OJMO

Open Journal of Mathematical Optimization

Manuel Berkemeier & Sebastian Peitz

Multi-Objective Trust-Region Filter Method for Nonlinear Constraints using Inexact Gradients

Volume 6 (2025), article no. 11 (36 pages)

https://doi.org/10.5802/ojmo.47

Article submitted on January 20, 2023, revised on October 31, 2024, accepted on November 10, 2025.

Open Journal of Mathematical Optimization is a member of the Centre Mersenne for Open Scientific Publishing
http://www.centre-mersenne.org/
e-ISSN: 2777-5860

CENTRE MERSENNE

Multi-Objective Trust-Region Filter Method for Nonlinear Constraints using Inexact Gradients

Manuel Berkemeier
Department of Computer Science
TU Dortmund University
Germany
manuel.berkemeier@tu-dortmund.de

Sebastian Peitz
Department of Computer Science
TU Dortmund University
Germany
sebastian.peitz@tu-dortmund.de

Abstract

In this article, we build on previous work to present an optimization algorithm for non-linearly constrained multi-objective optimization problems. The algorithm combines a surrogate-assisted derivative-free trust-region approach with the filter method known from single-objective optimization. Instead of the true objective and constraint functions, so-called *fully linear* models are employed, and we show how to deal with the gradient inexactness in the composite step setting, adapted from single-objective optimization as well. Under standard assumptions, we prove convergence of a subset of iterates to a quasi-stationary point and, if constraint qualifications hold, then the limit point is also a KKT-point of the multi-objective problem.

Digital Object Identifier 10.5802/ojmo.47

Keywords Multi-Objective Optimization, Multiobjective Optimization, Nonlinear Optimization, Derivative-Free Optimization, Trust-Region Method, Surrogate Models, Filter Method.

Acknowledgments Research funded by the DFG Priority Programme 2353 "Daring more intelligence" (project ID 501834605). We warmly thank the two anonymous reviewers for their insightful input, which has helped to vastly improve this article.

1 Introduction

Multi-objective optimization problems (MOPs) arise naturally in many areas of mathematics, engineering, in the natural sciences or in economics. The goal of multi-objective optimization (MOO) is to find acceptable trade-offs between the competing objectives of a MOP. Generally, there are multiple solutions constituting the so-called Pareto Set in variable space and the Pareto Front in objective space. In this article, we consider the problem

$$\min_{\boldsymbol{x}\in\mathbb{R}^n} \begin{bmatrix} f_1(\boldsymbol{x}) \\ \vdots \\ f_K(\boldsymbol{x}) \end{bmatrix} = \min_{\boldsymbol{x}\in\mathbb{R}^n} \boldsymbol{f}(\boldsymbol{x}) \quad \text{s.t.} \quad \boldsymbol{h}(\boldsymbol{x}) = [h_1(\boldsymbol{x}), \dots, h_M(\boldsymbol{x})]^{\mathsf{T}} = \boldsymbol{0},\ \boldsymbol{g}(\boldsymbol{x}) = [g_1(\boldsymbol{x}), \dots, g_P(\boldsymbol{x})]^{\mathsf{T}} \le \boldsymbol{0}, \qquad \text{(MOP)}$$

where all functions are twice continuously differentiable. A global Pareto-optimal point $\boldsymbol{x}^*$ is feasible and non-dominated, i.e., there is no other feasible $\boldsymbol{x} \neq \boldsymbol{x}^*$ with $\boldsymbol{f}(\boldsymbol{x}) \le \boldsymbol{f}(\boldsymbol{x}^*)$ and $f_{\mathfrak{l}}(\boldsymbol{x}) < f_{\mathfrak{l}}(\boldsymbol{x}^*)$ for some $\mathfrak{l} \in \{1, \dots, K\}$. Throughout this article we will refer to the feasible set as $\mathcal{X} := \{\boldsymbol{x} \in \mathbb{R}^n : \boldsymbol{h}(\boldsymbol{x}) = \boldsymbol{0} \text{ and } \boldsymbol{g}(\boldsymbol{x}) \le \boldsymbol{0}\}$.

There is a multitude of methods available to approximate a single solution or the entire Pareto Set/Front (or a superset thereof) and the choice of method is heavily dependent on the structure of the problem at hand and the demands of the person seeking a solution. The references [27, 29, 44, 56] all provide an extensive overview of the topic. Amongst others, there are scalarization approaches [28] and adaptions of single-objective descent methods to the multi-objective case for different problem classes as defined by the properties of their objectives and constraint functions, see [23, 33, 34, 35, 37, 38, 51, 76] for examples of well-known scalar techniques adjusted for constrained and unconstrained MOPs in the smooth and non-smooth case. For global approximations there are, e.g., evolutionary algorithms (see [13] for an overview and [19] for a prominent example) and structure-exploiting methods [39, 42, 53]. Of course, there is also research in combining global and local techniques [61, 66]. Whilst there are natural applications of MOO for machine learning tasks [1, 67], machine learning techniques can conversely be used to assist the search for optimal points [60]. In case of expensive objective or constraints, surrogate models can be employed [10, 20, 62].

In our setting we assume (some) objective and constraint functions to be computationally expensive and without exact derivative information available. This motivates the use of *derivative-free* optimization methods, which also have been adapted to the multi-objective case. For example, there are direct search algorithms [4, 7, 8, 17, 21, 68], implicit filtering [12] and linesearch [50] algorithms, or trust-region algorithms using derivative-informed approximations [57, 64, 70] or derivative-free surrogate models [58, 65, 69]. Based on such trust-region algorithms, we have in a previous article [6] presented a trust-region algorithm for problems with a feasible set that is convex and compact (or $\mathcal{X} = \mathbb{R}^n$). The algorithm uses fully linear models (e.g., Lagrange polynomials or radial basis functions (RBFs) as in [75]) to approximate the objective functions and passes the exact constraints to an inner solver. We build upon this work to accommodate general non-linear constraints by also modelling them and solving inexact sub-problems.

To this end, we transfer the techniques from [32] to the multi-objective case with inexact derivatives. Inexact derivative have already been handled in single-objective optimization in a similar manner. For example, the authors of [30] provide strong convergence results for inexact objectives and objective derivatives with similar model accuracy requirements to our case. We will also discuss the single-objective algorithm presented in [25], as it is also based on fully linear models and employs a special kind of criticality check. Our work, however, is more along the lines of [72], where the filter trust-region algorithm and the composite step framework are likewise modified to handle inexact derivatives of both the objective and constraint functions. But in [72] the derivatives (approximated via automatic differentiation) can eventually become exact, in contrast to our setting, which is why (without constraint qualifications) we can only prove convergence to a *quasi-stationary* point, similar to [26]. The algorithms in [26] and [36, Chapter 4] also use fully linear surrogate models satisfying the same error bounds as assumed in this article within single-objective filter algorithms. In [26], convergence to KKT-points is proven under Linear Independence Constraint Qualifications. Similarly, we will see that our convergence result can be proven under the Mangasarian-Fromovitz Constraint Qualifications. Such constraint qualifications are not needed in [36]. But to show convergence to critical points, in [36] an additional bound for the model accuracy is required, which is viable if the projection onto the feasible set can be done with arbitrary precision. In contrast, our approach uses projections on approximations of the linearized feasible set.

In next section, we will state the relevant optimality conditions for the multi-objective case. Afterwards, the main building blocks of our algorithm are described in Section 2. The algorithm itself is given in Section 3 and convergence is shown in Sections 4 and 5. Finally, numerical examples and comparisons are presented in Section 6 with a brief discussion in Section 7.

1.1 Optimality Conditions

In this subsection we introduce necessary conditions for a point to be locally Pareto-optimal according to the following definition.

▶ **Definition 1.** *The point $\boldsymbol{x}^* \in \mathcal{X}$ is locally Pareto-optimal, if there is a neighborhood $\mathcal{U}_\varepsilon = \{\boldsymbol{x} \in \mathcal{X} : \|\boldsymbol{x} - \boldsymbol{x}^*\| \le \varepsilon\}$ such that $\boldsymbol{x}^*$ is non-dominated in $\mathcal{U}_\varepsilon$, that is, there is no $\boldsymbol{x} \in \mathcal{U}_\varepsilon$ such that for all $\mathfrak{l} \in \{1, \dots, K\}$ it holds that $f_\mathfrak{l}(\boldsymbol{x}) \le f_\mathfrak{l}(\boldsymbol{x}^*)$ and $f_{\mathfrak{l}_*}(\boldsymbol{x}) < f_{\mathfrak{l}_*}(\boldsymbol{x}^*)$ for at least one index $\mathfrak{l}_* \in \{1, \dots, K\}$.*

To state the optimality criteria, we require the following assumption to hold, so that all functions are sufficiently smooth:

▶ **Assumption 2.** *The objective functions $f_\mathfrak{l}$, $\mathfrak{l} = 1, \dots, K$, and the constraint functions $g_\mathfrak{i}$, $\mathfrak{i} = 1, \dots, P$, and $h_\mathfrak{j}$, $\mathfrak{j} = 1, \dots, M$, are twice continuously differentiable in an open domain containing $\mathcal{X}$ and have Lipschitz continuous gradients on $\mathcal{X}$.*

There then is a formulation of Fermat's Theorem for optimization with multiple objectives:

▶ **Theorem 3** ([49, Theorem 5.2]). *Suppose Assumption 2 holds. If $\boldsymbol{x}^* \in \mathcal{X}$ is locally Pareto-optimal for* (MOP), *then there is no $\boldsymbol{d} \in \mathcal{T}_\mathcal{X}(\boldsymbol{x}^*)$ such that for all $\mathfrak{l} \in \{1, \dots, K\}$ it holds that $\boldsymbol{d}^\mathsf{T} \cdot \boldsymbol{\nabla} f_\mathfrak{l}(\boldsymbol{x}^*) < 0$, where $\mathcal{T}_\mathcal{X}(\boldsymbol{x}^*)$ is the* *tangent cone of $\mathcal{X}$ at $\boldsymbol{x}^*$, i.e., the set of vectors $\boldsymbol{v} \in \mathbb{R}^n$ for which there is a sequence $\boldsymbol{v}^{(k)} \in \mathcal{X}$ and scalars $\alpha_k > 0$ with*

$$\lim_{k\to\infty} \boldsymbol{v}^{(k)} = \boldsymbol{x}^*, \lim_{k\to\infty} \alpha_k = 0, \lim_{k\to\infty} \frac{\boldsymbol{v}^{(k)} - \boldsymbol{x}^*}{\alpha_k} = \boldsymbol{v}.$$

We call a point $\boldsymbol{x}^* \in \mathcal{X}$ satisfying the criterion in Theorem 3 *Pareto-critical* or stationary. Theorem 3 can be used to motivate the following problem to compute a descent direction and check for criticality:

$$\min_{\boldsymbol{d}\in\mathcal{T}_{\mathcal{X}}(\boldsymbol{x}), \|\boldsymbol{d}\|_2\le 1} \max_{\mathfrak{l}=1,\dots,K} \boldsymbol{d}^{\mathsf{T}} \cdot \nabla f_{\mathfrak{l}}(\boldsymbol{x}) \tag{1}$$

For a critical point $\boldsymbol{x} \in \mathcal{X}$ the optimal value is zero, else the minimizer is a multi-descent direction (cf. [2, Theorem 1.8.]). The choice of norm in problem (1) is not really important. We could instead use a linear norm and assume that constraint qualifications hold at $\boldsymbol{x} \in \mathcal{X}$, ensuring that the tangent cone equals the set of linearized directions

$$L(\boldsymbol{x}) = \left\{\boldsymbol{d} \in \mathbb{R}^n : \boldsymbol{d}^{\mathsf{T}} \cdot \nabla h_{\mathfrak{j}}(\boldsymbol{x}) = 0, \, \mathfrak{j} = 1, \dots, M, \; \boldsymbol{d}^{\mathsf{T}} \cdot \nabla g_{\mathfrak{i}}(\boldsymbol{x}) \le 0, \, \mathfrak{i} \in A(\boldsymbol{x})\right\},$$

with $A(\boldsymbol{x}) := \{\mathfrak{i} \in \{1, \dots, P\} : g_{\mathfrak{i}}(\boldsymbol{x}) = 0\}$, to obtain a linear problem. These constraint qualifications are also used to derive the KKT conditions for (MOP) from Theorem 3 [55, Theorem 2.3.]. See [33, 59] for prominent examples of descent algorithms using the set of linearized directions. In our algorithm, however, we do not use the sets $L(\boldsymbol{x}^{(k)})$. Instead, at $\boldsymbol{x}^{(k)}$ (which does not have to be feasible) we use the linearized feasible set,

$$\mathcal{L}(\boldsymbol{x}^{(k)}) := \left\{\boldsymbol{x}^{(k)} + \boldsymbol{d} \in \mathbb{R}^n : \boldsymbol{h}(\boldsymbol{x}^{(k)}) + \boldsymbol{H}(\boldsymbol{x}^{(k)})\cdot\boldsymbol{d} = \boldsymbol{0}, \; \boldsymbol{g}(\boldsymbol{x}^{(k)}) + \boldsymbol{G}(\boldsymbol{x}^{(k)})\cdot\boldsymbol{d} \le \boldsymbol{0}\right\},$$

where $\boldsymbol{H}(\boldsymbol{x}^{(k)})$ and $\boldsymbol{G}(\boldsymbol{x}^{(k)})$ denote the *full* Jacobian matrices of the constraints $\boldsymbol{h}$ and $\boldsymbol{g}$, respectively. The set $\mathcal{L}(\boldsymbol{x}^{(k)}) - \boldsymbol{x}^{(k)}$ is not necessarily a cone, but for feasible $\boldsymbol{x}^{(k)}$, it is a subset of $L(\boldsymbol{x}^{(k)})$, and intuitively it should not matter which of the sets is used near critical points. Indeed, the set $\mathcal{L}(\boldsymbol{x}^{(k)})$ appears in the sequential optimality condition in [54]. Similar to [26], we are hindered from using this sequential condition directly by the fact that $\boldsymbol{g}$ and $\boldsymbol{h}$ are actually approximated by local surrogate models (see next section). For now, assume that $\mathcal{L}_k$ is a surrogate-based *approximation* of $\mathcal{L}(\boldsymbol{x}^{(k)})$. We will show that our algorithm produces a sequence that approximates a feasible point, which is also *quasi-stationary*: the values of (1) tend to zero if we replace $\mathcal{T}_{\mathcal{X}}\big(\boldsymbol{x}^{(k)}\big)$ with $\mathcal{L}_k$. The convergence result is strengthened by requiring the Mangasarian-Fromovitz Constraint Qualifications (MFCQs) to hold at the limit point $\overline{\boldsymbol{x}}$. Then the solution set sequence of the problems resulting from substituting $\mathcal{L}_k$ into (1) is continuous at $\overline{\boldsymbol{x}}$, and a vanishing approximation error for $\mathcal{L}_k$ ensures that in the limit the solution set is the same as it is for the exact linearized set $\mathcal{L}(\overline{\boldsymbol{x}})$. Furthermore, the MFCQs make the limit point a KKT point (similar to [40]).

▶ **Theorem 4.** *Suppose that $\boldsymbol{x}$ is feasible and that suitable constraint qualifications hold, e.g., the MFCQs. If the linear optimization problem*

$$-\min_{\substack{\boldsymbol{d}\in\mathcal{L}(\boldsymbol{x})-\boldsymbol{x},\\ \|\boldsymbol{d}\|_\infty\le 1}} \max_{\mathfrak{l}=1,\dots,K} \boldsymbol{d}^{\mathsf{T}} \cdot \nabla f_{\mathfrak{l}}(\boldsymbol{x}) \tag{2}$$

has zero as its optimal value, then $\boldsymbol{x}$ is also a KKT-point of (MOP)*.*

Proof. We refrain from listing the full set of KKT conditions and instead refer to

- their first appearance for MOOs with inequality constraints, [46],
- an extension to equality constraints, [55],
- and [42, 53] for more contemporary notation.

Dropping the argument $\boldsymbol{x}$ for notational convenience, e.g., letting $\boldsymbol{g} = \boldsymbol{g}(\boldsymbol{x}) \le 0$ etc., and denoting by $\boldsymbol{F}, \boldsymbol{H}, \boldsymbol{G}$ the objective and constraint Jacobians, the linear problem (2) is equivalent to

$$\max_{\boldsymbol{d},\beta^-} \begin{bmatrix}\boldsymbol{0}_n^{\mathsf{T}} & 1\end{bmatrix} \begin{bmatrix}\boldsymbol{d}\\ \beta^-\end{bmatrix} \quad \text{s.t.} \quad \boldsymbol{d} \in \mathbb{R}^n, \; \beta^- \in \mathbb{R}, \; \begin{bmatrix} -\boldsymbol{I}_{n,n} & \boldsymbol{0}_n \\ \boldsymbol{I}_{n,n} & \boldsymbol{0}_n \\ \boldsymbol{F} & \boldsymbol{1}_K \\ \boldsymbol{H} & \boldsymbol{0}_M \\ \boldsymbol{G} & \boldsymbol{0}_P \end{bmatrix} \begin{bmatrix}\boldsymbol{d}\\ \beta^-\end{bmatrix} \begin{matrix} \le \\ \le \\ \le \\ = \\ \le \end{matrix} \begin{bmatrix} \boldsymbol{1}_n \\ \boldsymbol{1}_n \\ \boldsymbol{0}_K \\ \boldsymbol{0}_M \\ -\boldsymbol{g} \end{bmatrix}. \tag{P}$$

Consider also the dual problem:

$$\min_{\boldsymbol{y}^1,\dots,\boldsymbol{y}^5} \begin{bmatrix}\mathbf{1}_n^\top & \mathbf{1}_n^\top & \mathbf{0}_K^\top & \mathbf{0}_M^\top & -\boldsymbol{g}^\top\end{bmatrix}\begin{bmatrix}\boldsymbol{y}^1\\ \vdots \\ \boldsymbol{y}^5\end{bmatrix} \quad \text{s.t.} \quad \boldsymbol{y}^1 \ge \mathbf{0}_n, \boldsymbol{y}^2 \ge \mathbf{0}_n, \boldsymbol{y}^3 \ge \mathbf{0}_K, \boldsymbol{y}^4 \in \mathbb{R}^M, \boldsymbol{y}^5 \ge \mathbf{0}_P \quad \text{and}$$
$$\begin{bmatrix}-\boldsymbol{I}_{n,n} & \boldsymbol{I}_{n,n} & \boldsymbol{F}^\top & \boldsymbol{H}^\top & \boldsymbol{G}^\top \\ \mathbf{0}_n^\top & \mathbf{0}_n^\top & \mathbf{1}_K^\top & \mathbf{0}_M^\top & \mathbf{0}_P^\top\end{bmatrix}\begin{bmatrix}\boldsymbol{y}^1\\ \vdots \\ \boldsymbol{y}^5\end{bmatrix} = \begin{bmatrix}\mathbf{0}_n \\ 1\end{bmatrix}. \tag{D}$$

If 0 is the optimal value of (P), then $\beta^- = 0$. By strong duality, the dual problem is feasible with optimal value 0, implying $\boldsymbol{y}^1 = \mathbf{0}$, $\boldsymbol{y}^2 = \mathbf{0}$ and $-\boldsymbol{g}^\top \boldsymbol{y}^5 = 0$. The KKT equations immediately follow from the remaining constraints and from the complementary slackness property of dual solution pairs. ◀

2 Trust-Region Concepts and Surrogates

As mentioned in the previous section, we assume (at least some) of the objectives or constraints to be computationally expensive. To approximate a Pareto-critical point whilst avoiding expensive function evaluations, a trust-region approach is used: The true functions $\boldsymbol{f}, \boldsymbol{h}$ and $\boldsymbol{g}$ are modeled by $\widehat{\boldsymbol{f}}^{(k)}, \widehat{\boldsymbol{h}}^{(k)}$ and $\widehat{\boldsymbol{g}}^{(k)}$ respectively. These models are constructed to be sufficiently accurate within iteration-dependent trust-regions

$$B^{(k)} = B(\boldsymbol{x}^{(k)}; \Delta_{(k)}) := \left\{\boldsymbol{x} \in \mathbb{R}^n : \left\|\boldsymbol{x} - \boldsymbol{x}^{(k)}\right\|_{\mathrm{tr,k}} \le \Delta_{(k)}\right\}. \tag{3}$$

Using the surrogate models, a step $\boldsymbol{s}^{(k)}$ and a step-size $\sigma_{(k)}$ are determined, in such a way as to reduce the constraint violation or achieve an objective value reduction at the trial point $\boldsymbol{x}^{(k)} + \sigma_{(k)}\boldsymbol{s}^{(k)}$, which is tested as a candidate for the next iterate. The step $\sigma_{(k)}\boldsymbol{s}^{(k)}$ is computed so that the trial point is contained in both the trust-region and the approximate linearized feasible set at $\boldsymbol{x}^{(k)}$:

$$\mathcal{L}_k = \mathcal{L}_k(\boldsymbol{x}^{(k)}) = \left\{\boldsymbol{x}^{(k)} + \boldsymbol{s} \in \mathbb{R}^n : \widehat{\boldsymbol{h}}^{(k)}(\boldsymbol{x}^{(k)}) + \widehat{\boldsymbol{H}}_k(\boldsymbol{x}^{(k)}) \cdot \boldsymbol{s} = \mathbf{0},\ \widehat{\boldsymbol{g}}^{(k)}(\boldsymbol{x}^{(k)}) + \widehat{\boldsymbol{G}}_k(\boldsymbol{x}^{(k)}) \cdot \boldsymbol{s} \le \mathbf{0}\right\}, \tag{4}$$

where $\widehat{\boldsymbol{H}}_k$ and $\widehat{\boldsymbol{G}}_k$ are now the full Jacobians of the constraint function models. This introduces uncertainty and iterates might no longer be feasible. Hence, we treat the constraints as *relaxable*, and we might have to evaluate $\boldsymbol{f}$ (and $\boldsymbol{g}$ and $\boldsymbol{h}$) outside of $\mathcal{X}$, which motivates the next assumption. It bounds the trust-regions and ensures that all functions are available for all possible iterates, trust-region norms and radii. Note, that the algorithm actually can use two trust-region sizes, the preliminary size $\overline{\Delta}_{(k)}$ at the beginning of an iteration and $\Delta_{(k)} \le \overline{\Delta}_{(k)}$ after the *Criticality Routine*, which is described in Section 3.1.

▶ **Assumption 5.** *There is a constant $0 < \Delta_{\max} < \infty$ such that for every $k \in \mathbb{N}_0$ the trust region sizes conform to $0 < \Delta_{(k)} \le \overline{\Delta}_{(k)} < \Delta_{\max}$. All functions (the true functions and their models) are defined in these regions, i.e., on the set $\mathcal{C}(\mathcal{X}) = \bigcup_{k\in\mathbb{N}_0} B\big(\boldsymbol{x}^{(k)}; \overline{\Delta}_{(k)}\big)$.*

Note, that the trust-regions in (3) may be defined using iteration-dependent norms $\|\bullet\|_{\mathrm{tr,k}}$. Later, we will also introduce norm-constrained subproblems, where the norm $\|\bullet\|_k$ can vary, too. This allows for great flexibility in the actual implementation. We only require that the norms be uniformly equivalent to the Euclidean norm:

▶ **Assumption 6.** *There is a constant $\mathsf{c} \ge 1$ such that for all $\boldsymbol{x} \in \mathcal{C}(\mathcal{X})$ and all $k \in \mathbb{N}_0$ and $\|\bullet\|_* = \|\bullet\|_{\mathrm{tr,k}}$ or $\|\bullet\|_* = \|\bullet\|_k$ it holds that*

$$\frac{1}{\mathsf{c}}\|\boldsymbol{x}\|_* \le \|\boldsymbol{x}\|_2 \le \mathsf{c}\|\boldsymbol{x}\|_*. \tag{5}$$

▶ Remark 7. Any two norms that are uniformly equivalent to $\|\bullet\|_2$ with constant c are pairwise equivalent with constants $1/\mathsf{c}^2$ and c.

2.1 Fully Linear Models

In this subsection, we first want to explain what it means for the model functions to be sufficiently accurate. Although the true derivative information is not used in any computation, we will need it for the convergence analysis and hence require the following generalization of Assumption 2:

▶ **Assumption 8.** *The objective functions* $f_\mathfrak{l}$, $\mathfrak{l} = 1, \dots, K$, *the constraint functions* $g_\mathfrak{i}$, $\mathfrak{i} = 1, \dots, P$, *and* $h_\mathfrak{j}$, $\mathfrak{j} = 1, \dots, M$, *are twice continuously differentiable in an open domain containing* $\mathcal{C}(\mathcal{X})$ *and have Lipschitz continuous gradients on* $\mathcal{C}(\mathcal{X})$.

With Assumption 8 we can use [16, Definition 6.1.] to have our models satisfy error bounds w.r.t. to their construction radius:

▶ **Definition 9** (Fully Linear Models). *Let* $\Delta_{\max} > 0$ *be a given constant and let* $f\colon \mathbb{R}^n \to \mathbb{R}$ *be a scalar-valued function satisfying Assumption 8. A set of model functions* $\mathcal{M} = \{\hat{f}\colon \mathbb{R}^n \to \mathbb{R}\}$ *is called a fully linear class of models for* f *if the following hold:*

1. *There are positive constants* $\mathsf{e}, \dot{\mathsf{e}}$ *and* $L^{\hat{f}}$ *such that for any* $\Delta \in (0, \Delta_{\max}]$ *and for any* $\boldsymbol{x} \in \mathcal{C}(\mathcal{X})$ *there is a model function* $\hat{f} \in \mathcal{M}$, *with Lipschitz continuous gradient and corresponding Lipschitz constant bounded by* $L^{\hat{f}}$, *such that the error between the gradients and the error between the values satisfy*

$$\left\|\nabla f(\boldsymbol{\xi}) - \nabla \hat{f}(\boldsymbol{\xi})\right\|_2 \le \dot{\mathsf{e}}\Delta \quad \textit{and} \quad \left|f(\boldsymbol{\xi}) - \hat{f}(\boldsymbol{\xi})\right| \le \mathsf{e}\Delta^2, \qquad \forall\, \boldsymbol{\xi} \in B(\boldsymbol{x}; \Delta) \cap \mathcal{C}(\mathcal{X}).$$

2. *For this class* $\mathcal{M}$ *there exists a "model-improvement" algorithm that, in a finite, uniformly bounded number of steps can either establish that a given model* $\hat{f} \in \mathcal{M}$ *is fully linear on* $B(\boldsymbol{x}; \Delta)$ *or find a model* $\hat{f} \in \mathcal{M}$ *that is fully linear on* $B(\boldsymbol{x}; \Delta)$.

▶ **Definition 10.** *Let* $\Delta_{\max} > 0$ *be a given constant and let* $\boldsymbol{f} = [f_1, \dots, f_K]^\mathsf{T}$ *be a vector of* $K \ge 1$ *functions satisfying the requirements of Theorem 9 with classes* $\mathcal{M}_\mathfrak{l}$ *and constants* $(\mathsf{e}_\mathfrak{l}, \dot{\mathsf{e}}_\mathfrak{l}, L^{\hat{f}}_\mathfrak{l})$, $\mathfrak{l} = 1, \dots, K$. *Then*

$$\mathcal{M} = \{\hat{\boldsymbol{f}} = [\hat{f}_1, \dots, \hat{f}_K]^\mathsf{T} : \hat{f}_1 \in \mathcal{M}_1,\ \dots,\ \hat{f}_K \in \mathcal{M}_K\}$$

is a class of fully linear, vector-valued model functions with constants $\max_\mathfrak{l} \mathsf{e}_\mathfrak{l} > 0$, $\max_\mathfrak{l} \dot{\mathsf{e}}_\mathfrak{l} > 0$ *and* $\max_\mathfrak{l} L^{\hat{f}}_\mathfrak{l} > 0$. *A vector of functions* $\hat{\boldsymbol{f}} \in \mathcal{M}$ *is deemed fully linear, if all components* $\hat{f}_\mathfrak{l}$ *are fully linear. The improvement algorithms for* $\mathcal{M}_\mathfrak{l}$ *are applied component-wise.*

▶ Remark 11. In trust-region algorithms with derivatives, oftentimes Taylor polynomial approximations are used, and under standard assumptions they satisfy the error bounds. In fact, the error bounds are motivated by these traditional models. In contrast, the notion of fully linear models allows for more flexible surrogates, the construction of which usually depends not only on the iterate but also the current trust region radius.

A function that is fully linear for $\Delta_{(k)} > 0$ is automatically fully linear for any smaller radius $0 < \Delta < \Delta_{(k)}$. When the trust-region radius is bounded above, then the constants $\mathsf{e} > 0$ and $\dot{\mathsf{e}} > 0$ can be chosen large enough such that a fully linear model stays fully linear in enlarged trust-regions:

▶ **Lemma 12** ([16, Lemma 10.25]). *For* $\boldsymbol{x} \in \mathcal{C}(\mathcal{X})$ *and* $\Delta \in (0, \Delta_{\max}]$ *consider a function* f *and a fully linear model* $\hat{f}$ *of* f *with constants* $\mathsf{e}, \dot{\mathsf{e}}, L^{\hat{f}} > 0$. *Let* $L^f > 0$ *be a Lipschitz constant of* ∇f. *Assume without loss of generality (w.l.o.g.) that*

$$L^{\hat{f}} + L^f \le \mathsf{e} \textit{ and } \dot{\mathsf{e}}/2 \le \mathsf{e}. \tag{6}$$

Then $\hat{f}$ *is fully linear on* $B\big(\boldsymbol{x}; \tilde{\Delta}\big)$ *for any* $\tilde{\Delta} \in [\Delta, \Delta_{\max}]$ *with the same constants.*

▶ **Assumption 13.** *For any* $k \in \mathbb{N}_0$ *the models* $\hat{\boldsymbol{f}}^{(k)}$ *are fully linear on* $B\big(\boldsymbol{x}^{(k)}; \Delta_{(k)}\big)$ *as in Theorem 10 with constants* $\mathsf{e}_f, \dot{\mathsf{e}}_f$ *and* $L^{\boldsymbol{f}}$ *that are chosen large enough such that* (6) *is fulfilled globally. The same holds for the models* $\hat{\boldsymbol{g}}^{(k)}$ *of* $\boldsymbol{g}$ *with constants* $\mathsf{e}_g, \dot{\mathsf{e}}_g, L^{\boldsymbol{g}}$ *and the models* $\hat{\boldsymbol{h}}^{(k)}$ *of* $\boldsymbol{h}$ *with constants* $\mathsf{e}_h, \dot{\mathsf{e}}_h$ *and* $L^{\boldsymbol{h}}$. *We also assume that all models are* interpolating *at* $\boldsymbol{x}^{(k)}$.

2.2 Composite Step Approach and Sub-Problems

Just like in many previous articles on multi-objective trust-region algorithms [6, 64, 69, 70], we use the maximum-scalarization

$$\Phi(\boldsymbol{x}) = \Phi\big[\boldsymbol{f}\big](\boldsymbol{x}) := \max_{\mathfrak{l}=1,\dots,K} f_\mathfrak{l}(\boldsymbol{x}) \quad \text{and} \quad \Phi^{(k)}(\boldsymbol{x}) = \Phi\big[\hat{\boldsymbol{f}}^{(k)}\big](\boldsymbol{x}) := \max_{\mathfrak{l}=1,\dots,K} \hat{f}_\mathfrak{l}(\boldsymbol{x})$$

to determine objective reduction and prove convergence. The idea then is to find a step $\boldsymbol{s}^{(k)}$ that approximately solves

$$\min_{\boldsymbol{s}\in\mathbb{R}^n} \Phi\big[\widehat{\boldsymbol{f}}^{(k)}\big](\boldsymbol{x}^{(k)} + \boldsymbol{s}^{(k)}) \quad \text{s.t.} \quad \boldsymbol{s} \in \Big(\mathcal{L}_k - \boldsymbol{x}^{(k)}\Big),\ \|\boldsymbol{s}\|_{\mathrm{tr,k}} \le \Delta_{(k)},$$

with the inexact linearized feasible set defined in (4). Without constraints, inexact line-search can be used. In our case, we use composite-step approach and $\boldsymbol{s}^{(k)}$ is split into a normal component $\boldsymbol{n}^{(k)}$ towards feasibility and a descent direction $\boldsymbol{d}^{(k)}$ (see [32, 72] for details). Then, the normal step can be computed with

$$\min \|\boldsymbol{n}\|_2^2 \quad \text{s.t.} \quad \boldsymbol{n} \in \Big(\mathcal{L}_k - \boldsymbol{x}^{(k)}\Big). \tag{ITRN$^{(k)}$}$$

If a normal step $\boldsymbol{n}^{(k)}$ has been found (and if $\|\boldsymbol{n}\|_{\mathrm{tr,k}} \le \overline{\Delta}_{(k)}$) the descent direction $\boldsymbol{d}^{(k)}$ can be taken as the minimizer of

$$\omega\Big(\boldsymbol{x}_{\mathrm{n}}^{(k)}; \widehat{\boldsymbol{f}}^{(k)}, \mathcal{L}_k, \|\bullet\|_k\Big) = -\min_{\beta\in\mathbb{R},\boldsymbol{d}\in\mathbb{R}^n} \beta \quad \text{s.t.} \quad \widehat{\boldsymbol{F}}_k(\boldsymbol{x}_{\mathrm{n}}^{(k)}) \cdot \boldsymbol{d} \le \beta,\ \boldsymbol{d} \in \Big(\mathcal{L}_k - \boldsymbol{x}_{\mathrm{n}}^{(k)}\Big),\ \|\boldsymbol{d}\|_k \le 1, \tag{ITRT$^{(k)}$}$$

where $\mathcal{L}_k$ is the approximate linearized feasible set at $\boldsymbol{x}^{(k)}$ according to (4) and $\boldsymbol{x}_{\mathrm{n}}^{(k)}$ is the point $\boldsymbol{x}_{\mathrm{n}}^{(k)} = \boldsymbol{x}^{(k)} + \boldsymbol{n}^{(k)}$. We actually only compute the descent direction if there is enough movement possible after performing the normal step, i.e., the normal step must not be too large. We call such a step *compatible*, and it is defined with respect to the preliminary radius $\overline{\Delta}_{(k)}$:

▶ **Definition 14.** *Let* $\mathsf{c}_\Delta \in (0,1], \mathsf{c}_\mu > 0$ *and* $\mu \in (0,1)$ *be constants. The minimizer* $\boldsymbol{n}^{(k)}$ *of* (ITRN$^{(k)}$) *is called compatible (w.r.t.* $\overline{\Delta}_{(k)}$*) if*

$$\Big\|\boldsymbol{n}^{(k)}\Big\|_{\mathrm{tr,k}} \le \mathsf{c}_\Delta \overline{\Delta}_{(k)} \min\Big\{1, \mathsf{c}_\mu \overline{\Delta}_{(k)}^{\mu}\Big\}. \tag{7}$$

Similarly, we call (ITRN$^{(k)}$) *compatible if it has a non-empty solution-set and a minimizer that is compatible.*

Furthermore, we assume a normal step to exist if the true constraint violation is not too large, as measured by the infeasibility function

$$\theta(\boldsymbol{x}) := \max\Bigg\{0, \max_{\mathrm{j}=1,\dots,M} |h_{\mathrm{j}}(\boldsymbol{x})|, \max_{\mathrm{i}=1,\dots,P} g_{\mathrm{i}}(\boldsymbol{x})\Bigg\}.$$

▶ **Assumption 15** (Existence and Boundedness of Normal Step). *If* $\theta_k = \theta\big(\boldsymbol{x}^{(k)}\big) \le \delta_{\mathrm{n}}$*, for a constant* $\delta_{\mathrm{n}} > 0$*, then* $\boldsymbol{n}^{(k)}$ *exists and there is a constant* $\mathsf{c}_{\mathrm{ubn}} > 0$ *such that*

$$\Big\|\boldsymbol{n}^{(k)}\Big\|_{\mathrm{tr,k}} \le \mathsf{c}_{\mathrm{ubn}}\theta_k. \tag{8}$$

Assumption 15 is a standard assumption and the reasoning behind it can be found in [32]. Finally, the step-size $\sigma_{(k)}$ is determined in such a way that $\big\|\boldsymbol{x}^{(k)} + \boldsymbol{n}^{(k)} + \sigma_{(k)}\boldsymbol{d}^{(k)}\big\|_{\mathrm{tr,k}} \le \Delta_{(k)}$ and a *sufficient decrease condition* (for the objective surrogates) is satisfied, which is described and justified in Section 2.4. The term $\sigma_{(k)}\boldsymbol{d}^{(k)}$ is also called *tangential step* and altogether the step $\boldsymbol{s}^{(k)}$ results in the *trial point* $\boldsymbol{x}_{\mathrm{s}}^{(k)} = \boldsymbol{x}^{(k)} + \boldsymbol{s}^{(k)} = \boldsymbol{x}_{\mathrm{n}}^{(k)} + \sigma_{(k)}\boldsymbol{d}^{(k)}$.

2.3 Filter Mechanism

As mentioned already, iterates can become infeasible. We employ a so-called *filter* to drive them back towards the feasible set. Thus, the trial point $\boldsymbol{x}_{\mathrm{s}}^{(k)} = \boldsymbol{x}^{(k)} + \boldsymbol{s}^{(k)}$ is tested not only for actual decrease of the original functions but also against previous iterates stored in the filter $\mathcal{F}$. $\mathcal{F}$ is a set of tuples $(\theta_j, \Phi_j), j = 1, \dots, |\mathcal{F}|$, describing a forbidden area in image space. In fact, the tuples in $\mathcal{F}$ (w.r.t. $\boldsymbol{f}$) are currently non-dominated for the bi-objective optimization problem of minimizing both $\theta(\boldsymbol{x})$ and $\Phi\big[\boldsymbol{f}\big](\boldsymbol{x})$. The trial point $\boldsymbol{x}_{\mathrm{s}}^{(k)}$ is only acceptable for $\mathcal{F}$ if its value tuple is also (sufficiently) non-dominated. An acceptable trial point is further tested, and if it sufficiently reduces the *true* objectives it is kept as the next iterate. The current iterate might be added to the filter if the predicted objective decrease is small compared to the constraint violation, or if the latter is too large. More formally, we use the following definition:

▶ **Definition 16** (Multi-Objective Filter). *A filter $\mathcal{F}$ with constant $\gamma_\theta \in (0,1)$ for a function $\boldsymbol{f}\colon \mathbb{R}^n \to \mathbb{R}^K$ is a discrete set of tuples $\{(\theta_j, \Phi_j)\}_j \subset \mathbb{R}^2$ such that for all j the tuple $(\theta_j, \Phi_j) \in \mathcal{F}$ is acceptable for $\mathcal{F} \setminus \{(\theta_j, \Phi_j)\}$. A tuple (θ_j, Φ_j) is* acceptable *for $\mathcal{F}$ iff*

$$\theta_j \le (1-\gamma_\theta)\theta_i \quad \text{or} \quad \Phi_j \le \Phi_i - \gamma_\theta \theta_i \qquad \forall\, (\theta_i, \Phi_i) \in \mathcal{F}.$$

If a tuple (θ_j, Φ_j) is added to $\mathcal{F}$, then all redundant entries are removed, i.e., all tuples (θ_i, Φ_i) with

$$\theta_i \ge \theta_j \quad \text{and} \quad \Phi_i - \gamma_\theta\theta_i \ge \Phi_j - \gamma_\theta \theta_j.$$

A point *$\boldsymbol{x}$ is acceptable for the filter $\mathcal{F}$ iff the value tuple $(\theta(\boldsymbol{x}), \Phi[\boldsymbol{f}](\boldsymbol{x}))$ is acceptable for $\mathcal{F}$, and a point $\boldsymbol{x}$ is added to $\mathcal{F}$ by adding its value tuple.*

As can be seen from Theorem 16, a filter strengthens non-dominance testing by employing a positive offset $\gamma_\theta\theta_j$. Like with the extended definition of (weak) dominance in [34], feasible and infeasible points can be compared (in contrast to the alternative definition in [8]). Note, that we use a 2-dimensional filter in accordance with the maximum-scalarization. This makes the convergence analysis easier. It should, however, be possible to use a stricter, $(K+1)$-dimensional filter (similar to [41]) by using $\boldsymbol{f}(\bullet)$ instead of $\Phi[\boldsymbol{f}](\bullet)$. In Section 8.3, there is a simple draft of such a procedure. Likely, this requires further modifications to the algorithm, e.g., when testing if a step reduces the objective values to a satisfactory degree. For now, we stick to the procedure given in the next section.

2.4 Sufficient Decrease

In this subsection we want to explain what is meant by "sufficient decrease". In short, we have to relate the model predicted objective reduction to the criticality value. The criticality value is the negative optimal value of $(\text{ITRT}^{(k)})$, and it is thus dependent on the norm, which we are allowed to choose in a way best suited for the problem geometry or the inner solver(s). At this point, the importance of Assumption 6 becomes clear. It allows us to relate criticality values for different norms.

▶ **Lemma 17.** *Suppose Assumptions 5, 6, 8 and 13 hold. Denote by $\widehat{\omega}^{(k)} := \omega(\boldsymbol{x}_{\mathrm{n}}^{(k)}; \widehat{\boldsymbol{f}}^{(k)}, \mathcal{L}_k, \|\bullet\|_k)$ the optimal value of $(\text{ITRT}^{(k)})$ and by $\widehat{\omega}_2^{(k)}$ the optimal value if the 2-norm is used, $\widehat{\omega}_2^{(k)} = \omega(\boldsymbol{x}_{\mathrm{n}}^{(k)}; \widehat{\boldsymbol{f}}^{(k)}, \mathcal{L}_k, \|\bullet\|_2)$. There is a constant $\mathtt{w} \ge 1$ such that for any $k \in \mathbb{N}_0$ for which the normal step exists, it holds that*

$$\frac{1}{\mathtt{w}}\, \widehat{\omega}^{(k)} \le \widehat{\omega}_2^{(k)} \le \mathtt{w}\, \widehat{\omega}^{(k)}. \tag{9}$$

The proof of Theorem 17 can be found in the appendix. The lemma shows that we can relate the iteration dependent inexact critical values $\widehat{\omega}^{(k)}$ and $\widehat{\omega}_2^{(k)}$. Furthermore, we show in the appendix that it is then sensible to assume a sufficient decrease condition for the objective surrogate functions as per Assumption 19, as long as an additional (standard) assumption holds:

▶ **Assumption 18.** *The norm of all model Hessians of the objective function surrogates is uniformly bounded above, i.e., there is a positive constant $\mathtt{H} > 0$ such that for all $k \in \mathbb{N}_0$*

$$\left\| \nabla^2 \widehat{f}_{\mathfrak{l}}^{(k)}(\boldsymbol{\xi}) \right\|_2 \le \mathtt{H} \text{ for all } \mathfrak{l} = 1, \dots, K, \text{ and all } \boldsymbol{\xi} \in \mathcal{L}_k(\boldsymbol{x}^{(k)}) \cap B^{(k)}.$$

We know how to find suitable model functions that satisfy Assumption 18, including Lagrange interpolation polynomials or RBFs, so that finally we can state the sufficient decrease assumption:

▶ **Assumption 19** (Sufficient Decrease). *Suppose Assumptions 5, 8 and 13 hold and that $\Delta_{(k)} \in (0, \Delta_{\max})$. Let $\Phi^{(k)} := \Phi[\widehat{\boldsymbol{f}}^{(k)}]$. If $\boldsymbol{n}^{(k)}$ is compatible, $\boldsymbol{d}^{(k)}$ is a minimizer of $(\text{ITRT}^{(k)})$ at $\boldsymbol{x}^{(k)}$ for $\|\bullet\|_k$, and $\widehat{\omega}_2^{(k)}$ is the sign-flipped optimal value from $(\text{ITRT}^{(k)})$ using the 2-norm, then there is a step-length $\sigma_{(k)} \ge 0$ such that $\boldsymbol{x}^{(k)} + \boldsymbol{n}^{(k)} + \sigma_{(k)}\boldsymbol{d}^{(k)} \in \mathcal{L}_k \cap B^{(k)}$ and*

$$\Phi^{(k)}\left(\boldsymbol{x}_{\mathrm{n}}^{(k)}\right) - \Phi^{(k)}\left(\boldsymbol{x}_{\mathrm{n}}^{(k)} + \sigma_{(k)}\boldsymbol{d}^{(k)}\right) \ge \mathtt{c}_{\mathrm{sd}} \widehat{\omega}_2^{(k)} \min\left\{ \frac{\widehat{\omega}_2^{(k)}}{\mathtt{W}}, \Delta_{(k)}, 1 \right\}, \tag{10}$$

for constants $\mathtt{c}_{\mathrm{sd}} \in (0,1)$ and $\mathtt{W} \ge 1$.

Throughout the rest of this article we use a slightly modified measure for notational convenience:

▶ **Corollary 20** (Modified Criticality Measure). *For any $k \in \mathbb{N}_0$ and $\widehat{\omega}^{(k)}$ as in Theorem 17, define the criticality measure $\chi^{(k)} := \min\{1, \widehat{\omega}^{(k)}\}$ and denote by $\chi_2^{(k)}$ the corresponding value, if the 2-norm is used instead of $\|\bullet\|_k$. Then $\lim_{k\to\infty} \widehat{\omega}_2^{(k)} = 0$ if and only if $\lim_{k\to\infty} \chi_2^{(k)} = 0$, and if Assumption 19 holds, then it also follows (with $\mathtt{W} \geq 1$) that*

$$\Phi^{(k)}\left(\boldsymbol{x}_{\mathrm{n}}^{(k)}\right) - \Phi^{(k)}\left(\boldsymbol{x}_{\mathrm{n}}^{(k)} + \sigma_{(k)}\boldsymbol{d}^{(k)}\right) \geq \mathtt{c}_{\mathrm{sd}}\chi_2^{(k)} \min\left\{\frac{\chi_2^{(k)}}{\mathtt{W}}, \Delta_{(k)}\right\}.$$

3 Discussion of the Algorithm

The behavior of the algorithm is determined by several additional algorithmic parameters:

Parameter(s)	Description
$0 < \Delta_{(0)} \leq \Delta_{\max} < \infty$	initial and maximum trust-region radius
$0 < \gamma_0 \leq \gamma_1 < 1 \leq \gamma_2$	shrinking and growing parameters for the trust-regions
$\gamma_\theta \in (0,1)$	filter constant in Theorem 16
$0 < \nu_0 \leq \nu_1 < 1$	acceptance thresholds for trial point test
$0 < \varepsilon_\chi < 1,\ 0 \leq \varepsilon_\theta \leq \delta_{\mathrm{n}}$	thresholds for criticality test
$\kappa_\theta \in (0,1), \psi > \frac{1}{1+\mu}$	threshold parameters in (11)
$0 < \mathtt{B} < \mathtt{M}$ and $\alpha \in (0,1)$	Criticality Routine threshold factors and backtracking constant
$\mathtt{c}_{\mathrm{sd}} \in (0,1)$	*s*ufficient *d*ecrease constant in Assumption 19
$\mathtt{c}_\Delta \in (0,1), \mathtt{c}_\mu > 0, \mu \in (0,1)$	constants defining compatibility in Theorem 14
$\delta_{\mathrm{n}} > 0,\ \mathtt{c}_{\mathrm{ubn}} > 0$	existence of normal step in Assumption 15

Below, the *Criticality Test* is used instead of the intuitive stopping criterion " $\omega\left(\boldsymbol{x}_{\mathrm{n}}^{(k)}; \widehat{\boldsymbol{f}}^{(k)}, \mathcal{L}_k, \|\bullet\|_k\right) = 0$ " or "$\chi^{(k)} = 0$" because the surrogate models are inexact. If an iterate is nearly feasible and nearly critical for the surrogate problem, then the criticality routine is entered and the trust-region radius is reduced to make the models more precise. At a truly critical point the routine loops infinitely, but if a point is not critical for the true functions, we exit and continue with regular iterations. Our version of the routine is a bit different from the "unrolled" variant in [25]. Whilst their version is simpler, we avoid restoration in criticality loops. For further details of the Criticality Test and the Criticality Routine we refer to [16].

Algorithm

0. **Initialization:** Let $k \leftarrow 0$, $\mathcal{F} \leftarrow \emptyset$ and $\boldsymbol{n}^{(k)} = \mathtt{undef}$. Evaluate $\boldsymbol{f}(\boldsymbol{x}^{(k)}), \boldsymbol{g}(\boldsymbol{x}^{(k)}), \boldsymbol{h}(\boldsymbol{x}^{(k)})$ and compute $\theta_k = \theta(\boldsymbol{x})$. Build surrogate models $\widehat{\boldsymbol{f}}^{(k)}, \widehat{\boldsymbol{g}}^{(k)}, \widehat{\boldsymbol{h}}^{(k)}$ that are fully linear in $B^{(k)}$ with radius $\overline{\Delta}_{(k)} := \Delta_{(0)}$ and set $\overline{\chi}^{(k)} = \mathtt{undef}$.
1. **Compatibility Test:**
   - If $\boldsymbol{n}^{(k)} \neq \mathtt{undef}$ and $\boldsymbol{n}^{(k)}$ is compatible w.r.t. $\overline{\Delta}_{(k)}$, go to step 3.
   - If $\boldsymbol{n}^{(k)} = \mathtt{undef}$, try to compute $\boldsymbol{n}^{(k)}$. If $\boldsymbol{n}^{(k)}$ exists and is compatible w.r.t. $\overline{\Delta}_{(k)}$, go to step 3.
2. **Restoration:** Add $\boldsymbol{x}^{(k)}$ to the filter, set $\Delta_{(k)} \leftarrow \overline{\Delta}_{(k)}, \chi^{(k)} \leftarrow \overline{\chi}^{(k)}$, and attempt to find a *restoration step* $\boldsymbol{r}^{(k)}$ and $\overline{\Delta}_{(k+1)} > 0$ for which there exists a $\overline{\Delta}_{(k+1)}$-compatible normal step at $\boldsymbol{x}^{(k)} + \boldsymbol{r}^{(k)}$, and for which $\boldsymbol{x}^{(k)} + \boldsymbol{r}^{(k)}$ is acceptable for $\mathcal{F}$. Set $\boldsymbol{x}^{(k+1)} \leftarrow \boldsymbol{x}^{(k)} + \boldsymbol{r}^{(k)}$, keep $\overline{\Delta}_{(k+1)}$ and go to step 8.
3. **Descent Step:** Compute a descent direction $\boldsymbol{d}^{(k)}$ and $\overline{\chi}^{(k)}$ with (ITRT$^{(k)}$) and Theorem 20.
4. **Criticality Test:** If $\theta_k < \varepsilon_\theta$ **and** $\left(\overline{\chi}^{(k)} < \varepsilon_\chi \text{ and } \overline{\Delta}_{(k)} > \mathtt{M}\overline{\chi}^{(k)}\right)$, then enter the **Criticality Routine** to get $\Delta_{(k)}$ and $\chi^{(k)}$ and update $\boldsymbol{d}^{(k)}$ and $\boldsymbol{n}^{(k)}$. Else, set $\Delta_{(k)} \leftarrow \overline{\Delta}_{(k)}$ and $\chi^{(k)} \leftarrow \overline{\chi}^{(k)}$.
5. **Acceptance Test:** Compute a step-size $\sigma_{(k)} > 0$ such that Assumption 19 is fulfilled. Set $\boldsymbol{x}_{\mathrm{s}}^{(k)} = \boldsymbol{x}^{(k)} + \boldsymbol{n}^{(k)} + \sigma_{(k)}\boldsymbol{d}^{(k)}$. Compute $\boldsymbol{f}(\boldsymbol{x}_{\mathrm{s}}^{(k)})$ and $\theta\left(\boldsymbol{x}_{\mathrm{s}}^{(k)}\right)$.
   - If $\boldsymbol{x}_{\mathrm{s}}^{(k)}$ is not acceptable for the augmented filter $\mathcal{F} \cup \{(\theta_k, \Phi[\boldsymbol{f}](\boldsymbol{x}^{(k)}))\}$ OR
   - If

$$\Phi[\widehat{\boldsymbol{f}}^{(k)}](\boldsymbol{x}^{(k)}) - \Phi[\widehat{\boldsymbol{f}}^{(k)}](\boldsymbol{x}_{\mathrm{s}}^{(k)}) \geq \kappa_\theta \theta_k^\psi \tag{11}$$

AND

$$\rho_{(k)} := \frac{\Phi[\boldsymbol{f}](\boldsymbol{x}^{(k)}) - \Phi[\boldsymbol{f}](\boldsymbol{x}_{\mathrm{s}}^{(k)})}{\Phi[\widehat{\boldsymbol{f}}^{(k)}](\boldsymbol{x}^{(k)}) - \Phi[\widehat{\boldsymbol{f}}^{(k)}](\boldsymbol{x}_{\mathrm{s}}^{(k)})} < \nu_0, \tag{12}$$

keep $\boldsymbol{x}^{(k+1)} \leftarrow \boldsymbol{x}^{(k)}, \boldsymbol{n}^{(k+1)} \leftarrow \boldsymbol{n}^{(k)}$, choose $\overline{\Delta}_{(k+1)} \in [\gamma_0\Delta_{(k)}, \gamma_1\Delta_{(k)}]$, increment k and go to step 8.

6. **Filter Test:** If (11) fails, include $\boldsymbol{x}^{(k)}$ in the filter (θ -iteration).
7. **Iterate Updates:** Set $\boldsymbol{x}^{(k+1)} \leftarrow \boldsymbol{x}_{\mathrm{s}}^{(k)}$ and choose

$$\overline{\Delta}_{(k+1)} \in \begin{cases} [\gamma_0\Delta_{(k)}, \gamma_1\Delta_{(k)}] & \text{if } \rho_{(k)} < \nu_1, \\ [\Delta_{(k)}, \min\{\gamma_2\Delta_{(k)}, \Delta_{\max}\}] & \text{if } \rho_{(k)} \geq \nu_1. \end{cases} \tag{13}$$

8. **Model Updates:** Update the surrogates for $\boldsymbol{x}^{(k+1)}$ and $\overline{\Delta}_{(k+1)}$. If any of the constraint function models is dependent on the trust-region radius (see Theorem 11), then set $\boldsymbol{n}^{(k+1)} \leftarrow \texttt{undef}$ and $\overline{\chi}^{(k+1)} \leftarrow \texttt{undef}$. Finally, let $k \leftarrow k+1$ and go to step 1.

Note, that in contrast to regular trust-region methods, a trial point that does not pass the test (12) may still be accepted as the next iterate, namely in a θ-iteration. The restoration step $\boldsymbol{x}^{(k+1)} = \boldsymbol{x}^{(k)} + \boldsymbol{r}^{(k)}$ aims at reducing the constraint violation and is usually calculated by solving

$$\min_{\boldsymbol{x}} \theta(\boldsymbol{x}). \tag{R}$$

For a detailed discussion of the role of (11) and the restoration procedure, we refer to [32]. We only want to note a few important properties:

- No feasible iterate is ever added to the filter.
- Unless noted otherwise, we assume that the restoration procedure is always able to find a suitable restoration step.
- There are no two successive restoration iterations.
- Under Assumption 15, there can only ever be a finite number of sub-iterations in the restoration procedure until a compatible normal step exists.

3.1 Criticality Routine

The criticality routine is provided with the current models $\widehat{\boldsymbol{f}}^{(k)}, \widehat{\boldsymbol{g}}^{(k)}, \widehat{\boldsymbol{h}}^{(k)}$, the preliminary criticality value $\overline{\chi}^{(k)}$ from algorithm step 3, and the preliminary trust-region radius $\overline{\Delta}_{(k)}$. All the models as well as the normal step $\boldsymbol{n}^{(k)}$ may be modified within this sub-routine, and we set $\Delta_{(k)}$ and $\chi^{(k)}$.

When the Criticality Routine starts, we know that $\overline{\Delta}_{(k)} > \mathtt{M}\overline{\chi}^{(k)}$ and that $\boldsymbol{n}^{(k)}$ is compatible for $\overline{\Delta}_{(k)}$.

1. Let $j \leftarrow 0$ and set $\delta_j \leftarrow \overline{\Delta}_{(k)}$, $\chi_j \leftarrow \overline{\chi}^{(k)}$ and $\boldsymbol{n}_j \leftarrow \boldsymbol{n}^{(k)}$.
2. While $\delta_j > \mathtt{M}\chi_j$:
   a. Tentatively decrease the radius: $\delta_j^+ \leftarrow \alpha\delta_j = \alpha^{j+1}\overline{\Delta}_{(k)}$.
   b. Make the models fully linear for the new radius δ_j^+.
   c. Solve (ITRN$^{(k)}$) to get $\boldsymbol{n}_j^+$ for the new models and δ_j^+.
   If $\boldsymbol{n}_j^+$ is not compatible for δ_j^+, then `BREAK`.
   d. Set $j \leftarrow j+1$, $\delta_j \leftarrow \delta_{j-1}^+$ and keep $\boldsymbol{n}_j \leftarrow \boldsymbol{n}_{j-1}^+$.
   e. Solve (ITRT$^{(k)}$) and use Theorem 20 to get the criticality measure value χ_j for the new models from step 2b and for δ_j.
3. Keep the updated models as $\widehat{\boldsymbol{f}}^{(k)}, \widehat{\boldsymbol{g}}^{(k)}, \widehat{\boldsymbol{h}}^{(k)}$, set $\chi^{(k)} \leftarrow \chi_j$, $\boldsymbol{n}^{(k)} \leftarrow \boldsymbol{n}_j$ and choose

$$\Delta_{(k)} \leftarrow \min\Big\{\max\{\delta_j, \mathtt{B}\chi^{(k)}\}, \overline{\Delta}_{(k)}\Big\}. \tag{14}$$

There are two ways the routine can stop. If the radius is sufficiently small compared to the criticality, i.e., $\delta_j \leq \mathtt{M}\chi_j$, or if the next prospective normal step $\boldsymbol{n}_j^+$ is no longer compatible for the smaller radius δ_j^+. The second criterion ensures (inductively) that after the routine has stopped finitely, the normal step will be compatible for δ_j and for any radius $\Delta \in [\delta_j, \Delta_{\max}]$. Because it always holds that $\delta_j \leq \overline{\Delta}_{(k)}$ and $\delta_j \leq \max\{\delta_j, \mathtt{B}\chi^{(k)}\}$, it follows from (14) that $\boldsymbol{n}^{(k)}$ is compatible for $\Delta_{(k)}$.

One reason for the Criticality Test (and the Criticality Routine) is to have a lower bound on the inexact criticality if the constraint violation is sufficiently small, i.e., $\theta_k \leq \delta_{\mathrm{n}}$. Then, a normal compatible normal step always exists, and the routine can stop only because of the first condition, $\delta_j \leq \mathtt{M}\chi_j$.

▶ **Lemma 21.** *Suppose Assumptions 5, 6, 8, 13 and 15 hold. Let $k \in \mathbb{N}_0$ be the iteration index. If $\theta_k \le \delta_{\mathrm{n}}$, and if the Criticality routine does not run infinitely, then*

$$\chi^{(k)} \ge \min\left\{\frac{\Delta_{(k)}}{\mathtt{M}}, \varepsilon_\chi\right\}. \tag{15}$$

Proof. Equation (15) holds trivially if $\overline{\chi}^{(k)} \ge \varepsilon_\chi$. In this case, the Criticality Routine is not entered, we set $\chi^{(k)} = \overline{\chi}^{(k)}$ and $\Delta_{(k)} = \overline{\Delta}_{(k)}$, and (15) follows. Otherwise, the Criticality Routine must finish due to $0 < \delta_j \le \mathtt{M}\chi_j \le \mathtt{M}\chi^{(k)}$. By (14) we have

$$0 < \Delta_{(k)} \le \max\{\delta_j, \mathtt{B}\chi^{(k)}\} \overset{\mathtt{B}<\mathtt{M}}{\le} \max\{\delta_j, \mathtt{M}\chi^{(k)}\} \le \mathtt{M}\chi^{(k)}.$$

Division by $\mathtt{M}$ again implies (15). ◀

Finally, note that the models $\widehat{\boldsymbol{f}}^{(k)}, \widehat{\boldsymbol{g}}^{(k)}, \widehat{\boldsymbol{h}}^{(k)}$ are fully linear when the Criticality Routine has finished after a finite number of iterations (even if the radius is slightly increased above δ_j) because of Assumption 13.

4 Convergence to Quasi-Stationary Points

Note that the problem (ITRT$^{(k)}$) defining $\chi^{(k)}$ is similar to the criticality problem (1), but we compute the descent step starting at $\boldsymbol{x}_{\mathrm{n}}^{(k)}$ instead of $\boldsymbol{x}^{(k)}$, the surrogate model gradients are used to determine a model descent step within the approximated linearized feasible set and a variable vector norm $\|\bullet\|_k$ bounds the problem (which must not necessarily equal the trust-region norm $\|\bullet\|_{\mathrm{tr,k}}$). For the convergence analysis, the additional uncertainties are dealt with in two steps: The first part (the remainder of this section) is concerned with proving that there is a subsequence $\{\boldsymbol{x}^{(k_\ell)}\}$ of iterates converging to a quasi-stationary point. That is, it holds that

$$\lim_{\ell\to\infty} \theta(\boldsymbol{x}^{(k_\ell)}) = 0 \quad \text{and} \quad \lim_{\ell\to\infty} \omega\left(\boldsymbol{x}_{\mathrm{n}}^{(k_\ell)}; \widehat{\boldsymbol{f}}^{(k_\ell)}, \mathcal{L}_{k_\ell}, \|\bullet\|_\infty\right) = 0.$$

Afterwards, Section 5 builds on this result to show convergence to actual KKT-points.

4.1 Comparison with Related Algorithms

Most results in this section are "translated" from their single-objective pendants in [32] or [72]. The latter article is also concerned with inexact surrogates, which have error bounds that are slightly different from ours and can become exact eventually. Hence, we cannot simply apply the maximum-scalarization and be done (unfortunately). We also have to take care of the Criticality Routine and other, more subtle differences, such as the iteration-dependent norms. That is why we have decided to cite the results from [32, 72] explicitly whenever we introduce a similar one. If we deem the proofs very similar and easily transferable, we have stated so. Otherwise, hints are given on how to adapt them or the proofs are provided wholly for the sake of completeness.

Special mention has to be made of the single-objective algorithm in [25] that is further detailed in the dissertation [24]. This algorithm also employs a Criticality Routine and uses fully linear models. By encoding the surrogate modeling of black-box components as additional constraints to the original problem and using a nonlinear (even non-quadratic) subproblem for the computation of the normal step, the convergence analysis becomes easier due to the resulting inexactness bounds. At the time of writing, the nonlinear normal step computation did not suit our particular needs, but we think it easily possible and very worthwhile to also transfer their approach to the multi-objective case.

4.2 Final Definitions and Requirements

To actually investigate limit points of algorithmic iteration sequences we need Assumption 22, which guarantees their existence:

▶ **Assumption 22.** *The set $\mathcal{C}(\mathcal{X})$ is contained in a closed and bounded set.*

A detailed discussion of Assumption 22 and alternatives is given in [32]. If the iterates are contained in a bounded domain, the true functions (which are continuous according to Assumption 8) are bounded and so are their Lipschitz-continuous gradients. With Assumption 13 the models are fully linear and have to be bounded at the iterates as well, due to $\overline{\Delta}_{(k)} \le \Delta_{\max} < \infty$:

▶ **Corollary 23.** *If Assumptions 8, 13 and 22 and* $\overline{\Delta}_{(k)} \le \Delta_{\max} < \infty$ *hold, then the norm of all function and model gradients is uniformly bounded above, i.e., there is a positive constant* $\mathsf{c}_{\text{ubj}} > 0$ *such that for* $\varphi \in \{f_1, \dots, f_K, g_1, \dots, g_P, h_1, \dots, h_M\}$*, for all* $k \in \mathbb{N}_0$ *and* $\varphi^{(k)} \in \left\{\widehat{f}_1^{(k)}, \dots, \widehat{f}_K^{(k)}, \widehat{g}_1^{(k)}, \dots, \widehat{g}_P^{(k)}, \widehat{h}_1^{(k)}, \dots, \widehat{h}_M^{(k)}\right\}$ *it holds that*

$$\left\|\nabla\varphi(\boldsymbol{x}^{(k)})\right\| \le \mathsf{c}_{\text{ubj}} \quad \textit{and} \quad \left\|\nabla\varphi^{(k)}(\boldsymbol{x}^{(k)})\right\|_{\infty} \le \mathsf{c}_{\text{ubj}}.$$

Throughout this section, we consider the following iteration index sets:

▶ **Definition 24.** *The set of "successful" iteration indices is* $\mathcal{A} = \{k : \boldsymbol{x}^{(k+1)} = \boldsymbol{x}_{\text{s}}^{(k)}\}$*. The set of restoration indices is defined as* $\mathcal{R} = \{k : \boldsymbol{n}^{(k)}$ *does not exist or* (7) *fails, i.e.,* $\boldsymbol{n}^{(k)}$ *is incompatible*$\}$*. Finally, the set of filter indices (indices of iterations that modify the filter) is* $\mathcal{Z} = \{k : \boldsymbol{x}^{(k)}$ *is added to the filter* $\} \supseteq \mathcal{R}$*.*

Whenever we require "the Criticality Routine to finish finitely" in subsequent statements, then we want it to finish after a finite number of iterations and explicitly include the case that the routine is not even entered due to the Criticality Test failing in step 4 of the algorithm.

4.3 Convergence Analysis

Similar to [72] we can state "accuracy requirements" that bound the linearized constraint violation of the steps $\boldsymbol{n}^{(k)}$ and $\boldsymbol{s}^{(k)}$ by the trust-region radius $\Delta_{(k)}$:

▶ **Lemma 25** (Accuracy Requirements similar to [72, A.2.4])**.** *Suppose Assumptions 5, 6, 8 and 13 hold and that* $k \in \mathbb{N}_0$ *is an iteration index. Then there is a constant* $\mathsf{e}_{\text{err}} > 0$ *such that, if* $\boldsymbol{n}^{(k)}$ *exists as the solution to* (ITRN$^{(k)}$)*, it holds that*

$$\max\left\{\max_{\mathrm{j}} \left|h_{\mathrm{j}}(\boldsymbol{x}^{(k)}) + \nabla h_{\mathrm{j}}(\boldsymbol{x}^{(k)})^{\mathsf{T}}\boldsymbol{n}^{(k)}\right|, \max_{\mathrm{i}} g_{\mathrm{i}}(\boldsymbol{x}^{(k)}) + \nabla g_{\mathrm{i}}(\boldsymbol{x}^{(k)})^{\mathsf{T}}\boldsymbol{n}^{(k)}\right\} \le \mathsf{e}_{\text{err}}\overline{\Delta}_{(k)}\left\|\boldsymbol{n}^{(k)}\right\|_{\text{tr,k}}. \tag{16}$$

For any $k \in \mathbb{N}_0$*, for which* $\boldsymbol{n}^{(k)}$ *exists and satisfies* $\left\|\boldsymbol{n}^{(k)}\right\|_{\text{tr,k}} \le \overline{\Delta}_{(k)}$*, it also holds that*

$$\max\left\{\max_{\mathrm{j}} \left|h_{\mathrm{j}}(\boldsymbol{x}^{(k)}) + \nabla h_{\mathrm{j}}(\boldsymbol{x}^{(k)})^{\mathsf{T}}\boldsymbol{n}^{(k)}\right|, \max_{\mathrm{i}} g_{\mathrm{i}}(\boldsymbol{x}^{(k)}) + \nabla g_{\mathrm{i}}(\boldsymbol{x}^{(k)})^{\mathsf{T}}\boldsymbol{n}^{(k)}\right\} \le \mathsf{e}_{\text{err}}\overline{\Delta}_{(k)}^2, \tag{17}$$

and if the Criticality Routine finishes finitely and $\boldsymbol{s}^{(k)}$ *is the step* $\boldsymbol{n}^{(k)} + \sigma_{(k)}\boldsymbol{d}^{(k)}$ *with* $\boldsymbol{d}^{(k)}$ *computed using* (ITRT$^{(k)}$) *and* $\left\|\boldsymbol{s}^{(k)}\right\|_{\text{tr,k}} \le \Delta_{(k)}$*, we have*

$$\max\left\{\max_{\mathrm{j}} \left|h_{\mathrm{j}}(\boldsymbol{x}^{(k)}) + \nabla h_{\mathrm{j}}(\boldsymbol{x}^{(k)})^{\mathsf{T}}\boldsymbol{s}^{(k)}\right|, \max_{\mathrm{i}} g_{\mathrm{i}}(\boldsymbol{x}^{(k)}) + \nabla g_{\mathrm{i}}(\boldsymbol{x}^{(k)})^{\mathsf{T}}\boldsymbol{s}^{(k)}\right\} \le \mathsf{e}_{\text{err}}\Delta_{(k)}^2. \tag{18}$$

Proof. The constraints of (ITRN$^{(k)}$) ensure that

$$\widehat{\boldsymbol{h}}^{(k)}(\boldsymbol{x}^{(k)}) + \widehat{\boldsymbol{H}}_k(\boldsymbol{x}^{(k)})\cdot\boldsymbol{n}^{(k)} = \boldsymbol{0} \quad \text{and} \quad \widehat{\boldsymbol{g}}^{(k)}(\boldsymbol{x}^{(k)}) + \widehat{\boldsymbol{G}}_k(\boldsymbol{x}^{(k)})\cdot\boldsymbol{n}^{(k)} \le \boldsymbol{0}. \tag{19}$$

Because the models are interpolating (Assumption 13) we also have $\widehat{\boldsymbol{h}}^{(k)}(\boldsymbol{x}^{(k)}) = \boldsymbol{h}(\boldsymbol{x}^{(k)})$ and $\widehat{\boldsymbol{g}}^{(k)}(\boldsymbol{x}^{(k)}) = \boldsymbol{g}(\boldsymbol{x}^{(k)})$. Dropping the argument $\boldsymbol{x}^{(k)}$ to improve readability and then invoking the Cauchy–Schwartz inequality and the error-bounds of the models (for the preliminary radius $\overline{\Delta}_{(k)}$), we obtain

$$\begin{aligned}\max_{\mathrm{j}}\left|h_{\mathrm{j}}^{(k)} + \nabla h_{\mathrm{j}}^{(k)\mathsf{T}}\boldsymbol{n}^{(k)}\right| &= \left|h_{\mathrm{j}*}^{(k)} + \nabla h_{\mathrm{j}*}^{(k)\mathsf{T}}\boldsymbol{n}^{(k)} - 0\right| \overset{(19)}{=} \left|\left(\nabla h_{\mathrm{j}*}^{(k)} - \nabla\widehat{h}_{\mathrm{j}*}^{(k)}\right)^{\mathsf{T}}\boldsymbol{n}^{(k)}\right| \\ &\le \left\|\nabla h_{\mathrm{j}*}^{(k)} - \nabla\widehat{h}_{\mathrm{j}*}^{(k)}\right\|_2 \left\|\boldsymbol{n}^{(k)}\right\|_2 \le \dot{\mathsf{e}}_h\overline{\Delta}_{(k)}\mathsf{c}\left\|\boldsymbol{n}^{(k)}\right\|_{\text{tr,k}},\end{aligned}$$

where $\mathrm{j}*$ is the maximizing index. Similarly, we find that

$$\max_{\mathrm{i}} g_{\mathrm{i}}(\boldsymbol{x}^{(k)}) + \nabla g_{\mathrm{i}}(\boldsymbol{x}^{(k)})^{\mathsf{T}}\boldsymbol{n}^{(k)} \le \dot{\mathsf{e}}_g\overline{\Delta}_{(k)}\mathsf{c}\left\|\boldsymbol{n}^{(k)}\right\|_{\text{tr,k}},$$

and (16) follows with $\mathsf{e}_{\text{err}} = \mathsf{c}\max\{\dot{\mathsf{e}}_h, \dot{\mathsf{e}}_g\}$. Moreover, $\left\|\boldsymbol{n}^{(k)}\right\|_{\text{tr,k}} \le \overline{\Delta}_{(k)}$ leads to (17).

The second inequality (18) is derived analogously, respecting the fact that the step-size is chosen so that $\left\|\boldsymbol{s}^{(k)}\right\|_{\text{tr,k}} \le \Delta_{(k)}$ and that $\boldsymbol{x}^{(k)} + \boldsymbol{s}^{(k)}$ also satisfies the approximated linearized constraints. ◀

From the preceding accuracy results, a bound on the constraint violation can be derived:

▶ **Lemma 26** ([72, Lemma 4.3], [32, Lemma 3.1]). *Assume that the algorithm is applied to* (MOP) *and that Assumptions 5, 6, 8, 13, 15 and 22 hold. There is a constant* $\mathsf{c}_{\mathrm{ubj}} > 0$ *(independent of k) such that, if the normal step exists, it holds that*

$$\theta_k \le \left(\mathsf{e}_{\mathrm{err}}\overline{\Delta}_{(k)} + \mathsf{c}_{\mathrm{ubj}}\mathsf{c}\right)\left\|\boldsymbol{n}^{(k)}\right\|_{\mathrm{tr,k}}. \tag{20}$$

Proof. The proof works very similar to that of Lemma 4.3 in [72]. The bound (and the constant) follow from Theorem 25 by uniformly bounding the norm of all surrogate gradients. ◀

When the normal step is compatible, i.e., for iterations with $k \notin \mathcal{R}$, we can further refine the bound on θ_k:

▶ **Lemma 27** ([72, Lemma 4.4], [32, Lemma 3.4]). *Suppose that the algorithm is applied to* (MOP) *and that Assumptions 5, 6, 8, 13, 15 and 22 hold. Suppose further that $k \notin \mathcal{R}$ and that* (20) *holds. Then there is a constant* $\mathsf{c}_{\mathrm{ub}\theta} > 0$ *such that*

$$\theta_k \le \mathsf{c}_{\mathrm{ub}\theta}\max\left\{\overline{\Delta}_{(k)}^{1+\mu}, \overline{\Delta}_{(k)}^{2}\right\} \quad \textit{and} \quad \theta\left(\boldsymbol{x}^{(k)} + \boldsymbol{s}^{(k)}\right) \le \mathsf{c}_{\mathrm{ub}\theta}\Delta_{(k)}^2. \tag{21}$$

Proof. Since $k \notin \mathcal{R}$, one obtains from (20) and from (7) that

$$\begin{aligned}\theta_k &\le \left(\mathsf{e}_{\mathrm{err}}\overline{\Delta}_{(k)} + \mathsf{c}_{\mathrm{ubj}}\mathsf{c}\right)\left\|\boldsymbol{n}^{(k)}\right\|_{\mathrm{tr,k}} \le \left(\mathsf{e}_{\mathrm{err}}\overline{\Delta}_{(k)} + \mathsf{c}_{\mathrm{ubj}}\mathsf{c}\right)\mathsf{c}_{\Delta}\overline{\Delta}_{(k)}\min\left\{1, \mathsf{c}_{\mu}\overline{\Delta}_{(k)}^{\mu}\right\} \\ &\le \mathsf{c}_{\Delta}\left(\mathsf{e}_{\mathrm{err}}\overline{\Delta}_{(k)}^{2} + \mathsf{c}_{\mathrm{ubj}}\mathsf{c}\mathsf{c}_{\mu}\overline{\Delta}_{(k)}^{1+\mu}\right) \le \underbrace{\mathsf{c}_{\Delta}(\mathsf{e}_{\mathrm{err}} + \mathsf{c}_{\mathrm{ubj}}\mathsf{c}\mathsf{c}_{\mu})}_{=:\mathsf{c}^*_{\mathrm{ub}\theta}\ \text{(const.)}}\max\left\{\overline{\Delta}_{(k)}^{1+\mu}, \overline{\Delta}_{(k)}^{2}\right\}.\end{aligned}$$

For the second bound let c_{m} be any constraint component from $\boldsymbol{h}$ or $\boldsymbol{g}$. Due to Assumptions 5 and 8 we can construct a Taylor polynomial and use the Mean-Value-Theorem to obtain

$$c_{\mathsf{m}}(\boldsymbol{x}^{(k)} + \boldsymbol{s}^{(k)}) = c_{\mathsf{m}}(\boldsymbol{x}^{(k)}) + \nabla c_{\mathsf{m}}(\boldsymbol{x}^{(k)})^{\mathsf{T}}\boldsymbol{s}^{(k)} + \frac{\boldsymbol{s}^{(k)^{\mathsf{T}}}\nabla^2 c_{\mathsf{m}}(\boldsymbol{\xi})\,\boldsymbol{s}^{(k)}}{2} \tag{22}$$

for some $\boldsymbol{\xi}$ on the line segment from $\boldsymbol{x}^{(k)}$ to $\boldsymbol{x}^{(k)} + \boldsymbol{s}^{(k)}$. From (18) we know that $\left|c_{\mathsf{m}}(\boldsymbol{x}^{(k)}) + \nabla c_{\mathsf{m}}(\boldsymbol{x}^{(k)})^{\mathsf{T}}\boldsymbol{s}^{(k)}\right| \le \mathsf{e}_{\mathrm{err}}\Delta_{(k)}^2$. With Assumptions 8 and 22 the norm of all constraint Hessians in equation (22) can be bounded from above globally (say, by $2\kappa_H > 0$) so that the Cauchy–Schwartz inequality gives

$$\frac{\boldsymbol{s}^{(k)^{\mathsf{T}}}\nabla^2 c_{\mathsf{m}}(\boldsymbol{\xi})\,\boldsymbol{s}^{(k)}}{2} \le \frac{\left\|\boldsymbol{s}^{(k)}\right\|_2^2\left\|\nabla^2 c_{\mathsf{m}}(\boldsymbol{\xi})\right\|_2}{2} \le \kappa_H\Delta_{(k)}^2.$$

Hence, (21) follows with $\mathsf{c}_{\mathrm{ub}\theta} =: \max\{\mathsf{c}^*_{\mathrm{ub}\theta}, \mathsf{e}_{\mathrm{err}} + \kappa_H\}$. ◀

The requirements for the accuracy requirements to hold are also met within the Criticality Routine, allowing for the following corollary:

▶ **Corollary 28.** *Let $k \in \mathbb{N}_0$ be an iteration index and suppose the same requirements as in Theorem 26 hold. If the Criticality Routine is entered, then for any sub-iteration index $j \in \mathbb{N}_0$, it holds that*

$$\theta_k \le \mathsf{e}_{\mathrm{err}}\delta_j^2 + \mathsf{c}_{\mathrm{ubj}}\left\|\boldsymbol{n}_j^{(k)}\right\|_2, \tag{23}$$

If the Criticality Routine is not entered or if it has completed finitely, then it also holds that

$$\theta_k \le \mathsf{e}_{\mathrm{err}}\Delta_{(k)}^2 + \mathsf{c}_{\mathrm{ubj}}\left\|\boldsymbol{n}^{(k)}\right\|_2, \tag{24}$$

and if $k \notin \mathcal{R}$, then

$$\theta_k \le \mathsf{c}_{\mathrm{ub}\theta}\max\left\{\Delta_{(k)}^{1+\mu}, \Delta_{(k)}^2\right\}. \tag{25}$$

4.3.1 Convergence in the Criticality Routine

Our first important observation is that quasi-criticality is approximated if the criticality loop runs infinitely:

▶ **Lemma 29** ([6, Lemma 8]). *Assume that Assumptions 5, 6, 8, 13, 15 and 22 hold. Denote by χ_j the inexact criticality value from Theorem 20 in the criticality subroutine. If the criticality routine runs infinitely ($j \to \infty$) at $\boldsymbol{x}^{(k)}$, then*

$$\theta\Big(\boldsymbol{x}^{(k)}\Big) = 0 \text{ and } \lim_{j\to\infty} \delta_j = 0 \text{ and } \lim_{j\to\infty} \chi_j = 0.$$

Proof. As in the description of the Criticality Routine, let $j \in \mathbb{N}_0$ be the sub-iteration index, $\delta_j^+ := \alpha^{j+1}\overline{\Delta}_{(k)}$, and $\delta_j = \delta_j^+$ after the index j has been increased. In accordance with Theorem 29, denote by χ_j the updated (doubly inexact) criticality measure from step 1 and step 2e. There are two logically disjunct stopping criteria for the Criticality Routine.

Firstly, the Criticality Routine may stop if $\mathtt{M}\chi_j \ge \delta_j$ for some $j \in \mathbb{N}_0$. Conversely, if the routine loops infinitely, then it must hold that

$$\chi_j < \frac{\delta_j}{\mathtt{M}} = \alpha^j \frac{\overline{\Delta}_{(k)}}{\mathtt{M}} \qquad \text{for all } j \in \mathbb{N}_0.$$

Because of $\alpha \in (0,1)$ the right side goes to zero and then the second limit in Theorem 29 also follows.

Secondly, the routine stops if no compatible step $\boldsymbol{n}_j$ for the preliminary radius δ_j^+ can be found anymore. Vice versa, if the routine runs infinitely, there always is a compatible step with

$$\|\boldsymbol{n}_j\|_2 \le \mathtt{c}\mathtt{c}_\Delta \delta_j \min\big\{1, \mathtt{c}_\mu \delta_j^\mu\big\}.$$

This implies $\boldsymbol{n}_j^{(k)} \to 0$ for $j \to \infty$. From (23) it then follows that it must already hold that $\theta(\boldsymbol{x}^{(k)}) = 0$. ◀

4.3.2 Infinitely Many Filter Iterations

Outside of the criticality loop, we can show that feasibility is approached if the number of filter iterations is infinite:

▶ **Lemma 30** ([32, Lemma 3.3]). *Suppose that the algorithm is applied to* (MOP) *and that Assumptions 5, 8 and 22 hold. If $|\mathcal{Z}| = \infty$, then* $\lim_{k\to\infty, k\in\mathcal{Z}} \theta_k = 0$.

Proof. The proof is exactly the same as in [32], only $f(\bullet)$ has to be substituted by $\Phi\big[\boldsymbol{f}\big](\bullet)$. It also works with a $K+1$ dimensional filter that uses $\boldsymbol{f}$ instead of $\Phi\big[\boldsymbol{f}\big](\bullet)$. ◀

The next result shows that feasibility is approached for *any* subsequence of iteration indices. The result is a bit stronger than that found in [32] and not necessarily needed for the convergence proofs. A similar theorem can be found in [71].

▶ **Lemma 31.** *Suppose that the algorithm is applied to* (MOP) *and that Assumptions 5, 8 and 22 hold. If $|\mathcal{Z}| = \infty$, then $\lim_{k\to\infty} \theta_k = 0$.*

Proof. Assume, for sake of contradiction, that there is a subsequence of indices $\{k_\ell\}$ for which the constraint violation is bounded below

$$\theta_{k_\ell} \ge \varepsilon > 0 \qquad \forall\, k_\ell. \tag{26}$$

From Theorem 30 we know that there is some k_0 such that

$$\theta_k < \varepsilon \quad \text{for all } k \in \mathcal{Z} \text{ with } k \ge k_0. \tag{27}$$

Because there are infinitely many indices both in $\{k_\ell\}$ and $\mathcal{Z}$, each $k \in \{k_\ell\}$ (except maybe the first) must lie between a smallest filter index $\kappa_1 \in \mathcal{Z}$ and a largest index $\kappa_2 \in \mathcal{Z}$, respectively, $\kappa_1(k) \le k \le \kappa_2(k)$, and from (26) and (27) we can deduce that there is a $k_1 \ge k_0$ such that

$$k_0 < \kappa_1(k) < k < \kappa_2(k) \quad \text{for all } k \in \{k_\ell\} \text{ with } k \ge k_1.$$

Let $k \in \{k_\ell\}$ be an index with $k \ge k_1$. For all indices κ with $\kappa_1(k) < \kappa < \kappa_2(k)$, the iteration is not a filter iteration and hence the function Φ is "monotonic":

$$\Phi\Big(\boldsymbol{x}^{(\kappa)}\Big) - \Phi\Big(\boldsymbol{x}^{(\kappa+1)}\Big) \ge 0.$$

Moreover, there must be some smallest successful index $\kappa_2^*(k)$ with $k \le \kappa_2^*(k) < \kappa_2(k)$, reducing the constraint violation from above ε to below ε again. $\kappa_2^*(k)$ is not a filter index, so (11) succeeds and $\rho^{(\kappa_2^*(k))} \ge \nu_0 > 0$, resulting in

$$\Phi\Big(\boldsymbol{x}^{(\kappa_2^*(k))}\Big) - \Phi\Big(\boldsymbol{x}^{(\kappa_2^*(k)+1)}\Big) \ge \nu_0 \kappa_\theta \theta^{\psi}_{\kappa_2^*(k)} \ge \nu_0 \kappa_\theta \varepsilon^{\psi} > 0. \tag{28}$$

The function Φ might be increased eventually in the filter iteration $\kappa_2(k)$ (when accepting $\boldsymbol{x}^{(\kappa_2(k)+1)}$), but due to the monotonicity of the intermediate iterations it holds that $\Phi\big(\boldsymbol{x}^{(\kappa_2^*(k)+1)}\big) - \Phi\big(\boldsymbol{x}^{(\kappa_2(k))}\big) \ge 0$ and thus

$$\Phi\Big(\boldsymbol{x}^{(\kappa_2^*(k))}\Big) - \Phi\Big(\boldsymbol{x}^{(\kappa_2(k))}\Big) \ge \nu_0 \kappa_\theta \varepsilon^{\psi} > 0, \tag{29}$$

and $\boldsymbol{x}^{(\kappa_2(k))}$ is added to the filter.

For k from above, let $k_+ \in \{k_\ell\}$ be the smallest iteration index following k such that it holds for the enclosing filter indices that $\kappa_2(k) \le \kappa_1(k_+)$. There then must be a largest successful index $\kappa_1^*(k_+)$ with $\kappa_1(k_+) \le \kappa_1^*(k_+) < k_+$, such that the constraint violation is increased from below ε to above ε,

$$\theta_{\kappa_1^*(k_+)} < \varepsilon \le \theta_{\kappa_1^*(k_+)+1},$$

and from the filter mechanism we know that then the objective reduction has to be significant compared to *all* points in the filter with a smaller constraint violation, including that with index $\kappa_2(k)$ and $\theta_{\kappa_2(k)} < \varepsilon$:

$$\Phi\Big(\boldsymbol{x}^{(\kappa_2(k))}\Big) - \Phi\Big(\boldsymbol{x}^{(\kappa_1^*(k_+)+1)}\Big) \ge \gamma_\theta \theta_{\kappa_2(k)} \ge 0.$$

Using the same notation as above, (i.e., $\kappa_2^*(k_+)$ as in (28)), we deduce from the monotonicity of the intermediate iterations that $\Phi\big(\boldsymbol{x}^{(\kappa_2(k))}\big) - \Phi\big(\boldsymbol{x}^{(\kappa_2^*(k_+))}\big) \ge 0$ and thus, with (29),

$$\Phi\Big(\boldsymbol{x}^{(\kappa_2^*(k))}\Big) - \Phi\Big(\boldsymbol{x}^{(\kappa_2^*(k_+))}\Big) \ge \nu_0 \kappa_\theta \varepsilon^{\psi} > 0. \tag{30}$$

By repeating the above procedure, we see that it is possible to construct an infinite subsequence $\{\kappa_j\}$ of iteration indices from the κ_2^* values (of indices from $\{k_\ell\}$), for which $\Phi(\boldsymbol{x}^{(\kappa_j)})$ is strictly monotonically decreasing with a guaranteed (constant) objective reduction (30). This is a contradiction to Φ being bounded below as per Assumptions 8 and 22. ◀

What follows next is a series of auxiliary lemmata to finally show convergence of the inexact criticality measure when $|\mathcal{Z}| = \infty$. First, we have a bound on the surrogate objective change along the normal step:

▶ **Lemma 32** (Similar to a bound in [32, Lemma 3.5]). *Suppose the algorithm is applied to* (MOP), *that Assumptions 5, 6, 8, 13, 18 and 22 hold and that the Criticality Routine does not run infinitely. Suppose further that $k \notin \mathcal{R}$. Then it holds that*

$$\Big|\Phi^{(k)}(\boldsymbol{x}^{(k)}) - \Phi^{(k)}(\boldsymbol{x}_{\mathrm{n}}^{(k)})\Big| = \Big|\Phi^{(k)}(\boldsymbol{x}_{\mathrm{n}}^{(k)}) - \Phi^{(k)}(\boldsymbol{x}^{(k)})\Big| \le \mathtt{c}_{\mathrm{ubj}} \Big\|\boldsymbol{n}^{(k)}\Big\|_2 + \frac{1}{2}\mathtt{H}\Big\|\boldsymbol{n}^{(k)}\Big\|_2^2. \tag{31}$$

Proof. There are maximizing indices $\mathfrak{l}_1, \mathfrak{l}_2 \in \{1, \dots, K\}$ such that

$$\Phi^{(k)}(\boldsymbol{x}_{\mathrm{n}}^{(k)}) - \Phi^{(k)}(\boldsymbol{x}^{(k)}) = \widehat{f}^{(k)}_{\mathfrak{l}_1}(\boldsymbol{x}_{\mathrm{n}}^{(k)}) - \widehat{f}^{(k)}_{\mathfrak{l}_2}(\boldsymbol{x}^{(k)}) \le \widehat{f}^{(k)}_{\mathfrak{l}_1}(\boldsymbol{x}_{\mathrm{n}}^{(k)}) - \widehat{f}^{(k)}_{\mathfrak{l}_1}(\boldsymbol{x}^{(k)}).$$

Using a 2nd degree Taylor approximation of $\widehat{f}^{(k)}_{\mathfrak{l}_1}$ around $\boldsymbol{x}^{(k)}$ at $\boldsymbol{x}_{\mathrm{n}}^{(k)}$ results in

$$\begin{aligned}
\Phi^{(k)}(\boldsymbol{x}_{\mathrm{n}}^{(k)}) - \Phi^{(k)}(\boldsymbol{x}^{(k)}) &\le \widehat{f}^{(k)}_{\mathfrak{l}_1}(\boldsymbol{x}_{\mathrm{n}}^{(k)}) - \widehat{f}^{(k)}_{\mathfrak{l}_1}(\boldsymbol{x}^{(k)}) \\
&\le \Big|\boldsymbol{n}^{(k)\mathsf{T}} \nabla \widehat{f}^{(k)}_{\mathfrak{l}_1}(\boldsymbol{x}^{(k)})\Big| + \Big|\frac{1}{2}\boldsymbol{n}^{(k)\mathsf{T}} \nabla^2 \widehat{f}^{(k)}_{\mathfrak{l}_1}(\boldsymbol{x}^{(k)})\,\boldsymbol{n}^{(k)}\Big| \\
&\le \Big\|\nabla \widehat{\boldsymbol{f}}^{(k)}_{\mathfrak{l}_1}(\boldsymbol{x}^{(k)})\Big\|_2 \Big\|\boldsymbol{n}^{(k)}\Big\|_2 + \frac{1}{2}\Big\|\nabla^2 \widehat{f}^{(k)}_{\mathfrak{l}_1}(\boldsymbol{x}^{(k)})\Big\|_2 \Big\|\boldsymbol{n}^{(k)}\Big\|_2^2 \\
&\le \mathtt{c}_{\mathrm{ubj}} \Big\|\boldsymbol{n}^{(k)}\Big\|_2 + \frac{1}{2}\mathtt{H}\Big\|\boldsymbol{n}^{(k)}\Big\|_2^2,
\end{aligned}$$

where the last inequality comes from Assumption 18 and Theorem 23. Analogously, we can show

$$-\left(\Phi^{(k)}(\boldsymbol{x}_{\mathrm{n}}^{(k)}) - \Phi^{(k)}(\boldsymbol{x}^{(k)})\right) \le \widehat{f}_{\mathrm{l}_2}^{(k)}(\boldsymbol{x}^{(k)}) - \widehat{f}_{\mathrm{l}_2}^{(k)}(\boldsymbol{x}_{\mathrm{n}}^{(k)}) \le \mathtt{c}_{\mathrm{ubj}} \left\|\boldsymbol{n}^{(k)}\right\|_2 + \frac{1}{2}\mathtt{H}\left\|\boldsymbol{n}^{(k)}\right\|_2^2. \qquad ◀$$

The next lemma provides a sufficient decrease bound in case that the doubly inexact criticality measure is bounded below, and the radius is sufficiently small:

▶ **Lemma 33** ([32, Lemma 3.5]). *Suppose the algorithm is applied to* (MOP)*, that Assumptions 5, 6, 8, 13, 18, 19 and 22 hold and that the Criticality Routine does not run infinitely. Suppose further, that $k \notin \mathcal{R}$, that*

$$\chi^{(k)} \ge \frac{1}{\mathtt{w}}\chi_2^{(k)} > \frac{1}{\mathtt{w}}\epsilon \tag{LB}$$

for some $\epsilon > 0$ and that

$$\Delta_{(k)} \le \delta_m := \min\left\{ \frac{\epsilon}{\mathtt{W}}, \left(\frac{2\mathtt{c}_{\mathrm{ubj}}}{\mathtt{H}\mathtt{c}_\Delta \mathtt{c}_\mu}\right)^{\frac{1}{1+\mu}}, \left(\frac{\mathtt{c}_{\mathrm{sd}}\epsilon}{4\mathtt{c}_{\mathrm{ubj}}\mathtt{c}_\Delta \mathtt{c}_\mu}^{\frac{1}{1+\mu}}\right) \right\}. \tag{32}$$

Then

$$\Phi^{(k)}(\boldsymbol{x}^{(k)}) - \Phi^{(k)}(\boldsymbol{x}^{(k)} + \boldsymbol{s}^{(k)}) \ge \frac{1}{2}\mathtt{c}_{\mathrm{sd}}\epsilon\Delta_{(k)}. \tag{33}$$

Proof. The proof is similar to that given in [32]. The bounds (LB) and (32) are substituted into the sufficient decrease equation of Assumption 19. After some arithmetic transformations, the bound (33) follows from Theorem 32, (32), and the fact that $\boldsymbol{n}^{(k)}$ is compatible. ◀

Under similar conditions as in Theorem 33 the iteration will be successful:

▶ **Lemma 34** ([32, Lemma 3.6]). *Suppose the algorithm is applied to* (MOP)*, that Assumptions 5, 6, 8, 13, 18, 19 and 22 hold and that the Criticality Routine does not run infinitely. Suppose further, that $k \notin \mathcal{R}$, that* (LB) *holds again and that*

$$\Delta_{(k)} \le \delta_\rho := \min\left\{\delta_m, \frac{(1-\nu_1)\mathtt{c}_{\mathrm{sd}}\epsilon}{2\mathtt{e}_f}\right\}. \tag{34}$$

Then iteration k is successful, that is, $\rho_{(k)} \ge \nu_1$.

Proof. We can again follow the proof in [32], because from Assumption 13 we can conclude (cf. [69, Lemma 4.16]) that the model error bound holds also for the scalarization:

$$\left|\Phi\left[\boldsymbol{f}\right](\boldsymbol{\xi}) - \Phi\left[\widehat{\boldsymbol{f}}^{(k)}\right](\boldsymbol{\xi})\right| = \left|\Phi(\boldsymbol{\xi}) - \Phi^{(k)}(\boldsymbol{\xi})\right| \le \mathtt{e}_f\Delta_{(k)}^2 \quad \forall\, \boldsymbol{\xi} \in B^{(k)}. \tag{35}$$

◀

Further, if the radius is sufficiently small, then the test (11) will succeed (prohibiting a θ-iteration):

▶ **Lemma 35** ([32, Lemma 3.7]). *Suppose the algorithm is applied to* (MOP)*, that Assumptions 5, 6, 8, 13, 15, 18, 19 and 22 hold and that the Criticality Routine does not run infinitely. Suppose further, that $k \notin \mathcal{R}$, that* (LB) *holds again, that $\theta_k \le \delta_{\mathrm{n}}$, and that*

$$\Delta_{(k)} \le \delta_f := \min\left\{\delta_m, 1, \left(\frac{\mathtt{c}_{\mathrm{sd}}\epsilon}{2\kappa_\theta \mathtt{c}_{\mathrm{ub}\theta}^{\psi}}\right)^{\frac{1}{\psi(1+\mu)-1}}\right\}. \tag{36}$$

Then (11) *succeeds, i.e., $\Phi^{(k)}(\boldsymbol{x}^{(k)}) - \Phi^{(k)}(\boldsymbol{x}_{\mathrm{s}}^{(k)}) \ge \kappa_\theta\theta_k^{\psi}$.*

Proof. Theorem 33 provides

$$\Phi^{(k)}(\boldsymbol{x}^{(k)}) - \Phi^{(k)}(\boldsymbol{x}_{\mathrm{s}}^{(k)}) = \Phi^{(k)}(\boldsymbol{x}^{(k)}) - \Phi^{(k)}(\boldsymbol{x}^{(k)} + \boldsymbol{s}^{(k)}) \ge \frac{1}{2}\mathtt{c}_{\mathrm{sd}}\epsilon\Delta_{(k)}.$$

The small constraint violation $\theta_k \le \delta_{\mathrm{n}}$ and $k \notin \mathcal{R}$ imply (25). Because of $\Delta_{(k)} \le 1$ and $\mu \in (0,1)$ the bound (25) simplifies to $\theta_k \le \Delta_{(k)}^{1+\mu}$, and it follows that $\mathtt{c}_{\mathrm{ub}\theta}^{\psi}\Delta_{(k)}^{\psi(1+\mu)} \ge \theta_k^{\psi}$. The final inequality in (36) then leads to

$$\frac{1}{2}\mathtt{c}_{\mathrm{sd}}\epsilon\Delta_{(k)} \ge \kappa_\theta \mathtt{c}_{\mathrm{ub}\theta}\Delta_{(k)}^{\psi(1+\mu)},$$

which then implies (11). ◀

If additionally the constraint violation is small enough, then the filter pair at the trial point will not be filter-dominated by the pair at $\boldsymbol{x}^{(k)}$:

▶ **Lemma 36** ([32, Lemma 3.8]). *Suppose the algorithm is applied to* (MOP)*, that Assumptions 5, 6, 8, 13, 15, 18, 19 and 22 hold and that the Criticality Routine does not run infinitely. Suppose further, that $k \notin \mathcal{R}$, that* (LB) *holds again, that $\Delta_{(k)} \le \delta_\rho$ as in* (34) *and that $\theta_k \le \delta_{\mathrm{n}}$ and that*

$$\theta_k \le \delta_\theta := \min\left\{ \frac{1}{\mathsf{c}_{\mathrm{ub}\theta}} \left(\frac{\nu_1 \mathsf{c}_{\mathrm{sd}} \epsilon}{2\gamma_\theta} \right)^2, \frac{1}{\mathsf{c}_{\mathrm{ub}\theta}^{\frac{1}{\mu}}} \left(\frac{\nu_1 \mathsf{c}_{\mathrm{sd}} \epsilon}{2\gamma_\theta} \right)^{\frac{1+\mu}{\mu}} \right\}. \tag{37}$$

Then $\Phi\big(\boldsymbol{x}^{(k)}\big) - \Phi\big(\boldsymbol{x}^{(k)} + \boldsymbol{s}^{(k)}\big) \ge \gamma_\theta \theta_k$.

Proof. From Theorem 27 we deduce that $\frac{\theta_k}{\mathsf{c}_{\mathrm{ub}\theta}} \le \Delta_{(k)}^{1+\mu}$ or $\frac{\theta_k}{\mathsf{c}_{\mathrm{ub}\theta}} \le \Delta_{(k)}^2$ and thus

$$\Delta_{(k)} \ge \min\left\{ \left(\frac{\theta_k}{\mathsf{c}_{\mathrm{ub}\theta}} \right)^{\frac{1}{1+\mu}}, \left(\frac{\theta_k}{\mathsf{c}_{\mathrm{ub}\theta}} \right)^{\frac{1}{2}} \right\}. \tag{38}$$

With Theorem 34, it follows that

$$\begin{aligned} \Phi\Big(\boldsymbol{x}^{(k)}\Big) - \Phi\Big(\boldsymbol{x}^{(k)} + \boldsymbol{s}^{(k)}\Big) &\ge \nu_1 \Big(\Phi^{(k)}\Big(\boldsymbol{x}^{(k)}\Big) - \Phi^{(k)}\Big(\boldsymbol{x}^{(k)} + \boldsymbol{s}^{(k)}\Big) \Big) \overset{(33)}{\ge} \frac{1}{2} \nu_1 \mathsf{c}_{\mathrm{sd}} \epsilon \Delta_{(k)} \\ &\overset{(38)}{\ge} \frac{1}{2} \nu_1 \mathsf{c}_{\mathrm{sd}} \epsilon \min\left\{ \left(\frac{\theta_k}{\mathsf{c}_{\mathrm{ub}\theta}} \right)^{\frac{1}{1+\mu}}, \left(\frac{\theta_k}{\mathsf{c}_{\mathrm{ub}\theta}} \right)^{\frac{1}{2}} \right\} \overset{(37)}{\ge} \gamma_\theta \theta_k. \end{aligned}$$ ◀

In the preceding lemmata, it has always been assumed that the iteration index does not belong to the set of restoration indices $\mathcal{R}$. This is ensured if both the radius and the constraint violation are small enough:

▶ **Lemma 37** ([32, Lemma 3.9]). *Suppose the algorithm is applied to* (MOP)*, that Assumptions 5, 6, 8, 13, 15, 18, 19 and 22 hold and that the Criticality Routine does not run infinitely. Suppose that* (LB) *holds again. Suppose further that*

$$\Delta_{(k)} \le \delta_{\mathcal{R}} := \min\left\{ \gamma_0 \delta_\rho, \left(\frac{1}{\mathsf{c}_\mu} \right)^{\frac{1}{\mu}}, \left(\frac{(1-\gamma_\theta)\gamma_0^2 \mathsf{c}_\Delta \mathsf{c}_\mu}{\mathsf{c}_{\mathrm{ubn}} \mathsf{c}_{\mathrm{ub}\theta}} \right)^{\frac{1}{1-\mu}}, \right\} \tag{39}$$

and that

$$\theta_k \le \min\{\delta_\theta, \delta_{\mathrm{n}}\}. \tag{40}$$

If $k > 0$, then $k \notin \mathcal{R}$.

Proof. The proof works exactly as in [32], because the Criticality Routine is not entered in restoration iterations. ◀

The previous lemmata allow us to investigate two mutually exclusive cases defined by the number of filter iterations. The first convergence result is obtained for the case that there are infinitely many such iterations.

▶ **Lemma 38.** *Suppose the algorithm is applied to* (MOP)*, that Assumptions 5, 6, 8, 13, 15, 18, 19 and 22 hold and that the Criticality Routine does not run infinitely. Suppose $|\mathcal{Z}| = \infty$. For any subsequence of iteration indices $\{k_\ell\}$ with $|\{k_\ell\} \cap \mathcal{Z}| = \infty$ it holds that $\liminf_{\ell\to\infty} \Delta_{(k_\ell)} = 0$. Especially, we have $\lim_{\substack{k\to\infty,\\ k\in\mathcal{Z}}} \Delta_{(k)} = 0$.*

Proof. The proof is along the lines of [32, Lemma 3.10], but additionally takes into account the Criticality Routine. First, let $\{k_\ell\}$ a sequence of indices containing infinitely many filter indices, $|\{k_\ell\} \cap \mathcal{Z}| = \infty$, and assume that it was bounded away from zero:

$$\Delta_{(k_\ell)} \ge \Delta_{\min} > 0 \qquad \forall\, \ell \in \mathbb{N}_0. \tag{41}$$

From Theorem 31 and Assumption 15 it follows that

$$\lim_{\ell\to\infty} \left\| \boldsymbol{n}^{(k_\ell)} \right\|_{\mathrm{tr,k}} = 0 \overset{\text{Assumption 6}}{=} \lim_{\ell\to\infty} \left\| \boldsymbol{n}^{(k_\ell)} \right\|_2. \tag{42}$$

Consequently, for sufficiently large ℓ, the steps will become compatible and satisfy (7) and thus $k_\ell \notin \mathcal{R}$. Hence, Theorem 32 applies and inequality (31) holds again:

$$\left|\Phi^{(k_\ell)}\left(\boldsymbol{x}_{\mathrm{n}}^{(k_\ell)}\right) - \Phi^{(k_\ell)}\left(\boldsymbol{x}^{(k_\ell)}\right)\right| \le \mathtt{c}_{\mathrm{ubj}} \left\|\boldsymbol{n}^{(k_\ell)}\right\|_2 + \frac{1}{2}\mathtt{H}\left\|\boldsymbol{n}^{(k_\ell)}\right\|_2^2,$$

so that (42) gives

$$\lim_{\ell\to\infty}\left|\Phi^{(k_\ell)}\left(\boldsymbol{x}^{(k_\ell)}\right) - \Phi^{(k_\ell)}\left(\boldsymbol{x}_{\mathrm{n}}^{(k_\ell)}\right)\right| = 0. \tag{43}$$

Theorem 31 enables the Criticality Routine. Additionally, from the boundedness of $\Delta_{(k_\ell)}$ as per (41) and from Assumption 15, it also follows that, for large ℓ, the Criticality Routine will not exit because no compatible normal step exists anymore. This ensures that

$$\chi^{(k_\ell)} \ge \min\left\{\varepsilon_\chi, \frac{\Delta_{(k_\ell)}}{\mathtt{M}}\right\} \ge \min\left\{\varepsilon_\chi, \frac{\Delta_{\min}}{\mathtt{M}}\right\} =: \mathtt{z} > 0 \tag{44}$$

for large ℓ. We can use this fact in Assumption 19 and obtain

$$\Phi^{(k_\ell)}\left(\boldsymbol{x}_{\mathrm{n}}^{(k_\ell)}\right) - \Phi^{(k_\ell)}\left(\boldsymbol{x}_{\mathrm{s}}^{(k_\ell)}\right) \ge \underbrace{\mathtt{c}_{\mathrm{sd}}\mathtt{z}\min\left\{\frac{\mathtt{z}}{\mathtt{W}}, \Delta_{\min}\right\}}_{\text{const.}} > 0. \tag{45}$$

If we "add zero", then we obtain the decomposition

$$\Phi^{(k_\ell)}\left(\boldsymbol{x}^{(k_\ell)}\right) - \Phi^{(k_\ell)}\left(\boldsymbol{x}_{\mathrm{s}}^{(k_\ell)}\right) = \left(\Phi^{(k_\ell)}\left(\boldsymbol{x}^{(k_\ell)}\right) - \Phi^{(k_\ell)}\left(\boldsymbol{x}_{\mathrm{n}}^{(k_\ell)}\right)\right) + \left(\Phi^{(k_\ell)}\left(\boldsymbol{x}_{\mathrm{n}}^{(k_\ell)}\right) - \Phi^{(k_\ell)}\left(\boldsymbol{x}_{\mathrm{s}}^{(k_\ell)}\right)\right). \tag{46}$$

By looking at (43), we see that the first set of parenthesis vanishes, and that difference of values is the same in the limit:

$$\lim_{\ell\to\infty}\Phi^{(k_\ell)}\left(\boldsymbol{x}^{(k_\ell)}\right) - \Phi^{(k_\ell)}\left(\boldsymbol{x}_{\mathrm{s}}^{(k_\ell)}\right) = \lim_{\ell\to\infty}\Phi^{(k_\ell)}\left(\boldsymbol{x}_{\mathrm{n}}^{(k_\ell)}\right) - \Phi^{(k_\ell)}\left(\boldsymbol{x}_{\mathrm{s}}^{(k_\ell)}\right). \tag{47}$$

Because of $|\{k_\ell\} \cap \mathcal{Z}| = \infty$, there is a subsequence $\{k_j\} \subseteq \{k_\ell\}$ of filter indices, $k_j \in \mathcal{Z}$, for which it must hold that $k_j \in \mathcal{R}$ or that (11) fails. We have already shown that $k_j \notin \mathcal{R}$ for large j. Hence, for j sufficiently large, it must hold that $\kappa_\theta \theta_{k_j}^{\psi} > \Phi^{(k_j)}(\boldsymbol{x}^{(k_j)}) - \Phi^{(k_j)}(\boldsymbol{x}^{(k_j)} + \boldsymbol{s}^{(k_j)})$ and Theorem 31 implies that the left-hand side (LHS) must go to zero. So the limits in (47) are bounded above by 0. But this is a contradiction to (45). Hence, no infinite subsequence $\{k_\ell\}$ with $|\{k_\ell\} \cap \mathcal{Z}| = \infty$ and (41) can exist. ◀

▶ Remark 39. Without further assumptions on the restoration step it does not seem possible to show $\lim_{k\to\infty}\Delta_{(k)} = 0$ in the case that $|\mathcal{Z}| = \infty$. For any element of an index sequence $\{k_\ell\}$, bounded away from zero and with $|\{k_\ell\} \cap \mathcal{Z}| < \infty$, there might be a preceding restoration iteration with arbitrarily small radius itself, but possibly increasing it without any restriction, so that no contradiction can be derived. This difficulty has been observed in the literature before, see e.g. [71, Remark 7].

▶ **Lemma 40.** *Suppose the algorithm is applied to* (MOP)*, that Assumptions 5, 6, 8, 13, 15, 18, 19 and 22 hold and that the Criticality Routine does not run infinitely. Suppose that $|\mathcal{Z}| = \infty$ and that the doubly inexact criticality is bounded as per* (LB) *for all $k \in \mathcal{Z}$. Then the trust-region radius is also bounded below, i.e., there is some $\Delta_{\min} > 0$ such that $\Delta_{(k)} \ge \Delta_{\min}$ for all $k \in \mathcal{Z}$.*

Proof. If there are infinitely many filter indices $\mathcal{Z}$, then we know from Theorem 31 that $\theta_k \to 0$. Suppose $\{\Delta_{(k)}\}_{k\in\mathcal{Z}}$ is not bounded away from zero. Then there is a subsequence $\{k_\ell\} \subseteq \mathcal{Z}$ with $\lim_{\ell\to\infty}\Delta_{(k_\ell)} = 0$. Following the argumentation in [32, Lemma 3.10], we see that for ℓ large enough, Theorem 37 applies and guarantees that $k_\ell \notin \mathcal{R}$. At the same time, Theorem 35 applies for large ℓ, so that the test (11) always succeeds for $\boldsymbol{x}^{(k_\ell)}$. Hence, there is some $\ell_0 \in \mathbb{N}_0$ such that for all k_ℓ with $\ell \ge \ell_0$, it holds that $k_\ell \notin \mathcal{Z}$. This contradicts $\{k_\ell\} \subseteq \mathcal{Z}$. ◀

We are finally able to state the convergence result for the case of infinitely many filter iterations as a cumulative corollary derived from Theorems 30, 38 and 40:

▶ **Corollary 41** ([32, Lemma 3.10])**.** *Suppose the algorithm is applied to* (MOP)*, that Assumptions 5, 6, 8, 13, 15, 18, 19 and 22 hold and that the Criticality Routine does not run infinitely. Suppose further, that $|\mathcal{Z}| = \infty$. Then there is a subsequence $\{k_\ell\} \subseteq \mathcal{Z}$ of filter indices with*

$$\lim_{\ell\to\infty}\theta_{k_\ell} = 0, \quad \lim_{\ell\to\infty}\Delta_{(k_\ell)} = 0 \quad \textit{and} \quad \lim_{\ell\to\infty}\chi^{(k_\ell)} = 0.$$

4.3.3 Finitely Many Filter Iterations

We have shown that a quasi-stationary point is approached when there are infinitely many filter indices. We now concentrate on the case that there are only finitely many filter indices (and note that this then implies that there are only finitely many restoration indices as well). *From now on, k_0 is the last iteration index for which $\boldsymbol{x}^{(k_0-1)}$ is added to the filter.*

▶ **Lemma 42** ([32, Lemma 3.11])**.** *Suppose the algorithm is applied to* (MOP)*, that Assumptions 5, 6, 8, 13, 15, 18, 19 and 22 hold and that the Criticality Routine does not run infinitely. Suppose further, that $|\mathcal{Z}| < \infty$. Then*

$$\lim_{k\to\infty} \theta_k = 0. \tag{48}$$

Furthermore, $\boldsymbol{n}^{(k)}$ satisfies (20) *for all $k \geq k_0$ large enough.*

Proof. First consider the case that there are only finitely many successful indices $\mathcal{A}$. Then $\Delta_{(k)} \xrightarrow{k\to\infty} 0$. Because there are no restoration iterations for $k \geq k_0$, it follows from (7) that $\boldsymbol{n}^{(k)} \to 0$. Equation (24) finally gives $\theta_k \to 0$.

Now, suppose that $|\mathcal{A}| = \infty$ and consider any *successful* iteration with $k \geq k_0$. Then $\boldsymbol{x}^{(k)}$ is not added to the filter, and it follows from step 5 of the algorithm that $\rho_{(k)} \geq \nu_0$ and from the definition of $\rho_{(k)}$ and from (11) that

$$\Phi\left(\boldsymbol{x}^{(k)}\right) - \Phi\left(\boldsymbol{x}^{(k)} + \boldsymbol{s}^{(k)}\right) \geq \nu_0\left(\Phi^{(k)}\left(\boldsymbol{x}^{(k)}\right) - \Phi^{(k)}\left(\boldsymbol{x}^{(k)} + \boldsymbol{s}^{(k)}\right)\right) \geq \nu_0\kappa_\theta\theta_k^\psi \geq 0. \tag{49}$$

The sequence $\left\{\Phi\left(\boldsymbol{x}^{(k)}\right)\right\}_{k\geq k_0}$ is bounded below due to Assumptions 8 and 22, and it is monotonically decreasing because (11) always succeeds due to $k \geq k_0$. Thus, no new iterate is ever chosen with $\rho_{(k)} < \nu_0$. Hence, for the right-hand side (RHS) in (49) it follows that

$$\lim_{\substack{k\in\mathcal{A},\\ k\to\infty}} \Phi\left(\boldsymbol{x}^{(k)}\right) - \Phi\left(\boldsymbol{x}^{(k)} + \boldsymbol{s}^{(k)}\right) = 0. \tag{50}$$

The limit (48) follows from the RHS in (49) by noticing that $\theta_j = \theta_k$ for all non-successful indices $j \geq k$ with $\rho^{(j)} < \nu_0$. The bound (20) holds eventually because $\theta_k \leq \delta_{\mathrm{n}}$ for large k and then Theorem 26 applies. ◀

The next auxiliary lemma shows that the trust-region radius is bounded below if the asymptotically feasible iterates do not approach a quasi-stationary point. It is used afterwards to derive a contradiction.

▶ **Lemma 43** ([32, Lemma 3.12])**.** *Suppose the algorithm is applied to* (MOP)*, that Assumptions 5, 6, 8, 13, 15, 18, 19 and 22 hold and that the Criticality Routine does not run infinitely. Suppose further, that $|\mathcal{Z}| < \infty$ and that the doubly inexact criticality is bounded away from zero as in* (LB) *for all $k \geq k_0$. Then there is a $\Delta_{\min} > 0$ such that $\Delta_{(k)} \geq \Delta_{\min}$ for all $k \in \mathbb{N}_0$.*

Proof. We sketch how to adapt the proof in [32]: Theorem 42 ensures that for large enough k we have $\theta_k \leq \varepsilon_\theta$ and the Criticality Routine affects the criticality value and the radius. For these k it follows from (14) and (LB) that $\Delta_{(k)} \geq \min\left\{\mathsf{B}\widehat{\omega}^{(k)}, \overline{\Delta}_{(k)}\right\} \geq \min\left\{\mathsf{B}\epsilon, \overline{\Delta}_{(k)}\right\}$. Further, if $k \geq 1$ and $k-1 \notin \mathcal{R}$, then $\Delta_{(k)} \geq \min\left\{\mathsf{B}\epsilon, \gamma_0\Delta_{(k-1)}\right\}$. Suppose $k_1 \geq k_0$ is large enough that $\theta_k \leq \varepsilon_\theta$ and (40) is fulfilled for all $k \geq k_1$, i.e., $\theta_k \leq \min\{\varepsilon_\theta, \delta_\theta\}$ for all $k \geq k_1$. For the purpose of deriving a contradiction, one may now assume that $j \geq k_1$ is the first index with

$$\Delta_{(j)} \leq \gamma_0 \min\left\{\delta_\rho, \sqrt{\frac{(1-\gamma_\theta)\theta^{\mathcal{F}}}{\mathsf{c}_{\mathrm{ub}\theta}}}, \Delta_{(k_1)}, \mathsf{B}\epsilon\right\} =: \gamma_0\delta_s, \tag{51}$$

where δ_ρ is as defined in Theorem 34 and $\theta^{\mathcal{F}} := \min_{i\in\mathcal{Z}}\theta_i$ and proceed as in [32]. ◀

▶ **Lemma 44** ([32, Lemma 3.13])**.** *Suppose the algorithm is applied to* (MOP)*, that Assumptions 5, 6, 8, 13, 15, 18, 19 and 22 hold and that the Criticality Routine does not run infinitely. Suppose further, that $|\mathcal{Z}| < \infty$. Then* $\liminf_{k\to\infty} \Delta_{(k)} = 0$.

Proof. If there are only finitely many successful indices, then the result follows immediately from the radius update rules.

Thus, suppose that $|\mathcal{A}| = \infty$ and, for sake of contradiction, assume that there is a constant $\Delta_{\min} > 0$ such that

$$\Delta_{(k)} \geq \Delta_{\min} > 0 \qquad \forall\, k \in \mathbb{N}_0. \tag{52}$$

As before, we see that from Theorem 42 it follows for large k that the normal step $\boldsymbol{n}^{(k)}$ exists and satisfies (8) in Assumption 15. Then the same equations as in Theorem 42 hold, namely (49) and (50). Very much like in the proof of Theorem 41, we can again decompose the model decrease via (46). For the first term, (31) applies again, yielding $\lim_{k\to\infty}\left(\Phi^{(k)}(\boldsymbol{x}^{(k)}) - \Phi^{(k)}\left(\boldsymbol{x}_{\mathrm{n}}^{(k)}\right)\right) = 0$. From equations (49), (50) and from the model decrease decomposition it follows that

$$\lim_{\substack{k\to\infty,\\ k\in\mathcal{A}}} \left(\Phi^{(k)}\left(\boldsymbol{x}_{\mathrm{n}}^{(k)}\right) - \Phi^{(k)}\left(\boldsymbol{x}_{\mathrm{s}}^{(k)}\right)\right) = 0. \tag{53}$$

But the sufficient decrease condition (Assumption 19) still holds and (for large k and assuming (52) holds) the Criticality Routine ensures

$$\chi^{(k)} \geq \min\left\{\varepsilon_\chi, \frac{\Delta_{(k)}}{\mathtt{M}}\right\} \geq \min\left\{\varepsilon_\chi, \frac{\Delta_{\min}}{\mathtt{M}}\right\} =: \mathsf{z} > 0,$$

which can be plugged in and gives:

$$\Phi^{(k)}\left(\boldsymbol{x}_{\mathrm{n}}^{(k)}\right) - \Phi^{(k)}\left(\boldsymbol{x}_{\mathrm{s}}^{(k)}\right) \geq \mathsf{c}_{\mathrm{sd}}\mathsf{z}\min\left\{\frac{\mathsf{z}}{\mathtt{W}}, \Delta_{\min}\right\} > 0,$$

where the RHS is constant, contradicting (53). ◀

▶ **Corollary 45** ([32, Lemma 3.13]). *Suppose the algorithm is applied to* (MOP)*, that Assumptions 5, 6, 8, 13, 15, 18, 19 and 22 hold and that the Criticality Routine does not run infinitely. Suppose further, that $|\mathcal{Z}| < \infty$. Then*

$$\liminf_{k\to\infty} \chi^{(k)} = 0.$$

For a subsequence $\{k_\ell\} \subseteq \{k\}$ with $\chi^{(k_\ell)} \to 0$ it must also hold that $\lim_{\ell\to\infty} \Delta_{(k_\ell)} = 0$.

Proof. To derive a contradiction, it is assumed that $\chi^{(k)}$ is bounded away from zero as per (LB) for all $k \in \mathbb{N}_0$. Then the radius must be bounded away from zero due to Theorem 43, in contradiction to $\liminf_{k\to\infty} \Delta_{(k)} = 0$, as assured by Theorem 44. If $\{k_\ell\}$ is such that $\chi^{(k_\ell)} \to 0$, but we assume $\lim_{\ell\to\infty} \Delta_{(k_\ell)} = \Delta_{\min} > 0$, then the Criticality Routine again ensures (44) for large ℓ, which is a contradiction. ◀

▶ **Theorem 46** (Convergence to Quasi-Stationary Point). *Suppose the algorithm is applied to* (MOP) *and that Assumptions 5, 6, 8, 13, 15, 18, 19 and 22 hold. Let $\{\boldsymbol{x}^{(k)}\}$ be the sequence of iterates produced by the algorithm. Then either the restoration procedure in step 2 terminates unsuccessfully by converging to an infeasible, first-order critical point of problem* (R) *or there is a subsequence $\{k_\ell\}$ of indices for which $\lim_{\ell\to\infty} \boldsymbol{x}^{(k_\ell)} = \overline{\boldsymbol{x}}$ and $\overline{\boldsymbol{x}}$ is quasi-stationary, i.e.,*

$$\theta(\overline{\boldsymbol{x}}) = \lim_{\ell\to\infty} \theta_{k_\ell} = 0 \text{ and } \lim_{\ell\to\infty} \chi^{(k_\ell)} = 0,$$

and for which it also holds that $\lim_{\ell\to\infty} \Delta_{(k_\ell)} = 0$.

Proof. If the restoration procedure always terminates successfully, then the convergence to a quasi-stationary point follows from Assumption 22 and Theorems 29, 30, 41 and 45. ◀

5 Convergence to KKT-Points

In this section, we conclude that a quasi-stationary point is also a KKT-point under suitable constraint qualifications. In the first step, we show that we can easily replace the *objective* surrogates with their true counterparts.

▶ **Lemma 47.** *Suppose Assumptions 5, 6, 8, 13 and 15 hold and suppose that it holds for a subsequence $\{\boldsymbol{x}^{(k)}\}$ of iterates that*

$$\lim_{k\to\infty} \Delta_{(k)} = 0 \quad \text{and} \quad \lim_{k\to\infty} \widehat{\omega}_2^{(k)} = \lim_{k\to\infty} \omega\left(\boldsymbol{x}_{\mathrm{n}}^{(k)}; \widehat{\boldsymbol{f}}^{(k)}, \mathcal{L}_k, \|\bullet\|_2\right) = 0.$$

Then it also holds for the true objectives $\boldsymbol{f}$ *that*

$$\lim_{k\to\infty} \widehat{\omega}_2^{(k)} = \lim_{k\to\infty} \omega\left(\boldsymbol{x}_{\mathrm{n}}^{(k)}; \boldsymbol{f}, \mathcal{L}_k, \|\bullet\|_2\right) = 0.$$

Proof. Similar to [6, Lemma 7], we can show that for $k \in \mathbb{N}_0$ it must hold that

$$\left|\omega\left(\boldsymbol{x}_{\mathrm{n}}^{(k)}; \widehat{\boldsymbol{f}}^{(k)}, \mathcal{L}_k, \|\bullet\|_2\right) - \omega\left(\boldsymbol{x}_{\mathrm{n}}^{(k)}; \boldsymbol{f}, \mathcal{L}_k, \|\bullet\|_2\right)\right| = \left|\widehat{\omega}_2^{(k)} - \widehat{\omega}_2^{(k)}\right| \le \mathtt{s}\Delta_{(k)}$$

for some constant $\mathtt{s} > 0$ (independent of k). The triangle inequality yields

$$\left|\widehat{\omega}_2^{(k)}\right| \le \left|\widehat{\omega}_2^{(k)} - \widehat{\omega}_2^{(k)}\right| + \left|\widehat{\omega}_2^{(k)}\right| \le \mathtt{s}\Delta_{(k)} + \left|\widehat{\omega}_2^{(k)}\right|$$

and the RHS goes to zero for $k \to \infty$. ◀

Note that Theorem 47 applies both to a sequence of the main algorithm and to an infinite subsequence of the Criticality Routine (see Theorem 29). In our second step, we now replace the *constraint* surrogates with the original functions.

▶ **Lemma 48.** *Suppose Assumptions 5, 8, 13 and 15 hold. Let* $\{\boldsymbol{x}^{(k)}\} \subseteq \mathbb{R}^n$ *be an algorithmic sequence with* $\boldsymbol{x}^{(k)} \to \overline{\boldsymbol{x}} \in \mathcal{X}$, $\omega\left(\boldsymbol{x}_{\mathrm{n}}^{(k)}; \boldsymbol{f}, \mathcal{L}_k, \|\bullet\|_\infty\right) \to 0$ *and* $\Delta_{(k)} \to 0$*. Further, assume that the MFCQs hold at* $\overline{\boldsymbol{x}}$*. i.e., the rows of* $\boldsymbol{H}(\overline{\boldsymbol{x}})$ *are linearly independent, and there is a direction* $\boldsymbol{d} \in \mathbb{R}^n$ *such that* $\boldsymbol{H}(\overline{\boldsymbol{x}})\boldsymbol{d} = \boldsymbol{0}$ *and* $\boldsymbol{d}^\mathsf{T}\nabla g_{\mathrm{i}}(\overline{\boldsymbol{x}}) < 0$ *for all* i *with* $g_{\mathrm{i}}(\overline{\boldsymbol{x}}) = 0$*. Then* $\overline{\boldsymbol{x}}$ *is a KKT-point of* (MOP)*.*

Proof. In the following, we use $\widehat{\boldsymbol{H}}_k$ and $\widehat{\boldsymbol{G}}_k$ to denote the Jacobians of the surrogates $\widehat{\boldsymbol{h}}^{(k)}$ and $\widehat{\boldsymbol{g}}^{(k)}$, evaluated at $\boldsymbol{x}^{(k)}$. Likewise, $\boldsymbol{F}_k$ is the Jacobian of the true objective function at $\boldsymbol{x}^{(k)}$, while $\widehat{\boldsymbol{g}}_k \le \boldsymbol{0}_P$ is defined as $\widehat{\boldsymbol{g}}^{(k)}(\boldsymbol{x}^{(k)}) + \widehat{\boldsymbol{G}}_k \cdot \boldsymbol{n}^{(k)}$.

Assumption 15 gives $\boldsymbol{n}^{(k)} \to \boldsymbol{0}$. Because of this, and the error bounds in Assumption 13, we have that $\boldsymbol{x}^{(k)} \to \overline{\boldsymbol{x}}$ as $\Delta_{(k)} \to 0$, as well as $\widehat{\boldsymbol{H}}_k \to \boldsymbol{H} = \nabla\boldsymbol{h}(\overline{\boldsymbol{x}})$, $\widehat{\boldsymbol{G}}_k \to \boldsymbol{G} = \nabla\boldsymbol{h}(\overline{\boldsymbol{x}})$ and $\widehat{\boldsymbol{g}}_k \to \boldsymbol{g} = \boldsymbol{g}(\overline{\boldsymbol{x}})$.

By assumption, $\omega\left(\boldsymbol{x}_{\mathrm{n}}^{(k)}; \boldsymbol{f}, \mathcal{L}_k, \|\bullet\|_\infty\right) \to 0$. We thus have a sequence of linear programs,

$$\omega\left(\boldsymbol{x}_{\mathrm{n}}^{(k)}; \boldsymbol{f}, \mathcal{L}_k, \|\bullet\|_\infty\right) = \max_{\boldsymbol{d}\in\mathbb{R}^n, \beta^-\in\mathbb{R}} \begin{bmatrix}\boldsymbol{0}_n^\mathsf{T} & 1\end{bmatrix}\begin{bmatrix}\boldsymbol{d}\\ \beta^-\end{bmatrix} \quad \text{s.t.} \quad \begin{bmatrix}-\boldsymbol{I}_{n,n} & \boldsymbol{0}_n\\ \boldsymbol{I}_{n,n} & \boldsymbol{0}_n\\ \boldsymbol{F}_k & \boldsymbol{1}_K\\ \widehat{\boldsymbol{H}}_k & \boldsymbol{0}_M\\ \widehat{\boldsymbol{G}}_k & \boldsymbol{0}_P\end{bmatrix}\begin{bmatrix}\boldsymbol{d}\\ \beta^-\end{bmatrix} \begin{matrix}\le\\ \le\\ \le\\ =\\ \le\end{matrix} \begin{bmatrix}\boldsymbol{1}_n\\ \boldsymbol{1}_n\\ \boldsymbol{0}_K\\ \boldsymbol{0}_M\\ -\widehat{\boldsymbol{g}}_k\end{bmatrix},$$

the values of which go to zero. The dual problems are

$$\min_{\boldsymbol{y}^1,\boldsymbol{y}^2,\boldsymbol{y}^3,\boldsymbol{y}^5\ge\boldsymbol{0},\boldsymbol{y}^4\in\mathbb{R}^M} \begin{bmatrix}\boldsymbol{1}_n^\mathsf{T} & \boldsymbol{1}_n^\mathsf{T} & \boldsymbol{0}_K^\mathsf{T} & \boldsymbol{0}_M^\mathsf{T} & -\widehat{\boldsymbol{g}}_k^\mathsf{T}\end{bmatrix}\begin{bmatrix}\boldsymbol{y}^1\\ \vdots\\ \boldsymbol{y}^5\end{bmatrix} \quad \text{s.t.} \quad \begin{bmatrix}-\boldsymbol{I}_{n,n} & \boldsymbol{I}_{n,n} & \boldsymbol{F}_k^\mathsf{T} & \widehat{\boldsymbol{H}}_k^\mathsf{T} & \widehat{\boldsymbol{G}}_k^\mathsf{T}\\ \boldsymbol{0}_n^\mathsf{T} & \boldsymbol{0}_n^\mathsf{T} & \boldsymbol{1}_K^\mathsf{T} & \boldsymbol{0}_M^\mathsf{T} & \boldsymbol{0}_P^\mathsf{T}\end{bmatrix}\begin{bmatrix}\boldsymbol{y}^1\\ \vdots\\ \boldsymbol{y}^5\end{bmatrix} = \begin{bmatrix}\boldsymbol{0}_n\\ 1\end{bmatrix}. \tag{D$_k$}$$

By strong duality, the dual problem for $k \in \mathbb{N}_0$ is also always feasible, and its optimal value equals the primal optimal value. Suppose that $\left\{(\boldsymbol{y}_k^1, \ldots, \boldsymbol{y}_k^5)\right\}$ is a sequence of dual optimizers. By strong duality, it follows from $\omega\left(\boldsymbol{x}_{\mathrm{n}}^{(k)}; \boldsymbol{f}, \mathcal{L}_k, \|\bullet\|_\infty\right) \to 0$ that

$$\lim_{k\to\infty} \boldsymbol{y}_k^1 = \boldsymbol{0}, \quad \lim_{k\to\infty} \boldsymbol{y}_k^2 = \boldsymbol{0}, \quad \text{and} \quad \lim_{k\to\infty} -\widehat{\boldsymbol{g}}_k^\mathsf{T}\boldsymbol{y}_k^5 = 0.$$

First, we show that it must also hold for $i \in \{3, 4, 5\}$ that the sequences $\{\boldsymbol{y}^i\}$ are bounded (see also [26] for a similar idea). To derive a contradiction, assume that for some $i \in \{3, 4, 5\}$ the sequence $\{\boldsymbol{y}_k^i\}$ is not bounded. Define $\nu_k = \max\left\{1, \max_{i=3,4,5} \left\|\boldsymbol{y}_k^i\right\|_\infty\right\}$, and we get a bounded sequence of scaled variables

$$\widetilde{y}_k^i = \frac{1}{\nu_k} y_k^i \qquad i = 1, \ldots, 5.$$

Then, $\nu_k \xrightarrow{k\to\infty} \infty$, but we can take a subsequence $\mathcal{K}$ of indices such that for every $i \in \{3,4,5\}$ the sequence $\{\widetilde{\boldsymbol{y}}_k^i\}_{k\in\mathcal{K}}$ converges to $\overline{\boldsymbol{y}}^i$ and so that for one $i \in \{3,4,5\}$ the limit $\overline{\boldsymbol{y}}^i$ is not zero. Consequently, we have

$$\lim_{\substack{k\to\infty,\\ k\in\mathcal{K}}} (\widetilde{\boldsymbol{y}}_k^1, \ldots, \widetilde{\boldsymbol{y}}_k^5) = (\overline{\boldsymbol{y}}^1, \ldots, \overline{\boldsymbol{y}}^5) = (\mathbf{0}_n, \mathbf{0}_n, \overline{\boldsymbol{y}}^3, \overline{\boldsymbol{y}}^4, \overline{\boldsymbol{y}}^5), \quad \text{and} \quad \lim_{\substack{k\to\infty,\\ k\in\mathcal{K}}} -\widehat{\boldsymbol{g}}_k^\mathsf{T} \widetilde{\boldsymbol{y}}_k^5 = -\boldsymbol{g}^\mathsf{T} \overline{\boldsymbol{y}}^5 = 0. \tag{54}$$

For any dual feasible $[\boldsymbol{y}_k^1, \ldots, \boldsymbol{y}_k^5]$, the vector $[\widetilde{\boldsymbol{y}}_k^1, \ldots, \widetilde{\boldsymbol{y}}_k^5]$ is feasible for the problem

$$\min_{\tilde{\boldsymbol{y}}^1,\tilde{\boldsymbol{y}}^2,\tilde{\boldsymbol{y}}^3,\tilde{\boldsymbol{y}}^5\geq\mathbf{0},\tilde{\boldsymbol{y}}^4\in\mathbb{R}^M} \begin{bmatrix}\mathbf{1}_n^\mathsf{T} & \mathbf{1}_n^\mathsf{T} & \mathbf{0}_K^\mathsf{T} & \mathbf{0}_M^\mathsf{T} & -\widehat{\boldsymbol{g}}_k^\mathsf{T}\end{bmatrix} \begin{bmatrix}\widetilde{\boldsymbol{y}}^1\\ \vdots \\ \widetilde{\boldsymbol{y}}^5\end{bmatrix} \quad \text{s.t.} \quad \begin{bmatrix}\boldsymbol{I}_{n,n} & \boldsymbol{I}_{n,n} & \boldsymbol{F}_k^\mathsf{T} & \widehat{\boldsymbol{H}}_k^\mathsf{T} & \widehat{\boldsymbol{G}}_k^\mathsf{T}\\ \mathbf{0}_n^\mathsf{T} & \mathbf{0}_n^\mathsf{T} & \mathbf{1}_K^\mathsf{T} & \mathbf{0}_M^\mathsf{T} & \mathbf{0}_P^\mathsf{T}\end{bmatrix} \begin{bmatrix}\widetilde{\boldsymbol{y}}^1\\ \vdots \\ \widetilde{\boldsymbol{y}}^5\end{bmatrix} = \begin{bmatrix}\mathbf{0}_n\\ 1/\nu_k\end{bmatrix}.$$

In the limit, the last constraint becomes $\mathbf{1}_K^\mathsf{T} \cdot \overline{\boldsymbol{y}}^3 = 0$, the limiting problem is

$$\min_{\tilde{\boldsymbol{y}}^1,\tilde{\boldsymbol{y}}^2,\tilde{\boldsymbol{y}}^3,\tilde{\boldsymbol{y}}^5\geq\mathbf{0},\tilde{\boldsymbol{y}}^4\in\mathbb{R}^M} \begin{bmatrix}\mathbf{1}_n^\mathsf{T} & \mathbf{1}_n^\mathsf{T} & \mathbf{0}_K^\mathsf{T} & \mathbf{0}_M^\mathsf{T} & -\boldsymbol{g}^\mathsf{T}\end{bmatrix} \begin{bmatrix}\widetilde{\boldsymbol{y}}^1\\ \vdots \\ \widetilde{\boldsymbol{y}}^5\end{bmatrix} \quad \text{s.t.} \quad \begin{bmatrix}\boldsymbol{I}_{n,n} & \boldsymbol{I}_{n,n} & \boldsymbol{F}^\mathsf{T} & \boldsymbol{H}^\mathsf{T} & \boldsymbol{G}^\mathsf{T}\\ \mathbf{0}_n^\mathsf{T} & \mathbf{0}_n^\mathsf{T} & \mathbf{1}_K^\mathsf{T} & \mathbf{0}_M^\mathsf{T} & \mathbf{0}_P^\mathsf{T}\end{bmatrix} \begin{bmatrix}\widetilde{\boldsymbol{y}}^1\\ \vdots \\ \widetilde{\boldsymbol{y}}^5\end{bmatrix} = \begin{bmatrix}\mathbf{0}_n\\ 0\end{bmatrix}. \tag{D$'$}$$

The primal thus has constant objective value:

$$\max_{\boldsymbol{d}\in\mathbb{R}^n,\beta^-\in\mathbb{R}} 0 \qquad \text{s.t.} \qquad \begin{bmatrix}-\boldsymbol{I}_{n,n} & \mathbf{0}_n\\ \boldsymbol{I}_{n,n} & \mathbf{0}_n\\ \boldsymbol{F} & \mathbf{1}_K\\ \boldsymbol{H} & \mathbf{0}_M\\ \boldsymbol{G} & \mathbf{0}_P\end{bmatrix} \begin{bmatrix}\boldsymbol{d}\\ \beta^-\end{bmatrix} \begin{matrix}\leq\\ \leq\\ \leq\\ =\\ \leq\end{matrix} \begin{bmatrix}\mathbf{1}_n\\ \mathbf{1}_n\\ \mathbf{0}_K\\ \mathbf{0}_M\\ -\boldsymbol{g}\end{bmatrix}. \tag{P$'$}$$

By upper semi-continuity of the feasible set mappings (see [73]), $\overline{\boldsymbol{y}}$ is feasible for (D$'$), and we see from (54) and (P$'$) that it is also optimal. Let $\boldsymbol{d} \in \mathbb{R}^n$ be a direction with $\|\boldsymbol{d}\|_\infty \leq 1$ adhering to the MFCQs. Furthermore, let β be such that $(\boldsymbol{d}, \beta)$ is feasible for (P$'$). Then $(\boldsymbol{d}, \beta)$ is optimal for (P$'$), and we make the following observations:

- $\overline{\boldsymbol{y}}^3$ **must be zero,** because of the second constraint in (D$'$).
- $\overline{\boldsymbol{y}}^5$ **must be zero:** Because of the MFCQs it holds that $\boldsymbol{G}_{[\mathrm{i},:]} \cdot \boldsymbol{d} < 0$ whenever $g_\mathrm{i} = 0$. By complementary slackness, it follows for these indices i that the entries in $\overline{\boldsymbol{y}}^5$ must be 0. From (54) it also follows for the other indices that their values must be 0.
- But with $\overline{\boldsymbol{y}}^4 \neq \mathbf{0}$ it then follows from the first constraint in (D$'$) that $\boldsymbol{H}^\mathsf{T} \cdot \overline{\boldsymbol{y}}^4 = \mathbf{0}$ in contradiction to the MFCQs.

Hence, for all $i \in \{3,4,5\}$ the sequence $\{\boldsymbol{y}_k^i\}$ of (unscaled) Lagrange multipliers must also be bounded!

We thus can take a subsequence $\mathcal{K}$ so that $\{\boldsymbol{y}_k\}_{k\in\mathcal{K}}$ converges to some $(\mathbf{0}, \mathbf{0}, \overline{\boldsymbol{y}}_k^3, \overline{\boldsymbol{y}}_k^4, \overline{\boldsymbol{y}}^5)$ and (by upper semi-continuity of the feasible set mapping) the limit point is feasible for the limiting problem of (D$_k$), which happens to be (D). The optimal value at $(\mathbf{0}, \mathbf{0}, \overline{\boldsymbol{y}}_k^3, \overline{\boldsymbol{y}}_k^4, \overline{\boldsymbol{y}}_k^5)$ is 0 due to strong duality. Because the corresponding primal is (P), it follows from Theorem 4 that $\overline{\boldsymbol{x}}$ is a KKT point of (MOP). ◀

With Theorem 4, it is now easy to derive the main result:

▶ **Theorem 49** (Convergence to KKT-points). *Suppose the same assumptions as in Theorem 46 hold and that* $\{\boldsymbol{x}^{(k)}\}$ *is a quasi-stationary subsequence with limit point* $\overline{\boldsymbol{x}}$*. If the MFCQs hold at* $\overline{\boldsymbol{x}}$*, then* $\overline{\boldsymbol{x}}$ *is a KKT-point of* (MOP).

6 Numerical Examples

In this section, we provide numerical examples in order to evaluate the strengths and weaknesses of our algorithm. There are two examples using an implementation close to the description in Section 3. Additionally, we give comparisons with the `DFMO` algorithm [50] on the `TESTMO` set of test problems. This comparison is done not with the main algorithm, but with algorithms derived from it in order to generate multiple solutions.

6.1 Implementation Details

All algorithms are available in the research package `Compromise.jl` [5] in the Julia language. The code to generate the experiments can be found in a separate GitHub repository.[1] It also contains utilities to interact with the Fortran implementations of `DFMO` and `TESTMO` provided by the authors of [50].

Trust-regions are boxes, but we allow for the use of other norms in the objective of ($\text{ITRN}^{(k)}$) and the constraint in ($\text{ITRT}^{(k)}$). In fact, by default linear norms are used because this appears to improve the quality of solutions with the inner solver `HiGHS` [43]. For restoration, we currently use `COBYLA` [63] from the `NLopt` package [45].

The version of `Compromise.jl` used in this demonstration allows for RBF surrogate models according to the construction in [74]. It is also possible to use first or second degree Taylor models. These need derivative information, which we provide by means of finite difference approximations using `FiniteDiff.jl`.

The plots are generated with `Makie.jl` [18].

6.2 Constrained Two-Parabolas Problem

Variations of the two parabolas problem are popular test cases for MOO methods. The objectives are simply two n-variate quadratic functions and (in the unconstrained case) the Pareto Set is the line connecting their respective minima.

6.2.1 Inequality Constraint

The following constrained version is taken from [38].

$$\min_{\boldsymbol{x}\in\mathbb{R}^2} \begin{bmatrix} (x_1-2)^2+(x_2-1)^2 \\ (x_1-2)^2+(x_2+1)^2 \end{bmatrix} \quad \text{s.t.} \quad g(\boldsymbol{x}) = 1 - x_1^2 - x_2^2 \le 0. \tag{TPineq}$$

The feasible set of (TPineq) is $\mathbb{R}^2$ without the interior of the $\|\bullet\|_2$ unit ball, so it is non-convex. The Pareto critical set is the line connecting $[2,-1]$ and $[2,1]$ and the left boundary of the unit ball:

$$\mathcal{P}_c = \left\{ \begin{bmatrix} 2 \\ s \end{bmatrix} : s \in [-1,1] \right\} \bigcup \left\{ \begin{bmatrix} \cos t \\ \sin t \end{bmatrix} : t \in [\pi-\theta, \pi+\theta], \theta = \arctan\left(\frac{1}{2}\right) \right\}.$$

We have plotted the critical set with black lines in Figure 1.

For this problem, we first apply our algorithm three times with the same parameters and beginning at $\boldsymbol{x}_0 = [-2, 0.5]$, but with different model types. We compare RBF surrogate models with first and second degree Taylor polynomials of the objective and constraint functions. The RBF models use the cubic radial function $r \mapsto r^3$. The derivatives for the Taylor models are approximated using finite differences (to conform to the assumption that gradients are not available exactly). The algorithm parameters are

$$\begin{aligned} &\Delta_{(0)} = .5, \Delta_{\max} = 16, \gamma_0 = 0.25, \gamma_1 = 0.5, \gamma_2 = 2.0, \gamma_\theta = 0.1, \nu_0 = 0.01, \varepsilon_\chi = 10^{-4}, \varepsilon_\theta = 10^{-6}, \\ &\qquad \kappa_\theta = 10^{-4}, \psi = 2, \mathtt{B} = 1000, \mathtt{M} = 3000, \alpha = 0.5, \mathtt{c}_\Delta = 0.99, \mathtt{c}_\mu = 100, \mu = 0.01. \end{aligned} \tag{55}$$

The parameter δ_n is not required for computations, but assumed to have a value $\delta_\text{n} \ge \varepsilon_\theta$. These settings largely follow the recommendations in [32]. For these runs, the success threshold is $\nu_1 = 0.45$. We stop after 100 iterations or if the trust region radius is reduced to below 10^{-9}. Afterwards, we change the starting point to $x_0 = [-2, -0.5]$ and the success threshold to $\nu_1 = 0.9$ for three runs with the same kinds of surrogates. Finally, there is also a run starting at $x_0 = [-2, 0]$ with $\nu_1 = .45$.

The results are depicted in Figure 1. As can be seen, all runs converge to a point on the Pareto critical set and avoid the infeasible area. In terms of objective evaluations, the RBF models need the fewest function calls. In terms of iterations, the second degree Taylor polynomials perform best.

Almost all solutions lie on the connecting line of the individual objective minima. The only run converging to the left part of the critical set started out close to it. Of course, this behavior is in part caused by the problem geometry and the surrogate construction methods. But if our wish is to find non-dominated points rather than the superset of critical points, we benefit from the descent nature of the method.

---

[1] `https://github.com/manuelbb-upb/MOBenchmarks`

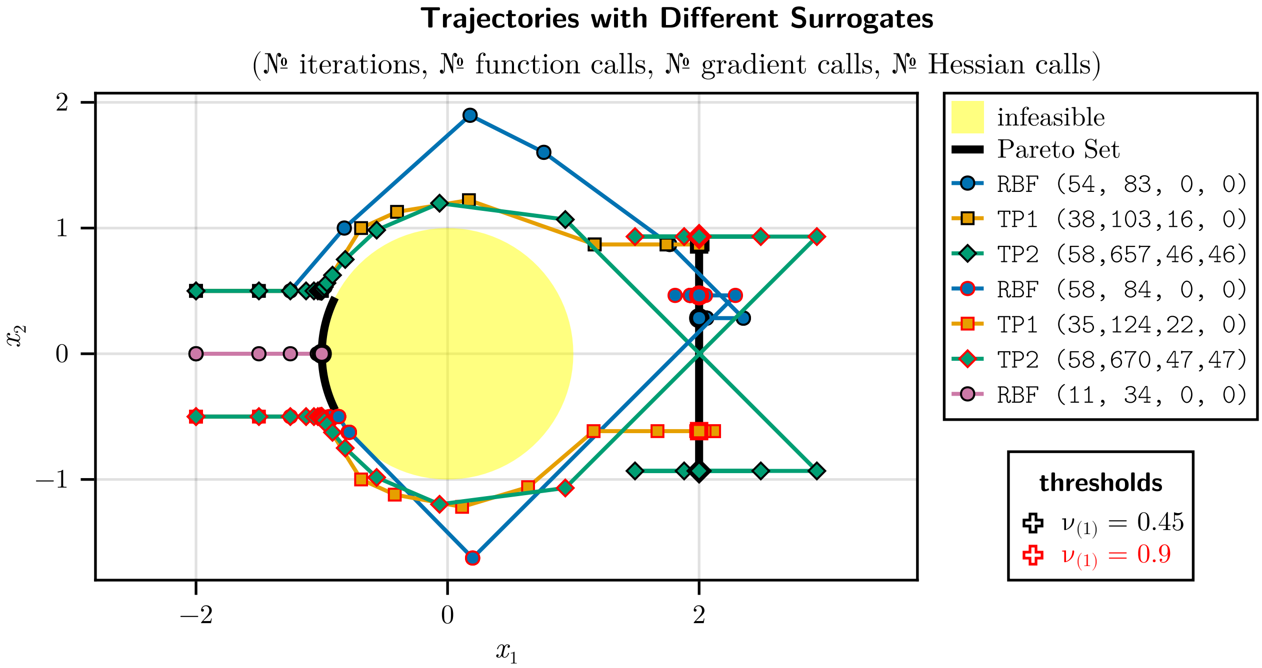

Figure 1 Optimization trajectories with different surrogate types for (TPineq). The upper legend shows the number of function calls in parentheses. Different strokecolors correspond to different values of the threshold parameter ν_1.

Compared to the Taylor models, the RBF models sometimes take large steps that do not necessarily reduce both objectives to the same degree. This is likely due to them providing more inexact gradients. Moreover, using $\|\bullet\|_\infty$ in (ITRT$^{(k)}$) induces a bias towards the corners of the unit ball.

We can also see the zig-zag motion close to critical points that is typical for gradient-based algorithm. It is very pronounced for second degree Taylor models. The Taylor models of degree 2 can be considered almost exact for (TPineq). Hence, the trial points achieve the predicted objective decrease and nearly every iteration is deemed successful, which makes the trust-regions have large radius and practically reduces our method to "vanilla" gradient-descent.

Finally, it is worthwile to compare the iteration trajectories to those in [38], where an active set strategy is used. In contrast to the results presented there, we stay further away from the infeasible area. A possible explanation lies in the fact that (in our algorithm) steps are computed with inexact approximations of *every* constraint function, instead of only considering the active constraints.

6.2.2 Equality Constraint

We can take the same objectives and turn the constraint into an equality constraint:

$$\min_{\boldsymbol{x}\in\mathbb{R}^2} \begin{bmatrix}(x_1-2)^2+(x_2-1)^2\\(x_1-2)^2+(x_2+1)^2\end{bmatrix} \quad \text{s.t.} \quad g(\boldsymbol{x}) = 1 - x_1^2 - x_2^2 = 0. \tag{TPeq}$$

Now, the feasible set is the $\|\bullet\|_2$ unit sphere. Subsequently, the connecting line of individual objective minima cannot be optimal anymore. Analysis shows that the Pareto critical set now consists of two segments of the unit sphere:

$$\mathcal{P}_c = \left\{ \begin{bmatrix}\cos t\\ \sin t\end{bmatrix} : t \in [\pi-\theta, \pi+\theta] \cup [2\pi-\theta, 2\pi+\theta],\ \theta = \arctan\left(\frac{1}{2}\right) \right\}.$$

We mostly keep the parameters (55) and set $\nu_1 = 0.9$. This time, for different starting points, we vary the initial trust region radius and the compatibility parameter $\mathtt{c}_\Delta$. Doing so allows us to illustrate the effect of these parameters on the restoration procedure (amongst others) by comparing trajectories and pairs of trajectories.

The results are depicted in Figure 2. First, this example illustrates well how restoration steps return the iterations back close to the feasible set. Secondly, we again notice a tendency towards the objective minima;

only the green trajectory ends on the left part of the Pareto Set. This is due to the significantly smaller initial trust region radius. Comparing the blue trajectory with the others confirms that a greater $\mathtt{c}_\Delta$ value is more permissive with respect to declaring the normal step compatible. Specifically, the purple trajectory has been set up symmetrically to the blue one. But due to $\mathtt{c}_\Delta$ being smaller, purple starts with a restoration right away. Orange starts near blue and can avoid initial restoration because it starts closer to the feasible set. But as with purple there are noticeably more restoration iterations overall, compared to blue. Comparing orange and green highlights that "compatibility" is a function not only of $\mathtt{c}_\Delta$ but of $\Delta_{(k)}$ as well. That is why for green, with the smaller initial radius, the first iteration is a restoration.

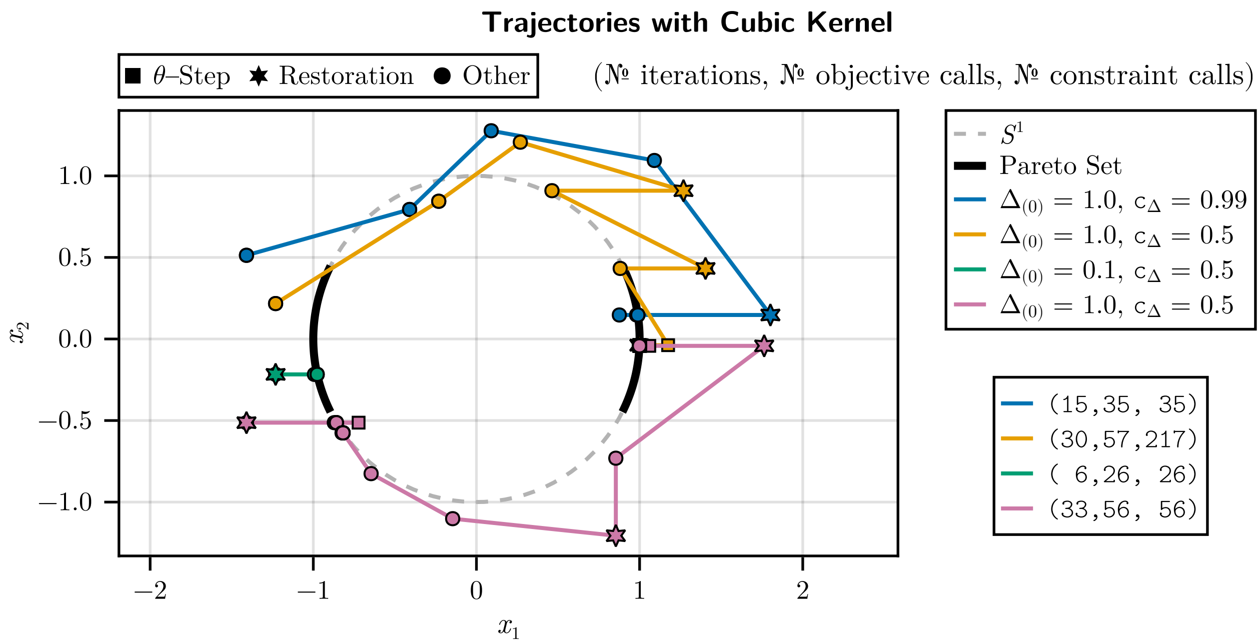

Figure 2 Four optimization trajectories for (TPeq) with cubic RBF surrogates and varying initial trust region radii and compatibility parameters.

6.3 Non-Convex Test-Problem "W3"

The second example is problem "W3" taken from [52]. The problem has box constraints and two non-linear inequality constraints. Whilst the unconstrained Pareto Front is simply a line in objective space, the constraints make it partially non-convex. We have chosen the problem with 3 variables and 2 objectives to compare our algorithm against a simple weighted-sum scalarization. It reads:

$$
\begin{aligned}
&\min_{\boldsymbol{x}\in[0,1]^3} \begin{bmatrix} f_1(\boldsymbol{x}) \\ f_2(\boldsymbol{x}) \end{bmatrix} = \min_{\boldsymbol{x}\in[0,1]^3} \begin{bmatrix} x_1 \\ d(\boldsymbol{x}) \cdot \left(1 - \frac{f_1(\boldsymbol{x})}{d(\boldsymbol{x})}\right) \end{bmatrix} \qquad \text{s.t.} \\
&c_1(\boldsymbol{x}) = f_1(\boldsymbol{x}) + f_2(\boldsymbol{x}) - 1.05 - 0.45\sin(0.75\pi \cdot l(\boldsymbol{x}))^6 \le 0, \\
&c_2(\boldsymbol{x}) = -f_1(\boldsymbol{x}) - f_2(\boldsymbol{x}) + 0.85 + 0.3\sin(0.75\pi \cdot l(\boldsymbol{x}))^2 \le 0, \\
&d(\boldsymbol{x}) = 1 + 2(x_2 + (x_1 - 0.5)^2 - 1)^2 + 2(x_3 + (x_2 - 0.5)^2 - 1)^2, \\
&l(\boldsymbol{x}) = \sqrt{2}(f_2(\boldsymbol{x}) - f_1(\boldsymbol{x})).
\end{aligned} \tag{MW3}
$$

Because the constraints c_1 and c_2 describe the feasible set in terms of the objective values we can show the attainable objective values in objective space in Figure 3. The critical set is again shown in black.

For 15 fixed starting points, we apply 3 different algorithms:

- Our trust-region algorithm `Compromise.jl` with default settings,
- a modified version of `Compromise.jl`, with normalized gradients and requiring strict descent,
- and a weighted-sum minimization of $^1/_2 f_1 + {}^1/_2 f_2$ with `COBYLA`.

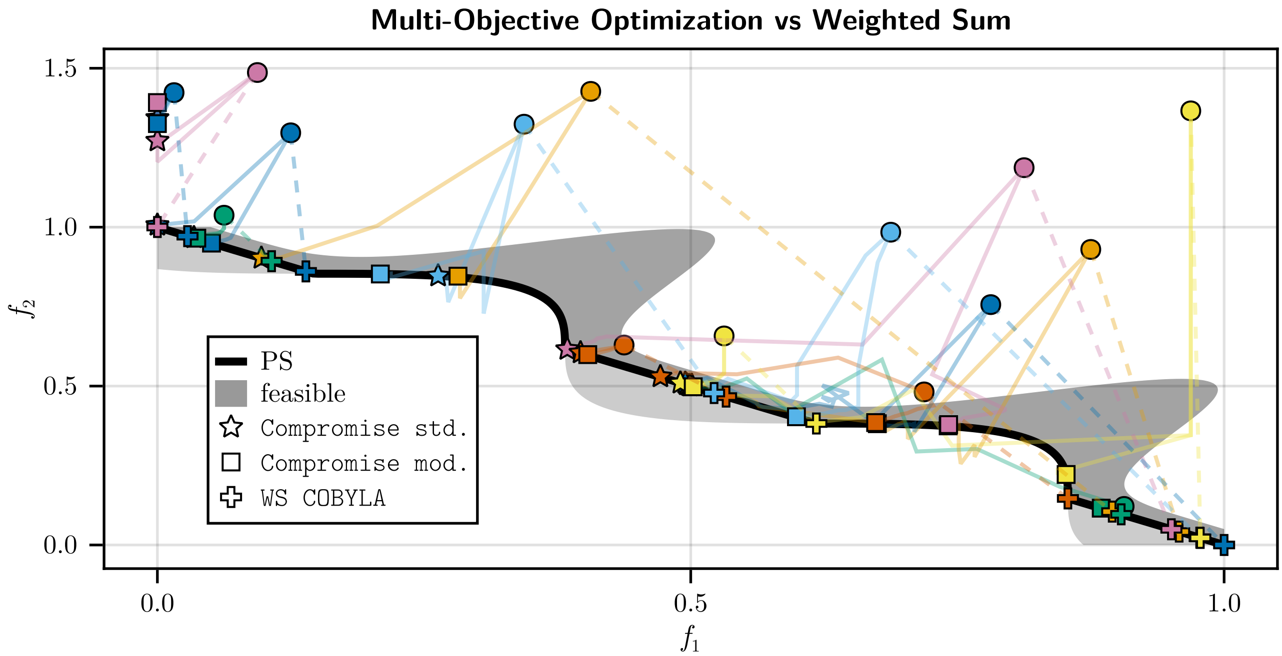

Figure 3 Results for the MW3 problem and comparison with the weighted-sum approach. Different marker symbols refer to different optimization techniques. Each initial objective vector is connected to the solution vector by a thin line. For `Compromise.jl`, the lines represent the image space trajectories. The gray area show all objective vectors that are feasible if we read the constraints as functions of f_1 and f_2. The darker area actually has valid pre-images.

As we can see from the results in Figure 3, the weighted-sum approach is only able to find convex parts of the Pareto Front. Both multi-objective descent techniques are able to find solutions on the non-convex parts. The standard version of `Compromise.jl` has only one solution in the non-convex region. In the standard version, we allow partial descent. By requiring strict descent, the modified version does not "run away" from the non-convex parts. Additionally, normalizing gradients before computing the steepest descent direction centers the steepest descent direction. Nonetheless, both versions of `Compromise.jl` show a typical phenomenon of gradient-based multi-objective optimization techniques: Some solutions are on the boundary and only weakly optimal.

6.4 Comparison with `DFMO` on `TESTMO` Suite

6.4.1 Solvers

The solver `DFMO` is a Fortran implementation of a derivative-free multi-objective optimization algorithm for problems with box constraints and nonlinear inequality constraints [50]. The code is available under an open license[2], which makes it a convenient candidate to test our algorithm against. `DFMO` combines a penalty approach with inexact line-search along quasi-random and coordinate directions to find a set of critical solutions. However, our method produces only a single critical output. We thus designed various "outer" algorithms based on the same principles.

The *sequential* outer algorithm starts with a set of initial points and sequentially performs inner updates according to the main algorithm from Section 3. That is, for each point, first a surrogate is constructed, and then we compute an inexact normal step and an inexact descent step. It is very similar to just executing multiple runs with the main algorithm, but allows us to share the evaluation database in a more effective manner. We refer to the sequential outer algorithm as `CSEQ`. A major drawback of `CSEQ` becomes obvious if nonlinear constraints are present: A single restoration phase can easily use up any computational budget.

[2] `https://github.com/DerivativeFreeLibrary/DFMO`

That is why we also test `CPEN`, the sequential outer algorithm applied to the box-constrained problem with penalty-augmented objectives

$$Z_\mathfrak{l}(\boldsymbol{x}) = f_\mathfrak{l}(\boldsymbol{x}) + \frac{1}{\varepsilon}\sum_{\mathfrak{i}=1}^{P}\max\{0, g_\mathfrak{i}(\boldsymbol{x})\} \qquad \text{for some } \varepsilon > 0.$$

The experiments are performed with $\varepsilon = 0.1$. Note that `CPEN` does no longer satisfy the assumption that the objectives are smooth. In between outer iterations we may perform non-dominance testing to not waste time on dominated solutions.

With `CPEN` we abandon the composite step approach, as no normal steps are required. But it seems reasonable that normal steps are beneficial to finding feasible solutions by exploiting the local structure of the feasible set. For this reason, we thought of a *set-based* outer algorithm, `CSET`. In contrast to `CSEQ`, more information between solutions is shared and we try to avoid restoration as much as possible. A high-level description of `CSET` is given in the Appendix, Section 8.3. Unfortunately, there is no convergence theory for it yet.

Compared to `DFMO`, all our algorithms suffer from a lack of *exploration* mechanisms. Any exploration happens by accident due to inexact surrogates. We try to compensate by starting with large initial populations.

6.4.2 Setup and Metrics

We use the test problem library `TESTMO` and largely follow the setup in [50]. From `TESTMO`, we take those 51 multi-objective problems with 3 or more variables. All problems have finite box constraints. We also generate 214 test problems with non-linear constraints by augmenting each of the 51 problems with one of six constraint functions provided in [50] and discarding infeasible problems. Each solver has a budget of 2000 evaluations. `DFMO` is run with default settings. The algorithms based on `Compromise.jl` have small acceptance threshold $\nu_0 = 10^{-4}$ but a large success threshold $\nu_1 = 0.9$. All functions are modelled with cubic RBF surrogates. For the `Compromise.jl` algorithms, we use the first $\min\{500, 100n\}$ Halton points as starting points.

We present the results of our experiments by means of performance profiles [17, 22] for performance indicators derived from different metrics. Loosely speaking, we define a performance indicator as a value $t_{p,s}$ that is lower the better solver $s \in \textbf{Solvers}$ performs on problem $p \in \textbf{Problems}$. The values are then transformed to obtain normalized performance values $r_{p,s}$. Finally, we plot the distribution functions

$$\rho_s(\tau) = \frac{1}{|\textbf{Problems}|}\left|\{p \in \textbf{Problems} : r_{p,s} \le \tau\}\right|,$$

starting at the lowest possible τ. Usually, $r_{p,s}$ is the performance *ratio*, obtained by dividing $t_{p,s}$ by the best possible value obtained among all solvers, i.e.,

$$r_{p,s} = \frac{t_{p,s}}{\min\{t_{p,s} :\ s \in \textbf{Solvers}\}}. \tag{T1}$$

Then, $\tau \ge 1$ and $\rho_s(1)$ is the percentage of problem instances for which s was the best. The solver s with highest $\rho_s(1)$ value is the best overall, whilst high values $\rho_s(\tau)$ for $\tau > 1$ indicate solver robustness. However, if $t_{p,s}$ can become 0, we instead use a min-max-normalization, i.e.,

$$r_{p,s} = \begin{cases} 0 & \text{if } \bar{t}_p - \underline{t}_p = 0,\\ \frac{t_{p,s}-\underline{t}_p}{\bar{t}_p-\underline{t}_p} & \text{otherwise,}\end{cases} \qquad \underline{t}_p = \min\{t_{p,s} :\ s \in \textbf{Solvers}\}, \bar{t}_p = \max\{t_{p,s} :\ s \in \textbf{Solvers}\}. \tag{T2}$$

Subsequently, $r_{p,s} \in [0,1]$ and $\tau \in [0,1]$, and $r_{p,s}$ relates the performance of s to both the best and worst solver for problem instance p.

From [17] we take the purity metric, and the spread metrics Γ and Δ. All these metrics require a reference front $\mathcal{Y}_p$ or extreme points. We build the reference front $\mathcal{Y}_p$ by taking the union of the approximate fronts $\mathcal{Y}_{p,s}$ returned by all solvers and removing the dominated points from this set.

The purity of a solver on a problem is the percentage of solutions of this solver that are also in the reference front. To make it suitable for performance profiles, $t_{p,s}$ is the reciprocal of that fraction, i.e., $t_{p,s} = |\mathcal{Y}_p|/|\mathcal{Y}_{p,s}\cap\mathcal{Y}_p|$, and $r_{p,s}$ is obtained from (T1).

The Γ metric relates to the maximum inter-solution distance within front approximations. The metric Δ is an indicator for how well approximate solution value vectors are distributed. In both cases $t_{p,s}$ can become zero, thus (T2) is used.

We consider two more metrics. The hyperarea ratio [3] is an indicator of how much volume is dominated by a solver front compared to the reference front. Like with the purity, we take the reciprocal and use (T1). The reference point for the hypervolume computation is constructed like in [9].

Moreover, we take the median of the constraint violation values of approximate solutions as a performance indicator, as well as the 3rd quartile values. In this case, we measure constraint violation as $\sum_i \max\{0, g_i(\boldsymbol{x})\}$. These values are normalized using (T2).

For every indicator except "constraint violation", we only take into consideration those solutions where the constraint violation is below 10^{-3}.

6.4.3 Results and Discussion

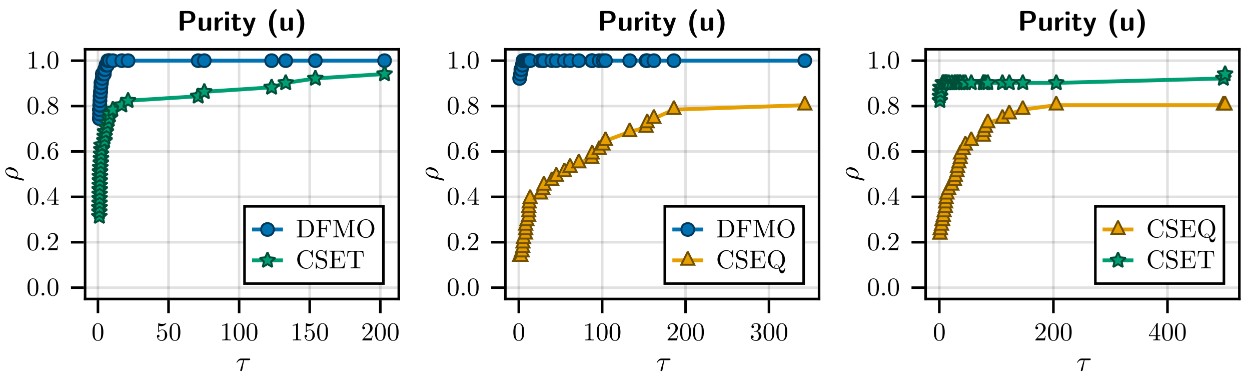

Figure 4 Pairwise purity comparison on unconstrained problems with $\tau \geq 1$.

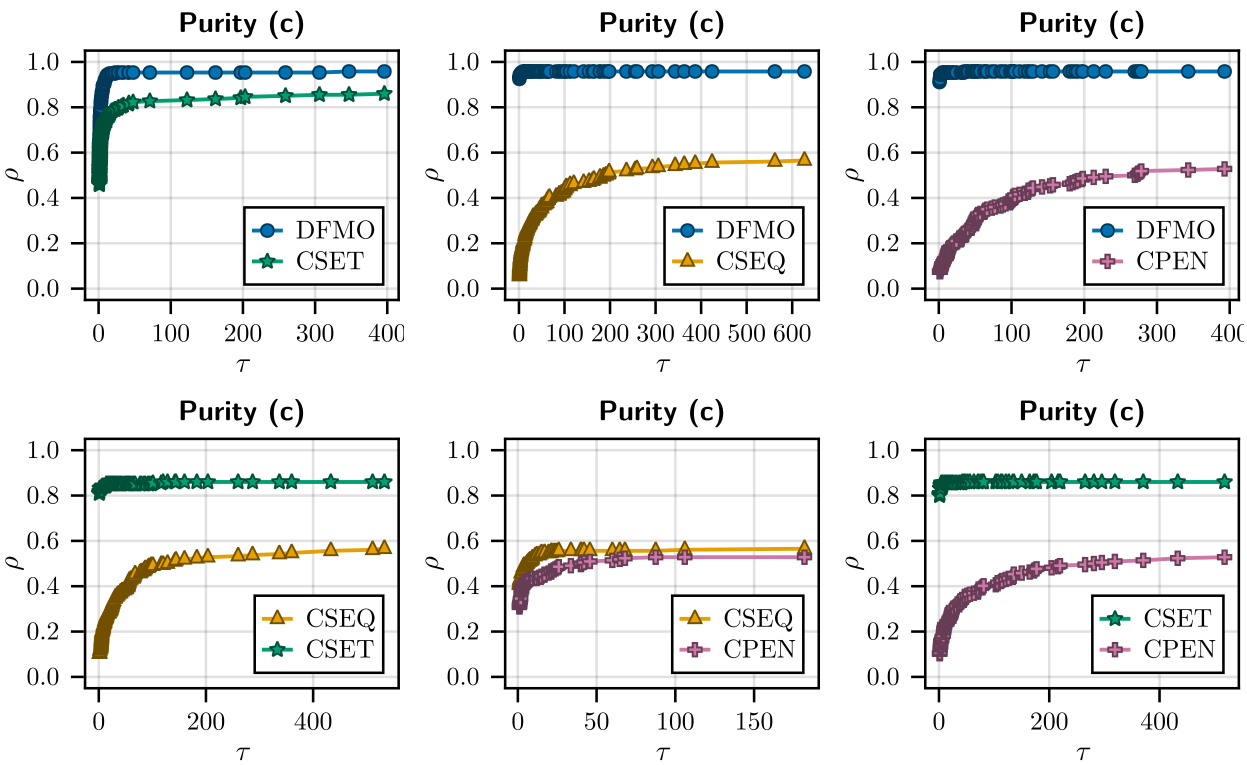

Figure 5 Pairwise purity comparison on constrained problems with $\tau \geq 1$.

As can be seen in Figure 4 and Figure 5, `DFMO` consistently outperforms all other solvers. The sequential outer algorithms seem to do very bad. Fortunately, `CSET` appears to find solutions that have better quality. It is still outperformed by `DFMO`, which might be due to `CSET` being more prone to local minima.

The plots for the hyperarea ratio indicator in Figure 6 confirm that `DFMO` produces the best solutions with `CSET` as the runner-up.

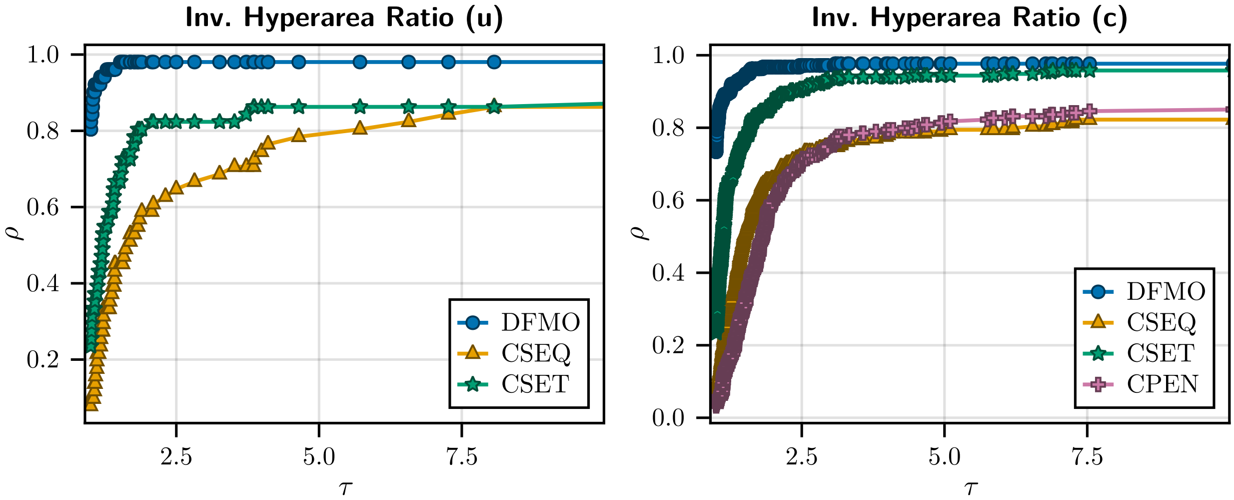

Figure 6 Performance based on the inverse hyperarea ratio for constrained (c) and unconstrained (u) problems with $\tau \geq 1$.

A similar trend is seen with the Γ spread in Figure 7, both in the constrained and unconstrained case. Again, `DFMO` outperforms all other solvers. Due to the quasi-random directions it uses, and its expansive line-search routine, it has much better exploration capabilities than the `Compromise.jl` algorithms.
At first glance, the sequential outer algorithms `CSEQ` and `CPEN` seem to do better regarding the Δ spread. But this is a fluke. The sequential algorithms usually find way fewer (and worse) solutions, whilst `DFMO` and `CSET` obtain more solutions that are sometimes clustered, with gaps in-between clusters. These gaps result in high Δ values, even if the cluster-gaps for `DFMO` or `CSET` are smaller than inter-solution gaps for `CSEQ` or `CPEN`.

Finally, both `DFMO` and `CSET` find mostly feasible solutions, see Figure 8. Surprisingly, `CPEN` is not too bad either. On the other hand, `CSEQ` behaves as expected. Likely, much of the budget is spent on restoration a single solution, or very few solutions, so that other points remain infeasible.

7 Conclusion and Outlook

In this article, we have presented an algorithm for non-linearly constrained MOPs. The method does not necessarily need derivative information of the objective or constraint functions but can use surrogate models instead. Additionally, a filter ensures convergence towards feasibility. We have proven convergence of an algorithmic subsequence to Pareto-criticality and confirmed the theoretical results with numerical experiments.

The experiments have shown both promising features of the algorithm and potential drawbacks:

1. Like many gradient-based MOO algorithms, there often seems to be bias towards individual objective minima. Thus, an important task for future research lies in trying to remedy this behavior.
2. In this regard, it would be desirable to be able to use alternative descent direction that allow guidance of the iterates or use momentum to accelerate convergence.
3. As a native MOO technique, our method can find critical points for non-convex problems and thus is preferable over some scalarization approaches like weighted-sum or (finite) p-norm scalarization. Note also that in the presence of non-linear constraint functions, *any* scalarization approach requires a capable single-objective solver.
4. Using fully linear models saves function evaluations compared to traditional modeling or derivative-approximation methods. We believe it possible to transfer the model construction optimizations from [6, 75] into our algorithm. This would allow for “model-improvement” steps with models that are not fully linear and could potentially save even more expensive function calls.
5. Unfortunately, our algorithm produces only one critical point. We have tried to come up with outer algorithms (`CSEQ`, `CSET`, `CPEN`) to obtain multiple solutions but they perform poorly against `DFMO`.
6. The algorithm is sensitive to the choice of algorithmic parameters, even more so if constraints are present. It seems advisable to perform a more structured hyper-parameter optimization.

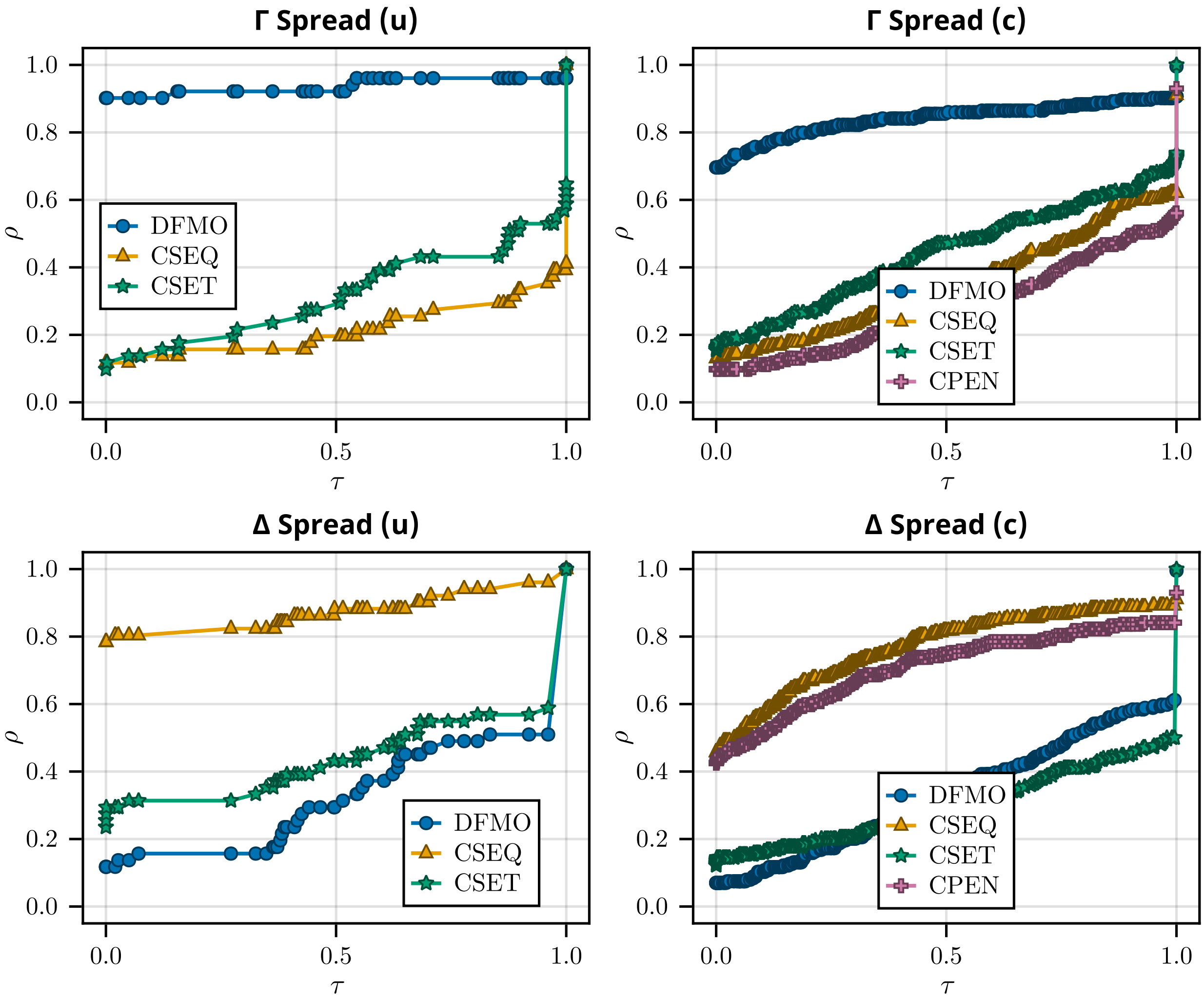

Figure 7 Spread performance for constrained (c) and unconstrained (u) problems with $\tau \in [0, 1]$.

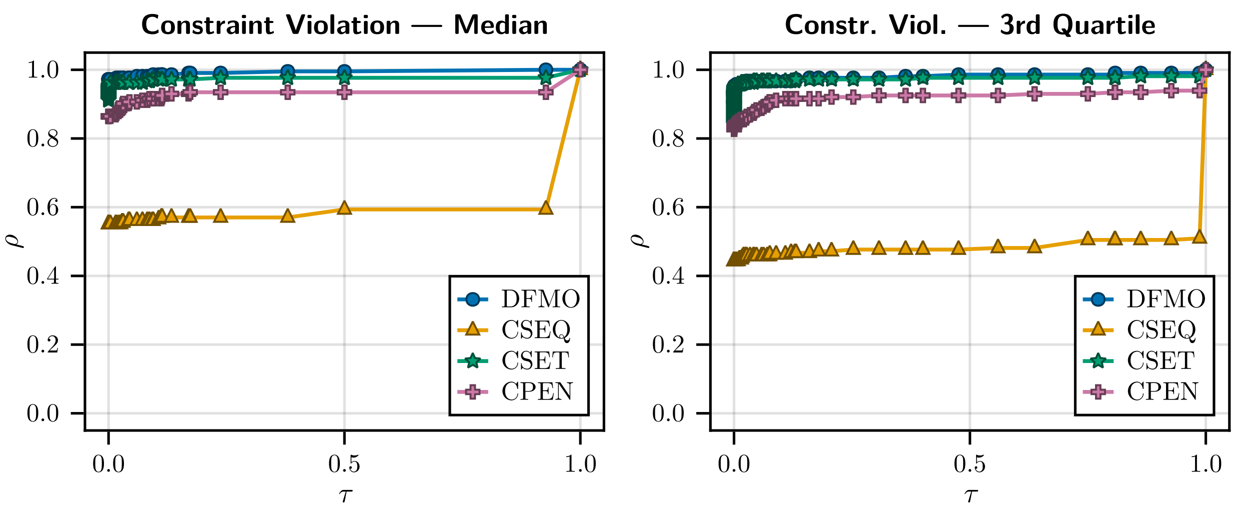

Figure 8 Constraint violation of solutions. Performance based on normalized median values of solutions (left) and normalized 3rd quartile/75th percentile values (right) with $\tau \in [0, 1]$.

7. Similarly, initialization for the `Compromise.jl` based methods should be investigated and optimized. The number of initial points, $\min\{500, 100n\}$, and the use of Halton sequences are motivated by informal trial-and-error experiments. Moreover, a relatively large initial set is required to mitigate the major drawbacks of the `Compromise.jl`: It helps to obtain a more diverse front, and expensive restoration can be avoided.

To elaborate on item 5, we want to justify the importance of our work despite the inferior performance against `DFMO`. The base algorithm provides a novel approach (and theory) for constrained derivative-free multi-objective optimization. Like with many other "baseline" algorithms, a single critical solution is found. Our modified, set-based algorithms are able to find multiple solutions but cannot compete with state-of-the-art solvers. However, these set-based algorithms constitute rather improvised drafts meant to enable any comparisons at all. In that light, it actually seems worthwhile to further develop and improve `CSET`. This requires exploration mechanism and robust theory. There are other "set-based" methods successfully building on simpler base algorithms [11, 47, 48, 57, 58], which further motivates future research in this direction.

Alternatively, it is worth mentioning, that one could think of using solutions obtained by `Compromise.jl` and its surrogates for local exploration of the Pareto Set by means of continuation. Furthermore, in Section 4.1 we have already talked about why an approach similar to that in [25] could proof beneficial. Lastly, a more theoretical question is whether or not our convergence results can be strengthened. To this end, using a slanted filter like in [30] might be useful.

8 Appendix

8.1 Equivalence of Inexact Criticality with Different Norms

To prove Theorem 17 we transfer the corresponding single-objective results from [15] to the multi-objective case. This works relatively straightforward, but to the best of our knowledge the multi-objective results have not yet been published anywhere, so the proofs are given in detail. First, we make use of the following auxiliary result:

▶ **Lemma 50** ([15, Lemma 2.2], [14, Theorem 12.1.5]). *Suppose Assumptions 5, 6, 8 and 13 hold and that ω is defined by*

$$\omega\left(\boldsymbol{x}_{\mathrm{n}}^{(k)}; \widehat{\boldsymbol{f}}^{(k)}, \mathcal{L}_k, \|\bullet\|_k, \vartheta\right) = -\min_{\boldsymbol{d}\in\mathbb{R}^n} \max_{\mathfrak{l}} \boldsymbol{d}^{\mathsf{T}} \cdot \nabla \widehat{f}_{\mathfrak{l}}(\boldsymbol{x}_{\mathrm{n}}^{(k)}) \quad s.t. \quad \boldsymbol{d} \in L_k, \quad \|\boldsymbol{d}\|_k \le \vartheta, \tag{56}$$

where $L_k = \left(\mathcal{L}_k - \boldsymbol{x}_{\mathrm{n}}^{(k)}\right)$. For any $k \in \mathbb{N}_0$, for which the normal step exists, the following statements hold:

1. *The function $\vartheta \mapsto \omega\left(\boldsymbol{x}_{\mathrm{n}}^{(k)}; \widehat{\boldsymbol{f}}^{(k)}, \mathcal{L}_k, \|\bullet\|_k, \vartheta\right)$ is continuous and non-decreasing for $\vartheta \ge 0$.*
2. *The function $\vartheta \mapsto \frac{\omega\left(\boldsymbol{x}_{\mathrm{n}}^{(k)}; \widehat{\boldsymbol{f}}^{(k)}, \mathcal{L}_k, \|\bullet\|_k, \vartheta\right)}{\vartheta}$ is non-increasing for $\vartheta > 0$.*

Proof. Let $k \in \mathbb{N}_0$ be an iteration index with algorithmic variables $(\boldsymbol{x}_{\mathrm{n}}^{(k)}, \mathcal{L}_k, \|\bullet\|_k)$.

1. Same as in [15], the feasible set mapping

$$\vartheta \mapsto D_k(\vartheta) = \left\{\boldsymbol{d} \in \mathbb{R}^n : \|\boldsymbol{d}\|_k \le \vartheta, \boldsymbol{d} \in \left(\mathcal{L}_k - \boldsymbol{x}_{\mathrm{n}}^{(k)}\right)\right\}$$

is continuous for all $\vartheta \ge 0$, and $D_k(\vartheta)$ is convex. The function $\Psi(\boldsymbol{d}, \vartheta) = \Psi(\boldsymbol{d}) = \max_{\mathfrak{l}} \boldsymbol{d}^{\mathsf{T}} \cdot \nabla \widehat{f}_{\mathfrak{l}}(\boldsymbol{x}_{\mathrm{n}}^{(k)})$ is defined on $\mathbb{R}^n \times \mathbb{R}_{\ge 0}$ and continuous, and for each $\vartheta \ge 0$ it is linear in $\boldsymbol{d}$. Similar to [14, Theorem 12.1.5], we can use results from [31], which guarantee the continuity of the optimal value

$$\min_{\boldsymbol{d}\in D_k(\vartheta)} \Psi(\boldsymbol{d}, \vartheta) = -\omega\left(\boldsymbol{x}_{\mathrm{n}}^{(k)}; \widehat{\boldsymbol{f}}^{(k)}, \mathcal{L}_k, \|\bullet\|_k, \vartheta\right).$$

By definition, ω is non-decreasing with respect to $\vartheta \ge 0$.

2. Consider $0 < \vartheta_1 < \vartheta_2$ and $\boldsymbol{d}_1, \boldsymbol{d}_2 \in \mathbb{R}^n$ such that

$$\omega\left(\boldsymbol{x}_{\mathrm{n}}^{(k)}; \widehat{\boldsymbol{f}}^{(k)}, \mathcal{L}_k, \|\bullet\|_k, \vartheta_1\right) = -\max_{\mathfrak{l}} \boldsymbol{d}_1^{\mathsf{T}} \cdot \nabla \widehat{f}_{\mathfrak{l}}(\boldsymbol{x}_{\mathrm{n}}^{(k)}), \quad \|\boldsymbol{d}_1\|_k \le \vartheta_1, \boldsymbol{d}_1 \in L_k, \tag{57}$$

$$\omega\left(\boldsymbol{x}_{\mathrm{n}}^{(k)}; \widehat{\boldsymbol{f}}^{(k)}, \mathcal{L}_k, \|\bullet\|_k, \vartheta_2\right) = -\max_{\mathfrak{l}} \boldsymbol{d}_2^{\mathsf{T}} \cdot \nabla \widehat{f}_{\mathfrak{l}}(\boldsymbol{x}_{\mathrm{n}}^{(k)}), \quad \|\boldsymbol{d}_2\|_k \le \vartheta_2, \boldsymbol{d}_2 \in L_k. \tag{58}$$

Because the set L_k is convex, it follows from $\vartheta_1/\vartheta_2 < 1$, $\mathbf{0} \in L_k$ and $\boldsymbol{d}_2 \in L_k$ that also $\vartheta_1/\vartheta_2 \boldsymbol{d}_2 \in L_k$. Moreover, it follows from (57) that

$$\left\|\frac{\vartheta_1}{\vartheta_2}\boldsymbol{d}_2\right\|_k = \frac{\vartheta_1}{\vartheta_2}\|\boldsymbol{d}_2\|_k \le \vartheta_1.$$

Thus, ${}^{\vartheta_1}/_{\vartheta_2}\boldsymbol{d}_2$ is feasible for the problem in (58). Consequently,

$$\frac{\omega\Big(\boldsymbol{x}_{\mathrm{n}}^{(k)};\widehat{\boldsymbol{f}}^{(k)},\mathcal{L}_k,\|\bullet\|_k,\vartheta_1\Big)}{\vartheta_1} \ge -\frac{1}{\vartheta_1}\max_{\mathfrak{l}}\frac{\vartheta_1}{\vartheta_2}\boldsymbol{d}_2^{\mathsf{T}}\cdot\nabla\widehat{f}_{\mathfrak{l}}(\boldsymbol{x}_{\mathrm{n}}^{(k)}) = \frac{\omega\Big(\boldsymbol{x}_{\mathrm{n}}^{(k)};\widehat{\boldsymbol{f}}^{(k)},\mathcal{L}_k,\|\bullet\|_k,\vartheta_2\Big)}{\vartheta_2}. \quad ◀$$

We can now proof Theorem 17, which states that uniformly equivalent norms imply uniformly equivalent inexact criticality values, i.e., there is $\mathtt{w} \ge 1$ with

$$\frac{1}{\mathtt{w}}\omega\Big(\boldsymbol{x}_{\mathrm{n}}^{(k)};\widehat{\boldsymbol{f}}^{(k)},\mathcal{L}_k,\|\bullet\|_k\Big) \le \omega\Big(\boldsymbol{x}_{\mathrm{n}}^{(k)};\widehat{\boldsymbol{f}}^{(k)},\mathcal{L}_k,\|\bullet\|_2\Big) \le \mathtt{w}\cdot\omega\Big(\boldsymbol{x}_{\mathrm{n}}^{(k)};\widehat{\boldsymbol{f}}^{(k)},\mathcal{L}_k,\|\bullet\|_k\Big), \quad \forall\, k. \tag{59}$$

Proof. The proof works similarly to the single-objective case [15, Theorem 3.2]. Let $k \in \mathbb{N}_0$ be such that the normal step exists. We first make the following observations:

1. The ball defined by $\|\boldsymbol{d}\|_2 \le \frac{1}{\mathsf{c}}$ is contained in the ball defined by $\|\boldsymbol{d}\|_k \le 1$, due to Assumption 6.
2. Likewise, the ball with $\|\boldsymbol{d}\|_k \le 1$ is contained in the ball defined by $\|\boldsymbol{d}\|_2 \le \mathsf{c}$.

According to (56), we define

$$\omega_{\max} = \omega\Big(\boldsymbol{x}_{\mathrm{n}}^{(k)};\widehat{\boldsymbol{f}}^{(k)},\mathcal{L}_k,\|\bullet\|_2,\mathsf{c}\Big) \quad\text{and}\quad \omega_{\min} = \omega\Big(\boldsymbol{x}_{\mathrm{n}}^{(k)};\widehat{\boldsymbol{f}}^{(k)},\mathcal{L}_k,\|\bullet\|_2,\mathsf{c}^{-1}\Big). \tag{60}$$

From the second statement in Theorem 50 it then follows that

$$\omega_{\max} \le \mathsf{c}^2\omega_{\min}. \tag{61}$$

If $\omega\Big(\boldsymbol{x}_{\mathrm{n}}^{(k)};\widehat{\boldsymbol{f}}^{(k)},\mathcal{L}_k,\|\bullet\|_2\Big) = \omega\Big(\boldsymbol{x}_{\mathrm{n}}^{(k)};\widehat{\boldsymbol{f}}^{(k)},\mathcal{L}_k,\|\bullet\|_k\Big)$ there is nothing to show. Hence, first we assume that

$$\omega\Big(\boldsymbol{x}_{\mathrm{n}}^{(k)};\widehat{\boldsymbol{f}}^{(k)},\mathcal{L}_k,\|\bullet\|_2\Big) < \omega\Big(\boldsymbol{x}_{\mathrm{n}}^{(k)};\widehat{\boldsymbol{f}}^{(k)},\mathcal{L}_k,\|\bullet\|_k\Big), \tag{62}$$

and we again take the respective minimizers $\boldsymbol{d}^{(k)}, \boldsymbol{d}_2 \in \mathbb{R}^n$ with

$$\omega\Big(\boldsymbol{x}_{\mathrm{n}}^{(k)};\widehat{\boldsymbol{f}}^{(k)},\mathcal{L}_k,\|\bullet\|_k\Big) = -\max_{\mathfrak{l}}\boldsymbol{d}^{(k)^{\mathsf{T}}}\cdot\nabla\widehat{f}_{\mathfrak{l}}(\boldsymbol{x}_{\mathrm{n}}^{(k)}), \quad \Big\|\boldsymbol{d}^{(k)}\Big\|_k \le 1,\ \boldsymbol{d}^{(k)} \in L_k, \tag{63}$$

$$\omega\Big(\boldsymbol{x}_{\mathrm{n}}^{(k)};\widehat{\boldsymbol{f}}^{(k)},\mathcal{L}_k,\|\bullet\|_2\Big) = -\max_{\mathfrak{l}}\boldsymbol{d}_2^{\mathsf{T}}\cdot\nabla\widehat{f}_{\mathfrak{l}}(\boldsymbol{x}_{\mathrm{n}}^{(k)}), \qquad \|\boldsymbol{d}_2\|_2 \le 1, \quad \boldsymbol{d}_2 \in L_k. \tag{64}$$

Then

$$\frac{1}{\mathsf{c}} \le \Big\|\boldsymbol{d}^{(k)}\Big\|_2 \le \mathsf{c} \quad\text{and}\quad \frac{1}{\mathsf{c}} \le \|\boldsymbol{d}_2\|_2 \le \mathsf{c}. \tag{65}$$

The upper bounds are trivial because of $\mathsf{c} \ge 1$. Suppose the first lower bound in (65) is violated, i.e., $\Big\|\boldsymbol{d}^{(k)}\Big\|_2 < {}^1/_{\mathsf{c}}$. According to observation 1, it then follows that $\Big\|\boldsymbol{d}^{(k)}\Big\|_k \le 1$. The vector $\boldsymbol{d}^{(k)} \in L_k$ is then also feasible for (64) and the optimality of $\boldsymbol{d}_2$ implies

$$\underbrace{-\max_{\mathfrak{l}}\boldsymbol{d}_2^{\mathsf{T}}\cdot\nabla\widehat{f}_{\mathfrak{l}}(\boldsymbol{x}_{\mathrm{n}}^{(k)})}_{=\omega\left(\boldsymbol{x}_{\mathrm{n}}^{(k)};\widehat{\boldsymbol{f}}^{(k)},\mathcal{L}_k,\|\bullet\|_2\right)} \ge \underbrace{-\max_{\mathfrak{l}}\boldsymbol{d}^{(k)^{\mathsf{T}}}\cdot\nabla\widehat{f}_{\mathfrak{l}}(\boldsymbol{x}_{\mathrm{n}}^{(k)})}_{=\omega\left(\boldsymbol{x}_{\mathrm{n}}^{(k)};\widehat{\boldsymbol{f}}^{(k)},\mathcal{L}_k,\|\bullet\|_k\right)},$$

in contradiction to (62).

Suppose the other lower bound in (65) is violated, i.e., $\|\boldsymbol{d}_2\|_2 < {}^1/_{\mathsf{c}}$. According to our observations from above, for $\boldsymbol{d}_2$ we also have $\|\boldsymbol{d}_2\|_k \le 1$, and it is therefore feasible for the problem in (63). With (62) we even see that it is strictly optimal, contradicting the optimality of $\boldsymbol{d}^{(k)}$ in (63). Thus, the second lower bound in (65) must hold, too.

Equation (65) shows that both vectors $\boldsymbol{d}^{(k)}$ and $\boldsymbol{d}_2$ are feasible for the problems defining $\omega_{\max}$ and $\omega_{\min}$ in (60). We again apply the definition (56) to see that

$$\omega_{\min} \le \omega\Big(\boldsymbol{x}_{\mathrm{n}}^{(k)};\widehat{\boldsymbol{f}}^{(k)},\mathcal{L}_k,\|\bullet\|_2\Big) \le \omega_{\max} \quad\text{and}\quad \omega_{\min} \le \omega\Big(\boldsymbol{x}_{\mathrm{n}}^{(k)};\widehat{\boldsymbol{f}}^{(k)},\mathcal{L}_k,\|\bullet\|_k\Big) \le \omega_{\max}.$$

With (62) we obtain a strict upper bound for the problem using $\|\bullet\|_2$, which we modify using (61):

$$\omega\Big(\boldsymbol{x}_{\mathrm{n}}^{(k)};\widehat{\boldsymbol{f}}^{(k)},\mathcal{L}_k,\|\bullet\|_2\Big) < \omega\Big(\boldsymbol{x}_{\mathrm{n}}^{(k)};\widehat{\boldsymbol{f}}^{(k)},\mathcal{L}_k,\|\bullet\|_k\Big) \le \omega_{\max} \le \mathsf{c}^2\omega_{\min} \le \mathsf{c}^2\omega\Big(\boldsymbol{x}_{\mathrm{n}}^{(k)};\widehat{\boldsymbol{f}}^{(k)},\mathcal{L}_k,\|\bullet\|_2\Big).$$

This implies

$$\frac{1}{\mathsf{c}^2}\omega\Big(\boldsymbol{x}_{\mathrm{n}}^{(k)}; \widehat{\boldsymbol{f}}^{(k)}, \mathcal{L}_k, \|\bullet\|_k\Big) \le \omega\Big(\boldsymbol{x}_{\mathrm{n}}^{(k)}; \widehat{\boldsymbol{f}}^{(k)}, \mathcal{L}_k, \|\bullet\|_2\Big) \le \mathsf{c}^2\omega\Big(\boldsymbol{x}_{\mathrm{n}}^{(k)}; \widehat{\boldsymbol{f}}^{(k)}, \mathcal{L}_k, \|\bullet\|_k\Big), \tag{66}$$

and we have shown that (59) holds for any constant $\mathtt{w}$ with $\mathtt{w} \ge \mathsf{c}^2 \ge 1$, in case that (62) is satisfied. The other case,

$$\omega\Big(\boldsymbol{x}_{\mathrm{n}}^{(k)}; \widehat{\boldsymbol{f}}^{(k)}, \mathcal{L}_k, \|\bullet\|_k\Big) < \omega\Big(\boldsymbol{x}_{\mathrm{n}}^{(k)}; \widehat{\boldsymbol{f}}^{(k)}, \mathcal{L}_k, \|\bullet\|_2\Big),$$

is treated analogously and also leads to (66). We conclude that Theorem 17 is valid. ◀

8.2 Sufficient Decrease via Backtracking

This section of the appendix is concerned with justifying the sufficient decrease bound (10) in Assumption 19. In the single-objective case, there are many possibilities to achieve the bound, some of which, e.g., exact and inexact line-search, we have previously shown to work in the multi-objective as well if the global feasible set is compact and convex [6]. Now at least the approximated linearized feasible sets are convex, which allows us to paraphrase and utilize the following result:

▶ **Lemma 51** (cf. [6, Lemma 3]). *Let L be a convex and compact set, let $\boldsymbol{d}$ be a descent direction for $\widehat{\boldsymbol{f}}$ at $\boldsymbol{x} \in L$, and let $\sigma \ge 0$ be a step-size such that $\boldsymbol{x} + {}^{\sigma}\!/_{\|\boldsymbol{d}\|} \cdot \boldsymbol{d} \in L$, where $\|\bullet\|$ is a vector norm that fulfills* (5)*. Then, for any fixed constants $\mathsf{a}, \mathsf{b} \in (0,1)$ and for $\Phi = \Phi\big[\widehat{\boldsymbol{f}}\big]$ there is an integer $j \in \mathbb{N}_0$ such that*

$$\Phi(\boldsymbol{x}) - \Phi\left(\boldsymbol{x} + \frac{\mathsf{b}^j\sigma}{\|\boldsymbol{d}\|}\boldsymbol{d}\right) \ge \mathsf{a}\frac{\sigma\mathsf{b}^j}{\|\boldsymbol{d}\|}\omega, \tag{67}$$

where $\omega = -\max_{\mathfrak{l}} \boldsymbol{d}^{\mathsf{T}} \cdot \nabla \widehat{f}_{\mathfrak{l}}(\boldsymbol{x})$. Moreover, there is a constant $\mathsf{c}_{\mathrm{sd}} \in (0,1)$ such that if j is the smallest $j \in \mathbb{N}_0$ that satisfies (67)*, then*

$$\Phi(\boldsymbol{x}) - \Phi\left(\boldsymbol{x} + \frac{\mathsf{b}^j\sigma}{\|\boldsymbol{d}\|}\boldsymbol{d}\right) \ge \mathsf{c}_{\mathrm{sd}}\omega \min\left\{\frac{\omega}{\|\boldsymbol{d}\|^2\mathsf{c}^2 H}, \frac{\sigma}{\|\boldsymbol{d}\|}\right\},$$

where

$$H = \max_{\boldsymbol{d} \in L - \boldsymbol{x}} \max_{\mathfrak{l}=1,\dots,K} \left\|\nabla^2 \widehat{f}_{\mathfrak{l}}(\boldsymbol{x} + \boldsymbol{d})\right\|_2 \quad s.t. \quad \|\boldsymbol{d}\| \le \mathsf{b}^j\sigma. \tag{68}$$

For Theorem 51 to be applicable in our setting, we take L as the intersection of current trust-region and approximated linearized feasible set. To make it work, we require the surrogate function $\widehat{\boldsymbol{f}}$ to be twice continuously differentiable and defined on $\mathcal{C}(\mathcal{X})$, which is guaranteed by Assumptions 5, 8 and 13. Furthermore, Assumption 18 may be used instead of (68) to bound the model Hessians, which appear due to a Taylor approximation of the components of $\widehat{\boldsymbol{f}}$ in the proof of Theorem 51.

We now have to deal with the possibly different norms $\|\bullet\|_{\mathrm{tr,k}}$ (defining the trust-region) and $\|\bullet\|_{\mathrm{tr}}$ (used in $(\mathrm{ITRT}^{(k)})$) and how to choose the initial backtracking step-length σ in Theorem 51. We want to choose $\sigma \ge 0$ as large as possible and so that for a descent direction $\boldsymbol{d}^{(k)}$ it holds that $\big\|\boldsymbol{n}^{(k)} + \sigma\boldsymbol{d}^{(k)}\big\|_{\mathrm{tr,k}} \le \Delta_{(k)}$ and $\boldsymbol{x}_{\mathrm{n}}^{(k)} + \sigma\boldsymbol{d}^{(k)} \in \mathcal{L}_k$. Luckily, there is the following bound on the optimal σ:

▶ **Lemma 52.** *For $\boldsymbol{x}^{(k)}$ let $\mathcal{L}_k$ be the linearized feasible set and $B^{(k)}$ be a trust-region of radius $\Delta_{(k)}$ w.r.t. $\|\bullet\|_{\mathrm{tr,k}}$. Let $\boldsymbol{d}^{(k)}$ be a minimizer of* $(\mathrm{ITRT}^{(k)})$ *with $\big\|\boldsymbol{d}^{(k)}\big\|_k \le 1$. Then there is an initial step-length $\overline{\sigma} \ge 0$ with $\boldsymbol{x}_{\mathrm{n}}^{(k)} + \frac{\overline{\sigma}}{\|\boldsymbol{d}^{(k)}\|_{\mathrm{tr,k}}}\boldsymbol{d}^{(k)} \in \mathcal{L}_k \cap B^{(k)}$ and*

$$\overline{\sigma} \ge \min\left\{\Delta_{(k)} - \left\|\boldsymbol{n}^{(k)}\right\|_{\mathrm{tr,k}}, \left\|\boldsymbol{d}^{(k)}\right\|_{\mathrm{tr,k}}\right\}. \tag{69}$$

Proof. There are, of course, better ways to determine $\overline{\sigma}$, but if $\big\|\boldsymbol{n}^{(k)} + \boldsymbol{d}^{(k)}\big\|_{\mathrm{tr,k}} \le \Delta_{(k)}$ we can always choose $\overline{\sigma} = \big\|\boldsymbol{d}^{(k)}\big\|_{\mathrm{tr,k}}$. If $\big\|\boldsymbol{n}^{(k)} + \boldsymbol{d}^{(k)}\big\|_{\mathrm{tr,k}} > \Delta_{(k)}$ we can equate either side of the triangle inequality

$$\left\|\boldsymbol{n}^{(k)}\right\|_{\mathrm{tr,k}} + \overline{\sigma} \ge \left\|\boldsymbol{n}^{(k)} + \frac{\overline{\sigma}}{\|\boldsymbol{d}^{(k)}\|_{\mathrm{tr,k}}}\boldsymbol{d}^{(k)}\right\|_{\mathrm{tr,k}}$$

with $\Delta_{(k)}$ and solve for $\overline{\sigma}$. ◀

Finally, we are able to derive the sufficient decrease bound (10) when backtracking is used. Of course, in case that $\boldsymbol{x}^{(k)}$ is critical, the bound (10) is automatically fulfilled. Else, we use Theorem 51 with $\bar{\sigma}$ satisfying (69). Assumption 6 and $\left\|\boldsymbol{d}^{(k)}\right\|_k \le 1$ together imply

$$\left\|\boldsymbol{d}^{(k)}\right\|_{\mathrm{tr,k}} \le \mathtt{c}^2 \left\|\boldsymbol{d}^{(k)}\right\|_k \le \mathtt{c}^2. \tag{70}$$

Theorem 51 and the fact that we assume $\boldsymbol{n}^{(k)}$ to be compatible lead to

$$\begin{aligned}
\Phi^{(k)}\left(\boldsymbol{x}_{\mathrm{n}}^{(k)}\right) - \Phi^{(k)}\left(\boldsymbol{x}_{\mathrm{n}}^{(k)} + \sigma_{(k)}\boldsymbol{d}^{(k)}\right) &\ge \widetilde{\mathtt{c}}_{\mathrm{sd}}\widehat{\omega}^{(k)} \min\left\{ \frac{\widehat{\omega}^{(k)}}{\left\|\boldsymbol{d}^{(k)}\right\|_{\mathrm{tr,k}}^2 \mathtt{c}^2\mathtt{H}}, \frac{\bar{\sigma}}{\left\|\boldsymbol{d}^{(k)}\right\|_{\mathrm{tr,k}}} \right\} \\
&\overset{(69)}{\ge} \widetilde{\mathtt{c}}_{\mathrm{sd}}\widehat{\omega}^{(k)} \min\left\{ \frac{\widehat{\omega}^{(k)}}{\left\|\boldsymbol{d}^{(k)}\right\|_{\mathrm{tr,k}}^2 \mathtt{c}^2\mathtt{H}}, \frac{\Delta_{(k)} - \left\|\boldsymbol{n}^{(k)}\right\|_{\mathrm{tr,k}}}{\left\|\boldsymbol{d}^{(k)}\right\|_{\mathrm{tr,k}}}, \frac{\left\|\boldsymbol{d}^{(k)}\right\|_{\mathrm{tr,k}}}{\left\|\boldsymbol{d}^{(k)}\right\|_{\mathrm{tr,k}}} \right\} \\
&\overset{(70),(7)}{\ge} \widetilde{\mathtt{c}}_{\mathrm{sd}}\widehat{\omega}^{(k)} \min\left\{ \frac{\widehat{\omega}^{(k)}}{\mathtt{c}^4\mathtt{c}^2\mathtt{H}}, \frac{1-\mathtt{c}_\Delta}{\mathtt{c}^2}\Delta_{(k)}, \frac{1}{\mathtt{c}^2} \right\} \\
&\overset{(9)}{\ge} \frac{(1-\mathtt{c}_\Delta)\widetilde{\mathtt{c}}_{\mathrm{sd}}}{\mathtt{w}\mathtt{c}^2}\widehat{\omega}_2^{(k)} \min\left\{ \frac{\widehat{\omega}_2^{(k)}}{(1-\mathtt{c}_\Delta)\mathtt{w}\mathtt{c}^4\mathtt{H}}, \Delta_{(k)}, \frac{1}{(1-\mathtt{c}_\Delta)} \right\}.
\end{aligned}$$

To obtain (10), we define $\mathtt{c}_{\mathrm{sd}} := \dfrac{\widetilde{\mathtt{c}}_{\mathrm{sd}}(1-\mathtt{c}_\Delta)}{\mathtt{w}\mathtt{c}^2} \in (0,1)$, and notice that $1/(1-\mathtt{c}_\Delta) \ge 1$, and that we can assume $\mathtt{W} := \mathtt{w}\mathtt{c}^4\mathtt{H} \ge 1$ w.l.o.g..

8.3 Set-Based Algorithm `CSET`

With `CSET` we have tried to construct a set-based algorithm that shares most of the features discussed in Section 3. Conceptually, it is a bit similar to the set-based trust-region algorithm in [57] and its surrogate-assisted variant in [58]. For constraint handling, we again use a filter, and exact derivatives are not required thanks to fully linear surrogates.

1. Given a finite set of initial points $\{\boldsymbol{x}_0, \boldsymbol{x}_1, \ldots\} \subset \mathbb{R}^n$, form the set $\mathcal{S} \subset \mathbb{R}^K$ by selecting only those points that are non-dominated with respect to $\boldsymbol{x} \mapsto [\theta(\boldsymbol{x}), \boldsymbol{f}(\boldsymbol{x})]$.
   Assign an initial positive trust-region radius to each point in $\mathcal{S}$.
   Initialize empty filter $\mathcal{F} = \{\} \subset \mathbb{R}^{K+1}$ and restoration seeds $\mathcal{R} = \{\} \subset \mathbb{R}^K$.
2. If $\mathcal{S} \neq \emptyset$, then choose a non-converged point $\boldsymbol{x} \in \mathcal{S}$ and go to step 4. If all points have converged, then `STOP`. If $\mathcal{S} = \emptyset$, go to step 3.
3. Find a point $\boldsymbol{x}$ and a radius $\Delta \in (0, \Delta_{\max}]$ such that $\boldsymbol{x}$ is acceptable for $\mathcal{F}$ and such that it has a compatible normal step for Δ. The seeds in $\mathcal{R}$ might provide valuable information concerning starting points. If a suitable point $\boldsymbol{x}$ is found, go to step 2. Otherwise, return `INFEASIBLE`.
4. For $\boldsymbol{x}$, ensure that all surrogates $\widehat{\boldsymbol{f}}, \widehat{\boldsymbol{g}}, \widehat{\boldsymbol{h}}$ are fully linear w.r.t. its trust-region and compute a normal step $\boldsymbol{n}$ with (ITRN$^{(k)}$). If no compatible normal step is found, set $\mathcal{S} \leftarrow \mathcal{S} \setminus \{\boldsymbol{x}\}$ and $\mathcal{R} \leftarrow \mathcal{R} \cup \{\boldsymbol{x}\}$, and go to step 2.
5. Compute a (partial) descent step $\boldsymbol{d}$ similar to (ITRT$^{(k)}$). If necessary, enter the Criticality Routine. Otherwise, find a step-size σ such that $\boldsymbol{x}_{\mathrm{s}} = \boldsymbol{x} + \boldsymbol{n} + \sigma\boldsymbol{d}$ has the sufficient decrease property for all or some objective surrogate indices $I \subseteq \{1, \ldots, K\}, I \neq \emptyset$.
6. If $\boldsymbol{x}_{\mathrm{s}}$ is not acceptable for $\mathcal{F} \cup \{[\theta(\boldsymbol{x}), \boldsymbol{f}(\boldsymbol{x})]\}$, then go to step 10.
7. Let
   $$D = \min_{\mathfrak{l} \in I} \widehat{f}_{\mathfrak{l}}(\boldsymbol{x}) - \widehat{f}_{\mathfrak{l}}(\boldsymbol{x}_{\mathrm{s}}) \quad \text{and} \quad \rho = \min_{\mathfrak{l} \in I} \frac{f_{\mathfrak{l}}(\boldsymbol{x}) - f_{\mathfrak{l}}(\boldsymbol{x}_{\mathrm{s}})}{\widehat{f}_{\mathfrak{l}}(\boldsymbol{x}) - \widehat{f}_{\mathfrak{l}}(\boldsymbol{x}_{\mathrm{s}})}.$$
   If $D \ge \kappa_\theta \theta(\boldsymbol{x})^\psi$ and $\rho < \nu_0$, go to step 10.
8. If $D < \kappa_\theta \theta(\boldsymbol{x})^\psi$, then include $\boldsymbol{x}$ in the filter $\mathcal{F}$ and prune $\mathcal{S}$.
9. Compute a new trust region radius similar to (13). Assign the new radius to $\boldsymbol{x}$. Add $\boldsymbol{x}_{\mathrm{s}}$ to $\mathcal{S}$, also using the new radius. Go to step 2.
10. Reduce the radius Δ associated with $\boldsymbol{x}$, $\Delta \leftarrow [\gamma_0\Delta, \gamma_1\Delta]$, and go to step 2.

There are no convergence guarantees, and we leave many details intentionally vague. In step 2, we might consider a point converged if its trust-region is very small. Of course, we can also stop after a certain number of iterations, or if some computation budget is exhausted. Stopping based on small trust-regions also prevents the Criticality Routine from looping infinitely, but there certainly are smarter ways of preventing a single solution from using the budget during Criticality loops. The restoration seeds $\mathcal{R}$ are optional, but can be used in step 3 to choose a starting point, e.g., based on constraint violation or normal step length. Whenever a point is added to $\mathcal{S}$, all points that become dominated w.r.t. $\boldsymbol{f}$ are removed. Similarly, the size of $\mathcal{R}$ can be reduced by removing seeds that are dominated w.r.t. $\boldsymbol{x} \mapsto [\theta(\boldsymbol{x}), \boldsymbol{f}(\boldsymbol{x})]$. A point $\boldsymbol{x}$ is acceptable for $\mathcal{F}$, if it is "non-dominated with envelope", i.e., if for all $[\theta_j, \boldsymbol{f}^j] \in \mathcal{F}$ it holds that

$$\theta(\boldsymbol{x}) \le \theta_j - \gamma_\theta \theta_j \quad \text{or (for some } \mathfrak{l}) \quad f_\mathfrak{l}(\boldsymbol{x}) \le f^j_\mathfrak{l} - \gamma_\theta \theta_j .$$

If $\boldsymbol{x}$ is added to $\mathcal{F}$, then all points dominated by the vector $[\theta(\boldsymbol{x}), \boldsymbol{f}(\boldsymbol{x})] - \gamma_\theta \theta$ are removed. Like in Section 3, we want all elements of $\mathcal{S}$ to be acceptable for $\mathcal{F}$, so in step 8 we remove those points from $\mathcal{S}$ that are no longer acceptable after filter modification. As before, no feasible iterate should ever be added to $\mathcal{F}$. Special care has to be taken when a parent $\boldsymbol{x}$ is not dominated by its child $\boldsymbol{x}_\mathrm{s}$. Either the radius has to be changed, or we have to make sure that the index set I is different when $\boldsymbol{x}$ is chosen next. Theoretically, a partial descent direction could be computed such that the Pareto Set or the Pareto Front is better explored. At the moment, exploration happens by accident: We still compute the steepest descent direction using fully linear surrogates, but accept step-sizes such that the Armijo condition is fulfilled only for some indices I. A more thought-out approach is subject to future research.

References

1 Isabela Albuquerque, Joao Monteiro, Thang Doan, Breandan Considine, Tiago Falk, and Ioannis Mitliagkas. Multi-Objective Training of Generative Adversarial Networks with Multiple Discriminators. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 202–211. PMLR, 2019.

2 Hédy Attouch, Guillaume Garrigos, and Xavier Goudou. A Dynamic Gradient Approach to Pareto Optimization with Nonsmooth Convex Objective Functions. *J. Math. Anal. Appl.*, 422(1):741–771, 2015.

3 Charles Audet, Jean Bigeon, Dominique Cartier, Sébastien Le Digabel, and Ludovic Salomon. Performance Indicators in Multiobjective Optimization. *Eur. J. Oper. Res.*, 292(2):397–422, 2020.

4 Charles Audet, Gilles Savard, and Walid Zghal. A Mesh Adaptive Direct Search Algorithm for Multiobjective Optimization. *Eur. J. Oper. Res.*, 204(3):545–556, 2010.

5 Manuel B. Berkemeier. CoMPrOMISE.Jl Github Repository, 2024. `https://github.com/manuelbb-upb/Compromise.jl`.

6 Manuel B. Berkemeier and Sebastian Peitz. Derivative-Free Multiobjective Trust Region Descent Method Using Radial Basis Function Surrogate Models. *Math. Comput. Appl.*, 26(2): article no. 31 (37 pages), 2021.

7 Jean Bigeon, Sébastien Le Digabel, and Ludovic Salomon. DMulti-MADS: Mesh Adaptive Direct Multisearch for Bound-Constrained Blackbox Multiobjective Optimization. *Comput. Optim. Appl.*, 79(2):301–338, 2021.

8 Jean Bigeon, Sébastien Le Digabel, and Ludovic Salomon. Handling of Constraints in Multiobjective Blackbox Optimization. *Comput. Optim. Appl.*, 89(1):69–113, 2024.

9 Carmo P. Brás and Ana Luísa Custódio. On the Use of Polynomial Models in Multiobjective Directional Direct Search. *Comput. Optim. Appl.*, 77(3):897–918, 2020.

10 Tinkle Chugh, Karthik Sindhya, Jussi Hakanen, and Kaisa Miettinen. A Survey on Handling Computationally Expensive Multiobjective Optimization Problems with Evolutionary Algorithms. *Soft Computing*, 23(9):3137–3166, 2019.

11 Guido Cocchi, Giampaolo Liuzzi, Stefano Lucidi, and Marco Sciandrone. On the Convergence of Steepest Descent Methods for Multiobjective Optimization. *Comput. Optim. Appl.*, 77(1):1–27, 2020.

12 Guido Cocchi, Giampaolo Liuzzi, Alessandra Papini, and Marco Sciandrone. An Implicit Filtering Algorithm for Derivative-Free Multiobjective Optimization with Box Constraints. *Comput. Optim. Appl.*, 69(2):267–296, 2018.

13 Carlos A. Coello Coello, David. A. van Veldhuizen, and Gary B. Lamont. *Evolutionary Algorithms for Solving Multi-Objective Problems*. Genetic and Evolutionary Computation. Springer, 2013.

14 Andrew R. Conn, Nicholas I. M. Gould, and Philippe Louis Toint. *Trust-Region Methods*. MPS/SIAM Series on Optimization. Society for Industrial and Applied Mathematics, 2000.

15 Andrew R. Conn, Nick Gould, Annick Sartenaer, and Philippe Louis Toint. Global Convergence of a Class of Trust Region Algorithms for Optimization Using Inexact Projections on Convex Constraints. *SIAM J. Optim.*, 3(1):164–221, 1993.

16 Andrew R. Conn, Katya Scheinberg, and Luís N. Vicente. *Introduction to Derivative-Free Optimization*. Number 8 in MPS/SIAM Series on Optimization. Society for Industrial and Applied Mathematics/Mathematical Programming Society, 2009.

17 Ana Luísa Custódio, José Firmino Aguilar Madeira, A. Ismael F. Vaz, and Luís N. Vicente. Direct Multisearch for Multiobjective Optimization. *SIAM J. Optim.*, 21(3):1109–1140, 2011.

18 Simon Danisch and Julius Krumbiegel. Makie.Jl: Flexible High-Performance Data Visualization for Julia. *J. Open Source Softw.*, 6(65): article no. 3349 (5 pages), 2021.

19 Kalyanmoy Deb, Amrit Pratap, Sameer Agarwal, and T. Meyarivan. A Fast and Elitist Multiobjective Genetic Algorithm: NSGA-II. *IEEE Trans. Evol. Comput.*, 6(2):182–197, 2002.

20 Kalyanmoy Deb, Proteek Chandan Roy, and Rayan Hussein. Surrogate Modeling Approaches for Multiobjective Optimization: Methods, Taxonomy, and Results. *Math. Comput. Appl.*, 26(1): article no. 5 (27 pages), 2020.

21 Sander Dedoncker, Wim Desmet, and Frank Naets. An Adaptive Direct Multisearch Method for Black-Box Multi-Objective Optimization. *Optim. Eng.*, 23(3):1411–1437, 2022.

22 Elizabeth D. Dolan and Jorge J. Moré. Benchmarking Optimization Software with Performance Profiles. *Math. Program.*, 91(2):201–213, 2002.

23 Luis Mauricio Graña Drummond and Benar Fux Svaiter. A Steepest Descent Method for Vector Optimization. *J. Comput. Appl. Math.*, 175(2):395–414, 2005.

24 John P. Eason. *A Trust Region Filter Algorithm for Surrogate-based Optimization*. PhD thesis, Carnegie Mellon University, 2018.

25 John P. Eason and Lorenz T. Biegler. A Trust Region Filter Method for Glass Box/Black Box Optimization. *AIChE J.*, 62(9):3124–3136, 2016.

26 Nélida E. Echebest, Maria L. Schuverdt, and R. P. Vignau. An Inexact Restoration Derivative-Free Filter Method for Nonlinear Programming. *Comput. Appl. Math.*, 36(1):693–718, 2017.

27 Matthias Ehrgott. *Multicriteria Optimization*. Springer, 2nd ed edition, 2005.

28 Gabriele Eichfelder. *Adaptive Scalarization Methods in Multiobjective Optimization*. Vector Optimization. Springer, 2008.

29 Gabriele Eichfelder. Twenty Years of Continuous Multiobjective Optimization in the Twenty-First Century. *EURO J. Comput. Optim.*, 9: article no. 100014, 2021.

30 Priscila S. Ferreira, Elizabeth W. Karas, Mael Sachine, and Francisco N. C. Sobral. Global Convergence of a Derivative-Free Inexact Restoration Filter Algorithm for Nonlinear Programming. *Optimization*, 66(2):271–292, 2017.

31 Anthony V. Fiacco. *Introduction to Sensitivity and Stability Analysis in Nonlinear Programming*, volume 165 of *Mathematics in Science and Engineering*. Elsevier, 1983.

32 Roger Fletcher, Nicholas I. M. Gould, Sven Leyffer, Philippe Louis Toint, and Andreas Wächter. Global Convergence of a Trust-Region SQP-Filter Algorithm for General Nonlinear Programming. *SIAM J. Optim.*, 13(3):635–659, 2002.

33 Jörg Fliege and Benar Fux Svaiter. Steepest Descent Methods for Multicriteria Optimization. *Math. Methods Oper. Res.*, 51(3):479–494, 2000.

34 Jörg Fliege and A. Ismael F. Vaz. A Method for Constrained Multiobjective Optimization Based on SQP Techniques. *SIAM J. Optim.*, 26(4):2091–2119, 2016.

35 Ellen H. Fukuda and Luis Mauricio Graña Drummond. A Survey on Multiobjective Descent Methods. *Pesqui. Oper.*, 34(3):585–620, 2014.

36 Sebastian R. Garreis. *Optimal Control under Uncertainty: Theory and Numerical Solution with Low-Rank Tensors*. PhD thesis, Technische Universität München, 2019. `https://mediatum.ub.tum.de/1452538`.

37 Bennet Gebken and Sebastian Peitz. An Efficient Descent Method for Locally Lipschitz Multiobjective Optimization Problems. *J. Optim. Theory Appl.*, 188(3):696–723, 2021.

38 Bennet Gebken, Sebastian Peitz, and Michael Dellnitz. A Descent Method for Equality and Inequality Constrained Multiobjective Optimization Problems. In *Numerical and Evolutionary Optimization – NEO 2017*, volume 785 of *Studies in Computational Intelligence*, pages 29–61. Springer, 2018.

39 Bennet Gebken, Sebastian Peitz, and Michael Dellnitz. On the Hierarchical Structure of Pareto Critical Sets. *AIP Conf. Proc.*, 2070: article no. 020041, 2019. Proceedings LEGO – 14th International Global Optimization Workshop.

40 Clóvis C. Gonzaga, Elizabeth W. Karas, and Márcia Vanti. A Globally Convergent Filter Method for Nonlinear Programming. *SIAM J. Optim.*, 14(3):646–669, 2004.

41 Nicholas I. M. Gould, Sven Leyffer, and Philippe Louis Toint. A Multidimensional Filter Algorithm for Nonlinear Equations and Nonlinear Least-Squares. *SIAM J. Optim.*, 15(1):17–38, 2004.

42 Claus Hillermeier. *Nonlinear Multiobjective Optimization: A Generalized Homotopy Approach*. Springer, 2001.

43 Qi Huangfu and J. A. Julian Hall. Parallelizing the Dual Revised Simplex Method. *Math. Program. Comput.*, 10(1):119–142, 2018.

44 Johannes Jahn. *Vector Optimization: Theory, Applications, and Extensions*. Springer, 2. ed edition, 2011.

45 Steven G. Johnson. The NLopt Nonlinear-Optimization Package, 2007. https://github.com/stevengj/nlopt.

46 Harold W. Kuhn and Albert W. Tucker. Nonlinear Programming. In *Proceedings of the Second Berkeley Symposium on Mathematical Statistics and Probability*, pages 481–492. University of California Press, 2014.

47 Matteo Lapucci and Pierluigi Mansueto. Improved Front Steepest Descent for Multi-Objective Optimization. *Oper.*

Res. Lett., 51(3):242–247, 2023.

48 Matteo Lapucci, Pierluigi Mansueto, and Davide Pucci. Effective Front-Descent Algorithms with Convergence Guarantees. `https://arxiv.org/abs/2405.08450`, 2024.

49 J. G. Lin. Maximal Vectors and Multi-Objective Optimization. *J. Optim. Theory Appl.*, 18(1):41–64, 1976.

50 Giampaolo Liuzzi, Stefano Lucidi, and Francesco Rinaldi. A Derivative-Free Approach to Constrained Multiobjective Nonsmooth Optimization. *SIAM J. Optim.*, 26(4):2744–2774, 2016.

51 L. R. Lucambio Pérez and Leandro F. Prudente. Nonlinear Conjugate Gradient Methods for Vector Optimization. *SIAM J. Optim.*, 28(3):2690–2720, 2018.

52 Zhongwei Ma. Evolutionary Constrained Multiobjective Optimization: Test Suite Construction and Performance Comparisons. *IEEE Trans. Evol. Comput.*, 23(6):972–986, 2019.

53 Adanay Martín and Oliver Schütze. Pareto Tracer: A Predictor–Corrector Method for Multi-Objective Optimization Problems. *Eng. Optim.*, 50(3):516–536, 2018.

54 José Mario Martínez and Benar Fux Svaiter. A Practical Optimality Condition Without Constraint Qualifications for Nonlinear Programming. *J. Optim. Theory Appl.*, 118(1):117–133, 2003.

55 Ioan Maruşciac. On Fritz John Type Optimality Criterion in Multi-Objective Optimization. *Math., Rev. Anal. Numér. Théor. Approximation, Anal. Numér. Théor. Approximation*, 11(1-2):109–114, 1982.

56 Kaisa Miettinen. *Nonlinear Multiobjective Optimization.* Springer, 2013.

57 Aboozar Mohammadi and Ana Luísa Custódio. A Trust-Region Approach for Computing Pareto Fronts in Multiobjective Optimization. *Comput. Optim. Appl.*, 87(1):149–179, 2024.

58 Aboozar Mohammadi, Davood Hajinezhad, and Alfredo Garcia. A Trust-Region Approach for Computing Pareto Fronts in Multiobjective Derivative-Free Optimization. *Optim. Lett.*, 19(2):233–266, 2025.

59 Hiro Mukai. Algorithms for Multicriterion Optimization. *IEEE Trans. Autom. Control*, 25(2):177–186, 1980.

60 Aviv Navon, Aviv Shamsian, Gal Chechik, and Ethan Fetaya. Learning the Pareto Front with Hypernetworks. `https://arxiv.org/abs/2010.04104v2`, International Conference on Learning Representations, 2021.

61 Sebastian Peitz and Michael Dellnitz. Gradient-Based Multiobjective Optimization with Uncertainties. In *NEO 2016*, volume 731 of *Studies in Computational Intelligence*, pages 159–182. Springer, 2018.

62 Sebastian Peitz and Michael Dellnitz. A Survey of Recent Trends in Multiobjective Optimal Control—Surrogate Models, Feedback Control and Objective Reduction. *Math. Comput. Appl.*, 23(2): article no. 30 (33 pages), 2018.

63 Michael James David Powell. A Direct Search Optimization Method That Models the Objective and Constraint Functions by Linear Interpolation. In Susana Gomez and Jean-Pierre Hennart, editors, *Advances in Optimization and Numerical Analysis*, pages 51–67. Springer, 1994.

64 Shaojian Qu, Mark Goh, and Bing Liang. Trust Region Methods for Solving Multiobjective Optimisation. *Optim. Methods Softw.*, 28(4):796–811, 2013.

65 Jong-Hyun Ryu and Sujin Kim. A Derivative-Free Trust-Region Method for Biobjective Optimization. *SIAM J. Optim.*, 24(1):334–362, 2014.

66 Oliver Schütze, Oliver Cuate, Adanay Martín, Sebastian Peitz, and Michael Dellnitz. Pareto Explorer: A Global/Local Exploration Tool for Many-Objective Optimization Problems. *Eng. Optim.*, 52(5):832–855, 2020.

67 Ozan Sener and Vladlen Koltun. Multi-Task Learning as Multi-Objective Optimization. `https://arxiv.org/abs/1810.04650`, 2019.

68 Everton J. Silva and Ana Luísa Custódio. An Inexact Restoration Direct Multisearch Filter Approach to Multiobjective Constrained Derivative-Free Optimization. *Optim. Methods Softw.*, 40(2):406–432, 2024.

69 Jana Thomann and Gabriele Eichfelder. A Trust-Region Algorithm for Heterogeneous Multiobjective Optimization. *SIAM J. Optim.*, 29(2):1017–1047, 2019.

70 Kely D. V. Villacorta, Paulo R. Oliveira, and Antoine Soubeyran. A Trust-Region Method for Unconstrained Multiobjective Problems with Applications in Satisficing Processes. *J. Optim. Theory Appl.*, 160(3):865–889, 2014.

71 Andreas Wächter and Lorenz T. Biegler. Line Search Filter Methods for Nonlinear Programming: Motivation and Global Convergence. *SIAM J. Optim.*, 16(1):1–31, 2005.

72 Andrea Walther and Lorenz T. Biegler. On an Inexact Trust-Region SQP-filter Method for Constrained Nonlinear Optimization. *Comput. Optim. Appl.*, 63(3):613–638, 2016.

73 Roger Jean-Baptiste Wets. On the Continuity of the Value of a Linear Program and of Related Polyhedral-Valued Multifunctions. In *Mathematical Programming Studies*, pages 14–29. Springer, 1985.

74 Stefan M. Wild. *Derivative-Free Optimization Algorithms For Computationally Expensive Functions.* PhD thesis, Cornell University, 2009.

75 Stefan M. Wild, Rommel G. Regis, and Christine A. Shoemaker. ORBIT: Optimization by Radial Basis Function Interpolation in Trust-Regions. *SIAM J. Sci. Comput.*, 30(6):3197–3219, 2008.

76 Outi Wilppu, Napsu Karmitsa, and M. Mäkelä. New Multiple Subgradient Descent Bundle Method for Nonsmooth Multiobjective Optimization. Turku Centre for Computer Science, 2014.