3.1.45) is possibly quite coarse, since we expect the difference \( \bar{\lambda}_i - \bar{\lambda}_i^\ell \) to become small for increasing \( \ell \), which is not captured by the a-posteriori error estimate. This becomes even more obvious if we plug a solution \( \bar{u}^\ell \) of the reduced-order problem (PSP\( ^\ell(z, r) \))
together with the corresponding Lagrange multiplier $\bar{\lambda}^\ell$ into the a-posteriori error estimate, cf. Remark 3.1.20 (iii): Instead of (3.1.45), we use the estimate
\[
\sum_{i=1}^{k} (\lambda_i - \bar{\lambda}_i^\ell) \left( \hat{J}_i(\bar{u}^\ell) - \hat{J}_i^\ell(\bar{u}^\ell) \right) \leq \max_{i \in \{1,\ldots,k\}} \left( \hat{J}_i(\bar{u}^\ell) - \hat{J}_i^\ell(\bar{u}^\ell) \right) - \sum_{i=1}^{k} \bar{\lambda}_i^\ell \left( \hat{J}_i(\bar{u}^\ell) - \hat{J}_i^\ell(\bar{u}^\ell) \right).
\]

(3.1.46)

Thus, especially for increasing $\ell$, i.e., for a decreasing error between the reduced-order and the full-order solution, we expect the a-posteriori error estimate to become less tight, since we lose the factors $\bar{\lambda}_i - \bar{\lambda}_i^\ell$ ($i = 1, \ldots, k$) in the estimate.

In conclusion, by looking at the qualitative comparison, we expect the a-posteriori error estimate for the WSP to be the tightest, since it uses the fewest estimates, which are – in a similar form – also used in the estimates for the ERPP and the PSP. For the latter two, we have to estimate the coercivity constant, which directly implies an overestimation of around two. Moreover, the quality of the estimate for the PSP depends on the error between the reduced-order and the full-order solution, while this is not the case for the WSP and the ERPP.

In order to test the a-posteriori error estimates quantitatively and confirm our theoretical findings in a realistic situation, we use the example from Section 2.1.4 in the following way: For generating the full-order reference solutions, we solve the example problem from Section 2.1.4 by the FE method. For the test run with the POD method, the minimizers of the cost functions $\hat{J}_1$, $\hat{J}_2$, $\hat{J}_3$ are computed by the FE method and the POD basis is then constructed by using snapshots of the state and adjoint equation at these minimizers. For choosing the number of basis functions, we use the relative energy criteria from Remark A.2.3 for different tolerances $\tau$. Throughout the rest of the optimization, this POD basis is not modified. The a-posteriori error estimates $\eta_{\text{WSP}}(\bar{u}^\ell)$, $\eta_{\text{ERPP}}(\bar{u}^\ell)$ and $\eta_{\text{PSM}}(\bar{u}^\ell)$ are then compared to the true error $\| \bar{u} - \bar{u}^\ell \|_U$ between the full-order and the reduced-order solution. Since there are already numerical results for the estimate for the WSP and the ERPP available in the literature (see, e.g., [KTGV18, TV09] for the WSP, and [Ban17, BBV17, Mak18, Spu19] for the ERPP), we mainly focus on the estimate for the PSP, but also compare its results to the other two estimates.

For the experiment, we choose $\tau \in \{1e^{-5}, 1e^{-6}, 1e^{-7}\}$ as the tolerance for the relative energy contained in the POD basis. These choices result in a POD basis of size $\ell = 9$, $\ell = 16$ and $\ell = 25$, respectively.

For the evaluation of the a-posteriori error estimate for the PSM, we use the formula from Theorem 3.1.19 together with Remark 3.1.20 (i): The Matlab-function `fmincon` is used to compute a local minimizer of the a-posteriori error estimate with respect to the Lagrange multiplier $\bar{\lambda}^\ell$. Note that no further solutions of the state and adjoint equations are needed for evaluating the a-posteriori error estimate for different values of $\bar{\lambda}^\ell$. Thus, the computational costs for solving this minimization problem are negligible.

In Figure 3.1(a), the true error together with the locally optimal and the 'standard' a-posteriori error estimate can be seen for the choice of $\tau = 1e^{-6}$ for all arising PSPs, i.e., the results for the three subproblems $(\hat{J}_1, \hat{J}_2)$, $(\hat{J}_1, \hat{J}_3)$ and $(\hat{J}_2, \hat{J}_3)$ as well as for the
entire problem \( \{ \hat{J}_1, \hat{J}_2, \hat{J}_3 \} \) are shown from left to right, separated by the dashed black lines. From this plot, it can be clearly seen that the a-posteriori error estimate is an upper bound on the error \( \| \bar{u}^\ell - \bar{u} \|_U \) for each of the PSPs. Moreover, we observe that minimizing the a-posterior error estimate does in most cases not lead to a significantly smaller error bound. However, since computing the optimal a-posteriori error estimate is cheap in practice, we still use it in our experiments.

The second interesting property is the overestimation of the error. By definition, the overestimation is always greater or equal than 1, where 1 is the ideal value. In Figure 3.1(b), the overestimation of the true error is depicted for the different choices \( \tau \in \{ 1e^{-5}, 1e^{-6}, 1e^{-7} \} \). There are several noteworthy aspects: As already argued in the beginning of this section, we can indeed observe that the overestimation becomes larger for an increasing basis size \( \ell \). Thus, the a-posteriori error estimate for the PSP can only be efficiently used for moderate error tolerances in the applications. Moreover, there is no clear connection between the value of \( \bar{\lambda}_1^\ell \) and the overestimation of the error. As the first cost function \( \hat{J}_1 \) only measures the control costs, its coercivity constant in (3.1.1) is exactly given by \( \sigma_{\bar{u}1}^{(1)} \), whereas we can only use estimates for the coercivity constants of \( \hat{J}_2 \) and \( \hat{J}_3 \). Thus, the estimate

\[
\sum_{i=1}^{3} \frac{\bar{\lambda}_i^\ell}{2} \sigma_{\bar{u}i}^{(i)} \left\| \bar{u} - \bar{u}^\ell \right\|_U^2 \leq \sum_{i=1}^{3} \frac{\bar{\lambda}_i^\ell}{2} \langle \nabla^2 \hat{J}_i(\bar{u}^\ell)(\bar{u} - \bar{u}^\ell) \rangle_\mathcal{U} \tag{3.1.47}
\]

is tighter if \( \bar{\lambda}_1^\ell \) is large in comparison to \( \bar{\lambda}_2^\ell \) and \( \bar{\lambda}_3^\ell \). In a similar way, the estimates of the coercivity constants for the WSP and the ERPP become tighter if \( \alpha_1 \) and \( \hat{J}_1(\bar{u}^\ell) - z_1 \) are large in comparison to \( \alpha_2, \alpha_3 \) and \( \hat{J}_2(\bar{u}^\ell) - z_2, \hat{J}_3(\bar{u}^\ell) - z_3 \), respectively. For these two methods, tighter estimates for the coercivity constants directly lead to tighter a-posteriori error estimates, as can be seen in the Figures 3.2(a),(c), where the overestimation of the respective a-posteriori error estimates in dependence of the relative size of \( \alpha_1 \) and \( \hat{J}_1(\bar{u}^\ell) - z_1 \) are displayed. Note that this was also observed in the literature, see, e.g.,
Figure 3.2: Left: Overestimation of the a-posteriori error estimates for $\ell = 16$ in dependence of the relative weight of $\hat{J}_1$. For the WSP and the PSP, the relative weight is given by the value of $\alpha_1$ and $\bar{\lambda}_1$, respectively. For the ERPP, it is given by $(\hat{J}_1(\bar{u}^\ell) - z_1)/\sum_{i=1}^3 (\hat{J}_i(\bar{u}^\ell) - z_i)$.

Right: Overestimation of the a-posteriori error estimates for the different basis sizes $\ell \in \{9, 16, 25\}$. 
holds \( \alpha \) some details of our implementation. Let us now continue with applying Algorithm 7 to the MOCP. To this end, we specify 3.1.5.2 Certified Error Control is only capable of ensuring moderately small approximation errors in practice. Thus, it overestimation of the estimate for the PSP increases if the true error decreases. Hence, on the one hand, for an increasing value of \( \lambda_1 \), the estimate (3.1.47) for the coercivity constant becomes tighter, but, on the other hand, the estimate (3.1.46) might become less tight so that it is not possible to predict how the overall tightness of the a-posteriori error estimate will behave in dependence of \( \lambda_1 \). This can be observed in Figure 3.2(e).

Consequently, the estimate (3.1.46) might become coarser for increasing \( \lambda_1 \). Hence, on the other hand, the estimate (3.1.46) might become less tight so that it is not possible to predict how the overall tightness of the a-posteriori error estimate will behave in dependence of \( \lambda_1 \). This can be observed in Figure 3.2(e).

As a final comparison, the overestimation for the three a-posteriori error estimates in dependence of the number of basis functions is shown in Figures 3.2(b),(d),(f). It is obvious that the estimates for the PSP and the ERPP are much tighter than the one for the PSP. One can also see the dependence of the overestimation on the relative size of \( \alpha_1 \) and \( \tilde{\lambda}_1(\tilde{u}^t) - z_1 \): The overestimation is the largest for the subproblem \((\tilde{J}_2, \tilde{J}_3)\), where it holds \( \alpha_1 = 0 \) and \( \tilde{\lambda}_1(\tilde{u}^t) - z_1 = 0 \) for all points, respectively. Moreover, the overestimations of the estimates for the PSP and the ERPP are independent of the true error. Hence, they can also be used to ensure very small approximation errors. In contrast to this, the overestimation of the estimate for the PSP increases if the true error decreases. Thus, it is only capable of ensuring moderately small approximation errors in practice.

3.1.5.2 Certified Error Control

Let us now continue with applying Algorithm 7 to the MOCP. To this end, we specify some details of our implementation.

[KTGV18] for the WSP and [BBV17] for the ERPP.

In contrast to this, we do not see this behavior for the PSP, cf. Figure 3.2(e). By looking at the a-posteriori error estimate based on Remark 3.1.20 (iii), this can be explained by the fact that the estimate (3.1.46) might become coarser for a larger value of \( \lambda_1^t \): Indeed, it holds \( \tilde{J}_1(\tilde{u}^t) = \tilde{J}_1^t(\tilde{u}^t) \), so that the estimate (3.1.46) can be rewritten as

\[
\sum_{i=2}^{3} (\lambda_i - \lambda_i^t) \left( \tilde{J}_i(\tilde{u}^t) - \tilde{J}_i^t(\tilde{u}^t) \right)
\leq \max \left( 0, \tilde{J}_2(\tilde{u}^t) - \tilde{J}_2^t(\tilde{u}^t), \tilde{J}_3(\tilde{u}^t) - \tilde{J}_3^t(\tilde{u}^t) \right) - \sum_{i=2}^{3} \lambda_i^t \left( \tilde{J}_i(\tilde{u}^t) - \tilde{J}_i^t(\tilde{u}^t) \right).
\]

For the right-hand side of this estimate, we have

\[
\max \left( 0, \tilde{J}_2(\tilde{u}^t) - \tilde{J}_2^t(\tilde{u}^t), \tilde{J}_3(\tilde{u}^t) - \tilde{J}_3^t(\tilde{u}^t) \right) - \sum_{i=2}^{3} \lambda_i^t \left( \tilde{J}_i(\tilde{u}^t) - \tilde{J}_i^t(\tilde{u}^t) \right)
\geq \max \left( 0, \tilde{J}_2(\tilde{u}^t) - \tilde{J}_2^t(\tilde{u}^t), \tilde{J}_3(\tilde{u}^t) - \tilde{J}_3^t(\tilde{u}^t) \right) \left( 1 - \lambda_2 - \lambda_3^t \right)
= \lambda_1^t \max \left( 0, \tilde{J}_2(\tilde{u}^t) - \tilde{J}_2^t(\tilde{u}^t), \tilde{J}_3(\tilde{u}^t) - \tilde{J}_3^t(\tilde{u}^t) \right),
\]

which is strictly monotonically increasing in \( \lambda_1^t \) if

\[
\max \left( \tilde{J}_2(\tilde{u}^t) - \tilde{J}_2^t(\tilde{u}^t), \tilde{J}_3(\tilde{u}^t) - \tilde{J}_3^t(\tilde{u}^t) \right) > 0.
\]

As a final comparison, the overestimation for the three a-posteriori error estimates in dependence of the number of basis functions is shown in Figures 3.2(b),(d),(f). It is obvious that the estimates for the WSP and the ERPP are much tighter than the one for the PSP. One can also see the dependence of the overestimation on the relative size of \( \alpha_1 \) and \( \tilde{\lambda}_1(\tilde{u}^t) - z_1 \): The overestimation is the largest for the subproblem \((\tilde{J}_2, \tilde{J}_3)\), where it holds \( \alpha_1 = 0 \) and \( \tilde{\lambda}_1(\tilde{u}^t) - z_1 = 0 \) for all points, respectively. Moreover, the overestimations of the estimates for the PSP and the ERPP are independent of the true error. Hence, they can also be used to ensure very small approximation errors. In contrast to this, the overestimation of the estimate for the PSP increases if the true error decreases. Thus, it is only capable of ensuring moderately small approximation errors in practice.

3.1.5.2 Certified Error Control

Let us now continue with applying Algorithm 7 to the MOCP. To this end, we specify some details of our implementation.
Termination conditions. Apart from an inaccurate POD basis, another source of error might be the inaccurate computation of the solution $\bar{u}^\ell$ by the optimization algorithm. To ensure that an inexact optimization is not the reason for an insufficiently accurate reduced-order solution, we use an additional termination condition for the optimization algorithms: With the exact same arguments as in the proofs of Theorems 3.1.6, 3.1.12 and 3.1.19, a-posteriori error estimates for the error $\|\bar{u}^\ell - u^{(i)}\|_u$ can be shown, where $u^{(i)}$ is the current iterate of the optimization algorithm for the reduced-order problem. For these estimates, only cheaply computable reduced-order quantities are needed so that they can be used as an additional termination condition in the optimization algorithm. The termination tolerance $\varepsilon_{\text{apost}}$ for these conditions needs to be coupled to the desired error tolerance $\varepsilon_U$. In particular, it should hold $\varepsilon_{\text{apost}} < \varepsilon_U$, since otherwise we cannot expect that $\|\bar{u} - \bar{u}^\ell\|_u < \varepsilon_U$ holds. In our experiments, we set $\varepsilon_{\text{apost}} = 0.5 \varepsilon_U$.

It is worth mentioning that we only keep the termination conditions used in Section 2.1.4 as a safeguard to prevent the optimization algorithm from iterating infinitely in case the a-posteriori error estimate never gets smaller than the tolerance $\varepsilon_{\text{apost}}$. This might happen, for example, due to the overestimation of the true error by the a-posteriori error estimate. Especially for the PSM, the error tolerance should not be chosen too small to prevent this from happening. Note that this situation did not occur for the parameter choices that we make in this section.

Computation of the initial POD basis. The first step of the ERPM and the PSM is to minimize the individual cost functions $\hat{J}_1$, $\hat{J}_2$ and $\hat{J}_3$ and for consistency we also start with solving these problems for the WSM. Minimizing $\hat{J}_1$ is trivial, since the optimal control is given by $\bar{u} = 0$. Since there is no POD basis available in the beginning of the minimization of $\hat{J}_2$, we perform one step with the FE algorithm and compute the iterate $u^{(1)}$. Then we use snapshots at this iterate to generate the POD basis with an energy tolerance of $\tau = 1e^{-5}$. Given this POD basis, we use Algorithm 5 to ensure that the computed minimizer of $\hat{J}_2$ satisfies the desired error tolerance. Although there is a POD basis available, we proceed similarly for the minimization of $\hat{J}_3$: We first do one step with the FE algorithm and then take snapshots at the first iterate to extend the POD basis obtained from the minimization of $\hat{J}_2$ by using Algorithm 6. Again, we use this POD basis and Algorithm 5 for the minimization of $\hat{J}_3$ to ensure that the error tolerance is satisfied.

Tolerances for adapting the POD basis. In all of the following experiments, we use the constants $\kappa_1 = 0.2$, $\kappa_2 = 0.1$ for adapting the POD basis, cf. lines 4 and 9 of Algorithm 7.

Using a fixed POD basis. To show the benefits of the certified error control by Algorithm 7 in comparison to an initially computed, fixed POD basis, we proceed in the following way: For computing the POD basis, we follow the procedure explained above with the only difference that we do not use Algorithm 5 to ensure the desired error tolerance for the minimizers of $\hat{J}_2$ and $\hat{J}_3$. Thus, the fixed POD basis is computed from snapshots at the first iterates of the minimization of $\hat{J}_2$ and $\hat{J}_3$. For determining the size of the POD basis, we use the energy tolerance $\tau \in \{1e^{-5}, 1e^{-7}\}$, which leads to a basis
size of \( \ell = 15 \) and \( \ell = 29 \), respectively. The rest of the multiobjective optimization is conducted with this fixed POD basis.

**Results for the WSM**

For the WSM, we choose the error tolerance \( \varepsilon_U = 2 \times 10^{-4} \). Thus, the termination condition for the a-posteriori error estimate within the optimization algorithm is set to \( \varepsilon_{apost} = 1 \times 10^{-4} \). In Algorithm 6, we choose \( \varepsilon_{rem} = 0 \), so that old basis functions are not removed when updating the POD basis. The rest of the tolerances are chosen as for the FE experiments in Section 2.1.4.

The results of the experiment are depicted in Figure 3.3. Figure 3.3(a) shows the a-posteriori error estimate together with the true error for every solved WSP. From this, we see that Algorithm 7 can indeed ensure that \( \eta_{WSM}^\ell(\bar{u}^\ell) < \varepsilon_U \) holds for all arising WSPs.

Especially by looking at the subproblems \((\hat{J}_1, \hat{J}_2)\), \((\hat{J}_1, \hat{J}_3)\) and \((\hat{J}_2, \hat{J}_3)\), one can clearly see that the POD basis is updated whenever the a-posteriori error estimate is about to exceed the tolerance. The number of used POD basis functions throughout Algorithm 7 is shown in Figure 3.3(b). From this plot, we can see the influence of the different subproblems, which are again separated by the dashed black lines:

From the Pareto optimal point 1 to 23, the subproblem \((\hat{J}_1, \hat{J}_2)\) is solved, i.e., the weights start at \( \alpha^1 = (23/24, 1/24, 0) \) and end at \( \alpha^{23} = (1/24, 23/24, 0) \). Since the minimizer of \( \hat{J}_1 \) is given by \( \bar{u} = 0 \), the WSPs with \( \alpha_1 \approx 1 \) are expected to have a solution being quite close to 0. Thus, the resulting dynamics are not very complex and can therefore be captured with relatively few basis functions. The bigger the influence of the second weight \( \alpha_2 \), the more complex dynamics arise, so that more POD basis functions are needed to capture them. This explains the increase of the number of POD basis functions.

From the Pareto optimal point 24 to 46, the subproblem \((\hat{J}_1, \hat{J}_3)\) is solved, starting from the weight \((23/24, 0, 1/24)\) and ending at \((1/24, 0, 23/24)\). Thus, the increasing number of needed POD basis functions can be explained with the same arguments as for the first subproblem.

The solution of the third subproblem \((\hat{J}_2, \hat{J}_3)\) can be seen from the Pareto optimal point 47 to 69. In here, WSPs with \( \alpha_1 = 0 \) are being solved. Since the weight in front of the control costs is only \( \sigma^{(2)}_{U} = 0.002 \) and \( \sigma^{(3)}_{U} = 0.001 \) for \( \hat{J}_2 \) and \( \hat{J}_3 \), respectively, the optimal controls take larger values, which implies more complex dynamics. Moreover, we showed that in this situation, the overestimation of the true error by the a-posteriori error estimate is the largest due to the overestimation of the coercivity constant. For these two reasons, more POD basis functions are needed to guarantee that the error tolerance is satisfied than in large parts of the first two subproblems. This results in a maximal number of 38 POD basis functions.

The remaining part of the plot shows the solutions to the original problem \((\hat{J}_1, \hat{J}_2, \hat{J}_3)\), where WSPs for weights \( \alpha > 0 \) are solved with the value of \( \alpha_1 \) monotonically increasing from 1/24 to 23/24. Thus, in the beginning of this part, there are still a lot of basis functions needed to satisfy the error tolerance, whereas this number decreases in the further course. Here the advantage of the possibility of reducing the number of POD basis functions in line 4 of Algorithm 7 becomes obvious: If we did not include this possibility,
all WSPs in the fourth part would be solved with 38 basis functions, which would increase the computational time noticeably.

Finally, the comparison of the results obtained from Algorithm 7 with the ones obtained from using fixed POD basis with sizes $\ell = 15$ and $\ell = 29$, respectively, can be seen in the Figures 3.3(c),(d). In Figure 3.3(c), the a-posteriori error estimates for the three different strategies are depicted, whereas Figure 3.3(d) shows the true errors. For a fixed POD basis with $\ell = 15$ basis functions, the percentage of points, for which both the a-posteriori error estimate and the true error is larger than the error tolerance, is around 90%. Choosing a larger but still fixed POD basis with $\ell = 29$ basis functions improves the results noticeably. However, we see that for all solutions of the subproblem $\left(\hat{J}_2, \hat{J}_3\right)$ both the a-posteriori error estimate and the true error are larger than the error tolerance. Of course, one could choose an even bigger fixed POD basis in the beginning to further improve the results and eventually come to a point, where all solutions satisfy the error tolerance. However, for any
given POD basis, there is no guarantee that this is indeed the case without evaluating the a-posteriori error estimate after solving each WSP. In particular, this cannot be guaranteed before solving the MOCP. In contrast to this, Algorithm 7 is able to control the number of POD basis functions automatically.

**Results for the ERPM**

For the experiment with the ERPM, we choose the same data as for the WSM. The results are similar to those of the WSM: Algorithm 7 ensures that the a-posteriori error estimate is smaller than the error tolerance for all of the computed Pareto optimal points, cf. Figure 3.4(a). Consequently, the true error also satisfies the error tolerance for all computed points. The evolution of the number of POD basis functions is displayed in Figure 3.4(b). As already explained for the WSM, the influence of the subproblems \((\hat{J}_1, \hat{J}_2), (\hat{J}_1, \hat{J}_3)\) and \((\hat{J}_2, \hat{J}_3)\) as well as the problem \((\hat{J}_1, \hat{J}_2, \hat{J}_3)\) can be seen in the number

![Graph](image1)

(a) A-posteriori estimate and true error for Algorithm 7.

![Graph](image2)

(b) Number of POD basis functions for Algorithm 7.

![Graph](image3)

(c) A-posteriori error estimates for different strategies.

![Graph](image4)

(d) True errors for different strategies.

**Figure 3.4:** Results of Algorithm 7 for the ERPM.
of used POD basis functions. The maximal number of 49 POD basis functions is needed for solving an ERPP in the end of the subproblem \((\hat{J}_1, \hat{J}_3)\), where the term \(\hat{J}_1(\hat{u}^\ell) - z_1\) is relatively small. After that, the number of needed basis functions stays constantly high while solving the subproblem \((\hat{J}_2, \hat{J}_3)\). For the problem \((\hat{J}_1, \hat{J}_2, \hat{J}_3)\), we observe that the number of basis functions is not monotonically decreasing as it is for the WSM. The reason is that the reference points are not ordered from 'hard' to 'easy' problems: One might jump from a reference point, for which \(\hat{J}_1(\bar{u}^\ell) - z_1\) is relatively small, to another reference point, for which \(\hat{J}_1(\bar{u}^\ell) - z_1\) is relatively large. Nevertheless, Algorithm 7 is able to automatically deal with this situation by decreasing or increasing the number of POD basis functions, respectively.

Comparing the results obtained from using Algorithm 7 with those obtained from using a fixed POD basis, we notice that \(\ell = 15\) and \(\ell = 29\) are not enough basis functions to satisfy the error tolerance in most of the computed Pareto optimal points, cf. Figures 3.4(c),(d). In fact, for \(\ell = 15\), for around 99\% of all points the a-posteriori error estimate and for around 98\% the true error exceed the error tolerance. These numbers improve to 20\% and 6\%, respectively, for \(\ell = 29\) basis functions. Again, the subproblem \((\hat{J}_2, \hat{J}_3)\) is the problematic subproblem, since its optimal controls generate the most complex dynamics. However, also for some points in the other subproblems, for which the relative size of \(\hat{J}_1(\bar{u}^\ell) - z_1\) is relatively small, the a-posteriori error estimate exceeds the error tolerance.

**Results for the PSM**

As already explained, we have to choose larger tolerances for the experiment with the PSM. To be precise, we set \(\varepsilon_U = 2e^{-3}\) and \(\varepsilon_{\text{apost}} = 1e^{-3}\). Moreover, we choose \(\varepsilon_{\text{rem}} = 1e^{-5}\) in Algorithm 6, so that old basis functions can be potentially removed from the POD basis whenever it is updated. This shall prevent the POD basis from growing too much.

The results of this experiment are depicted in Figure 3.5. In Figure 3.5(a), we see that the a-posteriori error estimate is smaller than the error tolerance for all computed Pareto optimal points. Consequently, also the true error satisfies the error tolerance for all points. Note that the true error is actually smaller than the a-posteriori error estimate by up to two orders of magnitude, which is due to the large overestimation of the error, especially when the error is small. Thus, even though the error tolerance was chosen to be larger than the one for the WSM and the ERPM, the number of needed basis functions even exceeds the other two methods. For the subproblems \((\hat{J}_2, \hat{J}_3)\) and \((\hat{J}_1, \hat{J}_2, \hat{J}_3)\), a maximum of 67 basis functions is needed to stay below the error tolerance, see Figure 3.5(b). Another reason for this might be that the PSPs are structurally harder to solve than the WSPs and the ERPPs, so that more accurate approximations of the cost functions and their gradients are needed to achieve the same accuracy as for the WSPs and the ERPPs. Note that the decrease from 67 to 54 basis functions in the middle of solving \((\hat{J}_1, \hat{J}_2, \hat{J}_3)\) is due to a POD basis update, in which some of the old basis functions were removed.

By looking at Figure 3.5(c), we notice that \(\ell = 15\) and \(\ell = 29\) basis functions are not enough to keep the a-posteriori error estimate below the error tolerance. In fact, the estimates are larger than the tolerance in around 99\% and 85\% of all points, respectively. However, the plot of the true error in Figure 3.5(d) reveals that the true error for \(\ell = 29\) basis functions
Comparison of performances

Finally, we compare the runtimes of the different strategies as well as the required FE solves of the state and adjoint equation in Table 3.1. First of all, we note that using a fixed POD basis is faster than using the FE method by a factor of around 25 – 60 for all three scalarization methods depending on the number of POD basis functions. The reason is that only 68 FE solves of the state and adjoint equations are needed. In comparison to this, Algorithm 7 needs significantly more computational time. On the one hand, evaluating the a-posteriori error estimate after each optimization problem is actually below the error threshold for all points. This emphasizes the importance of a tight a-posteriori error estimate w.r.t. the efficiency of Algorithm 7: If we had a tighter a-posteriori error estimate, we would need way less basis functions than the numbers shown in Figure 3.5(b) to satisfy the error tolerance.

Figure 3.5: Results of Algorithm 7 for the PSM.
### Table 3.1: Comparison of the computational time (with speed-up in comparison to the FE method) and the number of FE solves between the POD-based methods for the different scalarization methods. Last row: Results for only computing the a-posteriori error estimates.

<table>
<thead>
<tr>
<th>Method</th>
<th>WSM</th>
<th>Time [s]</th>
<th># FE</th>
<th>ERPM</th>
<th>Time [s]</th>
<th># FE</th>
<th>PSM</th>
<th>Time [s]</th>
<th># FE</th>
</tr>
</thead>
<tbody>
<tr>
<td>FE</td>
<td></td>
<td>2741 (-)</td>
<td>12817</td>
<td>5381 (-)</td>
<td>24740</td>
<td>39308 (-)</td>
<td>179363</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Alg. 7</td>
<td></td>
<td>331 (8.3)</td>
<td>1112</td>
<td>442 (12.2)</td>
<td>1160</td>
<td>7505 (5.2)</td>
<td>707</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ℓ = 15</td>
<td></td>
<td>47 (58.3)</td>
<td>68</td>
<td>108 (49.8)</td>
<td>68</td>
<td>968 (40.6)</td>
<td>68</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ℓ = 29</td>
<td></td>
<td>64 (42.8)</td>
<td>68</td>
<td>148 (26.4)</td>
<td>68</td>
<td>1207 (32.6)</td>
<td>68</td>
<td></td>
<td></td>
</tr>
<tr>
<td>a-post.</td>
<td></td>
<td>197 (-)</td>
<td>920</td>
<td>218 (-)</td>
<td>928</td>
<td>142 (-)</td>
<td>620</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

quite costly, since FE solutions of the state and adjoint equations need to be computed. In the last row of Table 3.1, we see that this is responsible for most of the FE solves within Algorithm 7 and already takes between 140 s and 218 s for the different methods, mainly depending on the number of solved optimization problems. On the other hand, whenever the a-posteriori error estimate is larger than the error tolerance, the POD basis is updated and the same optimization problem is solved again with the new basis, which increases the computational time additionally. Moreover, especially for the PSM, the number of POD basis functions is clearly larger than $\ell = 29$ for most of the multiobjective optimization, which additionally slows down the computation. To this end, note that the system matrices resulting from applying the POD method are dense, whereas the FE matrices are sparse. Thus, the POD method loses a lot of its speed-up already at a moderate number of basis functions. Nevertheless, in comparison to the FE method, the runtimes can still be reduced by factors of around 8 (WSM), 12 (ERPM) and 5 (PSM) by using Algorithm 7. At the same time, Algorithm 7 can guarantee that the desired error tolerance is maintained, which is not the case when using a fixed POD basis.

#### 3.2 RB-Based Multiobjective Parameter Optimization

While the POD method was applied to the MOCP of a parabolic PDE in the previous section, the RB method is more suitable for adaptively computing the reduced-order models when solving MPOPs of elliptic PDEs. The reason is that solving the elliptic PDE for a specific parameter only generates a single snapshot instead of a whole trajectory of snapshots. Thus, it is not reasonable to apply the POD method, whose idea is to find the dominant dynamics of a given set of snapshots, to adaptively compute the reduced-order model for these problems. For an introduction to the RB method, we refer to Appendix A.1, where not only the method, but also its application to elliptic PDEs and parameter optimization problems of elliptic PDEs is described. Here we will continue with applying the RB method to the MPOP from Section 2.2.3. To this end, assume that a reduced basis space $V^r$ is given so that we can obtain reduced-order approximations $\hat{J}_1^r, \ldots, \hat{J}_k^r$ of the cost functions.
\( \hat{J}_1, \ldots, \hat{J}_k \), see Appendix A.1.2. Then the reduced-order MPOP reads

\[
\min_{u \in \mathbb{U}_{ad}} \hat{J}^r(u) = \min_{u \in \mathbb{U}_{ad}} \left( \begin{array}{c}
\hat{J}_1^r(u) \\
\vdots \\
\hat{J}_k^r(u)
\end{array} \right). \tag{RB-MPOP}
\]

To fix the notation, let us introduce the WSP, the ERPP and the PSP for this problem:

Given \( \alpha \in \Delta_k \), the reduced-order WSP reads

\[
\min_{u \in \mathbb{U}_{ad}} (\hat{J}^r)^{g_{\alpha}}(u) := \min_{u \in \mathbb{U}_{ad}} \sum_{i=1}^k \alpha_i \hat{J}_i^r(u). \tag{WSP^r(\alpha)}
\]

For any \( z \in \mathbb{R}^k \), the reduced-order ERPP is given by

\[
\min_{u \in \mathbb{U}_{ad}} (\hat{J}^r)^{g_z}(u) := \min_{u \in \mathbb{U}_{ad}} \frac{1}{2} \sum_{i=1}^k \left( \hat{J}_i^r(u) - z_i \right)^2. \tag{ERPP^r(z)}
\]

As for the POD method, we denote by \( Z_{opt,(u)}^r \) the (weakly) Pareto admissible set for the problem (RB-MPOP).

Finally, for an arbitrary reference point \( z \in \mathbb{R}^k \), the reduced-order PSP reads

\[
\min_{(u,t)} t \\
\text{s.t.} \\
u \in \mathbb{U}_{ad}, \ t \in \mathbb{R}, \\
\hat{J}_i^r(u) - z_i \leq t, \ i = 1, \ldots, k. \tag{PSP^r(z,r)}
\]

The a-priori and a-posteriori error estimates for the three scalarization methods from Section 3.1 exploited the structure of the underlying linear-quadratic MOCP in two ways: First, the estimates made use of the fact that the cost functions are quadratic, and second, an estimate of the coercivity constant of the Hessian matrix, which is easily computable in the linear-quadratic case, was used. For non-convex problems – and thus, in particular, for the considered MPOP – one would have to proceed differently to obtain error estimates. In [KTV13], an a-posteriori error estimate for a non-convex problem with the cost function being the weighted sum of two non-convex functions was presented. In the proof of the estimate, the mean value theorem is used instead of the fact that the cost functions are quadratic. The estimation of the coercivity constant of the Hessian matrix is also possible, yet requires some additional effort, see also [ITV16, Tre17]. Here we will not proceed in this direction, but rather propose to use a Trust-Region (TR) RB algorithm. The idea of using a TR algorithm in the framework of model-order reduction is present in the literature since the early 2000s, cf. [AFS00, Fah00]. This approach was further exploited in more recent publications, cf. [BKMG+20, KMO+20, QGVW17, YM13]. In this section, we introduce and discuss the TR-RB algorithm recently presented in [BKMG+20], to which the author made substantial contributions. Subsequently, we show that this algorithm can be applied to solve the non-convex MPOP presented in Section 2.2.3 by the WSM,
3.2 RB-Based Multiobjective Parameter Optimization

3.2.1 A Trust-Region RB Algorithm

In this section, we will use the general notation \( J: \mathbb{U} \rightarrow \mathbb{R} \) for a cost function in order to avoid confusions with the multiobjective cost function \( \hat{J} \).

The general idea of standard TR methods (cf. [CGT00]) is to build a model function \( m^{(i)} \) of the original cost function \( J \) at the current iterate \( u^{(i)} \). Usually, this model function is a quadratic approximation of the cost function of the form

\[
m^{(i)}(d) := \frac{1}{2} \langle H(u^{(i)})d, d \rangle_{\mathbb{U}} + \langle \nabla J(u^{(i)}), d \rangle_{\mathbb{U}} + J(u^{(i)}) - J(u^{(i)}(0)),
\]

where \( H(u^{(i)}) \) is either the Hessian \( \nabla^2 J(u^{(i)}) \) or an approximation of it. This model is said to be trusted in a neighborhood around the current iterate \( u^{(i)} \), the so-called Trust-Region \( \Delta^{(i)} \). Then the minimizer \( d^{(i)} \) of the problem (also called the TR subproblem)

\[
\min_d m^{(i)}(d) \quad \text{s.t.} \quad u + d \in \mathbb{U}_{\text{ad}} \cap \Delta^{(i)}
\]

is computed and it is checked whether the new iterate is accepted by looking at the quotient

\[
\rho^{(i)} := \frac{J(u^{(i)} + d^{(i)}) - J(u^{(i)})}{m^{(i)}(d^{(i)}) - m^{(i)}(0)},
\]

which compares the achieved reduction in the cost function with the predicted reduction in the model function. Only if this quotient is large enough, the new iterate \( u^{(i+1)} = u^{(i)} + d^{(i)} \) is accepted and the TR is possibly increased. If it is too small, the iterate is rejected, the TR is decreased and the subproblem (3.2.1) is solved again.

This idea has been adapted to model-order reduction in the following way [AFS00, Fah00, YM13]: Instead of using a quadratic model function at the current iterate \( u^{(i)} \), a reduced-order version \( J^{r,(i)} \) of the cost function \( J \) is used as model function in the TR subproblem. This has the advantage that the reduced-order cost function \( J^{r,(i)} \) only needs to be a good approximation of the full-order cost function in a neighborhood of \( u^{(i)} \). In the publications [KMO+20, QGVW17, YM13], this is incorporated into the TR framework by demanding that the estimated relative error in the cost function is smaller than a given TR radius \( \delta^{(i)} \), i.e., the TR-RB subproblem reads

\[
\min_{u \in \mathbb{U}_{\text{ad}}} J^{r,(i)}(u) \quad \text{s.t.} \quad q^{(i)}(u) := \frac{\Delta J^{r,(i)}(u)}{J^{r,(i)}(u)} < \delta^{(i)},
\]

where \( \Delta J^{r,(i)}(u) \) is an estimate for the error \( |J(u) - J^{r,(i)}(u)| \).

Here we present and describe the TR-RB algorithm introduced in [BKM+20]. Note that the author made substantial contributions to the inclusion of the possibility of skipping
basis updates and the convergence analysis in [BKM+20, Section 3]. Let us also mention that the author did neither contribute to the non-conforming dual approach for the RB method in [KMO+20, Section 2], nor to the TR-RB algorithm [KMO+20, Algorithm 1], on which the algorithm is based. In contrast to the TR-RB algorithms in [QGVW17, YM13], which can only deal with unconstrained problems, the ones in [BKM+20, KMO+20] can also handle problems with parameter constraints.

Let us directly start with stating the TR-RB algorithm (cf. Algorithm 8) and explaining its details.

**Initial RB space at** $u^{(0)}$. If there is no RB space $V_{r,0}$ available, we generate the initial RB by using the FE solutions of the state and adjoint equations at $u^{(0)}$.

**Updating the RB space** $V_{r,i}$. By updating the RB space $V_{r,i}$ at $u^{(i+1)}$, we mean that the FE solutions of the state and adjoint equations at the iterate $u^{(i+1)}$ are computed and added to the RB space, cf. Algorithm 10.

**Reduced-order cost function** $J_{r,i}$. By the notation $J_{r,i}$, we refer to the reduced-order cost function which results from using the RB space $V_{r,i}$.

**The approximated generalized Cauchy (AGC) point.** In line 3, the so-called approximated generalized Cauchy (AGC) point is computed, which is defined in the following way.

**Definition 3.2.1.** Let $P_{\text{ad}} : \mathcal{U} \rightarrow \mathcal{U}_{\text{ad}}$ be the canonical projection onto the closed and convex set $\mathcal{U}_{\text{ad}}$.

**Definition 3.2.2.** Let $\kappa \in (0,1)$ and $\kappa_{\text{arm}} \in (0,1)$ be backtracking parameters. For the current iterate $u^{(i)}$ define $d^{(i)} := \nabla J_{r,i}(u^{(i)})$. Let $j \in \mathbb{N}$ be the smallest number, for which the two conditions

$$J_{r,i}\left( P_{\text{ad}}(u^{(i)} - \kappa^j d^{(i)}) \right) - J_{r,i}(u^{(i)}) \leq -\frac{\kappa_{\text{arm}}}{\kappa^j} \left\| P_{\text{ad}}(u^{(i)} - \kappa^j d^{(i)}) - u^{(i)} \right\|_U^2, \quad (3.2.3)$$
$$q^{(i)}(P_{\text{ad}}(u^{(i)} - \kappa^j d^{(i)})) \leq \delta^{(i)}, \quad (3.2.4)$$

are satisfied. Then we define the approximated generalized Cauchy (AGC) point as $u^{(i)}_{\text{AGC}} := P_{\text{ad}}(u^{(i)} - \kappa^j d^{(i)})$.

**Solving the TR-RB subproblem.** The TR-RB subproblem (3.2.2) of the current $i$-th iteration is solved in line 4. For this, we use the termination conditions

$$\left\| u - P_{\text{ad}}(u - \nabla J_{r,i}(u)) \right\|_U \leq \tau_{\text{sub}}, \quad (3.2.5)$$
$$\beta_{\text{bound}} \delta^{(i)} \leq q^{(i)}(u) \leq \delta^{(i)}, \quad (3.2.6)$$

where (3.2.5) is the typical first-order condition with tolerance $\tau_{\text{sub}} \in (0,1)$ and (3.2.6) is supposed to ensure that we do not spend too much time iterating in an area where the reduced-order cost function is a rather poor approximation of the full-order cost function anyway, see [QGVW17]. To this end, $\beta_{\text{bound}} \in (0,1)$ is chosen to be rather close to 1. In the numerical algorithm, this problem is solved by using a projected Newton-CG algorithm with the AGC point $u^{(i)}_{\text{AGC}}$ as a warm start.
Algorithm 8 TR-RB algorithm
1: Set $i = 0$ and Loop_flag=True;
2: while Loop_flag do
3:  Compute the AGC point $u_{\text{AGC}}^{(i)}$;
4:  Compute $u^{(i+1)}$ as solution of (3.2.2) with stopping criteria (3.2.5) & (3.2.6);
5:  if $\mathcal{J}^{r,(i)} (u^{(i+1)}) + \Delta \mathcal{J}^{r,(i)} (u^{(i+1)}) < \mathcal{J}^{r,(i)} (u_{\text{AGC}}^{(i)})$ then
6:     Accept $u^{(i+1)}$, set $\delta^{(i+1)} = \delta^{(i)}$, compute $\varrho^{(i)}$ and $g_h(u^{(i+1)})$;
7:     if $g_h(u^{(i+1)}) \leq \tau_{\text{FOC}}$ then
8:        Set Loop_flag=False;
9:     else if $\varrho^{(i)} \geq \eta \varrho^{(i)}$ then
10:        Enlarge the TR radius $\delta^{(i+1)} = \beta^{-1} \delta^{(i)}$;
11:        end if
12:     if not Skip_enrichment_flag then
13:        Update the RB space at $u^{(i+1)}$ to obtain $V^{r,(i+1)}$;
14:     else
15:        Set $V^{r,(i+1)} = V^{r,(i)}$;
16:     end if
17:  else if $\mathcal{J}^{r,(i)} (u^{(i+1)}) - \Delta \mathcal{J}^{r,(i)} (u^{(i+1)}) \geq \mathcal{J}^{r,(i)} (u_{\text{AGC}}^{(i)})$ then
18:      Reject $u^{(i+1)}$, shrink the radius $\delta^{(i)} = \beta \delta^{(i)}$ and go to line 3;
19:  else
20:     Compute $\mathcal{J}(u^{(i+1)})$, $g_h(u^{(i+1)})$ and $g^{(i)}$;
21:     if $g_h(u^{(i+1)}) \leq \tau_{\text{FOC}}$ then
22:        Set Loop_flag=False;
23:     else if Skip_enrichment_flag and $\varrho^{(i)} \geq \eta \varrho^{(i)}$ then
24:        Accept $u^{(i+1)}$ and set $\delta^{(i+1)} = \beta^{-1} \delta^{(i)}$;
25:     else if $\mathcal{J}(u^{(i+1)}) \leq \mathcal{J}^{r,(i)} (u_{\text{AGC}}^{(i)})$ then
26:        Accept $u^{(i+1)}$ and update the RB space at $u^{(i+1)}$ to obtain $V^{r,(i+1)}$;
27:     if $\varrho^{(i)} > \eta \varrho^{(i)}$ then
28:        Set $\delta^{(i+1)} = \beta^{-1} \delta^{(i)}$;
29:     else
30:        Set $\delta^{(i+1)} = \delta^{(i)}$;
31:     end if
32:  else
33:     Update the RB space at $u^{(i+1)}$ to obtain an updated version $V^{r,(i)}$;
34:     Reject $u^{(i+1)}$, set $\delta^{(i)} = \beta \delta^{(i)}$ and go to line 3;
35: end if
36: end if
37: end if
38: end while
39: Set $i = i + 1$;
Accepting the TR-RB step and updating the RB space. As it is common for TR methods, the question of when to accept the TR-RB step $u^{(i+1)}$ arises. The general condition for accepting the TR-RB step, which is needed in the proof of convergence, is that the solution $u^{(i+1)}$ of the TR-RB subproblem (3.2.2) satisfies the so-called error-aware sufficient decrease condition ([YM13])

$$\mathcal{J}^{r,(i+1)}(u^{(i+1)}) \leq \mathcal{J}^{r,(i)}(u_{\text{AGC}}^{(i)}),$$  

(3.2.7)

where $u^{(i+1)}$ is plugged into the new reduced-order cost function $\mathcal{J}^{r,(i+1)}$, which arises from using the (possibly updated) RB space $V^{r,(i+1)}$ on the left-hand side of (3.2.7). Thus, the question of accepting the TR-RB step is connected with the question of when to update the RB space $V^{r,(i)}$ and consequently, the reduced-order cost function $\mathcal{J}^{r,(i)}$. It is not advisable to update the RB space in every iteration to prevent it from growing too large, which would make the algorithm less efficient. Rather, we found that the following three conditions should hold at the new iterate to guarantee convergence of the algorithm analytically and to make it numerically stable: First,

$$\frac{|g(u^{(i+1)}) - g^{r,(i+1)}(u^{(i+1)})|}{g^{r,(i+1)}(u^{(i+1)})} \leq \tau_g$$  

(3.2.8)

for some $\tau_g > 0$, where

$$g(u) := \|u - P_{U_{\text{ad}}}(u - \nabla \mathcal{J}(u))\|_U,$$

$$g^{r,(i+1)}(u) := \|u - P_{U_{\text{ad}}}(u - \nabla \mathcal{J}^{r,(i+1)}(u))\|_U.$$  

This condition ensures that the relative deviation of the first-order critical conditions for the full-order and the reduced-order problem at the iterate $i + 1$ is bounded independently of $i$.

Second,

$$\frac{\|\nabla \mathcal{J}^{r,(i+1)}(u^{(i+1)}) - \nabla \mathcal{J}(u^{(i+1)})\|_U}{\|\nabla \mathcal{J}^{r,(i+1)}(u^{(i+1)})\|_U} \leq \min\{\tau_{\text{grad}}, \beta_{\text{grad}} \delta^{(i+1)}\}$$  

(3.2.9)

for some $\tau_{\text{grad}}, \beta_{\text{grad}} \in (0,1)$, which ensures that the gradient at the current iterate is approximated accurately enough.

Lastly, we demand

$$q^{(i+1)}(u^{(i+1)}) \leq \beta_q \delta^{(i+1)},$$  

(3.2.10)

for some $\beta_q \in (0,1)$, which implies that the starting point $u^{(i+1)}$ for solving the $(i + 1)$-st TR-RB subproblem (3.2.2) satisfies the TR-condition and is not already too close to the boundary of the TR.

In Algorithm 8, the TR-RB step $u^{(i+1)}$ is accepted if the four conditions (3.2.7), (3.2.8), (3.2.9) and (3.2.10) can be satisfied at the same time (by either updating the RB space
at \( u^{(i+1)} \) or not). To this end, we check the conditions (3.2.8), (3.2.9) and (3.2.10) at the cost function \( \mathcal{J}^{r,(i)} \) and use the boolean variable

\[
\text{Skip}_\text{enrichment\_flag}(i) := \left( q^{(i)}(u^{(i+1)}) \leq \beta_q \delta^{(i+1)} \right) \quad \text{and} \quad \left( \frac{|g(u^{(i+1)}) - g^{r,(i)}(u^{(i+1)})|}{g^{r,(i)}(u^{(i+1)})} \leq \tau_g \right) \quad \text{and} \quad \left( \frac{\|\nabla \mathcal{J}^{r,(i)}(u^{(i+1)}) - \nabla \mathcal{J}(u^{(i+1)})\|_U}{\|\nabla \mathcal{J}^{r,(i)}(u^{(i+1)})\|_U} \leq \min\{\tau_{\text{grad}}, \beta_{\text{grad}} \delta^{(i+1)}\} \right).
\]

Additionally, we make use of the fact that the reduced-order cost function is exact at \( u^{(i+1)} \) after updating the RB space, i.e., that it holds \( \mathcal{J}^{r,(i)}(u^{(i+1)}) = \mathcal{J}(u^{(i+1)}) \) and \( \nabla \mathcal{J}^{r,(i)}(u^{(i+1)}) = \nabla \mathcal{J}(u^{(i+1)}) \). For more details and a thorough proof that the conditions (3.2.7), (3.2.8) and (3.2.10) are satisfied whenever the TR-RB step is accepted by Algorithm 8, we refer to [BKM+20, Lemmata 3.4 & 3.7]. The respective proof for the condition (3.2.9) can be obtained analogously to [BKM+20, Lemma 3.4].

**Enlarging/Shrinking the TR radius.** If the TR-RB step \( u^{(i+1)} \) is rejected, the TR radius \( \delta^{(i)} \) is shrunk to \( \delta^{(i+1)} = \beta_{\text{TR}} \delta^{(i)} \) with some shrinking factor \( \beta_{\text{TR}} \in (0,1) \). On the other hand, if the TR-RB step \( u^{(i+1)} \) is accepted, there is the possibility of enlarging the TR radius for the next TR-RB subproblem. To this end, we compute the ratio

\[
\varrho^{(i)} := \frac{\mathcal{J}(u^{(i)}) - \mathcal{J}(u^{(i+1)})}{\mathcal{J}^{r,(i)}(u^{(i)}) - \mathcal{J}^{r,(i)}(u^{(i+1)})},
\]

which does not involve any further solves of the state equation, since all the involved quantities have already been computed before. If \( \varrho^{(i)} \geq \eta_\varrho \) for some \( \eta_\varrho \in [0.75,1) \), then the TR radius is enlarged to \( \delta^{(i+1)} := \beta_{\text{TR}}^{-1} \delta^{(i)} \).

**Termination condition for Algorithm 8.** For a given tolerance \( \tau_{\text{FOC}} \in (0,1) \), Algorithm 8 terminates at the TR-RB step \( u^{(i+1)} \) if

\[
g(u^{(i+1)}) < \tau_{\text{FOC}}
\]

holds. Note that this is the typical first-order condition for optimization problems with control constraints.

After explaining Algorithm 8 in detail, we can tend to the convergence analysis. To this end, we need some assumptions on the optimization problem and the cost function. The first assumption restricts us to optimization problems with box constraints. This restriction is needed for two reasons: First, we utilize the specific form of the projection operator \( P_{U_{ad}} \), and second, we can make use of the fact that the admissible set is compact.

**Assumption 3.1.** Assume that the admissible set \( U_{ad} \) is of the form \( U_{ad} = [u^a, u^b] \subset \mathbb{R}^p \) for some \( u^a, u^b \in \mathbb{R}^p \) with \( u^a \leq u^b \).
Moreover, in view of the TR-condition in the TR-RB subproblem (3.2.2), we require all (reduced-order) cost functions to be strictly positive for every parameter value.

**Assumption 3.2.** Assume that the cost functions $J$ and $J^{r,(i)}$ ($i \in \mathbb{N}$) are strictly positive, i.e., that there is $\varepsilon > 0$ such that

$$J(u) \geq \varepsilon \quad \text{and} \quad J^{r,(i)}(u) \geq \varepsilon$$

hold for all $u \in U_{\text{ad}}$ and all $i \in \mathbb{N}$.

For the convergence of the projected Newton-CG method, which is used to solve the TR-RB subproblems, we require that the reduced-order cost functions $J^{r,(i)}$ are twice continuously Fréchet differentiable. Moreover, we have to assume the continuity of the gradient $\nabla J$ and the uniform Lipschitz continuity of the reduced-order gradients $\nabla J^{r,(i)}$ to be able to show convergence of Algorithm 8.

**Assumption 3.3.** Assume that the cost function $J$ is continuously Fréchet differentiable. Moreover, let the reduced-order cost functions $J^{r,(i)}$ be twice continuously Fréchet differentiable for all $i \in \mathbb{N}$ and assume that the gradient $\nabla J^{r,(i)}$ is uniformly Lipschitz continuous, i.e., that it holds

$$\left\| \nabla J^{r,(i)}(u) - \nabla J^{r,(i)}(v) \right\|_U \leq C_L \| u - v \|_U$$

for all $u, v \in U_{\text{ad}}$ and a Lipschitz constant $C_L > 0$, which is independent of $i$.

Theoretically, it might happen that Algorithm 8 either never finds an acceptable solution of the TR-RB subproblem (3.2.2) or that the TR radius is converging to 0 in the course of the algorithm. To exclude these cases, we assume that there is a minimal TR radius, for which we can always find an acceptable solution of the TR-RB subproblem.

**Assumption 3.4.** Assume that there is $\delta_{\text{min}} > 0$ such that for every $i \in \mathbb{N}$ there is a TR radius $\delta^{(i)} \geq \delta_{\text{min}}$, for which there is a solution $u^{(i+1)}$ of the TR-RB subproblem (3.2.2) which is accepted by Algorithm 8.

To guarantee that an AGC point can always be computed in finitely many backtracking steps (with the maximal number of required backtracking steps being independent of the current iteration $i$), we need the following uniform continuity assumption on the functions $q^{(i)}$.

**Assumption 3.5.** Assume that the family of functions $(q^{(i)})_{i \in \mathbb{N}}$ is uniformly continuous w.r.t. the parameter $u$ and the index $i$, i.e., that

$$\forall \varepsilon > 0 : \exists \eta = \eta(\varepsilon) > 0 : \forall i \in \mathbb{N} : \forall u, v \in U_{\text{ad}} : \left( \| u - v \| < \eta \Rightarrow \left| q^{(i)}(u) - q^{(i)}(v) \right| < \varepsilon \right)$$

(3.2.13) holds.
Remark 3.2.3. It is hard to verify Assumptions 3.4 and 3.5 in practice. To avoid an infinite loop in Algorithm 8, we define a minimal TR radius $\delta_{\text{min}}$ close to the machine precision and terminate the algorithm as soon as the TR radius falls below $\delta_{\text{min}}$. However, note that this forced termination was never triggered in our practical experiments.

Concerning Assumption 3.5, it is known that the function $q^{(i)}$ is uniformly continuous for every $i \in \mathbb{N}$, since the admissible set $\mathcal{U}_{\text{ad}}$ is compact by Assumption 3.1, the error estimator $\Delta^{(i)}_J$ is continuous (cf. [KMO+20, Remark 2.6]) and the reduced-order function $\hat{J}^{(i)}_r$ is continuous and strictly positive by Assumptions 3.2 and 3.3. Thus, in Assumption 3.5, we only have to demand this continuity to be uniform in $i$. In a practical implementation, where the full-order problem is obtained from a high- but finite-dimensional FE discretization, this is automatically satisfied, since the reduced-order cost function is exact after finitely many updates of the RB space.

Under these assumptions, it is possible to show global convergence of the TR-RB method to a first-order critical point, cf. [BKM+20, Theorem 3.8]. As a direct consequence, we then obtain that Algorithm 8 terminates after finitely many steps.

Theorem 3.2.4. Let Assumptions 3.1 – 3.5 be satisfied. Then every accumulation point $\bar{u}$ of the sequence of iterates $(u^{(i)})_{i \in \mathbb{N}}$ is a first-order critical point for the full-order optimization problem, i.e., it holds

$$\|\bar{u} - P_{\mathcal{U}_{\text{ad}}} (\bar{u} - \nabla J(\bar{u}))\|_{\mathcal{U}} = 0.$$ 

In particular, Algorithm 8 terminates after finitely many steps.

Proof. For a proof of the convergence to a first-order critical point we refer to the proof of [BKM+20, Theorem 3.8]. Furthermore, since the admissible set $\mathcal{U}_{\text{ad}}$ is compact by Assumption 3.1, we know that there is at least one accumulation point $\bar{u}$ of the sequence $(u^{(i)})_{i \in \mathbb{N}}$. Thus, Algorithm 8 terminates after finitely many steps.

3.2.1.1 Application of the TR-RB Algorithm to the Scalarization Methods

In this section, we show that the TR-RB algorithm can be applied to the three scalarization methods for solving the MPOP presented in Section 2.2.3. Note that the admissible set $\mathcal{U}_{\text{ad}}$ might be unbounded in the problem formulation (MPOP). Thus, we assume that $\mathcal{U}_{\text{ad}}$ is bounded in the following, i.e., that the lower and upper bounds $u_a$ and $u_b$ do not take the values $-\infty$ and $\infty$, respectively, which is anyway the case for the specific example for which we conducted the numerical experiments in Section 2.2.4. This immediately implies that $\mathcal{U}_{\text{ad}}$ is compact.

For proving Assumption 3.3, we need the following result.

Lemma 3.2.5. There are constants $C_J, C_{\nabla J}, C_{\nabla^2 J} > 0$ such that for any $i \in \{1, \ldots, k\}$, any $u \in \mathcal{U}_{\text{ad}}$ and any choice of the RB space $V^r$ it holds

$$\|J^r_i(u)\| \leq C_J, \quad \|\nabla J^r_i(u)\|_{\mathcal{U}} \leq C_{\nabla J}, \quad \|\nabla^2 J^r_i(u)\|_{L^2(\mathcal{U})} \leq C_{\nabla^2 J}.$$
Proof. In Theorem A.1.1, it is stated that $\|S_r(u)\|$ is bounded independently of the RB space $V^r$. The same property can be shown for $\|A_r(u)\|$, $\|(S')'(u)h\|$ and $\|(A')'(u)h\|$. Together with the representations of the gradient and the Hessian from Corollary A.1.9 and the fact that $U_{ad}$ is compact, this proves the claim. 

Application to the WSM

We show that Assumptions 3.2 and 3.3 are satisfied for $(WSP_r(\alpha))$ for any $\alpha \in \Delta_k$.

**Lemma 3.2.6.** Let $\alpha \in \Delta_k$ be arbitrary. Then it holds $(\hat{J}^r)_{\alpha}^0(u) \geq 0$ for all $u \in U_{ad}$ for any RB space $V^r$. In particular, Assumption 3.2 can be satisfied by adding any positive constant to $(\hat{J}^r)_{\alpha}^0$, which does not change the first-order critical points.

**Proof.** This follows directly from the facts that the weight $\alpha$ is non-negative and that $\hat{J}^r_i(u) \geq 0$ holds for all $u \in U_{ad}$ and all $i \in \{1, \ldots, k\}$. 

By using Lemma 3.2.5, it is possible to show that Assumption 3.3 holds as well.

**Lemma 3.2.7.** Let $\alpha \in \Delta_k$ be arbitrary. Then Assumption 3.3 is satisfied.

**Proof.** The cost functions $\hat{J}_1, \ldots, \hat{J}_k$ are twice continuously Fréchet differentiable by Lemma 2.2.11. Thus, the function $\hat{J}_{\alpha}^0$ is also twice continuously Fréchet differentiable. With the same reasoning, we can also show that the cost functions $\hat{J}_1^r, \ldots, \hat{J}_k^r$ are twice continuously Fréchet differentiable. It directly follows that $(\hat{J}^r_{\alpha})^i$ is also twice continuously Fréchet differentiable for all $i \in \mathbb{N}$. The Hessian

$$\nabla^2 (\hat{J}^r_{\alpha})^i(u) = \sum_{j=1}^{k} \alpha_i \nabla^2 \hat{J}_j^r(u)$$

is bounded independently of $u$ and $i$, cf. Lemma 3.2.5. Thus, the uniform Lipschitz continuity of the gradients $\nabla (\hat{J}_{\alpha})^r$ follows from the mean value theorem. 

**Remark 3.2.8.** In the TR-RB algorithm, we need an a-posteriori error estimate $\Delta_{\alpha}^r(u)$ for the reduced-order cost function $J^r(u)$. Note that it holds

$$J(u) = \sum_{j=1}^{k} \alpha_j \hat{J}_j(u) \quad \text{and} \quad J^r(u) = \sum_{j=1}^{k} \alpha_j \hat{J}_j^r(u)$$

for $(WSP(\alpha))$ for any $\alpha \in \Delta_k$. By Theorem A.1.11, we have an a-posteriori error estimate for the individual cost functions

$$|J_j(u) - \hat{J}_j^r(u)| \leq \Delta_{\alpha}^r(u) \quad \text{for all } u \in U_{ad} \text{ and all } j \in \{1, \ldots, k\}.$$
Thus, we can use the a-posteriori error estimate
\[ |J(u) - J^{r,(i)}(u)| \leq \sum_{j=1}^{k} \alpha_j \Delta_j^{r,(i)}(u) =: \Delta_j^{r,(i)}(u) \]
for all \( u \in U_{ad} \) in this case.

**Application to the ERPM**

Let us show that Assumptions 3.2 and 3.3 are satisfied for \((\text{ERPP}^r(z))\) for any reference point \( z \in \bigcap_{r \in \mathbb{N}} Z_{opt,w}^r \cap Z_{opt,w} \).

**Lemma 3.2.9.** Let \( z \in \bigcap_{r \in \mathbb{N}} Z_{opt,w}^r \cap Z_{opt,w} \) be arbitrary. Then Assumption 3.2 can be satisfied by adding any positive constant to \((\hat{J}^r)^{g_z}\).

**Proof.** By the definition of the cost function, it follows directly that \((\hat{J}^r)^{g_z}(u) \geq 0\) holds for all \( u \in U_{ad} \) and any RB space \( V^r \), which proves the claim. \( \square \)

The proof of Assumption 3.3 can be conducted with the same arguments as for the WSP.

**Lemma 3.2.10.** Let \( z \in \bigcap_{r \in \mathbb{N}} Z_{opt,w}^r \cap Z_{opt,w} \) be arbitrary. Then Assumption 3.3 holds.

**Proof.** The application of the Hessian of the Euclidean reference point cost function to an arbitrary direction \( h \in U \) is given by
\[ \nabla^2(\hat{J}^{r,(i)})^{g_z} h = \sum_{j=1}^{k} (\hat{J}^{r,(i)}_j(u) - z_j) \nabla^2 \hat{J}^{r,(i)}_j(u) h + \langle \nabla \hat{J}^{r,(i)}_j(u), h \rangle \nabla \hat{J}^{r,(i)}_j(u) \]
for any \( u \in U_{ad} \) and \( i \in \mathbb{N} \). Thus, we can again use Lemma 3.2.5 to conclude that the Hessian is bounded independently of \( i \). By applying the mean value theorem, we obtain that the gradients \( \nabla(\hat{J}^{r,(i)})^{g_z} \) are Lipschitz continuous uniformly in \( i \). \( \square \)

**Remark 3.2.11.** We need an a-posteriori error estimate \( \Delta_j^{r,(i)}(u) \) for the reduced-order cost function \( J^{r,(i)} \) in the TR-RB algorithm. For any \( z \in \bigcap_{r \in \mathbb{N}} Z_{opt,w}^r \cap Z_{opt,w} \), we have
\[ J(u) = \frac{1}{2} \sum_{j=1}^{k} (\hat{J}_j(u) - z_j)^2 \quad \text{and} \quad J^{r,(i)}(u) = \frac{1}{2} \sum_{j=1}^{k} (\hat{J}_j^{r,(i)}(u) - z_j)^2 \]
for \((\text{ERPP}(z))\). By using the a-posteriori error estimates for the individual cost functions
\[ |\hat{J}_j(u) - \hat{J}_j^{r,(i)}(u)| \leq \Delta_j^{r,(i)}(u) \quad \text{for all} \ u \in U_{ad} \text{ and all} \ j \in \{1, \ldots, k\}, \]
which were presented in Theorem A.1.11, a simple calculation reveals
\[ |J(u) - J^{r,(i)}(u)| \leq \sum_{j=1}^{k} \frac{1}{2} \left( \Delta_j^{r,(i)}(u) \right)^2 + |\hat{J}_j^{r,(i)}(u) - z_j| \Delta_j^{r,(i)}(u) =: \Delta_j^{r,(i)}(u) \]
for all \( u \in U_{ad} \), which can be used as a-posteriori error estimate in this case. \( \diamond \)
3. MOR for PDE-Constrained Multiobjective Optimization

Application to the PSM

Recall that for solving the Pascoletti-Serafini problem (PSP($z,r$)) numerically, we use an augmented Lagrangian algorithm, cf. Algorithm 11 in Appendix B, in which the augmented Lagrangian subproblems

$$\min_{(u,t,s) \in \mathcal{X}} \mathcal{L}_A((u,t,s),\lambda;\mu) \quad \text{s.t.} \quad (u,t,s) \in \mathcal{X}_{ad}$$

are repeatedly solved for different Lagrange multipliers $\lambda$ and penalty parameters $\mu$.

We want to use Algorithm 11 and combine it with the TR-RB algorithm to compute a first-order critical point of (PSP($z,r$)). The idea is to apply the TR-RB algorithm to each of the augmented Lagrangian subproblems (3.2.14). To this end, given a Lagrange multiplier $\lambda \in \mathbb{R}^k$ and a penalty parameter $\mu > 0$, we define the reduced-order augmented Lagrangian (cf. (B.1.2)) by

$$\mathcal{L}'_A((u,t,s),\lambda;\mu) := t + \sum_{i=1}^{k} \lambda_i \left( \hat{J}_i(u) - z_i - t + s_i \right) + \frac{\mu}{2} \sum_{i=1}^{k} \left( \hat{J}_i(u) - z_i - t + s_i \right)^2,$$

which leads to the reduced-order augmented Lagrangian subproblem (cf. (B.1.3))

$$\min_{(u,t,s) \in \mathcal{X}} \mathcal{L}'_A((u,t,s),\lambda;\mu) \quad \text{s.t.} \quad (u,t,s) \in \mathcal{X}_{ad}.$$  

Note that the admissible set $\mathcal{X}_{ad} = \mathbb{U}_{ad} \times \mathbb{R} \times [0,\infty)^k$ is unbounded, so that Assumption 3.1 is not satisfied. Therefore, we introduce a problem, which is equivalent to (PSP($z,r$)) and for which the admissible set of the augmented Lagrangian subproblems is bounded.

We start by showing that the variable $t$ can be constrained to a compact set. So let a reference point $z \in \mathbb{R}^k$ be given. By choosing an arbitrary $\tilde{u} \in \mathbb{U}_{ad}$ and defining $\tilde{t} := \max_{i \in \{1,\ldots,k\}} \hat{J}_i(\tilde{u}) - z_i$, we get that $\hat{J}_i(\tilde{u}) - z_i \leq \tilde{t}$ for all $i \in \{1,\ldots,k\}$, so that $(\tilde{u},\tilde{t})$ is admissible for (PSP($z,r$)). Therefore, for the optimal $\tilde{t}$, we have $\tilde{t} \leq \tilde{t}$ and we can define the upper bound by $t^{\max} := \tilde{t} + 1$. On the other hand, we know that the cost functions $\hat{J}_1,\ldots,\hat{J}_k$ are non-negative. Thus, for any admissible tuple $(u,t) \in \mathbb{U}_{ad} \times \mathbb{R}$ it holds $-z_i \leq \hat{J}_i(u) - z_i \leq t$ for all $i \in \{1,\ldots,k\}$. Hence, we can define the lower bound $t^{\min} := -\min_{i \in \{1,\ldots,k\}} z_i - 1$.

In conclusion, we have just shown that the problem (PSP($z,r$)) is equivalent to

$$\min_{(u,t)} \quad t$$

$$\text{s.t.} \quad u \in \mathbb{U}_{ad}, \quad t \in [t^{\min},t^{\max}],$$

$$\hat{J}_i(u) - z_i \leq t, \quad i = 1,\ldots,k.$$  

We still have to deal with the problem that the admissible set for the slack variables $s$ is given by $[0,\infty)^k$. However, the partial derivative of the augmented Lagrangian $\mathcal{L}_A$ with respect to $s_i$ is given by

$$\partial_{s_i} \mathcal{L}_A((u,t,s),\lambda;\mu) = \lambda_i + \mu \left( \hat{J}_i(u) - z_i - t + s_i \right) \geq \lambda_i + \mu (-z_i - t^{\max} + s_i),$$
so that \( \mathcal{L}_A \) is strictly monotonically increasing in \( s_i \) for \( s_i > -\frac{\lambda}{\mu} + z_i + t_{\text{max}} =: s_i^{\text{max}} \). Thus, given the Lagrange multiplier \( \lambda \) and the penalty parameter \( \mu \), we can restrict the slack variable \( s_i \) to the interval \([0, s_i^{\text{max}}]\) without tampering with the solvability and the solution of the augmented Lagrangian subproblem. By setting \( \tilde{X}_{\text{ad}} := \mathcal{U}_{\text{ad}} \times [t_{\text{min}}, t_{\text{max}}] \times [0, s_i^{\text{max}}] \), the equivalent formulation for the augmented Lagrangian subproblem corresponding to (3.2.17) reads

\[
\min_{(u,t,s) \in \tilde{X}} \mathcal{L}_A((u,t,s), \lambda; \mu) \quad \text{s.t.} \quad (u,t,s) \in \tilde{X}_{\text{ad}}.
\] (3.2.18)

Similarly, the reduced-order augmented Lagrangian subproblem is given by

\[
\min_{(u,t,s) \in \tilde{X}} \mathcal{L}_A^r((u,t,s), \lambda; \mu) \quad \text{s.t.} \quad (u,t,s) \in \tilde{X}_{\text{ad}}.
\] (3.2.19)

As a corollary, we immediately get:

**Corollary 3.2.12.** Assumption 3.1 is satisfied for the problems (3.2.18) and (3.2.19).

Moreover, we can also show that Assumption 3.2 holds by possibly adding a constant to the augmented Lagrangian function.

**Lemma 3.2.13.** Let the Lagrange multiplier \( \lambda \) and the penalty parameter \( \mu \) be given. Then there is a constant \( C = C(\lambda, \mu) \) and \( \varepsilon > 0 \) such that

\[
\mathcal{L}_A((u,t,s), \lambda; \mu) + C > \varepsilon \quad \text{and} \quad \mathcal{L}_A^r((u,t,s), \lambda; \mu) + C > \varepsilon
\] (3.2.20)

hold for all \((u,t,s) \in \tilde{X}_{\text{ad}}\) and any RB space \( V^r \), where we only used that \( t \) is bounded and that the augmented Lagrangian functions are quadratic functions in the terms \( c_i(u,t,s) = \hat{J}_i(u) - z_i - t + s_i \) and \( c_i^r(u,t,s) = \hat{J}_i^r(u) - z_i - t + s_i \) for all \( i \in \{1, \ldots, k\} \), respectively. Thus, by defining \( C := -t_{\text{min}} + \sum_{i=1}^{k} \frac{\lambda_i^2}{2\mu_i} + 1 \), we get that (3.2.20) is satisfied for \( \varepsilon = 1 \). \( \square \)

In practice, we rarely observe that the augmented Lagrangian becomes negative. Thus, we only add the constant \( C \) if it is required. Lastly, we prove that Assumption 3.3 holds as well.

**Lemma 3.2.14.** Let the Lagrange multiplier \( \lambda \) and the penalty parameter \( \mu \) be given. Then Assumption 3.3 is satisfied.
Proof. From Lemma 2.2.11, we know that the cost functions \( \hat{J}, \ldots, \hat{J}_k \) are twice continuously Fréchet differentiable. Thus, the function \( (u, t, s) \rightarrow \mathcal{L}_{\lambda}(u, t, s, \lambda; \mu) \) is also twice continuously Fréchet differentiable as a concatenation of twice continuously Fréchet differentiable functions. In a similar way, we can show that the reduced-order augmented Lagrangians \( \mathcal{L}_{A}^{(i)}((\cdot, \cdot, \cdot), \lambda; \mu) \) are also twice continuously Fréchet differentiable for all \( i \in \mathbb{N} \). For the Hessian of the reduced-order augmented Lagrangian, we have

\[
\nabla^2 \mathcal{L}_{A}^{(i)}((u, t, s), \lambda; \mu)(h_0, h_1, h_2) = \\
\left( \sum_{j=1}^{k} \left[ \lambda_j + \mu c_j^{(i)} \right] \nabla^2 \hat{J}_j^{(i)}(u) h_0 + \mu (d_j^{r,(i)} - h_1 + h_2^2) \nabla \hat{J}_j^{(i)}(u) \right) \\
k_\mu h_1 - \mu \sum_{j=1}^{k} \left[ d_j^{r,(i)} + h_2^2 \right] \\
\mu \left[ d_1^{r,(i)} + h_1^2 - h_1 \right] \\
\vdots \\
\mu \left[ d_k^{r,(i)} + h_1^2 - h_1 \right]
\]

for any \( h = (h_0, h_1, h_2) \in \mathbb{U} \times \mathbb{R} \times \mathbb{R}^k \), where \( c_j^{r,(i)} := \hat{J}_j^{(i)}(u) - z_j - t + s_j \) and \( d_j^{r,(i)} := \langle \nabla \hat{J}_j^{(i)}(u), h_0 \rangle \) for \( j \in \{1, \ldots, k\} \). Thus, together with Lemma 3.2.5, it can be concluded that the Hessian \( \nabla^2 \mathcal{L}_{A}^{(i)}((u, t, s), \lambda; \mu) \) can be bounded independently of \( (u, t, s) \) and \( i \). Using once again the mean value theorem, this allows us to conclude that the gradients \( \nabla \mathcal{L}_{A}^{(i)}((\cdot, \cdot, \cdot), \lambda; \mu) \) are Lipschitz continuous with the Lipschitz constant \( C_L \) being independent of \( i \).

\[ \square \]

Remark 3.2.15. For \( (\text{PSP}(z, r)) \), we have that the (reduced-order) cost function in the TR-RB algorithm is given by

\[
\mathcal{J}(u, t, s) = \mathcal{L}_{\lambda}(u, t, s, \lambda; \mu) \quad \text{and} \quad \mathcal{J}^{r,(i)}(u, t, s) = \mathcal{L}_{A}^{(i)}((u, t, s), \lambda; \mu)
\]

for any reference point \( z \in \mathbb{R}^k \), any Lagrange multiplier \( \lambda \in \mathbb{R}^k \) and any penalty parameter \( \mu > 0 \). Again, using the a-posteriori error estimates for the individual cost functions

\[
\left| \hat{J}_j(u) - \hat{J}_j^{(i)}(u) \right| \leq \Delta_{\hat{J}_j^{(i)}}(u) \quad \text{for all } u \in \mathbb{U}_{ad} \text{ and all } j \in \{1, \ldots, k\},
\]

which are shown in Theorem A.1.11, it is straight-forward to conclude

\[
\left| \mathcal{J}(u, t, s) - \mathcal{J}^{r,(i)}(u, t, s) \right| \leq \sum_{j=1}^{k} \left( \lambda_j + c \left| \hat{J}_j^{(i)}(u) - z_j - t + s_j \right| \right) \Delta_{\hat{J}_j^{(i)}}(u) \\
+ \sum_{j=1}^{k} \frac{c}{2} \left( \Delta_{\hat{J}_j^{(i)}}(u) \right)^2 =: \Delta_{\mathcal{J}}^{r,(i)}(u)
\]

for all \( u \in \mathbb{U}_{ad} \), which can be used as a-posteriori error estimate in the TR-RB algorithm. \( \diamond \)
3.2 RB-Based Multiobjective Parameter Optimization

3.2.2 Numerical Experiments

In this section, we test the TR-RB Algorithm 8 numerically by applying it to the MPOP from Section 2.2.4. From a theoretical point of view, we could show that the TR-RB algorithm can be successfully applied to the WSM, the ERPM and the PSM. Here we restrict ourselves to the application to the PSM.

For comparability, all the parameters of the problem are chosen the same as in Section 2.2.4. For the TR-RB algorithm, we make the following choices:

- The initial TR radius is chosen as $\delta(0) = 0.1$, the tolerance for increasing the TR radius is set to $\eta = 0.75$ and the factor for shrinking and enlarging the TR radius to $\beta_{\text{incr}} = 0.5$. For the minimal TR radius we use $\delta_{\min} = 1e-16$.
- For the Armijo backtracking strategy, we use the constants $\kappa_{\text{arm}} = 1e^{-4}$ and $\kappa = 0.5$.
- The tolerance of the first-order condition is set to $\tau_{\text{FOC}} = \tau_{\text{FOC,sub}}^{(i)}$, where $\tau_{\text{FOC,sub}}^{(i)}$ is the tolerance for the first-order condition of the current augmented Lagrangian subproblem. Moreover, we choose $\tau_{\text{sub}} = 0.5 \tau_{\text{FOC}}$ as the tolerance of the first-order condition of the TR-subproblem and $\beta_{\text{bound}} = 0.9$ as the constant in (3.2.6).
- For checking the necessity of updating the RB space, we choose $\tau_g = 1$, $\tau_{\text{grad}} = 0.1$, $\beta_{\text{grad}} = 0.2$ and $\beta_q = 0.005$.

The TR-RB algorithm automatically modifies the RB space within the optimization, i.e., in the case of the PSP, within an augmented Lagrangian subproblem. However, in principle, we are free in the choice of the initial RB space for every PSP or all of the augmented Lagrangian subproblems, respectively. For our numerical tests, we always use the RB space obtained from the previous augmented Lagrangian subproblem as the initial RB space for the next augmented Lagrangian subproblem. For handling the RB space between two different PSPs, we investigate three different approaches. Note that for all approaches, we start with the minimization of the individual cost functions $\hat{J}_1$, $\hat{J}_2$ and $\hat{J}_3$ by the TR-RB method, i.e., formally, we solve the WSPs to the weights $(1, 0, 0)$, $(0, 1, 0)$ and $(0, 0, 1)$, respectively.

**Using one RB space.** The first idea is to use a single RB space $V^r$ for the entire multiobjective optimization. This basis is bit by bit enriched by the TR-RB algorithm either when solving the WSPs in the beginning or within the augmented Lagrangian subproblems of the PSPs. In this way, the RB space becomes a better and better approximation of the FE space so that we expect that less updates are needed in later PSPs.

**Using one RB space with restart.** For minimizing the three individual cost functions, we follow the approach from above, i.e., we generate one RB space $V^r$ while minimizing $\hat{J}_1$, which is then also used for minimizing $\hat{J}_2$ and so on. Denote by $V^*_r$ the resulting RB space after minimizing all three cost functions. For every occurring PSP, we take $V^*_r$ as initial RB space. In this way, we want to ensure that the information obtained during the minimization of the individual cost functions by the TR-RB algorithm remain in the RB space, while the basis size does not grow too much.
Using multiple (local) RB spaces. Lastly, we investigate the idea of using multiple RB spaces, which are local in parameter space, cf. [BDPV18, HDO11]. In contrast to the previous approaches, when minimizing the individual cost functions, we do not take the previously obtained RB space as an initial RB space for the next minimization, but generate three different RB spaces $V_{r_1}, V_{r_2}, V_{r_3}$. Hence, in the beginning of every PSP, we have a family of RB spaces $(V_{r_i})_{i \in \{1,...,m\}}$ and decide for the RB space $V_{r_i}$ if

$$q(0)(u(0)) < \beta q(0)$$

holds for a previously specified maximal number of basis functions $r_{\text{max}} \in \mathbb{N}$. In our numerical test below, we use $r_{\text{max}} = 45$. If several RB spaces satisfy these two conditions, we take the one with the smallest value of $q(0)(u(0))$. If no such RB space exists, a new one $V_{r_{m+1}}$ is initialized by using the FE solutions of the state and adjoint equations at the initial value $u(0)$. In this way, we ensure that the chosen RB space is a good model around the initial value. Additionally, by bounding the maximal dimension of the RB spaces, we encourage that several RB spaces of moderate size are generated, each of which is well-suited for a certain parameter range.

3.2.2.1 Numerical results

In Figure 3.6(b), the approximation of the Pareto front obtained by using the TR-RB algorithm with local RB spaces is shown. We observe that the hole in the upper left part of the Pareto front does not appear, in contrast to the approximation obtained by the FE method, see Figure 3.6(a). The reason is that actually none of the PSPs corresponding to reference points in the left upper part converged to the tip point of the lower right part, which was observed for the FE method in Section 2.2.4. This indicates that for this example, the TR-RB algorithm tends to stay in the region of the initial value, whereas jumps in parameter space might occur when solving the augmented Lagrangian subproblems as described in Appendix B. However, we do not have an analytical explanation for this and it might as well depend on the example. Apart from this, no visible differences can be seen. Note that the same holds true for the Pareto fronts obtained by using the TR-RB algorithm together with one RB space and one RB space with restart, respectively.

This is confirmed by the error plots between the optimal parameters for the FE and the TR-RB methods in Figure 3.7(a). Again, the dotted black lines separate the errors for the subproblems $(\hat{J}_1, \hat{J}_2)$, $(\hat{J}_1, \hat{J}_3)$, $(\hat{J}_2, \hat{J}_3)$ and the entire problem $(\hat{J}_1, \hat{J}_2, \hat{J}_3)$ from each other. For almost all computed Pareto optimal parameters, the errors are in a range of $10^{-4} - 10^{-6}$, which also confirms the convergence result for the TR-RB algorithm from Theorem 3.2.4. Moreover, the errors are almost the same for the three different approaches. The spikes, which occur for all three TR-RB methods, indicate the points, for which the FE algorithm converged to the tip point of the lower right part of the Pareto front, while the TR-RB algorithms stayed in the upper left part.

In Figure 3.7(b), the dimension of the used RB space $V^r$ after solving each PSP is shown. Recall that for the subproblems with two cost functions, the PSP to every reference point is solved twice, once going along the Pareto front from top to bottom and once from bottom.
3.2 RB-Based Multiobjective Parameter Optimization

Figure 3.6: Approximations of the Pareto fronts of the MPOP from Section 2.2.4 for the FE and the TR-RB method.

As expected, the approach using one RB space yields a good approximation after some iterations, so that less updates are necessary. However, the downside is the large dimension of the RB space throughout the entire multiobjective optimization.

For the second approach of using one RB space with a restart before solving every PSP, the initial RB space $V_r$ has a dimension of 23. As one can see, some basis updates are necessary in almost every PSP, especially when solving PSPs corresponding to reference points of the problem ($\hat{J}_1, \hat{J}_2, \hat{J}_3$). Nevertheless, the advantage is that a relatively small number of basis functions is required to achieve the desired accuracy of the solution.

Using multiple local RB spaces is a hybrid between the first two approaches. On the one hand, using the same RB space for several optimization problems implies that it is a good approximation of the FE space for a certain parameter range. Thus, it can be used for quite a few optimization problems without having to update it. On the other hand, the RB spaces do not get too large to slow down the computation, since we bound the maximal dimension of the RB space. In this example, in total seven different RB spaces are generated. In Figure 3.8, the differently colored reference points indicate which of the RB spaces was used for which reference point. One can nicely see the different clusters for the different RB spaces, which underlines that they are of local nature. Note that the points in red are the ones excluded due to Remark 1.7.56.

Let us now compare the results for the TR-RB algorithms with the ones obtained by using a greedy algorithm to compute the RB space $V^r$ in an offline phase, cf. Section A.1.3.1.
3. MOR for PDE-Constrained Multiobjective Optimization

(a) Control errors $\|\bar{u} - \tilde{u}\|_U$ between the FE and the TR-RB solutions for the different TR-RB strategies.

(b) Dimension of the RB space $V^r$ for different TR-RB strategies.

(c) Control errors $\|\bar{u} - \tilde{u}\|_U$ between the FE and the RB solutions for the TR-RB one basis strategy and the greedy method.

Figure 3.7: Results for the different TR-RB strategies and the RB greedy method.

For the error indicator in the greedy algorithm, we choose

$$\Delta(V^r, u) := \max \left( \Delta_{st}, \max_{i \in \{1, \ldots, k\}} \Delta_{adj, i} \right),$$

where $\Delta_{st}$ and $\Delta_{adj, i}$ are a-posteriori error estimates for the errors in the state equation and the adjoint equation corresponding to $\hat{J}_i$, respectively. The three-dimensional parameter domain is discretized using 10 points per dimension, which results in a training set $\mathcal{U}_{train}$ containing 1000 distinct parameter values. For this training set, the greedy Algorithm 9 is run for the tolerances $\varepsilon_{\text{greedy}} \in \{0.1, 0.03, 0.01\}$, which results in RB spaces of dimension $\text{dim}(V^r) \in \{36, 42, 54\}$, respectively. These RB spaces are then used to solve the MOP by the augmented Lagrangian Algorithm 11 with the same parameter choices as for the numerical experiments in Section 2.2.4.
In Figure 3.7(c), the plots of the errors between the optimal parameters can be seen together with the ones for the TR-RB algorithm using a single RB space. We observe that the quality of the results of the TR-RB algorithm is comparable with the greedy method using the tolerance \( \varepsilon_{\text{greedy}} = 0.03 \), for which the RB space has 42 basis functions. However, when computing the RB space in an offline phase, we do not have any a-priori guarantee about the quality of the results, which is the major drawback of this approach.

By looking at Table 3.2, we see that all three TR-RB methods reduce the number of FE solves by a factor of around 20 in comparison to the FE method. However, in contrast to this reduction, the resulting reduction in computational time is relatively small (a factor of 1.7 – 2.4). There are several reasons for this:

First, as mentioned in the end of Section 2.2.4, the code has not been optimized with respect to performance. Since the resulting overhead is independent of whether the RB or the FE method is used, this reduces the computational speed-up of the RB-based methods with respect to the FE method. For this reason, even the method using a fixed RB space, which was computed in the offline-phase by the greedy algorithm in around 20 s, only leads to a speed-up factor of around 4. Note furthermore that this issue did not play a crucial role for the parabolic MOCP, since solving the (reduced-order) parabolic PDEs was a lot more costly than the resulting overhead.

Second, the TR-RB method is based on comparing the RB model with the FE model not only after every iteration by directly computing FE solutions, but also within solving the TR-RB subproblem (3.2.2) by using an a-posteriori error estimate. Although the evaluation of the a-posteriori error estimate only depends on the dimension of the RB space, it still requires a significant amount of computational time, since it has to be evaluated very often.

This is directly connected to the third reason, namely that not only the reduction of FE solves but also the dimension of the RB space and the number of iterations have a big influence on the computational time. In fact, in the TR-RB algorithm, one has to find a balance between these two points. On the one hand, having a large RB space usually yields a good approximation of the FE cost function and its gradient. Thus, relatively few
Table 3.2: Computational time, speed-up with respect to the FE-method, average number of iterations and the number of FE solves for the different methods.

<table>
<thead>
<tr>
<th>Method</th>
<th>Time [s]</th>
<th>Speed-Up</th>
<th>Avg. # it.</th>
<th># FE solves</th>
</tr>
</thead>
<tbody>
<tr>
<td>FE</td>
<td>1357</td>
<td>–</td>
<td>19.5</td>
<td>435853</td>
</tr>
<tr>
<td>TR-RB One Basis</td>
<td>789</td>
<td>1.7</td>
<td>24.8</td>
<td>20738</td>
</tr>
<tr>
<td>TR-RB One Basis Restart</td>
<td>636</td>
<td>2.1</td>
<td>26.5</td>
<td>21664</td>
</tr>
<tr>
<td>TR-RB Local Basis, $r_{\text{max}} = 45$</td>
<td>566</td>
<td>2.4</td>
<td>25.6</td>
<td>20803</td>
</tr>
<tr>
<td>Greedy $\varepsilon_{\text{greedy}} = 0.03$ ($r = 42$)</td>
<td>350</td>
<td>3.9</td>
<td>19.1</td>
<td>47</td>
</tr>
</tbody>
</table>

Iterations of the TR-RB algorithm can be expected. However, due to the high dimension of the RB space, these iterations are quite costly. On the other hand, having a small RB space implies less computational effort for a single iteration, but possibly increases the number of iterations, since the FE cost function and its gradient are badly approximated at the beginning of the optimization. These two conflicting issues can be observed by looking at the results for the approaches of using one basis and of using one basis with a restart before every optimization. While the first one needs in average less iterations, the RB space of the second one has a much smaller dimension, which leads to the iterations being faster. In the end, we observe that restarting the basis pays off in terms of computational time. Using multiple local RB spaces is a hybrid of these two approaches, since the dimension of the RB spaces is kept quite small, while it still yields a decent approximation of the FE cost function and its gradient in the beginning of the optimization. For this example, we see that this hybrid strategy works the best.

In conclusion, we observe that all three TR-RB methods can be successfully applied to the present MPOP, since, on the one hand, their results can be qualitatively compared to those of the FE method, and, on the other hand, they reduce the number of FE solves by a factor of around 20. Nevertheless, we also see that a careful handling of the RB space is needed to gain a satisfying computational speed-up when using the TR-RB algorithms. Since the TR-RB algorithm involves computing a-posteriori error estimates within the iterations and solving the state and adjoint equation by the FE method after every iteration, it is clear that the reduction of the computational time cannot be of the same order as for a fixed RB space, which was computed by a greedy algorithm in an offline-phase. However, as already mentioned, we have no control of the approximation error if we use a greedy algorithm for constructing the RB space. Moreover, for higher dimensional parameters spaces, a standard greedy algorithm is not feasible anymore, since a representative training set $\mathcal{U}_{\text{train}}$ of the admissible set $\mathcal{U}_{\text{ad}}$ is needed. In this case, the use of adaptive greedy procedures (see, e.g., [HSZ14]) is required. In contrast to this, the TR-RB algorithm can in its present form also be efficiently used for high-dimensional parameter spaces, see, e.g., [BKM⁺20, KMO⁺20].
Conclusion and Outlook

In this thesis, we have investigated three different scalarization methods for solving both convex and non-convex MOPs with an arbitrary number of cost functions. We have shown that the resulting algorithms can be successfully applied to a linear-quadratic MOCP and a non-convex MPOP. Since the computational effort for solving these problems is high when using standard FE-based discretizations of the PDE, we have also studied the application of MOR techniques. Here our focus was on extending methods from the literature to the context of PDE-constrained multiobjective optimization with the goals of significantly reducing the computational costs while guaranteeing a small error between the reduced-order and the full-order solutions throughout the multiobjective optimization.

The main results of Chapter 1 are the two (numerical) hierarchical algorithms for solving MOPs with an arbitrary number of cost functions by the ERPM and the PSM, cf. Algorithms 1 & 2 and Algorithms 3 & 4, respectively. The idea for these algorithms is based on the hierarchical structure of the Pareto front, which was already used in [KSd15, MAL12, MGGS09] to develop similar algorithms for the PSM.

Given a convex framework for the ERPM, we have used the connection between an extended ERPP and the WSP to prove a characterization of a Pareto sufficient set of reference points by using solutions of subproblems, cf. Theorem 1.6.38. This characterization served as a basis for designing the hierarchical algorithm and allowed us to prove analytically that it computes the entire Pareto front, cf. Theorem 1.6.63. In a numerical implementation, we have used a grid structure for the reference points. Since the characterization of reference points can be lead back to the componentwise comparisons of points in Corollary 1.6.65, this allows for an efficient computation of reference points for an arbitrary number of cost functions in practice. Moreover, together with the Lipschitz continuity of the solution mapping of the ERPP, this implies a guaranteed coverage of the Pareto front, which can be specified by choosing the grid size of the reference point grid, see Remark 1.6.70.

For the PSM, we have proceeded in a similar way and obtained sufficient and necessary conditions for a Pareto sufficient set of reference points, which are only based on solutions of subproblems, cf. Corollary 1.7.53. However, in the general non-convex case, there is a gap between the necessary and sufficient condition. In practice, this implies that potentially a superset of the Pareto sufficient set of reference points is computed. However, for the convex framework we have shown in Corollary 1.7.62 that this gap is closed if Assumption 1.4 is satisfied. This assumption is hard to verify and we have observed that
it is often not satisfied in practice. However, we have seen that the algorithm still works in this case and only suffers from being less efficient. For the resulting algorithm, we could analytically show in Theorem 1.7.63 that it computes the entire Pareto front. In the same way as for the ERPM, the grid structure of the reference points allowed us to use simple componentwise comparisons in the selection procedure of the reference points. In particular, all relevant reference points are selected in a numerical implementation, cf. Theorem 1.7.68. By using this result, we could also show a guaranteed coverage of the Pareto front in Remark 1.7.69, which can be specified by choosing the grid size of the reference point grid.

To summarize, the results in [KSd15, MAL12, MGGS09] were modified and extended in the following ways:

- We could show analytically that the entire Pareto front is computed by the hierarchical algorithm. Such a result is missing in the above mentioned publications. Moreover, in [KSd15, MGGS09], the authors made the assumption that the Pareto front is connected. In contrast to this, our proposed algorithms only require assumptions on the cost functions, which are easier to verify in practice. In particular, the assumptions for the PSM allow for a disconnected Pareto front, which can therefore also be dealt with by our algorithm.

- As described above, we use componentwise comparisons for computing the reference points, which is computationally cheap. In constrast to this, linear programs were used in [KSd15, MGGS09] and a so-called Centroidal Voronoi Tessellation was used in [MAL12] for the selection process, which can both become costly for a larger number of objective functions. Moreover, the authors did not show that all relevant reference points can be obtained.

At this point, note that the hierarchical algorithms for the ERPM and the PSM can be parallelized in two ways for both our approach and the ones in [KSd15, MAL12, MGGS09]: Assume that all subproblems with at most \( l - 1 \) cost functions have been solved by the ERPM or the PSM. Then all subproblems with \( l \) cost functions can be solved in parallel, since this only depends on data from the previously solved subproblems with at most \( l - 1 \) cost functions. Moreover, all reference points are computed at once for a given subproblem. Thus, solving the ERPPs and PSPs within one subproblem can also be parallelized.

In numerical examples with polynomial cost functions, we have verified the analytical results for the two hierarchical algorithms. In particular, the guaranteed coverage of the Pareto front could be confirmed. Moreover, we have seen that the ERPM and the PSM outperform the WSM in terms of coverage and uniformity of the approximation of the Pareto front for both examples. In fact, despite the lack of analytical results, the ERPM can still produce quite good approximations of the Pareto front in the non-convex example, which are clearly superior to those of the WSM.
Two model problems for PDE-constrained MOPs were introduced and studied in Chapter 2.

The first one is a linear-quadratic MOCP of a parabolic diffusion-advection equation. Numerical experiments for a problem with three cost functions revealed that the ERPM and the PSM outperform the WSM in terms of the approximation quality of the Pareto front. However, due to the complicated structure of the PSPs, the computational time of the PSM is clearly higher than for the ERPM, so that the ERPM is to be favored in this example.

The second model problem is a non-convex MPOP, where the governing PDE is an elliptic diffusion-reaction equation. For a numerical example with three cost functions, we have seen that the WSM fails to approximate the non-convex parts of the Pareto front in this case. Similarly to the non-convex example from Chapter 1, the ERPM could still produce a good approximation of the Pareto front. However, the approximation quality was very sensitive with respect to the chosen distance of the reference points from the Pareto front. Thus, a good approximation of the Pareto front cannot always be guaranteed for the ERPM. For the PSM, we have observed that it approximates the Pareto front nicely as predicted by the theory if the global minimizers of the arising PSPs can be computed. Here we have observed that the utilized augmented Lagrangian penalty algorithm is very sensitive with respect to the choice of the initial value. In particular, even by choosing the initial value relatively close to the global minimizer, it might still happen that a local minimizer is computed. This can be explained by the augmentation of the Lagrangian function, which enables jumps in the parameter space.

The application of MOR to the two model problems was the focus of Chapter 3.

For the application of the POD method to the linear-quadratic MOCP from Section 2.2, we could extend already known a-priori and a-posteriori results for the WSP and the ERPP to the PSP. This allowed us to design an algorithm for controlling the approximation error between the full-order and the reduced-order solutions of the arising scalarized optimization problems by combining the a-posteriori error estimates with a simple update scheme for the POD basis. The proposed algorithm has reduced the computational time for all three scalarization methods significantly in comparison to the FE method while the given error tolerance is satisfied at the same time. However, since the a-posteriori error estimate for the PSP is shown both analytically and numerically to be less efficient in comparison to the ones for the WSP and the ERPP – especially if the error between the full-order and the reduced-order solution is small – the error tolerance for the PSM has to be chosen larger than the ones for the WSM and the ERPM in practice.

For the non-convex MPOP, we have presented and described the TR-RB algorithm from [BKM+20]. We have shown that the convergence result can be applied to all three scalarization methods. In numerical experiments, this has been verified for the PSM. Additionally, we have investigated three different approaches of handling the RB space during the multiobjective optimization. Here it turned out that the use of several local
RB spaces can be beneficial in terms of computational time, since they allow for a good approximation with a reasonably small basis size.

**Outlook**

There are several arising interesting questions and tasks that might be the subject of further studies.

For the ERPM, choosing the reference points on shifted coordinate planes turned out to produce less uniform approximations of the Pareto front in comparison to an iterative procedure for two cost functions. Thus, one could investigate if this iterative procedure can be generalized to more than two cost functions by using the analytical results shown in Section 1.6. Alternatively, one could also try to extend the approach presented here to the case when the set $D$ is a hyperplane, see [Eic08, KSd15, MAL12, MGGS09].

We have observed that the augmented Lagrangian penalty method for solving the PSPs is very sensitive with respect to the choice of the initial value for the optimization in the non-convex case. This leads to problems concerning the computation of the global minimizer. Thus, the implementation of a global optimization algorithm would be beneficial.

In a next step, the PSM could be applied to non-smooth PDE-constrained optimization problems, which arise, e.g., when using the max-PDE, cf. [Ber19]. From an analytical point of view, differentiability of the objective functions is not required, so that all theoretical results for the hierarchical algorithm also hold for non-smooth problems satisfying the non-convex framework. One occurring issue in practice might be to solve the arising PSPs, since one has to combine non-smooth objective functions with a non-smooth scalarization function.

For the certified error control presented in Algorithm 7 for the linear-quadratic MOCP, it would be interesting to test other methods – such as, e.g., optimality-system POD ([KV08]) – for updating the POD basis than the one proposed in Algorithm 6. Moreover, one could try to extend the proposed TR-RB algorithm to the application of the POD method to MOCPs. From a theoretical point of view, the infinite-dimensional control space poses a problem that has to be tackled. In a practical algorithm, the error estimator of the cost function, which we use here inside the TR-RB subproblems, is a potential problem, since the overestimation of a-posteriori error estimates for parabolic problems can be quite high.

In [BKM+20, KMO+20], a non-conforming dual approach was used in the RB approximation, i.e., different RB spaces were used for the state and adjoint equation. To overcome the resulting inconsistency in the computation of the gradients, an additional error term has to be added to the cost function. The advantage of this approach is that lower-dimensional RB spaces can be used, since they only need to be good approximations for the state or the adjoint equation, respectively. Thus, one could investigate the application of this approach to the MPOP. In particular, the combination of using the non-conforming dual approach and the use of local RB spaces could be interesting.
Model-Order Reduction

Nowadays, the increase in computational power allows for more and more complex problems to be solved [SvdVR08]. A good example for this observation is the field of PDE-constrained optimization [HPUU09, Trö10]. When optimizing with PDE constraints, it is inevitable to solve the underlying PDE numerous times for different input parameters. Even on modern computers, this can often not be done in an acceptable amount of time. One idea to circumvent this problem is the use of model-order reduction [SvdVR08]. In there, the high-dimensional model of the PDE is replaced by a low-dimensional surrogate model based on previously obtained data from the high-dimensional model. In contrast to the FE space, which is supposed to be a good approximation of the entire function space, the low-dimensional reduced-order space aims only at approximating the characteristics of solutions of the PDE.

In this chapter, we give an overview on two techniques for model-order reduction: The reduced basis (RB) method [HRS16, PR07, QMN16, RHP07] and proper orthogonal decomposition (POD) [HLBR12, KV99, LC18, Pin08, Sir87].

A.1 The Reduced Basis Method

Especially in the field of parametrized PDEs, the RB method has become a popular tool for model-order reduction [Haa17, RHP07]. It was first studied in the 1980s [FR83, NP80] and has attracted much attention since then. Since the early 2000s, more and more publications have focused not only on reduction but also on (a-posteriori) error control of the RB method [DH15, Gre05]. This allowed the method to be applied to various problems, including PDE-constrained parameter optimization problems [KG14a, KG14b, QGVW17].

In Section A.1.1, we introduce the RB method and show in Section A.1.2 how it can be applied to an elliptic PDE-constrained parameter optimization problem. In particular, we also present a-posteriori error estimates for the relevant quantities. Finally, in Section A.1.3, we describe two different approaches for computing a reduced basis in practice.

Note that we strongly follow [Haa17, Section 2] in this section.
A.1.1 The RB Method for Parametrized Elliptic PDEs

We present the basic idea of the RB method by looking at the elliptic model problem (2.2.1) from Section 2.2. For this problem, we derived its general weak formulation (for details we refer to Section 2.2.1): Given a parameter $u \in \mathcal{U} := \mathbb{R}^m \times \mathbb{R}$ with $u > 0$, find $y \in V := H^1(\Omega)$ such that

$$a(u; y, \varphi) = F(\varphi) \quad \text{for all } \varphi \in V. \quad \text{(A.1.1)}$$

Furthermore, we showed that there exists a set $\mathcal{U}_{eq} \subset \mathcal{U}$ such that (A.1.1) has a unique solution $y = y(u) \in V$ for all parameters $u \in \mathcal{U}_{eq}$, which allowed us to define the solution operator $S : \mathcal{U}_{eq} \to V$. Thus, we can define the so-called solution manifold $\mathcal{M} := \{S(u) \mid u \in \mathcal{U}_{eq}\}$.

The idea of the RB method is to find a low-dimensional subspace $V^r \subset V$, which is a good approximation of the solution manifold $\mathcal{M}$. This subspace is generated by using snapshots, i.e., solutions $S(u_1), \ldots, S(u_r)$ of the equation (A.1.1) to different parameter values $u_1, \ldots, u_r \in \mathcal{U}_{eq}$, so that we have $V^r = \text{span}\{S(u_1), \ldots, S(u_r)\}$. If the snapshots $S(u_1), \ldots, S(u_r)$ are linearly independent, the space $V^r$ is $r$-dimensional and the snapshots trivially form a basis. Note that in practice, one often performs an orthonormalization of the snapshots $S(u_1), \ldots, S(u_r)$ for numerical stability. For more details on the computation of the RB space $V^r$ and, in particular, on the choice of the parameters $u_1, \ldots, u_r \in \mathcal{U}_{eq}$, we refer to Section A.1.3.

Given an RB space $V^r$, we obtain the reduced-order version of (A.1.1) by a Galerkin projection: Given a parameter $u \in \mathcal{U}_{eq}$, find $y^r \in V^r$ such that

$$a(u; y^r, \psi) = F(\psi) \quad \text{for all } \psi \in V^r. \quad \text{(A.1.2)}$$

For the reduced-order equation, we can also show unique solvability for all parameters $u \in \mathcal{U}_{eq}$.

**Theorem A.1.1.** For any $u \in \mathcal{U}_{eq}$ there is a unique solution $y^r = y^r(u) \in V^r$ of (A.1.2). Moreover, the estimate

$$\|y^r\|_V \leq C \|F\|_V, \quad \text{(A.1.3)}$$

holds with a constant $C > 0$, which can be chosen independently of $u$, $F$ and $r$.

**Proof.** This can be concluded with the same arguments as Theorem 2.2.2. \hfill \Box

Theorem A.1.1 allows us to define a solution operator for the reduced-order equation.

**Definition A.1.2.** Define the solution operator $S^r : \mathcal{U}_{eq} \to V^r$, which maps any parameter $u \in \mathcal{U}_{eq}$ to the unique solution $y^r = S^r(u) \in V^r$ of (A.1.2).
A.1 The Reduced Basis Method

An important tool for both the generation of a reduced basis and the use of the RB method in PDE-constrained optimization is the a-posteriori estimation of the error between the full-order and the reduced-order solution. The following result shows such an a-posteriori error estimate for the state equation.

**Theorem A.1.3.** Let \( u \in U_{eq} \) be arbitrary and denote by \( \alpha(u) \) the coercivity constant of the bilinear form \( a(u; \cdot, \cdot) \). By Remark 2.2.4, it holds \( \alpha(u) \geq \alpha_{\text{min}} > 0 \). Let the residual \( r_{st}(u; \cdot) \in V' \) be given by

\[
r_{st}(u; \varphi) := F(\varphi) - a(u; S^r(u), \varphi)
\]

for all \( \varphi \in V \). Then it holds

\[
\|S(u) - S^r(u)\|_V \leq \Delta_{st}(u) := \|r_{st}(u; \cdot)\|_{V'} / \alpha(u). \tag{A.1.4}
\]

**Proof.** A proof of this statement can be found in [Haa17, Proposition 2.19].

**Remark A.1.4.**

(i) Note that we only need the reduced-order state \( S^r(u) \), but not the full-order state \( S(u) \), for evaluating the a-posteriori error estimate (A.1.4).

(ii) For our example, the computation of the coercivity constant \( \alpha(u) \) is cheap, see Lemma 2.2.1. In more general examples, this might not be the case. Thus, one often uses a quickly computable lower bound \( \alpha_{LB}(u) \) instead. Possible methods for computing such a lower bound are, e.g., the min-theta approach (cf. [Haa17, Proposition 2.35]) or the Successive Constraint Method (SCM) (cf. [RHP07, Section 10]).

(iii) In practice, the computation of the term \( \|r_{st}(u; \cdot)\|_{V'} \) poses a problem. In the infinite-dimensional setting, it is in general not possible to compute this norm. Moreover, even if the space \( V \) is a finite-dimensional FE space, the numerical costs for computing \( \|r_{st}(u; \cdot)\|_V \) depend on the dimension of the FE space, which is not desirable in the context of RB. However, in our case, one can make use of the so-called parameter-separability of the bilinear form \( a(u; \cdot, \cdot) \) to precompute certain quantities such that the computational costs for evaluating \( \|r_{st}(u; \cdot)\|_{V'} \) only depend on the dimension of the RB space. For details, we refer to, e.g., [RHP07, Section 9.4].

A.1.2 Application of the RB Method to PDE-Constrained Parameter Optimization Problems

Given a reduced basis \( V^r \), we now show how the RB method can be applied to the PDE-constrained parameter optimization problem (2.2.7) from Section 2.2.2. The idea is simply to replace the solution operator \( S \) of the full-order equation by the reduced-order solution operator \( S^r \) in the cost function \( \hat{J} \), cf. Definition 2.2.8.
Definition A.1.5. We define the essential reduced-order cost function \( \hat{J}_r : U_{eq} \to \mathbb{R} \) by
\[
\hat{J}_r(u) := J(S^r(u), u) = \frac{\sigma_\Omega}{2} \|S^r(u) - y_\Omega\|_H^2 + \frac{\sigma}{2} \|u\|_U^2.
\]
If Assumption 2.2 is satisfied, this allows us to introduce the reduced-order parameter optimization problem
\[
\min_{u \in U_{eq}} \hat{J}_r(u) \quad \text{s.t.} \quad u \in U_{ad}.
\] (A.1.5)

Similarly to Section 2.2.2.1, we can introduce the adjoint equation to find an evaluable representation of the gradient of the cost function \( \hat{J}_r \).

Definition A.1.6. For \( u \in U_{eq} \) we define the reduced-order adjoint equation by
\[
a(u; \psi, p^r) = \langle \sigma_\Omega(S^r(u) - y_\Omega), \psi \rangle_H \quad \text{for all} \quad \psi \in V^r.
\] (A.1.6)

Theorem A.1.7. For any \( u \in U_{eq} \) there is a unique solution \( p^r = p^r(u) \in V^r \) of (A.1.6). Moreover, the estimate
\[
\|p^r\|_V \leq C \|u\|_U
\]
holds with a constant \( C > 0 \), which is independent of \( u \) and \( r \).

Proof. The unique solvability can be shown with the same arguments as in Theorem 2.2.13. Moreover, by using the same theorem together with the a-priori estimate (A.1.3) of the reduced-order state equation, the a-priori estimate can be shown.

Definition A.1.8. We denote the continuous solution operator of the reduced-order adjoint equation by \( A^r : U_{eq} \to V \). Note that the continuity constant can be chosen independently of \( r \).

Corollary A.1.9. Let \( u \in U_{eq} \) be arbitrary. Then it holds
\[
\nabla \hat{J}_r(u) = -\partial_u a(u; S^r(u), A^r(u)) + \sigma_\Omega u \in U.
\]

Proof. This follows with the same arguments as Lemma 2.2.16.

In the last part of this section, we focus on the a-posteriori error estimation for the adjoint equation, the cost function and its gradient.

Theorem A.1.10. Let \( u \in U_{eq} \) be arbitrary and denote by \( \alpha(u) > 0 \) the coercivity constant of the bilinear form \( a(u; \cdot, \cdot) \). The residual \( r_{adj}(u; \cdot) \in V^r \) of the adjoint equation is given by
\[
r_{adj}(u; \varphi) := \langle \sigma_\Omega(S^r(u) - y_\Omega), \varphi \rangle_H - a(u; \varphi, A^r(u)).
\]
for all \( \varphi \in V \). Then it holds
\[
\|A(u) - A^r(u)\|_V \leq \Delta_{adj}(u) := \frac{\|r_{adj}(u; \cdot)\|_{V^r} + \sigma_\Omega \Delta_{st}(u)}{\alpha(u)}.
\]
Proof. For a proof of this statement we refer to, e.g., [KMO+20, Proposition 2.4].

Similarly, there is also an a-posteriori error estimate for the error in the cost function and its gradient.

**Theorem A.1.11.** Let \( u \in \mathcal{U}_{eq} \) be arbitrary. Then it holds
\[
\left| J(u) - \hat{J}(u) \right| \leq \Delta_{st}(u) \left\| r_{adj}(u; \cdot) \right\|_{V^r} + \sigma_{\Omega} \Delta_{st}(u)^2 =: \Delta_{Jr}(u).
\]
Moreover, we have
\[
\left\| \nabla \hat{J}(u) - \nabla \hat{J}(u) \right\|_{\mathfrak{H}} \leq \left\| \partial_u a(u; \cdot, \cdot) \right\| \left( \left\| S^r(u) \right\|_{V^r} \Delta_{adj}(u) + \Delta_{st}(u) \Delta_{adj}(u) \right) + \Delta_{st}(u) \left\| A^r(u) \right\|_{V^r}) =: \Delta_{\nabla Jr}(u).
\]

**Proof.** A proof of the a-posteriori error estimate for the cost function can be found in [KMO+20, Proposition 2.5]. The estimate for the gradient can be derived similarly to [KMO+20, Proposition 2.7] or [BGD+19, Section 3.3].

### A.1.3 Basis Generation

So far, we did not specify how the reduced basis \( V^r \) can be computed in practice. As already stated, the most common way is to compute snapshots \( S(u_1), \ldots, S(u_r) \) to certain parameter values \( u_1, \ldots, u_r \in \mathcal{U}_{eq} \) and define \( V^r \) as the span of these snapshots. In PDE-constrained optimization, it is also of interest to have a good approximation of the cost function and its gradient, since most optimization algorithms work with gradient information. So, in order to obtain a good approximation of the solution of the full-order optimization problem, it is often inevitable to have an RB space, which approximates the gradient of the cost function well. Therefore, in these applications, it is necessary to not only include snapshots of the state, but also of the adjoint equation.

In this section, we discuss the choice of the parameters and the corresponding snapshots which are used for generating the basis. In particular, we present two different approaches: the greedy algorithm and an adaptive basis computation. Both approaches have in common that they need an error indicator \( \Delta(V^r, u) \), which represents an approximation error at the parameter \( u \) when using the RB space \( V^r \). This error indicator strongly depends on the application and should be chosen carefully. A simple example for this error indicator could be, e.g., the error in the state \( \left\| S(u) - S^r(u) \right\|_{V^r} \), or the a-posteriori estimate \( \Delta_{st}(u) \) for this error. In the framework of PDE-constrained optimization, where also the approximation of the cost function and its gradient is of interest, the error indicator should also contain the error of the adjoint equation. This could be realized by choosing \( \Delta(V^r, u) = \max (\left\| S(u) - S^r(u) \right\|_{V^r}, \left\| A(u) - A^r(u) \right\|_{V^r}) \), or \( \Delta(V^r, u) = \max (\Delta_{st}, \Delta_{adj}) \), respectively.
A.1.3.1 The Greedy Algorithm

The idea of the greedy algorithm ([VPRP03, BMP+12]) is to find a subspace $V^r$, which approximately solves the minimization problem

$$\min_{V^r \subset V} \max_{w \in \mathcal{U}_{ad}} \Delta(V^r, u)$$

for a given dimension $r$. To this end, a finite training set $\mathcal{U}_{train} \subset \mathcal{U}_{ad}$ is defined, which is supposed to be a good approximation of the parameter set $\mathcal{U}_{ad}$. For computing the first RB space $V^1$, a random parameter $u^1 \in \mathcal{U}_{train}$ is chosen and the corresponding snapshots are computed. Given an RB space $V^r$, we look at the maximal error indicator over all training parameters $\max_{u \in \mathcal{U}_{train}} \Delta(V^r, u)$. As long as this is greater than a predefined error tolerance $\varepsilon_{\text{greedy}}$, the parameter for which the error indicator is the largest is chosen and snapshots corresponding to this parameter are added to the basis. This procedure is summarized in Algorithm 9.

**Algorithm 9 Greedy Algorithm**

**Require:** Parameter training set $\mathcal{U}_{train} \subset \mathcal{U}_{ad}$, error indicator $\Delta(V^r, u)$, desired error tolerance $\varepsilon_{\text{greedy}}$

1. Choose $u_1 \in \mathcal{U}_{train}$ randomly and compute $y_1 := \mathcal{S}(u_1)$;
2. Define $\tilde{y}_1 := y/\|y\|_V$;
3. Set $\Psi^1 := \{\tilde{y}_1\}$ and $V^1 := \text{span}\{\tilde{y}_1\}$;
4. Set $r := 1$;
5. **while** $\max_{u \in \mathcal{U}_{train}} \Delta(V^r, u) > \varepsilon_{\text{greedy}}$ **do**
6. Set $r \leftarrow r + 1$;
7. Determine $u_r \in \arg \max_{u \in \mathcal{U}_{train}} \Delta(V^r, u)$;
8. Compute $y_r := \mathcal{S}(u_r)$;
9. Denote by $\tilde{y}_r$, the orthonormalization of $y_r$ w.r.t. $V^{r-1}$;
10. Set $\Psi^r \leftarrow \Psi^{r-1} \cup \{\tilde{y}_r\}$ and $V^r \leftarrow V^{r-1} \oplus \text{span}\{\tilde{y}_r\}$;
11. **end while**

**Remark A.1.12.** (i) Since the finite training set $\mathcal{U}_{train}$ is supposed to be a good approximation of the parameter set $\mathcal{U}_{ad}$, the standard greedy algorithm is only practical for low-dimensional parameter sets. Nevertheless, the extension to higher-dimensional parameter sets is possible by using adaptive greedy procedures, cf. [HSZ14].

(ii) If the error indicator contains the exact errors of the involved quantities, the greedy algorithm is called strong. On the other hand, if the error indicator only uses a-posteriori error estimates, the greedy algorithm is called weak. Consequently, a strong greedy algorithm is computationally more expensive than a weak greedy algorithm, since the full-order quantities need to be computed explicitly. However, it is probable that the RB space generated with a strong greedy algorithm has a smaller dimension than the RB space of a weak greedy algorithm because of the overestimation of the a-posteriori estimates. In particular, if one wants to use a weak greedy algorithm, it is important to have a reasonably sharp a-posteriori estimate.
A.1 The Reduced Basis Method

(iii) Using a greedy algorithm is in the typical spirit of the RB method of splitting the computation into an offline and an online phase. In the offline phase, the expensive calculations, i.e., the greedy algorithm, are performed, while in the online phase, only fast reduced-order solves are necessary.

(iv) As described in the beginning of this section, it is sometimes also necessary to add solutions of the adjoint equation to the reduced basis. In this case, one also computes \( p_1 := A(u_1) \) and \( p_r := A(u_r) \) in lines 1 and 8 and adds these snapshots to the basis as well in lines 3 and 10.

A.1.3.2 Adaptive Basis Computation

Lately, more and more attention was drawn to adaptively computing a reduced basis, especially in the field of PDE-constrained optimization (cf., e.g., [KMO + 20, QGVW17, YM13]). The idea is that during an optimization procedure, the RB cost function only needs to be a good approximation of the full-order cost function along the optimization path. At the current iterate, this means that it must only be accurate in a local neighborhood of the current parameter value. Therefore, it is actually not necessary to perform a costly greedy algorithm, which ensures that the error is small in the entire parameter domain. Algorithm 10 visualizes how this approach can be incorporated into a general optimization algorithm. Note that we directly include solutions of the adjoint equation to the reduced basis. The reason is that one usually also wants to have a good approximation of the gradient of the cost function in these approaches, which can only be obtained if solutions of the adjoint equation are also included into the RB space.

Algorithm 10 Adaptive Basis Computation

Require: Initial value for the optimization \( u_o \), error indicator \( \Delta(V_r, u) \), desired error tolerance \( \varepsilon_{\text{adap}} \)

1: Compute \( y_o := \mathcal{S}(u_o), p_o := \mathcal{A}(u_o) \) and the orthonormalizations \( \tilde{y}_o \) and \( \tilde{p}_o \);
2: Set \( \Psi^r := \{ \tilde{y}_o, \tilde{p}_o \} \) and \( V_r := \text{span}\{\tilde{y}_o, \tilde{p}_o\} \);
3: Set \( i := 0 \) and \( u^{(0)} := u_o \);
4: while Termination criteria for optimization not fulfilled do
5: Compute next iterate \( u^{(i+1)} \) with the RB cost function based on \( V_r \);
6: if \( \Delta(V_r, u^{(i+1)}) > \varepsilon_{\text{adap}} \) then
7: Compute \( y_{i+1} := \mathcal{S}(u^{(i+1)}) \) and \( p_{i+1} := \mathcal{A}(u^{(i+1)}) \);
8: Denote by \( \tilde{y}_{i+1} \) the orthonormalization of \( y_{i+1} \) w.r.t. \( V_r \);
9: Set \( \Psi^r := \Psi^r \cup \{\tilde{y}_{i+1}\} \) and \( V^r := V^r \oplus \text{span}\{\tilde{y}_{i+1}\} \);
10: Denote by \( \tilde{p}_{i+1} \) the orthonormalization of \( p_{i+1} \) w.r.t. \( V_r \);
11: Set \( \Psi^r := \Psi^r \cup \{\tilde{p}_{i+1}\} \) and \( V^r := V^r \oplus \text{span}\{\tilde{p}_{i+1}\} \);
12: end if
13: end while

Remark A.1.13. (i) As for the greedy algorithm, there is again a weak and a strong version of this approach: In the weak version, a-posteriori estimates are used as indicator, which avoid the computation of the expensive full-order data, whereas the error indicator uses the true errors in the strong version.
(ii) Note that this approach avoids the typical online/offline decomposition of the RB method. It only uses full-order evaluations inside the optimization routine in order to guarantee the exactness of the RB cost function. Therefore, this approach is also suitable for high-dimensional parameter spaces.

\[\Box\]

## A.2 Proper Orthogonal Decomposition

Proper orthogonal decomposition (POD), which is also known by the names of Karhunen-Loève decomposition or principal component analysis, is a widely used model-order reduction technique for dynamical systems. In the 1960s, it was introduced by Lumley [Lum67] in the context of identifying coherent structures in turbulent flow. Given an arbitrary data set, the POD method computes an optimal orthogonal basis of the subspace spanned by the data in a certain least-squares sense, which will be explained in this chapter.

Sirovich [Sir87] introduced the idea of using snapshots, i.e., spatial solutions of the PDE at certain time instances, as data for computing the POD space. Therefore, the POD method can be seen as a particular method for computing the RB space in case that a whole set of snapshots is available at the same time. This occurs, in particular, for time-dependent PDEs, for which one obtains not only one snapshot, but a whole trajectory of snapshots when solving the PDE for one input parameter. Thus, the POD method is widely used for model-order reduction of dynamical systems, see, e.g., [GHV21, HLBR12, LC18, Pin08]. Moreover, the POD method is also successfully applied to PDE-constrained optimization problems, cf. [GV17, HV08, TV09].

In Section A.2.1, we introduce the POD method itself, before we show how to apply it to linear evolution equations in Section A.2.2 and to linear-quadratic optimal control problems in Section A.2.3. Note that this introduction is oriented at [GV17] and strongly follows the outline of Section A.1 for the RB method.

### A.2.1 The POD Method

Let $\mathcal{X}$ be a separable Hilbert space, $T > 0$ a final time and $y_1, \ldots, y_p \in L^2(0, T; \mathcal{X})$ be given snapshots, e.g., solutions to a PDE for different input parameters. Then we define the linear subspace

$$ V := \left\{ y \in \mathcal{X} \mid \exists \omega_1, \ldots, \omega_p \in L^2(0, T) : y = \sum_{k=1}^p \int_0^T \omega_k(t) y^k(t) \, dt \right\} $$

and call it the *snapshot subspace* with dimension $d \in \mathbb{N} \cup \{\infty\}$. Given $\ell \in \{1, \ldots, d\}$, the idea of the POD method is to find a subspace $V^\ell \subset \mathcal{X}$ with dimension $\ell$, represented by an orthonormal basis $\{\psi_i\}_{i=1}^\ell$, such that the mean square error between the snapshots $y_1, \ldots, y_p$ and their projections onto the subspace $V^\ell$ is minimized. Mathematically, the orthonormal basis can be computed by solving the minimization problem

$$ \min_{\psi_1, \ldots, \psi_\ell \in \mathcal{X}} \sum_{k=1}^p \int_0^T \left\| y^k(t) - \sum_{i=1}^\ell \langle y^k(t), \psi_i \rangle_{\mathcal{X}} \psi_i \right\|_{\mathcal{X}}^2 \, dt, \quad \text{s.t.} \quad \langle \psi_i, \psi_j \rangle_{\mathcal{X}} = \delta_{ij} \text{ for } 1 \leq i, j \leq \ell. \quad (A.2.1) $$
**Definition A.2.1.** A solution \( \{ \psi_i \} \) of the minimization problem (A.2.1) is called a POD basis of rank \( \ell \). The corresponding subspace \( V^\ell := \text{span}\{ \psi_1, \ldots, \psi_\ell \} \) is called a POD subspace of rank \( \ell \).

As it turns out, we can compute a POD basis of rank \( \ell \) for any \( \ell \in \{ 1, \ldots, d \} \) by looking at the eigenfunctions of the operator \( \mathcal{R} : \mathcal{X} \to \mathcal{X} \) defined by

\[
\mathcal{R} \psi := \sum_{k=1}^{p} \int_0^T \langle y^k(t), \psi \rangle_X y^k(t) \, dt
\]

for all \( \psi \in \mathcal{X} \).

**Theorem A.2.2.** The operator \( \mathcal{R} \) is compact, non-negative and self-adjoint. Moreover, there are non-negative eigenvalues \( \{ \bar{\lambda}_i \} \in \mathbb{N} \) and associated orthonormal eigenfunctions \( \{ \bar{\psi}_i \} \in \mathbb{N} \), which form a basis of \( \mathcal{X} \), such that

\[
\mathcal{R} \bar{\psi}_i = \bar{\lambda}_i \bar{\psi}_i \quad (i \in \mathbb{N}), \quad \bar{\lambda}_1 \geq \cdots \geq \bar{\lambda}_d > \bar{\lambda}_{d+1} = \cdots = 0.
\]

For every \( \ell \in \{ 1, \ldots, d \} \), the first \( \ell \) eigenfunctions \( \{ \bar{\psi}_i \}^\ell_{i=1} \) are a POD basis of rank \( \ell \).

Lastly, we have the identity

\[
\sum_{k=1}^{p} \int_0^T \left\| y^k(t) - \sum_{i=1}^\ell \langle y^k(t), \bar{\psi}_i \rangle_X \bar{\psi}_i \right\|_X^2 \, dt = \sum_{i=\ell+1}^{d} \bar{\lambda}_i. \tag{A.2.2}
\]

**Proof.** For a proof, we refer to [GV17, Theorem 2.13].

**Remark A.2.3.** (i) Even if the dimension \( d \) of the snapshot subspace \( \mathcal{V} \) is finite, Theorem A.2.2 allows us to theoretically compute the POD subspace \( V^\ell \) for any \( \ell \in \mathbb{N} \).

(ii) In practice, there are different ways of computing a POD basis by using a discretized version of Theorem A.2.2: In the simplest case, given the matrix \( \mathcal{Y} \) containing the discrete snapshots, one can either solve the eigenvalue problems \( \mathcal{Y}^T \mathcal{Y} \) or \( \mathcal{Y} \mathcal{Y}^T \), or compute a singular value decomposition (SVD) of the matrix \( \mathcal{Y} \). For details, we refer to [GV17, Section 2].

(iii) In applications, the identity (A.2.2) can be used to estimate how many basis functions are needed to obtain a good approximation of the snapshot subspace: Given a tolerance \( 0 < \tau \ll 1 \), choose the smallest \( \ell \) such that

\[
\frac{\sum_{i=\ell+1}^{d} \bar{\lambda}_i}{\sum_{i=1}^{d} \bar{\lambda}_i} \leq \tau
\]

holds. This formula can be interpreted as demanding that more than \( 1 - \tau \) of the relative 'energy' of the snapshots is contained in the POD subspace.  

\[\diamondsuit\]
A.2.2 Application of POD to Linear Evolution Equations

We recall the situation from Section 2.1.1, where we introduced the linear parabolic advection-diffusion equation (2.1.1). In its abstract weak formulation, the equation reads

\[ \langle y_t(t), \varphi \rangle_{V', V} + a(t; y(t), \varphi) = \langle F(t) + B(u(t)), \varphi \rangle_{V', V} \quad \text{f.a.} \; \varphi \in V \text{ a.e. in } (0, T), \]
\[ y(0) = y_0 \quad \text{in } H, \tag{A.2.3} \]

with the spaces \( V := H^1(\Omega) \) and \( H := L^2(\Omega) \), cf. Section 2.1.1. In the following, we show how the POD method can be applied to this framework.

In this situation, canonical choices for the Hilbert space \( \mathcal{X} \) are either \( \mathcal{X} = V \) or \( \mathcal{X} = H \). Reasonable choices for the snapshots \( y^1, \ldots, y^k \in L^2(0, T; \mathcal{X}) \) are, e.g., solutions to the state equation (A.2.3) for different input parameters \( u^1, \ldots, u^k \). Now assume that a POD basis \( \{\psi_i\}_{i=1}^{\ell} \) of rank \( \ell \) with corresponding POD subspace \( V^\ell \) has been computed. Then the reduced-order version of (A.2.3) is obtained by a Galerkin projection of the equation onto \( V^\ell \). Thus, for the reduced-order equation, we want to find \( y^\ell \in W(0, T) \) with \( y^\ell(t) \in V^\ell \) for almost all \( t \in (0, T) \) such that

\[ \langle y^\ell_t(t), \psi \rangle_{V', V} + a(t; y^\ell(t), \psi) = \langle F(t) + B(u(t)), \psi \rangle_{V', V} \quad \text{f.a.} \; \psi \in V^\ell \text{ a.e. in } (0, T), \]
\[ \langle y^\ell(0), \psi \rangle_{H} = \langle y_0, \psi \rangle_{H} \quad \text{f.a.} \; \psi \in V^\ell \tag{A.2.4} \]

holds. Similarly to Section 2.1.1, it is possible to show that this equation has a unique solution, which depends continuously on the data \( F, u \) and \( y_0 \), cf. Theorem 2.1.4.

**Theorem A.2.4.** For any \( u \in \mathcal{U} \) there is a unique solution \( y^\ell = y^\ell(u) \in W(0, T) \) of (A.2.4). Moreover, there is a constant \( C > 0 \), which is independent of \( u, F, y_0 \) and \( \ell \), such that the estimate

\[ \|y^\ell\|_{W(0,T)} \leq C \left( \|u\|_{\mathcal{U}} + \|F\|_{L^2(0,T;V')} + \|y_0\|_{H} \right) \tag{A.2.5} \]

is satisfied.

**Proof.** A proof of a similar version this statement can be found in [Eva10, pp. 354 – 356, Theorems 1 and 2].

In Definition 2.1.5, we saw that we can split the solution of (A.2.3) into a \( u \)-independent part \( \hat{y} \) and and a part \( Su \) depending linearly on \( u \). Theorem A.2.4 allows us to do the same for the reduced-order equation (A.2.4).

**Definition A.2.5.** We denote the unique solution of (A.2.4) for \( u = 0 \) by \( \hat{y} \in W(0, T) \). Moreover, we define the linear solution operator \( S^\ell : \mathcal{U} \to W(0, T) \) of the equation

\[ \langle y^\ell_t(t), \psi \rangle_{V', V} + a(t; y^\ell(t), \psi) = \langle B(u(t)), \psi \rangle_{V', V} \quad \text{f.a.} \; \psi \in V^\ell \text{ a.e. in } (0, T), \]
\[ y^\ell(0) = 0 \quad \text{in } H, \tag{A.2.6} \]

which maps any control \( u \in \mathcal{U} \) to the unique solution \( y^\ell = S^\ell(u) \) of (A.2.6).
**Definition A.2.6.** Similarly to Definition 2.1.7, we define the linear operators

\[ S^\ell_Q : \mathcal{U} \to L^2(0,T; H), \quad S^\ell_Q u := \Phi_Q S^\ell u, \]

\[ S^\ell_T : \mathcal{U} \to H, \quad S^\ell_T u := \Phi_T S^\ell u, \]

where the operators \( \Phi_Q \) and \( \Phi_T \) have been defined in Definition 2.1.1.

We can show the following a-priori convergence result.

**Theorem A.2.7.** Assume that we are given a POD subspace \( V^\ell \) for any \( \ell \in \mathbb{N} \). Let \( u \in \mathcal{U} \) be arbitrary. If \( Su \in H^1(0,T; V) \), then we have

\[
\lim_{\ell \to \infty} \| S^\ell_Q u - S_Q u \|_{L^2(0,T; H)} = 0,
\]

\[
\lim_{\ell \to \infty} \| S^\ell_T u - S_T u \|_H = 0.
\]

**Proof.** A proof of this statement can be found in [GV17, Theorem 3.11 3.)].

**Remark A.2.8.** If the POD subspaces \( V^\ell \) \((\ell \in \mathbb{N})\) are generated using the snapshots \( y^1 = Su \) (and potentially \( y^2 = (Su)_t \)), then it is actually possible to show a-priori convergence rates for the errors \( \| S^\ell_Q u - S_Q u \|_{L^2(0,T; H)} \) and \( \| S^\ell_T u - S_T u \|_H \). At this point, we refer to [GV17, pp. 29 – 32] for more details. ♦

### A.2.3 Application of POD to Linear-Quadratic Optimal Control Problems

Now we show how to apply POD to the linear-quadratic optimal control problem (2.1.6) from Section 2.1.2. To obtain the reduced-order optimal control problem, we replace the full-order solution operators \( S_Q \) and \( S_T \) in the cost function \( \hat{J} \) with the reduced-order solution operators \( S^\ell_Q \) and \( S^\ell_T \).

**Definition A.2.9.** Based on the essential cost function \( \hat{J} \) from Definition 2.1.9, let the essential reduced-order cost function \( \hat{J}^\ell \) \( : \mathcal{U} \to \mathbb{R} \) be given by

\[
\hat{J}^\ell(u) := \frac{\sigma_Q}{2} \left\| S^\ell_Q u - \tilde{y}_Q \right\|_{L^2(0,T; H)}^2 + \frac{\sigma_T}{2} \left\| S^\ell_T u - \tilde{y}_T \right\|_H^2 + \frac{\sigma_u}{2} \| u \|_{L^2(0,T; \mathbb{R}^m)}^2.
\]

for all \( u \in \mathcal{U} \). Then the reduced-order optimal control problem reads

\[
\min_{u \in \mathcal{U}} \hat{J}^\ell(u) \quad \text{s.t.} \quad u \in \mathcal{U}_{ad}. \tag{A.2.7}
\]

Based on the convergence result from Theorem A.2.7, we can show the following result for the cost function.

**Theorem A.2.10.** Assume that we are given a POD subspace \( V^\ell \) for any \( \ell \in \mathbb{N} \). Let \( u \in \mathcal{U} \) be arbitrary. If \( Su \in H^1(0,T; V) \), then we have

\[
\lim_{\ell \to \infty} \left| \hat{J}^\ell(u) - \hat{J}(u) \right| = 0.
\]
Proof. The proof of this theorem follows with similar arguments as in the proof of [Ban17, Theorem 5.31].

Analogously to the full-order cost function, the reduced-order adjoint equation of (A.2.3) can be used to evaluate the gradient of the reduced-order cost function $\hat{J}^\ell$.

**Definition A.2.11.** Let $V^\ell$ be a POD subspace and $u \in U$ be arbitrary. Then the equation

$$-\langle p^\ell(t), \psi \rangle_{V',V} + a(t; \psi, p^\ell(t)) = \sigma_Q \langle (S^\ell_Q u)(t) - \tilde{y}_Q(t), \psi \rangle_H \quad \text{f.a. } \psi \in V^\ell \text{ a.e. in } (0,T),$$

$$\langle p(T), \psi \rangle_H = \sigma_T \langle S^\ell_T u - \tilde{y}_T, \psi \rangle_H \quad \text{f.a. } \psi \in V^\ell$$

is called the adjoint equation of (A.2.3).

**Theorem A.2.12.** Let $V^\ell$ be a POD subspace and $u \in U$ be arbitrary. Then (A.2.8) has a unique solution $p^\ell = p^\ell(u) \in W(0,T)$ with $p^\ell(t) \in V^\ell$ for almost all $t \in (0,T)$. Moreover, there is a constant $C > 0$, which is independent of $\ell$, such that the estimate

$$\|p^\ell\|_{W(0,T)} \leq C \left( \|\sigma_Q (S^\ell_Q u - \tilde{y}_Q)\|_{L^2(0,T;H)} + \|\sigma_T (S^\ell_T u - \tilde{y}_T)\|_H \right)$$

(A.2.9)

holds.


**Definition A.2.13.** Let $V^\ell$ be a POD subspace. Then we denote by $\hat{p}^\ell \in W(0,T)$ the unique solution of (A.2.8) for $u = 0$. Moreover, we define the linear solution operator $A^\ell: U \rightarrow W(0,T)$, which maps any control $u \in U$ to the unique solution $p^\ell = A^\ell u$ of

$$-\langle p^\ell(t), \psi \rangle_{V',V} + a(t; \psi, p^\ell(t)) = \sigma_Q \langle (S^\ell_Q u)(t), \psi \rangle_H \quad \text{f.a. } \psi \in V^\ell \text{ a.e. in } (0,T),$$

$$\langle p^\ell(T), \psi \rangle_H = \sigma_T \langle S^\ell_T u, \psi \rangle_H \quad \text{f.a. } \psi \in V^\ell.$$  

(A.2.10)

By using the convergence result from Theorem A.2.7 for the state equation, we can show the following convergence result for the adjoint equation.

**Theorem A.2.14.** Let the snapshot subspace $V$ be generated by arbitrary snapshots and assume that we are given a POD subspace $V^\ell$ for any $\ell \in \mathbb{N}$. Let $u \in U$ be arbitrary. If $Au, \hat{p}, Su \in H^1(0,T;V)$, then we have

$$\lim_{\ell \to \infty} \left\| A^\ell u + \hat{p}^\ell - Au - \hat{p} \right\|_{L^2(0,T;H)} = 0.$$  

Proof. At this point we refer to [GV17, Theorem 4.15] for a proof.

With the help of the adjoint equation, we obtain a representation of the gradient and the Hessian of $\hat{J}^\ell$. 

Lemma A.2.15. Let $V^\ell$ be a POD subspace and $u \in \mathcal{U}$ be arbitrary. Then it holds
\[
\nabla \hat{J}^\ell(u) = B^* (A^\ell u + \hat{p}^\ell) + \sigma_{\mathcal{U}} u \quad \in \mathcal{U},
\]
\[
\nabla^2 \hat{J}^\ell(u) = B^* A^\ell + \sigma_{\mathcal{U}} \text{id}_{\mathcal{U}} \quad \in L(\mathcal{U}, \mathcal{U}').
\]

Proof. The proof uses the exact same arguments as the proof of Corollary 2.1.19. \qed

As an immediate consequence of Theorem A.2.14 and Lemma A.2.15, we can conclude the convergence of the gradient of the reduced-order cost function to the gradient of the full-order cost function.

Corollary A.2.16. Let the assumptions of Theorem A.2.14 be satisfied. Then it holds
\[
\lim_{\ell \to \infty} \left\| \nabla \hat{J}^\ell(u) - \nabla \hat{J}(u) \right\|_{\mathcal{U}} = 0.
\]
Appendix B

An Augmented Lagrangian Penalty Method

In this chapter, we present an augmented Lagrangian penalty method, which is taken from [NW06, Chapter 17.4], for solving the Pascoletti-Serafini problem \((\text{PSP}(z, r))\). Recall that the problem formulation reads

\[
\begin{align*}
\min_{(u, t)} & \quad t \\
\text{s.t.} & \quad u \in \mathcal{U}_{\text{ad}}, \; t \in \mathbb{R}, \\
& \quad \hat{J}_i(u) - z_i \leq t r_i, \quad i = 1, \ldots, k.
\end{align*}
\]

(PSP\((z, r))\)

Note that we do not use the reformulated Pascoletti-Serafini problem

\[
\min_{u \in \mathcal{U}_{\text{ad}}} \hat{J}_{g_{z, r}}(u) := \min_{u \in \mathcal{U}_{\text{ad}}} \max_{i \in \{1, \ldots, k\}} \frac{1}{r_i} (J_i(u) - z_i)
\]

to avoid the non-differentiability of the max-function. By introducing the slack variables \(s_1, \ldots, s_k\), the problem \((\text{PSP}(z, r))\) can be reformulated to

\[
\begin{align*}
\min_{(u, t, s)} & \quad t \\
\text{s.t.} & \quad u \in \mathcal{U}_{\text{ad}}, \; t \in \mathbb{R}, \; s \in \mathbb{R}^k \\
& \quad c_i(u, t, s) := \hat{J}_i(u) - z_i - t r_i + s_i = 0, \quad i = 1, \ldots, k.
\end{align*}
\]

We continue by defining the variable space \(\mathcal{X} := \mathcal{U} \times \mathbb{R} \times \mathbb{R}^k\), where every \(x \in \mathcal{X}\) can be written as \(x = (u, t, s)\), and the admissible set \(\mathcal{X}_{\text{ad}} := \mathcal{U}_{\text{ad}} \times \mathbb{R} \times \mathbb{R}^k\). For this problem, we can formulate the Lagrangian function as follows

\[
\mathcal{L}((u, t, s), \lambda) := t + \sum_{i=1}^{k} \lambda_i c_i(u, t, s),
\]

where \(\lambda \in \mathbb{R}^k\) is the Lagrange multiplier. Following the lines of [NW06, Chapter 17.4], the augmented Lagrangian of this problem reads

\[
\mathcal{L}_A((u, t, s), \lambda; \mu) := t + \sum_{i=1}^{k} \lambda_i c_i(u, t, s) + \frac{\mu}{2} \sum_{i=1}^{k} c_i(u, t, s)^2,
\]

(B.1.2)
where $\mu > 0$ is a penalty parameter to ensure that the equality constraints are satisfied up to a desired precision. The idea is to repeatedly solve the subproblems
\[
\min_{(u,t,s)\in \mathcal{X}} \mathcal{L}_A((u,t,s), \lambda; \mu) \quad \text{s.t.} \quad (u,t,s) \in \mathcal{X}_{ad}
\] (B.1.3)
approximately and then update the Lagrange multiplier $\lambda$ and the penalty parameter $\mu$ until the termination criteria
\[
\|c(x)\|_{\mathbb{R}^k} < \tau_{EC},
\]
(B.1.4)
\[
\|x - P_{\mathcal{X}_{ad}}(x - \nabla_x \mathcal{L}_A(x, \lambda; \mu))\|_\mathcal{X} < \tau_{FOC}
\]
(B.1.5)
are satisfied for some tolerances $\tau_{EC}, \tau_{FOC} \in (0,1)$, where $P_{\mathcal{X}_{ad}} : \mathcal{X} \to \mathcal{X}_{ad}$ is the canonical projection onto $\mathcal{X}_{ad}$. The detailed procedure is summarized in Algorithm 11, which is directly taken from [NW06, Algorithm 17.4].

**Algorithm 11 Augmented Lagrangian penalty algorithm**

**Require:** Initial value $x^{(0)} = (u^{(0)}, t^{(0)}, s^{(0)})$, initial Lagrange multiplier $\lambda^{(0)}$, initial penalty parameter $\mu^{(0)}$, increasing factor for penalty parameter $\mu_{incr}$, initial tolerance for equality constraint $\tau_{EC,sub}^{(0)}$, initial termination tolerance $\tau_{FOC,sub}^{(0)}$ for the subproblem (B.1.3), termination tolerances $\tau_{EC}, \tau_{FOC}$

1: Set $i = 0$ and Loop_flag=True;
2: while Loop_flag do
3: Compute a solution $x^{(i+1)}$ of the subproblem (B.1.3) with termination criteria
4: if $\|c(x^{(i+1)})\|_{\mathbb{R}^k} < \tau_{EC,sub}^{(i)}$ then
5: if (B.1.4) and (B.1.5) hold for $x = x^{(i+1)}$ then
6: Set Loop_flag=False and go to line 15;
7: end if
8: Set $\lambda^{(i+1)} = \lambda^{(i)} + \mu^{(i)}c(x^{(i+1)})$ and $\mu^{(i+1)} = \mu^{(i)}$;
9: Set $\tau_{EC,sub}^{(i+1)} = \tau_{EC,sub}^{(i)}/\mu^{(i+1)}$ and $\tau_{FOC,sub}^{(i+1)} = \tau_{FOC,sub}^{(i)}/\mu^{(i+1)}$;
10: else
11: $\lambda^{(i+1)} = \lambda^{(i)}$ and $\mu^{(i+1)} = \mu_{incr} \mu^{(i)}$;
12: Set $\tau_{EC,sub}^{(i+1)} = 1/(\mu^{(i+1)})^{0.1}$ and $\tau_{FOC,sub}^{(i+1)} = 1/\mu^{(i+1)}$;
13: end if
14: Set $i = i + 1$;
15: end while

The idea of the algorithm is to enlarge the penalty parameter $\mu^{(i)}$ after the $i$-th iteration if the current tolerance for the violation of the equality constraint is exceeded, see lines 11 – 12. Otherwise, only the Lagrange multiplier is updated as shown in the lines 8 – 9.

As suggested in [NW06, Section 17.4], we use a Trust-Region method for solving the subproblem (B.1.3) in line 3. In our algorithm, we use the TR subproblem
\[
\min_{d \in \mathcal{X}} m^{(i)}(d) := \frac{1}{2} \langle \nabla_x^2 \mathcal{L}_A(x^{(i)}, \lambda^{(i)}; \mu^{(i)}) d, d \rangle_{\mathcal{X}} + \langle \nabla_x \mathcal{L}_A(x^{(i)}, \lambda^{(i)}; \mu^{(i)}), d \rangle_{\mathcal{X}},
\]
s.t. $x^{(i)} + d \in \mathcal{X}_{ad}$, $\|d\|_\infty \leq \Delta^{(i)}$, 

The detailed procedure is summarized in Algorithm 11, which is directly taken from [NW06, Algorithm 17.4].
which slightly deviates from the one proposed in [NW06], since it turned out to be a bit faster in our numerical experiments. To solve the TR subproblem, first a projected gradient method is used to compute an (approximate) Cauchy point, whose active indices are kept fixed. Note that we use the procedure of computing an exact Cauchy point described in [NW06, Section 16.7] for problems with a low-dimensional control space, e.g., for parameter optimization problems of an elliptic PDE. For problems with a high-dimensional control space, which occur, e.g., in optimal control problems of parabolic PDEs, this procedure might lead to a large computational effort, since new state and adjoint equations need to be solved whenever a new index of the control becomes active. In this case, the strategy presented in [CGT00, Algorithm 12.2.2] is more advisable, since it uses an inaccurate yet less costly bisection strategy for computing an approximate Cauchy point. Afterwards, a CG-Steihaug method (cf. [NW06, Algorithm 7.2]) is conducted on the inactive indices of the (approximate) Cauchy point, which results in the TR step $d^{(i)}_{TR}$. Afterwards, the quotient

$$\rho^{(i)} := \frac{L_A(x^{(i)} + d^{(i)}_{TR}, \lambda^{(i)}; \mu^{(i)}) - L_A(x^{(i)}, \lambda^{(i)}; \mu^{(i)})}{m^{(i)}(d^{(i)}_{TR}) - m^{(i)}(0)}$$

is computed, which indicates the relation between the actual reduction in the cost function and the predicted reduction in the model function. It is used as an indicator, whether the TR step is accepted or not and whether the TR radius is enlarged, remains the same, or is even reduced. To be precise, for given tolerances $0 < \eta_{acc} \leq \bar{\eta} < \bar{\eta} < 1$, we

- accept the TR step and set $x^{(i+1)} = x^{(i)} + d^{(i)}$ if $\rho^{(i)} \geq \eta_{acc}$,
- reduce the TR radius by a factor $\beta \in (0, 1)$, i.e., $\Delta^{(i+1)} = \beta \Delta^{(i)}$ if $\rho^{(i)} \leq \bar{\eta}$,
- set $\Delta^{(i+1)} = \Delta^{(i)}$ if $\bar{\eta} < \rho^{(i)} < \bar{\eta}$,
- increase the TR radius to $\Delta^{(i+1)} = \beta^{-1} \Delta^{(i)}$ if $\rho^{(i)} \geq \bar{\eta}$.

For more details, we refer to Section 17.4 and its references to Sections 7.1, 16.7 and 18.6 in [NW06].
B. An Augmented Lagrangian Penalty Method
Bibliography


BIBLIOGRAPHY


