Worst-Case Complexity of High-Order Algorithms for Pareto-Front Reconstruction

Andrea Cristofari*, Marianna De Santis†, Stefano Lucidi‡, Giampaolo Liuzzi‡

*Department of Civil Engineering and Computer Science Engineering
University of Rome "Tor Vergata"
Via del Politecnico, 1, 00133 Rome, Italy
E-mail: andrea.cristofari@uniroma2.it

†Department of Information Engineering
Università degli studi di Firenze
Via di Santa Marta 3, 50139 Firenze, Italy
E-mail: marianna.desantis@unifi.it

‡Department of Computer, Control and Management Engineering
Sapienza Università di Roma
Via Ariosto 25, 00185 Roma, Italy
E-mail: liuzzi@diag.uniroma1.it, E-mail: lucidi@diag.uniroma1.it

Abstract. In this paper, we are concerned with a worst-case complexity analysis of a-posteriori algorithms for unconstrained multiobjective optimization. Specifically, we propose an algorithmic framework that generates sets of points by means of pth-order models regularized with a power $p+1$ of the norm of the step. Through a tailored search procedure, several trial points are generated at each iteration and they can be added to the current set if a decrease is obtained for at least one objective function. Building upon this idea, we devise two algorithmic versions: at every iteration, the first tries to update all points in the current set, while the second tries to update only one point. Under Lipschitz continuity of the derivatives of the objective functions, we derive worst-case complexity bounds for both versions. For the first one, we show that at most $\mathcal{O}(\epsilon^{-m(p+1)/p})$ iterations are needed to generate a set where all points are ϵ-approximate Pareto-stationary, with m being the number of objective functions, requiring at most $\mathcal{O}(|X(\epsilon)|\epsilon^{-m(p+1)/p})$ function evaluations, where $X(\epsilon)$ is the largest set of points computed by the algorithm. Additionally, at most $\mathcal{O}(\epsilon^{-(p+1)/p})$ iterations are needed to generate at least one ϵ-approximate Pareto-stationary point, requiring at most $\mathcal{O}(|X(\epsilon)|\epsilon^{-(p+1)/p})$ function evaluations. For the second version, we get $\mathcal{O}(\epsilon^{-m(p+1)/p})$ worst-case complexity bounds on the number of iterations and function evaluations for generating at least one ϵ-approximate Pareto-stationary point. Our results align with those for single objective optimization and generalize those obtained for methods that produce a single Pareto-stationary point.

Keywords. Multiobjective optimization. High-order methods. Worst-case iteration complexity.

MSC2000 subject classifications. 90C29. 90C30. 65Y20.

1 Introduction

In this paper, we consider unconstrained multiobjective optimization problems of the following form:

$$\min_{x\in\mathbb{R}^n} F(x) := (f_1(x), \ldots, f_m(x))^T, \tag{1}$$

with (possibly non-convex) objective functions $f_i\colon \mathbb{R}^n \to \mathbb{R}$, $i = 1, \ldots, m$ and $m \geq 2$.

For the solution of Problem (1), where we want to simultaneously minimize a number of conflicting objective functions, a common approach is represented by the so-called descent methods. They are characterized by the use of suitable directions that allow all objective functions to decrease when using an appropriate stepsize. Such a paradigm generates a sequence of points converging, over appropriate subsequences, to a Pareto-stationary point.

In particular, for first-order descent methods, a steepest descent method was originally proposed in [16], computing a descent direction by minimizing a first-order model with second-order regularization. When the gradients of the objective functions are Lipschitz continuous, the stationarity violation was proved in [17, 20] to converge to zero at a $\mathcal{O}(k^{-1/2})$ rate and provide a worst-case iteration complexity of $\mathcal{O}(\epsilon^{-2})$ to produce an approximate solution with precision $\epsilon > 0$. Extending that approach to the use of higher order derivatives, a descent algorithm was proposed in [4] where, at each iteration, a descent direction can be computed by approximately minimizing a pth-order model regularized with a power $p + 1$ of the norm of the step. Assuming that the pth derivatives of the objective functions are Hölder continuous with parameter $\beta \in [0, 1]$, it was shown a worst-case iteration complexity of $\mathcal{O}(\epsilon^{-(p+\beta)/(p+\beta-1)})$ for generating an approximate solution with precision $\epsilon > 0$, thus yielding to a $\mathcal{O}(\epsilon^{-(p+1)/p})$ worst-case complexity when the pth derivatives of the objective function are Lipschitz continuous (i.e., $\beta = 1$). These complexity bounds align with those obtained for single-objective optimization (see, e.g., [3, 5, 8, 19, 26]).

The above mentioned schemes for multiobjective optimization are able to produce a single Pareto-stationary point. This may be unsatisfactory when the goal is to approximate the entire Pareto set rather than computing just a single point. For this reason, several algorithms that build sequences of sets were recently proposed in the literature [9, 11, 10, 14, 13, 21, 22, 23, 25].

Following this trend, here we design and analyze an algorithmic approach that builds sequences of sets of points by means of high-order regularized models. In the proposed scheme, we are given a set of points X_k at every iteration k and, chosen any point in X_k (possibly all), we try to produce new points to be added to the set by minimizing an appropriate model where the pth-order Taylor approximation is regularized by a power $p+1$ of the norm of the step. Specifically, to select appropriate values for the regularization parameters, we generate several trial points using a search procedure that repeatedly increases the regularization parameters until a dominating point is obtained, that is, such that all the objective values are lower than the initial ones. A distinguishing feature of our approach is that not all the trial points computed in that search phase are discarded. Instead, they can be added to the current set of points if they make *at least one objective function* decrease.

For this paradigm, we consider two algorithmic versions: at every iteration k, the first version tries to update all points at its disposal in the current set X_k, while the second version

chooses only one point from X_k. Loosely speaking, the two versions meet two different needs: the first one aims to produce a more accurate approximation of the Pareto front but requires a major computational effort in practice. Conversely, the second strategy is more efficient from a computational standpoint.

Under Lipschitz continuity of the derivatives of the objective functions, for both versions we derive worst-case iteration and function evaluation complexity bounds for generating a set of approximate Pareto-stationary points. To the best of the authors' knowledge, this represents the first attempt to provide a worst-case analysis for algorithms that generate sets of points and use high order derivatives.

1.1 Main Contributions

Let us summarize the main contributions of the paper.

1. We devise an algorithmic framework for Pareto front reconstruction in unconstrained multiobjective optimization using high-order regularized models. Specifically, we give two algorithmic versions which differ in how the current set of points is updated at each iteration: while the first version tries to update each point of the set, the second one—a lighter version—tries to update only one arbitrarily chosen point of the set. As discussed earlier, these two versions are expected to achieve different levels of accuracy in the Pareto front reconstruction and different efficiency.

2. For both algorithmic versions, we provide a worst-case iteration and function evaluation complexity analysis for generating a set containing approximate Pareto-stationary points. Our results show that, for the algorithm that updates all points at its disposal at every iteration, we generate a set where all points are ϵ-approximate Pareto-stationary after at most $\mathcal{O}(\epsilon^{-m(p+1)/p})$ iterations, requiring at most $\mathcal{O}(|X(\epsilon)|\epsilon^{-m(p+1)/p})$ function evaluations, where $X(\epsilon)$ is the largest set of points computed by the algorithm. Moreover, to generate at least one ϵ-approximate Pareto-stationary point, we show a worst-case complexity of $\mathcal{O}(\epsilon^{-(p+1)/p})$ for the number of iterations and $\mathcal{O}(|X(\epsilon)|\epsilon^{-(p+1)/p})$ for the number of function evaluations. For the light algorithm, which is much less expensive, only the latter type of complexity bounds can be established (i.e., to generate at least one ϵ-approximate Pareto-stationary point), which are $\mathcal{O}(\epsilon^{-m(p+1)/p})$ for both the number of iterations and the number of function evaluations.

Moreover, we investigate the use of approximate conditions to solve the (possibly non-convex) high-order regularized models inexactly, thus mitigating the difficulty connected with their practical solution. Our analysis shows that the complexity bounds remain the same (up to the constant factors) when approximate minimizers are used.

1.2 Notation and contents

Given a vector $v \in \mathbb{R}^n$, we denote by v_i the ith entry of v, while $\|v\|$ denotes the Euclidean norm of v. Given two vectors $u, v \in \mathbb{R}^m$, then $u \leq v$ ($u < v$) if and only if $u_i \leq v_i$ ($u_i < v_i$) for all $i = 1, \ldots, m$. Given two vectors $x, y \in \mathbb{R}^n$, we say that x dominates y, and we

write $x \prec y$, if $F(x) \leq F(y)$ and $F(x) \neq F(y)$. Given a set of points $X \subseteq \mathbb{R}^n$, we define $F(X) := \{z \in \mathbb{R}^m : \ z = F(x) \text{ for } x \in X\}$. The volume of a set $X \subseteq \mathbb{R}^n$ is denoted by $\text{Vol}(X)$. By $\mathbf{1}$ we denote the vector of all ones of appropriate dimensions, typically $\mathbf{1} \in \mathbb{R}^m$. We indicate by $\mathbb{R}^m_+$ the set of vectors in $\mathbb{R}^m$ having all components strictly positive.

The rest of the paper is organized as follows. In Section 2, we recall some concepts on multiobjective optimization problems and point out some results on quadratically regularized models. In Section 3, we analyze high-order regularized models. In Section 4, we describe the proposed algorithmic approach and carry out a worst-case complexity analysis, using both exact and inexact minimizers of the regularized model. In Section 5, we describe a light version of the algorithm and give worst-case complexity bounds. In Section 6, we report some numerical examples to show the behavior of the proposed algorithms. Finally, some conclusions are drawn in Section 7.

2 Preliminaries

In this section we introduce some preliminary material. First, let us recall the definitions of Pareto-optimality and -stationarity for Problem (1).

Definition 1 (Pareto optimality)**.** *A point $x^* \in \mathbb{R}^n$ is said to be Pareto optimal (or efficient) for Problem* (1) *if there does not exist $x \in \mathbb{R}^n$ such that $x \prec x^*$, that is $F(x) \leq F(x^*)$, with $F(x) \neq F(x^*)$.*

Proposition 1. *Let $x^* \in \mathbb{R}^n$ be a Pareto optimal point for Problem* (1)*. Then, for any $d \in \mathbb{R}^n$, an index $j \in \{1, \ldots, m\}$ exists such that*

$$\nabla f_j(x^*)^T d \geq 0.$$

Definition 2 (Pareto-stationary point)**.** *A point $x^* \in \mathbb{R}^n$ is Pareto-stationary for Problem* (1) *if, for any $d \in \mathbb{R}^n$, an index $j \in \{1, \ldots, m\}$ exists such that*

$$\nabla f_j(x^*)^T d \geq 0.$$

Now, starting from the above results, we can also give the definition and a characterization of efficient sets.

Definition 3 (Efficient set)**.** *A set of points $\mathcal{E} \subseteq \mathbb{R}^n$ is said to be an efficient set if every point $x \in \mathcal{E}$ is a Pareto optimal point of Problem* (1)*.*

Proposition 2. *Let $\mathcal{E} \subseteq \mathbb{R}^n$ be an efficient set for Problem* (1)*. Then, for any $x \in \mathcal{E}$ and $d \in \mathbb{R}^n$, there exists an index $i \in \{1, \ldots, m\}$ such that $\nabla f_i(x)^T d \geq 0$.*

Stronger conditions to characterize the efficient set $\mathcal{E}$ were also given in [12]. Here, according to Proposition 2, we give the following definition of a Pareto-stationary set.

Definition 4 (Pareto-stationary set)**.** *Let $S \subseteq \mathbb{R}^n$ be a non-empty set. We say that S is a Pareto-stationary set for Problem* (1) *if, for any $x \in S$ and $d \in \mathbb{R}^n$, there exists an index $i \in \{1, \ldots, m\}$ such that $\nabla f_i(x)^T d \geq 0$.*

To measure the quality of a set of points in the image space (i.e., $\mathbb{R}^m$) of a multiobjective problem, it is common to use the hypervolume indicator. Before giving its definition, we need the following assumption.

Assumption 1. *For all $i = 1, \dots, m$, and given $x_0 \in \mathbb{R}^n$, it holds that*

$$-\infty < f_i^{\min} := \inf\{f_i(x) \colon x \in \mathbb{R}^n\},$$
$$+\infty > f_i^{\max} := \sup\{f_i(x) \colon x \in \mathbb{R}^n \text{ and } x_0 \not\prec x\}.$$

Let us also denote $f_{\max} := (f_1^{\max}, \dots, f_m^{\max})^T \in \mathbb{R}^m$ and $f_{\min} := (f_1^{\min}, \dots, f_m^{\min})^T \in \mathbb{R}^m$.

Definition 5 (Hypervolume indicator [14, Definition 3.1])**.** *The hypervolume indicator for some set $A \subset \mathbb{R}^m$ and a reference point $\rho \in \mathbb{R}^m$ that is dominated by all the points in A is defined as*

$$HI(A) := \mathrm{Vol}\left(\{b \in \mathbb{R}^m : b \leq \rho \text{ and } \exists a \in A : a \leq b\}\right) = \mathrm{Vol}\left(\bigcup_{a \in A} [a, \rho]\right),$$

where $[a, \rho] = \{y \in \mathbb{R}^m : \ a_i \leq y_i \leq \rho_i, \ i = 1, \dots, m\}$.

A key result is expressed in the following lemma. It relates the increase of the hypervolume in the image space when, given a set of points, a new point is added which is nondominated with respect to all other points of the set.

Lemma 1 ([14, Lemma 3.1])**.** *Let $Y \subseteq \mathbb{R}^n$, $x \in \mathbb{R}^n \setminus Y$, and $\nu > 0$ such that*

$$F(x) \not\prec F(y) - \nu \mathbf{1} \quad \forall y \in Y.$$

Let the reference point be $\rho = f_{\max} + \alpha \mathbf{1}$ (with $\alpha > 0$ being sufficiently large). Then,

$$HI(F(Y \cup \{x\})) - HI(F(Y)) \geq \nu^m.$$

Note that, under Assumption 1, given the efficient set $\mathcal{E} \subseteq \mathbb{R}^n$ of Problem (1), $\overline{HI} \in \mathbb{R}$ exists such that

$$HI(F(\mathcal{E})) \leq \overline{HI} < +\infty. \tag{2}$$

2.1 Quadratically regularized models

In this subsection, we give some results on quadratically regularized models applied to Problem (1), laying the groundwork for the proposed algorithmic framework, which extends those concepts to higher order models dealing with sets of points.

To begin with, given $x \in \mathbb{R}^n$, consider the following problem:

$$\min_{v \in \mathbb{R}^n} \max_{i=1,\dots,m} \nabla f_i(x)^T v + \frac{1}{2}\|v\|^2. \tag{3}$$

We see that (3) consists of a first-order term with quadratic regularization. It is a strongly convex problem and has a unique optimal solution, which will be denoted by $v(x)$, that is,

$$v(x) \in \operatorname*{argmin}_{v \in \mathbb{R}^n} \max_{i=1,\dots,m} \nabla f_i(x)^T v + \frac{1}{2}\|v\|^2. \tag{4}$$

From [16], we get the following result.

Proposition 3 ([16]). *Given $x \in \mathbb{R}^n$,the following holds:*

(i) the optimal value of Problem (3) *is non-positive,*

(ii) x^* is Pareto-stationary if and only if $\|v(x^*)\| = 0$,

(iii) the mapping $x \to v(x)$ is continuous.

In view of the above proposition, we see that Problem (3) has a twofold role: it provides an amount of Pareto-stationarity violation at a given point x and, if $v(x) \neq 0$, the latter can be used as a descent direction.

We also note that, if $m = 1$, then $v(x) = -\nabla f_1(x)$, thus reducing to the anti-gradient direction for single-objective optimization.

Moreover, Problem (3) can be equivalently reformulated as

$$\begin{aligned} &\min_{(v,\alpha)\in\mathbb{R}^{n+1}} \alpha + \frac{1}{2}\|v\|^2 \\ &\text{s.t. } \nabla f_i(x)^T v \leq \alpha, \quad i = 1, \ldots, m. \end{aligned} \tag{5}$$

It is straightforward to verify that v is an optimal solution of (3) if and only if there exists $\alpha \in \mathbb{R}$ such that (v, α) is an optimal solution of (5). Moreover, for Problem (5), we can define the Lagrangian function

$$L(v, \alpha, \lambda) = \alpha + \frac{1}{2}\|v\|^2 + \sum_{i=1}^{m} \lambda_i \Big(\nabla f_i(x)^T v - \alpha \Big)$$

and give the following KKT conditions:

$$v + \sum_{i=1}^{m} \lambda_i \nabla f_i(x) = 0, \tag{6a}$$

$$\sum_{i=1}^{m} \lambda_i = 1, \tag{6b}$$

$$\lambda_i (\nabla f_i(x)^T v - \alpha) = 0, \quad i = 1, \ldots, m, \tag{6c}$$

$$\nabla f_i(x)^T v - \alpha \leq 0, \quad i = 1, \ldots, m, \tag{6d}$$

$$\lambda_i \geq 0, \quad i = 1, \ldots, m. \tag{6e}$$

Let us conclude this section by giving a key result which we will use for high-order models to analyze the first-order stationarity, expressed in terms of $\|v(x)\|$ in view of Proposition 3.

Lemma 2. *Given $x \in \mathbb{R}^n$, let $v(x)$ be defined as in* (4)*. For all $u \in \mathbb{R}^m$ such that $\sum_{i=1}^m u_i = 1$ and $u \geq 0$, we have that*

$$\|v(x)\| \leq \left\| \sum_{i=1}^{m} u_i \nabla f_i(x) \right\|.$$

Proof. Since $v(x)$ is the unique optimal solution of Problem (3), then an optimal solution of (5) is given by $(v(x), \alpha(v(x)))$, with $\alpha(v) := \max_{i=1,\dots,m} \nabla f_i(x)^T v$. Since (5) is a linearly constrained problem, then the constraint qualification holds and $(v(x), \alpha(v(x)))$ satisfies the KKT conditions (6a)–(6e) with a multiplier vector $\bar{\lambda} \in \mathbb{R}^m$ (see, e.g., [1]). In particular, from (6a), (6b) and (6e), we can write

$$v(x) = -\sum_{i=1}^{m} \bar{\lambda}_i \nabla f_i(x), \quad \sum_{i=1}^{m} \bar{\lambda}_i = 1, \quad \bar{\lambda} \geq 0. \tag{7}$$

Now, consider the dual of Problem (5), given by

$$\begin{aligned} \max_{\lambda \in \mathbb{R}^m} & -\frac{1}{2} \left\| \sum_{i=1}^{m} \lambda_i \nabla f_i(x) \right\|^2 \\ \text{s.t.} & \sum_{i=1}^{m} \lambda_i = 1, \\ & \lambda \geq 0. \end{aligned} \tag{8}$$

From strong duality, it follows that $\bar{\lambda}$ is an optimal solution for the dual Problem (8). Hence,

$$-\left\| \sum_{i=1}^{m} \bar{\lambda}_i \nabla f_i(x) \right\|^2 \geq -\left\| \sum_{i=1}^{m} u_i \nabla f_i(x) \right\|^2 \quad \forall u \in \mathbb{R}^m \text{ such that } \sum_{i=1}^{m} u_i = 1 \text{ and } u \geq 0.$$

Recalling (7), the left-hand side of the above inequality is equal to $-\|v(x)\|^2$, yielding to the desired result. □

3 High-order regularized models

From an algorithmic perspective, a solution of Problem (3) can be used to obtain a descent direction for all objective functions (see Proposition 3). Building upon this idea, a steepest descent method was defined in [16] by moving, at each iteration k, from the current point x_k along a direction $v(x_k)$ computed as in (4) and using a stepsize obtained by a line search procedure. Under the assumption that ∇f_i, $i = 1, \dots, m$, are Lipschitz continuous, a convergence rate of $\mathcal{O}(k^{-1/2})$ for the first-order stationarity violation was proved for such a scheme in [17], leading to a worst-case iteration complexity of $\mathcal{O}(\epsilon^{-2})$ to drive the first-order stationarity violation below a given $\epsilon > 0$. The same convergence rate and worst-case iteration complexity were then obtained in [20] for a broader class of first-order methods.

Since Problem (3) consists of a first-order term with quadratic regularization, it appears natural to define a more general framework using models of order $p \geq 1$ with regularization of order $p + 1$, extending ideas from single-objective optimization [3]. Such an approach was investigated in [4] under Hölder continuity of the derivatives (thus including Lipschitz continuity as a special case). However, while the algorithm designed in [4] is able to produce only a single Pareto-stationary point, here we want to define an algorithm able to approximate the Pareto front.

Let us assume that, for a given $p \geq 1$ and all $i = 1, \ldots, m$, the pth derivatives of f_i, denoted by $\nabla^p f_i$, are Lipschitz continuous with constant $L_i > 0$. Namely,

$$\|\nabla^p f_i(x) - \nabla^p f_i(y)\|_{[p]} \leq L_i \|x - y\| \quad \forall x, y \in \mathbb{R}^n,$$

where, using a standard notation (see, e.g., [3, 7]), $\|\cdot\|_{[p]}$ is the tensor norm recursively induced on the space of pth-order tensors, which is given by

$$\|T\|_{[p]} := \max_{\|v_1\|=\cdots=\|v_p\|=1} |T[v_1, \ldots, v_p]|,$$

with $T[v_1, \cdots, v_j]$ indicating the tensor of order $p - j \geq 0$ resulting from the application of the pth-order tensor T to the vectors $v_1, \ldots, v_j$.

Then, given a point $x \in \mathbb{R}^n$ and a regularization parameter vector $\sigma \in \mathbb{R}^m_+$, we can consider the following problem:

$$\min_{s \in \mathbb{R}^n} \max_{i=1,\ldots,m} \sum_{j=1}^{p} \frac{1}{j!} \nabla^j f_i(x)[s]^j + \frac{\sigma_i}{p!} \|s\|^{p+1}, \tag{9}$$

where $\nabla^j f_i(x)[s]^j$ stands for $\nabla^j f_i(x)$ applied j times to the vector s.

To simplify the notation, given $x \in \mathbb{R}^n$ and $\sigma \in \mathbb{R}^m_+$, let us define

$$\begin{aligned} m_i^p(s; x, \sigma_i) &:= \sum_{j=1}^{p} \frac{1}{j!} \nabla^j f_i(x)[s]^j + \frac{\sigma_i}{p!} \|s\|^{p+1}, \quad i = 1, \ldots, m, \\ m^p(s; x, \sigma) &:= \max_{i=1,\ldots,m} m_i^p(s; x, \sigma_i), \end{aligned}$$

so that Problem (9) can be rewritten as

$$\min_{s \in \mathbb{R}^n} m^p(s; x, \sigma).$$

We see that each $m_i^p(s; x, \sigma_i)$ is obtained by computing the Taylor series $T_i^p(x, s)$ of the function $f_i(x + s)$ at x truncated at order p, then subtracting $f_i(x)$ (which is constant with respect to s) and adding a regularization term of order $p + 1$. Namely,

$$m_i^p(s; x, \sigma_i) = T_i^p(x, s) - f_i(x) + \frac{\sigma_i}{p!} \|s\|^{p+1}, \quad i = 1, \ldots, m,$$

where

$$T_i^p(x, s) := f_i(x) + \sum_{j=1}^{p} \frac{1}{j!} \nabla^j f_i(x)[s]^j, \quad i = 1, \ldots, m.$$

Note that, when $p = 1$ in (9), we recover the strongly convex Problem (3) (up to a constant in the regularization terms). However, when $p > 1$, Problem (9) is no longer convex. In the next result, we highlight that the optimal value of $m^p(; x, \sigma)$ with respect to s cannot be positive.

Proposition 4. *Given $x \in \mathbb{R}^n$ and $\sigma \in \mathbb{R}^m_+$, let $\bar{s}$ be an optimal solution of Problem (9). Then,*

$$m^p(\bar{s}; x, \sigma) \leq m^p(0; x, \sigma) = 0,$$

that is, $m^p_i(\bar{s}; x, \sigma_i) \leq m^p_i(0; x, \sigma_i) = 0$ for all $i = 1, \ldots, m$.

Proof. It follows straightforwardly from the definition of Problem (9). □

Reasoning similarly as in Section 2.1, we can give an equivalent reformulation of Problem (9) as

$$\begin{aligned} &\min_{(s,t)\in\mathbb{R}^{n+1}} \ t \\ &\text{s.t. } m^p_i(s; x, \sigma_i) \leq t, \quad i = 1, \ldots, m. \end{aligned} \tag{10}$$

For Problem (10), we define the Lagrangian function as

$$L(s, t, \lambda) = t + \sum_{i=1}^{m} \lambda_i (m^p_i(s; x, \sigma_i) - t),$$

so that the KKT conditions can be expressed as follows:

$$\sum_{i=1}^{m} \lambda_i \nabla_s m^p_i(s; x, \sigma_i) = 0, \tag{11a}$$

$$\sum_{i=1}^{m} \lambda_i = 1, \tag{11b}$$

$$\lambda_i (m^p_i(s; x, \sigma_i) - t) = 0, \quad i = 1, \ldots, m, \tag{11c}$$

$$m^p_i(s; x, \sigma_i) - t \leq 0, \quad i = 1, \ldots, m, \tag{11d}$$

$$\lambda_i \geq 0, \quad i = 1, \ldots, m, \tag{11e}$$

where $\nabla_s m^p_i(s; x, \sigma_i)$ denotes the gradient of $m^p_i(s; x, \sigma_i)$ with respect to s.

In the following result, we point out that a constraint qualification holds for Problem (10), so that the KKT conditions (11a)–(11e) are indeed necessary for optimality.

Proposition 5. *Given $x \in \mathbb{R}^n$ and $\sigma \in \mathbb{R}^m_+$, let (s, t) be an optimal solution of Problem (10). Then, there exists $\lambda \in \mathbb{R}^m$ such that KKT conditions (11a)–(11e) are satisfied.*

Proof. Since (s, t) is an optimal solution for Problem (10), using the Fritz-John conditions [1], there exist $\lambda_0, \lambda_1, \ldots, \lambda_m \in \mathbb{R}$ such that

$$\sum_{i=1}^{m} \lambda_i \nabla_s m^p_i(s; x, \sigma_i) = 0, \tag{12a}$$

$$\sum_{i=1}^{m} \lambda_i = \lambda_0, \tag{12b}$$

$$\lambda_i (m^p_i(s; x, \sigma_i) - t) = 0, \quad i = 1, \ldots, m, \tag{12c}$$

$$m^p_i(s; x, \sigma_i) - t \leq 0, \quad i = 1, \ldots, m, \tag{12d}$$

$$\lambda_i \geq 0, \quad i = 0, 1, \ldots, m, \tag{12e}$$

$$(\lambda_0, \lambda_1, \ldots, \lambda_m) \neq (0, 0, \ldots, 0). \tag{12f}$$

To prove the result, we need to show that $\lambda_0 > 0$ (so that, by dividing each λ_i by λ_0, the KKT conditions (11a)–(11e) hold). Reasoning by contradiction, assume that $\lambda_0 = 0$. In view of (12b) and (12e), it follows that $\lambda_i = 0$ for all $i = 0, 1, \ldots, m$, thus contradicting (12f). □

Let us finally state useful properties of $f_1, \ldots, f_m$ coming from known results on functions with Lipschitz continuous pth-order derivatives [3, 7]: for all $i = 1, \ldots, m$, it holds that

$$|f_i(x+s) - T_i^p(x,s)| \leq \frac{L_i}{p!}\|s\|^{p+1}, \quad \forall x, s \in \mathbb{R}^n, \tag{13}$$

$$\|\nabla f_i(x+s) - \nabla_s T_i^p(x,s)\| \leq \frac{L_i}{(p-1)!}\|s\|^{p}, \quad \forall x, s \in \mathbb{R}^n. \tag{14}$$

Moreover, in the remainder of the paper, we will denote the m-dimensional vector

$$L := \begin{bmatrix} L_1 & \ldots & L_m \end{bmatrix}^T$$

and use the value $L^{\max}$ defined as follows:

$$L^{\max} := \max_{i=1,\ldots,m} L_i. \tag{15}$$

4 HOP: a High-Order algorithm for Pareto-front reconstruction

In this section, we describe the proposed *High-Order algorithm for Pareto-front reconstruction*, that is, Algorithm HOP. At each iteration k, we are given a set X_k of mutually nondominated points and we try to update X_k by means of the *Regularized Search* (RS) procedure. Specifically, for any non-Pareto-stationary point x_k in X_k, the RS procedure generates several trial points by computing (possibly approximate) minimizers of the regularized model $m^p(s; x_k, \sigma)$ with respect to s using different choices for the regularization parameters σ. More precisely, we execute a while loop where we repeatedly increase the regularization parameters associated with the objective functions that do not satisfy a sufficient decrease condition in the trial points. The while loop stops only when all the objective functions satisfy the sufficient decrease condition, then returning $s(x_k)$ such that $x_k + s(x_k) \prec x_k$. Notably, the trial points produced in the while loop are not necessarily discarded. Instead, any trial point that makes at least one objective function decrease is considered a candidate to be included in the set X_{k+1} for the next iteration.

Observe that RS procedure initially chooses the regularization parameter vector in $[\sigma_l, \sigma_u]$, with σ_l and σ_u being vectors of finite positive real numbers. In the remainder of the paper, we denote

$$\sigma_u^{\max} := \max_{i=1,\ldots,m} (\sigma_u)_i,$$
$$\sigma_l^{\min} := \min_{i=1,\ldots,m} (\sigma_l)_i.$$

Algorithm 1 High-Order algorithm for Pareto-front reconstruction (HOP)

given $X_0 \subseteq \mathbb{R}^n$
set $k = 0$
while X_k is not Pareto-stationary **do**
    set $\tilde{X}_k = X_k$
    for all $x_k \in X_k$ **do**
        if x_k is not Pareto-stationary **then**
            compute $Y(x_k)$, $s(x_k)$, $\sigma(x_k)$ by `Regularized Search`(x_k, X_k)
            set $\tilde{X}_k \leftarrow \tilde{X}_k \cup Y(x_k)$
        end if
    end for
    set X_{k+1} as the nondominated points in $\tilde{X}_k$
    set $k \leftarrow k + 1$
end while

Algorithm 2 `Regularized Search`(x, X)

input: x, X; **output:** Y, s, σ
given $\eta \in (0,1)$, $\delta \in (0,1)$ and $0 < (\sigma_l)_i \leq (\sigma_u)_i < \infty$, $i = 1, \ldots, m$
set $j = 0$, $Y^j = X$ and choose $\sigma^j \in [\sigma_l, \sigma_u]$
compute a (possibly approximate) minimizer s^j of $m^p(s; x, \sigma^j)$
while $F(x + s^j) \not\leq F(x) - \eta \dfrac{\|s^j\|^{p+1}}{p!}\sigma^j$ **do**
    if $F(x + s^j) \not\succ F(y) - \eta \dfrac{\|s^j\|^{p+1}}{p!}\sigma^j$ for all $y \in Y^j$ **then**
        set $Y^{j+1} = Y^j \cup \{x + s^j\}$
    else
        set $Y^{j+1} = Y^j$
    end if
    set σ^{j+1} such that, for all $i = 1, \ldots, m$,

$$(\sigma^{j+1})_i = \begin{cases} (\sigma^j)_i/\delta & \text{if } f_i(x + s^j) > f_i(x) - \eta\dfrac{\|s^j\|^{p+1}}{p!}(\sigma^j)_i \\ (\sigma^j)_i & \text{otherwise} \end{cases}$$

    compute a (possibly approximate) minimizer s^{j+1} of $m^p(s; x, \sigma^{j+1})$
    set $j \leftarrow j + 1$
end while
set $Y^{j+1} = Y^j \cup \{x + s^j\}$
return $Y = Y^{j+1}$, $s = s^j$ and $\sigma = \sigma^j$

4.1 Using exact minimizers of the regularized model

First, we assume for simplicity that, for any $\sigma \in \mathbb{R}^m_+$, each s^j computed in the RS procedure is a global minimizer of the regularized model $m^p(s; x, \sigma^j)$ with respect to s, that is,

$$s^j \in \operatorname*{argmin}_{s \in \mathbb{R}^n} m^p(s; x, \sigma^j). \tag{16}$$

We now show how to relate the norm of an optimal solution s of $\min_{s\in\mathbb{R}^n} m^p(s; x, \sigma)$ to $\|v(x + s)\|$, the latter measuring the amount of first-order stationarity violation at $x + s$ according to Proposition 3.

Proposition 6. *Given $x \in \mathbb{R}^n$ and $\sigma \in \mathbb{R}^m_+$, let $\bar{s} \in \operatorname{argmin}_{s\in\mathbb{R}^n} m^p(s; x, \sigma)$. Then,*

$$\|v(x + \bar{s})\| \leq \left(\frac{(p+1)\max_{i=1,\dots,m} \sigma_i}{p!} + \frac{L^{max}}{(p-1)!} \right) \|\bar{s}\|^p,$$

where $v(\cdot)$ is defined as in (4).

Proof. Let $t := m^p(\bar{s}; x, \sigma)$. Since $\bar{s} \in \operatorname{argmin}_{s\in\mathbb{R}^n} m^p(s; x, \sigma)$, it follows that $(\bar{s}, t)$ is an optimal solution of Problem (10). Using Proposition 5, there exists a KKT multiplier vector $\lambda \in \mathbb{R}^m$ satisfying (11a)–(11e) at $(\bar{s}, t)$. Using Lemma 2 and the triangle inequality, we have that

$$\begin{aligned}
\|v(x + \bar{s})\| &\leq \left\| \sum_{i=1}^m \lambda_i \nabla f_i(x + \bar{s}) \right\| \\
&= \left\| \sum_{i=1}^m \lambda_i (\nabla_s T_i^p(x, \bar{s}) + \nabla f_i(x + \bar{s}) - \nabla_s T_i^p(x, \bar{s})) \right\| \\
&\leq \left\| \sum_{i=1}^m \lambda_i \nabla_s T_i^p(x, \bar{s}) \right\| + \left\| \sum_{i=1}^m \lambda_i (\nabla f_i(x + \bar{s}) - \nabla_s T_i^p(x, \bar{s})) \right\|.
\end{aligned} \tag{17}$$

To upper bound the first norm in the last term of (17), using the triangle inequality we can write

$$\begin{aligned}
&\left\| \sum_{i=1}^m \lambda_i \nabla_s T_i^p(x, \bar{s}) \right\| = \\
&\left\| \sum_{i=1}^m \lambda_i \Big(\nabla_s T_i^p(x, \bar{s}) + \frac{(p+1)\sigma_i}{p!} \|\bar{s}\|^{p-1} \bar{s} - \frac{(p+1)\sigma_i}{p!} \|\bar{s}\|^{p-1} \bar{s} \Big) \right\| \leq \\
&\left\| \sum_{i=1}^m \lambda_i \Big(\nabla_s T_i^p(x, \bar{s}) + \frac{(p+1)\sigma_i}{p!} \|\bar{s}\|^{p-1} \bar{s} \Big) \right\| + \sum_{i=1}^m \lambda_i \frac{(p+1)\sigma_i}{p!} \|\bar{s}\|^p = \\
&\left\| \sum_{i=1}^m \lambda_i (\nabla_s m_i^p(\bar{s}; x, \sigma_i)) \right\| + \sum_{i=1}^m \lambda_i \frac{(p+1)\sigma_i}{p!} \|\bar{s}\|^p \leq \\
&\frac{(p+1)\max_{i=1,\dots,m} \sigma_i}{p!} \|\bar{s}\|^p,
\end{aligned} \tag{18}$$

where we have used (11e) in the first inequality, while the last inequality follows from (11a) and (11b). To upper bound the second norm in the last term of (17), using the triangle inequality we can write

$$\begin{aligned}\left\|\sum_{i=1}^{m}\lambda_i(\nabla f_i(x+\bar{s})-\nabla_s T_i^p(x,\bar{s}))\right\| &\leq \sum_{i=1}^{m}\lambda_i\left\|\nabla f_i(x+\bar{s})-\nabla_s T_i^p(x,\bar{s})\right\| \\ &\leq \sum_{i=1}^{m}\lambda_i\frac{L_i}{(p-1)!}\|\bar{s}\|^p \\ &\leq \frac{L^{\max}}{(p-1)!}\|\bar{s}\|^p,\end{aligned} \tag{19}$$

where, in the first two inequalities, we have used (11e) and (14), respectively, while the third inequality follows from (11b) and the definition of $L^{\max}$ given in (15). Combining (17), (18) and (19), the desired result follows. □

Remark 1. *Proposition 6 guarantees that, for any $\sigma \in \mathbb{R}^m_+$, a point $x \in \mathbb{R}^n$ is Pareto-stationary for Problem* (1) *if $0 \in \operatorname{argmin}_{s\in\mathbb{R}^n} m^p(s;x,\sigma)$ (see item (ii) of Proposition 3).*

The following Propositions 7–8 show that the while loop in the RS procedure ends after solving a finite number of regularized models. In particular, for each non-Pareto-stationary point $x_k \in X_k$, we get $s(x_k)$ such that $x_k + s(x_k) \prec x_k$.

Proposition 7. *Given $x \in \mathbb{R}^n$, assume that $\sigma_i \geq (1-\eta)^{-1}L_i$, $i = 1,\dots,m$, with $\eta \in (0,1)$, and $\bar{s} \in \operatorname{argmin}_{s\in\mathbb{R}^n} m^p(s;x,\sigma)$. Then,*

$$F(x+\bar{s}) \leq F(x) - \eta\,\frac{\|\bar{s}\|^{p+1}}{p!}\sigma.$$

Proof. Without loss of generality, assume that $\bar{s} \neq 0$ (otherwise, x would be Pareto-stationary for Problem (1), according to Remark 1, and the result would trivially follow). Consider any objective index $i \in \{1,\dots,m\}$. Define

$$\rho_i := \frac{f_i(x) - f_i(x+\bar{s})}{T_i^p(x,0) - T_i^p(x,\bar{s})}. \tag{20}$$

Observe that

$$T_i^p(x,0) = f_i(x)$$

and

$$T_i^p(x,0) - T_i^p(x,\bar{s}) = f_i(x) - T_i^p(x,\bar{s}) = -m_i^p(\bar{s};x,\sigma_i) + \frac{\sigma_i}{p!}\|\bar{s}\|^{p+1} \geq \frac{\sigma_i}{p!}\|\bar{s}\|^{p+1}, \tag{21}$$

where the last inequality follows from Proposition 4. Hence, we get

$$1-\rho_i = \frac{f_i(x+\bar{s}) - T_i^p(x,\bar{s})}{-m_i^p(\bar{s};x,\sigma_i) + (\sigma_i/p!)\|\bar{s}\|^{p+1}} \leq \frac{f_i(x+\bar{s}) - T_i^p(x,\bar{s})}{(\sigma_i/p!)\|\bar{s}\|^{p+1}}.$$

In the last term, the numerator can be upper bounded by $(L_i/p!)\|\bar{s}\|^{p+1}$ using (13), leading to

$$1 - \rho_i \leq \frac{L_i}{\sigma_i},$$

Since, by hypothesis, we have $\sigma_i \geq L_i/(1-\eta)$, with $1-\eta > 0$, it follows that $1-\rho_i \leq 1-\eta$, that is, $\rho_i \geq \eta$. Namely, recalling (20) and (21), we have that

$$f_i(x) - f_i(x+\bar{s}) \geq \eta \frac{\sigma_i}{p!}\|\bar{s}\|^{p+1},$$

where the last inequality follows from Proposition 4 and the fact that $\eta > 0$. □

Proposition 8. *Assume that each s^j in the RS procedure satisfies* (16)*. Given a non-Pareto-stationary point $x_k \in X_k$ generated at iteration k of Algorithm HOP, the RS procedure computes $\sigma(x_k) \leq \max\{(\delta(1-\eta))^{-1}L, \sigma_u\}$ such that*

$$F(x_k + s(x_k)) \leq F(x_k) - \eta \frac{\|s(x_k)\|^{p+1}}{p!}\sigma(x_k).$$

Proof. It follows from Proposition 7 and the instructions of the RS procedure. □

Remark 2. *For any iteration k and non-Pareto-stationary point $x_k \in X_k$, it follows from Proposition 8 and the instructions of the RS procedure that*

$$(\sigma(x_k))_i \leq \sigma^{\max} := \max\{(\delta(1-\eta))^{-1}L^{max}, \sigma_u^{\max}\}, \quad i = 1, \ldots, m. \tag{22}$$

Remark 3. *The maximum number of function evaluations n_F required by the RS procedure at every iteration is upper bounded by 1 plus the maximum number of times we increase the regularization parameter for each objective function f_i, $i = 1, \ldots, m$. Hence, if each s^j in the RS procedure satisfies* (16)*, by Proposition 8 we have*

$$n_F := \max_{i=1,\ldots,m} \left\lceil \frac{1}{\log \delta} \log\left(\frac{(\sigma_l)_i}{\max\{(\delta(1-\eta))^{-1}L_i, (\sigma_u)_i\}} \right) \right\rceil + 1. \tag{23}$$

Now we show that, at every iteration k, there is no point $x \in X_k$ such that $x \prec x_k + s(x_k)$, that is, new nondominated points can be added to the current set X_k.

Proposition 9. *Assume that each s^j in the RS procedure satisfies* (16)*. Given a non-Pareto-stationary point $x_k \in X_k$ generated at iteration k of Algorithm HOP, we have that*

$$F(x_k + s(x_k)) \not\succ F(y) - \eta \frac{\|s(x_k)\|^{p+1}}{p!}\sigma(x_k), \quad \forall y \in X_k.$$

Proof. Since $x_k \in X_k$ is non-Pareto-stationary for Problem (1), by Proposition 8 we can write

$$f_i(x_k + s(x_k)) \leq f_i(x_k) - \eta \frac{\|s(x_k)\|^{p+1}}{p!}(\sigma(x_k))_i, \quad \forall i = 1, \ldots, m. \tag{24}$$

Since, by the instructions of the algorithm, there is no point in $X_k \setminus \{x_k\}$ that dominates x_k, then, for any given $y \in X_k \setminus \{x_k\}$, an index $\hat{\imath} \in \{1, \ldots, m\}$ exists such that

$$f_{\hat{\imath}}(y) > f_{\hat{\imath}}(x_k),$$

which, by (24), leads to

$$f_{\hat{\imath}}(x_k + s(x_k)) < f_{\hat{\imath}}(y) - \eta \frac{\|s(x_k)\|^{p+1}}{p!} (\sigma(x_k))_{\hat{\imath}}, \quad \forall y \in X_k \setminus \{x_k\}. \tag{25}$$

The desired result is thus obtained by combining (24) and (25). □

Using Propositions 6 and 8, we now show that, whenever a new point is added to the set $Y^j(x_k)$ during the while loop in the RS procedure, the Pareto-stationarity violation can be related to the decrease obtained in one objective function. Furthermore, by Proposition 1, this decrease corresponds to an increase in the hypervolume of the set $F(Y^j(x_k))$. This is formalized in the following result.

Proposition 10. *Assume that each s^j in the RS procedure satisfies* (16)*. Given a non-Pareto-stationary point $x_k \in X_k$ generated at iteration k of Algorithm HOP, consider the jth iteration of the RS procedure invoked with $x = x_k$ and $X = X_k$. Let $\sigma^j(x_k) := \sigma^j$, $s^j(x_k) := s^j$ and $Y^j(x_k) := Y^j$. If $Y^{j+1}(x_k) \neq Y^j(x_k)$, using $v(\cdot)$ defined as in* (4)*, then*

(i) an index $\hat{\imath} \in \{1, \ldots, m\}$ exists such that

$$f_{\hat{\imath}}(x_k) - f_{\hat{\imath}}(x_k + s^j(x_k)) \geq c \|v(x_k + s^j(x_k))\|^{(p+1)/p},$$

(ii) $HI(F(Y^{j+1}(x_k))) - HI(F(Y^j(x_k))) \geq c^m \|v(x_k + s^j(x_k))\|^{m(p+1)/p}$,

where

$$c := \left(\eta \frac{\sigma_l^{\min}}{p!}\right) \left(\frac{(p+1)\sigma^{\max}}{p!} + \frac{L^{max}}{(p-1)!}\right)^{-(p+1)/p} \tag{26}$$

and $\sigma^{\max}$ is defined as in (22)*.*

Proof. Since $Y^{j+1}(x_k) \neq Y^j(x_k)$, from line 5 of the RS procedure (using $y = x_k$), we have that $\hat{\imath} \in \{1, \ldots, m\}$ exists such that

$$f_{\hat{\imath}}(x_k) - f_{\hat{\imath}}(x_k + s^j(x_k)) \geq \eta \frac{\|s^j(x_k)\|^{p+1}}{p!} \sigma_{\hat{\imath}}^j(x_k) \geq \eta \frac{\sigma_l^{\min}}{p!} \|s^j(x_k)\|^{p+1}. \tag{27}$$

Moreover, from Proposition 6 and Remark 2, we can write

$$\begin{aligned}
\|v(x_k + s^j(x_k))\| &\leq \left(\frac{(p+1)\max_{i=1,\ldots,m}(\sigma^j(x_k))_i}{p!} + \frac{L^{\max}}{(p-1)!}\right) \|s^j(x_k)\|^p \\
&= \left(\frac{(p+1)\max_{i=1,\ldots,m}(\sigma(x_k))_i}{p!} + \frac{L^{\max}}{(p-1)!}\right) \|s^j(x_k)\|^p \\
&\leq \left(\frac{(p+1)\sigma^{\max}}{p!} + \frac{L^{\max}}{(p-1)!}\right) \|s^j(x_k)\|^p,
\end{aligned}$$

that is,

$$\|s^j(x_k)\|^{(p+1)} \geq \left(\frac{(p+1)\sigma^{\max}}{p!} + \frac{L^{\max}}{(p-1)!}\right)^{-(p+1)/p} \|v(x_k + s^j(x_k))\|^{(p+1)/p}.$$

Hence, item (i) follows by combining the above inequality with (27), while item (ii) follows from item (i) and the fact that, from Lemma 1, we have

$$HI(F(Y^{j+1}(x_k))) - HI(F(Y^j(x_k))) \geq \left(f_{\hat{\imath}}(x_k) - f_{\hat{\imath}}(x_k + s^j(x_k))\right)^m.$$

□

Remark 4. *For any $k \geq 0$ and any non-Pareto-stationary point $x_k \in X_k$, from Propositions 9–10 the following holds:*

(i) $f_i(x_k) - f_i(x_k + s(x_k)) \geq c\|v(x_k + s(x_k))\|^{(p+1)/p}$ *for all* $i = 1, \ldots, m$,

(ii) $HI(F(X_{k+1})) - HI(F(X_k)) \geq c^m(\|v(x_k + s(x_k))\|)^{m(p+1)/p}$.

In particular, $x_k + s(x_k)$ ensures a decrease for all objective functions since it is the point produced in the last iteration of the RS procedure invoked with $x = x_k$ and $X = X_k$.

We are now ready to derive worst-case complexity bounds for Algorithm HOP in two different scenarios. Namely, we are interested in upper bounding the number of iterations and function evaluations needed to generate a set X_k satisfying one of the following two properties:

- all points in X_k are ϵ-approximate Pareto-stationary, as shown in Theorem 1;
- at least one point $x_k \in X_k$ is such that $x_k + s(x_k)$ is an ϵ-approximate Pareto-stationary point, as shown in Theorem 2.

Theorem 1. *Assume that each s^j in the RS procedure satisfies* (16). *Let $\epsilon > 0$ and, for Algorithm HOP, define*

$$K'_\epsilon := \{k \colon \|v(x_k + s(x_k))\| \geq \epsilon \text{ for at least one } x_k \in X_k\}.$$

Let NF'_ϵ be the number of functions evaluations performed by Algorithm HOP up to the first iteration $\bar{k} \notin K'_\epsilon$. If Assumption 1 holds, then

$$|K'_\epsilon| \leq \left\lfloor \frac{\overline{HI} - HI_0}{c^m} \epsilon^{-m(p+1)/p} \right\rfloor,$$
$$NF'_\epsilon \leq n_F |X(\epsilon)| \left\lfloor \frac{\overline{HI} - HI_0}{c^m} \epsilon^{-m(p+1)/p} \right\rfloor,$$

where $\overline{HI}$, n_F and c are given in (2), (23) *and* (26), *respectively, while $|X(\epsilon)| := \max_{k=0,\ldots,\bar{k}} |X_k|$ and $HI_0 := HI(F(X_0))$.*

Proof. Let us consider an iteration r and write

$$HI(F(X_{r+1})) - HI_0 = \sum_{k=0}^{r}(HI(F(X_{k+1})) - HI(F(X_k))).$$

Taking the limit for $r \to \infty$ and using item (ii) in Remark 4, for any non-Pareto-stationary $x_k \in X_k$ we can write

$$\begin{aligned}
\overline{HI} - HI_0 &\geq \sum_{k=0}^{\infty}(HI(F(X_{k+1})) - HI(F(X_k))) \\
&\geq \sum_{k=0}^{\infty} c^m \|v(x_k + s(x_k))\|^{m(p+1)/p} \\
&\geq \sum_{k\in K'_\epsilon} c^m \|v(x_k + s(x_k))\|^{m(p+1)/p} \\
&\geq \sum_{k\in K'_\epsilon} c^m \epsilon^{m(p+1)/p} \\
&= |K'_\epsilon| c^m \epsilon^{m(p+1)/p},
\end{aligned}$$

which proves the upper bound on $|K'_\epsilon|$.

To upper bound NF'_ϵ, first note that $\bar{k} \leq |K'_\epsilon|$. Furthermore, as pointed out in Remark 3, the maximum number of function evaluations n_F required by the RS procedure at each iteration is given as in (23). Then, the desired result follows by observing that, at any iteration k, the RS procedure is called at most $|X_k|$ times. □

Theorem 2. *Assume that each s^j in the RS procedure satisfies* (16)*. Let $\epsilon > 0$ and, for Algorithm HOP, define*

$$K''_\epsilon := \{k\colon \|v(x_k + s(x_k))\| \geq \epsilon \text{ for all } x_k \in X_k\}.$$

Let NF''_ϵ be the number of functions evaluations performed by Algorithm HOP up to the first iteration $\bar{k} \notin K''_\epsilon$. If Assumption 1 holds, then

$$\begin{aligned}
|K''_\epsilon| &\leq \left\lfloor \left(\frac{\min_{i=1,\dots,m}(f_i(x_0) - f_i^{min})}{c} \right) \epsilon^{-(p+1)/p} \right\rfloor, \\
NF''_\epsilon &\leq n_F |X(\epsilon)| \left\lfloor \left(\frac{\min_{i=1,\dots,m}(f_i(x_0) - f_i^{min})}{c} \right) \epsilon^{-(p+1)/p} \right\rfloor,
\end{aligned}$$

where n_F and c are given in (23) *and* (26)*, respectively, while $X(\epsilon)$ is defined as in Theorem 1.*

Proof. Using item (i) of Remark 4 and the instructions for computing the set X_{k+1} at every iteration k, we can define a sequence $\{x_k\} \subseteq \mathbb{R}^n$ such that, starting from any $x_0 \in X_0$, we have that $F(x_{k+1}) \leq F(x_k + s(x_k))$ and $x_{k+1} \in X_{k+1}$ for all $k \geq 0$. Moreover, still using item (i) of Remark 4, we have

$$f_i(x_k) - f_i(x_{k+1}) \geq f_i(x_k) - f_i(x_k + s(x_k)) \geq c\|v(x_{k+1})\|^{(p+1)/p}, \quad i = 1, \dots, m,$$

where $v(\cdot)$ is defined as in (4). Now, let $\hat{\imath} \in \{1, \ldots, m\}$ be such that

$$f_{\hat{\imath}}(x_0) - f_{\hat{\imath}}^{\min} = \min_{i=1,\ldots,m} (f_i(x_0) - f_i^{\min}).$$

Note that $f_{\hat{\imath}}(x_0) - f_{\hat{\imath}}^{\min} < \infty$ by Assumption 1. Then, for any $k \geq 0$, summing up all the objective decreases until iteration k, we can write

$$f_{\hat{\imath}}(x_0) - f_{\hat{\imath}}^{\min} \geq f_{\hat{\imath}}(x_0) - f_{\hat{\imath}}(x_{k+1}) = \sum_{\ell=0}^{k} (f_{\hat{\imath}}(x_\ell) - f_{\hat{\imath}}(x_{\ell+1})) \geq c \sum_{\ell=0}^{k} \|v(x_{\ell+1})\|^{(p+1)/p}. \quad (28)$$

Letting $k \to \infty$ in (28), we obtain

$$f_{\hat{\imath}}(x_0) - f_{\hat{\imath}}^{\min} \geq c \sum_{\ell=0}^{\infty} \|v(x_{\ell+1})\|^{(p+1)/p} \geq c \sum_{\ell \in K_\epsilon''} \|v(x_k)\|^{(p+1)/p} \geq c |K_\epsilon''| \epsilon^{(p+1)/p},$$

hence proving the upper bound on $|K_\epsilon''|$.

The upper bound on NF_ϵ'' can be obtained by the same reasoning as in the proof of Theorem 1, noting that $\bar{k} \leq |K_\epsilon''|$. □

Remark 5. *Assumption 1 can be weakened in Theorem 2. Namely, Theorem 2 still holds if we assume that $\hat{\jmath} \in \{1, \ldots, m\}$ exists such that $f_{\hat{\jmath}}^{min} > -\infty$.*

From Theorem 1 we see that, to produce a set X_k where all points are ϵ-approximate Pareto-stationary, the maximum number of iterations and function evaluations required by Algorithm HOP are $\mathcal{O}(\epsilon^{-m(p+1)/p})$ and $\mathcal{O}(|X(\epsilon)|\epsilon^{-m(p+1)/p})$, respectively, thus depending on both the number of objective functions m and the model order p. Interestingly, the dependency on m does not appear in the worst-case complexity bounds to generate at least one ϵ-approximate Pareto-stationary point, which are $\mathcal{O}(\epsilon^{-(p+1)/p})$ for the number of iterations and $\mathcal{O}(|X(\epsilon)|\epsilon^{-(p+1)/p})$ for the number of functions evaluations, as stated in Theorem 2. Note the latter result aligns with the worst-case complexity bounds given in [4] for algorithms that generate a single point.

4.2 Using inexact minimizers

In this section, we analyze Algorithm HOP assuming that s is chosen as an approximate minimizer of $m^p(s; x, \sigma)$ for any point x and regularization parameter vector σ. Specifically, we will show that we maintain the same worst-case complexity as in the above subsection (up to constant factors).

Here, given a point $x \in \mathbb{R}^n$ and a regularization parameter vector $\sigma \in \mathbb{R}^m_+$, we just require

s and λ_i, $i = 1, \ldots, m$, to satisfy the following conditions:

$$m^p(s; x, \sigma) \leq m^p(0; x, \sigma) = 0, \tag{29a}$$

$$\sum_{i=1}^{m} \lambda_i \leq \theta, \tag{29b}$$

$$\Big\| \sum_{i=1}^{m} \lambda_i \nabla_s m_i^p(s; x, \sigma_i) \Big\| \leq \tau \|s\|^p, \tag{29c}$$

$$\lambda_i \geq 0, \quad i = 1, \ldots, m, \tag{29d}$$

with given $\theta \in [1, \infty)$ and $\tau \in [0, \infty)$.

We see that (29a) imposes a decrease for $m^p(s; x, \sigma)$, while (29b)–(29d) express approximate KKT conditions (cf. the original KKT conditions given in (11a)–(11e)). In practice, to obtain s satisfying (29a)–(29d), we can perform an inexact minimization of $m^p(s; x, \sigma)$ with respect to s and get λ as an approximate KKT multiplier vector, thus avoiding the issues connected with global optimization.

Then, Propositions 6 and 7 can be reformulated according to the new approximate conditions, leading to the same worst-case complexity bounds (up to constant factors) by the same arguments used in the previous subsection.

Proposition 11. *Given $x \in \mathbb{R}^n$ and $\sigma \in \mathbb{R}^m_+$, let $\bar{s}$ satisfying* (29)*. Then,*

$$\|v(x + \bar{s})\| \leq \left[\tau + \theta \left(\frac{(p+1) \max_{i=1,\ldots,m} \sigma_i}{p!} + \frac{L^{max}}{(p-1)!} \right) \right] \|\bar{s}\|^p,$$

where $v(\cdot)$ is defined as in (4)*.*

Proof. We can reason as in the proof of Proposition 6, with the only difference being in the fact that the last term in (18) must be multiplied by θ, in view of (29b), and added to $\tau \|\bar{s}\|^p$, in view of (29c), while the last term in (19) should be multiplied by θ, in view of (29b). □

Proposition 12. *Given a non-Pareto-stationary point $x \in \mathbb{R}^n$, assume that $\sigma_i \geq (1-\eta)^{-1} L_i$, $i = 1, \ldots, m$, with $\eta \in (0, 1)$, and $\bar{s}$ satisfying* (29)*. Then,*

$$F(x + \bar{s}) \leq F(x) - \eta \frac{\|\bar{s}\|^{p+1}}{p!} \sigma.$$

Proof. It is identical to the proof of Proposition 7 (just observing that (29a) is now a condition on s rather than coming from Proposition 4). □

Remark 6. *Replacing Propositions 6–7 by Propositions 11–12, respectively, has only the effect to generate, instead of the constant c defined in* (26)*, the following new constant:*

$$\tilde{c} := \left(\eta \frac{\sigma_l^{\min}}{p!} \right) \left[\tau + \theta \left(\frac{(p+1)\sigma^{\max}}{p!} + \frac{L^{max}}{(p-1)!} \right) \right]^{-(p+1)/p}. \tag{30}$$

Namely, if each s^j in the RS procedure satisfies (29)*, then the results of Theorems 1–2 hold with c replaced by $\tilde{c}$.*

5 LHOP: a light version of HOP

Each iteration of Algorithm HOP may be computationally expensive as the RS procedure needs to be applied to every point in X_k. In this section, we propose a variant called Light HOP (LHOP) where, at each iteration, the RS procedure is applied to a single point in the set X_k rather than to all of them. Note that, in the proposed version, we do not specify any rule to choose such a point, allowing any user-specified criterion to be applied.

Algorithm 3 Light High-Order algorithm for Pareto-front reconstruction (LHOP)

1: **given** $X_0 \subseteq \mathbb{R}^n$
2: set $k = 0$
3: **while** all $x_k \in X_k$ are non-Pareto-stationary **do**
4: choose $x_k \in X_k$
5: compute $Y(x_k)$, $s(x_k)$, $\sigma(x_k)$ by `Regularized Search`(x_k, X_k)
6: set X_{k+1} as the nondominated points in $X_k \cup Y(x_k)$
7: set $k \leftarrow k + 1$
8: **end while**

The worst-case complexity bounds for Algorithm LHOP are summarized in the next result.

Theorem 3. *Given $\epsilon > 0$, define K''_ϵ and NF''_ϵ as in Theorem 2. If Assumption 1 holds, then*

$$|K''_\epsilon| \leq \left\lfloor \frac{\overline{HI} - HI_0}{\hat{c}^m} \epsilon^{-m(p+1)/p} \right\rfloor,$$
$$NF''_\epsilon \leq n_F \left\lfloor \frac{\overline{HI} - HI_0}{\hat{c}^m} \epsilon^{-m(p+1)/p} \right\rfloor,$$

with

$$\hat{c} = \begin{cases} c & \text{if each } s^j \text{ in the RS procedure satisfies (16)}, \\ \tilde{c} & \text{if each } s^j \text{ in the RS procedure satisfies (29)}, \end{cases}$$

where n_F, c and $\tilde{c}$ are given in (23), (26) and (30), respectively.

Proof. If each s^j in the RS procedure satisfies (16), then the proof is identical to the one of Theorem 1 with K'_ϵ replaced with K''_ϵ since Algorithm LHOP picks one point $x_k \in X_k$ at every iteration k, allowing us to state that

$$HI(F(X_k \cup \{x_k + s(x_k)\})) - HI(F(X_k)) \geq c^m (\|v(x_k + s(x_k))\|)^{m(p+1)/p} \geq c^m \epsilon^{m(p+1)/p}$$

holds for all $k \in K''_\epsilon$ (but note that this is not true for all $k \in K'_\epsilon$). If each s^j in the RS procedure satisfies (29), then the reasoning is the same as before by taking into account Remark 6. □

For Algorithm LHOP, we can only provide worst-case complexity bounds for generating at least one ϵ-approximate Pareto-stationary point. In particular, according to Theorem 3,

Problem name	P.nr.	n	m	ref
T1	1	2	2	[13]
T2	2	2	2	[13]
LOVISON1 (unc)	3	2	2	[13]
LOVISON3 (unc)	4	2	2	[13]
FES2	5	10	3	[13]
IKK1	6	2	3	[13]
ZLT1	7	4	3	[13]
BROWNDEN vs ALLINITU	8	4	2	[15]
BROWNAL vs ARWHEAD	9	10	2	[15]
BROWNAL vs VARDIM	10	10	2	[15]
ARWHEAD vs VARDIM	11	10	2	[15]
ZANGWIL2 vs ROSENBR	12	2	2	[15]
ZANGWIL2 vs CUBE	13	2	2	[15]
ZANGWIL2 vs WAYSEA1	14	2	2	[15]
ROSENBR vs WAYSEA1	15	2	2	[15]
ROSENBR vs CUBE	16	2	2	[15]
WAYSEA1 vs CUBE	17	2	2	[15]

Table 1: Description of the problems used in the numerical experience. Column labels are: 'P.nr.' for problem number; 'n' for the number of variables; 'm' for the number of objective functions; 'ref' for the reference where the problem definition can be found.

the worst-case complexity is $\mathcal{O}(\epsilon^{-m(p+1)/p})$ for both the number of iterations and the number of function evaluations. Hence, compared to the results given for HOP in Theorem 2, the maximum number of iterations for LHOP is worse since it depends on m, while the maximum number of function evaluations, although also dependent on m, does not depend on $|X(\epsilon)|$. However, recall that LHOP performs a less exhaustive analysis of the points in the current set X_k at each iteration k.

6 Numerical examples

In this section, we report a preliminary numerical experience to compare the two algorithms HOP and LHOP and assess their behavior. By virtue of the results obtained, we will also try to put into perspective the different complexity results.

Problems used in the numerical experience We use two classes of multiobjective problems, that is, problems that are inherently multiobjective (from the collections proposed in [13] and [27]) and multiobjective problems built using two functions selected from the CUTEst [18] test set as proposed in [15].

Implementation details Both the algorithms HOP and LHOP have been implemented in Matlab© (version 2026a) and tested on an intel® Xeon® w3-2425 with 12 cores and 12 GB RAM running Linux Ubuntu 24.04 LTS. In the implementation of HOP and LHOP, we made the following choices:

- *Stationarity tolerance.* We use the value $\epsilon = 10^{-5}$;
- *Ordering of X_k.* Two strategies for ordering points in X_k are considered, identified by the parameter `spread_option` $\in \{1, 2\}$:

  `spread_option` $= 1$ means that, at every iteration k, we order X_k on the basis of the *spread* measure as defined in [14]; then, in LHOP, we select the point which is adjacent to the largest gap in the current Pareto front in objective space;

  `spread_option` $= 2$ means that, at every iteration k, we order X_k on the basis of the stationarity measure of the points $x_k \in X_k$; then in LHOP we select the worst point in terms of stationarity measure;

- *Choice of regularized search parameters.* Ideally, $[\sigma_l, \sigma_u]$ should approximate the interval where the reciprocal of the (local) Lipschitz constants lie (see Proposition 8). In the absence of this information, one can draw inspiration from what is commonly done for regularized methods in single-objective optimization, which typically update the regularization parameter by means of rules inherited from trust-region schemes (see, e.g., [3, 6]). In our case, since we consider a set of points X_k at each iteration k, we use a reference value given by the mean $\bar{\sigma}_k$ of the regularization parameters returned by the Regularized Search over all the points in X_k. Then, at iteration $k+1$, we initialize the Regularized Search with $\sigma = \bar{\sigma}_k/2$ for all points in X_{k+1}. Note that this implicitly defines an interval $[\sigma_l, \sigma_u] \subset (0, \infty)^m$ containing the initial value of σ, since we start with a positive value for σ at iteration $k = 0$ (so that σ_l is a positive vector) and, in view of Remark 2, the regularization parameters are upper bounded by finite values (so that σ_u is a finite vector). Moreover, we use $\delta = 0.5$ in the Regularized Search, in order to double the value of each regularization parameter whenever a sufficient decrease is not obtained. Note that since in LHOP we select a single point from X_k at any iteration k, the mean vector $\bar{\sigma}_k$ coincides with $\sigma(x_k)$.

Hypervolume computation To compute the hypervolume, we first need to define the reference vector in objective space. To this end, let us denote with $\mathcal{P}$ the set of multiobjective problems and with $\mathcal{S}$ the set of solvers in the comparison. For every multiobjective problem $p \in \mathcal{P}$ and solver $s \in \mathcal{S}$, let $\mathcal{F}_{p,s}$ be the Pareto front approximation found by solver s for problem p. We define

$$\mathcal{F}_p = \bigcup_{s \in \mathcal{S}} \mathcal{F}_{p,s}.$$

Then, we define the ideal $\tilde{y}^{I,p}$ and the nadir $\tilde{y}^{N,p}$ points for problem p as follows:

$$\tilde{y}^{I,p} = \left(\min_{y \in \mathcal{F}_p} y_1, \min_{y \in \mathcal{F}_p} y_2, \ldots, \min_{y \in \mathcal{F}_p} y_{m_p} \right)^T,$$
$$\tilde{y}^{N,p} = \left(\max_{y \in \mathcal{F}_p} y_1, \max_{y \in \mathcal{F}_p} y_2, \ldots, \max_{y \in \mathcal{F}_p} y_{m_p} \right)^T + \mathbf{1},$$

where m_p denotes the number of objective functions of problem p. According to Definition 5, the hypervolume obtained by a solver $s \in \mathcal{S}$ is $HI(\mathcal{F}_{p,s})$ where the reference vector is selected

as $\rho = \tilde{y}^{N,p}$. Note that the hypervolume could be biased by the possible different scalings of the objective functions for problem p. To avoid privileging one objective function over the others in the computation of the hypervolume (as proposed in [2]), we define the following transformation $T : \mathbb{R}^{m_p} \to \mathbb{R}^{m_p}$ such that $T(y) = z$ with

$$z_i = \begin{cases} (y_i - \tilde{y}_i^{I,p})/(\tilde{y}_i^{N,p} - \tilde{y}_i^{I,p}) & \text{if } \tilde{y}_i^{N,p} - \tilde{y}_i^{I,p} \neq 0 \\ y_i - \tilde{y}_i^{I,p} & \text{otherwise.} \end{cases} \tag{31}$$

Then, given a problem $p \in \mathcal{P}$ and a solver $s \in \mathcal{S}$, the hypervolume $HV_{p,s}$ is computed as

$$HV_{p,s} = HI(T(\mathcal{F}_{p,s})) = \text{Vol}\left(\bigcup_{y \in \mathcal{F}_{p,s}} [T(y), T(\tilde{y}^{N,p})] \right).$$

Results on the test problems The results obtained by running the solvers HOP and LHOP on the 17 problems are reported in Tables 2–3. In particular, for each solver, we consider $p = 1, 2$, and `spread_option` $= 1, 2$.

Moreover, the column headers *it*, *nP*, *nF*, *HV* represent, respectively, number of iterations, number of nondominated points found, number of function evaluations, and hypervolume of the dominated region. In the tables, we highlight in bold the largest hypervolume found by the solvers HOP and LHOP. As it can be observed, with $p = 1$, HOP achieves the best hypervolume on 16 out of the 34 test problems, whereas LHOP is the best-performing solver on only 8 problems. When the polynomial order is increased to $p = 2$, HOP attains the best hypervolume on 20 out of 34 problems, while LHOP does so on only 4.

It is worth emphasizing that, according to the established complexity results, HOP requires a higher number of function evaluations per iteration than LHOP. However, HOP consistently provides better hypervolume values.

Moreover, the order p of the approximating polynomials has a clear impact on the ability of both solvers to identify stationary Pareto fronts. In particular, using $p = 2$ consistently yields smaller stationarity measures than $p = 1$, indicating that high-order approximations are more effective in driving the iterates toward stationarity.

To assess the effect of the polynomial order p on the performance of the proposed methods, we aggregate the hypervolume results in Table 4. As can be seen, the best hypervolume values are most frequently obtained when $p = 1$.

This behavior can be explained by considering the dual task of multiobjective optimization. On the one hand, the goal is to identify Pareto stationary points; on the other hand, it is equally important to generate a well-distributed set of high-quality nondominated solutions that provides a good approximation of the true Pareto front. The latter aspect is effectively captured by the hypervolume indicator. Consequently, when $p = 1$, the methods encourage solution sets with better hypervolume values, even if this comes at the expense of stationarity. In contrast, increasing the polynomial order to $p = 2$ shifts the balance toward satisfying the stationarity conditions more accurately. As will be shown in the following, this improvement in stationarity is accompanied by a less pronounced advantage in terms of hypervolume.

To evaluate the Pareto stationarity of the solutions produced by the algorithms, we collect, for each method, all the nondominated points generated at the final iteration. The empirical cumulative distribution function of the corresponding stationarity measures is then reported in Figure 1. More specifically, given a solver $s \in \mathcal{S}$, the cumulative distribution of stationarity measure of the nondominated points obtained by solver s is

$$d_s(\alpha) = \frac{1}{|\mathcal{F}_s|}|\{y \in \mathcal{F}_s : \ y = F(x), \|v(x)\| \leq \alpha\}|,$$

where $\mathcal{F}_s = \bigcup_{p\in\mathcal{P}} \mathcal{F}_{p,s}$ and $v(\cdot)$ is defined as in (4). As shown in Figure 1, when linear models (i.e., $p = 1$) are employed, both variants of HOP are outperformed by LHOP with `spread_option` $= 1$, while they outperform LHOP with `spread_option` $= 2$. This behavior is consistent with the design of the algorithms. In particular, LHOP with `spread_option` $= 2$ always selects the nondominated point with the largest stationarity measure, thereby explicitly prioritizing the reduction of the worst stationarity value. Although this strategy is intended to mimic the behavior of HOP, the available function evaluation budget is often exhausted before comparable stationarity levels can be achieved. Conversely, LHOP with `spread_option` $= 1$ selects points according to the spread criterion, repeatedly improving the stationarity of the same subset of solutions and consequently producing a more favorable overall distribution of stationarity measures.

The advantage of HOP over LHOP becomes more evident when quadratic models (i.e., $p = 2$) are employed. As illustrated in Figure 1, both variants of HOP consistently outperform both variants of LHOP in terms of stationarity. This behavior is expected since, for $p = 2$, both methods are designed to reproduce the behavior of Newton's method in the multiobjective optimization setting. The superior performance of HOP can be attributed to its more thorough refinement of the current nondominated set at each iteration, which enables a more effective reduction of the stationarity measure.

Summing up, we could say that HOP generally provides better hypervolume values than LHOP, while LHOP has a lower per-iteration cost.

7 Conclusions

In this paper, we have presented an algorithmic framework for Pareto front reconstruction in unconstrained multiobjective optimization, which generates a set of points using high-order regularized models. At every iteration, the proposed scheme uses a search procedure where several trial points are computed starting from those included in the current set and, if some conditions are satisfied, the trial points can be added to the current set. We have analyzed both the cases where the regularized models are solved exactly and inexactly, giving worst-case complexity bounds. Then, a lighter version of the method has also been investigated where, at each iteration, the search procedure is applied only to one point of the current set rather than to all of them. In the paper, we focused on the general nonconvex case, while the analysis for the convex or strongly convex case may be the subject of future research.

As a final remark, we would like to emphasize that our analysis does not rely on linked sequences, which are commonly used when dealing with a-posteriori algorithms for multiobjective problems (see, e.g., [10, 14, 24, 25]). Instead, our analysis uses the fact that, whenever

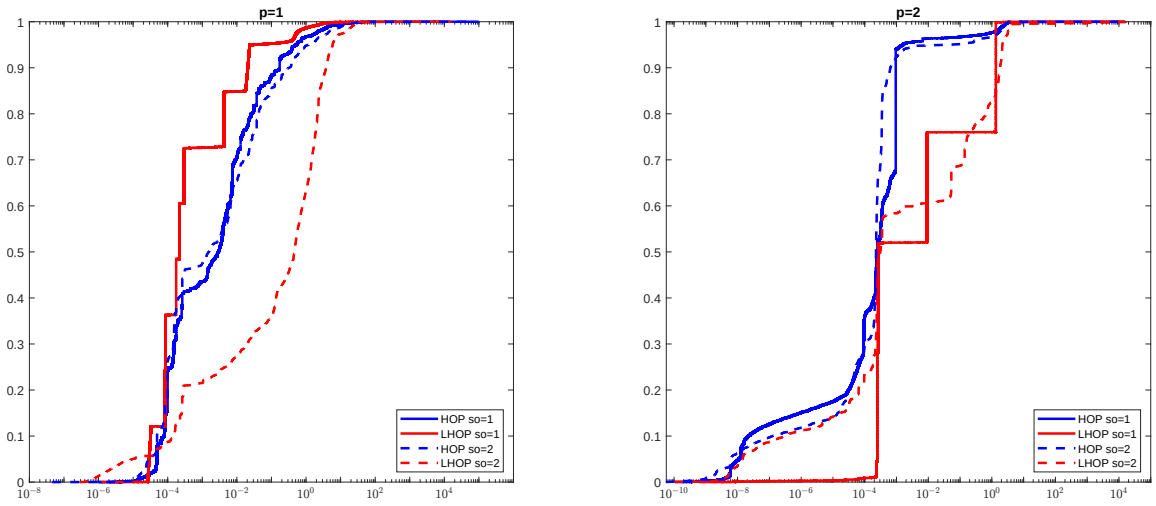

Figure 1: Distribution of stationarity measure of the nondominated points obtained by the different variants of the algorithms HOP and LHOP. Left plot is for $p = 1$, whereas right plot is for $p = 2$.

a point is added to the current set, we can relate the Pareto-stationary violation to the increase in the hypervolume of the set in the image space. This extends a common technique used in the analysis of algorithms for single-objective optimization and algorithms for multi-objective optimization that generate a single Pareto-stationary point, where the stationary violation is typically related to the decrease in the objective function(s).

	HOP				LHOP			
P.nr.	it	nP	nF	HV	it	nP	nF	HV
`spread_option`= 1								
1	17	4767	*	**0.77**	4973	4980	*	0.65
2	6	4297	*	0.80	4978	60	*	**0.82**
3	5	2468	*	**0.72**	4934	25	*	0.71
4	7	769	*	**0.21**	4975	4967	*	0.00
5	6	750	*	**0.71**	4957	65	*	0.32
6	4	9179	*	**0.98**	4951	4998	*	0.97
7	16	6469	*	>0.99	4979	4999	*	>0.99
8	6	252	*	**0.99**	4925	27	*	0.91
9	5	3131	*	0.90	4962	57	*	**0.99**
10	26	2550	*	>0.99	4958	30	*	>0.99
11	5	5444	*	0.83	4993	775	*	>**0.99**
12	8	4189	*	**0.95**	4964	4986	*	0.91
13	7	988	*	**0.95**	4967	4984	*	0.91
14	4	1456	*	**0.82**	4946	92	*	0.81
15	8	5216	*	0.98	4349	1111	*	0.98
16	29	4896	*	>0.99	4959	4882	*	>0.99
17	26	2777	*	>0.99	4747	4245	*	>0.99
`spread_option`= 2								
1	16	1976	*	**0.76**	4878	312	*	0.72
2	6	4713	*	0.80	4980	676	*	**0.82**
3	4	5341	*	0.71	4912	93	*	0.71
4	7	1573	*	**0.12**	4959	60	*	0.10
5	6	719	*	**0.68**	4338	1704	*	0.48
6	4	9192	*	**0.98**	4791	636	*	0.97
7	16	6460	*	>0.99	4979	65	*	>0.99
8	8	147	*	0.99	4404	1201	*	>**0.99**
9	5	3086	*	0.89	4813	426	*	>**0.99**
10	24	2768	*	>0.99	4962	34	*	>0.99
11	5	5450	*	0.82	4867	720	*	>**0.99**
12	8	4325	*	**0.95**	4947	203	*	0.90
13	8	3460	*	**0.94**	4967	87	*	0.90
14	5	1709	*	**0.81**	4946	96	*	0.80
15	8	5187	*	0.98	4195	2008	*	**0.98**
16	18	5502	*	>0.99	4944	59	*	>0.99
17	28	5168	*	0.99	4870	86	*	0.99

Table 2: Comparison of HOP and LHOP when $p = 1$. The '*' symbol in the nF column means the code used all the 10000 allowed function evaluations. For each problem, the largest hypervolumes found by the solvers are highlighted in bold.

	HOP				LHOP			
P.nr.	it	nP	nF	HV	it	nP	nF	HV
`spread_option`= 1								
1	7	1589	*	0.01	4944	686	*	0.01
2	1191	5	*	0.77	4955	8	*	**0.78**
3	4	5363	*	0.71	4869	4904	*	0.71
4	3	4	22	0.12	9	3	19	0.12
5	4	998	*	**0.72**	4907	4929	*	0.50
6	9	264	*	**0.44**	4951	6	*	0.12
7	13	1007	*	**0.73**	4888	4896	*	0.68
8	265	24	*	**0.86**	7228	6	*	0.84
9	17	1344	*	**0.93**	4981	31	*	0.48
10	11	1	24	>0.99	11	1	24	>0.99
11	11	1320	*	**0.95**	4936	4983	*	0.84
12	5	15	61	**0.93**	18	10	40	0.85
13	5	19	71	0.82	25	12	56	**0.83**
14	57	1037	*	**0.82**	7350	15	*	0.80
15	9	5611	*	**0.93**	6	1	26	0.92
16	11	1	33	**0.96**	4957	66	*	0.92
17	12	445	*	**0.77**	4954	45	*	0.01
`spread_option`= 2								
1	7	1539	*	0.01	4551	125	*	0.01
2	1191	5	*	**0.76**	4946	7	*	0.74
3	3	2373	*	0.71	4813	141	*	0.71
4	3	4	22	0.12	8	3	18	0.12
5	4	1043	*	**0.71**	4794	270	*	0.61
6	6	1469	*	0.44	4944	38	*	**0.46**
7	13	1061	*	**0.73**	4959	49	*	0.17
8	80	80	*	**0.87**	8788	6	*	0.85
9	8	403	*	**0.92**	4981	27	*	0.25
10	11	1	24	>0.99	11	1	24	>0.99
11	10	1538	*	**0.95**	4956	39	*	0.86
12	5	14	60	**0.93**	18	10	40	0.85
13	3	17	63	0.76	27	15	61	**0.78**
14	196	138	*	0.80	4995	274	*	0.80
15	10	5721	*	**0.94**	4892	133	*	0.92
16	11	1	33	0.99	13	25	64	0.99
17	13	585	*	**0.86**	4954	46	*	0.01

Table 3: Comparison of HOP and LHOP when $p = 2$. The '*' symbol in the nF column means the code used all the 10000 allowed function evaluations. For each problem, the largest hypervolumes found by the solvers are highlighted in bold.

	$p=1$				$p=2$			
	spread_option=1		spread_option=2		spread_option=1		spread_option=2	
P.nr.	HOP	LHOP	HOP	LHOP	HOP	LHOP	HOP	LHOP
1	**0.80**	0.71	**0.80**	0.77	0.01	0.01	0.01	0.01
2	0.80	**0.82**	0.80	**0.82**	0.76	0.77	0.76	0.74
3	**0.73**	0.72	**0.73**	0.72	0.71	0.71	0.71	0.71
4	**0.21**	<0.01	<0.01	<0.01	<0.01	<0.01	<0.01	<0.01
5	**0.66**	0.30	0.64	0.47	0.55	0.38	0.55	0.47
6	**0.98**	0.97	**0.98**	0.97	0.84	0.73	0.84	0.52
7	0.48	0.48	0.48	0.48	0.68	0.63	**0.71**	0.24
8	>**0.99**	0.92	0.99	>**0.99**	>**0.99**	0.92	>**0.99**	0.92
9	0.89	0.99	0.89	>**0.99**	0.96	0.96	0.96	0.95
10	>**0.99**	>**0.99**	>**0.99**	>**0.99**	>**0.99**	>**0.99**	>**0.99**	>**0.99**
11	0.82	0.99	0.82	>**0.99**	0.99	0.97	0.99	0.98
12	>**0.99**	>**0.99**	>**0.99**	>**0.99**	0.99	0.98	0.99	0.98
13	**0.95**	0.91	0.94	0.90	0.93	0.93	0.93	0.93
14	**0.82**	0.80	**0.82**	0.81	**0.82**	0.81	**0.82**	**0.82**
15	**0.96**	**0.96**	**0.96**	**0.96**	0.94	0.92	**0.96**	0.94
16	>**0.99**	>**0.99**	>**0.99**	>**0.99**	0.41	0.39	0.41	0.41
17	>**0.99**	>**0.99**	>**0.99**	>**0.99**	0.83	0.01	0.86	0.01

Table 4: Comparison of the methods with respect to the order p of the models employed. For each problem, the largest hypervolumes found by the solvers are highlighted in bold.

References

[1] D. P. Bertsekas. *Nonlinear programming*. Athena Scientific, Belmont, MA, 1999.

[2] J. Bigeon, S. Le Digabel, and L. Salomon. DMulti-MADS: Mesh adaptive direct multisearch for bound-constrained blackbox multiobjective optimization. *Computational Optimization and Applications*, 79(2):301–338, 2021.

[3] E. G. Birgin, J. L. Gardenghi, J. M. Martínez, S. A. Santos, and P. L. Toint. Worst-case evaluation complexity for unconstrained nonlinear optimization using high-order regularized models. *Mathematical Programming*, 163:359–368, 2017. doi: 10.1007/s10107-016-1065-8.

[4] L. Calderón, M. A. Diniz-Ehrhardt, and J. M. Martínez. On high-order model regularization for multiobjective optimization. *Optimization Methods and Software*, 37(1): 175–191, 2022. doi: 10.1080/10556788.2020.1719408.

[5] C. Cartis, N. I. Gould, and P. L. Toint. Adaptive cubic regularisation methods for unconstrained optimization. Part II: worst-case function-and derivative-evaluation complexity. *Mathematical programming*, 130(2):295–319, 2011. doi: 10.1007/s10107-009-0337-y.

[6] C. Cartis, N. I. Gould, and P. L. Toint. Adaptive cubic regularisation methods for unconstrained optimization. Part I: motivation, convergence and numerical results. *Mathematical Programming*, 127(2):245–295, 2011. doi: 10.1007/s10107-009-0286-5.

[7] C. Cartis, N. I. M. Gould, and P. L. Toint. Second-order optimality and beyond: characterization and evaluation complexity in convexly constrained nonlinear optimization. *Foundations of Computational Mathematics*, 18:1073–1107, 2018. doi: 10.1007/s10208-017-9363-y.

[8] C. Cartis, N. I. Gould, and P. L. Toint. Universal Regularization Methods: Varying the Power, the Smoothness and the Accuracy. *SIAM Journal on Optimization*, 29(1): 595–615, 2019.

[9] G. Cocchi and M. Lapucci. An augmented Lagrangian algorithm for multi-objective optimization. *Computational Optimization and Applications*, 77(1):29–56, 2020. doi: 10.1007/s10589-020-00204-z.

[10] G. Cocchi, G. Liuzzi, S. Lucidi, and M. Sciandrone. On the convergence of steepest descent methods for multiobjective optimization. *Computational Optimization and Applications*, 77:1–27, 2020. doi: 10.1007/s10589-020-00192-0.

[11] G. Cocchi, M. Lapucci, and P. Mansueto. Pareto front approximation through a multiobjective augmented Lagrangian method. *EURO Journal on Computational Optimization*, 9:100008, 2021. doi: 10.1016/j.ejco.2021.100008.

[12] A. Cristofari, M. De Santis, and S. Lucidi. On Necessary Optimality Conditions for Sets of Points in Multiobjective Optimization. *Journal of Optimization Theory and Applications*, pages 1–20, 2024. doi: 10.1007/s10957-024-02478-3.

[13] A. L. Custódio, J. A. Madeira, A. I. F. Vaz, and L. N. Vicente. Direct multisearch for multiobjective optimization. *SIAM Journal on Optimization*, 21(3):1109–1140, 2011. doi: 10.1137/10079731X.

[14] A. L. Custódio, Y. Diouane, R. Garmanjani, and E. Riccietti. Worst-case complexity bounds of directional direct-search methods for multiobjective optimization. *Journal of Optimization Theory and Applications*, 188(1):73–93, 2021. doi: 10.1007/

s10957-020-01781-z.

[15] M. De Santis, G. Eichfelder, and M. Porcelli. Objective-function free multi-objective optimization: Rate of convergence and performance of an adagrad-like algorithm. *arXiv preprint arXiv:2602.05893*, 2026.

[16] J. Fliege and B. F. Svaiter. Steepest descent methods for multicriteria optimization. *Mathematical Methods of Operations Research*, 51:479–494, 2000. doi: 10.1007/s001860000043.

[17] J. Fliege, A. I. F. Vaz, and L. N. Vicente. Complexity of gradient descent for multiobjective optimization. *Optimization Methods and Software*, 34(5):949–959, 2019. doi: 10.1080/10556788.2018.1510928.

[18] N. I. M. Gould, D. Orban, and Ph. L. Toint. CUTEst: a constrained and unconstrained testing environment with safe threads for mathematical optimization. *Computational Optimization and Applications*, 60(3):545–557, 2015. doi: 10.1007/s10589-014-9687-3.

[19] G. N. Grapiglia, J. Yuan, and Y.-x. Yuan. On the convergence and worst-case complexity of trust-region and regularization methods for unconstrained optimization. *Mathematical Programming*, 152(1):491–520, 2015.

[20] M. Lapucci. Convergence and complexity guarantees for a wide class of descent algorithms in nonconvex multi-objective optimization. *Operations Research Letters*, 54: 107115, 2024. doi: 10.1016/j.orl.2024.107115.

[21] M. Lapucci and P. Mansueto. Improved front steepest descent for multi-objective optimization. *Operations Research Letters*, 51(3):242–247, 2023. doi: 10.1016/j.orl.2023.03.001.

[22] M. Lapucci, P. Mansueto, and D. Pucci. Effective front-descent algorithms with convergence guarantees. *SIAM Journal on Optimization*, 36(2):597–625, 2026.

[23] G. Liuzzi and S. Lucidi. Worst-case complexity analysis of derivative-free methods for multi-objective optimization. *arXiv preprint arXiv:2505.17594*, 2025.

[24] G. Liuzzi, S. Lucidi, and F. Rinaldi. A derivative-free approach to constrained multiobjective nonsmooth optimization. *SIAM J. Optim.*, 26(4):2744–2774, 2016. doi: 10.1137/15M1037810.

[25] A. Mohammadi and A. L. Custódio. A trust-region approach for computing Pareto fronts in multiobjective optimization. *Computational Optimization and Applications*, 87 (1):149–179, 2024. doi: 10.1007/s10589-023-00510-2.

[26] Y. Nesterov and B. T. Polyak. Cubic regularization of Newton method and its global performance. *Mathematical Programming*, 108(1):177–205, 2006. doi: 10.1007/s10107-006-0706-8.

[27] J. Thomann and G. Eichfelder. A trust-region algorithm for heterogeneous multiobjective optimization. *SIAM Journal on Optimization*, 29(2):1017–1047, 2019. doi: 10.1137/18M1173277.