ROM-based multiobjective optimization of elliptic PDEs via numerical continuation

Stefan Banholzer\textsuperscript{1}, Bennet Gebken\textsuperscript{2}, Michael Dellnitz\textsuperscript{2}, Sebastian Peitz\textsuperscript{2}, and Stefan Volkwein\textsuperscript{1}

\textsuperscript{1}Department of Mathematics and Statistics, University of Konstanz, Germany
\textsuperscript{2}Department of Mathematics, Paderborn University, Germany

June 24, 2019

Abstract

Multiobjective optimization plays an increasingly important role in modern applications, where several objectives are often of equal importance. The task in multiobjective optimization and multiobjective optimal control is therefore to compute the set of optimal compromises (the Pareto set) between the conflicting objectives. Since the Pareto set generally consists of an infinite number of solutions, the computational effort can quickly become challenging when the objectives are costly to evaluate as is the case for models governed by partial differential equations (PDEs). To decrease the numerical effort to an affordable amount, surrogate models can be used to replace the expensive PDE evaluations. Existing multiobjective optimization methods using model reduction are limited either to low parameter dimensions or to few (ideally two) objectives. In this article, we present a combination of the reduced basis model reduction method with a continuation approach using inexact gradients. The resulting approach can handle an arbitrary number of objectives while yielding a significant reduction in computing time.

1. Introduction

The dilemma of deciding between multiple, equally important goals is present in almost all areas of engineering and economy. A prominent example comes from production, where we want to produce a product at minimal cost while simultaneously preserving a high quality. In the same manner, multiple goals are present in most technical applications, maximizing the velocity while minimizing the energy consumption of electric vehicles\cite{24} being only one of many examples. These conflicting goals result in multiobjective optimization problems (MOPs)\cite{9}, where we want to optimize all objectives simultaneously. Since the objectives are in general contradictory, there exists an infinite number of optimal compromises. The set of these compromise solutions is called the Pareto set, and the goal in multiobjective optimization is to approximate this set in an efficient manner, which is significantly more expensive than solving a single objective problem. Due to this, the development of efficient numerical approximation methods is an active area of research, and methods range from scalarization\cite{9,14} over set-oriented approaches\cite{8} and continuation\cite{14} to evolutionary algorithms\cite{7}. Recent advances have paved the way to new challenging application areas for multiobjective optimization such as feedback control or problems constrained by partial differential equations (PDEs); cf.\cite{22} for a survey.

In the presence of PDE constraints, the computational effort can quickly become infeasible such that special means have to be taken in order to accelerate the computation. To this
end, surrogate models form a promising approach for significantly reducing the computational effort. A widely used approach is to directly construct a mapping from the parameter to the objective space using as few function evaluations of the expensive model as possible, cf. [30, 6] for extensive reviews. In the case of PDE constraints, an alternative approach is via dimension reduction techniques such as Proper Orthogonal Decomposition (POD) [29, 18] or the reduced basis (RB) method [11]. In these methods, a small number of high-fidelity solutions is used to construct a low-dimensional surrogate model for the PDE which can be evaluated significantly faster while guaranteeing convergence using error estimates. In recent years, several methods have been proposed where model reduction is used in multiobjective optimization and optimal control. In [17] and [16], scalarization using the so-called weighted sum method was combined with RB and POD, respectively. In [1, 2], convex problems were solved using reference point scalarization and POD, and set-oriented approaches were used in [4, 3]. A comparison of both was performed in [23] for the Navier–Stokes equations.

In this article we combine an extension of the continuation methods presented in [14, 28] to inexact gradients (Section 2) with a reduced basis approach for elliptic PDEs (Section 3). To deal with the error introduced by the RB approach, we combine the KKT conditions for MOPs with error estimates for the RB method to obtain a tight superset of the Pareto set. For the example considered here, the proposed method yields a speed-up factor of approximately 63 compared to the direct solution of the expensive problem (Section 4). Additionally, our approach allows us to control the quality of the result by controlling the errors for each objective function individually.

2. A continuation method for MOPs with inexact objective gradients

In this section, we will begin by briefly introducing the basic concepts of multiobjective optimization upon which we will build in this article (see [9, 14] for detailed introductions). Afterwards, we will discuss the continuation method for MOPs and present two modifications of it that can deal with inexact gradient information.

2.1. Multiobjective optimization

The goal of multiobjective optimization is to minimize several conflicting criteria at the same time. In other words, we want to minimize an objective \( J = (J_1, ..., J_k) : \mathbb{R}^n \to \mathbb{R}^k \) that is vector valued. It maps the variable space \( \mathbb{R}^n \) to the image space \( \mathbb{R}^k \). In contrast to single-objective optimization (i.e., \( k = 1 \)), there exists no natural total order of the image space \( \mathbb{R}^k \) for \( k > 1 \). As a result, the classical concept of optimality has to be generalized:

**Definition 2.1.** (a) \( \bar{u} \in \mathbb{R}^n \) is called (globally) Pareto optimal if there is no other point \( u \in \mathbb{R}^n \) such that \( J_i(u) \leq J_i(\bar{u}) \) for all \( i \in \{1, ..., k\} \) and \( J_j(u) < J_j(\bar{u}) \) for some \( j \in \{1, ..., k\} \).

(b) The set \( P \) of all Pareto optimal points is called the Pareto set. Its image under \( J \) is the Pareto front.

The Pareto set is the solution of the multiobjective optimization problem (MOP)

\[
\min_{u \in \mathbb{R}^n} J(u).
\]  

Constrained MOPs can be formulated analogously by restricting \( u \) in Definition 2.1 to a subset \( U \subseteq \mathbb{R}^n \). Similar to the scalar-valued case, if \( J \) is differentiable, we can use the derivative of \( J \) to obtain necessary conditions for Pareto optimality, the Karush-Kuhn-Tucker (KKT) conditions [14]:

---

2
Theorem 2.2. Let \( \bar{u} \) be a Pareto optimal point of (MOP). Then there exist multipliers
\[
\alpha \in \Delta_k := \left\{ \alpha \in (\mathbb{R}^{\geq 0})^k : \sum_{i=1}^k \alpha_i = 1 \right\}
\]
such that
\[
DJ(\bar{u})^\top \alpha = \sum_{i=1}^k \alpha_i \nabla J_i(\bar{u}) = 0.
\]

(KKT)

For \( k = 1 \), this reduces to the well-known optimality condition \( \nabla J(\bar{u}) = 0 \). If \( J \) is non-convex, then the points satisfying (KKT) form a proper superset of the Pareto set \( P \):

Definition 2.3. If \( \bar{u} \in \mathbb{R}^n \) and \( \bar{\alpha} \in \Delta_k \) satisfy (KKT), then \( \bar{u} \) is called Pareto critical with corresponding KKT vector \( \bar{\alpha} \), containing the KKT multipliers \( \bar{\alpha}_i \), \( i \in \{1, \ldots, k\} \). The set \( P_c \) of all Pareto critical points is called the Pareto critical set.

When solving an MOP, an initial step can be to compute the Pareto critical set. This set possesses additional structure which can be exploited in numerical schemes. Introducing the function
\[
F: \mathbb{R}^n \times (\mathbb{R}^{\geq 0})^k \to \mathbb{R}^{n+1}, (u,\alpha) \mapsto \left( \sum_{i=1}^k \alpha_i \nabla J_i(u) \right),
\]
we see that Pareto critical points and their corresponding KKT vectors can be described as the zero level set of \( F \). As shown by Hillermeier [14], this has the following implication:

Theorem 2.4. Let \( J \) be twice continuously differentiable.

(a) Let \( \mathcal{M} := \{(u,\alpha) \in \mathbb{R}^n \times (\mathbb{R}^{\geq 0})^k : F(u,\alpha) = 0\} \). If the Jacobian of \( F \) has full rank everywhere, i.e.,
\[
\text{rk}(DF(u,\alpha)) = n + 1 \quad \forall (u,\alpha) \in \mathcal{M},
\]
then \( \mathcal{M} \) is a \( (k-1) \)-dimensional differentiable submanifold of \( \mathbb{R}^{n+k} \). The tangent space of \( \mathcal{M} \) at \( (u,\alpha) \) is given by
\[
T_{(u,\alpha)} \mathcal{M} = \ker(DF(u,\alpha)).
\]

(b) Let \( (u,\alpha) \in \mathcal{M} \) such that (1) holds in \( (u,\alpha) \). Then there is an open set \( U \subseteq \mathbb{R}^n \times \mathbb{R}^k \) with \( (u,\alpha) \in U \) such that \( \mathcal{M} \cap U \) is a manifold as in (a). In other words, \( \mathcal{M} \) locally possesses a manifold structure in all points satisfying (1).

Theorem 2.4 forms the basis for the continuation method we use in this article.

2.2. Continuation method with exact gradients

We only give a brief description of the method here and refer to [28] and [14] for details. By Theorem 2.4, the Pareto critical set is – except for the boundary – the projection of the differentiable manifold \( \mathcal{M} \subseteq \mathbb{R}^n \times \mathbb{R}^k \) onto its first \( n \) components. In [10] it has been shown that generically, this also holds for the first-order approximations, i.e., the projection of the tangent space of \( \mathcal{M} \) yields the tangent cone of \( P_c \). Given a Pareto critical point \( \bar{u} \in P_c \), this means that we can find first-order candidates for new Pareto critical points in the vicinity of \( \bar{u} \) by moving in the projected tangent space of \( \mathcal{M} \). The idea of the continuation method is to do this iteratively to explore the entire Pareto critical set.

Instead of approximating \( P_c \) by a set of points, we use a set-oriented numerical approach; cf. [28] for details. This has the key advantage that it is easy to check whether a certain part of the set has already been computed, which is difficult when working with points. Additionally, a
covering of \( P_c \) by boxes makes it easy to obtain (and exploit) its topological properties. In the approach, we evenly divide the variable space \( \mathbb{R}^n \) into hypercubes or boxes \( B \) with radius \( r > 0 \):

\[
B(r) := \{ [-r, r]^n + (2i_1r, ..., 2i_nr)^\top : (i_1, ..., i_n) \in \mathbb{Z}^n \}.
\] (2)

**Remark 2.5.** For ease of notation and readability, we will only consider the case where points \( u \in \mathbb{R}^n \) are contained in single boxes. In other words, we only consider the case where \( u \) is in the interior of a box and not in the intersection of multiple boxes. Since this is the generic case, this has no impact on the numerical methods we will propose later. ■

For \( u \in \mathbb{R}^n \) let \( B(u, r) \) be the box containing \( u \). We want to compute the subset of \( B(r) \) covering the Pareto critical set for a given radius \( r \), i.e.,

\[
B_c(r) := \{ B \in B(r) : P_c \cap B \neq \emptyset \}.
\]

Since we are interested in a covering via boxes instead of an approximation via points, when moving in a tangent direction of the critical set, we will search for **tangent boxes** instead of single points. For \( u \in \mathbb{R}^n \) let

\[
N(u, r) := \{ B \in B(r) : B(u, r) \cap B \neq \emptyset \}
\]

be the set of neighboring boxes of \( B(u, r) \). Starting from a box \( B(\bar{u}, r) \) containing a critical point \( \bar{u} \) with KKT vector \( \bar{\alpha} \), we want to explore the neighboring boxes covering the projected tangent space at \( \bar{u} \), i.e.,

\[
B'(\bar{u}, r) = \{ B' \in B(r) : B' \in N(\bar{u}, r), \ B' \cap \bar{u} + \text{pr}_u(T(\bar{u}, \bar{\alpha})M) \neq \emptyset \}.
\] (3)

Here, \( \text{pr}_u : \mathbb{R}^{n+k} \to \mathbb{R}^n \) is the projection of the tangent space onto the first \( n \) components, i.e., the variable space. The typical situation is visualized in Figure 1.

![Figure 1: Tangent boxes (black) of the initial box (grey) containing \( \bar{u} \), which is contained in the Pareto critical set \( P_c \) (dashed). The red line indicates the projection of the tangent space of \( M \) onto the variable space.](image)

As the tangent space of the Pareto critical set is only a linear approximation, a corrector step is required to verify that a given tangent box actually contains part of the Pareto critical set. This means that there has to be at least one \( u \in B \) satisfying (KKT). To this end, for a box \( B \), we consider the problem

\[
\min_{u \in B, \alpha \in \Delta_k} \| DJ(u)\top \alpha \|_2^2
\] (PC-Box)
Let $\theta(B)$ be the optimal value of this problem. Then obviously

$$\theta(B) = 0 \iff B \cap P_c \neq \emptyset.$$  

In particular, if $\theta(B) = 0$ and $(\bar{u}, \bar{\alpha})$ is the solution of (PC-Box), then $\bar{u}$ is Pareto critical with corresponding KKT vector $\bar{\alpha}$. After solving (PC-Box) in each tangent box, all boxes with $\theta(B) = 0$ are added to a queue and a new iteration of the method is started with the first element in the queue. The method stops when the queue is empty, i.e., when there is no neighboring box of the current set of boxes that contains part of the Pareto critical set. For the remainder of this article, we will refer to this method as the exact continuation method.

### 2.3. Continuation method with inexact gradients

Using ROM to solve the state equation of an MOP of an elliptic PDE will introduce an error in the objective functions and the corresponding gradients, which has to be taken into account in order to ensure Pareto criticality of the solution. We here present a method that calculates a tight superset of the Pareto critical set via numerical continuation, using upper bounds for the errors in the approximated gradients. Formally, we now assume that for each gradient $\nabla J_i$, we only have an approximation $\nabla J_{ri}$ such that

$$\sup_{u \in \mathbb{R}^n} \left\| \nabla J_i(u) - \nabla J_{ri}(u) \right\|_2 \leq \epsilon_i, \ i \in \{1, \ldots, k\},$$  

with upper bounds $\epsilon = (\epsilon_1, \ldots, \epsilon_k) \in \mathbb{R}^k$. Let $P_c$ and $P_{rc}$ be the Pareto critical sets corresponding to $(\nabla J_i)_i$ and $(\nabla J_{ri})_i$, respectively. The following lemma shows how these error bounds translate to error bounds for the KKT conditions:

**Lemma 2.6.** Let $\bar{u} \in \mathbb{R}^n$ be Pareto critical for $J$ with KKT vector $\bar{\alpha} \in \Delta_k$. Then

$$\left\| DJ_{ri}(\bar{u}) \bar{\alpha} \right\|_2 \leq \sum_{i=1}^k \bar{\alpha}_i \epsilon_i \leq \|\epsilon\|_\infty.$$  

**Proof.** From the estimate

$$\left\| DJ_{ri}(\bar{u}) \bar{\alpha} \right\|_2 = \left\| DJ_{ri}(\bar{u}) \bar{\alpha} - DJ(\bar{u}) \bar{\alpha} \right\|_2 = \sum_{i=1}^k (\nabla J_{ri}(\bar{u}) - \nabla J_i(\bar{u}))^\top \bar{\alpha}_i \right\|_2 \leq \sum_{i=1}^k \| \nabla J_{ri}(\bar{u}) - \nabla J_i(\bar{u}) \|_2 \bar{\alpha}_i \leq \sum_{i=1}^k \bar{\alpha}_i \epsilon_i \leq \|\epsilon\|_\infty$$

we derive the claim.

**Remark 2.7.** Lemma 2.6 can be generalized to equality and inequality constrained MOPs using the constrained version of the optimality conditions from [13]. In this case, in the norm on the left-hand side of the inequality in Lemma 2.6, one additionally has to add a linear combination of the gradients of the equality and inequality constraints.

Lemma 2.6 shows that we have to weaken the conditions for Pareto criticality of the reduced objective function to obtain a superset of the actual Pareto critical set $P_c$. Formally, let

$$P_{1r} := \left\{ u \in \mathbb{R}^n : \min_{\alpha \in \Delta_k} \left\| DJ_{ri}(u) \bar{\alpha} \right\|_2^2 \leq \epsilon_i^2 \right\},$$

$$P_{2r} := \left\{ u \in \mathbb{R}^n : \min_{\alpha \in \Delta_k} \left( \left\| DJ_{ri}(u) \bar{\alpha} \right\|_2^2 - (\bar{\alpha}^\top \epsilon)^2 \right) \leq 0 \right\}.$$
Example 2.9. Let for a simple MOP (cf. [21]).

The condition for a point being in $P_1^r$ only depends on the maximal error $\|\epsilon\|_\infty$ and can be seen as a relaxed version of the KKT conditions for the inexact objective function. In contrast to this, the condition in $P_2^r$ actually considers the individual error bounds. By Lemma 2.6, $P_c \subseteq P_2^r \subseteq P_1^r$ and $P_c^r \subseteq P_2^r \subseteq P_1^r$.

i.e., both $P_1^r$ and $P_2^r$ are supersets of $P_c$ and $P_c^r$ (the points $\bar{u}$ for which the inexact gradients satisfy (KKT)). In fact, $P_2^r$ is a tight superset of $P_c$ in the following sense:

**Lemma 2.8.** Let $\bar{u} \in P_2^r$. Then there is some continuously differentiable $\bar{J} : \mathbb{R}^n \rightarrow \mathbb{R}^k$ with

$$\sup_{\alpha \in \mathbb{R}^n} \|\nabla \bar{J}_i(u) - \nabla J_i^r(u)\|_2 \leq \epsilon_i \quad \forall i \in \{1,\ldots,k\}$$

such that $\bar{u}$ is Pareto critical for $\bar{J}$.

**Proof.** Let

$$\bar{\alpha} \in \arg\min_{\alpha \in \Delta_k} \left(\| DJ^r(u)^T \alpha \|_2^2 - (\alpha^T \epsilon)^2 \right),$$

$$\nu := DJ^r(\bar{u})^T \bar{\alpha},$$

$$g(u) := -\left(\frac{1}{\bar{\alpha}^T \epsilon} \sum_{i=1}^n u_i \epsilon_i \right) \epsilon,$$

$$\bar{J}(u) := J^r(u) + g(u).$$

Since $\bar{u} \in P_2^r$ by assumption, we have $\|\nu\|_2 \leq \bar{\alpha}^T \epsilon$. Thus

$$\|\nabla \bar{J}_i(u) - \nabla J_i^r(u)\|_2 = \|\nabla g_i(u)\|_2 = \frac{\epsilon_i}{\bar{\alpha}^T \epsilon} \|\nu\|_2 \leq \epsilon_i \forall u \in \mathbb{R}^n \text{ and } \forall i \in \{1,\ldots,k\},$$

and

$$DJ^r(\bar{u})^T \bar{\alpha} = \nu + \sum_{i=1}^k \bar{\alpha}_i \nabla g_i(\bar{u}) = \nu - \sum_{i=1}^k \bar{\alpha}_i \epsilon_i \epsilon = 0,$$

which proves the lemma.

Lemma 2.8 shows that for each point $\bar{u}$ in $P_2^r$, there is an objective function satisfying the error bounds (4) for which $\bar{u}$ is Pareto critical. As a result, $P_2^r$ is the tightest superset of $P_c$ we can hope for if we only have the estimates in (4). The following example shows both supersets for a simple MOP (cf. [21]).

**Example 2.9.** Let

$$J^r : \mathbb{R}^2 \rightarrow \mathbb{R}^2, \quad u \mapsto \begin{pmatrix} (u_1 - 1)^2 + (u_2 - 1)^4 \\ (u_1 + 1)^2 + (u_2 + 1)^2 \end{pmatrix}.$$ 

We consider the two error bounds $\epsilon_1 = (0.2, 0.05)^T$ and $\epsilon_2 = (0, 0.2)^T$. The corresponding supersets $P_1^r$ and $P_2^r$ are shown in Figure 2.

As $\|\epsilon_1\|_\infty = \|\epsilon_2\|_\infty = 0.2$, $P_1^r$ is identical for both error bounds. Considering each component of $J^r$ individually, the critical points of $J_1^r$ and $J_2^r$ are located at $u^1 = (1,1)^T$ and $u^2 = (-1, -1)^T$, respectively. For $P_2^r$, we see that the difference between $P_2^r$ and $P_2^c$ becomes smaller the closer we get to the critical point of the objective function with the smaller error bound. This can be expected, as the influence (or weight) of $\nabla J_1^r(u)$ in the KKT conditions (KKT) becomes larger the closer $u$ is to $u^1$. In particular, in Figure 2(b), the difference between $P_2^r$ and $P_2^c$ at $(1,1)^T$ becomes zero, as $\epsilon_1^2 = 0$. \hfill \Box
If we set $\epsilon_i = \|\epsilon\|_\infty$ for all $i \in \{1, \ldots, k\}$, then $P^r_1 = P^r_2$. Thus, we will from now on only consider $P^r_2$. As shown in the previous example, the “dimension” of $P^r_2$ is higher than the “dimension” of $P^r_c$. More precisely, $P^r_2$ contains the closure of an open subset of $\mathbb{R}^n$, which is shown in the following lemma:

**Lemma 2.10.** Let $\nabla J^r_i$ be continuous for all $i \in \{1, \ldots, k\}$. Let

$$A := \left\{ u \in \mathbb{R}^n : \min_{\alpha \in \Delta_k} \left( \|DJ^r(u)^\top \alpha\|_2^2 - (\alpha^\top \epsilon)^2 \right) < 0 \right\}.$$ 

Then

(a) $P^r_2$ is closed. In particular, $\overline{A} \subseteq P^r_2$.

(b) $A$ is open.

**Proof.** (a) The case $P^r_2 = \emptyset$ is trivial, so we assume that $P^r_2 \neq \emptyset$. Let $\bar{u} \in \overline{P^r_2}$. Then there is a sequence $(u^i)_i \in P^r_2$ with $\lim_{i \to \infty} u^i = \bar{u}$. Consider the sequence $(\alpha^i)_i \in \Delta_k$ with

$$\alpha^i \in \arg\min_{\alpha \in \Delta_k} \left( \|DJ^r(u^i)^\top \alpha\|_2^2 - (\alpha^\top \epsilon)^2 \right).$$

By compactness of $\Delta_k$, we can assume w.l.o.g. that there is some $\bar{\alpha} \in \Delta_k$ with $\lim_{i \to \infty} \alpha^i = \bar{\alpha}$. Let

$$\Psi : \mathbb{R}^n \times \Delta_k \to \mathbb{R}, \quad (u, \alpha) \mapsto \|DJ^r(u)^\top \alpha\|_2^2 - (\alpha^\top \epsilon)^2.$$ 

By our assumption, $\Psi$ is continuous. From $\Psi(u^i, \alpha^i) < 0$ for all $i \in \mathbb{N}$ it follows that $\Psi(\bar{u}, \bar{\alpha}) \leq 0$, which yields $\bar{u} \in P^r_2$.

(b) The case $A = \emptyset$ is again trivial such that we assume $A \neq \emptyset$. Let $\bar{u} \in A$ with

$$\bar{\alpha} \in \arg\min_{\alpha \in \Delta_k} \left( \|DJ^r(\bar{u})^\top \alpha\|_2^2 - (\alpha^\top \epsilon)^2 \right).$$

Let $\psi : \mathbb{R}^n \to \mathbb{R}$, $u \mapsto \|DJ^r(u)^\top \bar{\alpha}\|_2^2 - (\bar{\alpha}^\top \epsilon)^2$. Then $\psi(\bar{u}) < 0$ and by our assumption, $\psi$ is continuous. Therefore, there is some open set $U \subseteq \mathbb{R}^n$ with $\bar{u} \in U$ such that $\psi(u) < 0$ for all $u \in U$. Since

$$\min_{\alpha \in \Delta_k} \left( \|DJ^r(u)^\top \alpha\|_2^2 - (\alpha^\top \epsilon)^2 \right) \leq \psi(u) < 0 \quad \forall u \in U$$

we have $U \subseteq A$ such that $A$ is open.

---

**Figure 2:** $P^r_1$ and $P^r_2$ for different error bounds $\epsilon$
We will now present two strategies for the numerical computation of $P_r^2$. Analogously to the case with exact gradients, we will approximate $P_r^2$ via the box covering $B_c^r(r) := \{ B \in \mathcal{B}(r) : B \cap P_r^2 \neq \emptyset \}$.

2.3.1. Strategy 1

The idea of our first method is to mimic the exact continuation method to calculate $B_c^r$. For this, there are mainly two modifications we have to make:

1. By Lemma 2.10, $P_r^2$ is not a lower-dimensional object in $\mathbb{R}^n$, so it makes no sense to use tangent information to find first-order candidates as in $\text{(3)}$. Instead, we have to consider all neighboring boxes.

2. The problem $\text{(PC-Box)}$ has to be replaced by a problem that checks the defining inequality of $P_r^2$.

As a replacement for $\text{(PC-Box)}$, we consider the following problem:

$$\min_{u \in B, \alpha \in \Delta_k} \| DJ(u) \alpha \|^2_2 - (\alpha^\top \epsilon)^2. \quad (\epsilon\text{PC-Box})$$

Let $\theta_\epsilon(B)$ be the optimal value of this problem. Note that $\theta_\epsilon(B) < 0$ is sufficient to verify that a box $B$ contains part of $P_r^2$. As a result, we do not need to solve $\text{(\epsilonPC-Box)}$ exactly. For example, when using an iterative method for the solution of $\text{(\epsilonPC-Box)}$, we can stop when the function value is negative. The above mentioned changes yield Algorithm 1.

**Algorithm 1** Strategy 1: Box-Continuation Algorithm with Inexact Gradients

Given: Radius $r > 0$ of boxes.

1. Choose an initial point $u_0 \in P_r^2$ and initialize $\mathcal{B} = \{ B(u_0, r) \}$ and a queue $Q = \{ u_0 \}$.
2. while $Q \neq \emptyset$ do
3. Remove the first element $\bar{u}$ from $Q$.
4. for $B' \in N(\bar{u}, r) \setminus \mathcal{B}$ do
5. Solve $\text{\epsilon\text{PC-Box}}$ for $B'$. Let $\theta_\epsilon(B')$ be the optimal value and $(u', \alpha')$ be the solution.
6. if $\theta_\epsilon(B') \leq 0$ then
7. Add $u'$ to $Q$ and $B'$ to $\mathcal{B}$.
8. end if
9. end for
10. end while

Due to the loss of low-dimensionality of $P_r^2$, the formulation of the continuation method becomes much simpler. As a consequence, it is straightforward to show that Algorithm 1 yields the desired covering $B_c^r(r)$.

When executing the exact continuation method directly using inexact gradients (i.e., forgetting about the inexactness) and comparing it to Algorithm 1 (with the same box radius), the former will generally be much faster than the latter. A suitable way to evaluate the run time is to compare the number of times Problems $\text{(PC-Box)}$ and $\text{(\epsilonPC-Box)}$ need to be solved, respectively, as they require the majority of the computing time and are equally difficult to solve. (Here, we assume that both problems are solved with equal precision.) For each box added to the collection $\mathcal{B}$ in either algorithm, one of these problems has to be solved. Consequently, the longer run time of Algorithm 1 is partly due to the fact that $P_r^2$ is a superset of $P_c^r$, which means that more
boxes are required to cover $P_r^2$ than $P_r^2$. However, even if the error bounds $\epsilon$ are small such that $P_r^2$ and $P_r^2$ are almost equal, Algorithm 1 will be slower. This is due to the fact that instead of only the tangent boxes, all neighboring boxes have to be tested with (PC-Box) in each loop of Algorithm 1. While this does not matter in the interior of $P_r^2$ (as all neighboring boxes are in fact in $P_r^2$ in that case), it is very inefficient at the boundary of $P_r^2$. This is the motivation for the second strategy.

2.3.2. Strategy 2

By Lemma 2.10 $P_r^2$ has the same dimension as the space of variables $\mathbb{R}^n$. This means that it can be described much more efficiently by its topological boundary $\partial P_r^2$. To be more precise, $\mathbb{R}^n \setminus \partial P_r^2$ consists of different connected components that lie either completely inside or completely outside $P_r^2$. So if we know $\partial P_r^2$, we merely have to test one point of each connected component if it is contained in $P_r^2$ or not to completely determine $\partial P_r^2$. Therefore, the idea of our second strategy is to only compute $\partial P_r^2$.

Let

$$\varphi : \mathbb{R}^n \to \mathbb{R}, \quad u \mapsto \min_{\alpha \in \Delta_k} \left( \|DJ^r(u)\alpha\|_2^2 - (\alpha^\top \epsilon)^2 \right).$$

This map is well-defined since $\Delta_k$ is compact, i.e., the minimum always exists. By Lemma 2.10 we have $\partial P_r^2 \subseteq \varphi^{-1}(0)$. Our goal is to compute $\varphi^{-1}(0)$ via a continuation approach. To this end, we first have to show that $\varphi$ is differentiable. We will do this by investigating the properties of the optimization problem in (5), i.e., of the problem

$$\min_{\alpha \in \mathbb{R}^k} \omega(\alpha),$$

s.t. $\sum_{i=1}^k \alpha_i = 1$,

$$\alpha_i \geq 0 \quad \forall i \in \{1, \ldots, k\},$$

for

$$\omega(\alpha) := \|DJ^r(u)\alpha\|_2^2 - (\alpha^\top \epsilon)^2 = \alpha^\top (DJ^r(u)DJ^r(u)^\top - \epsilon^\top)\alpha.$$

This leads to the following result.

**Theorem 2.11.** Let $\bar{u} \in \varphi^{-1}(0)$ such that (6) has a unique solution $\bar{\alpha} \in \Delta_k$ with $\bar{\alpha}_i > 0$ for all $i \in \{1, \ldots, k\}$. Let (6) be uniquely solvable in a neighborhood of $\bar{u}$. Then there is an open set $U \subseteq \mathbb{R}^n$ with $\bar{u} \in U$ such that $\varphi|_U$ is continuously differentiable.

**Proof.** See Appendix A \square

For a standard continuation approach, we also have to show that $\varphi^{-1}(0)$ is a manifold. By the Level Set Theorem (cf. [19], Corollary 5.14), to properly show that $\varphi^{-1}(0)$ is a manifold in a neighborhood of some $\bar{u} \in \varphi^{-1}(0)$, we would have to show that $D\varphi|_U(\bar{u}) \neq 0$ (cf. [21]). From the theoretical point of view, this poses a problem as there is no obvious way to achieve this. In practice however, we can test this by checking if the norm of $D\varphi|_U(\bar{u})$ is below a certain threshold. If this is the case, and if $\varphi^{-1}(0)$ is indeed not a manifold, we again have to consider all neighboring boxes as tangent boxes as in strategy 1. Otherwise, if $D\varphi|_U(\bar{u}) \neq 0$, we can compute the tangent space $T_{\bar{u}}$ of $\varphi^{-1}(0)$ at $\bar{u}$ via

$$T_{\bar{u}} = \ker(D\varphi(\bar{u})).$$

Finally, in analogy to (PC-Box) and (PC-Box), we will use the following problem to test if a box $B$ contains part of $\partial P_r^2$:

$$\min_{u \in B} \varphi(u)^2.$$  \hspace{1cm} (\partial \epsilon \text{PC-Box})

The resulting continuation method is presented in Algorithm 2.
Remark 2.12. 1. Since for every evaluation of $\phi$ the solution of the quadratic problem \((6)\) has to be computed, \((\partial_{\epsilon} \text{PC-Box})\) is significantly more difficult to solve than \((\epsilon \text{PC-Box})\). Additionally, we are looking for the points $u$ where $\phi(u) = 0$, i.e., where the problem \((6)\) is not positive definite. This increased difficulty of Strategy 2 is compensated by the fact that far fewer boxes have to be checked with \((\partial_{\epsilon} \text{PC-Box})\) than with \((\epsilon \text{PC-Box})\) in Strategy 1.

2. When all $\epsilon_i = \bar{\epsilon}$ are equal, $\phi(u) = -\bar{\epsilon}^2$ for all $u \in P_r^c$, i.e., $\phi$ is constant on the Pareto critical set $P_r^c$. This means that local solvers may fail to find a minimum of $\phi$ when the box $B$ in \((\partial_{\epsilon} \text{PC-Box})\) has a nonempty intersection with $P^c$. An obvious but expensive way to circumvent this problem is to start the local solver multiple times with different initial points. Alternatively, one can use sufficient conditions for a box $B$ containing part of $\phi^{-1}(0)$ before actually solving \((\partial_{\epsilon} \text{PC-Box})\). For example, by the intermediate value theorem, if there are two points in $B$ where $\phi$ has different sign, we immediately know that $\phi(u) = 0$ for some $u \in B$. (But note that for this method, we still need to find a point in $\phi^{-1}(0) \cap B$ to be able to calculate the tangent space of $\phi^{-1}(0)$).

3. In practice, error bounds which are zero can cause problems for the stability of Strategy 2. For example, in Figure 2(b), the width of $P_{r_2}^c$ becomes arbitrarily small near $(1,1)^T$. As a result, Strategy 2 may jump between different parts of the boundary and thus miss certain parts. Additionally, since the boundary of $P_r^c$ typically intersects the Pareto critical set $P_c$ in this case, \((\partial_{\epsilon} \text{PC-Box})\) may be difficult to solve (as in 2.). Thus, in practice, one should use error bounds that are slightly larger than zero, even if the corresponding gradients are exact.

Algorithm 2 Strategy 2: Boundary-Continuation Algorithm for Inexact Gradients

Given: Radius $r > 0$ of boxes.

1. Choose an initial point $u_0 \in \partial P_r^c$ and initialize $B = \{B(u_0, r)\}$ and a queue $Q = \{u_0\}$.

2. while $Q \neq \emptyset$ do
3. Remove the first element $\bar{u}$ from $Q$.
4. If $\|D\phi(\bar{u})\|_2$ is small set $T = \mathbb{R}^n$. Otherwise, compute the tangent space $T = \text{ker}(D\phi(\bar{u}))$.

5. Predictor:
6. Find all neighboring boxes of $B(\bar{u}, r)$ that have a nonempty intersection with $\bar{u} + T$ and have not been considered before, i.e.,

\[
B'(\bar{u}, r) = \{B' \in B(r) : B' \cap B(\bar{u}, r) \neq \emptyset, B' \cap \bar{u} + T \neq \emptyset\} \setminus B.
\]

7. Corrector:
8. for $B' \in B'(\bar{u}, r)$ do
9. Solve \((\partial_{\epsilon} \text{PC-Box})\) for $B'$. Let $\theta(B')$ be the optimal value and $u'$ be the solution.
10. if $\theta(B') = 0$ then
11. Add $u'$ to $Q$ and $B'$ to $B$.
12. end if
13. end for
14. end while
2.4. Globalization approach

Note that all algorithms presented in this section so far approximate either $P_c$, $P_r^2$ or $\partial P_r^2$ by starting in an initial point $u_0$ and then locally exploring in all (tangent) directions. Thus, if the set we want to approximate is disconnected, we can only compute the connected component that contains $u_0$. In the following, we will describe how we can solve this problem, i.e., how our methods can be globalized.

As mentioned earlier, an advantage of using boxes in the continuation method instead of points is the fact that it is easy to detect whether a region has already been explored. In particular, this allows us to start the continuation in multiple initial points at the same time, by simply adding all of them to the queue $Q$ in step 1 of Algorithms 1 or 2 (and initializing the covering $B$ with the corresponding boxes). As a result, to globalize our methods, we merely have to find an initial set $U_0$ of points such that the intersection of $U_0$ with each connected component is nonempty.

For obtaining an initial set, we make use of the optimization problems that verify if a box contains part of the set we want to approximate, i.e., the problems (PC-Box), $(\epsilon$PC-Box) and $(\partial\epsilon$PC-Box). The idea is to consider a box covering as in (2) with large radius $R$ and then simply test each box for relevant points using these problems. Let $B_0$ be a compact superset of the set that we want to approximate (i.e., of $P_c$, $P_r^2$ or $\partial P_r^2$), e.g., a large outer box. For ease of notation, we assume that $B_0$ is a union of boxes in $B(R)$. For the case of the Pareto critical set $P_c$, i.e., the globalization of the exact continuation method, the resulting method is presented in Algorithm 3. The corresponding globalization methods for Algorithm 1 and 2 are obtained by replacing (PC-Box) in step 3 by $(\epsilon$PC-Box) and $(\partial\epsilon$PC-Box), respectively.

**Algorithm 3 Global Initialization**

Given: Outer box $B_0$, Radius $R > 0$ of boxes.

1. Initialize $U_0 = \emptyset$.
2. for $B \in B(R)$ with $B \cap B_0 \neq \emptyset$ do
3. Solve (PC-Box) for $B$. Let $\theta(B)$ be the optimal value and $\bar{u}$ be the solution.
4. if $\theta(B) = 0$ then
5. Add $\bar{u}$ to $U_0$.
6. end if
7. end for

The radius $R$ has to be chosen such that for each connected component, there is at least one box in our covering that only has an intersection with the desired component. In theory, $R$ can obviously become very small if two different connected components are very close to each other. In this case, Algorithm 3 becomes infeasible to use, as the number of boxes that have to be tested becomes too large. In practice however, the components are often sufficiently far apart such that a large radius is sufficient and only few boxes have to be considered.

For the globalization of the exact continuation method and Algorithm 1 we only have to take the non-connectivity of $P_c$ and $P_r^2$ into account. For Algorithm 2 an additional problem may arise since the boundary $\partial P_r^2$ does not necessarily need to be smooth. Non-smoothness of $\partial P_r^2$ is caused by points in which $\varphi$ is not differentiable. (By Theorem 2.11 these are points where the solution of $(\partial\epsilon$PC-Box) is not unique.) In these points, $\partial P_r^2$ does not posses a tangent space, and our method will be unable to continue. As a result, we have to ensure in the initialization of Algorithm 2 that we choose an initial point in $U_0$ on each smooth component of $\partial P_r^2$. Visually, these can be thought of as the faces of $P_r^2$.

We conclude this section with some remarks on the practical use of Algorithm 3.

**Remark 2.13.** 1. For MOPs with a high-dimensional variable space, Algorithm 3 quickly becomes infeasible due to the exponential growth of the number of boxes in $B(R)$. For these
cases, an initialization based on points instead of boxes should be used, for example by applying methods from global optimization to modified versions of \((\text{PC-Box})\), \((\epsilon\text{PC-Box})\) and \((\partial \epsilon\text{PC-Box})\), where \(u\) is not constrained to a box \(B\).

2. Instead of directly looping over all boxes in step 2 of Algorithm 3, in some cases it might be more beneficial to first execute a few steps of the subdivision algorithm (cf. \[8\]) to quickly discard boxes that are far away from the Pareto critical set.

3. Multiobjective optimization of an elliptic PDE using the RB method

In this section we will present a multiobjective (parameter) optimization problem of an elliptic advection-diffusion-reaction equation and show how the reduced basis method can be applied in view of the continuation method for inexact gradients from Section 2.3 (see Algorithms 1 and 2).

3.1. Multiobjective optimization of an elliptic PDE

Given a domain \(\Omega \subset \mathbb{R}^d\), \(d \in \{2,3\}\), we consider the problem

\[
\begin{align*}
\min_{y,u} \quad & J(y,u) := \\
\text{s.t.} \quad & -\sum_{i=1}^{m'} \kappa_i \chi_{\Omega_i}(x) \Delta y(x) + c b(x) \cdot \nabla y(x) + r y(x) = f(x) \quad \text{for } x \in \Omega, \\
& \frac{\partial y}{\partial n}(x) = 0 \quad \text{for } x \in \partial \Omega,
\end{align*}
\]

and the bilateral box constraints

\[
u_a \leq u \leq u_b,
\]

where \(u = (u_1, \ldots, u_m) = (\kappa_1, \ldots, \kappa_{m'}, c, r) \in \mathbb{R}^m\) is the parameter of dimension \(m := m' + 2\), \(U_{ad} := \{u \in \mathbb{R}^m \mid u_a \leq u \leq u_b\}\) is the admissible parameter set, and \(y \in L^2(\Omega) =: H\) is the state variable.

The domain \(\Omega\) is divided into \(m'\) pairwise disjoint subdomains \(\Omega = \Omega_1 \cup \ldots \cup \Omega_{m'}\), such that \(\kappa_i\) is the diffusion coefficient on \(\Omega_i\). The vector field \(b \in L^\infty(\Omega, \mathbb{R}^d)\) is the given advection, whose strength and orientation can be controlled by the parameter \(c \in \mathbb{R}\). Moreover, the reaction coefficient is given by the parameter \(r > 0\), and \(f \in H\) is the inhomogeneity on the right-hand side of the equation. On the boundary we impose homogeneous Neumann boundary conditions.

The cost functions \(J_1, \ldots, J_{k-1} : H \times U_{ad} \rightarrow \mathbb{R}^k\) are of tracking type with respect to the desired states \(y_1, \ldots, y_{k-1} \in H\), and the cost function \(J_k : H \times U_{ad} \rightarrow \mathbb{R}\) measures the parameter cost. Setting \(V := H^1(\Omega)\) and using the parameter-dependent bilinear form \(a(u, \cdot, \cdot) : V \times V \rightarrow \mathbb{R}\) defined by

\[
a(u, \varphi, \psi) := \sum_{i=1}^{m} u_i a_i(\varphi, \psi)
\]

\[
:= \sum_{i=1}^{m'} \kappa_i \int_{\Omega_i} \nabla \varphi(x) \cdot \nabla \psi(x) \, dx + c \int_{\Omega} b(x) \cdot \nabla \varphi(x) \psi(x) \, dx
\]

\[
+ r \int_{\Omega} \varphi(x) \psi(x) \, dx,
\]

and the corresponding cost function

\[
J_k(y, u) = \frac{1}{2} \left\| y - y_d \right\|^2_{L^2(\Omega)}.
\]
for all \( u \in U_{ad} \) and \( \varphi, \psi \in V \), and the linear functional \( F : V \to \mathbb{R} \) given by \( F(\varphi) := \langle f, \varphi \rangle_H \) for all \( \varphi \in V \), we can write (EPDE) in its weak formulation as: Find \( y \in V \) such that
\[
a(u; y, \varphi) = F(\varphi) \quad \text{for all } \varphi \in V
\] is satisfied. It is possible to show the unique solvability of (7) under some conditions on the parameter \( u \).

**Theorem 3.1.** There are \( \kappa_{\min} \in (0, \infty)^m \), \( c_{\min}, c_{\max} \in \mathbb{R} \) with \( c_{\min} < c_{\max} \) and \( r_{\min} \in (0, \infty) \) such that (7) has a unique solution \( y(u) \in V \) for every parameter \( u = (\kappa, c, r) \in \mathbb{R}^m \) with \( \kappa > \kappa_{\min} \), \( c_{\min} < c < c_{\max} \) and \( r > r_{\min} \).

**Proof.** It is straightforward to show that for all parameters \( u \in \mathbb{R}^m \) the bilinear form \( a(u; \cdot, \cdot) \) and the linear functional \( F \) are continuous, and that there are \( \kappa_{\min} \in (0, \infty)^m \), \( c_{\min}, c_{\max} \in \mathbb{R} \) with \( c_{\min} < c_{\max} \) and \( r_{\min} \in (0, \infty) \) such that \( a(u; \cdot, \cdot) \) is coercive for all \( u = (\kappa, c, r) \in \mathbb{R}^m \) with \( \kappa > \kappa_{\min}, c_{\min} < c < c_{\max} \) and \( r > r_{\min} \). Now the Lax-Milgram Theorem can be applied to show the unique solvability of (7).

With Theorem 3.1 in mind we can introduce the solution operator of the elliptic PDE.

**Definition 3.2.** Define the set \( U_{eq} := (\kappa_{\min}, \infty) \times (c_{\min}, c_{\max}) \times (r_{\min}, \infty) \) with the constants from Theorem 3.1. Let \( S : U_{eq} \to V \hookrightarrow H \) be defined as the solution operator of (7), i.e., the function \( y := S(u) \) solves the weak formulation (7) for any parameter \( u \in U_{eq} \).

**Remark 3.3.** In the following we suppose that it holds \( U_{ad} \subset U_{eq} \).

Using the explicit dependence of the state \( y \) on the parameter \( u \) for all \( u \in U_{ad} \), the essential cost functions \( J_1, \ldots, J_k : U_{ad} \to \mathbb{R} \) can be defined.

**Definition 3.4.** For any \( i \in \{1, \ldots, k\} \) let the essential cost function \( J_i : U_{ad} \to \mathbb{R} \) be given by \( J_i(u) := J_i(S(u), u) \) for all \( u \in U_{ad} \).

For applying the continuation method from Section 2, which is based on Theorem 2.4, to solve this multiobjective parameter optimization problem, the cost functions \( J_1, \ldots, J_k \) need to be twice continuously differentiable. This is the statement of the next lemma.

**Lemma 3.5.** The cost functions \( J_1, \ldots, J_k \) are twice continuously differentiable.

**Proof.** It is clear that the cost function \( J_k \) is twice continuously differentiable. Furthermore, it is possible to show that the solution operator \( S \) of (7) is twice continuously differentiable (this can be shown by rewriting (7) in the form \( e(y, u) = 0 \) and then using the implicit function theorem, cf. [15, Section 1.6]). From this it immediately follows that the cost functions \( J_1, \ldots, J_{k-1} \) are twice continuously differentiable as well.

For later use, we need an explicit formula for the gradients \( \nabla J_1, \ldots, \nabla J_k \). Therefore, we introduce the so-called adjoint equation for all \( i \in \{1, \ldots, k-1\} \): Find \( p \in V \) such that it holds
\[
a(u; \varphi; p) = \langle y_i^1 - S(u), \varphi \rangle_H \quad \text{for all } \varphi \in V.
\] (8)

With the same arguments as in Theorem 3.1 it is possible to show that (8) has a unique solution for all \( u \in U_{eq} \).

**Definition 3.6.** Denote by \( A_i : U_{eq} \to V \hookrightarrow H \) the solution operator of the adjoint equation (8) for all \( i \in \{1, \ldots, k-1\} \).
Now a small computation shows that
\[ J'_i(u)h = \langle S(u) - y'_d, S'(u)h \rangle_H = \partial_u a(u; S(u), A_i(u))h, \]
which yields
\[ \nabla J_i(u) = \begin{pmatrix} \partial_u a(u; S(u), A_i(u))e_1 \\ \vdots \\ \partial_u a(u; S(u), A_i(u))e_m \end{pmatrix} = \begin{pmatrix} a_1(S(u), A_i(u)) \\ \vdots \\ a_m(S(u), A_i(u)) \end{pmatrix} \]
for all \( i \in \{1, \ldots, k - 1\} \). Lastly, it is obvious that \( \nabla J_k(u) = u \).

3.2. The reduced basis method

For computing the Pareto critical set of the problem \( \text{MPOP} \) by the exact continuation method introduced in Section 2.2, the problem (PC-Box) has to be solved numerous times. However, already one gradient evaluation of all cost functions \( \nabla J_1(u), \ldots, \nabla J_k(u) \) involves the solution of one state and \( k - 1 \) adjoint equations. Thus, using a finite element discretization for the weak formulations (7) and (8), which leads to large linear equation systems, is numerically very costly and time consuming. Therefore, the use of reduced-order modelling (ROM) is a common tool to lower the computational costs.

The idea of ROM is to use a low-dimensional subspace \( V^r \subset V \) as a surrogate for the infinite-dimensional space \( V \) in the weak formulations (7) and (8). Given a finite-dimensional reduced-order space \( V^r \subset V \), the reduced-order state equation reads: Find \( y^r \in V^r \) such that
\[ a(u; y^r, \varphi) = F(\varphi) \quad \text{for all } \varphi \in V^r \]
is satisfied.

With the same arguments as in Theorem 3.1 it can be shown that (10) has a unique solution for all \( u \in U_{eq} \). Therefore, we can follow the procedure of Section 3.1 and introduce the solution operator \( S' : U_{eq} \to V^r \subset V \leftarrow H \) of the ROM state equation (10) and consequently the ROM essential cost functions \( J'_1, \ldots, J'_k \), which are defined by \( J'_i(u) := \langle S'(u), u \rangle \) for all \( u \in U_{ad} \) and all \( i \in \{1, \ldots, k\} \). Again, it can be shown that the functions \( J'_1, \ldots, J'_k \) are twice continuously differentiable so that they fit into the framework of Theorem 2.4. The gradient of the cost functions can also be displayed by the reduced-order adjoint equations
\[ a(u; \varphi, p^r) = \langle y'_d - S'(u), \varphi \rangle_H \quad \text{for all } \varphi \in V^r, \]
for all \( i \in \{1, \ldots, k - 1\} \), whose solution operator we denote by \( A'_i : U_{eq} \to V^r \subset V \leftarrow H \). With this definition it holds
\[ \nabla J'_i(u) = \begin{pmatrix} \partial_u a(u; S'(u), A'_i(u))e_1 \\ \vdots \\ \partial_u a(u; S'(u), A'_i(u))e_m \end{pmatrix} = \begin{pmatrix} a_1(S'(u), A'_i(u)) \\ \vdots \\ a_m(S'(u), A'_i(u)) \end{pmatrix} \]
for all \( i \in \{1, \ldots, k - 1\} \). Moreover, we have \( \nabla J'_k(u) = u = \nabla J_k(u) \).

In this paper we use a particular model-order reduction technique, namely the reduced basis (RB) method (see e.g. \cite{20, 13, 25}). In the RB method the snapshot space \( V^r \) is spanned by solutions of the state equation and the adjoint equations to different parameter values \( u \in U_{ad} \). The reduced basis is then given by an orthonormal basis \( (\Phi_1, \ldots, \Phi_N) \) of the space \( V^r \).

By using the RB method we introduce an error in the state equation, which transfers to the cost functions, its gradients and eventually to the Pareto critical set, which we want to compute. In
Section 2.3, two strategies were presented to deal with the inflicted inexactness in the gradients of the multiobjective optimization problem. Both are based on the estimates [4] for the errors in the gradients of the cost functions. Thus, when applying the RB method we need to ensure these estimates. This is done by using the well-known greedy algorithm (cf. [5]). Given a sufficiently fine finite parameter training set \( \mathcal{P} \subset \mathcal{U}_{ad} \) new solution snapshots are computed until the error in the gradients of all cost functions is smaller than the predefined error tolerance for all parameters in \( \mathcal{P} \). The parameter for the new snapshots is thereby chosen as the one for which the error in the gradient is the largest. The procedure is summarized in Algorithm 4.

**Algorithm 4 Greedy Algorithm**

Given: Parameter set \( \mathcal{P} \subset \mathcal{U}_{ad} \), greedy tolerances \( \varepsilon_1, \ldots, \varepsilon_k > 0 \).

1. Choose \( u \in \mathcal{P} \), compute \( \mathcal{S}(u), \mathcal{A}_1(u), \ldots, \mathcal{A}_{k-1}(u) \).
2. Set \( V^r = \text{span}\{\mathcal{S}(u), \mathcal{A}_1(u), \ldots, \mathcal{A}_{k-1}(u)\} \) and compute the reduced basis by orthonormalization.
3. while \( \max_{u \in \mathcal{P}} \max_{i \in \{1, \ldots, k-1\}} \| \nabla J_i(u) - \nabla J_i^r(u) \|_2 > \varepsilon_i \) do
   4. Choose \( (\bar{u}, i) = \arg \max_{u \in \mathcal{P}, i \in \{1, \ldots, k-1\}} \| \nabla J_i(u) - \nabla J_i^r(u) \|_2 \).
   5. Compute \( \mathcal{S}(\bar{u}) \) and \( \mathcal{A}_i(\bar{u}) \).
   6. Set \( V^r = \text{span}\{V^r \cup \{\mathcal{S}(\bar{u}), \mathcal{A}_i(\bar{u})\}\} \) and compute the reduced basis by orthonormalization.
7. end while

### 3.3. Error estimation for the gradients

In the greedy procedure in Algorithm 4, the error between the full-order and the reduced-order gradients has to be evaluated. There are two strategies to do so.

1. The full-order gradients are computed and stored at the beginning of the greedy procedure. Therefore, in each greedy iteration, only the reduced-order gradients have to be computed and the error can be easily evaluated. Of course, this implies large computational costs at the beginning of the greedy procedure. This method is called strong greedy algorithm (cf. [12, 5]).

2. An a-posteriori error estimator for the errors in the gradient is used, which can be efficiently evaluated. This results in computational costs for the greedy algorithm, which only depend on the reduced-order dimension \( N \).

To be able to follow the second strategy we introduce a rigorous a-posteriori error estimator for the error in the gradient of the cost functions. Using the gradient representations [9] and [12] we can write for \( i \in \{1, \ldots, k-1\} \)

\[
\| \nabla J_i(u) - \nabla J_i^r(u) \|_2^2 = \sum_{j=1}^m |a_j(\mathcal{S}(u), \mathcal{A}_i(u)) - a_j(\mathcal{S}^r(u), \mathcal{A}_i^r(u))|^2.
\]

Due to the bilinearity and the continuity of \( a_1, \ldots, a_m \) and the triangle inequality, we can further write

\[
|a_j(\mathcal{S}(u), \mathcal{A}_i(u)) - a_j(\mathcal{S}^r(u), \mathcal{A}_i^r(u))| \\
\leq |a_j(\mathcal{S}(u) - \mathcal{S}^r(u), \mathcal{A}_i(u))| + |a_j(\mathcal{S}(u) - \mathcal{S}^r(u), \mathcal{A}_i(u) - \mathcal{A}_i^r(u))| \\
+ |a_j(\mathcal{S}^r(u), \mathcal{A}_i(u) - \mathcal{A}_i^r(u))| \\
\leq C_j \left( \| \mathcal{S}(u) - \mathcal{S}^r(u) \|_V \| \mathcal{A}_i(u) \|_V + \| \mathcal{S}(u) - \mathcal{S}^r(u) \|_V \| \mathcal{A}_i(u) - \mathcal{A}_i^r(u) \|_V \\
+ \| \mathcal{S}^r(u) \|_V \| \mathcal{A}_i(u) - \mathcal{A}_i^r(u) \|_V \right)
\]

(13)

(14)
for all $j \in \{1, \ldots, m\}$.
Therefore, we need a-posteriori error estimators for the state and the adjoint equations in order to be able to estimate the approximation error induced in the gradients. To this end, we use the following well-known estimators (cf. [26]).

\[
\|S(u) - S^r(u)\|_V \leq \frac{\|r_S(u)\|_V}{\alpha(u)} =: \Delta_S(u),
\]
\[
\|A_i(u) - A_i^r(u)\|_V \leq \frac{\|r_{A_i}(u)\|_V}{\alpha(u)} + \Delta_{A_i}(u) =: \Delta_{A_i}(u),
\]

where the residuals $r_S(u)$ and $r_{A_i}(u)$ are given by

\[
\langle r_S(u), \varphi \rangle_{V', V} := F(\varphi) - a(u; S^r(u), \varphi)
\]
\[
\langle r_{A_i}(u), \varphi \rangle_{V', V} := (y_i^d - S^r(u), \varphi)_H - a(u; \varphi, A_i^r(u))
\]

for all $\varphi \in V$.

For methods on how to estimate $\alpha(u)$ and to evaluate the terms $\|r_S(u)\|_V$ and $\|r_{A_i}(u)\|_V$ efficiently, we refer for example to [26].

**Remark 3.7.** Since $J_k = J_r^k$, the gradients of the two functions also coincide, so that the $\nabla J_k$ is approximated exactly by $\nabla J_r^k$. ■

### 4. Numerical results

In this section we will numerically investigate the application of the continuation method presented in Section 2 to the PDE-constrained multiobjective optimization problem using the reduced basis method in Section 3.

For the discretization of the state and adjoint equations we used linear finite elements with 714 degrees of freedom.

#### 4.1. Generation of the reduced basis

For investigating the generation of the reduced basis by the greedy algorithm in Algorithm 4 we consider the MPOP

\[
\begin{pmatrix}
J_1(u) \\
J_2(u)
\end{pmatrix} = \begin{pmatrix}
\frac{1}{2} \|S(u) - y_1^d\|^2_H \\
\frac{1}{2} \|u\|^2_R
\end{pmatrix}
\]

with $u = (\kappa_1, \kappa_2, c, r), \, \Omega_1 = (0,1) \times (0,0.5), \, \Omega_2 = (0,1) \times (0.5,1)$, and the admissible parameter set

$U_{ad} = \{u = (\kappa_1, \kappa_2, c, r) \in \mathbb{R}^4 \mid 0.2 \leq \kappa_i \leq 5 \, (i = 1,2), \, c = 0, \, r = 0.5\}$.

The reason for setting $c = 0$ in this example is that the coercivity constant $\alpha(u)$ of the bilinear form $a(u; \cdot, \cdot)$ is explicitly given by $\alpha(u) = \min\{\kappa_1, \kappa_2, r\}$ for all $u \in U_{ad}$, so that we expect a good efficiency of the error estimator of both the state and adjoint equations. This is verified by the results shown in Figure 3(a), where the efficiency of the error estimator for both equations is shown for a given reduced basis for 1000 randomly chosen parameter values. However, the resulting efficiency of the error estimator for the error in the gradient is between $10^3$ and $10^6$ (see Figure 3(b)) and thus not well suited for a greedy procedure, which depends on a good error estimation. The huge overestimation of the error estimator is mainly due to the use of the triangle inequality (13) and the continuity estimates (14), as can be seen in Figure 3(b).

Compared to the strong greedy algorithm, we can see in Table 4 that this overestimation results
Overestimation of errors in state and adjoint equation

\begin{figure}[h]
\centering
\includegraphics[width=\textwidth]{figure3}
\caption{Overestimations for 1000 randomly selected parameter values}
\end{figure}

Table 1: Number of basis functions for different error bounds

| Error bound | Strong Greedy | Error Estimate |
|-------------|---------------|----------------|
| $\varepsilon = 1 e^{-6}$ | 24 | 56 |
| $\varepsilon = 1 e^{-5}$ | 20 | 50 |
| $\varepsilon = 1 e^{-4}$ | 16 | 40 |
| $\varepsilon = 1 e^{-3}$ | 12 | 32 |
| $\varepsilon = 1 e^{-2}$ | 12 | 26 |
| $\varepsilon = 1 e^{-1}$ | 10 | 20 |

in far more basis elements than actually needed to reach the given error bound. Since we want to investigate the influence of the error bounds in the estimate (4) on the problem, we want that the estimate (4) is satisfied sharply by the RB. Therefore, we will not use the error estimator to generate the basis, but instead use the strong greedy algorithm.

4.2. Application of the continuation methods to an MPOP

For the numerical investigation of the continuation method applied to a PDE-constrained multi-objective parameter optimization problem together with the use of the reduced basis method, we consider the MPOP

\[
\begin{pmatrix}
    J_1(u) \\
    J_2(u) \\
    J_3(u) \\
    J_4(u)
\end{pmatrix} =
\begin{pmatrix}
    \frac{1}{2} \| S(u) - S((0.7, 0.8, 0.5)) \|_{L^2}^2 \\
    \frac{1}{2} \| S(u) - S((2, 0.5, 0.5)) \|_{L^2}^2 \\
    \frac{1}{2} \| S(u) - S((3, -0.5, 0.5)) \|_{L^2}^2 \\
    \frac{1}{2} \| u \|_{H^3}^2
\end{pmatrix}
\]

(16)

with $u = (\kappa, c, r)$ and

\[ U_{ad} = \{ u = (\kappa, c, r) \in \mathbb{R}^3 \mid 0.5 \leq \kappa \leq 3, -1 \leq c \leq 1, r = 0.5 \}, \]

i.e., the reaction parameter $r$ is a constant so that we only optimize the diffusivity in the whole domain $\Omega$ and the strength and orientation of the advection field $b$. Thus, this can be seen as a problem with two parameters.

As described before, the reduced basis is generated by the strong greedy Algorithm 4 where the error bounds $\varepsilon_1, \ldots, \varepsilon_4$ are chosen in accordance with the estimate (4). As a reference, the exact solution of (16) (via exact continuation and FEM discretization of the weak formulations) is shown in Figure 4.
Remark 4.1. Since (16) is constrained to a box, we have to use a constrained version of the exact continuation method (cf. [14]) to calculate Pareto critical points that lie on the boundaries of (16). But note that for this example, all Pareto critical points on the boundary are also Pareto critical if we ignore the constraints. In other words, for each Pareto critical point \( \bar{u} \) on the boundary, there is a sequence of Pareto critical points in the interior that converges to \( \bar{u} \). By continuity of \( DJ \), the gradients of the (active) inequality constraints in the KKT conditions can be ignored. As a result, we can treat (16) as an unconstrained problem that we only solve in a certain area. ■

As a first test, we will compare the time needed to compute the exact solution of (16) with the time needed for Strategy 1 and 2. For the error bounds we choose \( \epsilon = (0.03,0.03,0.01,0.01) \) and for the box radius we choose \( r = \frac{3-0.5}{2^5} \approx 0.0049 \). The results are shown in Figure 5.

Figure 5: Results of Strategy 1 (left) and 2 (right) for the MPOP (16) with \( \epsilon = (0.03,0.03,0.01,0.01) \)

All three methods were implemented in Matlab. For the solution of the subproblems (PC-Box), (\( \epsilon \)-PC-Box) and (\( \partial \epsilon \)-PC-Box), the SQP-Algorithm of fmincon was used. (For increased stability during the continuation, each subproblem where the SQP-Algorithm found an optimal value larger than zero was restarted using the Interior-Point-Method and the Active-Set-Method of fmincon). The runtime, number of boxes and number of subproblems needed are shown in Table 2. When comparing Strategy 1 and Strategy 2, we see that Strategy 2 needs about 20 times fewer boxes and solutions of subproblems than Strategy 1. This is to be expected, since Strategy 2 only computes a covering of the boundary of \( P_2^r \), i.e., of a lower dimensional set. When comparing the actual runtime, Strategy 2 is about 5 times faster than Strategy 1, since the subproblems in
Table 2: Comparison of the performance of the exact continuation method, Strategy 1 and Strategy 2 for Example 16. The number of subproblems is split up in subproblems for the continuation and initialization (cf. Section 2.4).

| Algorithm    | # Boxes | # Subproblems | Runtime (in seconds) |
|--------------|---------|---------------|----------------------|
| Exact cont.  | 15916   | 18721 + 25    | 17501s               |
| Strategy 1   | 21750   | 24490 + 25    | 1426s                |
| Strategy 2   | 899     | 1027 + 225    | 276s                 |

Strategy 2 are more expensive to solve than the ones in Strategy 1 (cf. Remark 2.12). Finally, Strategy 2 is about 63 times faster than the exact continuation method with FEM discretization, illustrating the large increase in efficiency we gain from our approach.

Although it is a lot quicker to use inexact gradients from ROM instead of the exact gradients via FEM, it is important to keep in mind that our methods are computing a superset of the actual Pareto critical set. For example, in Figure 5, the right side of the lower connected component is only approximated poorly by $P_2^r$. Therefore, we will now investigate the influence of the error bounds $\epsilon = (\epsilon_1, \epsilon_2, \epsilon_3, \epsilon_4)$ on $P_2^r$, by applying Strategy 2 with reduced bases for different values of $\epsilon$. Note that in all our tests we set $\epsilon_4 = 0.01$, although the error in the gradient of the fourth cost function is zero for all parameters. This is done to make the solution of $(\partial_0 P_0 - \partial_0 P_1)$ numerically stable (cf. Remark 2.12).

The results of our experiment can be seen in Figure 6. Generally, as expected, the boundary $\partial P_2^r$ encloses the Pareto critical set $P_c$ sharper and sharper for decreasing $\epsilon$. Moreover, we observe that it is crucial to choose an $\epsilon$ which is not too large: For the value $\epsilon = (0.1, 0.1, 0.1, 0.01)$ the shape of the boundary $\partial P_2^r$ implies that the set $P_2^r$ is connected, i.e., we lose the topological information that the Pareto critical set actually consists of two connected components. Decreasing $\epsilon$ to $\epsilon = (0.0885, 0.0885, 0.0885, 0.01)$ we are in the limit case in which the boundary $\partial P_2^r$ touches the box constraints at around $(2.3, 1)$, so that this is the approximate $\epsilon$ for which we regain the basic topological information of a disconnected Pareto critical set.

If we compare the results for $\epsilon = (0.03, 0.03, 0.03, 0.01), \epsilon = (0.03, 0.03, 0.01, 0.01)$ and $\epsilon = (0.03, 0.01, 0.01, 0.01)$, the influence of changing one component of $\epsilon$ becomes obvious. For $\epsilon = (0.03, 0.03, 0.03, 0.01)$ the set $\partial P_2^r$ encloses the set $P_c$ quite sharply at the upper connected component and at the left part of the lower connected component, where the second and third component of the corresponding KKT-multipliers $\alpha$ are small. On the other hand, in the right part of the lower connected component of $P_c$, where the second and third component of the corresponding KKT-multipliers are relatively large, the deviation of $\partial P_2^r$ to $P_c$ is still large. Consequently, first reducing $\epsilon_3$ and then also $\epsilon_2$ from 0.03 to 0.01 leads to a clearly visible sharper enclosing of this part of $P_c$.

5. Conclusion and outlook

In this article, we present a way to efficiently solve multiobjective parameter optimization problems of elliptic PDEs by combining the reduced basis method from PDE-constrained optimization with the continuation approach from multiobjective optimization, which computes a box covering of the Pareto critical set. Using the RB method in this setting introduces an error in the objective functions and their gradients that has to be considered when solving the MOP. To this end, we require that the reduced basis guarantees error bounds for the gradients of the objective functions. These error bounds are then incorporated into the KKT optimality conditions for MOPs to derive a tight superset $P_2^r$ of the actual Pareto (critical) set. This superset can be computed using a straightforward modification of the continuation method for MOPs (Strategy 1). Since $P_2^r$ has the same dimensions as the variable space of the MOP, we afterwards
Figure 6: Results of Strategy 2 for different values of $\epsilon$

present a second method that only computes the boundary $\partial P_2^r$ of $P_2^r$ (Strategy 2). We do this by showing that $\partial P_2^r$ can be written as the level set of a differentiable mapping, which again enables the use of a continuation approach to compute it. For constructing the reduced basis, we use a greedy procedure which incorporates, and thus ensures, the error bounds for the gradients of the objective functions.

Our numerical tests show that the presented a-posteriori error estimator for the error in the gradients is not well-suited for the application in a greedy procedure due to its bad efficiency. Therefore, a strong greedy algorithm is used to build the reduced basis. Concerning the solution of the MPOP we investigate two aspects: First, the runtimes of our methods are compared. In our case, Strategy 1 is about 13 times and Strategy 2 about 63 times faster than the exact solution of the MPOP (via the classical continuation method with FEM discretization). Second, the influence of the error bound for the gradients of the objective functions is investigated. As expected, a smaller error bound leads to a tighter covering of the Pareto critical set. Moreover, we observe that single components of the error bound strongly influence the tightness of the covering in areas, in which the corresponding components of the KKT-multipliers are large. Thus, by individually adapting the single components of the error bound, we can nicely control the tightness of the covering.

For future work, there are some theoretical and practical aspects that should be investigated further:

- As mentioned in Remark 2.12, in certain situations there can be difficulties when solving the problem $\partial\epsilon P_2$-Box. In these situations, specialized methods that take these difficulties into account should be developed and used instead of standard methods for constrained optimization.

- If the number of objectives of the MPOP is larger than the number of variables, it may be possible to combine our approaches in this article with the hierarchical decomposition of the Pareto critical set presented in [10].

- The development of a more efficient a-posteriori error estimator for the error in the gradients of the objective functions would allow to use it in the greedy procedure. In that way,
the expensive strong greedy procedure would be avoided in the offline phase. One way to do so might be the application of localized RB methods, see e.g. [20].

- As explained in the globalization approach in Section 2.4, we have to use multiple initial points to ensure that we find all connected components of $P^*_2$ (and faces of $\partial P^*_2$). Due to the local nature of the continuation method, this approach can potentially be parallelized, increasing the efficiency of our methods even more.

- If a decision maker is present with a certain preference, it may be worth to steer our continuation method in a direction that results from that preference instead of approximating the complete Pareto set. For the case with exact gradients, this was done in [27].

References

[1] S. Banholzer, D. Beermann, and S. Volkwein. POD-Based Bicriterial Optimal Control by the Reference Point Method. *IFAC-PapersOnLine*, 49(8):210–215, 2016.

[2] S. Banholzer, D. Beermann, and S. Volkwein. POD-Based Error Control for Reduced-Order Bicriterial PDE-Constrained Optimization. *Annual Reviews in Control*, 44:226–237, 2017.

[3] D. Beermann, M. Dellnitz, S. Peitz, and S. Volkwein. POD-based multiobjective optimal control of PDEs with non-smooth objectives. In *Proceedings in Applied Mathematics and Mechanics (PAMM)*, pages 51–54, 2017.

[4] D. Beermann, M. Dellnitz, S. Peitz, and S. Volkwein. Set-Oriented Multiobjective Optimal Control of PDEs using Proper Orthogonal Decomposition. In *Reduced-Order Modeling (ROM) for Simulation and Optimization*, pages 47–72. Springer, 2018.

[5] A. Buffa, Y. Maday, A. T. Patera, C. Prudhomme, and G. Turinici. A priori convergence of the greedy algorithm for the parametrized reduced basis method. *ESAIM: Mathematical Modelling and Numerical Analysis*, 46(3):595–603, 2012.

[6] T. Chugh, K. Sindhiya, J. Hakanen, and K. Miettinen. A survey on handling computationally expensive multiobjective optimization problems with evolutionary algorithms. *Soft Computing*, pages 1–30, 2017.

[7] C. A. Coello Coello, G. B. Lamont, and D. A. Van Veldhuizen. *Evolutionary Algorithms for Solving Multi-Objective Problems*, volume 2. Springer Science & Business Media, 2007.

[8] M. Dellnitz, O. Schütze, and T. Hestermeyer. Covering Pareto Sets by Multilevel Subdivision Techniques. *Journal of Optimization Theory and Applications*, 124(1):113–136, Jan 2005.

[9] M. Ehrgott. *Multicriteria optimization*. Springer Berlin Heidelberg New York, 2 edition, 2005.

[10] B. Gebken, S. Peitz, and M. Dellnitz. On the hierarchical structure of Pareto critical sets. *Journal of Global Optimization*, 73(4):891–913, 2019.

[11] M. A. Grepl and A. T. Patera. A posteriori error bounds for reduced-basis approximations of parametrized parabolic partial differential equations. *ESAIM: Mathematical Modelling and Numerical Analysis*, 39(1):157–181, 2005.

[12] B. Haasdonk, J. Salomon, and B. Wohlmuth. A reduced basis method for the simulation of american options. In *Numerical Mathematics and Advanced Applications 2011*, pages 821–829, Berlin, Heidelberg, 2013. Springer Berlin Heidelberg.

[13] J. Hesthaven, G. Rozza, and B. Stamm. Certified reduced basis methods for parametrized partial differential equations. *SpringerBriefs in Mathematics*, 2016.

[14] C. Hillermeier. *Nonlinear Multiobjective Optimization: A Generalized Homotopy Approach*. Birkhäuser Basel, 2001.

[15] M. Hinze, R. Pinnau, M. Ulbrich, and S. Ulbrich. *Optimization with PDE Constraints*. Springer, 2009.
Acknowledgment

This research was funded by the DFG Priority Programme 1962 “Non-smooth and Complementarity-based Distributed Parameter Systems”.
A. Proof of Theorem 2.11

To prove Theorem 2.11, we first have to investigate some of the properties of the optimization problem (6). This problem is quadratic with linear equality and inequality constraints. We will first investigate the uniqueness of the solution in the following lemma.

Lemma A.1. Let \( u \in \varphi^{-1}(0) \) and let \( \alpha^1 \) and \( \alpha^2 \) be two solutions of (6) with \( \alpha^1 \neq \alpha^2 \). Then \( \omega(\alpha) = 0 \) for all \( \alpha \in \text{span}(\{\alpha^1, \alpha^2\}) \) and

\[
\text{span}(\{\alpha^1, \alpha^2\}) \cap \ker(DJ(u)^\top) \neq \emptyset.
\]

Proof. For \( c_1, c_2 \in \mathbb{R} \setminus \{0\} \) we have

\[
\omega(c_1 \alpha^1 + c_2 \alpha^2) = (c_1 \alpha^1 + c_2 \alpha^2)^\top (DJ'(u)DJ'(u)^\top - \epsilon \epsilon^\top)(c_1 \alpha^1 + c_2 \alpha^2)
\]

\[
= c_2^2 \omega(\alpha^1) + 2(c_1 \alpha^1 \top DJ'(u)DJ'(u)^\top c_2 \alpha^2 - c_1 \alpha^1 \top \epsilon \epsilon^\top c_2 \alpha^2) + c_2^2 \omega(\alpha^2)
\]

\[
= 2(c_1 \alpha^1 \top DJ'(u)DJ'(u)^\top c_2 \alpha^2 - c_1 \alpha^1 \top \epsilon \epsilon^\top c_2 \alpha^2)
\]

\[
= 2c_1c_2((DJ'(u)^\top \alpha^1)(DJ'(u)^\top \alpha^2) - (\epsilon^\top \alpha^1)(\epsilon^\top \alpha^2)).
\]

From \( \omega(\alpha^1) = \omega(\alpha^2) = 0 \) it follows that \( \epsilon^\top \alpha^1 = \|DJ'(u)^\top \alpha^1\| \) and \( \epsilon^\top \alpha^2 = \|DJ'(u)^\top \alpha^2\| \). Let \( \angle \) be the angle between \( DJ'(u)^\top \alpha^1 \) and \( DJ'(u)^\top \alpha^2 \). Then

\[
\omega(c_1 \alpha^1 + c_2 \alpha^2)
\]

\[
= 2c_1c_2(cos(\angle)\|DJ'(u)^\top \alpha^1\|\|DJ'(u)^\top \alpha^2\| - \|DJ'(u)^\top \alpha^1\|\|DJ'(u)^\top \alpha^2\|)
\]

\[
= 2c_1c_2(cos(\angle) - 1)\|DJ'(u)^\top \alpha^1\|\|DJ'(u)^\top \alpha^2\|.
\]

Assume \( cos(\angle) \neq 1 \) (i.e., \( cos(\angle) - 1 < 0 \)), \( \|DJ'(u)^\top \alpha^1\| \neq 0 \) and \( \|DJ'(u)^\top \alpha^2\| \neq 0 \). If we choose \( c_1 = t \) and \( c_2 = 1 - t \) for \( t \in (0, 1) \), then \( t \alpha^1 + (1 - t) \alpha^2 \in \Delta_k \) and \( \omega(t \alpha^1 + (1 - t) \alpha^2) < 0 \), which contradicts \( u \in \varphi^{-1}(0) \). If \( \|DJ'(u)^\top \alpha^1\| = 0 \) or \( \|DJ'(u)^\top \alpha^2\| = 0 \) then (17) holds for \( \hat{\alpha} = \alpha^1 \) or \( \hat{\alpha} = \alpha^2 \), respectively. If \( cos(\angle) - 1 = 0 \) then \( DJ'(u)^\top \alpha^1 \) and \( DJ'(u)^\top \alpha^2 \) are linearly dependent, so there are \( c_1, c_2 \in \mathbb{R} \setminus \{0\} \) such that \( DJ'(u)^\top \alpha^1 = 0 \) for \( \alpha = c_1 \alpha^1 + c_2 \alpha^2 \). In particular, in any case we must have \( \omega(\alpha) = 0 \) for all \( \alpha \in \text{span}(\{\alpha^1, \alpha^2\}) \).

The previous lemma implies that for \( k = 2 \), the solution of (6) for \( u \in \varphi^{-1}(0) \) is non-unique iff \( DJ'(u)DJ'(u)^\top - \epsilon \epsilon^\top = 0 \). For \( k > 2 \), we can only have non-uniqueness if (17) holds. If we consider the dimensions of the spaces in (17), we see that in the generic case, it can only hold if

\[
\dim(\text{span}(\{\alpha^1, \alpha^2\}) \cap \ker(DJ(u)^\top)) \geq 1
\]

\[
\iff 2 + k - rk(DJ(u)^\top) - k \geq 1
\]

\[
\iff rk(DJ(u)^\top) \leq 1,
\]

i.e., if all gradients of the objectives are linearly dependent in \( u \). This motivates us to assume that in general, the solution of (6) is unique for almost all \( u \in \varphi^{-1}(0) \).

We will now investigate the differentiability of \( \varphi \). Our strategy is to apply the implicit function theorem to the KKT conditions of (6) to obtain a differentiable function \( \phi \) that maps a point \( u \in \mathbb{R}^n \) onto the solution of (6) in \( u \). This would imply the differentiability of \( \varphi \) via concatenation with \( \omega \). An obvious problem here is the fact that (6) has inequality constraints which, when activated or deactivated under variation of \( u \), lead to non-differentiability in \( \phi \). Note that an inequality constraint being active means that one component of \( \alpha \) is zero, i.e., one of the objective functions has no impact on the current problem. Thus, for our theoretical purposes,
if there is an active inequality constraint in (6) we will just ignore the corresponding objective function. This approach is strongly related to the hierarchical decomposition of the Pareto critical set (cf. [10]).

For the reasons mentioned above, we will now consider the case where the solution of (6) is strictly positive in each component. The following lemma shows a technical result that will be used in a later proof.

**Lemma A.2.** Let \( u \in \varphi^{-1}(0) \) and let \( \bar{\alpha} \in \Delta_k \) be a solution of (6) with \( \alpha_i > 0 \) \( \forall i \in \{1, ..., k\} \). Then \( \bar{\alpha} \) is unique if and only if there is no \( \beta \in \mathbb{R}^k \setminus \{0\} \) with \( \omega(\beta) = 0 \) and \( \sum_{i=1}^k \beta_i = 0 \).

**Proof.** We will show that \( \alpha \) is non-unique if and only if there is some \( \beta \in \mathbb{R}^k \) with \( \omega(\beta) = 0 \) and \( \sum_{i=1}^k \beta_i = 0 \).

\( \Rightarrow \): Let \( \bar{\alpha} \) be another solution of (6). Then, as in the proof of Lemma A.1 we must have \( \omega(c_1\bar{\alpha} + c_2\bar{\alpha}) = 0 \) for all \( c_1, c_2 \in \mathbb{R} \). This means we can choose \( \beta = \bar{\alpha} - \bar{\alpha} \).

\( \Leftarrow \): Let \( \beta \in \mathbb{R}^k \) with \( \omega(\beta) = 0 \) and \( \sum_{i=1}^k \beta_i = 0 \). Let \( s > 0 \) be small enough such that \( \bar{\alpha} + s\beta \in \Delta_k \). Then, as in (18), we have

\[
\omega(\bar{\alpha} + s\beta) = 2s(\cos(\angle) - 1)\|DJ^r(u)^\top \bar{\alpha}\|\|DJ^r(u)^\top \beta\| \leq 0.
\]

Since by assumption \( \varphi(u) = 0 \) we must have \( \omega(\bar{\alpha} + s\beta) = 0 \), so \( \bar{\alpha} + s\beta \) is another solution of (6). \( \square \)

To be able to use the KKT conditions of (6) to obtain its solution, we have to make sure that these conditions are sufficient. Since (6) is a quadratic problem, this means we have to show that the matrix in the objective \( \omega \) is positive semidefinite.

**Lemma A.3.** Let \( u \in \varphi^{-1}(0) \) and let \( \bar{\alpha} \in \Delta_k \) be the unique solution of (6) with \( \bar{\alpha}_i > 0 \) \( \forall i \in \{1, ..., k\} \). Then \( \omega(\beta) \geq 0 \) for all \( \beta \in \mathbb{R}^k \). In particular, \( DJ(u)DJ(u)^\top - \epsilon \epsilon^\top \) is positive semidefinite.

**Proof.** Assume there is some \( \beta \in \mathbb{R}^k \) with \( \omega(\beta) < 0 \), i.e., \( \epsilon^\top \beta > \|DJ^r(u)^\top \beta\| \). We distinguish between two cases:

Case 1: \( \sum_{i=1}^k \beta_i = 0 \): Similar to the proof of Lemma A.1 we get

\[
\omega(\bar{\alpha} + s\beta) < 2s((DJ^r(u)^\top \bar{\alpha})(DJ^r(u)^\top \beta) - (\epsilon^\top \bar{\alpha})(\epsilon^\top \beta)) < 2s(\cos(\angle) - 1)\|DJ^r(u)^\top \bar{\alpha}\|\|DJ^r(u)^\top \beta\| \leq 0
\]

for all \( s > 0 \). In particular, since \( \bar{\alpha} \) is positive, there is some \( \bar{s} > 0 \) such that \( \bar{\alpha} + \bar{s}\beta \notin \Delta_k \) with \( \omega(\bar{\alpha} + \bar{s}\beta) < 0 \), which is a contradiction.

Case 2: \( \sum_{i=1}^k \beta_i \neq 0 \). W.l.o.g. assume that \( \sum_{i=1}^k \beta_i = 1 \). Consider

\( \bar{\omega} : \mathbb{R} \to \mathbb{R}, \ s \mapsto \omega(\bar{\alpha} + s(\beta - \bar{\alpha})) \)

Then \( \bar{\omega}(0) = 0 \) and \( \bar{\omega}(1) < 0 \). By assumption we must have \( \bar{\omega}(s) > 0 \) for all \( s \) such that \( \bar{\alpha} + s(\beta - \bar{\alpha}) \in \Delta_k \). By continuity of \( \bar{\omega} \) there must be some \( s^* \) with \( \bar{\omega}(s^*) = 0 \). Let \( \bar{\beta} := \bar{\alpha} + s^*(\beta - \bar{\alpha}) \). Using (18) we get

\[
\omega(\bar{\alpha} + ts^*(\beta - \bar{\alpha})) = \omega((1-t)\bar{\alpha} + t\bar{\beta}) = 2t(1-t)(\cos(\angle) - 1)\|DJ^r(u)^\top \bar{\alpha}\|\|DJ^r(u)^\top \bar{\beta}\| \leq 0
\]

for all \( t \in (0, 1) \), which is a contradiction. \( \square \)

The previous results now allow us to prove Theorem 2.11.
Theorem 2.11  Let $\bar{u} \in \varphi^{-1}(0)$ such that (6) has a unique solution $\bar{\alpha} \in \Delta_k$ with $\bar{\alpha}_i > 0$ for all $i \in \{1, ..., k\}$. Let (6) be uniquely solvable in a neighborhood of $\bar{u}$. Then there is an open set $U \subseteq \mathbb{R}^n$ with $\bar{u} \in U$ such that $\varphi|_U$ is continuously differentiable.

Proof. The KKT conditions for (6) are

$$
(DJ(u)DJ(u)^T - \epsilon\epsilon^T)\alpha - \begin{pmatrix} \lambda + \mu_1 \\ \vdots \\ \lambda + \mu_k \end{pmatrix} = 0,
$$

$$
\sum_{i=1}^k \alpha_i - 1 = 0,
$$

$$
\alpha_i \geq 0 \forall i \in \{1, ..., k\},
$$

$$
\mu_i \geq 0 \forall i \in \{1, ..., k\},
$$

for $\lambda \in \mathbb{R}$ and $\mu \in \mathbb{R}^k$. By Lemma A.3 these conditions are sufficient for optimality. By our assumption there is an open set $U'$ with $\bar{u} \in U'$ such that the solution of (6) is unique and positive. Thus, on $U'$, (19) is equivalent to

$$
(DJ(u)DJ(u)^T - \epsilon\epsilon^T)\alpha - \begin{pmatrix} \lambda \\ \vdots \\ \lambda \end{pmatrix} = 0,
$$

$$
\sum_{i=1}^k \alpha_i - 1 = 0.
$$

for some $\lambda \in \mathbb{R}$. This system can be rewritten as $G(u, (\alpha, \lambda)) = 0$ for

$$
G : \mathbb{R}^n \times \mathbb{R}^{k+1} \to \mathbb{R}^{k+1}, \quad (u, (\alpha, \lambda)) \mapsto \left((DJ(u)DJ(u)^T - \epsilon\epsilon^T)\alpha - \begin{pmatrix} \lambda \\ \vdots \\ \lambda \end{pmatrix}, \sum_{i=1}^k \alpha_i - 1\right).
$$

Derivating $G$ with respect to $(\alpha, \lambda)$ yields

$$
D_{(\alpha, \lambda)}G(u, (\alpha, \lambda)) = \left((DJ(u)DJ(u)^T - \epsilon\epsilon^T)(-1, \ldots, -1)^T, 0\right) \in \mathbb{R}^{(k+1) \times (k+1)}.
$$

Let $\bar{\lambda} \in \mathbb{R}$ such that $G(\bar{u}, (\bar{\alpha}, \bar{\lambda})) = 0$. (Note that uniqueness of $\bar{\alpha}$ implies uniqueness of $\bar{\lambda}$ here.) For $D_{(\alpha, \lambda)}G(\bar{u}, (\bar{\alpha}, \bar{\lambda}))$ to be singular, there would have to be some $v = (v^1, v^2) \in \mathbb{R}^{k+1}$ with

$$
0 = D_{(\alpha, \lambda)}G(\bar{u}, (\bar{\alpha}, \bar{\lambda}))v = \left((DJ(\bar{u})DJ(\bar{u})^T - \epsilon\epsilon^T)v^1 - (v^2, \ldots, v^2)^T, \sum_{i=1}^k v_i^1\right)
$$

and thus

$$
0 = v^1^T(DJ(\bar{u})DJ(\bar{u})^T - \epsilon\epsilon^T)v^1 - v^1^T(v^2, \ldots, v^2)^T
$$

$$
= v^1^T(DJ(\bar{u})DJ(\bar{u})^T - \epsilon\epsilon^T)v^1 - v_2 \sum_{i=1}^k v_i^1
$$

$$
= w(v^1).
$$

By Lemma A.2 this is a contradiction to the assumption that $\bar{\alpha}$ is a unique solution of (6). So $D_{(\alpha, \lambda)}G(\bar{u}, (\bar{\alpha}, \lambda))$ has to be regular. This means we can apply the implicit function theorem.
to obtain open sets $U \subseteq U' \subseteq \mathbb{R}^n$, $V \subseteq \mathbb{R}^{k+1}$ with $\bar{u} \in U$, $(\bar{\alpha}, \bar{\lambda}) \in V$ and a continuously differentiable function $\phi = (\phi_\alpha, \phi_\lambda) : U \to V$ with
\[
G(u, (\alpha, \lambda)) = 0 \iff (\alpha, \lambda) = \phi(u) \quad \forall u \in U, (\alpha, \lambda) \in V.
\]
In particular,
\[
\varphi|_U(u) = \min_{\alpha \in \Delta_k} \left( \|DJ(u)^\top \alpha\|^2 - (\alpha^\top \epsilon)^2 \right) = \|DJ(u)^\top \phi_\alpha(u)\|^2 - (\phi_\alpha(u)^\top \epsilon)^2,
\]
so $\varphi|_U$ is continuously differentiable.

**Remark A.4.** From the proof of Theorem 2.11 we can even derive an explicit formula for the derivative of $\varphi|_U$ in $\bar{u}$: First of all, the derivative of the implicit function $\phi$ is given by
\[
D\phi(\bar{u}) = -G(\alpha, \lambda)(\bar{u}, (\bar{\alpha}, \bar{\lambda}))^{-1}G_u(\bar{u}, (\bar{\alpha}, \bar{\lambda})) = -\left( DJ(\bar{u})DJ(\bar{u})^\top - \epsilon\epsilon^\top \begin{pmatrix} 1 & \cdots & 1 \end{pmatrix}_{1 \times k} \right)^{-1} \cdot \begin{pmatrix} \bar{\alpha}^\top DJ(\bar{u}) \nabla^2 J_1(\bar{u}) \\ \vdots \\ \bar{\alpha}^\top DJ(\bar{u}) \nabla^2 J_k(\bar{u}) \end{pmatrix} + \left( DJ(\bar{u}) \sum_{i=1}^k \bar{\alpha}_i \nabla^2 J_i(\bar{u}) \right)_{1 \times n}.
\]
By applying the chain rule to (20) we obtain
\[
D\varphi|_U(\bar{u}) = 2\left( DJ(\bar{u})^\top \bar{\alpha} \right)^\top \sum_{i=1}^k \bar{\alpha}_i \nabla^2 J_i(\bar{u}) + \left( 2(DJ(\bar{u})^\top \bar{\alpha})^\top DJ(\bar{u})^\top - 2(\bar{\alpha}^\top \epsilon)^\top \right) D\phi_\alpha(\bar{u})
\]
\[
= 2\left( DJ(\bar{u})^\top \bar{\alpha} \right)^\top \sum_{i=1}^k \bar{\alpha}_i \nabla^2 J_i(\bar{u}) + 2\bar{\alpha}^\top \left( DJ(\bar{u})DJ(\bar{u})^\top - \epsilon\epsilon^\top \right) D\phi_\alpha(\bar{u})
\]
\[
= 2\left( DJ(\bar{u})^\top \bar{\alpha} \right)^\top \sum_{i=1}^k \bar{\alpha}_i \nabla^2 J_i(\bar{u}) + 2(\bar{\lambda}, \ldots, \bar{\lambda}) D\phi_\lambda(\bar{u}).
\]