Mathematical and Computational Applications

Article

A Trust Region Reduced Basis Pascoletti-Serafini Algorithm for Multi-Objective PDE-Constrained Parameter Optimization

Stefan Banholzer, Luca Mechelli and Stefan Volkwein *

Department of Mathematics and Statistics, University of Konstanz, Universitätsstraße 10, 78464 Konstanz, Germany; stefan.banholzer@uni-konstanz.de (S.B.); luca.mechelli@uni-konstanz.de (L.M.)
* Correspondence: stefan.volkwein@uni-konstanz.de

Abstract: In the present paper non-convex multi-objective parameter optimization problems are considered which are governed by elliptic parametrized partial differential equations (PDEs). To solve these problems numerically the Pascoletti-Serafini scalarization is applied and the obtained scalar optimization problems are solved by an augmented Lagrangian method. However, due to the PDE constraints, the numerical solution is very expensive so that a model reduction is utilized by using the reduced basis (RB) method. The quality of the RB approximation is ensured by a trust-region strategy which does not require any offline procedure, in which the RB functions are computed in a greedy algorithm. Moreover, convergence of the proposed method is guaranteed and different techniques to prevent the excessive growth of the number of basis functions are explored. Numerical examples illustrate the efficiency of the proposed solution technique.

Keywords: non-convex multi-objective optimization; partial differential equations; Pascoletti-Serafini method; augmented Lagrangian; reduced basis method; trust-region strategy

Citation: Banholzer, S.; Mechelli, L.; Volkwein, S. A Trust Region Reduced Basis Pascoletti-Serafini Algorithm for Multi-Objective PDE-Constrained Parameter Optimization. *Math. Comput. Appl.* **2022**, *27*, 39. https://doi.org/10.3390/mca27030039

Academic Editors: Simona Perotto, Gianluigi Rozza and Antonia Larese

Received: 18 January 2022
Accepted: 29 April 2022
Published: 3 May 2022

1. Introduction

Multi-objective optimization plays an important role in many applications, e.g., in industry, medicine or engineering. One of the mentioned examples is the minimization of costs with simultaneous quality optimization in production or the minimization of CO_2 emission in energy generation and simultaneous cost minimization. These problems lead to multi-objective optimization problems (MOPs), where we want to achieve an optimal compromise with respect to all given objectives at the same time. Normally, the different objectives are contradictory such that there exists an infinite number of optimal compromises. The set of these compromises is called the *Pareto set*. The goal is to approximate the Pareto set in an efficient way, which turns out to be more expensive than solving a single objective optimization problem.

Since MOPs are of great importance, there exist several algorithms to solve them. Among the most popular methods are scalarization methods, which transform MOPs into scalar problems. For example, in the weighted sum method [1–3], convex combinations of the original objectives are optimized. However, in our case the multi-objective optimization problem

$$\min \hat{J}(u) = \big(\hat{J}_1(u), \ldots, \hat{J}_k(u)\big)^T \quad \text{subject to (s.t.)} \quad u \in \mathcal{U}_{\mathsf{ad}} \tag{\textbf{MOP}}$$

is non-convex with a bounded, non-empty, convex and closed set $\mathcal{U}_{\mathsf{ad}}$. To solve (MOP) a suitable scalarization method in that case is the Pascoletti-Serafini (PS) scalarization [4,5]: For a chosen *reference point* $z \in \mathbb{R}^k$ and a given *target direction* $r \in \mathbb{R}^k$ with $r_i > 0$ for all $i \in \{1, \ldots, k\}$ the Pascoletti-Serafini problem is given by

$$\min t \quad \text{s.t.} \quad (t, u) \in \mathbb{R} \times \mathcal{U}_{\mathsf{ad}} \text{ and } \hat{J}(u) - z \leq t\, r. \tag{$\mathbf{P}^{\mathsf{PS}}_{z,r}$}$$

In the present paper ($\mathbf{P}_{z,r}^{\mathsf{PS}}$) is solved by an augmented Lagrangian approach. However, in our case the evaluation of the objective $\hat{J}$ requires the solution of an elliptic partial differential equation (PDE) for the given parameter u. This implies further that for the computation of the gradients $\nabla\hat{J}_i$, $i = 1, \ldots, k$, adjoint PDEs have to be solved; cf. [6]. Here, surrogate models offer a promising tool to reduce the computational effort significantly [7]. Examples are dimensional reduction techniques such as the Reduced Basis (RB) method [8,9]. In an offline phase, a low-dimensional surrogate model of the PDE is constructed by using, e.g., the greedy algorithm, cf. [8,10,11]. In the online phase, only the RB model is used to solve the PDE, which saves a lot of computing time.

Since the early 2000s the combination of model order reduction with trust-region algorithms in the setting of PDE-constrained optimization is present in the literature, cf. [12,13]. The idea in these methods is to replace the usual quadratic model function in each trust-region step with the reduced-order approximation of the cost function. More recent publications followed and enhanced this approach by using a-posteriori error estimates of the cost function and its gradient, cf. [14,15]. These works were the starting point for the trust-region reduced basis methods developed in [16–18]. Let us mention that [19,20] have proposed similar methods for the combination of reduced-order and trust-region methods based on previous works on trust-region algorithms for PDE-constrained optimization under uncertainty, cf. [21,22]. In contrast to the approach followed by [14–18], these methods do not use rigorous a-posteriori error estimates but rather asymptotic error indicators which still allow for a global convergence result. Here we propose an extension of the method in [16] for solving multi-objective PDE-constrained parameter optimization problems, which is based on a combination of the trust-region reduced basis method presented in [17,18] and the PS method. In particular, we discuss different strategies to handle the increasing number of reduced basis functions, which is crucial in order to guarantee good performances of the algorithm. Notice that our approach is designed for applications, where we have to solve the multi-objective PDE-constrained parameter optimization problem once. For that reason, our trust-region reduced basis method does not rely on any offline computations. These proposed strategies are not only interesting in the field of multi-objective optimization by the PS method, but can also be used in other applications where many PDE-constrained optimization problems must be solved and it is hence crucial to keep the number of reduced basis functions small enough, as, e.g., in model predictive control; cf. [23].

The paper is organized as follows: In Section 2 we introduce a general MOP and explain the PS method, in particular, a hierarchical version of the PS algorithm which turns out to be very efficient in the numerical realization. The concrete PDE-constrained MOP is investigated in Section 3. The trust-region RB method and its combination with the PS method is described in Section 4. Convergence is ensured and the algorithmic realization of the approach is explained. Numerical examples are discussed in detail in Section 5. Finally, we draw some conclusions.

2. Multi-Objective Optimization

Let $(\mathcal{U}, \langle \cdot\,, \cdot \rangle_{\mathcal{U}})$ be a real Hilbert space, $\mathcal{U}_{\mathsf{ad}} \subset \mathcal{U}$ non-empty, convex and closed, $k \geq 2$ arbitrary and $\hat{J}_1, \ldots, \hat{J}_k \colon \mathcal{U}_{\mathsf{ad}} \subset \mathcal{U} \to \mathbb{R}$ be given real-valued functions. In this manuscript, we assume also that $\mathcal{U}_{\mathsf{ad}}$ is bounded. This is an assumption we will require later for the convergence of our method. Note that one can derive similar results of this section if $\mathcal{U}_{\mathsf{ad}}$ is unbounded by introducing additional assumptions; cf. [16]. To shorten the notation, we write $\hat{J} := (\hat{J}_1, \ldots, \hat{J}_k)^T \colon \mathcal{U}_{\mathsf{ad}} \to \mathbb{R}^k$. In the following, we deal with the multi-objective optimization problem

$$\min \hat{J}(u) \quad \text{s.t.} \quad u \in \mathcal{U}_{\mathsf{ad}}. \tag{\textbf{MOP}}$$

Definition 1. (a) *The functions* $\hat{J}_1, \ldots, \hat{J}_k$ *are called* cost *or* objective functions. *Analogously, the vector-valued function* $\hat{J}\colon \mathcal{U}_{\mathsf{ad}} \to \mathbb{R}^k$ *is named the* (multi-objective) cost *or* (multi-objective) objective function.

(b) *The Hilbert space* $\mathcal{U}$ *is named the* admissible space, *the set* $\mathcal{U}_{\mathsf{ad}}$ *is called the* admissible set *and a vector* $u \in \mathcal{U}_{\mathsf{ad}}$ *is called* admissible.

(c) *The space* $\mathbb{R}^k$ *is named the* objective space *and the image set* $\hat{J}(\mathcal{U}_{\mathsf{ad}})$ *is called the* objective set. *A vector* $y = \hat{J}(u) \in \hat{J}(\mathcal{U}_{\mathsf{ad}})$ *is called* objective point.

Definition 2 (Partial ordering on $\mathbb{R}^k$)**.** *On* $\mathbb{R}^k$ *we define the partial ordering* $\leq$ *as*

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

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

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

For convenience, we write

$$x \lneqq y :\iff (x \leq y \;\; \& \;\; x \neq y)$$

for all $x, y \in \mathbb{R}^k$ *and define the two sets* $\mathbb{R}^k_{\leq} := \{y \in \mathbb{R}^k \mid y \leq 0\}$, $\mathbb{R}^k_{\lneqq} := \{y \in \mathbb{R}^k \mid y \lneqq 0\}$. *Analogously, the relations* $\geq$, $>$ *and* $\gneqq$ *as well as the sets* $\mathbb{R}^k_{\geq}$ *and* $\mathbb{R}^k_{\gneqq}$ *are defined.*

Definition 3 (Pareto optimality)**.**

(a) *An admissible vector* $\bar{u} \in \mathcal{U}_{\mathsf{ad}}$ *and its corresponding objective point* $\bar{y} := \hat{J}(\bar{u}) \in \hat{J}(\mathcal{U}_{\mathsf{ad}})$ *are called* (locally) weakly Pareto optimal *if there is no* $\tilde{u} \in \mathcal{U}_{\mathsf{ad}}$ *(in a neighborhood of* $\bar{u}$*) with* $\hat{J}(\tilde{u}) < \hat{J}(\bar{u})$. *The sets*

$$\mathcal{U}_{\mathsf{opt,w}} := \{u \in \mathcal{U}_{\mathsf{ad}} \mid u \textit{ is weakly Pareto optimal}\} \subset \mathcal{U}_{\mathsf{ad}},$$
$$\mathcal{U}_{\mathsf{opt,w,loc}} := \{u \in \mathcal{U}_{\mathsf{ad}} \mid u \textit{ is locally weakly Pareto optimal}\} \subset \mathcal{U}_{\mathsf{ad}}$$

are said to be the weak Pareto set *and the* locally weak Pareto set, *respectively. The sets*

$$\mathcal{J}_{\mathsf{opt,w}} := \hat{J}(\mathcal{U}_{\mathsf{opt,w}}) \subset \mathbb{R}^k, \quad \mathcal{J}_{\mathsf{opt,w,loc}} := \hat{J}(\mathcal{U}_{\mathsf{opt,w,loc}}) \subset \mathbb{R}^k,$$

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

(b) *An admissible vector* $\bar{u} \in \mathcal{U}_{\mathsf{ad}}$ *and its corresponding objective point* $\bar{y} := \hat{J}(\bar{u}) \in \hat{J}(\mathcal{U}_{\mathsf{ad}})$ *are called* (locally) Pareto optimal *if there is no* $\tilde{u} \in \mathcal{U}_{\mathsf{ad}}$ *(in a neighborhood of* $\bar{u}$*) with* $\hat{J}(\tilde{u}) \lneqq \hat{J}(\bar{u})$. *The sets*

$$\mathcal{U}_{\mathsf{opt}} := \{u \in \mathcal{U}_{\mathsf{ad}} \mid u \textit{ is Pareto optimal}\} \subset \mathcal{U}_{\mathsf{ad}},$$
$$\mathcal{U}_{\mathsf{opt,loc}} := \{u \in \mathcal{U}_{\mathsf{ad}} \mid u \textit{ is locally Pareto optimal}\} \subset \mathcal{U}_{\mathsf{ad}}$$

are called the Pareto set *and the* local Pareto set, *respectively. The sets*

$$\mathcal{J}_{\mathsf{opt}} := \hat{J}(\mathcal{U}_{\mathsf{opt}}) \subset \mathbb{R}^k, \quad \mathcal{J}_{\mathsf{opt,loc}} := \hat{J}(\mathcal{U}_{\mathsf{opt,loc}}) \subset \mathbb{R}^k$$

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

If we talk about the different notions of (local) (weak) Pareto optimality in one sentence, we use the notation $\mathcal{U}_{\mathsf{opt,(w),(loc)}}$ to keep the sentence compact. Analogously, $\mathcal{U}_{\mathsf{opt,(w),loc}}$, $\mathcal{U}_{\mathsf{opt,(loc)}}$, $\mathcal{J}_{\mathsf{opt,(w),(loc)}}$ etc. are to be understood. An example with the different concepts of Pareto optimality can be found in [16] (Example 1.2.6).

The next theorem about a sufficient condition for the existence of Pareto optimal points goes back to [24]. It also appears in a similar form in [25,26].

Theorem 1. *Suppose that there is $y \in \hat{J}(\mathfrak{U}_{\mathsf{ad}}) + \mathbb{R}^k_{\geq}$ such that the set $(y - \mathbb{R}^k_{\geq}) \cap (\hat{J}(\mathfrak{U}_{\mathsf{ad}}) + \mathbb{R}^k_{\geq})$ is compact. Then it holds $\mathfrak{J}_{\mathsf{opt}} \neq \emptyset$.*

Proof. This is a slight generalization of [1] (Theorem 2.10) using the argument that adding $\mathbb{R}^k_{\geq}$ to the set $\hat{J}(\mathfrak{U}_{\mathsf{ad}})$ does not change the Pareto front $\mathfrak{J}_{\mathsf{opt}}$. □

Given any $y = \hat{J}(u) \in \hat{J}(\mathfrak{U}_{\mathsf{ad}})$ with $y \notin \mathfrak{J}_{\mathsf{opt}}$, it follows directly from the definition of Pareto optimality that there is $\bar{y} = \hat{J}(\bar{u}) \in \hat{J}(\mathfrak{U}_{\mathsf{ad}})$ with $\bar{y} \lneq y$. However, even if the Pareto front $\mathfrak{J}_{\mathsf{opt}}$ is not empty (e.g., since the assumptions of Theorem 1 are satisfied), it is not clear that there is $\bar{y} \in \mathfrak{J}_{\mathsf{opt}}$ with $\bar{y} \lneq y$. If this property holds for all $y \in \hat{J}(\mathfrak{U}_{\mathsf{ad}}) \setminus \mathfrak{J}_{\mathsf{opt}}$, the set $\mathfrak{J}_{\mathsf{opt}}$ is said to be *externally stable*; cf. [1,26].

Definition 4. *The set $\mathfrak{J}_{\mathsf{opt}}$ is said to be* externally stable *if for every $y \in \hat{J}(\mathfrak{U}_{\mathsf{ad}})$ there is $\bar{y} \in \mathfrak{J}_{\mathsf{opt}}$ with $\bar{y} \leq y$. This is equivalent to $\hat{J}(\mathfrak{U}_{\mathsf{ad}}) \subset \mathfrak{J}_{\mathsf{opt}} + \mathbb{R}^k_{\geq}$.*

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 2. *If for every $y \in \hat{J}(\mathfrak{U}_{\mathsf{ad}}) + \mathbb{R}^k_{\geq}$ the set $(y - \mathbb{R}^k_{\geq}) \cap (\hat{J}(\mathfrak{U}_{\mathsf{ad}}) + \mathbb{R}^k_{\geq})$ is compact, then $\mathfrak{J}_{\mathsf{opt}}$ is externally stable.*

Proof. For a proof of a similar version of this theorem, we refer to [1] (Theorem 2.21). □

Among the methods to solve multi-objective optimization problems, the ones based on scalarization techniques are frequently appearing in the literature. Let us mention here the weighted-sum method [1,3], the Euclidian reference point method [27] and the PS method [4,5]. Since in our case the set $\hat{J}(\mathfrak{U}_{\mathsf{ad}}) + \mathbb{R}^k_{\geq}$ is non-convex, we apply the PS method which is proven to be able to solve a non-convex (**MOP**).

2.1. The PS Method

For a chosen *reference point* $z \in \mathbb{R}^k$ and a given *target direction* $r \in \mathbb{R}^k_{>}$ the PS problem is given by

$$\min t \quad \text{s.t.} \quad (t,u) \in \mathbb{R} \times \mathfrak{U}_{\mathsf{ad}} \text{ and } \hat{J}(u) - z \leq t\, r. \tag{$\mathbf{P}^{\mathsf{PS}}_{z,r}$}$$

Analogously, we can define the PS problem as a scalarization problem. For $z \in \mathbb{R}^k$ and $r \in \mathbb{R}^k_{>}$ we define the scalarization function

$$g_{z,r} \colon \mathbb{R}^k \to \mathbb{R}, \quad x \mapsto g_{z,r}(x) := \max_{1 \leq i \leq k} \frac{1}{r_i}(x_i - z_i),$$

and the *PS scalarized function*

$$\hat{J}^{g_{z,r}}(u) := g_{z,r}(\hat{J}(u)) = \max_{1 \leq i \leq k} \frac{1}{r_i}(\hat{J}_i(u) - z_i) \quad \text{for } u \in \mathfrak{U}_{\mathsf{ad}}.$$

Then the *reformulated PS problem* is given by

$$\min \hat{J}^{g_{z,r}}(u) \quad \text{s.t.} \quad u \in \mathfrak{U}_{\mathsf{ad}}. \tag{$\mathbf{RP}^{\mathsf{PS}}_{z,r}$}$$

The following theorem proved in [16] (Theorem 1.7.3) ensures the equivalence between ($\mathbf{P}^{\mathsf{PS}}_{z,r}$) and ($\mathbf{RP}^{\mathsf{PS}}_{z,r}$).

Theorem 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 ($\mathbf{P}^{\mathsf{PS}}_{z,r}$), then $\bar{u}$ is a global (local) solution of ($\mathbf{RP}^{\mathsf{PS}}_{z,r}$) with minimal function value $\bar{t}$. On the other hand, if $\bar{u}$ is a global (local) solution of ($\mathbf{RP}^{\mathsf{PS}}_{z,r}$), then $(\bar{u}, \bar{t})$ with $\bar{t} := \max_{1 \leq i \leq k}(\hat{J}_i(\bar{u}) - z_i)/r_i$ is a global (local) solution of ($\mathbf{P}^{\mathsf{PS}}_{z,r}$).*

Assumption 1. *The cost functions $\hat{J}_1, \dots, \hat{J}_k$ are weakly lower semi-continuous and bounded from below.*

Theorem 4. *Let* Assumption 1 be satisfied and $z \in \mathbb{R}^k$ as well as $r \in \mathbb{R}^k_>$ be arbitrary. Then ($\mathbf{RP}^{\mathsf{PS}}_{z,r}$) has a global solution $\bar{u} \in \mathfrak{U}_{\mathsf{opt}}$.

Proof. A proof of this statement can be found in [16] (Corollary 1.7.12). □

The previous result also shows that the existing global solution of ($\mathbf{RP}^{\mathsf{PS}}_{z,r}$) belongs to the Pareto set. To guarantee a good reconstruction of the Pareto set by the PS method, one needs that, given a (weakly) Pareto optimal point, it is possible to choose the parameters z and r such that this point solves ($\mathbf{RP}^{\mathsf{PS}}_{z,r}$). This is stated in [16] (Theorem 1.7.13), which we report here for clearness.

Theorem 5. *Let $\bar{u} \in \mathfrak{U}_{\mathsf{opt,w}}$ 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 solution of ($\mathbf{RP}^{\mathsf{PS}}_{z,r}$) for the reference point $z := \hat{J}(\bar{u}) - \bar{t}r$. If even $\bar{u} \in \mathfrak{U}_{\mathsf{opt}}$, any other global solution $\tilde{u}$ of ($\mathbf{RP}^{\mathsf{PS}}_{z,r}$) satisfies $\hat{J}(\tilde{u}) = \hat{J}(\bar{u})$.*

Remark 1. *We refer the reader to* [16] *(Lemma 1.7.15) for the derivation of first-order necessary optimality condition for a global solution of ($\mathbf{P}^{\mathsf{PS}}_{z,r}$).*

Thus, the PS method can compute in principle every (locally) (weak) Pareto optimal point so that many algorithms based on PS method have been proposed. Here we only mention the ones which are related to (but differ from) our proposed technique. Our main idea is to keep the parameter r fixed, while varying the reference point z. This was also proposed in [4], but the method turns out to be, on the one hand, not numerically efficient for $k > 2$ and, on the other hand, not numerically applicable in some cases for $k > 2$. In [28], the authors provide assumptions on the Pareto front to ensure that 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 solution to subproblems. Their idea was then to find these trade-off points first and then compute the rest of the Pareto front. A similar idea but with the use of Centroidal Voronoi Tessellation was presented by [29]. Finally, [30] shows and fixes some problematic behavior associated to the algorithm in [28]. We follow the idea of the mentioned contributions of hierarchically solving subproblems of (**MOP**), but with the focus of finding a set of reference points, by looking at subproblems, for which we can obtain Pareto optimal points. We are then not interested in finding 'boundary' points (i.e., the trade-off limits) of the Pareto front and then filling its 'interior' as in [28–30], but rather to partly generalize this approach. In what follows, we characterize which reference points are necessary and/or sufficient for computing the entire (local) (weak) Pareto front. First, we recall the following well-defined solution mappings of ($\mathbf{RP}^{\mathsf{PS}}_{z,r}$); cf. [16] (Definition 1.7.16).

Definition 5. *We define the set-valued mappings*

$$\begin{aligned}
\mathcal{Q}_{\mathsf{opt,w}} &: \mathbb{R}^k \rightrightarrows \mathfrak{U}_{\mathsf{opt,w}}, && z \mapsto \{u \in \mathfrak{U}_{\mathsf{ad}} \mid u \textit{ is a global solution of } (\mathbf{RP}^{\mathsf{PS}}_{z,r})\},\\
\mathcal{Q}_{\mathsf{opt,w,loc}} &: \mathbb{R}^k \rightrightarrows \mathfrak{U}_{\mathsf{opt,w,loc}}, && z \mapsto \{u \in \mathfrak{U}_{\mathsf{ad}} \mid u \textit{ is a local solution of } (\mathbf{RP}^{\mathsf{PS}}_{z,r})\},\\
\mathcal{Q}_{\mathsf{opt,(loc)}} &: \mathbb{R}^k \rightrightarrows \mathfrak{U}_{\mathsf{opt,(loc)}}, && z \mapsto \mathcal{Q}_{\mathsf{opt,w,(loc)}}(z) \cap \mathfrak{U}_{\mathsf{opt,(loc)}}.
\end{aligned}$$

From Theorem 3, it follows that $\mathcal{Q}_{\mathsf{opt,(w),(loc)}}(\mathbb{R}^k) = \mathfrak{U}_{\mathsf{opt,(w),(loc)}}$, i.e., by solving ($\mathbf{RP}^{\mathsf{PS}}_{z,r}$) for all $z \in \mathbb{R}^k$, we obtain all (locally), (weakly) Pareto optimal points. Furthermore, if Assumption 1 is satisfied, we infer from Theorem 4 that $\mathcal{Q}_{\mathsf{opt,(w),(loc)}}(z) \neq \emptyset$ for all $z \in \mathbb{R}^k$. We also introduce the notion of a (locally) (weakly) Pareto sufficient set for the PS method.

Definition 6. *A set $Z \subset \mathbb{R}^k$ is called* (locally) (weakly) Pareto sufficient *if we have* $\mathcal{Q}_{\mathsf{opt},(\mathsf{w}),(\mathsf{loc})}(Z) = \mathcal{U}_{\mathsf{opt},(\mathsf{w}),(\mathsf{loc})}$.

Hence, a (locally) (weakly) Pareto sufficient set contains the reference points which allow us to compute the entire (local) (weak) Pareto front. Clearly, the set $\mathbb{R}^k$ is (locally) (weakly) Pareto sufficient, but this fact is not computationally useful. The next lemma gives a first condition towards this computational efficiency.

Lemma 1. *Let $Z \subset \mathbb{R}^k$ be arbitrary. Z is (locally) (weakly) Pareto sufficient, if*

$$\forall \bar{u} \in \mathcal{U}_{\mathsf{opt},(\mathsf{w}),(\mathsf{loc})} \colon \exists t \in \mathbb{R} \colon \hat{J}(\bar{u}) - tr \in Z. \tag{1}$$

Proof. Let $Z \subset \mathbb{R}^k$ be such that (1) holds. Let $\bar{u} \in \mathcal{U}_{\mathsf{opt},(\mathsf{w}),(\mathsf{loc})}$ be arbitrary. We need to show that there is a $z \in Z$ with $\bar{u} \in \mathcal{Q}_{\mathsf{opt},(\mathsf{w}),(\mathsf{loc})}(z)$. Indeed, by (1) there is $t \in \mathbb{R}$ with $z := \hat{J}(\bar{u}) - tr \in Z$ and by Theorem 5 we already have $\bar{u} \in \mathcal{Q}_{\mathsf{opt},(\mathsf{w}),(\mathsf{loc})}(z)$. □

To proceed we introduce the concepts of ideal point and shifted ideal point, which will first be used to define a set of shifted coordinate planes D. On this set we can then define a set of reference points $\mathcal{Z}^D_{\mathsf{opt},(\mathsf{w}),(\mathsf{loc})}$ which turns out to be an optimal Pareto sufficient set (The word 'optimal' here means that removing any point from the set will cause the loss of the Pareto sufficient property).

Definition 7.

(a) *We define the* ideal objective point $y^{\mathsf{id}} \in \mathbb{R}^k \cup \{-\infty\}$ *by* $y^{\mathsf{id}}_i := \inf_{u \in \mathcal{U}_{\mathsf{ad}}} \hat{J}_i(u)$ *for all* $i \in \{1, \ldots, k\}$.

(b) *For an arbitrary vector* $\tilde{d} \in \mathbb{R}^k_>$ *define the* shifted ideal point $\tilde{y}^{\mathsf{id}} := y^{\mathsf{id}} - \tilde{d}$. *Let $D_i \subset \mathbb{R}^k$ be given by* $D_i := \{y \in \mathbb{R}^k \mid y \geq \tilde{y}^{\mathsf{id}},\ y_i = \tilde{y}^{\mathsf{id}}_i\}$ *for all* $i \in \{1, \ldots, k\}$. *Then the set $D \subset \mathbb{R}^k$ is defined by* $D := \bigcup_{i=1}^k D_i$.

(c) *We define* $\mathcal{Z}^D_{\mathsf{opt},(\mathsf{w}),(\mathsf{loc})} := \{z \in D \mid \exists \bar{u} \in \mathcal{U}_{\mathsf{opt},(\mathsf{w}),(\mathsf{loc})} \colon \exists t \in \mathbb{R} \colon z = \hat{J}(\bar{u}) - tr\}$.

(d) *For any $y \in \mathbb{R}^k$ we set* $t^D(y) := \min_{i \in \{1,\ldots,k\}} (y_i - \tilde{y}^{\mathsf{id}}_i)/r_i \in \mathbb{R}$.

Remark 2. *It is proved in* [16] *(Lemma 1.7.24) that*

$$\mathcal{Z}^D_{\mathsf{opt},(\mathsf{w}),(\mathsf{loc})} = \left\{\hat{J}(\bar{u}) - t^D(\hat{J}(\bar{u}))\, r \,\middle|\, \bar{u} \in \mathcal{U}_{\mathsf{opt},(\mathsf{w}),(\mathsf{loc})}\right\}.$$

Furthermore, the set $\mathcal{Z}^D_{\mathsf{opt},(\mathsf{w}),(\mathsf{loc})}$ is (locally) (weakly) Pareto sufficient and there is a Lipschitz continuous bijection between $\mathcal{Z}^D_{\mathsf{opt}}$ and the Pareto front $\mathcal{J}_{\mathsf{opt}}$. Unfortunately, there is no bijection between $\mathcal{Z}^D_{\mathsf{opt},(\mathsf{w}),(\mathsf{loc})}$ and $\mathcal{J}_{\mathsf{opt},(\mathsf{w}),(\mathsf{loc})}$, but the set $\mathcal{Z}^D_{\mathsf{opt},(\mathsf{w}),(\mathsf{loc})}$ is still (locally) (weakly) Pareto sufficient. Therefore, it is anyway possible to use it for the computation of the Pareto front.

2.2. Hierarchical PS Method

Due to Definition 7 and Remark 2 the set $\mathcal{Z}^D_{\mathsf{opt},(\mathsf{w}),(\mathsf{loc})}$ can only by computed once the set $\mathcal{U}_{\mathsf{opt},(\mathsf{w}),(\mathsf{loc})}$ is available. Clearly, this characterization of $\mathcal{Z}^D_{\mathsf{opt},(\mathsf{w}),(\mathsf{loc})}$ is not useful for a numerical algorithm since the availability of $\mathcal{U}_{\mathsf{opt},(\mathsf{w}),(\mathsf{loc})}$ means that we have already solved (**MOP**). Fortunately, in [16,31] it is shown that the Pareto set has a hierarchical structure. This means that the (weak) Pareto front and the (weak) Pareto sets of (**MOP**) are contained in the set of all (weak) Pareto fronts and (weak) Pareto sets of all of its subproblems. This particular structure of the Pareto set can be exploited to set up a hierarchical algorithm for obtaining a superset of $\mathcal{Z}^D_{\mathsf{opt},(\mathsf{w}),(\mathsf{loc})}$ without having to compute the entire (local) (weak) Pareto set $\mathcal{U}_{\mathsf{opt},(\mathsf{w}),(\mathsf{loc})}$ first. We start the explanation of the hierarchical algorithm by introducing the notion of a subproblem and related notations.

Definition 8. *For the index set $I \subset \{1, \dots, k\}$ we denote by $\hat{J}^I$ the multi-objective cost function $(\hat{J}_i)_{i \in I} \colon \mathcal{U}_{\mathsf{ad}} \to \mathbb{R}^I$, and call the problem*

$$\min \hat{J}^I(u) \quad \textit{s.t.} \quad u \in \mathcal{U}_{\mathsf{ad}} \tag{$\mathbf{MOP}_I$}$$

a subproblem of (**MOP**). *For $I, K \subset \{1, \dots, k\}$ with $K \subset I$,*

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

(b) *the set $\mathcal{U}^I_{\mathsf{opt,(w),(loc)}} := \{u \in \mathcal{U}_{\mathsf{ad}} \mid u$ is (loc.) (weak.) Pareto optimal for ($\mathbf{MOP}_I$)$\}$ denotes the* (local) (weak) Pareto set *and the set $\mathcal{J}^I_{\mathsf{opt,(w),(loc)}} := \hat{J}^I(\mathcal{U}^I_{\mathsf{opt,(w),(loc)}}) \subset \mathbb{R}^I$ denotes the* (local) (weak) Pareto front of the subproblem ($\mathbf{MOP}_I$).

(c) *the* (local) (weak) nadir objective point for the subproblem ($\mathbf{MOP}_I$) *is defined by*

$$y_i^{\mathsf{nad},I,\mathsf{(w),(loc)}} := \sup\{y_i \mid y \in \mathcal{J}^I_{\mathsf{opt},(w),(loc)}\} \quad \textit{for all } i \in I.$$

Given a subproblem ($\mathbf{MOP}_I$) it is straight-forward to define the PS problem for this setting.

Definition 9. *Let $I \subset \{1, \dots, k\}$ be arbitrary. For a given reference point $z \in \mathbb{R}^I$ and target direction $r \in \mathbb{R}^I_>$, we define the* PS problem for ($\mathbf{MOP}_I$) *by*

$$\min t \quad \textit{s.t.} \quad (t, u) \in \mathbb{R} \times \mathcal{U}_{\mathsf{ad}} \textit{ and } \hat{J}^I(u) - z \le tr^I. \tag{$\mathbf{P}^{\mathsf{PS}}_{I,z,r}$}$$

Again, it is possible to show that ($\mathbf{P}^{\mathsf{PS}}_{I,z,r}$) is equivalent (in the sense of Theorem 3*) to the problem*

$$\min \left(\max_{i \in I} \frac{1}{r_i} (\hat{J}_i(u) - z_i) \right) \quad \textit{s.t.} \quad u \in \mathcal{U}_{\mathsf{ad}}. \tag{$\mathbf{RP}^{\mathsf{PS}}_{I,z,r}$}$$

Let us mention that the statements proved in Section 2.1 can be adapted for the PS method for the subproblems. Similarly, we can also generalize the definition of the shifted coordinate plane D and the (locally) (weakly) Pareto sufficient set of reference points $\mathcal{Z}^D_{\mathsf{opt,(w),(loc)}}$ to this setting.

Definition 10. *Let $I \subset \{1, \dots, k\}$ be arbitrary. Given the vector $\tilde{d} \in \mathbb{R}^k_>$ and the shifted ideal point $\tilde{y}^{\mathsf{id}} \in \mathbb{R}^k$, which were both introduced in Definition 7, let $D^I_i \subset \mathbb{R}^I$ be given by*

$$D^I_i := \{y \in \mathbb{R}^I \,|\, y \ge (\tilde{y}^{\mathsf{id}})^I,\, y_i = \tilde{y}^{\mathsf{id}}_i\} \quad \textit{for } i \in I.$$

Then the set $D^I \subset \mathbb{R}^I$ is defined by $D^I := \bigcup_{i \in I} D_i$. Moreover, for all $K \subset \{1, \dots, k\}$ we define the sets

$$\mathcal{Z}^{D^I,K}_{\mathsf{opt,(w),(loc)}} := \{z \in D^I \,|\, \exists \bar{u} \in \mathcal{U}^K_{\mathsf{opt,(w),(loc)}} \colon \exists t \in \mathbb{R} \colon z = \hat{J}^I(\bar{u}) - tr^I\}.$$

To ease the notation, we write $\mathcal{Z}^{D^I}_{\mathsf{opt,(w),(loc)}} := \mathcal{Z}^{D^I,I}_{\mathsf{opt,(w),(loc)}}$. If $I = \{1, \dots, k\}$ we set $\mathcal{Z}^{D,K}_{\mathsf{opt,(w),(loc)}} := \mathcal{Z}^{D^I,K}_{\mathsf{opt,(w),(loc)}}$ and $\mathcal{Z}^{D}_{\mathsf{opt,(w),(loc)}} := \mathcal{Z}^{D^I,I}_{\mathsf{opt,(w),(loc)}}$. Finally, for any $y \in \mathbb{R}^I$ we set $t^{D^I}(y) := \min_{i \in I} \frac{y_i - \tilde{y}^{\mathsf{id}}_i}{r_i} \in \mathbb{R}$.

Note that also Remark 2 can be rewritten for the subproblems.

The main ingredient of the hierarchical PS method is the result that a superset of $\mathcal{Z}^{D^I}_{\mathsf{opt,(w),(loc)}}$ can be computed by using the sets $\mathcal{U}^K_{\mathsf{opt,(w),(loc)}}$ for all $K \subsetneq I$. In other words, in contrast to Definition 10 only the Pareto optimal solutions to all subproblems–but not the problem itself–are needed to compute the (locally) (weakly) Pareto sufficient set of reference points $\mathcal{Z}^{D^I}_{\mathsf{opt,(w),(loc)}}$ for the subproblem ($\mathbf{MOP}_I$). The very technical details of the analytical derivation and verification of this result are omitted here to ease and shorten the presentation. For a reader interested in the details we refer to [16] (Sections 1.7.4.2–1.7.4.4). Building on this result, the idea of the hierarchical PS method is to iteratively solve subproblems with increasing number of cost functions. During this procedure the required reference points for the current subproblem can be computed by using the Pareto optimal solutions of all of its subproblems as described above.

Before we formulate the hierarchical algorithm, we give the necessary numerical condition in order to compute a numerical approximation of the set $\mathcal{Z}^{D^I}_{\mathsf{opt,(w),(loc)}}$ by using the numerical solution to all subproblems.

To do so, we introduce a grid on D^I as follows.

Definition 11. *Let $I \subset \{1, \dots, k\}$ be arbitrary. For a given grid size $h > 0$ and any $i \in I$, we define*

$$\mathcal{Z}^{h,I}_i := \left\{ z \in D^I_i \,\middle|\, \forall j \in I \setminus \{i\} \colon \left(\exists k \geq 0 \colon z_j = \bar{y}^{\mathsf{id}}_j + \frac{h}{2} + kh \right) \,\&\, \left(z_j \leq y^{nad,I,w}_j - \bar{t}^i r_j \right) \right\}.$$

Furthermore, we set $\mathcal{Z}^{h,I} := \bigcup_{i \in I} \mathcal{Z}^{h,I}_i$. If $I = \{1, \dots, k\}$, we write $\mathcal{Z}^h := \mathcal{Z}^{h,I}$.

The idea is to only choose reference points that lie on the grid $\mathcal{Z}^{h,I}$ and do not satisfy the condition

$$\exists K \subsetneq I \colon \exists (\bar{u}, \bar{t}, \bar{z}) \in \mathcal{UTZ}^{\mathsf{num}}(K) \colon z^K = \bar{z}^K \,\&\, z^{I \setminus K} \geq \hat{J}^{I \setminus K}(\bar{u}) - \bar{t} r^{I \setminus K}, \tag{2}$$

where $\mathcal{UTZ}^{\mathsf{num}}(K)$ is a numerical approximation of $\mathcal{UTZ}(K) = \{(u, \tilde{d}_j, \bar{y}^{\mathsf{id}}_j) \mid u \in \tilde{\mathfrak{U}}_{\mathsf{opt,w}}(I)\}$. An explanation for excluding points based on (2) can be found in [16] (Section 1.7.4.5). Finally, we describe the proposed numerical hierarchical PS method in Algorithm 1.

Remark 3. *In [32], the author introduce three different quality criteria for the numerical implementation of a scalarization method, which we discuss here for the presented hierarchical PS method.*

(a) ***Coverage:*** *Every part of the Pareto set and front has to be represented in the sets $\mathfrak{U}^{\mathsf{num}}_{\mathsf{opt,w}}$ and $\mathfrak{J}^{\mathsf{num}}_{\mathsf{opt,w}}$, respectively. This can be measured by*

$$\mathsf{cov}(\mathfrak{J}_{\mathsf{opt,(w),(loc)}}) := \max_{\bar{y} \in \mathfrak{J}_{\mathsf{opt,(w),(loc)}}} \min_{y \in \mathfrak{J}^{\mathsf{num}}_{\mathsf{opt,(w),(loc)}}} \|\bar{y} - y\|.$$

In the case of Algorithm 1, we have that $\mathsf{cov}(\mathfrak{J}_{\mathsf{opt,(w),(loc)}}) = \mathcal{O}(h)$ (cf. [16] (Remark 1.7.69-(a))).

(b) ***Uniformity:*** *The points on the Pareto set and front should be distributed (almost) equidistantly; cf. [16] (Remark 1.7.69-(b)).*

(c) ***Cardinality:*** *The number of points contained in the numerical approximation should be reasonable. In the case of Algorithm 1 is not possible to estimate a-priori the number of elements computed by the method. It is possible to show a bound which can be computed when the nadir objective point $y^{\mathsf{nad,(w)}}$ is known (cf. [16], Remark 1.7.69-(c)).*

Algorithm 1 Solving (**MOP**) numerically by the hierarchical PS method

```
for $j = 1 : k$ do
    Set $I := \{j\}$;
    Compute $\mathcal{U}^{\mathsf{num}}_{\mathsf{opt,w}}(I) = \{u \mid u \text{ minimizes } \hat{J}_j\}$;
    Choose $\tilde{d}_j$, compute $y^{\mathsf{id}}_j$ and set $\tilde{y}^{\mathsf{id}}_j = y^{\mathsf{id}}_j - \tilde{d}_j$;
    Set $\mathcal{UTZ}^{\mathsf{num}}(I) = \{(u, \tilde{d}_j, \tilde{y}^{\mathsf{id}}_j) \mid u \in \mathcal{U}^{\mathsf{num}}_{\mathsf{opt,w}}(I)\}$;
end for
for $i = 2 : k$ do
    for all $I \subset \{1, \dots, k\}$ with $|I| = i$ do
        Initialize $\mathcal{U}^{\mathsf{num}}_{\mathsf{opt,w}}(I) = \bigcup_{K \subsetneq I} \mathcal{U}^{\mathsf{num}}_{\mathsf{opt,w}}(K)$ and $\mathcal{UTZ}^{\mathsf{num}}(I) = \emptyset$;
        Compute the reference points $Z^{\mathsf{num}}(I) = \{z \in \mathcal{Z}^{h,I} \mid \neg(2)\}$;
        while $Z^{\mathsf{num}}(I) \neq \emptyset$ do
            Choose $z \in Z^{\mathsf{num}}(I)$ and remove $z$ from $Z^{\mathsf{num}}(I)$;
            Solve $(\mathbf{P}^{\mathsf{PS}}_{I,z,r})/(\mathbf{RP}^{\mathsf{PS}}_{I,z,r})$;
            Set $\mathcal{U}^{\mathsf{num}}_{\mathsf{opt,w}}(I) \leftarrow \mathcal{U}^{\mathsf{num}}_{\mathsf{opt,w}}(I) \cup \mathcal{Q}^{I}_{\mathsf{opt,w}}(z)$;
            Set
            $\mathcal{UTZ}^{\mathsf{num}}(I) \leftarrow \mathcal{UTZ}^{\mathsf{num}}(I) \cup \{(\bar{u}, \bar{t}, z) \mid (\bar{u}, \bar{t}) \text{ gl. sol. of } (\mathbf{P}^{\mathsf{PS}}_{I,z,r})\}$;
            Add solutions of PSPs with respect to redundant reference points: Set
            $\mathcal{UTZ}^{\mathsf{num}}(I) \leftarrow \mathcal{UTZ}^{\mathsf{num}}(I) \cup \{(\bar{u}, \bar{t}, \tilde{z}) \mid (\bar{u}, \bar{t}) \text{ gl. sol. of } (\mathbf{P}^{\mathsf{PS}}_{I,z,r}),$
                                    $\tilde{z} \in Z^{\mathsf{num}}(I) \cap [z - (\bar{t}r^I - (\hat{J}^I(\bar{u}) - z)), z]\}$;
            Remove redundant reference points: Set
            $Z^{\mathsf{num}}(I) \leftarrow Z^{\mathsf{num}}(I) \setminus [z - (\bar{t}r^I - (\hat{J}^I(\bar{u}) - z)), z]$ for all $\bar{u} \in \mathcal{Q}^{I}_{\mathsf{opt,(w)}}(z)$;
        end while
    end for
end for
if computeParetoFront == true then
    Remove all $u \in \mathcal{U}^{\mathsf{num}}_{\mathsf{opt,w}}(\{1, \dots, k\})$ with $u \notin \mathcal{U}_{\mathsf{opt}}$ by a non-dominance test;
end if
```

3. The Non-Convex Parametric PDE-Constrained MOP

Before defining our exemplary MOP, we introduce the PDE model which will later serve as an equality constraint. Let $\Omega \subset \mathbb{R}^d$, $d \in \{2, 3\}$, be a bounded domain with Lipschitz-continuous boundary $\Gamma = \partial\Omega$. Furthermore, let $\Omega_1, \dots, \Omega_m$ be a pairwise disjoint decomposition of the domain Ω and set $\Gamma_i := \partial\Omega_i \cap \partial\Omega$ for all $i = 1, \dots, m$. Then we are interested in the following elliptic diffusion-reaction equation with Robin boundary condition:

$$-\nabla \cdot \left(\sum_{i=1}^{m} u_i^{\kappa} \chi_{\Omega_i}(\boldsymbol{x}) \nabla y(\boldsymbol{x}) \right) + u^r\, r(\boldsymbol{x}) y(\boldsymbol{x}) = f(\boldsymbol{x}) \qquad \text{a.e. in } \Omega, \tag{3a}$$

$$u_i^{\kappa} \frac{\partial y}{\partial \boldsymbol{n}}(\boldsymbol{s}) + \alpha y(\boldsymbol{s}) = \alpha y_a(\boldsymbol{s}) \qquad \text{a.e. on } \Gamma_i. \tag{3b}$$

For every $i \in \{1, \dots, m\}$, the parameter $u_i^{\kappa} > 0$ represents the diffusion coefficient on the subdomain Ω_i. By $r \in L^\infty(\Omega)$, we denote a reaction function, which is supposed to satisfy $r > 0$ a.e. in Ω and is controlled by the scalar parameter $u^r > 0$. On the right-hand side of (3a), we have the source term $f \in L^2(\Omega)$. The constant $\alpha > 0$ in (3b) models the heat exchange with the outside of the domain Ω, 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 the vector $u = (u^\kappa, u^r)^T$ with $u^\kappa = (u_1^\kappa, \dots, u_m^\kappa)^T \in \mathbb{R}^m$. Setting $H = L^2(\Omega)$ and $V = H^1(\Omega)$ the weak formulation of (3) is

$$a(u; y, \varphi) = \mathcal{F}(\varphi) \quad \text{for all } \varphi \in V \tag{4}$$

for any $u \in \mathfrak{U}$. In (4) the parameter-dependent symmetric bilinear form $a(u;\cdot,\cdot)\colon V \times V \to \mathbb{R}$ is given by

$$a(u;\varphi,\psi) := \sum_{i=1}^{m} u_i^\kappa \int_{\Omega_i} \nabla\varphi(\boldsymbol{x}) \cdot \nabla\psi(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x} + u^r \int_\Omega r(\boldsymbol{x})\varphi(\boldsymbol{x})\psi(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x} + \alpha \int_\Gamma \varphi(\boldsymbol{s})\psi(\boldsymbol{s})\,\mathrm{d}\boldsymbol{s}$$

for all $\varphi, \psi \in V$ and $u \in \mathfrak{U}$. The linear functional $\mathcal{F} \in V'$ is defined by

$$\mathcal{F}(\varphi) := \int_\Omega f(\boldsymbol{x})\varphi(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x} + \alpha \int_\Gamma y_a(\boldsymbol{s})\varphi(\boldsymbol{s})\,\mathrm{d}\boldsymbol{s} \quad \text{for all } \varphi \in V.$$

Lemma 2. (a) *For all $u \in \mathfrak{U}$ it holds*

$$\|a(u;\cdot,\cdot)\|_{L(V,V')} \le C\|u\|_{\mathfrak{U}}$$

with a constant $C > 0$, which does not depend on u.

(b) *For all $u \in \mathfrak{U}$ with $u^\kappa > 0$ in $\mathbb{R}$ and $u^r > 0$, it holds*

$$a(u;\varphi,\varphi) \ge \min(u_1^\kappa, \ldots, u_m^\kappa, u^r)\|\varphi\|_V^2 \quad \textit{for all } \varphi \in V.$$

(c) *The mapping $\mathcal{F} \in V'$ is well-defined.*

Proof. All statements follow from similar arguments of [33] (Lemma 1.4), where related operators were considered in the parabolic case. □

Theorem 6. *Let $u \in \mathfrak{U}$ with $u > 0$ be arbitrary. Then there is a unique solution $y = y(u) \in V$ of (3). Moreover, the estimate*

$$\|y\|_V \le C\Big(\|f\|_{L^2(\Omega)} + \|y_a\|_{L^2(\Gamma)}\Big) \tag{5}$$

holds with a constant $C > 0$, which depends continuously on u, but is independent of f and y_a.

Proof. The claims follow from the Lax-Milgram theorem (cf. [34]) and Lemma 2. □

Definition 12. *Let $u^\kappa_{min} \in (0,\infty)^m$ and $u^r_{min} > 0$ be arbitrary. Then we define the closed set*

$$\mathfrak{U}_{\mathsf{eq}} := \{u \in \mathfrak{U} \mid u^\kappa \ge u^\kappa_{min},\, u^r \ge u^r_{min}\}.$$

In view of Theorem 6, *it is possible to define the solution operator $\mathcal{S}\colon \mathfrak{U}_{\mathsf{eq}} \to V$, which maps any parameter $u \in \mathfrak{U}_{\mathsf{eq}}$ to the unique solution $y = \mathcal{S}(u) \in V$ of* (4).

Remark 4. *Due to Lemma* 2, we can conclude that $a(u;\varphi,\varphi) \ge \alpha_{\mathsf{min}}\|\varphi\|_V^2$ for all $\varphi \in V$ and $u \in \mathfrak{U}_{\mathsf{eq}}$, where $\alpha_{\mathsf{min}} := \min\big((u^\kappa_{min})_1, \ldots, (u^\kappa_{min})_m, u^r\big) > 0$. In particular, the constant C in (5) can be chosen independently of u if we restrict ourselves to parameters $u \in \mathfrak{U}_{\mathsf{eq}}$.

Theorem 7. *The solution operator $\mathcal{S}\colon \mathfrak{U}_{\mathsf{eq}} \to V$ is twice continuously Fréchet differentiable. For the first derivative $\mathcal{S}'\colon \mathfrak{U}_{\mathsf{eq}} \to L(\mathfrak{U}, V)$, we have that for any $u \in \mathfrak{U}_{\mathsf{eq}}$ and $h \in \mathfrak{U}$ the function $y^h := \mathcal{S}'(u)h \in V$ solves the equation*

$$a(u;y^h,\varphi) = -\partial_u a(u;\mathcal{S}(u),\varphi)h \quad \textit{for all } \varphi \in V.$$

The second derivative $\mathcal{S}'' \colon \mathfrak{U}_{\mathsf{eq}} \to L(\mathfrak{U}, L(\mathfrak{U}, V))$ is given as follows: For any $u \in \mathfrak{U}_{\mathsf{eq}}$ and $h_1, h_2 \in \mathfrak{U}$, the function $y^{h_1,h_2} := \mathcal{S}''(u)(h_1, h_2)$ solves the equation

$$a(u; y^{h_1,h_2}, \varphi) = -\partial_u a(u; \mathcal{S}'(u)h_1, \varphi)h_2 - \partial_u a(u; \mathcal{S}'(u)h_2, \varphi)h_1 \quad \textit{for all } \varphi \in V.$$

Remark 5. *By $\partial_u a$ we denote the partial derivative of the mapping a w.r.t. the parameter u. Since a is linear in u, it holds*

$$\partial_u a(u; \varphi, \psi)h = a(h; \varphi, \psi), \quad \partial_u^2 a(u; \varphi, \psi) = 0 \in L(\mathfrak{U}, \mathfrak{U}')$$

for all $u, h \in \mathfrak{U}$ and all $\varphi, \psi \in V$. In particular, we can identify $\partial_u a(u; \varphi, \psi) \in \mathfrak{U}'$ by

$$\partial_u a(u; \varphi, \psi) = \begin{pmatrix} \int_{\Omega_1} \nabla\varphi(\boldsymbol{x}) \cdot \nabla\psi(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x} \\ \vdots \\ \int_{\Omega_m} \nabla\varphi(\boldsymbol{x}) \cdot \nabla\psi(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x} \\ \int_{\Omega} r(\boldsymbol{x})\varphi(\boldsymbol{x})\psi(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x} \end{pmatrix} \in \mathfrak{U}$$

by using the Riesz representation theorem.

We are now ready to state the multiobjective parametric PDE-constrained optimization problem (MPPOP). Let $k \in \mathbb{N}$ be fixed and

$$\sigma_\Omega^{(1)}, \ldots, \sigma_\Omega^{(k)} \geq 0 \quad \text{as well as} \quad \sigma_\mathfrak{U}^{(1)}, \ldots, \sigma_\mathfrak{U}^{(k)} \geq 0$$

be non-negative weights. Furthermore, denote by $y_\Omega^{(1)}, \ldots, y_\Omega^{(k)} \in H$ the desired states and by $u_d^{(1)}, \ldots, u_d^{(k)} \in \mathfrak{U}$ the desired parameters. Then we define the multiobjective essential cost functions $\hat{J}_1, \ldots, \hat{J}_k \colon \mathfrak{U}_{\mathsf{eq}} \to \mathbb{R}$ by

$$\hat{J}_i(u) := \frac{\sigma_\Omega^{(i)}}{2} \|\mathcal{S}(u) - y_\Omega^{(i)}\|_H^2 + \frac{\sigma_\mathfrak{U}^{(i)}}{2} \|u - u_d^{(i)}\|_\mathfrak{U}^2 \quad \text{for all } u \in \mathfrak{U}_{\mathsf{eq}} \text{ and } i \in \{1, \ldots, k\}.$$

Moreover, u_a, u_b with $u_a \leq u_b$ are lower and upper bounds on the parameter u which we assume to be finite. We define $\mathfrak{U}_{\mathsf{ad}} := \{u \in \mathfrak{U} \mid u_a \leq u \leq u_b\}$ and we assume that $\mathfrak{U}_{\mathsf{ad}} \subset \mathfrak{U}_{\mathsf{eq}}$ holds. Note that $\mathfrak{U}_{\mathsf{ad}}$ is a closed, convex and bounded set because of the finiteness assumption on u_a and u_b. We are interested in solving

$$\min_{u \in \mathfrak{U}_{\mathsf{ad}}} \hat{J}(u) = \min_{u \in \mathfrak{U}_{\mathsf{ad}}} \left(\hat{J}_1(u), \ldots, \hat{J}_k(u)\right)^T. \tag{MPPOP}$$

Note that, thanks to the assumptions on $\mathfrak{U}_{\mathsf{ad}}$ and $\sigma_\mathfrak{U}^{(i)}$, the costs $\hat{J}_1, \ldots, \hat{J}_k$ satisfy Assumption 1. This problem fits into the framework of non-convex multiobjective optimization and Algorithm 1 can be applied. The non-convexity comes from the way the bilinear form depends on the parameter u. This makes, in fact, the solution mapping non-linear and thus the MPPOP non-convex. To close this section, we derive the expression of the gradient and Hessian of the cost functionals $\hat{J}_1, \ldots, \hat{J}_k$. We define the i-th adjoint equation and its solution operator as

Definition 13. *For $i = 1, \ldots, k$, the solution operator of the i-th adjoint equation is $\mathcal{A}_i \colon \mathfrak{U}_{\mathsf{eq}} \to V$, where for any given $u \in \mathfrak{U}_{\mathsf{eq}}$, $p^{(i)} := \mathcal{A}_i(u)$ solves the equation*

$$a(u; \varphi, p^{(i)}) = \langle \sigma_\Omega^{(i)}(\mathcal{S}(u) - y_\Omega^{(i)}), \varphi \rangle_H \quad \textit{for all } \varphi \in V. \tag{6}$$

As shown in [16], this operators satisfy the two following results:

Lemma 3. *The solution operator $\mathcal{A}_i \colon \mathcal{U}_{\mathsf{eq}} \to V$ is continuously Fréchet differentiable for all $i = 1, \ldots, k$. For all $i = 1, ...k$, for the first derivative $\mathcal{A}_i' \colon \mathcal{U}_{\mathsf{eq}} \to L(\mathcal{U}, V)$, we have that for any $u \in \mathcal{U}_{\mathsf{eq}}$ and $h \in \mathcal{U}$ the function $p^{(i),h} := \mathcal{A}_i'(u)h \in V$ solves the equation*

$$a(u; \varphi, p_i^{(i),h}) = -\partial_u a(u; \varphi, \mathcal{A}_i(u))h + \sigma_\Omega \langle \mathcal{S}'(u)h, \varphi \rangle_{V',V} \quad \text{for all } \varphi \in V. \tag{7}$$

Corollary 1. *Let $\mathcal{U}_{\mathsf{ad}} \subset \mathcal{U}_{\mathsf{eq}}$, $u \in \mathcal{U}_{\mathsf{ad}}$ and $h \in \mathcal{U}$ be arbitrary. Then for $i = 1, \ldots, k$ the cost functions $\hat{J}_i$ are twice continuously Fréchet differentiable and it holds*

$$\begin{aligned}
\nabla \hat{J}_i(u) &= -\partial_u a(u; \mathcal{S}(u), \mathcal{A}_i(u)) + \sigma_{\mathcal{U}}(u - u_d^{(i)}) \in \mathcal{U}, \\
\nabla^2 \hat{J}_i(u)h &= -\partial_u a(u; \mathcal{S}'(u)h, \mathcal{A}_i(u)) - \partial_u a(u; \mathcal{S}(u), \mathcal{A}_i'(u)h) + \sigma_{\mathcal{U}}^{(i)} h \in \mathcal{U}.
\end{aligned}$$

where we use the representation of $\partial_u a(u; \mathcal{S}(u), \mathcal{A}_i(u)) \in \mathcal{U}'$ in $\mathcal{U}$, cf. Remark 5*.*

The RB Method for MPPOP

One of the limitations of solving the MPPOP directly with the PS method is the high computational cost. Algorithm 1, in fact, requires to solve the state and adjoint equation a large number of times in order to efficiently approximate the Pareto set. Unfortunately, the numerical evaluation of the state and adjoint solution operators is costly due to the high number of degrees of freedom required to apply, for example, the FE method. For this reason, we use the RB method. In the following we explain how the RB method can be applied to our model. From Theorem 6, we know that the weak form of the state equation admits a unique solution for any control $u \in \mathcal{U}_{\mathsf{eq}}$. This allows us to define the solution operator $\mathcal{S} : \mathcal{U}_{\mathsf{eq}} \to V$. Now, let us consider the so-called solution manifold $\mathcal{M} := \{\mathcal{S}(u) \mid u \in \mathcal{U}_{\mathsf{eq}}\} \subset V$. The goal of the RB method is to provide a low-dimensional subspace $V^\ell \subset V$, which is a good approximation of $\mathcal{M}$. The subspace V^ℓ is defined as the span of linearly independent snapshots $\mathcal{S}(u_1), \ldots, \mathcal{S}(u_\ell)$ for selected parameters $u_1, \ldots, u_\ell \in \mathcal{U}_{\mathsf{eq}}$. Clearly, V^ℓ has dimension ℓ and the snapshots constitute its basis. Let us postpone the discussion on how to select good parameters for generating V^ℓ. Given an RB space V^ℓ, we obtain the reduced-order state equation by a Galerkin projection:

$$a(u; y^\ell, \psi) = \mathcal{F}(\psi) \quad \text{for all } \psi \in V^\ell. \tag{8}$$

Also for the reduced-order equation, we have unique solvability for all parameters $u \in \mathcal{U}_{\mathsf{eq}}$. The solution map $\mathcal{S}^\ell \colon \mathcal{U}_{\mathsf{eq}} \to V^\ell$, which maps any parameter $u \in \mathcal{U}_{\mathsf{eq}}$ to the unique solution $y^\ell = \mathcal{S}^\ell(u) \in V^\ell$ of (8), is then well-defined. We can similarly define a reduced-order adjoint equation and essential cost functional. For $i = 1, \ldots, k$, we define the essential reduced-order cost functions $\hat{J}_i^\ell \colon \mathcal{U}_{\mathsf{eq}} \to \mathbb{R}$ by

$$\hat{J}_i^\ell(u) := \frac{\sigma_\Omega^{(i)}}{2} \|\mathcal{S}^\ell(u) - y_\Omega^{(i)}\|_H^2 + \frac{\sigma_{\mathcal{U}}^{(i)}}{2} \|u - u_d^{(i)}\|_{\mathcal{U}}^2,$$

the reduced-order adjoint equation by

$$a(u; \psi, p^{(i),\ell}) = \langle \sigma_\Omega^{(i)}(\mathcal{S}^\ell(u) - y_\Omega^{(i)}), \psi \rangle_H \quad \text{for all } \psi \in V^\ell \tag{9}$$

and the reduced-order adjoint solution operator $\mathcal{A}_i^\ell : \mathcal{U}_{\mathsf{eq}} \to V$. Following Corollary 1, it is possible to represent the gradient and the Hessian of the essential reduced-order cost functions $\hat{J}_i^\ell$ for $i = 1, \ldots, k$ by simply replacing the operators $\mathcal{S}$ and $\mathcal{A}_i$ by their respective reduced-order versions $\mathcal{S}^\ell$ and $\mathcal{A}_i^\ell$. There are still two aspects which remain to be clarified: first, how to generate an RB space which guarantees a good approximation of the state and adjoint solution manifolds and, second, how to estimate a-posteriori (i.e., without explicitly evaluating the full-order solution operators $\mathcal{S}$ and $\mathcal{A}$) the error of such an approximation.

For the first aspect, one can think of building an RB space either prior to solving the reduced-order optimization problem or while solving it. The first approach is the so-called offline/online decomposition; cf. [35]. This technique exploits a greedy algorithm in the offline phase, which iteratively searches for the parameter for which the approximation error between the full- and reduced-order state and adjoint variables is the largest. Then, the RB space is enriched (by solving the full-order state and adjoint equations at the respective parameter and orthonormalizing the newly computed snapshots with respect to the current RB basis) until a pre-defined tolerance for the approximation error is reached. Once the RB space is computed, the online phase can start: the optimization problem is solved fast on the reduced-order level. Although this technique is still widely used in literature, it shows many disadvantages in the context of optimization. At first, it suffers from the curse of dimensionality: for a high-dimensional parameter space it is too costly to explore the entire parameter space with a greedy procedure. At second, it is counter-intuitive to prepare an RB space which is accurate enough for any parameter, when usually the optimization method follows a (short) pattern in the parameter space to find the solution or when the Pareto set is contained in some local regions of the parameter space, as often in the case of non-convex multiobjective problems. While it is true that the computational costs of an offline phase could be amortized in the context of multiobjective optimization for a reasonably small dimension of the parameter space due to the vast amount a scalarized PS problems that need to be solved in the online phase, the disadvantage of the offline-online splitting in this setting is the lack of control of the accuracy of the Pareto optimal solutions. Indeed, to the best of our knowledge there are no suitable error indicators for the greedy algorithm to guarantee a certified accuracy of the reduced-order Pareto optimal points w.r.t. full-order ones. Luckily, the focus has shifted recently towards adapting the RB space while proceeding with the optimization method. This procedure is followed, e.g., by the methods presented in [14,15,17,18]. The advantage of these methods with respect to methods based on an offline-online splitting is that they compute first-order critical points of the full-order optimization problems. Let us specify that in [14,17,18] the authors proposed and progressively improved an RB method combined with a TR algorithm, based on more general results presented in [15]. Such a method constructs the RB space adaptively while the optimizer is computing the optimal solution. Our focus here is on further improving the method in [17], which can be considered the most general among the TR-RB methods.

For any of the above-mentioned methods, a-posteriori error estimates are crucial to compute upper bounds of the approximation error made by the RB space in reconstructing the solution for a given parameter without any full-order solution at hand. In case of optimization, one is also interested in estimating the error in reconstructing the cost functional and its gradient. For our model, we can use the following estimates:

Theorem 8. *Let $u \in \mathcal{U}_{\mathsf{ad}}$ be arbitrary and denote by $\alpha(u)$ the coercivity constant of the bilinear form $a(u;\cdot,\cdot)$. By* Remark 4, *it holds $\alpha(u) \geq \alpha_{\mathsf{min}} > 0$. Let the residual $r_{\mathsf{st}}(u;\cdot) \in V'$ be given by $r_{\mathsf{st}}(u;\varphi) := \mathcal{F}(\varphi) - a(u;\mathcal{S}^\ell(u),\varphi)$ for all $\varphi \in V$. Then it holds*

$$\big\|\mathcal{S}(u) - \mathcal{S}^\ell(u)\big\|_V \leq \Delta_{\mathsf{st}}(u) := \frac{\big\|r_{\mathsf{st}}(u;\cdot)\big\|_{V'}}{\alpha(u)}. \tag{10}$$

For $i = 1, \dots, k$ the residual $r^{(i)}_{\mathsf{adj}}(u;\cdot) \in V'$ of the adjoint equations is given by $r^{(i)}_{\mathsf{adj}}(u;\varphi) := \langle \sigma^{(i)}_\Omega(\mathcal{S}^\ell(u) - y^{(i)}_\Omega), \varphi\rangle_H - a(u;\varphi,\mathcal{A}^\ell_i(u))$ for all $\varphi \in V$. Then it holds

$$\big\|\mathcal{A}_i(u) - \mathcal{A}^\ell_i(u)\big\|_V \leq \Delta^{(i)}_{\mathsf{adj}}(u) := \frac{\big\|r^{(i)}_{\mathsf{adj}}(u;\cdot)\big\|_{V'} + \sigma^{(i)}_\Omega \Delta_{\mathsf{st}}(u)}{\alpha(u)}.$$

Furthermore, for $i = 1, \dots, k$ we have

$$\begin{aligned} |\hat{J}_i(u) - \hat{J}_i^\ell(u)| &\leq \Delta_{\mathsf{st}}(u) \|r_{\mathsf{adj}}^{(i)}(u;\cdot)\|_{V'} + \sigma_\Omega^{(i)} \Delta_{\mathsf{st}}(u)^2 =: \Delta_{\hat{J}_i^\ell}(u), \\ \|\nabla \hat{J}_i(u) - \nabla \hat{J}_i^\ell(u)\|_{\mathfrak{U}} &\leq \|\partial_u a(u;\cdot,\cdot)\| \Big(\|\mathcal{S}^\ell(u)\|_V \Delta_{\mathsf{adj}}^{(i)}(u) + \Delta_{\mathsf{st}}(u) \Delta_{\mathsf{adj}}^{(i)}(u) \\ &\qquad + \Delta_{\mathsf{st}}(u) \|\mathcal{A}_i^\ell(u)\|_V \Big) =: \Delta_{\nabla \hat{J}_i^\ell}(u). \end{aligned}$$

Proof. A proof of the a-posteriori error estimates for the state and adjoint can be found in [35]. For the cost function and the gradient, we refer to [18] (Proposition 2.5). □

Note that we only need the reduced-order state and adjoint state to evaluate the a-posteriori error estimates. For our example, the computation of the coercivity constant $\alpha(u)$ is cheap, see Lemma 2. In more general examples, this might not be the case. Thus, one often uses a quickly computable lower bound $\alpha_{\mathsf{LB}}(u)$ instead. Possible methods for computing such a lower bound are, e.g., the min-theta approach (cf. [35]) or the Successive Constraint Method (SCM) (cf. [36]). In situations in which the computation or the estimation of the coercivity constant is complicated, the TR-RB algorithms presented in [19,20] have the advantage that they do not require the computation or estimation of the coercivity constant but only rely on asymptotic error estimates consisting of residual based error indicators. Note finally that the computation of the terms $\|r_{\mathsf{st}}(u;\cdot)\|_{V'}$ and $\|r_{\mathsf{adj}}^{(i)}(u;\cdot)\|_{V'}$ is not possible in an infinite-dimensional setting. Even after discretization with the FE method, the cost of computing such a term depends on the dimension of the full-order model, which contradicts the request of having a computationally cheap estimate. However, in our case, the parameter-separability of the bilinear form $a(u;\cdot,\cdot)$ can be exploited to preassemble certain quantities in such a way that the computational cost for evaluating $\|r_{\mathsf{st}}(u;\cdot)\|_{V'}$ and $\|r_{\mathsf{adj}}^{(i)}(u;\cdot)\|_{V'}$ only depends on the dimension of the RB space; see, e.g., [36]. Finally, we apply the RB method to (**MPPOP**): for a given RB space V^ℓ the reduced-order MPPOP reads

$$\min \hat{J}^\ell(u) = \big(\hat{J}_1^\ell(u), \dots, \hat{J}_k^\ell(u)\big)^T \quad \text{s.t.} \quad u \in \mathfrak{U}_{\mathsf{ad}}. \tag{$\mathbf{MPPOP}^\ell$}$$

For an arbitrary reference point $z \in \mathbb{R}^k$ and target direction $r \in \mathbb{R}^k$, the reduced-order PS problem reads

$$\min_{(u,t)} t \quad \text{s.t.} \quad (t,u) \in \mathbb{R} \times \mathfrak{U}_{\mathsf{ad}} \text{ and } \hat{J}_i^\ell(u) - z_i \leq t, \quad i = 1, \dots, k. \tag{$\mathbf{P}_{z,r}^{\mathsf{PS},\ell}$}$$

One could then outline an algorithm similar to Algorithm 1 by using an offline/online splitting. Because of the above-mentioned disadvantages, we focus on combining the PSPs with the TR-RB method from [17] and extend it with respect to the method in [16]. The TR method introduces new aspects to the RB implementation, such as the adaptive construction of the RB space; see the next section for further details.

4. The TR-RB Method

We briefly introduce the method from [17] and clarify how to apply this in combination with the PS method. In Section 4.2 we highlight our extension to this method and how this can reduce the computational time. The basic idea of a TR method is to compute a first-order critical point of a costly optimization problem by iteratively solving some cheap-to-solve approximations in local regions of the admissible space, where these model approximations can be trusted (i.e., are accurate enough). In such a way, one can derive a global method, which converges in a finite number of steps. For each outer iteration $j \geq 0$ of the TR method, the cheap approximation of the objective is generally indicated by $m^{(j)}$ and the trust regions are described by a radius $\delta^{(j)}$. To simplify the exposition, let us stick with the case $\mathcal{U} = \mathbb{R}^m \times \mathbb{R}$, as in Section 3. The TR method solves then, for each $j \geq 0$, the following constrained optimization sub-problems

$$\min_{v \in \mathcal{U}} m^{(j)}(v) \quad \text{s.t.} \quad \|v\|_2 \leq \delta^{(j)}, \tilde{u} := u^{(j)} + v \in \mathcal{U}_{\text{ad}}. \tag{11}$$

Under suitable assumptions, problem (11) admits a unique solution $\bar{v}^{(j)}$, which is used to compute the next outer iteration $u^{(j+1)} = u^{(j)} + \bar{v}^{(j)}$. To further simplify the presentation of the algorithm in [17], let us present it for a general cost functional $\mathcal{J}$. Later in this section we will give more details about its application to the MPPOP and the PS method. The TR-RB version of problem (11) is

$$\min_{\tilde{u} \in \mathcal{U}_{\text{ad}}} \mathcal{J}^{\ell,(j)}(\tilde{u}) \quad \text{s.t.} \quad q^{(j)}(\tilde{u}) := \frac{\Delta_{\mathcal{J}^{\ell,(j)}}(\tilde{u})}{\mathcal{J}^{\ell,(j)}(\tilde{u})} \leq \delta^{(j)}, \tag{12}$$

where $\mathcal{J}^{\ell,(j)}(\tilde{u})$ is the reduced-order cost functional w.r.t. the reduced-order model at the j-th iteration and $\Delta_{\mathcal{J}^{\ell,(j)}}(\tilde{u})$ is an estimate for the error $|\mathcal{J}(\tilde{u}) - \mathcal{J}^{\ell,(j)}(\tilde{u})|$. Looking at (12), one clearly sees that the role of the model function $m^{(j)}$ is played by the reduced-order model cost functional. This is perfectly in line with the TR spirit of having a cheap-to-solve approximation of the original optimization problem. The trust regions are defined instead through the RB error estimator, which is in fact the way we use to check the quality of the approximation. Let us mention at this point that there are also different approaches to this. In [19,20] the authors incorporated the usual trust-region constraints as seen in (11) into a TR-RB algorithm. In [18] also the importance of introducing a correction term on the RB level is discussed to improve the performance of the method. We point out that this only has to be done if one chooses two separate RB spaces for state and adjoint equations (see also [17]). This will not be the case for our application. In Algorithm 2, we report the method from [17]. In what follows, we guide the reader through the features of the algorithm. At first, we need to initialize the reduced-order model at the initial guess $u^{(0)}$. This means computing $\mathcal{S}(u^{(0)})$ and $\mathcal{A}_i(u^{(0)})$ for $i = 1, \ldots, k$ and generating the RB space $V^{\ell,(0)}$ as their span. Similarly, updating the RB space $V^{\ell,(j)}$ at the point $u^{(j+1)}$ means computing the full-order quantities $\mathcal{S}(u^{(j+1)})$ and $\mathcal{A}_i(u^{(j+1)})$ for $i = 1, \ldots, k$ and adding them to the RB space by a Gram-Schmidt orthonormalization.

In Line 3 of Algorithm 2, it is required to compute the so-called approximated generalized Cauchy (AGC) point. We report here its definition according to [15,18].

Definition 14. *Let $\kappa \in (0,1)$ and $\kappa_{\text{arm}} \in (0,1)$ be backtracking parameters. For the current iterate $u^{(j)}$ define $d^{(j)} := \nabla \mathcal{J}^{\ell,(j)}(u^{(j)})$. Let $\alpha \in \mathbb{N}$ be the smallest number for which the two conditions*

$$\mathcal{J}^{\ell,(j)}(P_{\mathcal{U}_{\text{ad}}}(u^{(j)} - \kappa^\alpha d^{(j)})) - \mathcal{J}^{\ell,(j)}(u^{(j)}) \leq -\frac{\kappa_{\text{arm}}}{\kappa^\alpha} \|P_{\mathcal{U}_{\text{ad}}}(u^{(j)} - \kappa^\alpha d^{(j)}) - u^{(j)}\|^2_{\mathcal{U}}, \tag{13}$$

$$q^{(j)}(P_{\mathcal{U}_{\text{ad}}}(u^{(j)} - \kappa^\alpha d^{(j)})) \leq \delta^{(j)} \tag{14}$$

are satisfied, where $P_{\mathcal{U}_{\text{ad}}} : \mathcal{U} \to \mathcal{U}_{\text{ad}}$ is the canonical projection onto the closed and convex set $\mathcal{U}_{\text{ad}}$. Then we define the AGC point *as $u^{(j)}_{\text{AGC}} := P_{\mathcal{U}_{\text{ad}}}(u^{(j)} - \kappa^\alpha d^{(j)})$.*

The TR-RB subproblem (12) is then solved in Line 4 using a projected Newton-CG algorithm with the AGC point as a warm start and the following termination criteria

$$\|u - P_{\mathcal{U}_{\text{ad}}}(u - \nabla \mathcal{J}^{\ell,(j)}(u))\|_{\mathcal{U}} \leq \tau_{\text{sub}}, \quad \beta_{\text{bound}} \delta^{(j)} \leq q^{(j)}(u) \leq \delta^{(j)}. \tag{15}$$

The first condition in (15) is the standard first-order criticality condition with tolerance $\tau_{\text{sub}} \in (0,1)$ and the second one was already introduced in [14] to avoid too many iterations close to the TR boundary, which is generally an area where we are already starting to trust the model function less. The parameter β_{bound} is usually chosen to be close to one exactly for this purpose.

Algorithm 2 TR-RB algorithm

```
1:  Initialize the reduced-order model at $u^{(0)}$, set $j = 0$ and Loop_flag=True;
2:  while Loop_flag do
3:    Compute the AGC point $u^{(j)}_{\text{AGC}}$ ;
4:    Compute $u^{(j+1)}$ as solution of (12) with stopping criteria (15);
5:    if $\mathcal{J}^{\ell,(j)}(u^{(j+1)}) + \Delta_{\mathcal{J}^{\ell,(j)}}(u^{(j+1)}) < \mathcal{J}^{\ell,(j)}(u^{(j)}_{\text{AGC}})$ then
6:      Accept $u^{(j+1)}$, set $\delta^{(j+1)} = \delta^{(j)}$, compute $\varrho^{(j)}$ and $g(u^{(j+1)})$;
7:      if $g(u^{(j+1)}) \leq \tau_{\text{FOC}}$ then
8:        Set Loop_flag=False;
9:      else
10:       if $\varrho^{(j)} \geq \eta_\varrho$ then
11:         Enlarge the TR radius $\delta^{(j+1)} = \beta_1^{-1}\delta^{(j)}$;
12:       end if
13:       if not Skip_enrichment_flag$(j)$ then
14:         Update the RB model at $u^{(j+1)}$ ;
15:       end if
16:     end if
17:   else if $\mathcal{J}^{\ell,(j)}(u^{(j+1)}) - \Delta_{\mathcal{J}^{\ell,(j)}}(u^{(j+1)}) > \mathcal{J}^{\ell,(j)}(u^{(j)}_{\text{AGC}})$ then
18:     if $\beta_1\delta^{(j)} \leq \delta_{\min}$ or Skip_enrichment_flag$(j-1)$ then
19:       Update the RB model at $u^{(j+1)}$;
20:     end if
21:     Reject $u^{(j+1)}$, shrink the radius $\delta^{(j+1)} = \beta_1\delta^{(j)}$ and go to 4;
22:   else
23:     Compute $\mathcal{J}(u^{(j+1)})$, $g(u^{(j+1)})$, $\varrho^{(j)}$ and set $\delta^{(j+1)} = \beta_1^{-1}\delta^{(j)}$;
24:     if $g(u^{(j+1)}) \leq \tau_{\text{FOC}}$ then
25:       Set Loop_flag=False;
26:     else
27:       if Skip_enrichment_flag$(j)$ and $\varrho^{(j)} \geq \eta_\varrho$ then
28:         Accept $u^{(j+1)}$;
29:       else if $\mathcal{J}(u^{(j+1)}) \leq \mathcal{J}^{\ell,(j)}(u^{(j)}_{\text{AGC}})$ then
30:         Accept $u^{(j+1)}$ and update the RB model ;
31:         if $\varrho^{(j)} < \eta_\varrho$ then
32:           Set $\delta^{(j+1)} = \delta^{(j)}$;
33:         end if
34:       else
35:         if $\beta_1\delta^{(j)} \leq \delta_{\min}$ or Skip_enrichment_flag$(j-1)$ then
36:           Update the RB model at $u^{(j+1)}$;
37:         end if
38:         Reject $u^{(j+1)}$, set $\delta^{(j+1)} = \beta_1\delta^{(j)}$ and go to 4;
39:       end if
40:     end if
41:   end if
42:   Set $j = j + 1$;
43: end while
```

An important aspect of TR methods is the decision to accept or reject the step $u^{(j+1)}$. Generally, one asks for the so-called sufficient decrease condition $\mathcal{J}^{\ell,(j+1)}(u^{(j+1)}) \leq \mathcal{J}^{\ell,(j)}(u^{(j)}_{\text{AGC}})$; cf. [15]. Note that this condition requires to update the RB space before being sure that the step will be accepted. If it is rejected, then we performed a costly update without the possibility of exploiting it. Because of this fact, Ref. [14] proposed a sufficient

(Line 5) and a necessary (Line 17) condition for the sufficient decrease condition. In [18] it is also noted that the full-order quantities $\mathcal{J}(u^{(j+1)})$ and $\nabla\mathcal{J}(u^{(j+1)})$ are cheaply available after updating the RB space. Additionally, Ref. [17] introduced the possibility of skipping a redundant enrichment, which is particularly useful at the late stage of the method, where we are close to the optimum. This will prevent the dimension of the RB space from growing too fast, so that the cheap-to-solve property is preserved. The three conditions to be checked in order to decide whether to skip the update of the RB space are contained in the following skipping parameter

$$\begin{aligned}\mathsf{Skip\_enrichment\_flag}(j) := \left(q^{(j)}(u^{(j+1)}) \le \beta_q \delta^{(j+1)}\right) \quad \mathsf{and} \\ \left(\frac{|g(u^{(j+1)}) - g^{\ell,(j)}(u^{(j+1)})|}{g^{\ell,(j)}(u^{(j+1)})} \le \tau_g\right) \quad \mathsf{and} \\ \left(\frac{\|\nabla\mathcal{J}^{\ell,(j)}(u^{(j+1)}) - \nabla\mathcal{J}(u^{(j+1)})\|_{\mathcal{U}}}{\|\nabla\mathcal{J}^{\ell,(j)}(u^{(j+1)})\|_{\mathcal{U}}} \le \min\{\tau_{\mathsf{grad}}, \beta_{\mathsf{grad}}\delta^{(j+1)}\}\right).\end{aligned}$$

where $\beta_q, \beta_{\mathsf{grad}}, \tau_g, \tau_{\mathsf{grad}} \in (0,1)$ are given parameters and

$$g(u) := \|u - P_{\mathcal{U}_{\mathsf{ad}}}(u - \nabla\mathcal{J}(u))\|_{\mathcal{U}}, \quad g^{\ell,(j)}(u) := \|u - P_{\mathcal{U}_{\mathsf{ad}}}(u - \nabla\mathcal{J}^{\ell,(j)}(u))\|_{\mathcal{U}}.$$

Note also that $g(u) = 0$ is nothing else than the standard first-order condition for optimization problems with constraints on the parameter set. This is the reason why Algorithm 2 terminates when $g(u^{(j+1)}) < \tau_{\mathsf{FOC}}$ holds with $\tau_{\mathsf{FOC}} \in (0,1)$. For more details on the skipping condition, we refer to [17]. Typically, TR methods also have the option of shrinking (enlarging) the TR radius $\delta^{(j)}$ with some factor $\beta_1 \in (0,1)$ ($\beta_1^{-1} > 1$, respectively). In the case of Algorithm 2, we shrink the radius if a point is rejected. We also compute the ratio

$$\varrho^{(j)} := \frac{\mathcal{J}(u^{(j)}) - \mathcal{J}(u^{(j+1)})}{\mathcal{J}^{\ell,(j)}(u^{(j)}) - \mathcal{J}^{\ell,(j)}(u^{(j+1)})}.$$

If this ratio is greater than a parameter $\eta_\varrho \in [0.75, 1]$, then the radius is enlarged. Algorithm 2 is proved to be convergent given some technical assumptions on the problem. We summarize everything in the following theorem (cf. [17]).

Theorem 9. *Suppose that $\mathcal{U}_{\mathsf{ad}} = [u^a, u^b] \subset \mathbb{R}^P$ for some $u^a, u^b \in \mathbb{R}^P$ with $u^a \le u^b$. Assume that $\mathcal{J}$ and $\mathcal{J}^{\ell,(j)}$ ($j \in \mathbb{N}$) are strictly positive, $\mathcal{J}$ is continuously Fréchet differentiable and $\mathcal{J}^{\ell,(j)}$ is even twice continuously Fréchet differentiable for all $j \in \mathbb{N}$. Moreover, $\nabla\mathcal{J}^{\ell,(j)}$ is uniformly Lipschitz-continuous with respect to j. Suppose that there is $\delta_{\mathsf{min}} > 0$ such that for every $j \in \mathbb{N}$ there exists a TR radius $\delta^{(j)} \ge \delta_{\mathsf{min}}$, for which there is a solution $u^{(j+1)}$ of the TR-RB subproblem (12) which is accepted by* Algorithm 2*. Assume that the family of functions $(q^{(j)})_{j\in\mathbb{N}}$ is uniformly continuous w.r.t. the parameter u and the index j. Then every accumulation point $\bar{u}$ of the sequence of iterates $(u^{(j)})_{j\in\mathbb{N}}$ is a first-order critical point for the full-order optimization problem, i.e., it holds*

$$\|\bar{u} - P_{\mathcal{U}_{\mathsf{ad}}}(\bar{u} - \nabla\mathcal{J}(\bar{u}))\|_{\mathcal{U}} = 0.$$

In particular, Algorithm 2 *terminates after finitely many steps.*

Although many of the assumptions in Theorem 9 are quite technical for the proof, one can show that they are reasonable in the case of the RB method; cf. [17].

4.1. The TR-RB Algorithm Applied to the PS Method

In this section we show how Algorithm 2 can be applied to the PS method. To this end, we recall the following lemma from [16].

Lemma 4. *There are constants $C_J, C_{\nabla J}, C_{\nabla^2 J} > 0$ such that for any $j \in \{1, \dots, k\}$, any $u \in \mathcal{U}_{\mathsf{ad}}$ and any choice of the RB space V^ℓ it holds*

$$\big|\hat{J}_i^\ell(u)\big| \le C_J, \quad \big\|\nabla \hat{J}_i^\ell(u)\big\|_{\mathcal{U}} \le C_{\nabla J}, \quad \big\|\nabla^2 \hat{J}_i^\ell(u)\big\|_{L(U)} \le C_{\nabla^2 J}.$$

Lemma 4 immediately implies that the reduced-order gradient is uniformly Lipschitz-continuous with respect to ℓ. We have to solve ($\mathbf{P}_{z,r}^{\mathsf{PS}}$). We follow the approach in [16], where the target direction $r = (1, \dots, 1)$ is chosen and an augmented Lagrangian method is used. Provided a penalty parameter $\mu > 0$, the augmented Lagrangian for ($\mathbf{P}_{z,r}^{\mathsf{PS}}$) is

$$\mathcal{L}_A((u,t,s), \lambda; \mu) := t + \sum_{i=1}^{k} \lambda_i c_i(u,t,s) + \frac{\mu}{2} \sum_{i=1}^{k} c_i(u,t,s)^2 \tag{16}$$

with $c_i(u,t,s) = \hat{J}_i(u) - z_i - t + s_i$. The idea is to iteratively solve the subproblems

$$\min \mathcal{L}_A((u,t,s), \lambda; \mu) \quad \text{s.t.} \quad (u,t,s) \in \mathcal{U}_{\mathsf{ad}} \times \mathbb{R} \times \mathbb{R}^k_{\ge} \tag{17}$$

approximately and then update the Lagrange multiplier λ and the penalty parameter μ until the termination criteria

$$\|c(u,t,s)\|_{\mathbb{R}^k} < \tau_{\mathsf{EC}}, \tag{18}$$

$$\big\|(u,t,s) - P_{\mathsf{ad}}\big((u,t,s) - \nabla_{(u,t,s)} \mathcal{L}_A((u,t,s), \lambda; \mu)\big)\big\|_{\mathcal{U} \times \mathbb{R} \times \mathbb{R}^k} < \tau_{\mathsf{FOC}} \tag{19}$$

are satisfied for some tolerances $\tau_{\mathsf{EC}}, \tau_{\mathsf{FOC}} \in (0,1)$, where $P_{\mathsf{ad}} \colon \mathcal{U} \times \mathbb{R} \times \mathbb{R}^k \to \mathcal{U}_{\mathsf{ad}} \times \mathbb{R} \times \mathbb{R}^k_{\ge}$ is the canonical projection onto $\mathcal{U}_{\mathsf{ad}} \times \mathbb{R} \times \mathbb{R}^k_{\ge}$. For further details, we refer to [16] (Appendix B). We want to combine then the augmented Lagrangian method with the TR-RB algorithm to solve problem ($\mathbf{P}_{z,r}^{\mathsf{PS}}$). To do so, we apply Algorithm 2 to solve each subproblem (17). We first define the reduced-order augmented Lagrangian

$$\mathcal{L}_A^\ell((u,t,s), \lambda; \mu) := t + \sum_{i=1}^{k} \lambda_i c_i^\ell(u,t,s) + \frac{\mu}{2} \sum_{i=1}^{k} c_i^\ell(u,t,s)^2, \tag{20}$$

with $c_i^\ell(u,t,s) = \hat{J}_i^\ell(u) - z_i - t + s_i$, which leads to the reduced-order subproblem

$$\min \mathcal{L}_A^\ell((u,t,s), \lambda; \mu) \quad \text{s.t.} \quad (u,t,s) \in \mathcal{U}_{\mathsf{ad}} \times \mathbb{R} \times \mathbb{R}^k_{\ge}. \tag{21}$$

Note that in this case the admissible set $\mathcal{U}_{\mathsf{ad}} \times \mathbb{R} \times \mathbb{R}^k_{\ge}$ is unbounded, which collides with the first assumption of Theorem 9. Nevertheless, Ref. [16] showed that the ($\mathbf{P}_{z,r}^{\mathsf{PS}}$) problem is also equivalent to

$$\min t \quad \text{s.t.} \quad (t,u) \in [t^{\mathsf{min}}, t^{\mathsf{max}}] \times \mathcal{U}_{\mathsf{ad}} \text{ and } \hat{J}(u) - z \le t. \tag{22}$$

There is still the problem that the admissible set for the slack variables s is given by $[0, \infty)^k$. However, computing the partial derivative of the augmented Lagrangian $\mathcal{L}_A$ with respect to s_i, we obtain

$$\partial_{s_i} \mathcal{L}_A((u,t,s), \lambda; \mu) = \lambda_i + \mu\big(\hat{J}_i(u) - z_i - t + s_i\big) \ge \lambda_i + \mu(-z_i - t^{\mathsf{max}} + s_i).$$

Thus, $\mathcal{L}_A$ is strictly monotonically increasing in s_i for $s_i > -\lambda_i/\mu + z_i + t_{\mathsf{max}} =: s_i^{\mathsf{max}}$. Thus, given the Lagrange multiplier λ and the penalty parameter μ, we can restrict the slack variable s_i to the interval $[0, s_i^{\mathsf{max}}]$. This will not cause any modification to the solvability and the solution of the augmented Lagrangian subproblem. By setting $\mathcal{X}_{\mathsf{ad}} := \mathcal{U}_{\mathsf{ad}} \times [t^{\mathsf{min}}, t^{\mathsf{max}}] \times [0, s^{\mathsf{max}}]$, the equivalent formulation for the augmented Lagrangian subproblem corresponding to (22) reads

$$\min_{(u,t,s)\in\mathcal{X}_{\mathsf{ad}}} \mathcal{L}_A((u,t,s),\lambda;\mu). \tag{23}$$

Similarly, the reduced-order augmented Lagrangian subproblem is given by

$$\min \mathcal{L}^\ell_A((u,t,s),\lambda;\mu) \quad \text{s.t.} \quad (u,t,s)\in\mathcal{X}_{\mathsf{ad}}. \tag{24}$$

Therefore, the goal is to apply Algorithm 2 to solve the subproblem (23). To this end, we define $x = (u,t,s) \in \mathcal{U}\times\mathbb{R}\times\mathbb{R}^k$, $\mathcal{J}(x) = \mathcal{L}_A(x,\lambda;\mu)$ and $\mathcal{J}^{\ell,(j)}(x) = \mathcal{L}^{\ell,(j)}_A(x,\lambda;\mu)$ for any reference point $z\in\mathbb{R}^k$, any Lagrange multiplier $\lambda\in\mathbb{R}^k_{\geq}$ and any penalty parameter $\mu > 0$. Furthermore, using the a-posteriori estimates of the individual objectives (cf. Theorem 8), we have that

$$\begin{aligned}\big|\mathcal{J}(x)-\mathcal{J}^{\ell,(j)}(x)\big| \leq{}& \sum_{j=1}^k \Big(\lambda_j + c\big|\hat{J}^{\ell,(j)}_j(u)-z_j-t+s_j\big|\Big)\Delta_{\hat{J}^{\ell,(j)}_j}(u)\\ &+\sum_{j=1}^k \frac{c}{2}\Big(\Delta_{\hat{J}^{\ell,(j)}_j}(u)\Big)^2 =: \Delta^{\ell,(j)}_{\mathcal{J}}(u)\end{aligned}$$

for all $u\in\mathcal{U}_{\mathsf{ad}}$, which can be used as a-posteriori error estimate in the TR-RB algorithm. According to Theorem 9, we still need to show the strict positivity of the costs $\mathcal{J}$ and $\mathcal{J}^{\ell,(j)}$ and the uniform Lipschitz continuity of the gradient $\nabla\mathcal{J}^{\ell,(j)}$. For the first, we note that the objectives $\mathcal{J}$ and $\mathcal{J}^{\ell,(j)}$ are bounded from below by $C := t^{\mathsf{min}} - \sum_{i=1}^k \lambda_i^2/(2\mu_i)$. Since C depends only on fixed parameters of the optimization problems, we can add $C+1$ to the cost functions to obtain strict positivity. Obviously, this will not change the minimizers. The second property is a bit more technical and we prove it in the following lemma.

Lemma 5. *Let the Lagrange multiplier λ and the penalty parameter μ be given. Then the function $\mathcal{J}(\cdot) := \mathcal{L}_A(\cdot,\lambda;\mu)$ is twice continuously Fréchet-differentiable for all $j\in\mathbb{N}$ and the gradient $\nabla\mathcal{J}^{\ell,(j)}$ is uniformly Lipschitz continuous with respect to j.*

Proof. Due to Corollary 1 the cost functions $\hat{J}_1,\ldots,\hat{J}_k$ are twice continuously Fréchet-differentiable. Thus, the function $(u,t,s)\mapsto\mathcal{L}_A((u,t,s),\lambda;\mu)$ is also twice continuously Fréchet-differentiable as a composition of twice continuously Fréchet-differentiable functions. Similarly, the reduced-order augmented Lagrangians $\mathcal{L}^{\ell,(j)}_A((\cdot,\cdot,\cdot),\lambda;\mu)$ are also twice continuously Fréchet-differentiable for all $j\in\mathbb{N}$. We have that

$$\nabla^2\mathcal{L}^{\ell,(j)}_A((u,t,s),\lambda;\mu)(h^u,h^t,h^s) = \begin{pmatrix} \sum\limits_{j=1}^k \Big(\big(\lambda_j+\mu c^{\ell,(j)}_j\big)\nabla^2\hat{J}^{\ell,(j)}_j(u)h^u + \mu\big(d^{\ell,(j)}_j - h^t + h^s_j\big)\nabla\hat{J}^{\ell,(j)}_j(u)\Big)\\ k\mu h^t - \mu\sum\limits_{j=1}^k\big(d^{\ell,(j)}_j + h^s_j\big)\\ \mu\big(d^{\ell,(j)}_1 + h^s_1 - h^t\big)\\ \vdots\\ \mu\big(d^{\ell,(j)}_k + h^s_k - h^t\big)\end{pmatrix}$$

for any $h = (h^u,h^t,h^s)\in\mathcal{U}\times\mathbb{R}\times\mathbb{R}^k$, where $c^{\ell,(j)}_j := \hat{J}^{\ell,(j)}_j(u) - z_j - t + s_j$ and $d^{\ell,(j)}_j := \langle\nabla\hat{J}^{\ell,(j)}(u),h^u\rangle_{\mathcal{U}}$ for $j\in\{1,\ldots,k\}$. Using Lemma 4, we obtain that the Hessian matrix $\nabla^2\mathcal{L}^{\ell,(j)}_A((u,t,s),\lambda;\mu)$ can be bounded independently of (u,t,s) and j. Using the mean value theorem, we can conclude that the gradients $\nabla\mathcal{L}^{\ell,(j)}_A((\cdot,\cdot,\cdot),\lambda;\mu)$ are Lipschitz-continuous with constant C_L uniformly in j. □

As a consequence of Theorem 9, we have that Algorithm 2 applied to solve the augmented Lagrangian subproblem (23) converges after finitely many steps to a first-order critical point of (23).

Remark 6. *Algorithm 2 constructs and updates the RB space during the optimization procedure. In the case of the PS method, we are free to choose what to do for the space constructed during the TR-RB procedure. For example, we can use it for the next augmented Lagrangian subproblem (and also for the next reference point). We explored different ideas (see also [16]), but we report here only the two most interesting and efficient ones:*

(1) *Use one common RB space for all the subproblems and reference points, i.e., use a single space V^ℓ (which is, of course, updated in the process) for solving the MOP. This strategy acquires efficiency in terms of reconstructing the full-order parameter space during the iterations. Therefore, thanks to the possibility of skipping an enrichment (which is the costly part in Algorithm 2), we expect more and more speed-up, together with accuracy, as the algorithm proceeds.*

(2) *Use multiple (local) RB spaces. This idea is already exploited by [16,37,38]. In this case, we do not use the previously obtained RB space for the next minimization problem. We generate instead k initial spaces $V_1^\ell, \ldots, V_k^\ell$, resulting from the minimization (Note that this procedure does not require extra computational cost, since we need to solve these problems for the hierarchical PS method anyway) of the objectives $\hat{J}_1, \ldots, \hat{J}_k$. Then at the beginning of every PS problem, we can decide to use the space V_i^ℓ for which $q^{(0)}(u^{(0)}) < \beta_q \delta^{(0)}$ and $\dim V_i^\ell \leq \ell_{\mathsf{max}}$, with $\ell_{\mathsf{max}} \in \mathbb{N}$ being a predefined maximal number of basis functions. If several spaces satisfy these conditions then we select the one for which the value $q^{(0)}(u^{(0)})$ is the smallest. If instead there is no space fulfilling these conditions, we initialize a new space V_{k+1}^ℓ by using the full-order quantities $\mathcal{S}(u^{(0)})$ and $\mathcal{A}_i(u^0)$ for $i = 1, \ldots, k$.*

Although these two techniques are already efficient, we noticed that there is a common problem: the number of RB basis functions might grow too fast and prevent a good speed-up for the solution. In particular, this is the case for the first strategy. To fix this issue, we propose different strategies to remove basis functions from V^ℓ in Section 4.2. This approach was not considered in [14,16–18] *and to our knowledge it has not been addressed in the literature yet. In reduced-order optimization, instead, this is meaningful, since the reduced-order model might grow too fast; see, e.g.,* [33], *in the case of proper orthogonal decomposition.*

4.2. How to Reduce the Number of Basis Functions

We point out that what is described in this section can also generally be applied to Algorithm 2 from [17] without any relation to the PS method. In particular, the strategies for reducing the number of basis functions presented in this section cannot only be used for PDE-constrained multi-objective optimization problems, but also for any other problem formulation containing PDE-constrained optimization problems. Therefore, we use again the general notation $\mathcal{J}$ for the cost, as it was done in the beginning of this section. The methodology to remove a basis function comes from the observation that some basis elements might not be used during the optimization process. Suppose that we start from a point $u^{(0)}$ very far from the optimum. Clearly, after j iterations the point $u^{(j)}$ is in a completely different region of the admissible set compared to the one of the starting point. Hence, the basis functions built for $u^{(0)}$ might give a negligible contribution in spanning the reduced-order model at the point $u^{(j)}$. If this is the case, we can expect that these functions will not play any further role also for the subsequent points and therefore they can be removed to reduce the dimension of the RB space. Our methodologies for removing basis functions are then based on Remark 6 and try to check which basis functions give a negligible contribution for the current iteration of the TR-RB algorithm. Notice that every technique we propose from now on will be applied after updating the RB space in the TR-RB algorithm. The aim is to modify the updated RB space in order to provide a new RB space, where the number of basis functions is reduced.

Technique T1. The first proposed technique is based on the computation of the so-called Fourier coefficients. Given $v \in V$ and a set of orthonormal basis functions $\{\psi_n\}_{n=1}^{\ell} \subset V^{\ell}$, the n-th Fourier coefficient is defined as $c_{\mathcal{F}}^{(n)}(v) := \langle v, \psi_n \rangle_V$. Now, T1 consists in computing $c_{\mathcal{F}}^{(n)}(\mathcal{S}(u^{(j+1)}))$ and $c_{\mathcal{F}}^{(n)}(\mathcal{A}_i(u^{(j+1)}))$, $i = 1, \dots, k$, for $n = 1, \dots, \ell$ and remove the basis function ψ_n for which

$$\zeta^{(n)} := \max\left\{ \frac{c_{\mathcal{F}}^{(n)}(\mathcal{S}(u^{(j+1)}))^2}{\sum_{\eta=1}^{\ell} c_{\mathcal{F}}^{(\eta)}(\mathcal{S}(u^{(j+1)}))^2}, \max_{i=1,\dots,k} \left\{ \frac{c_{\mathcal{F}}^{(n)}(\mathcal{A}_i(u^{(j+1)}))^2}{\sum_{\eta=1}^{\ell} c_{\mathcal{F}}^{(\eta)}(\mathcal{A}_i(u^{(j+1)}))^2} \right\} \right\}$$

is below a certain tolerance. Note, in fact, that the Fourier coefficients indicate the order of magnitude of the contribution of a given basis function in reconstructing the new snapshots that we want to add to update the RB. Strategy T1 is also based on the assumption that the snapshots, which we want to include in an update, are the most relevant for the new TR subproblem, because they correspond to the last accepted optimization step $u^{(j+1)}$. The advantage of T1 is that the required Fourier coefficients are already available from the Gram-Schmidt orthogonalization performed during the update of the RB space. There is, anyway, a possible drawback of T1 due to the tolerance we set: it can happen that also important basis functions are removed although one thinks that the tolerance is small enough. Because of this, we would like to have a criteria to decide in an unbiased way which basis functions should be removed.

Technique T2. This approach is based on the idea that once a point $u^{(j+1)}$ is accepted by the TR-RB algorithm and the RB space is updated, we will compute a provisional AGC point $u_{\mathsf{AGC}}^{(j+1),\mathsf{prov}}$ (cf. Definition 14) with respect to the previously updated RB space. One robustness criteria that we demand is that after removing basis functions, this provisional AGC point is still inside the new TR (Note that the TR depends on the reduced-order model due to the inequality constraint in (12) and, therefore, changes if we remove basis functions), although it might not coincide with the actual AGC point $u_{\mathsf{AGC}}^{(j+1)}$ that we compute after removing basis functions according to Line 3 in Algorithm 2 (Note that the reduced-order cost function changes by removing a basis function, so that also the first term in (13) differs after this removal). If we do not demand this robustness criteria, we can expect a deterioration of the TR performances due to lack of accuracy of the RB model in the steepest descent direction. Another important aspect is to guarantee the convergence of the TR-RB method, which implies checking that the conditions for accepting the point $u^{(j+1)}$ are still fulfilled, although we removed basis functions.

In summary, the difference with respect to T1 is then to remove basis functions starting from the one with the smallest value of $\zeta^{(n)}$ and proceeding in ascending order until one of the following conditions is satisfied

$$\frac{\Delta_{\mathcal{J}^{\ell-\mathsf{rem},(j+1)}}(u_{\mathsf{AGC}}^{(j+1),\mathsf{prov}})}{\mathcal{J}^{\ell-\mathsf{rem},(j+1)}(u_{\mathsf{AGC}}^{(j+1),\mathsf{prov}})} > \beta_q \delta^{(j+1)}, \tag{25a}$$

$$\frac{\Delta_{\nabla\mathcal{J}^{\ell-\mathsf{rem},(j+1)}}(u_{\mathsf{AGC}}^{(j+1),\mathsf{prov}})}{\|\nabla\mathcal{J}^{\ell-\mathsf{rem},(j+1)}(u_{\mathsf{AGC}}^{(j+1),\mathsf{prov}})\|_{\mathcal{U}}} > \min\{\tau_{\mathsf{grad}}, \beta_{\mathsf{grad}}\delta^{(j+1)}\}, \tag{25b}$$

$$\frac{\|\nabla\mathcal{J}^{\ell-\mathsf{rem},(j+1)}(u^{(j+1)}) - \nabla\mathcal{J}(u^{(j+1)})\|_{\mathcal{U}}}{\|\nabla\mathcal{J}^{\ell-\mathsf{rem},(j+1)}(u^{(j+1)})\|_{\mathcal{U}}} > \min\{\tau_{\mathsf{grad}}, \beta_{\mathsf{grad}}\delta^{(j+1)}\}, \tag{25c}$$

$$\frac{|g(u^{(j+1)}) - g^{\ell-\mathsf{rem},(j+1)}(u^{(j+1)})|}{g^{\ell-\mathsf{rem},(j+1)}(u^{(j+1)})} > \tau_g, \tag{25d}$$

$$\mathcal{J}^{\ell-\mathsf{rem},(j+1)}(u^{(j+1)}) > \mathcal{J}^{\ell,(j)}(u_{\mathsf{AGC}}^{(j)}), \tag{25e}$$

$$\mathcal{J}^{\ell-\mathsf{rem},(j+1)}(u_{\mathsf{AGC}}^{(j+1),\mathsf{prov}}) - \mathcal{J}(u^{(j+1)}) > -\kappa_{\mathsf{arm}}\|u_{\mathsf{AGC}}^{(j+1),\mathsf{prov}} - u^{(j+1)}\|_{\mathcal{U}}^2. \tag{25f}$$

If one of the conditions (25) holds we re-add the basis function to the RB space and finish the removal continuing with the TR-RB procedure. T2 is summarized in Algorithm 3.

Algorithm 3 Summary of T2

```
1: Follow the steps in Algorithm 2 until the RB model is updated at $u^{(j+1)}$;
2: Compute a provisional AGC point $u_{\mathsf{AGC}}^{(j+1),\mathsf{prov}}$ by using the reduced-order cost function
   w.r.t. the updated RB model;
3: Compute $\zeta^{(n)}$ for $n \in \{1, \dots, \ell\}$;
4: while None of the conditions in (25) is fullfiled do
5:     Out of all remaining basis functions, remove the one with the smallest value of $\zeta^{(n)}$
       from the RB space;
6: end while
7: Add the last removed basis function to the RB space;
8: Proceed with Algorithm 2 with the RB space obtained performing Steps 2–7;
```

Let us explain the meaning of (25). At first, the superindex $\ell - \mathsf{rem}$ indicates that the space used to compute the quantity is the RB space obtained after removing a basis function. Condition (25a) is to check that the provisional AGC point will remain inside an accurate-enough region of the TR. Condition (25b) is in the spirit of (25a) but for the gradient of the objective. Conditions (25c) and (25d) are based on the skipping enrichment criteria and are checked to ensure convergence and robustness of the method after the removal. For a similar issue we need to check that the sufficient decrease condition is fulfilled as well (cf. (25e)). Finally, (25f) is to enforce that the provisional AGC point is still a Cauchy point. In such a way, we are sure that Algorithm 2 converges even after performing the basis removal (cf. [17,18]). In this sense, T2 introduces an unbiased way to deal with the technique introduced in T1. There are still a few aspects one should comment on before implementing T2. At first, note that all the above-mentioned conditions are cheaply computable, since they are based either on reduced-order quantities or the appearing full-order quantities are available because of the RB update. At second, conditions (25a) and (25b) request efficient and reliable error estimators. Although for the PS method the efficiency of $\Delta_{\mathcal{J}}^{\ell,(j)}$ is acceptable, it is not the same for an error estimator $\Delta_{\nabla\mathcal{J}}^{\ell,(j)}$ based on the a-posteriori estimates of the gradients of the individual objectives. These estimators generally produce a huge overestimation, which makes them useless in practice. We notice, in fact, that condition (25b) is immediately triggered in the case of the PS method and we can not remove any basis function. This is the reason why we solved this issue by two different related approaches:

Technique T2a. We replace the numerator of (25b) by

$$\|\nabla\mathcal{J}^{\ell-\mathsf{rem},(j)}(u_{\mathsf{AGC}}^{(j+1),\mathsf{prov}}) - \nabla\mathcal{J}(u_{\mathsf{AGC}}^{(j+1),\mathsf{prov}})\|_{\mathcal{U}'}$$

which is the true error we wanted to estimate, but it is unfortunately costly. It requires the computation of the full-order quantities $\mathcal{S}(u_{\mathsf{AGC}}^{(j+1),\mathsf{prov}})$ and $\mathcal{A}_i(u_{\mathsf{AGC}}^{(j+1),\mathsf{prov}})$, $i = 1, \dots, k$.

Technique T2b. We replace the numerator of (25b) by

$$\|\nabla\mathcal{J}^{\ell-\mathsf{rem},(j)}(u_{\mathsf{AGC}}^{(j+1),\mathsf{prov}}) - \nabla\mathcal{J}^{\ell,(j+1)}(u_{\mathsf{AGC}}^{(j+1),\mathsf{prov}})\|_{\mathcal{U}}$$

which is a cheap approximation of the true error that we suppose to be reliable only after enough steps of Algorithm 2, however.

Clearly, if one has a good estimation of the gradient at hand, T2 can be still used in its original form.

Technique T3. Another drawback of T2 is the fact that we first need to remove the basis function in order to check (25). This implies that when we stop the removal, we need to add back the last basis function which was removed, because it is containing important information; cf. Line 7 of Algorithm 3. This results in a waste of time for the modified Algorithm 2. We decide to add the option of introducing numerical tolerances for each of the conditions (25). In such a way, the modified algorithm will generally stop before an important basis function is removed at the price of possibly leaving one or a few redundant basis functions in the RB space. We think that this is a meaningful modification regarding the time that is wasted reintroducing the removed basis function into the RB space; cf. Section 5. We indicate this last strategy as T3.

5. Numerical Experiments

In this section we test Algorithm 2 and compare it with the results obtained in [16] (Section 3.2.2). We use the same numerical setting, which we briefly report here. Let the domain Ω be the two-dimensional unit square, split into four different subdomains $\Omega_1 = (0, 0.5) \times (0, 0.5)$, $\Omega_2 = (0, 0.5) \times (0.5, 1)$, $\Omega_3 = (0.5, 1) \times (0, 0.5)$ and $\Omega_4 = (0.5, 1) \times (0.5, 1)$. For each Ω_i, we consider a corresponding diffusion coefficient $u_i^\kappa \in \mathbb{R}$ in (3) for $i = 1, \ldots, 4$. The reaction term $r(x)$ is set to be constantly equal to 1 for any $x \in \Omega$. We impose homogeneous Neumann boundary conditions (i.e., $\alpha = 0$) and a source term $f(x) = \sum_{i=1}^4 c_i \chi_{\Omega_i}(x)$ with $c_1 \approx 2.76$, $c_2 \approx -0.96$, $c_3 \approx 0.51$ and $c_4 \approx -1.66$ generated randomly in order to obtain a problem 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. For (**MPPOP**) we choose the following three objectives

$$\hat{J}_1(u) := \frac{1}{2} \|\mathcal{S}(u) - y_\Omega^{(1)}\|_H^2 + \frac{\varepsilon}{2} \|u - u_d^{(1)}\|_{\mathcal{U}}^2,$$
$$\hat{J}_2(u) := \frac{1}{2} \|\mathcal{S}(u) - y_\Omega^{(2)}\|_H^2 + \frac{\varepsilon}{2} \|u - u_d^{(2)}\|_{\mathcal{U}}^2, \quad \hat{J}_3(u) := \frac{0.05}{2} \|u - u_d^{(3)}\|_{\mathcal{U}}^2$$

with $\varepsilon = 0.002$, the desired states

$$y_\Omega^{(1)}(x) := \chi_{(0,0.5)\times(0,1)}(x), \quad y_\Omega^{(2)}(x) := \chi_{(0.5,1)\times(0,1)}(x),$$

and the desired parameter values

$$u_d^{(1)} = u_d^{(2)} := (2, 0, 0, 0, 0.3)^T, \quad u_d^{(3)} := (2, 1, 1, 1, 0.3)^T.$$

The lower and upper parameter bounds are given by

$$u_a = (2, 0.1, 0.1, 0.1, 0.3)^T \quad \text{and} \quad u_b = (2, 4, 4, 4, 0.3)^T,$$

respectively. This implies that $u_1^\kappa = 2$ and $u^r = 0.3$ are seen as constants and we only optimize over the three parameters u_2^κ, u_3^κ and u_4^κ. Note furthermore, that the desired parameters $u_d^{(1)} = u_d^{(2)}$ are not admissible. In fact, as for the parameters of the source term, they were chosen such that the resulting Pareto front is non-convex.

For the choice of the initial value for PSPs corresponding to reference points for the entire problem $(\hat{J}_1, \hat{J}_2, \hat{J}_3)$ we do the following: Let $\bar{u}^i$ be the minimizer of $\hat{J}_i$ for $i = 1, 2, 3$. Recall that the sets D_i have been introduced in Definition 7-(ii). Then, if $z \in D_i$, we choose $\bar{u}^i$ as the initial value for solving ($\mathbf{P}_{z,r}^{\mathsf{PS}}$). We additionally choose the shifting vectors $\tilde{d} = 0.001 \cdot (1, 1, 1)^T$, while the grid size h for the reference point grid is set to $h_{PSM} = 0.003$.

5.1. Parameter Choices for the TR-RB Algorithm

There are many parameters used in the TR-RB algorithm, which we will specify and briefly comment on in this section.

- The initial TR radius is chosen as $\delta^{(0)} = 0.1$, the tolerance for increasing the TR radius is set to $\eta_\varrho = 0.75$ and the factor for shrinking the TR radius to $\beta_1 = 0.5$. For the minimal TR radius we use $\delta_{\min} = 1 \times 10^{-16}$.

- For the Armijo backtracking strategy, we use the constants $\kappa_{\mathsf{arm}} = 1 \times 10^{-4}$ and $\kappa = 0.5$.
- The tolerance of the first-order condition is set to $\tau_{\mathsf{FOC}} = \tau^{(i)}_{\mathsf{FOC,sub}}$, where $\tau^{(i)}_{\mathsf{FOC,sub}}$ is the tolerance for the first-order condition of the current augmented Lagrangian subproblem. Moreover, we choose $\tau_{\mathsf{sub}} = 0.5\,\tau_{\mathsf{FOC}}$ as the tolerance of the first-order condition of the TR-subproblem and $\beta_{\mathsf{bound}} = 0.9$ as the constant in (15).
- For checking the necessity of updating the RB space, we choose $\tau_g = 1$, $\tau_{\mathsf{grad}} = 0.1$, $\beta_{\mathsf{grad}} = 0.2$ and $\beta_q = 0.005$.
- The tolerance chosen in T1 (cf. Section 4.2) for the Fourier coefficient is 10^{-6}. Similarly, we choose the same tolerance for T3 in order to break the removal algorithm before deleting important basis functions, i.e., we subtract it on the right-hand side of (25a)–(25f).

We notice in our numerical experiments that the method without basis removal is quite robust in terms of computational time and required PDE solves with respect to all the parameters except for the ratio between the first-order conditions of the current augmented Lagrangian subproblem and the TR-subproblem $\tau_{\mathsf{FOC}}/\tau_{\mathsf{sub}}$. In our experiments we choose this ratio to be 2, but we observe that a too large ratio (already 5 is sufficient) slows down the method considerably. The reason is that the TR-subproblems are solved with too much accuracy in this case which needs a lot of numerical effort but does not benefit the overall optimization. Regarding the techniques introduced in Section 4.2, T1 heavily depends on the choice of the tolerance for truncating the Fourier coefficient. The smaller the tolerance the less basis functions are removed. Anyway, if we remove too many basis functions (e.g., tolerance of 10^{-4}), T1 becomes less stable and the method needs more iterations to converge which generally corresponds in more enrichment steps which slow it down. Conversely, removing few basis functions (e.g., 10^{-8}) implies no significant differences between T1 and the method without removal. Contrarily to T1, techniques T2, T2a and T2b are only based on the same parameters which influence the behavior of the algorithm without removal. Their performances are also robust with respect to all these parameters in terms of basis functions removed. For T3 the same discussion applies, but this method is also sensitive to the tolerance chosen to break the removal algorithm before deleting presumed important basis functions. On one hand, if this tolerance is too high (e.g., 10^{-2}), the method will not remove a significant number of basis functions to influence the performances of the algorithm. On the other hand, if this is too low (e.g., 10^{-8}) T3 will be essentially equivalent to T2.

5.2. Numerical Ressults

In this section, we focus mainly on the comparison of our proposed TR-RB variants, briefly commenting on full-order versus reduced-order model. For detailed comments and results on the PS method applied on the FE and RB level, we refer to [16] (Section 3.2.2). At first, to validate our approach, we show in Figure 1 the obtained Pareto fronts by using the method in [16] (left) and our method (right). As one can see, there is no visible difference. The approximation error is, in fact, of the order of 10^{-6} for a Pareto point computed by all the proposed techniques (i.e., T1, T2a, T2b and T3) on average. This can be essentially explained by the fact that the termination criteria for Algorithm 2 relies on the full-order model. Therefore, any computed point is first-order critical for the FE model, up to the chosen stopping tolerance. Let us remark that this is not typical for model order reduction, where generally there is an additional approximation error due to the reduced-order model inaccuracy.

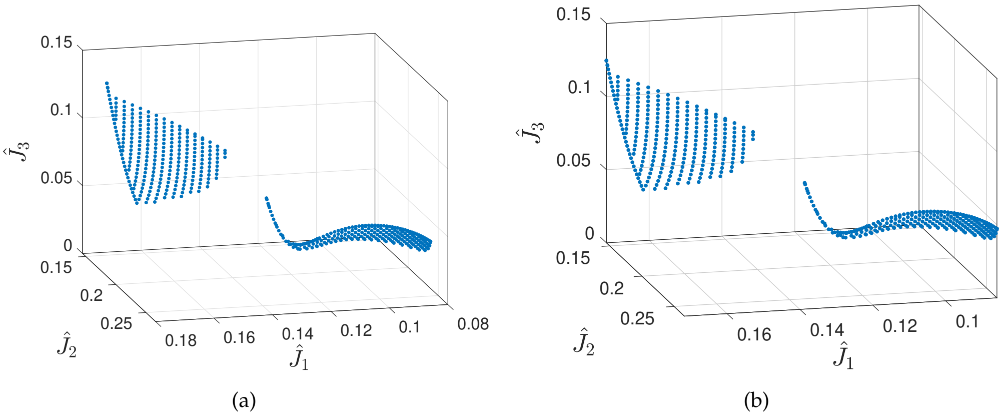


Figure 1. (**a**) Algorithm 2 no Removal local RB spaces. (**b**) Algorithm 2 T3 local RB spaces.

In Figure 2 we compare the computational time of Algorithm 2 for all the proposed techniques (cf. Section 4.2) against the full-order FE model and the algorithm in [16]. Concerning the FE method, we can save between 41% and 59% of the computational time. Considering the fact that we do not have an approximation error in reconstructing the Pareto points, we get the same result in approximately half of the time by using any of the TR-RB variants. This speed-up will also increase with an increasing number of degrees of freedom for the FE method, since the number of required FE solves of the PDE is significally smaller for the TR-RB algorithms than for the FE method; cf. Table 1.

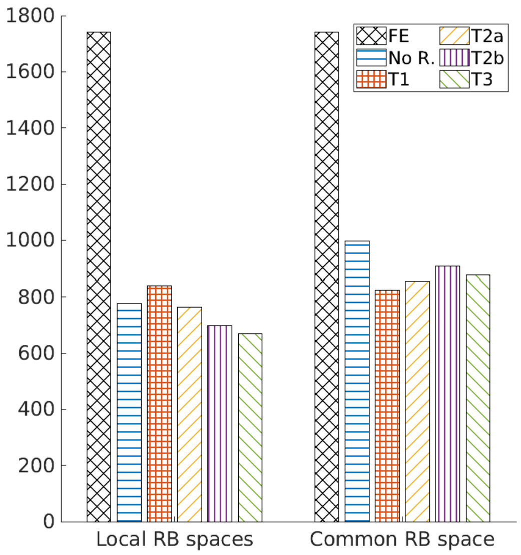


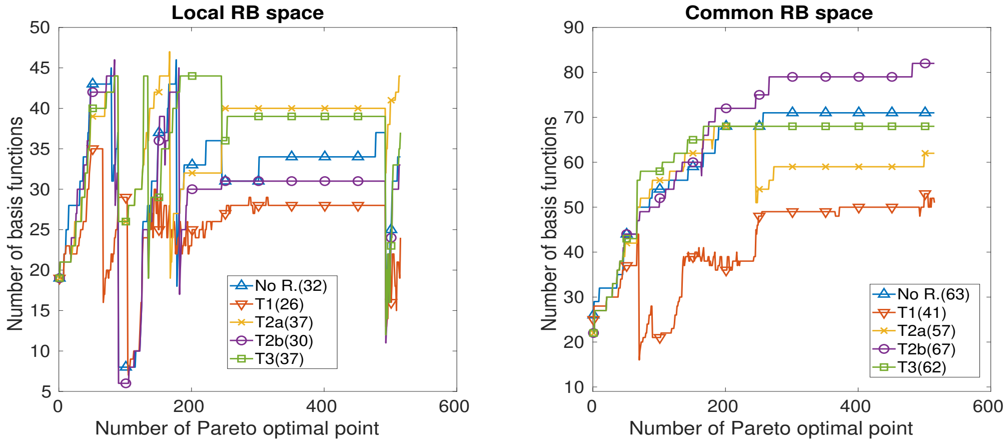
Figure 2. Computational times in seconds for Algorithm 2 with or without basis removal and using the two strategies in Remark 6 for initializing the RB space.

Table 1. Total PDE and only FE solves for the tested methods.

Method	# Total PDE Solves	# FE Solves
FE	433378	433378
Common RB Space No R.	493254	20743
Common RB Space T1	493282	20786
Common RB Space T2a	497032	20838
Common RB Space T2b	497032	20752
Common RB Space T3	493985	20792
Local RB Space No R.	497072	20773
Local RB Space T1	497589	20893
Local RB Space T2a	507064	21226
Local RB Space T2b	507064	20857
Local RB Space T3	502911	21023

Furthermore, we get a speed-up of the TR-RB algorithm by using the proposed techniques for reducing the number of basis functions in almost all cases. Depending on the strategy from Remark 6, one technique performs better than the others. Here we try to explain this phenomena in detail. Let us focus on the common RB space first. In this case, every technique helps in saving computational time. This is clearly the effect of removing redundant basis functions, which are particularly frequently included using a large common RB space. This is the reason why T1 appears to be the most effective, since it is the cheapest among the techniques (as we said it does not imply additional cost to be checked). T2a is more robust, but it comes with the price of evaluating the full-order gradient at the new AGC point and thus results to be slower than T1. Apparently, T2b should overcome this problem, but the inaccuracy of the RB space in the beginning yields a bad approximation of (25b), resulting in removing too many basis functions which leads to a worse approximation for the consecutive steps. This worsening of the approximation results in a way larger number of enrichment steps towards the end of the algorithm, which also negatively influences the computational time. T3 is comparable with T2a, meaning that for this example we are removing many basis functions in only a few instances, rather than frequently removing a few basis functions. Figure 3b confirms the above remarks for the case of a common RB space. In this figure we report the number of basis functions obtained at the end of Algorithm 2 while this is applied to compute each Pareto optimal point in the PS method.

Now, let us focus on the left group of columns in Figure 2 (and thus on Figure 3a), which corresponds to the computational times in the case of using local RB spaces (cf. Remark 6). This case is a bit more delicate, since the use of local RB spaces makes it more difficult to interpret the results. Here the problem of T1 is emerging. The fact that this technique removes a number of basis functions without any robustness criteria implies that the method slows down. In the case of local spaces, in fact, we do not have the same amount of redundant basis functions as it can occur for a common RB space. Therefore, we should only remove the basis functions which are actually redundant. As one can note in Figure 3a, T1 removes a significantly larger amount of basis functions in comparison to the other techniques. Here the criteria introduced in T2a play their role in a positive way. We can counteract the effect of T1 in such a way that the computational time is comparable to the one in [16]. The further simplification introduced in T2b helps to get an additional speed-up. In contrast to the common RB space, here we have local spaces which provide a sufficiently good accuracy for approximating (25b) also in the beginning of the optimization. This is then beneficial for the algorithm, since the cost of computing the criteria in T2b is way cheaper than T2a, where we need full-order solves of the state and adjoint equation to compute the gradient at the new AGC point. Additionally, T3 further improves T2a and T2b in terms of computational time, since in the case of local RB spaces it is more probable that we indeed remove only a few basis functions but more frequently than in the case of one common RB space. In this case, it is important to have tolerances that let us stop before removing an important basis function and save time for reintroducing it in the RB space.



(a) (b)

Figure 3. Number of basis functions used to compute each Pareto optimal point. (**a**) Local RB space. (**b**) Common RB space. In brackets: average number of basis functions.

In conclusion, comparing our fastest method (i.e., Algorithm 2 with local RB spaces and T3) to the slowest (i.e., using [16] with a common RB space) we get essentially the same results (the approximation error is 10^{-6}) saving approximately 30% of the computational time, which is roughly 300 s. This shows how one should invest time and resources in providing efficient techniques for reducing the number of basis functions in the RB space, while using an adaptive TR-RB algorithm. Particularly in the case of multiobjective optimization, this becomes crucial for a large number of cost functionals k. To obtain the same resolution of the Pareto front as in Figure 1 for a large k, we will need to solve the PSPs for many more points, implying higher risk of having redundant basis functions.

6. Conclusions

We showed the applicability and convergence of the TR-RB algorithm in the context multi-objective PDE-constrained parameter optimization problem. We presented and analyzed novel ways of reducing the dimension of the RB space during the optimization procedure. To our knowledge, basis reduction strategies have not been proposed yet for the RB method, although it is common for other model order reduction techniques. Such a removal significantly improved the performances of the TR-RB algorithm in the context of multiobjective optimization, leading faster to an accurate solution than the already existing techniques. The presented example contained only three parameters to be optimized. However, based on the results in [17] (Section 4.4) for an example with 28 parameters and on the various examples in [39] (Sections 3.5.4–3.5.6), we expect all of the TR-RB methods to scale well with an increasing number of parameters. As for the multi-objective optimization by the PS method, the numerical effort grows exponentially with the number of cost functions k, but is independent of the number of parameters m if $m \geq k - 1$. Moreover, the presented removal techniques of reduced basis functions can also be extended to other applications in which sequential parametric PDE-constrained optimization problems must be solved. In future work, one can try to extend the convergence theory for the presented TR-RB algorithm to a larger class of PDEs than the one presented here, as, e.g., parabolic PDEs [14] or non-affine parameter-to-state couplings. Due to the general formulation of the convergence result we are optimistic that this is possible. Moreover, one can try to achieve further improvements concerning robustness of the method and deriving tighter a-posteriori error estimators, in particular for the gradient of the cost function. This is also of great interest in the RB community. Another interesting idea could be to incorporate the usual trust-region condition based on the (Euclidean) distance from the current iterate into the presented TR-RB algorithm. In [19] the usual trust-region condition was actually performing slightly better than a residual-based error estimate as the trust-region constraint for some of the considered problems. Despite the fact that we use not only a residual-based error estimate but an error estimate of the actual cost function, a comparison between the different approaches is definitely of interest.

Author Contributions: Conceptualization, S.B., L.M. and S.V.; methodology, S.B., L.M. and S.V.; software, S.B. and L.M.; formal analysis, S.B., L.M. and S.V.; investigation, S.B., L.M. and S.V.; writing—original draft preparation, S.B., L.M. and S.V.; writing—review and editing, S.B., L.M. and S.V.; funding acquisition, S.V. All authors have read and agreed to the published version of the manuscript.

Funding: The authors acknowledge funding by the Deutsche Forschungsgemeinschaft (DFG) for the project Localized Reduced Basis Methods for PDE-constrained Parameter Optimization under contract VO 1658/6-1.

Acknowledgments: The authors thank Tim Keil, Mario Ohlberger and Felix Schindler from University of Münster (Germany) for the fruitful exchange of ideas on the topic.

Conflicts of Interest: The authors declare no conflict of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript, or in the decision to publish the results.

Abbreviations

The following abbreviations are used in this manuscript:

AGC	Approximated generalized Cauchy
CG	Conjugate gradient
FE	Finite element
MOP	Multiobjective optization problem
MPPOP	Multiobjective parametric PDE-constrained optimization problem
PDE	Partial Differential Equation
PS	Pascoletti-Serafini
RB	Reduced basis
s.t.	subject to
TR	Trust-region

References

1. Ehrgott, M. *Multicriteria Optimization*, 2nd ed.; Springer: Berlin/Heidelberg, Germany, 2005.
2. Miettinen, K. *Nonlinear Multiobjective Optimization*; Kluwer Academic Publishers: Cambridge, MA, USA, 1999.
3. Zadeh, L. Optimality and non-scalar-valued performance criteria. *IEEE Trans. Autom. Control* **1963**, *8*, 59–60.
4. Eichfelder, G. *Adaptive Scalarization Methods in Multiobjective Optimization*; Springer: Berlin/Heidelberg, Germany, 2008.
5. Pascoletti, A.; Serafini, P. Scalarizing vector optimization problems. *J. Optim. Theory Appl.* **1984**, *42*, 499–524.
6. Hinze, M.; Pinnau, R.; Ulbrich, M.; Ulbrich, S. *Optimization with PDE Constraints*; Springer Science + Business Media B.V.: Berlin/Heidelberg, Germany, 2009.
7. Schilders, W.H.; Van der Vorst, H.A.; Rommes, J. *Model Order Reduction*; Springer: Berlin/Heidelberg, Germany, 2008.
8. Hesthaven, J.S.; Rozza, G.; Stamm, B. *Certified Reduced Basis Methods for Parametrized Partial Differential Equations*; SpringerBriefs in Mathematics: Heidelberg, Germany, 2016.
9. Patera, A.T.; Rozza, G. *Reduced Basis Approximation and a Posteriori Error Estimation for Parametrized Partial Differential Equations*; MIT Pappalardo Graduate Monographs in Mechanical Engineering: Cambridge, MA, USA, 2007.
10. Banholzer, S.; Gebken, B.; Reichle, L.; Volkwein, S. ROM-based inexact subdivision methods for PDE-constrained multiobjective optimization. *Math. Comput. Appl.* **2021**, *26*, 32.
11. Iapichino, L.; Ulbrich, S.; Volkwein, S. Multiobjective PDE-constrained optimization using the reduced-basis method. *Adv. Comput. Math.* **2017**, *43*, 945–972.
12. Schu, M. Adaptive Trust-Region POD Methods and Their Application in Finance. Ph.D. Thesis, University of Trier, Trier, Germany, 2012. Available online: https://ubt.opus.hbz-nrw.de/opus45-ubtr/frontdoor/deliver/index/docId/574/file/PhD_Thesis_Schu.pdf (accessed on 28 April 2022).
13. Arian, E.; Fahl, M.; Sachs, W.S. *Trust-Region Proper Orthogonal Decomposition for Flow Controls*; Techincal Report No. 2000–2025; Institute for Computer Applications in Science and Engineering, NASA Langley Research Center: Hampton, VA, USA, 2000.

14. Qian, E.; Grepl, M.; Veroy, K.; Willcox, K. A certified trust region reduced basis approach to PDE-constrained optimization. *SIAM J. Sci. Comput.* **2017**, *39*, S434–S460.
15. Yue, Y.; Meerbergen, K. Accelerating optimization of parametric linear systems by model order reduction. *SIAM J. Optimiz.* **2013**, *23*, 1344–1370.
16. Banholzer, S. ROM-Based Multiobjective Optimization with PDE Constraints. Ph.D. Thesis, University of Konstanz, Konstanz, Germany, 2021. Available online: http://nbn-resolving.de/urn:nbn:de:bsz:352-2-1g98y1ic7inp29 (accessed on 28 April 2022).
17. Banholzer, S.; Keil, T.; Mechelli, L.; Ohlberger, M.; Schindler, F.; Volkwein, S. An adaptive projected Newton non-conforming dual approach for trust-region reduced basis approximation of PDE-constrained parameter optimization. *arXiv* **2020**, arXiv:2012.11653.
18. Keil, T.; Mechelli, L.; Ohlberger, M.; Schindler, F.; Volkwein, S. A non-conforming dual approach for adaptive trust-region reduced basis approximation of PDE-constrained optimization. *ESAIM M2AN* **2021**, *55*, 1239–1269.
19. Yano, M.; Huang, T.; Zahr, M.J. A globally convergent method to accelerate topology optimization using on-the-fly model reduction. *Comput. Methods Appl. Mech. Eng.* **2021** *375*, 113635.
20. Zahr, M.J.; Carlberg, K.T.; Kouri, D.P. An efficient, globally convergent method for optimization under uncertainty using adaptive model reduction and sparse grids. *SIAM/ASA J. Uncertain. Quantif.* **2019**, *7*, 877–912.
21. Kouri, D.P.; Heinkenschloss, M.; Ridzal, D.; van Bloemen Waanders, B.G. A trust-region algorithm with adaptive stochastic collocation for PDE optimization under uncertainty. *SIAM J. Sci. Comput.* **2013** *35*, A1847–A1879.
22. Kouri, D.P.; Heinkenschloss, M.; Ridzal, D.; van Bloemen Waanders, B.G. Inexact objective function evaluations in a trust-region algorithm for PDE-constrained optimization under uncertainty. *SIAM J. Sci. Comput.* **2014** *36*, A3011–3029.
23. Grüne, L.; Pannek, J. *Nonlinear Model Predictive Control: Theory and Algorithms*, 2nd ed.; Springer: London, UK, 2016.
24. Borwein, J.M. On the existence of Pareto efficient points. *Math. Oper. Res.* **1983**, *8*, 64–73.
25. Hartley, R. On cone-efficiency, cone-convexity and cone-compactness. *SIAM J. Appl. Math.* **1978**, *34*, 211–222.
26. Sawaragi, Y.; Nakayama, H.; Tanino, T. *Theory of Multiobjective Optimization*; Elsevier: Amsterdam, The Netherlands, 1985.
27. Wierzbicki, A.P. The Use of Reference Objectives in Multiobjective Optimization. In *Multiple Criteria Decision Making Theory and Application*; Springer: Berlin/Heidelberg, Germany, 1980; pp. 468–486.
28. Mueller-Gritschneder, D.; Graeb, H.; Schlichtmann, U. A successive approach to compute the bounded Pareto front of practical multiobjective optimization problems. *SIAM J. Optim.* **2009**, *20*, 915–934.
29. de Motta, R.S.; Afonso, S.M.B.; Lyra, P.R.M. A modified NBI and NC method for the solution of N-multiobjective optimization problems. *Struct. Multidiscip. Optim.* **2012**, *46*, 239–259.
30. Khaledian, K.; Soleimani-damaneh, M. A new approach to approximate the bounded Pareto front. *Math. Method Oper. Res.* **2015**, *82*, 211–228.
31. Lowe, T.J.; Thisse, J.-F.; Ward, J.E.; Wendell, R.E. On efficient solutions to multiple objective mathematical programs. *Manag. Sci.* **1984**, *30*, 1346–1349.
32. Sayın, S. Measuring the quality of discrete representations of efficient sets in multiple objective mathematical programming. *Math. Program.* **2000**, *87*, 543–560.
33. Mechelli, L. POD-Based State-Constrained Economic Model Predictive Control of Convection-Diffusion Phenomena. Ph.D. Thesis, University of Konstanz, Konstanz, Germany, 2019. Available online: http://nbn-resolving.de/urn:nbn:de:bsz:352-2-2zoi8n9sxknm1 (accessed on 28 April 2022).
34. Evans, L.C. *Partial Differential Equations*; American Mathematical Society: Providence, RI, USA, 2010.
35. Haasdonk, B. Reduced basis methods for parametrized PDEs—A tutorial introduction for stationary and instationary problems. In *Model Order Reduction and Approximation: Theory and Algorithms*; Benner, P., Ohlberger, M., Cohen, A., Willcox, K., Eds.; SIAM: Philadelphia, PA, USA, 2017; pp. 65–136.
36. Rozza, G.; Huynh, D.B.P.; Patera, A.T. Reduced basis approximation and a posteriori error estimation for affinely parametrized elliptic coercive partial differential equations. *Arch. Comput. Method E* **2008**, *15*, 229–275.
37. Beermann, D.; Dellnitz, M.; Peitz, S.; Volkwein, S. Set-oriented multi- objective optimal control of PDEs using proper orthogonal decomposition. In *Reduced-Order Modeling (ROM) for Simulation and Optimization*; Keiper, W., Milde, A., Volkwein, S., Eds.; Springer International Publishing: Berlin/Heidelberg, Germany, 2018; pp. 47–72.
38. Haasdonk, B.; Dihlmann, M.; Ohlberger, M. A training set and multiple bases generation approach for parameterized model reduction based on adaptive grids in parameter space. *Math. Comput. Model. Dyn.* **2011**, *17*, 423–442.
39. Keil, T. Adaptive Reduced Basis Methods for Multiscale Problems and Large-Scale PDE-Constrained Optimization. Ph.D. Thesis, WWU Münster, Münster, Germany, 2022.