

Worst-Case Complexity Bounds of Directional Direct-Search Methods for Multiobjective Optimization

Ana Luísa Custódio, Youssef Diouane, Rohollah Garmanjani, Elisa Riccietti

▶ **To cite this version:**

Ana Luísa Custódio, Youssef Diouane, Rohollah Garmanjani, Elisa Riccietti. Worst-Case Complexity Bounds of Directional Direct-Search Methods for Multiobjective Optimization. Journal of Optimization Theory and Applications, 2020, pp.0. 10.1007/s10957-020-01781-z . hal-03040932

HAL Id: hal-03040932
https://hal.science/hal-03040932

Submitted on 4 Dec 2020

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Open Archive Toulouse Archive Ouverte (OATAO)

OATAO is an open access repository that collects the work of some Toulouse researchers and makes it freely available over the web where possible.

This is an author's version published in: https://oatao.univ-toulouse.fr/26983

Official URL: https://doi.org/10.1007/s10957-020-01781-z

To cite this version :

Custódio, Ana Luísa and Diouane, Youssef and Garmanjani, Rohollah and Riccietti, Elisa Worst-Case Complexity Bounds of Directional Direct-Search Methods for Multiobjective Optimization. (2020) Journal of Optimization Theory and Applications. ISSN 0022-3239

Any correspondence concerning this service should be sent to the repository administrator: tech-oatao@listes-diff.inp-toulouse.fr

up ates

Worst-Case Complexity Bounds of Directional Direct-Search Methods for Multiobjective Optimization

Ana Luísa Custódio[1] · **Youssef Diouane**[2] · **Rohollah Garmanjani**[1] · **Elisa Riccietti**[3]

Abstract
Direct Multisearch is a well-established class of algorithms, suited for multiobjective derivative-free optimization. In this work, we analyze the worst-case complexity of this class of methods in its most general formulation for unconstrained optimization. Considering nonconvex smooth functions, we show that to drive a given criticality measure below a specific positive threshold, Direct Multisearch takes at most a number of iterations proportional to the square of the inverse of the threshold, raised to the number of components of the objective function. This number is also proportional to the size of the set of linked sequences between the first unsuccessful iteration and the iteration immediately before the one where the criticality condition is satisfied. We then focus on a particular instance of Direct Multisearch, which considers a more strict criterion for accepting new nondominated points. In this case, we can establish a better worst-case complexity bound, simply proportional to the square of the inverse of the threshold, for driving the same criticality measure below the considered threshold.

Keywords Multiobjective unconstrained optimization · Derivative-free optimization methods · Directional direct-search · Worst-case complexity · Nonconvex smooth optimization

Mathematics Subject Classification 90C29 · 90C30 · 90C56 · 90C60

1 Introduction

Multiobjective optimization is a challenging domain in nonlinear optimization [1,2], when there are different conflicting objectives that need to be optimized. Difficulties

Communicated by Xinmin Yang.

✉ Ana Luísa Custódio
alcustodio@fct.unl.pt

Extended author information available on the last page of the article

increase if derivatives are not available, neither can be numerically approximated due to the associated computational cost or to the presence of noise [3]. We are then in the domain of multiobjective derivative-free optimization, which often appears in problems where the objective function is evaluated through numerical simulation (for complementary information on single-objective derivative-free optimization methods, see [4–6]).

We are interested in establishing worst-case complexity (WCC) bounds for directional direct-search, a class of derivative-free optimization methods, when used for solving unconstrained multiobjective optimization problems. Each iteration of this class of algorithms can be divided into a search step and a poll step, being the former optional. In fact, the convergence properties of these methods rely on the procedure implemented in the poll step [7]. The objective function is evaluated at a finite set of points, corresponding to directions with good geometrical properties, scaled by a stepsize parameter. The decision of accepting or rejecting a new evaluated point is solely based on the objective function value, no model is built for the objective function, neither any attempt of estimating derivatives is considered [7]. The criterion for accepting a new evaluated point makes use of the partial order induced by the concept of Pareto dominance (cf. Definition 2.1).

In the last decades, there has been a growing interest in evaluating the performance of optimization algorithms in the worst-case scenario (see, for instance, [8–16]). Usually, the performance of an algorithm is measured by the number of iterations (or function evaluations) required to drive either some criticality measure below a given positive threshold or the function value below the threshold distance to the optimal function value.

In single-objective nonconvex smooth unconstrained optimization, Nesterov [15, Example 1.2.3] derived a WCC bound of $\mathcal{O}\left(\epsilon^{-2}\right)$ for gradient descent algorithms. A similar bound has been achieved for trust-region [17] and line-search [18] methods. Nesterov and Polyak [19] investigated the use of cubic regularization techniques, and then Cartis et al. [20] proposed a generalization to an adaptive regularized framework using cubics. For the latter class of methods, by considering second-order algorithmic variants, this bound was improved to $\mathcal{O}\left(\epsilon^{-3/2}\right)$, including a derivative-free approach where derivatives are approximated by finite-differences [8].

In the context of single-objective derivative-free optimization, directional direct search was the first class of algorithms for which worst-case complexity bounds were established [16]. The author considered the broad class of directional direct-search methods which use sufficient decrease as globalization strategy and established that this class of algorithms shares, in terms of ϵ, the worst-case complexity bound of steepest descent for the unconstrained minimization of a nonconvex smooth function. The complexity of directional direct-search methods for the optimization of convex smooth functions has been addressed in [9]. The bound of $\mathcal{O}\left(\epsilon^{-2}\right)$ has been improved to $\mathcal{O}\left(\epsilon^{-1}\right)$, which is identical, in terms of ϵ, to the one of steepest descent, under convexity. Complexity results have also been established for the nonsmooth case. In [11], a class of smoothing direct-search methods for the unconstrained optimization of nonsmooth functions was proposed and it was shown that the worst-case complexity of this procedure is roughly one order of magnitude worse than the one for directional

direct-search or the steepest descent method, when applied to smooth functions. Other types of direct-search methods have been analyzed in the literature. A probabilistic descent directional direct-search algorithm has been proposed in [13], which is characterized by the fact that poll directions only guarantee descent with a certain fixed probability. The authors establish a worst-case complexity bound of $\mathcal{O}\left(\epsilon^{-2}\right)$, which holds with a high probability. A restricted version of directional direct-search methods, where no stepsize increase is allowed, along with a worst-case complexity analysis has also been studied in [14]. In [21], considering stepsizes independent from the results of each iteration, the authors propose and analyze a random derivative-free optimization algorithm which evaluates three points per iteration and enjoys a worst-case complexity bound of $\mathcal{O}\left(\epsilon^{-2}\right)$.

As for the worst-case complexity of derivative-based methods for solving unconstrained multiobjective optimization problems, it has been shown in [12] that trust-region methods provide a worst-case complexity bound of $\mathcal{O}\left(\epsilon^{-2}\right)$. A similar bound has been derived in [10] and improved to $\mathcal{O}\left(\epsilon^{-1}\right)$ or $\mathcal{O}(\log(\epsilon^{-1}))$, assuming convexity or strong convexity of the different objective function components. In [22], complexity bounds have also been derived for p–order regularization methods, this time under a Hölder continuity assumption on the derivatives of the objective function components.

Regarding the WCC of multiobjective derivative-free optimization algorithms, a first work [23] showed that an optimal worst-case algorithm for Lipschitz functions can be reduced to the computation of centers of balls producing an uniform cover of the feasible region. A biobjective optimization algorithm for single variable, twice continuously differentiable functions was proposed and analyzed in [24]. The authors prove that after $\ell \in \mathbb{N}$ function evaluations, the number of points that do not belong to the Pareto front is of $\mathcal{O}(\log(\ell)^2)$.

In this work, we first establish a worst-case complexity bound for the original Direct Multisearch (DMS) [7] class of methods. We show that the DMS algorithm takes at most $\mathcal{O}\left(|L(\epsilon)|\epsilon^{-2m}\right)$ iterations for driving a criticality measure below $\epsilon > 0$, where $|L(\epsilon)|$ represents the cardinality of the set of linked sequences between the first unsuccessful iteration and the iteration immediately before the one where the criticality condition is satisfied. We then focus on a particular instance of this class of algorithms, which considers a more restrictive condition to accept new nondominated points. For that, we resort to the standard min–max formulation of the multiobjective optimization problem, which is widely used in the literature (e.g., see [25] and [12, Sect. 4.2] for multiobjective trust-region methods or [1] for additional references). We are able to establish that this particular instance of DMS enjoys a worst-case complexity bound of $\mathcal{O}\left(\epsilon^{-2}\right)$ for driving the same criticality measure below $\epsilon > 0$. This bound is identical, in terms of ϵ, to the one derived for multiobjective gradient descent [10] and trust-region [12] methods.

With regard to the strategy used to establish the WCC of the min–max formulation, we highlight that it is not equivalent to a straightforward application of the technique used for single-objective optimization to the scalar function obtained by considering the maximum of the components of the objective function. In particular, the analysis in [16], which establishes the WCC of directional direct-search for single-objective

optimization, relies on the differentiability of the objective function, which does not hold when a min–max formulation is considered. However, the analysis we propose takes into account the differentiability of the single components.

The remaining of the paper is organized as follows. In Sect. 2, we recall some known results on multiobjective optimization, which will be used throughout the paper. The complexity analysis of DMS in its most general form will be established in Sect. 3. Section 4 introduces the min–max formulation and establishes a worst-case complexity bound for it. Some conclusions are drawn in Sect. 5.

2 Preliminaries

Let us consider the unconstrained multiobjective derivative-free optimization problem, defined as

$$\min \quad F(x) := (f_1(x), \ldots, f_m(x))^\top \quad \text{s.t.} \quad x \in \mathbb{R}^n, \tag{1}$$

where $m \geq 2$, and each $f_i : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$, $i \in I := \{1, \ldots, m\}$ is a continuously differentiable function with Lipschitz continuous gradient.

When solving a multiobjective optimization problem of type (1), the goal is to identify a *local Pareto minimizer* [26], i.e., a point $x^* \in \mathbb{R}^n$ such that it does not exist another point x in a neighborhood $\mathcal{N}$ of x^* that dominates x^*, according to Definition 2.1.

Definition 2.1 (*Pareto dominance*) We say that point x dominates point x^* when $F(x) \prec_F F(x^*)$, i.e., when $F(x^*) - F(x) \in \mathbb{R}^m_+ \backslash \{0\}$.

Point x^* is then a local Pareto minimizer, if there is a neighborhood $\mathcal{N}$ of x^* where x^* is nondominated, meaning $F(x) \not\prec_F F(x^*)$ for all $x \in \mathcal{N}$.

A necessary condition for $x^* \in \mathbb{R}^n$ to be a local Pareto minimizer is [26]:

$$\forall d \in \mathbb{R}^n, \ \exists i_d \in I \ : \ \nabla f_{i_d}(x^*)^\top d \geq 0. \tag{2}$$

A point satisfying (2) is called a *Pareto critical point* [26]. We are then interested in finding Pareto critical points. In what comes next, $\|\cdot\|$ will denote the vector or matrix ℓ_2-norm.

Following [26], to characterize Pareto critical points, we are going to use, for a given $x \in \mathbb{R}^n$, the function:

$$\mu(x) := -\min_{\|d\| \leq 1} \max_{i \in I} \nabla f_i(x)^\top d. \tag{3}$$

Fliege and Svaiter [26] showed how some properties of $\mu(x)$, as reported in the following lemma, relate to the concept of Pareto critical points. We denote by $\mathcal{F}(x)$ the solution set of (3).

Lemma 2.1 [26, Lemma 3] *For a given $x \in \mathbb{R}^n$, assume that, for all $i \in I$, the function f_i is continuously differentiable at x and let $\mu(x)$ be defined as in (3). Then,*

1. $\mu(x) \geq 0$;
2. *If x is a Pareto critical point of* (1)*, then* $0 \in \mathcal{F}(x)$ *and* $\mu(x) = 0$;
3. *If x is not a Pareto critical point of* (1)*, then* $\mu(x) > 0$ *and for any* $d \in \mathcal{F}(x)$ *we have*

$$\nabla f_j(x)^\top d \leq \max_{i \in I} \nabla f_i(x)^\top d < 0, \ \forall j \in I,$$

   i.e., d is a descent direction of (1)*;*
4. *The function $x \mapsto \mu(x)$ is continuous;*
5. *If x_k converges to $\bar{x}$, $d_k \in \mathcal{F}(x_k)$ and d_k converges to $\bar{d}$, then $\bar{d} \in \mathcal{F}(\bar{x})$.*

Function μ can then be used to provide information about Pareto criticality of a given point and plays a role similar to the one of the norm of the gradient in single-objective optimization.

The following lemma describes the relationship between function μ and the norm of the gradient of the components of F.

Lemma 2.2 *For a given $x \in \mathbb{R}^n$ and $\epsilon > 0$, assume that, for all $i \in I$, $\nabla f_i(x)$ is well defined. If $\mu(x) > \epsilon$, then $\|\nabla f_i(x)\| > \epsilon$, for all $i \in I$.*

Proof If $\mu(x) > \epsilon$, then

$$\min_{\|d\| \leq 1} \max_{i \in I} \nabla f_i(x)^\top d < -\epsilon.$$

As a consequence, it exists $\bar{d}$ such that $\|\bar{d}\| \leq 1$ and

$$\max_{i \in I} \nabla f_i(x)^\top \bar{d} < -\epsilon,$$

that is, for all $i \in I$, we have

$$-\nabla f_i(x)^\top \bar{d} > \epsilon.$$

Hence, as $-\nabla f_i(x)^\top \bar{d} \leq \|\nabla f_i(x)\| \|\bar{d}\| \leq \|\nabla f_i(x)\|$, the thesis follows. □

In the following, we set $\mu_k := \mu(x_k)$, for all $k \geq 0$.

3 Direct Multisearch

Direct Multisearch (DMS) was proposed in [7] and generalizes directional direct-search to multiobjective derivative-free optimization. It is a general class of methods that can encompass many algorithmic variants, depending, for instance, on the globalization strategy considered. In this work, we will require sufficient decrease for accepting new points, via the use of a forcing function $\rho :]0, +\infty[\longrightarrow]0, +\infty[$. Following [4], ρ is a continuous nondecreasing function, satisfying $\rho(t)/t \to 0$ when $t \downarrow 0$. We consider the typical forcing function $\rho(t) = ct^p$, with $p > 1$, and $c > 0$.

DMS makes use of the strict partial order induced by the cone $\mathbb{R}^m_+$. Let $D(L) \subset \mathbb{R}^m$ be the image of the set of points dominated by a list of evaluated points L and let $D(L; a)$ be the set of points whose distance in the ℓ_∞-norm to $D(L)$ is no larger than $a > 0$. Algorithm 1 corresponds to an instance of the original DMS [7] method, which uses a globalization strategy based on the imposition of a sufficient decrease condition.

Algorithm 1: DMS using sufficient decrease as globalization strategy.

Initialization

Choose $x_0 \in \mathbb{R}^n$ with $f_i(x_0) < +\infty, \forall i \in I$, $\alpha_0 > 0$ an initial stepsize, $0 < \beta_1 \leq \beta_2 < 1$ the coefficients for stepsize contraction and $\gamma \geq 1$ the coefficient for stepsize expansion. Let $\mathcal{D}$ be a set of positive spanning sets. Initialize the list of nondominated points and corresponding stepsize parameters $L_0 = \{(x_0; \alpha_0)\}$.

For $k = 0, 1, 2, \ldots$

1. **Selection of an iterate point:** Order the list L_k according to some criteria and select the first item $(x; \alpha) \in L_k$ as the current iterate and stepsize parameter (thus setting $(x_k; \alpha_k) = (x; \alpha)$).
2. **Search step:** Compute a finite set of points $\{z_s\}_{s \in S}$ and evaluate F at each point in S. Compute L_{trial} by removing all dominated points, using sufficient decrease, from $L_k \cup \{(z_s; \alpha_k) : s \in S\}$ and selecting a subset of the remaining nondominated points. If $L_{trial} \neq L_k$ declare the iteration (and the search step) as successful, set $L_{k+1} = L_{trial}$, and skip the poll step.
3. **Poll step:** Choose a positive spanning set D_k from the set $\mathcal{D}$. Evaluate F at the poll points belonging to $\{x_k + \alpha_k d : d \in D_k\}$. Compute L_{trial} by removing all dominated points, using sufficient decrease, from $L_k \cup \{(x_k + \alpha_k d; \alpha_k) : d \in D_k\}$ and selecting a subset of the remaining nondominated points. If $L_{trial} \neq L_k$ declare the iteration (and the poll step) as successful and set $L_{k+1} = L_{trial}$. Otherwise, declare the iteration (and the poll step) as unsuccessful and set $L_{k+1} = L_k$.
4. **Stepsize parameter update:** If the iteration was successful, then maintain or increase the corresponding stepsize parameter, by considering $\alpha_{k,new} \in [\alpha_k, \gamma\alpha_k]$. Replace all the new points $(x_k + \alpha_k d; \alpha_k)$ in L_{k+1} by $(x_k + \alpha_k d; \alpha_{k,new})$, when success is coming from the poll step, or $(z_s; \alpha_k)$ in L_{k+1} by $(z_s; \alpha_{k,new})$, when success is coming from the search step. Replace also $(x_k; \alpha_k)$, if in L_{k+1}, by $(x_k; \alpha_{k,new})$.
   Otherwise, decrease the stepsize parameter, by choosing $\alpha_{k,new} \in [\beta_1\alpha_k, \beta_2\alpha_k]$, and replace the poll pair $(x_k; \alpha_k)$ in L_{k+1} by $(x_k; \alpha_{k,new})$.

DMS declares an iteration as successful when there are modifications in the list of nondominated points, meaning that a new point x was accepted, such that $F(x) \notin D(L; \rho(\alpha))$, where α represents a stepsize parameter associated with the current iteration.

For analyzing the worst-case behavior of the algorithms presented in this paper, we will need the following assumptions with regard to the component functions in (1).

Assumption 3.1 For all $i \in I$, the function f_i is continuously differentiable with Lipschitz continuous gradient with constant L_i. Set $L_{\max} = \max_{i \in I} L_i$.

Assumption 3.2 For all $i \in I$, the function f_i is lower and upper bounded in $\{x \in \mathbb{R}^n : F(x) \notin D(\{x_0\})\}$, with lower bound $f_i^{\min}$ and upper bound $f_i^{\max}$. Let $F^{\min} := \min\{f_1^{\min}, \ldots, f_m^{\min}\}$ and $F^{\max} := \max\{f_1^{\max}, \ldots, f_m^{\max}\}$.

Assumption 3.3 The set $\{x \in \mathbb{R}^n : F(x) \notin D(\{x_0\})\}$ is compact.

At an unsuccessful iteration of Algorithm 1, none of the components of the objective function is improved, since no new point is added to the list. However, the use of Pareto dominance to accept new points implies that successful iterations do not necessarily correspond to points that improve all components of the objective function. In fact, at some successful iterations, some of these components could increase the corresponding value. Nevertheless, at every successful iteration, the hypervolume (see Definition 3.1 or [27]) corresponding to the current list of nondominated points always increases.

Definition 3.1 [3, Definition 5.2] The hypervolume indicator (or S–metric, from 'Size of space covered') for some (approximation) set $A \subset \mathbb{R}^m$ and a reference point $r \in \mathbb{R}^m$ that is dominated by all the points in A is defined as:

$$\mathrm{HI}(A) \;:=\; \mathrm{Vol}\{b \in \mathbb{R}^m : b \leq r \wedge \exists a \in A : a \leq b\} \;=\; \mathrm{Vol}\left(\bigcup_{a \in A} [a, r]\right).$$

The inequalities should be understood componentwise, $\mathrm{Vol}(\cdot)$ denotes the Lebesgue measure of a m–dimensional set of points, and $[a, r]$ denotes the interval box with lower corner a and upper corner r.

Define $F(L)$ as the image set of a list of points L, i.e., $F(L) := \{F(x) : (x, \alpha) \in L\}$. We will consider $r = (f_1^{\max}, \ldots, f_m^{\max})$, when computing a hypervolume. Lemma 3.1 quantifies the increase in the hypervolume, associated with successful iterations.

Lemma 3.1 *In Algorithm 1, for a successful iteration $k \geq 0$, we have*

$$\mathrm{HI}(F(L_{k+1})) - \mathrm{HI}(F(L_k)) \;\geq\; (\rho(\alpha_k))^m .$$

Proof If k is a successful iteration, then $L_{k+1} \neq L_k$. Let $x \in L_{k+1}$ be such that $x \notin L_k$. In this situation, $F(x) \notin D(L_k; \rho(\alpha_k))$.

Thus, $B_\infty(F(x), \rho(\alpha_k)) \cap D(L_k) = \emptyset$, where $B_\infty(F(x), \rho(\alpha_k))$ represents the ℓ_∞ ball centered at $F(x)$, with radius $\rho(\alpha_k)$. This means that at least a hypercube of volume $\rho(\alpha_k)^m$ was added to the dominated region (the one belonging to $F(x) + (\mathbb{R}_0^+)^m$). □

Figure 1 illustrates the situation, where for a biobjective problem, at a successful iteration, the previous condition is satisfied as an equality. The initial list of nondominated points is formed by the two points represented by the dots. The point corresponding to the star, in the interior of the shaded region, was accepted as a new nondominated point, since it satisfies the sufficient decrease condition. Thus, the hypervolume corresponding to the new set of nondominated points has increased exactly in $\rho(\alpha_k)^2$.

As it is done in classical directional direct-search [4], we assume that all positive spanning sets considered by the algorithm include bounded directions. In multiobjective optimization, the cone of descent directions for all components of the objective function can be as narrow as one would like (see Remark 3.1). So, we need to assume density of the directions at a given limit point, as it is considered in the convergence analysis of DMS [7].

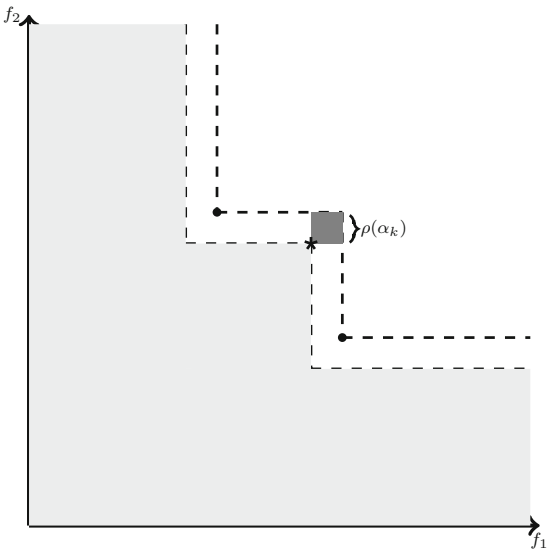


Fig. 1 Hypervolume increase at a successful iteration of Algorithm 1

Definition 3.2 A subsequence of iterates $\{x_k\}_{k\in K}$, corresponding to unsuccessful poll steps, is said to be a refining subsequence if $\{\alpha_k\}_{k\in K}$ converges to zero.

The existence of at least one convergent refining subsequence is a direct consequence of Assumption 3.3 and the use of sufficient decrease for accepting new nondominated points. Refining directions are limits of normalized poll directions associated with the refining subsequence. Without loss of generality, we will assume that all the positive spanning sets considered have normalized directions.

Assumption 3.4 Consider Algorithm 1 and let x^* be the limit point of a convergent refining subsequence. Assume that the set of refining directions associated with x^* is dense in the unit sphere.

We will make use of the following result, which establishes a relationship between the stepsize parameter at an unsuccessful iteration of a directional direct-search method and $\mu_{D_k}(x_k)$, an approximation to $\mu_k = \mu(x_k)$ which only considers the poll directions.

Lemma 3.2 *Let Assumption 3.1 hold. Let k be an unsuccessful iteration of Algorithm 1, D_k be the positive spanning set considered, and $\alpha_k > 0$ be the corresponding stepsize. Define*

$$\mu_{D_k}(x) := -\min_{d\in D_k, \|d\|\leq 1} \max_{i\in I} \nabla f_i(x)^\top d. \tag{4}$$

Then,

$$\mu_{D_k}(x_k) \leq \left(\frac{L_{\max}}{2}\alpha_k + \frac{\rho(\alpha_k)}{\alpha_k}\right). \tag{5}$$

Proof If iteration k is unsuccessful, then for each direction $d_k \in D_k$ there is an index $i(d_k) \in I$ such that

$$f_{i(d_k)}(x_k + \alpha_k d_k) \geq f_{i(d_k)}(x_k) - \rho(\alpha_k).$$

Hence, for each direction $d_k \in D_k$,

$$\begin{aligned} 0 &\leq f_{i(d_k)}(x_k + \alpha_k d_k) - f_{i(d_k)}(x_k) + \rho(\alpha_k) \\ &= \int_0^1 \nabla f_{i(d_k)}(x_k + t\alpha_k d_k)^\top \alpha_k d_k \, dt + \rho(\alpha_k) \end{aligned}$$

Adding $-\alpha_k \nabla f_{i(d_k)}(x_k)^\top d_k$ to both sides yields:

$$\begin{aligned} &-\alpha_k \nabla f_{i(d_k)}(x_k)^\top d_k \\ &\leq \int_0^1 \left(\nabla f_{i(d_k)}(x_k + t\alpha_k d_k)^\top \alpha_k d_k - \alpha_k \nabla f_{i(d_k)}(x_k)^\top d_k \right) dt + \rho(\alpha_k) \\ &\leq \alpha_k^2 \frac{L_{\max}}{2} \|d_k\|^2 + \rho(\alpha_k), \end{aligned}$$

so that

$$-\nabla f_{i(d_k)}(x_k)^\top d_k \leq \alpha_k \frac{L_{\max}}{2} \|d_k\|^2 + \frac{\rho(\alpha_k)}{\alpha_k}.$$

Then,

$$\max_{i \in I} \nabla f_i(x_k)^\top d_k \geq -\alpha_k \frac{L_{\max}}{2} \|d_k\|^2 - \frac{\rho(\alpha_k)}{\alpha_k}$$

This is true for all $d_k \in D_k$ so the thesis holds. □

Hereafter, we set $\mu_{D_k} := \mu_{D_k}(x_k)$. In the current work, WCC bounds will be derived for driving μ_{D_k} below $\epsilon > 0$. However, the goal is to establish bounds for having $\mu_k \leq \epsilon$. For this purpose, we consider Assumption 3.5. A somehow similar assumption has already been used within the context of trust-region derivative-free methods for multiobjective optimization (see [28, Assumption 4.8]).

Assumption 3.5 There exists $\mathcal{C}_1 > 0$ such that

$$|\mu_{D_k} - \mu_k| \leq \mathcal{C}_1 \mu_{D_k}, \quad \forall k \geq 0. \tag{6}$$

Remark 3.1 We note that Assumption 3.5 requires the nonnegativity of μ_{D_k} at every iteration, which may not hold. In such cases, additional directions could be added to the positive spanning sets considered as poll directions. Such procedure is supported by Assumption 3.4.

However, there are cases where Assumption 3.5 can be easily satisfied. Let us consider the following biobjective function:

$$F(x) := \frac{1}{2}\left(\|x - c_1\|^2, \|x - c_2\|^2\right)^\top,$$

where $c_1 = (-1, 1)^\top$ and $c_2 = -c_1$. This is a biobjective version of a single-objective variant of the Dennis–Woods function [29] introduced in [4] (see also [5]). The single-objective function has been used to show that coordinate search (which considers, at every iteration, the positive spanning set $D_k = [I - I]$, where I represents the identity matrix) stalls at any point $(a, a)^\top$, where $a \neq 0$ (see [4,5]). We will show that, for this biobjective problem, when considering the coordinate directions as positive spanning set, Assumption 3.5 holds for the majority of points in $\mathbb{R}^2$.

When applying DMS, for simplicity, we drop the iteration index k and assume that the set of poll directions is $D = [I - I]$. Then, for any $x \in \mathbb{R}^2$, one has $\mu_D(x) = -\min\{x_1 + 1, x_2 + 1, -x_1 + 1, -x_2 + 1\}$. Thus, as long as x belongs to $\mathcal{B} = \{x \in \mathbb{R}^2 : |x_1| > 1 \vee |x_2| > 1\}$, $\mu_D(x)$ will be positive. On the other hand,

$$\begin{aligned}\mu(x) &= \max_{\|d\|=1} \min\{-\nabla f_1(x)^\top d, -\nabla f_2(x)^\top d\} \\ &\leq \min\left\{\max_{\|d\|=1} -\nabla f_1(x)^\top d, \max_{\|d\|=1} -\nabla f_2(x)^\top d\right\}\end{aligned}$$

We will show that the assumption holds for all the points in $\mathcal{B}$ such that $x_1 > 1$ and $x_2 > 1$. For the other points in $\mathcal{B}$ a similar reasoning can be applied. If $x_1 > 1$ and $x_2 > 1$, one gets that $\mu_D(x) = \max\{x_1 - 1, x_2 - 1\}$ and

$$\mu(x) \leq \min\left\{\mu^1(x), \mu^2(x)\right\}, \quad \text{where}$$

$$\mu^1(x) := (x_1 - 1)\sqrt{\frac{(x_1 - 1)^2}{(x_1 - 1)^2 + (x_2 + 1)^2}} + (x_2 + 1)\sqrt{\frac{(x_2 + 1)^2}{(x_1 - 1)^2 + (x_2 + 1)^2}},$$

$$\mu^2(x) := (x_1 + 1)\sqrt{\frac{(x_1 + 1)^2}{(x_1 + 1)^2 + (x_2 - 1)^2}} + (x_2 - 1)\sqrt{\frac{(x_2 - 1)^2}{(x_1 + 1)^2 + (x_2 - 1)^2}}.$$

Then, it holds

$$\begin{aligned}\frac{\mu^1(x)}{\mu_D(x)} &\leq \frac{(x_1 - 1) + (x_2 + 1)}{x_1 - 1} \leq \frac{x_1 + x_2}{x_1 - 1}, \\ \frac{\mu^2(x)}{\mu_D(x)} &\leq \frac{(x_1 + 1) + (x_2 - 1)}{x_2 - 1} \leq \frac{x_1 + x_2}{x_2 - 1}.\end{aligned}$$

Hence, in this case, inequality (6) holds with $\mathcal{C}_1 \geq \max\left\{\frac{x_1+x_2}{x_1-1}, \frac{x_1+x_2}{x_2-1}\right\} - 1$. If x is far from the border of $\mathcal{B}$, this constant will assume reasonable values. For example, if we assume $2 \leq x_1, x_2 \leq 5$, then $\mathcal{C}_1 = 6$.

Difficulties arise when $x \notin \mathcal{B}$ (in which case $\mu_D \leq 0$) or when x is close to the border of $\mathcal{B}$ (which makes the constant $\mathcal{C}_1$ large). In particular, if $x = (a, a)^\top$ and $a \to 0$, the cone of descent directions will become as narrow as one would like. In such cases, it is advisable to rotate the set of polling directions. For example, to make sure that Assumption 3.5 holds at any point $x = (a, a)^\top$ with $|a| < 1$, a possibility would be to choose, for $l \geq 1$, 2^{l-1} maximal positive basis $\{R_i D\}_{0 \leq i \leq 2^{l-1}-1}$, where

$$R_i = \begin{bmatrix} \cos(i\theta) & -\sin(i\theta) \\ \sin(i\theta) & \cos(i\theta) \end{bmatrix},$$

with $\theta = \frac{\pi}{2^l}$ the angle between the generators of the cone of descent directions. Therefore, when $a \to 0$ (or equivalently $\theta \to 0$), Assumption 3.5 is satisfied at the cost of increasing the number of function evaluations.

In the following theorem, we will derive a bound on the number of successful iterations required to drive μ_k below a given small positive threshold. For each pair of indexes $k_1 < k_2$, we will denote by $U_{k_1}(k_2)$ and $S_{k_1}(k_2)$ the set of unsuccessful and successful iterations from k_1 to k_2, respectively. We will also denote by k_0 the index of the first unsuccessful iteration. We remark that the existence of such index is ensured by Assumption 3.2, from which one can prove that Algorithm 1 generates a sequence of iterates satisfying $\liminf_{k \to +\infty} \alpha_k = 0$ [7].

Theorem 3.1 *Consider the application of Algorithm 1 to problem (1), with the choice of forcing function $\rho(t) = ct^p$, $p > 1$, $c > 0$. Let Assumptions 3.1, 3.2, and 3.5 hold. Let k_0 be the index of the first unsuccessful iteration. Given any $\epsilon \in]0, 1[$, assume that $\mu_{k_0} > \epsilon$ and let j_1 be the first iteration after k_0 such that $\mu_{j_1+1} \leq \epsilon$. Then, to achieve $\mu_{j_1+1} \leq \epsilon$ starting from k_0, Algorithm 1 takes at most $|S_{k_0}(j_1)| = \mathcal{O}\left(\epsilon^{-\frac{pm}{\min(p-1,1)}}\right)$ successful iterations.*

Proof Let us assume that $\mu_k > \epsilon$, for $k = k_0, \ldots, j_1$. Using Assumption 3.5 we have

$$\epsilon < \mu_k = |\mu_k - \mu_{D_k}| + \mu_{D_k} \leq (\mathcal{C}_1 + 1)\mu_{D_k}. \tag{7}$$

Hence, we obtain $\mu_{D_k} > \epsilon/(1 + \mathcal{C}_1)$.

In view of Lemma 3.2, for an unsuccessful iteration k, we have

$$\mu_{D_k} \leq \left(\frac{L_{\max}}{2}\alpha_k + \frac{\rho(\alpha_k)}{\alpha_k}\right).$$

Thus,

$$\frac{\epsilon}{1 + \mathcal{C}_1} < \left(\frac{L_{\max}}{2}\alpha_k + \frac{\rho(\alpha_k)}{\alpha_k}\right),$$

which then implies, when $\alpha_k < 1$,

$$\epsilon < \mathcal{L}_1 \alpha_k^{\min(p-1,1)},$$

where $\mathcal{L}_1 = (1+\mathcal{C}_1)\left(\frac{L_{\max}}{2}+c\right)$. If $\alpha_k \geq 1$, then $\alpha_k > \epsilon$. Hence, by combining the two cases ($\alpha_k \geq 1$ and $\alpha_k < 1$) and having $\epsilon < 1$, when k is an unsuccessful iteration, we have

$$\alpha_k \;>\; \mathcal{L}_2 \epsilon^{\frac{1}{\min(p-1,1)}}, \tag{8}$$

where $\mathcal{L}_2 = \min\left(1, \mathcal{L}_1^{-\frac{1}{\min(p-1,1)}}\right)$.

Let k be a successful iteration and $U_{k_0}(k) = \{k_0, k_1, \ldots, k_u\}$ with $k_u < k$ be the set of unsuccessful iterations from k_0 to k. From Lemma 3.1 and by the choice of forcing function,

$$\begin{aligned} \mathrm{HI}(F(L_{k+1})) - \mathrm{HI}(F(L_{k_u})) \;&\geq\; (k-k_u)(c(\min_{k_u+1\leq t\leq k}\alpha_t)^p)^m \\ &\geq\; |S_{k_u}(k)|(c\beta_1^p\alpha_{k_\ell}^p)^m, \quad \text{for some} \quad 0\leq \ell \leq u. \end{aligned}$$

Notice that the second inequality holds as it is possible to backtrack from any iteration $t \in \{k_u+1, \ldots, k\}$ to some previous unsuccessful $k_\ell \in U_{k_0}(k)$ iteration and have $\alpha_t \geq \beta_1\alpha_{k_\ell}$.

Thus, in view of (8),

$$\mathrm{HI}(F(L_{k+1})) - \mathrm{HI}(F(L_{k_u})) \;\geq\; |S_{k_u}(k)|\left(c\beta_1^p\mathcal{L}_2^p\epsilon^{\frac{p}{\min(p-1,1)}}\right)^m.$$

By a similar reasoning, for $1 \leq i \leq u$, we obtain

$$\mathrm{HI}(F(L_{k_i})) - \mathrm{HI}(F(L_{k_{i-1}})) \;\geq\; |S_{k_{i-1}}(k_i)|\left(c\beta_1^p\mathcal{L}_2^p\epsilon^{\frac{p}{\min(p-1,1)}}\right)^m.$$

Therefore, using the two inequalities above for $k = k_0, \ldots, j_1$, we obtain

$$\mathrm{HI}(F(L_{j_1+1})) - \mathrm{HI}(F(L_{k_0})) \;\geq\; |S_{k_0}(j_1)|\left(c\beta_1^p\mathcal{L}_2^p\epsilon^{\frac{p}{\min(p-1,1)}}\right)^m.$$

Since $\left(F^{\max}-F^{\min}\right)^m \geq \mathrm{HI}(F(L_{j_1+1})) - \mathrm{HI}(F(L_{k_0}))$, the proof is completed. □

Now, in order to obtain a bound on the total number of iterations for driving μ_k below a given threshold, it remains to find a bound on the number of unsuccessful iterations. For that, we will adapt the definition of linked sequences, introduced in [30].

Definition 3.3 Consider $\{L_k\}_{k\in\mathbb{N}}$ the sequence of sets of nondominated points generated by Algorithm 1. A linked sequence between iterations i and j ($i < j$) is a finite sequence $\{(x_{l_k}, \alpha_{l_k})\}_{k\in\{1,\ldots,n_l\}}$ of maximum length such that $n_l \leq j-i+1$, $(x_{l_k}, \alpha_{l_k}) \in \cup_{r=i}^{j} L_r$ for all $k \in \{1, \ldots, n_l\}$, and for any $k \in \{2, \ldots, n_l\}$ and $r \in \{i+1, \ldots, j\}$, the pair $(x_{l_k}, \alpha_{l_k}) \in L_r$ is generated by $(x_{l_{k-1}}, \alpha_{l_{k-1}}) \in L_p$, with $i \leq p < r$.

Theorem 3.2 establishes a bound on the number of unsuccessful iterations required for driving μ_k below a given threshold.

Theorem 3.2 *Let all the assumptions of Theorem 3.1 hold. Then, to achieve $\mu_k \leq \epsilon$ starting from k_0, Algorithm 1 takes at most*

$$|U_{k_0}(j_1)| \;\leq\; |\mathbb{L}_{k_0}^{j_1}| \left\lceil -\frac{\log(\gamma)}{\log(\beta_2)}|S_{k_0}(j_1)| - \frac{\log(\alpha_{l_1})}{\log(\beta_2)} + \frac{\log\left(\beta_1 \mathcal{L}_2 \epsilon^{\frac{1}{\min(p-1,1)}}\right)}{\log(\beta_2)} \right\rceil$$

unsuccessful iterations, where α_{l_1} denotes the stepsize associated to one of the points in L_{k_0} and $|\mathbb{L}_{k_0}^{j_1}|$ the cardinality of the set of all linked sequences between k_0 and j_1.

Proof Let $\{(x_{l_k}, \alpha_{l_k})\}_{k\in\{1,\ldots,n_l\}}$ be a linked sequence between k_0 and j_1. Let $S_{k_0}(n_l)$ and $U_{k_0}(n_l)$ be, respectively, the set of successful and unsuccessful iterations in the sequence. Assume $U_{k_0}(n_l) \neq \emptyset$. Since, for any $1 \leq k < n_l - 1$, either $\alpha_{l_{k+1}} \leq \beta_2 \alpha_{l_k}$ (if the iteration is unsuccessful) or $\alpha_{l_{k+1}} \leq \gamma \alpha_{l_k}$ (if the iteration is successful), we obtain by induction

$$\alpha_{l_{n_l}} \;\leq\; \alpha_{l_1} \gamma^{|S_{k_0}(n_l)|} \beta_2^{|U_{k_0}(n_l)|},$$

which, in turn, implies from $\log(\beta_2) < 0$

$$|U_{k_0}(n_l)| \;\leq\; -\frac{\log(\gamma)}{\log(\beta_2)}|S_{k_0}(n_l)| - \frac{\log(\alpha_{l_1})}{\log(\beta_2)} + \frac{\log(\alpha_{l_{n_l}})}{\log(\beta_2)}.$$

From $\log(\beta_2) < 0$ and the lower bound (8) on α_k, we obtain

$$|U_{k_0}(n_l)| \;\leq\; -\frac{\log(\gamma)}{\log(\beta_2)}|S_{k_0}(n_l)| - \frac{\log(\alpha_{l_1})}{\log(\beta_2)} + \frac{\log\left(\beta_1 \mathcal{L}_2 \epsilon^{\frac{1}{\min(p-1,1)}}\right)}{\log(\beta_2)}.$$

The last inequality holds trivially when $U_{k_0}(n_l) = \emptyset$.

Let $\mathbb{L}_{k_0}^{j_1}$ denote the set of indexes of all linked sequences between k_0 and j_1. Then,

$$\begin{aligned}
|U_{k_0}(j_1)| &\leq \sum_{l\in\mathbb{L}_{k_0}^{j_1}} |U_{k_0}(n_l)| \\
&\leq\; |\mathbb{L}_{k_0}^{j_1}| \max_{l\in\mathbb{L}_{k_0}^{j_1}} |U_{k_0}(n_l)| \\
&\leq\; |\mathbb{L}_{k_0}^{j_1}| \max_{l\in\mathbb{L}_{k_0}^{j_1}} \left\lceil -\frac{\log(\gamma)}{\log(\beta_2)}|S_{k_0}(n_l)| - \frac{\log(\alpha_{l_1})}{\log(\beta_2)} + \frac{\log\left(\beta_1 \mathcal{L}_2 \epsilon^{\frac{1}{\min(p-1,1)}}\right)}{\log(\beta_2)} \right\rceil.
\end{aligned}$$

Since $n_l \leq j_1 - k_0 + 1$, one has $|S_{k_0}(n_l)| \leq |S_{k_0}(j_1)|$ for all $l \in \mathbb{L}_{k_0}^{j_1}$. Thus,

$$|U_{k_0}(j_1)| \leq |\mathbb{L}_{k_0}^{j_1}| \left[-\frac{\log(\gamma)}{\log(\beta_2)} |S_{k_0}(j_1)| - \frac{\log(\alpha_{l_1})}{\log(\beta_2)} + \frac{\log\left(\beta_1 \mathcal{L}_2 \epsilon^{\frac{1}{\min(p-1,1)}}\right)}{\log(\beta_2)} \right].$$

□

In the previous bound, the size of the second term in the sum can be easily bounded. In fact, from Theorem 3.1 we know that there is a finite number of successful iterations, before driving μ below the given threshold. The increase in the stepsize can be controlled by setting $\gamma = 1$ or by considering an upper bound for the stepsize itself (see [9]).

Combining Theorems 3.1 and 3.2, it can be seen that Algorithm 1 takes at most $\mathcal{O}\left(|L(\epsilon)|\epsilon^{-\frac{pm}{\min(p-1,1)}}\right)$ iterations to bring $\mu_k < \epsilon$ for some $k \geq 0$, where $|L(\epsilon)|$ represents the size of the set of linked sequences between the first unsuccessful iteration and the iteration immediately before the one where the criticality condition is satisfied. The best complexity bound is then derived by setting $p = 2$, which leads to the complexity bound of $\mathcal{O}\left(|L(\epsilon)|\epsilon^{-2m}\right)$. The WCC bounds of Algorithm 1 in terms of the number of function evaluations are established in the following corollary.

Corollary 3.1 *Let all the assumptions of Theorem 3.1 hold. To achieve $\mu_k < \epsilon$, Algorithm 1 takes at most $\mathcal{O}\left(|L(\epsilon)|\epsilon^{-\frac{pm}{\min(p-1,1)}}\right)$ iterations (and $\mathcal{O}\left(n|L(\epsilon)|\epsilon^{-\frac{pm}{\min(p-1,1)}}\right)$ function evaluations). When $p = 2$ this bound is $\mathcal{O}\left(|L(\epsilon)|\epsilon^{-2m}\right)$ (and $\mathcal{O}\left(n|L(\epsilon)|\epsilon^{-2m}\right)$ function evaluations).*

Remark 3.2 With regard to the number of function evaluations in Corollary 3.1, since the computational cost of evaluating each component of the objective function might not be the same, we have considered the computational cost of F and count the number of times that it is evaluated at each iteration, rather than counting separately the number of component functions evaluations.

One can see that the bound for DMS, in terms of ϵ, does not conform with the bound $\mathcal{O}\left(\epsilon^{-2}\right)$ for the gradient descent derived in [10] for problem (1). One of the main reasons behind this difference is the fact that DMS declares an iteration as successful if at least one of the components of the objective function could be improved sufficiently, whereas in [10] the algorithm, which uses a backtracking approach for determining the right stepsize parameter, moves to a new point if all the components of the objective function could be improved sufficiently. It should also be noted that DMS will compute an approximation to a complete local Pareto front, whereas the multiobjective gradient descent algorithm finds a single Pareto critical point. This explains the dependence on $|L(\epsilon)|$ for the WCC bounds derived.

If a more demanding criterion is considered to accept new nondominated points, a complexity bound identical, in terms of ϵ, to the one derived in [10] for the gradient descent can be established for DMS. In the next section, we will propose a direct-search framework, which corresponds to a particular instance of DMS, and presents a

worst-case complexity bound of $\mathcal{O}\left(n\epsilon^{-2}\right)$, when considering the number of function evaluations.

4 A Min–Max Direct-Search Framework for Multiobjective Optimization

In this section, instead of considering problem (1) directly, we use a min–max formulation:

$$\min_{x\in\mathbb{R}^n} f(x)$$

with

$$f(x) := \max_{i\in I} f_i(x).$$

Algorithm 2 considers a direct-search (DS) approach with a stricter criterion for accepting new nondominated points. In this case, rather than an approximation to the complete Pareto front, only one Pareto critical point will be computed for problem (1). For simplicity, the forcing function $\rho(t) = \frac{c}{2}t^2$, with $c > 0$, has been considered and the (optional) search step has not been included in the algorithmic description. However, the subsequent results could be established for a more general setting, such as the one of Algorithm 1, once that the strict condition for accepting new nondominated points is used.

Algorithm 2: Min–max DS for multiobjective optimization.

Initialization

Choose $x_0 \in \mathbb{R}^n$ with $f_i(x_0) < +\infty, \forall i \in I$, $\alpha_0 > 0$ an initial stepsize, $0 < \beta_1 \leq \beta_2 < 1$ the coefficients for stepsize contraction and $\gamma \geq 1$ the coefficient for stepsize expansion. Let $\mathcal{D}$ be a set of positive spanning sets and $c > 0$ a constant used in the sufficient decrease condition.

For $k = 0, 1, 2, \ldots$

1. **Poll step:** Choose a positive spanning set D_k from the set $\mathcal{D}$. Evaluate F at the poll points belonging to $\{x_k + \alpha_k d : d \in D_k\}$. If it exists $d_k \in D_k$ such that

$$f(x_k + \alpha_k d_k) < f(x_k) - \frac{c}{2}\alpha_k^2,$$

then declare the iteration as successful and set $x_{k+1} = x_k + \alpha_k d_k$. Otherwise, declare the iteration as unsuccessful and set $x_{k+1} = x_k$.

2. **Stepsize parameter update:** If the iteration was successful, then maintain or increase the corresponding stepsize parameter, by considering $\alpha_{k+1} \in [\alpha_k, \gamma\alpha_k]$.
Otherwise decrease the stepsize parameter, by choosing $\alpha_{k+1} \in [\beta_1\alpha_k, \beta_2\alpha_k]$.

Algorithm 2 can be regarded as a particular instance of Algorithm 1, where no search step is performed, the list L_k is a singleton, corresponding to the current iterate

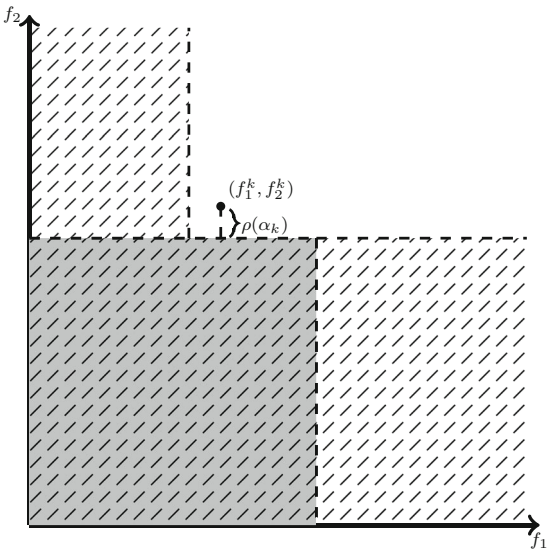


Fig. 2 Selecting a new nondominated point in the min–max direct-search framework

and stepsize parameter $(x_k; \alpha_k)$, with a particular choice of L_{trial} as a subset of the set of computed nondominated points. Figure 2 illustrates the latter claim for a biobjective optimization problem. Consider $F(x_k) = (f_1^k, f_2^k)$ as the objective function value at the current iterate and $\rho(\alpha_k)$ as the current value of the forcing function. The shaded region corresponds to the image of the subset of nondominated points, from which a new iterate can be selected. This set is a subset of the corresponding set in Algorithm 1 (represented by the hatch-lined area). Such restriction leads to a better worst-case complexity bound, compared to the general formulation of DMS.

A result similar to Lemma 3.2 can be established for Algorithm 2.

Lemma 4.1 *Under Assumption 3.1, suppose that the k-th iteration of Algorithm 2 is unsuccessful. Let D_k be the positive spanning set considered, and $\alpha_k > 0$ be the corresponding stepsize. Then,*

$$\mu_{D_k}(x_k) \leq \frac{1}{2}(L_{\max} + c)\,\alpha_k.$$

Proof If iteration k is unsuccessful, then for all directions $d_k \in D_k$

$$f(x_k + \alpha_k d_k) \geq f(x_k) - \frac{c}{2}\alpha_k^2.$$

Hence, there exists $i(d_k) \in I$ such that

$$f_{i(d_k)}(x_k + \alpha_k d_k) \geq f_{i(d_k)}(x_k) - \frac{c}{2}\alpha_k^2.$$

The remainder of the proof is similar to the one of Lemma 3.2. □

The following lemma states that the sequence $\sum_{k=0}^{\infty} \alpha_k^2$ is finite, where $\{\alpha_k\}_{k\geq 0}$ is generated by Algorithm 2. The proof is identical to the one of [13, Lemma 4.1], but we include it for completeness.

Lemma 4.2 [13, Lemma 4.1] *Under Assumption 3.2, the sequence of* $\{\alpha_k\}_{k\geq 0}$ *generated by Algorithm 2 satisfies*

$$\sum_{k=0}^{\infty} \alpha_k^2 \;\leq\; \Omega \;:=\; \frac{\gamma^2}{1-\beta_2^2}\left(\gamma^{-2}\alpha_0^2 + \frac{2}{c}(f(x_0) - F^{\min})\right),$$

where γ, β_2, c *are defined in Algorithm 2.*

Proof We begin by proving that the series $\sum_{k\in S} \alpha_k^2$ is finite. To this end, recall that for every successful iteration, we have

$$f(x_k) - f(x_{k+1}) \geq \frac{c}{2}\alpha_k^2.$$

Moreover, denoting by S the set corresponding to the indexes of successful iterations, since the iterate does not change between two successful iterations, we also have for any $K \geq 0$:

$$\sum_{\substack{k\in S \\ k\leq K}} f(x_k) - f(x_{k+1}) = \sum_{k\leq K} f(x_k) - f(x_{k+1}) = f(x_0) - f(x_{K+1}) \leq f(x_0) - F^{\min},$$

where the last inequality results from Assumption 3.2. As result, we obtain

$$f(x_0) - F^{\min} \geq \sum_{k\in S} f(x_k) - f(x_{k+1}) \geq \sum_{k\in S} \frac{c}{2}\alpha_k^2.$$

Thus, $\sum_{k\in S} \alpha_k^2 \leq \frac{2}{c}(f(x_0) - F^{min}) < \infty$.

To analyze the full series, we consider the set $S := \{k_0, k_1, k_2, \ldots, \}$, where $k_i \geq 0$ is the index of the i-th successful iteration and $k_0 = -1$ is an artificial index corresponding to $\alpha_{-1} = \gamma^{-1}\alpha_0$. With this notation, given the updating rules on α_k, we have that

$$\sum_{k=0}^{\infty} \alpha_k^2 = \sum_{i=0}^{\infty} \sum_{k=k_i+1}^{k_{i+1}} \alpha_k^2 \leq \sum_{i=0}^{\infty} \sum_{k=k_i+1}^{k_{i+1}} \gamma^2 \beta_2^{2(k-k_i-1)} \alpha_{k_i}^2 \leq \frac{\gamma^2}{1-\beta_2^2} \sum_{i=0}^{\infty} \alpha_{k_i}^2.$$

We conclude by observing that

$$\sum_{i=0}^{\infty} \alpha_{k_i}^2 = \gamma^{-2}\alpha_0^2 + \sum_{k\in S} \alpha_k^2 \leq \gamma^{-2}\alpha_0^2 + \frac{2}{c}(f(x_0) - F^{\min}).$$

□

Finally, in the main result of this section, we will prove that Algorithm 2 takes at most $\mathcal{O}(\epsilon^{-2})$ iterations for driving μ below $\epsilon > 0$. Similarly to Algorithm 1, Algorithm 2 cannot be proven globally convergent to a Pareto critical point for an arbitrary choice of positive spanning sets as sets of poll directions, as one can easily present examples where the cone of descent directions, considering all components of the objective function, can be arbitrarily narrow (see Remark 3.1).

Theorem 4.1 *Let Assumptions 3.1, 3.2, and 3.5 hold. For $\epsilon \in]0, 1[$, let k_ϵ be the first iteration index such that $\mu_{k_\epsilon+1} \leq \epsilon$. Then,*

$$k_\epsilon \leq \frac{2}{c\alpha_0^2}\left(f(x_0) - F^{\min}\right) + \frac{\Omega(L_{\max} + c)^2(\mathcal{C}_1 + 1)^2}{4\beta_1^2}\epsilon^{-2},$$

where Ω is defined as in Lemma 4.2.

Proof If $k_\epsilon = 0$, the result trivially holds. Therefore, we assume in what follows that $k_\epsilon > 0$.

For any unsuccessful iteration of index $k \leq k_\epsilon$, we have from Lemma 4.1 that

$$\alpha_k^2 \geq \frac{4\mu_{D_k}^2}{(L_{\max} + c)^2}. \tag{9}$$

Since $\mu_k > \epsilon$, in view of (7), we have $\mu_{D_k} > \epsilon/(\mathcal{C}_1 + 1)$. Therefore, using (9), we have

$$\alpha_k^2 \geq \frac{4\epsilon^2}{(L_{\max} + c)^2(\mathcal{C}_1 + 1)^2}. \tag{10}$$

Considering the updating rules on the stepsize, for any successful iteration of index $k_\epsilon \geq k > j_1$, where j_1 is the index of the first unsuccessful iteration, there exists an index of an unsuccessful iteration $j(k) \leq k$ (with possibly $j(k) = j_1$) such that $\alpha_k \geq \beta_1\alpha_{j(k)}$. Putting this together with (10) yields:

$$\forall k \in S, k_\epsilon \geq k > j_1, \quad \alpha_k^2 \geq \frac{4\beta_1^2\epsilon^2}{(L_{\max} + c)^2(\mathcal{C}_1 + 1)^2},$$

where S denotes the set of successful iterations. Using now the result of Lemma 4.2, we have:

$$\Omega \geq \sum_{k=0}^{\infty}\alpha_k^2 \geq \sum_{k=j_1+1}^{k_\varepsilon}\alpha_k^2 \geq (k_\varepsilon - j_1)\frac{4\beta_1^2\epsilon^2}{(L_{\max} + c)^2(\mathcal{C}_1 + 1)^2}.$$

Thus,

$$k_\varepsilon - j_1 \leq \frac{\Omega(L_{\max} + c)^2(\mathcal{C}_1 + 1)^2}{4\beta_1^2}\epsilon^{-2}.$$

Since j_1 is the index of the first unsuccessful iteration, one can trivially show that $j_1 \leq \frac{2}{c\alpha_0^2}\left(f(x_0) - F^{\min}\right)$. Then, the thesis follows. □

The previous theorem allows us to establish a WCC bound in terms of the number of function evaluations for Algorithm 2 (see also Remark 3.2).

Corollary 4.1 *Let all the assumptions of Theorem 4.1 hold. To achieve $\mu_k < \epsilon$, Algorithm 2 takes at most $\mathcal{O}\left(n\epsilon^{-2}\right)$ function evaluations.*

5 Conclusions

In this work, we analyzed the worst-case complexity of some direct-search derivative-free algorithms for unconstrained multiobjective nonconvex smooth optimization problems. In the case of Direct Multisearch [7], we derived a complexity bound of $\mathcal{O}(|L(\epsilon)|\epsilon^{-2m})$ for driving a criticality measure below $\epsilon > 0$. We then proposed a min–max approach to the multiobjective derivative-free optimization problem, which proved to be a particular instance of Direct Multisearch, but presented a worst-case complexity bound of $\mathcal{O}(\epsilon^{-2})$ for driving the same criticality measure below $\epsilon > 0$. This result is identical, in terms of ϵ, to the one established in [10] for gradient descent, considering the same class of problems. For the (strongly) convex case, where all the components of the objective function are (strongly) convex, it remains as an open question whether similar complexity bounds to those derived in [10] could be established for the algorithms considered in this paper.

Acknowledgements The authors would like to thank the three anonymous referees, whose comments and suggestions much improved the quality of the paper. Funding support for Ana Luísa Custódio and Rohollah Garmanjani was provided by national funds through FCT–Fundação para a Ciência e a Tecnologia I. P., under the scope of Projects PTDC/MAT-APL/28400/2017 and UIDB/00297/2020. Support for Elisa Riccietti was provided by TOTAL E&P.

References

1. Marler, R.T., Arora, J.S.: Survey of multi-objective optimization methods for engineering. Struct. Multidiscip. Optim. **26**, 369–395 (2004)
2. Ehrgott, M.: Multicriteria Optimization. Springer, Heidelberg (2005)
3. Custódio, A.L., Emmerich, M., Madeira, J.F.A.: Recent developments in derivative-free multiobjective optimization. Comput. Technol. Rev. **5**, 1–30 (2012)
4. Kolda, T.G., Lewis, R.M., Torczon, V.: Optimization by direct search: new perspectives on some classical and modern methods. SIAM Rev. **45**, 385–482 (2003)
5. Conn, A.R., Scheinberg, K., Vicente, L.N.: Introduction to Derivative-Free Optimization. MPS-SIAM Series on Optimization. SIAM, Philadelphia (2009)
6. Audet, C., Hare, W.: Derivative-Free and Blackbox Optimization. Springer Series in Operations Research and Financial Engineering. Springer, Cham (2017)
7. Custódio, A.L., Madeira, J.F.A., Vaz, A.I.F., Vicente, L.N.: Direct multisearch for multiobjective optimization. SIAM J. Optim. **21**, 1109–1140 (2011)
8. Cartis, C., Gould, N.I.M., Toint, Ph.L.: On the oracle complexity of first-order and derivative-free algorithms for smooth nonconvex minimization. SIAM J. Optim. **22**, 66–86 (2012)
9. Dodangeh, M., Vicente, L.N.: Worst case complexity of direct search under convexity. Math. Program. **155**, 307–332 (2016)

10. Fliege, J., Vaz, A.I.F., Vicente, L.N.: Complexity of gradient descent for multiobjective optimization. Optim. Methods Softw. **34**, 949–959 (2019)
11. Garmanjani, R., Vicente, L.N.: Smoothing and worst-case complexity for direct-search methods in nonsmooth optimization. IMA J. Numer. Anal. **33**, 1008–1028 (2013)
12. Grapiglia, G.N., Yuan, J., Yuan, Y.: On the convergence and worst-case complexity of trust-region and regularization methods for unconstrained optimization. Math. Program. **152**, 491–520 (2015)
13. Gratton, S., Royer, C.W., Vicente, L.N., Zhang, Z.: Direct search based on probabilistic descent. SIAM J. Optim. **25**, 1515–1541 (2015)
14. Konečný, J., Richtárik, P.: Simple Complexity Analysis of Simplified Direct Search. Technical reports, arXiv:1410.0390 (2014)
15. Nesterov, Y.: Introductory Lectures on Convex Optimization. Applied Optimization. Kluwer Academic Publishers, Boston (2004)
16. Vicente, L.N.: Worst case complexity of direct search. EURO J. Comput. Optim. **1**, 143–153 (2013)
17. Gratton, S., Sartenaer, A., Toint, Ph.L.: Recursive trust-region methods for multiscale nonlinear optimization. SIAM J. Optim. **19**, 414–444 (2008)
18. Cartis, C., Sampaio, Ph.R., Toint, Ph.L.: Worst-case evaluation complexity of non-monotone gradient-related algorithms for unconstrained optimization. Optimization **64**, 1349–1361 (2015)
19. Nesterov, Y., Polyak, B.T.: Cubic regularization of Newton method and its global performance. Math. Program. **108**, 177–205 (2006)
20. Cartis, C., Gould, N.I.M., Toint, Ph.L.: Adaptive cubic regularisation methods for unconstrained optimization. Part II: worst-case function- and derivative-evaluation complexity. Math. Program. **130**, 295–319 (2011)
21. Bergou, E.H., Gorbunov, E., Richtárik, P.: Stochastic three points method for unconstrained smooth optimization. SIAM J. Optim. **30**, 2726–2749 (2020)
22. Calderón, L., Diniz-Ehrhardt, M.A., Martínez, J.M.: On high-order model regularization for multiobjective optimization. Optim. Methods Softw. (2020). https://doi.org/10.1080/10556788.2020.1719408
23. Žilinskas, A.: On the worst-case optimal multi-objective global optimization. Optim. Lett. **7**, 1921–1928 (2013)
24. Calvin, J.M., Žilinskas, A.: On efficiency of a single variable bi-objective optimization algorithm. Optim. Lett. **14**, 259–267 (2020)
25. Villacorta, K.D.V., Oliveira, P.R., Soubeyran, A.: A trust-region method for unconstrained multiobjective problems with applications in satisficing processes. J. Optim. Theory Appl. **160**, 865–889 (2014)
26. Fliege, J., Svaiter, B.F.: Steepest descent methods for multicriteria optimization. Math. Methods Oper. Res. **51**, 479–494 (2000)
27. Zitzler, E.: Evolutionary Algorithms for Multiobjective Optimization: Methods and Applications. Ph.D. Thesis, Swiss Federal Institute of Technology Zurich, Switzerland (1999)
28. Thomann, J., Eichfelder, G.: A trust-region algorithm for heteregeneous multiobjective optimization. SIAM J. Optim. **29**, 1017–1047 (2019)
29. Dennis Jr., J.E., Woods, D.J.: Optimization on microcomputers: the Nelder–Mead simplex algorithm. In: Wouk, A. (ed.) New Computing Environments: Microcomputers in Large-Scale Computing, pp. 116–122. SIAM, Philadelphia (1987)
30. Liuzzi, G., Lucidi, S., Rinaldi, F.: A derivative-free approach to constrained multiobjective nonsmooth optimization. SIAM J. Optim. **26**, 2744–2774 (2016)

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Affiliations

Ana Luísa Custódio[1] · Youssef Diouane[2] · Rohollah Garmanjani[1] · Elisa Riccietti[3]

Youssef Diouane
youssef.diouane@isae-supaero.fr

Rohollah Garmanjani
r.garmanjani@fct.unl.pt

Elisa Riccietti
elisa.riccietti@enseeiht.fr

1 Department of Mathematics, FCT-UNL-CMA, Campus de Caparica, 2829-516 Caparica, Portugal

2 ISAE-SUPAERO, Université de Toulouse, 31055 Toulouse Cedex 4, France

3 IRIT - ENSEEIHT, Université de Toulouse, 31071 Toulouse Cedex 7, France