**JOTA manuscript No.**
(will be inserted by the editor)

# Worst-case complexity bounds of directional direct-search methods for multiobjective derivative-free optimization

**A. L. Custódio · Y. Diouane · R. Garmanjani · E. Riccietti**

Received: date / Accepted: date

**Abstract** Direct Multisearch (DMS) 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. Considering nonconvex smooth functions, we will show that the DMS algorithm takes at most $\mathcal{O}(\epsilon^{-2m})$ iterations for driving a criticality measure below $\epsilon > 0$ (here $m$ represents the number of components of the objective function). We then focus on a particular instance of DMS, which considers a more strict criterion for accepting new nondominated points. In this case, we can establish a better worst-case complexity bound of $\mathcal{O}(\epsilon^{-2})$ for driving the same criticality measure below $\epsilon > 0$.

**Keywords:** Derivative-free methods Directional direct-search Worst-case complexity Nonconvex smooth optimization

## 1 Introduction

In this work, we consider the following multiobjective derivative-free optimization problem

$$\begin{aligned} \min \quad & F(x) \equiv (f_1(x), \ldots, f_m(x))^\top \\ \text{s.t.} \quad & x \in \mathbb{R}^n, \end{aligned} \tag{1}$$

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. We assume that function derivatives are not available for use, neither can be numerically approximated.

A. L. Custódio
Department of Mathematics, FCT-UNL-CMA, Campus de Caparica, 2829-516 Caparica, Portugal.
Support for this author was provided by Fundação para a Ciência e a Tecnologia (Portuguese Foundation for Science and Technology) under the projects PTDC/MAT-APL/28400/2017 and UID/MAT/00297/2019.
E-mail: alcustodio@fct.unl.pt

Y. Diouane
ISAE-SUPAERO, Université de Toulouse, 31055 Toulouse Cedex 4, France. E-mail: youssef.diouane@isae.fr

R. Garmanjani
Department of Mathematics, FCT-UNL-CMA, Campus de Caparica, 2829-516 Caparica, Portugal.
Support for this author was provided by Fundação para a Ciência e a Tecnologia (Portuguese Foundation for Science and Technology) under the projects PTDC/MAT-APL/28400/2017 and UID/MAT/00297/2019.
E-mail: r.garmanjani@fct.unl.pt

E. Riccietti
IRIT - ENSEEIHT, Université de Toulouse, 31071 Toulouse Cedex 7, France.
Support for this author was provided by TOTAL E&P.
E-mail: elisa.riccietti@enseeiht.fr

We are interested in establishing worst-case complexity (WCC) bounds for directional direct-search methods, when used for solving problem (1). Each iteration of this class of algorithms can be divided in a search 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 [1]. 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 and in the concept of Pareto dominance [1]. No model is built for the objective function, neither any attempt of estimating derivatives is considered.

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 [2–10]). Usually, the performance of an algorithm is measured by the number of iterations/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 [9] derived a WCC bound of $\mathcal{O}\left(\epsilon^{-2}\right)$ for gradient descent algorithms. A similar bound has been achieved for trust-region [11] and line-search [12] methods. Nesterov and Polyak [13] investigated the use of cubic regularization techniques and then Cartis *et al* [14] 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 [2]. In the context of derivative-free optimization, directional direct-search was the first class of algorithms for which worst-case complexity bounds were established [10]. 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 the worst-case complexity bound of steepest descent for the unconstrained minimization of a smooth function. The complexity of directional direct-search methods for the optimization of convex functions as been addressed in [3]. The bound of $\mathcal{O}\left(\epsilon^{-2}\right)$ has been improved to $\mathcal{O}\left(\epsilon^{-1}\right)$, which is identical to the one of steepest descent, under convexity. Complexity results have also been established for the nonsmooth case. In [5] 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 direct-search along with a worst-case complexity bound of $\mathcal{O}\left(\epsilon^{-2}\right)$ has been established in [7]. 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 [8]. In [15], parallelization allows to establish WCC bounds independent of problem dimension.

As for the worst-case complexity of derivative-based methods for solving multiobjective optimization problems of type (1), it has been shown in [6] that trust-region methods provide a worst-case complexity bound of $\mathcal{O}\left(\epsilon^{-2}\right)$. A similar bound has been derived in [4], and improved to $\mathcal{O}\left(\epsilon^{-1}\right)$ or $\mathcal{O}(\log \epsilon^{-1})$, assuming convexity or strong convexity of the different components $f_i, i \in I$ of the objective function. In [16], 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.

To the best of our knowledge, no attempt in literature has been made for analyzing the worst-case complexity of multiobjective derivative-free optimization algorithms. In this work, we first establish a worst-case complexity bound for the original Direct Multisearch (DMS) [1] class of methods. We will show that the DMS algorithm takes at most $\mathcal{O}\left(\epsilon^{-2m}\right)$ iterations for driving a criticality measure below $\epsilon > 0$. 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 [17], which is widely used in the literature (e.g., see [18] and [6, §4.2] for multiobjective trust-region methods or [17] 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 to the one derived for multiobjective gradient descent methods [6, 4].

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 the maximum of the objectives. In particular, the analysis in [10], 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 Section 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 Section 3. Section 4 introduces the min-max formulation and establishes a worst-case complexity bound for it. Some conclusions are drawn in Section 5.

## 2 Preliminaries

When solving a multiobjective optimization problem of type (1), the goal is to identify a *local Pareto optimum* [19], i.e. a point $x^* \in \mathbb{R}^n$ such that it does not exist another point $x$ in a neighborhood of $x^*$ that dominates $x^*$, meaning that $F(x) \not\prec_F F(x^*)$ for all $x$ in that neighborhood. In fact, we say that point $x$ dominates point $x^*$ when $F(x) \prec_F F(x^*)$, i.e., when $F(x^*) - F(x) \in \mathbb{R}^m_+ \setminus \{0\}$.

A necessary condition for $x^* \in \mathbb{R}^n$ to be a local Pareto optimum is [19]:

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

A point satisfying (2) is called a *Pareto critical point* [19]. We are then interested in finding Pareto critical points. In what comes next, $\|\cdot\|$ will denote the vector or matrix $\ell_2$-norm.

Following [19], to characterize Pareto points, we are going to use, for a given $x \in \mathbb{R}^n$, the function:

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

Fliege and Svaiter [19] 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** *[19, 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 $\mu$ 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 $\mu$ 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 [1] 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, and for that a forcing function $\rho : (0, +\infty) \longrightarrow (0, +\infty)$ will be required. Following [20], $\rho$ 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 $\ell_\infty$ norm to $D(L)$ is no larger than $a > 0$. Algorithm 1 corresponds to an instance of the original DMS [1] method, using a globalization strategy based on imposing 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) 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) successful and set $L_{k+1} = L_{trial}$. Otherwise, declare the iteration (and the poll step) unsuccessful and set $L_{k+1} = L_k$.
4. **Stepsize parameter update:** If the iteration was successful then maintain or increase the corresponding stepsize parameters, by considering $\alpha_{k,new} \in [\alpha_k, \gamma\alpha_k]$ and replacing 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. 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 $\alpha$ 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** *The functions $f_1, \ldots, f_m$ are lower and upper bounded in a set containing $D(L_0) = D(\{(x_0; \alpha_0)\})$, with lower bound $f_i^{\min}$ and upper bound $f_i^{\max}$ for each function $f_i, i \in I$. Let $F^{\min} = \min\{f_1^{\min}, \ldots, f_m^{\min}\}$ and $F^{\max} = \max\{f_1^{\max}, \ldots, f_m^{\max}\}$.*

We also assume, as it is done in classical directional direct-search [20], that all positive spanning sets considered by the algorithm include bounded directions and have cosine measure bounded away from zero.

**Assumption 3.3** *All positive spanning sets $D_k$ used for polling satisfy $0 < d_{\min} \leq \|d\| \leq d_{\max}$, for all $d \in D_k$ and $\mathrm{cm}(D_k) \geq \kappa > 0$, where*

$$\mathrm{cm}(D_k) \equiv \min_{v\neq 0} \max_{d\in D_k} \frac{d^\top v}{\|d\|\|v\|}.$$

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) corresponding to the current list of nondominated points always increases.

**Definition 3.1** [21, Definition 4.2] 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:

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

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 $r$.

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

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

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 area corresponding to the set of nondominated points has increased exactly in $\rho(\alpha_k)^2$.

We will make use of the following result, which is a direct consequence of the one presented in [20, 22], that establishes a relationship between the size of the gradient of the objective function and the stepsize parameter at an unsuccessful iteration of a directional direct-search method.

**Theorem 3.1** *Let Assumptions 3.1 and 3.3 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. Then, there is $i(k) \in I$ such that:*

$$\|\nabla f_{i(k)}(x_k)\| \;\leq\; \kappa^{-1}\left(\frac{L_{\max}}{2}\alpha_k d_{\max} + \frac{\rho(\alpha_k)}{\alpha_k d_{\min}}\right). \tag{4}$$

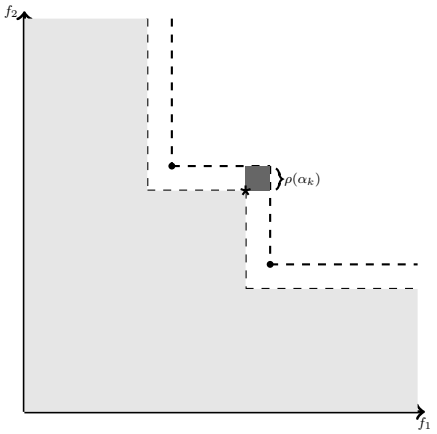


Fig. 1: Hypervolume increase at a successful iteration.

In the following theorem, we will derive a bound on the number of successful iterations required to drive $\mu_k$ below a given small positive threshold.

**Theorem 3.2** *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 3.3 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 $|S_{j_1}(k_0)| = \mathcal{O}\left(\epsilon^{-\frac{pm}{\min(p-1,1)}}\right)$ successful iterations.*

*Proof* Let us assume that $\mu_k > \epsilon$, for $k = k_0, \ldots, j_1$.

In view of Theorem 3.1, for an unsuccessful iteration $k$, we have

$$\|\nabla f_{i(k)}(x_k)\| \ \leq \ \kappa^{-1}\left(\frac{L_{\max}}{2}\alpha_k d_{\max} + d_{\min}^{-1} c\alpha_k^{p-1}\right), \text{ for some } i(k) \in I. \tag{5}$$

Hence, by applying Lemma 2.2, we obtain

$$\epsilon \ \leq \ \kappa^{-1}\left(\frac{L_{\max}}{2}\alpha_k d_{\max} + d_{\min}^{-1} c\alpha_k^{p-1}\right), \tag{6}$$

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

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

where $\mathcal{L}_1 = \kappa^{-1}\left(\frac{L_{\max}}{2}d_{\max} + d_{\min}^{-1}c\right)$. If $\alpha_k \geq 1$, then $\alpha_k \geq \epsilon$. 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

$$\alpha_k \ \geq \ \mathcal{L}_2 \epsilon^{\frac{1}{\min(p-1,1)}}, \tag{7}$$

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 appearing from $k_0$ to $k$. From (7), Definition 3.1, and by the choice of forcing function,

$$\begin{aligned}
\mathrm{HI}(L_k) - \mathrm{HI}(L_{k_u}) \ &\geq \ (k - k_u - 1)(c\alpha_{k_{u+1}}^p)^m \\
&\geq \ |S_k(k_u)|(c\beta_1^p \alpha_{k_u}^p)^m \\
&\geq \ |S_k(k_u)|\left(c\beta_1^p \mathcal{L}_2^p \epsilon^{\frac{p}{\min(p-1,1)}}\right)^m,
\end{aligned}$$

where $S_k(k_u)$ represents the set of successful iterations from $k_u$ to $k$. By a similar reasoning, for $1 \leq i \leq u$, we obtain

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

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

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

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

Now, in order to obtain a bound on the total number of iterations for driving $\mu_k$ below a given threshold, it remains to find a bound on the number of unsuccessful iterations, which is derived in the next result. The proof is identical to the one of [10, Theorem 3].

**Theorem 3.3** *Let all the assumptions of Theorem 3.2 hold. Then, to achieve $\mu_k \leq \epsilon$ starting from $k_0$, Algorithm 1 takes at most*

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

*unsuccessful iterations.*

Combining Theorems 3.2 and 3.3, it can be seen that Algorithm 1 takes at most $\mathcal{O}\left(\epsilon^{-\frac{pm}{\min(p-1,1)}}\right)$ iterations to bring $\mu_k < \epsilon$ for some $k \geq 0$. The best complexity bound is then derived by setting $p = 2$, which leads to the bound of $\mathcal{O}\left(\epsilon^{-2m}\right)$.

**Corollary 3.1** *Let all the assumptions of Theorem 3.2 hold. To achieve $\mu_k < \epsilon$, Algorithm 1 takes at most $\mathcal{O}\left(\epsilon^{-\frac{pm}{\min(p-1,1)}}\right)$ iterations. When $p = 2$ this bound is $\mathcal{O}\left(\epsilon^{-2m}\right)$.*

One can see that the bound for DMS does not conform with the bound $\mathcal{O}\left(\epsilon^{-2}\right)$ for the gradient descent derived in [4] for problem (1). The main reason 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 [4] 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 find a Pareto front, whereas the multiobjective gradient descent algorithm finds a Pareto critical point.

Indeed, being strict in defining an iteration as a successful one would lead to the same complexity bound as that of gradient descent. 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 $\mathcal{O}\left(\epsilon^{-2}\right)$.

## 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:

$$\min f(x)$$

with

$$f(x) \equiv \max_{i \in I} f_i(x). \tag{8}$$

Algorithm 2 considers a Direct-Search (DS) approach with a stricter criterion for accepting new nondominated points, since the sufficient decrease condition should be satisfied by all components of the objective function. In this case, rather than a Pareto front, only one Pareto critical point will be computed

**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 $\alpha_{k+1} \in [\beta_1\alpha_k, \beta_2\alpha_k]$.

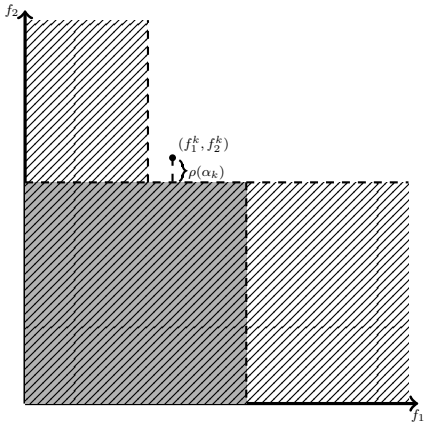


Fig. 2: Selecting a new nondominated point in the min-max direct-search framework.

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 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 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, comparing to the general formulation of DMS.

The following lemma is an auxiliary result for the remaining proofs and establishes a relationship between the positive spanning set and the gradients of the components of the objective function at a given iteration.

**Lemma 4.1** *Let $D_k$ be a positive spanning set and assume that $\nabla f_i(x_k)$ is well-defined for all $i \in I$. There exists a direction $\tilde{d}_k \in D_k$ such that*

$$\operatorname{cm}(D_k) d_{\min} \min_{i\in I} \|\nabla f_i(x_k)\| \leq -\max_{i\in I} \nabla f_i(x_k)^\top \tilde{d}_k. \tag{9}$$

*Proof* Let

$$\tilde{\mu}_k \equiv -\min_{d\in D_k} \max_{i\in I} \nabla f_i(x_k)^\top d,$$

and $\tilde{d}_k$ be its solution, that is $\tilde{d}_k \in D_k$ and

$$\tilde{\mu}_k = -\max_{i\in I} \nabla f_i(x_k)^\top \tilde{d}_k. \tag{10}$$

By definition of $\tilde{\mu}_k$, one has

$$\tilde{\mu}_k = \max_{d\in D_k} \left( -\max_{i\in I} \nabla f_i(x_k)^\top d \right) \geq -\max_{i\in I} \nabla f_i(x_k)^\top d, \quad \forall d \in D_k. \tag{11}$$

From the definition of cosine measure of a positive spanning set, $\operatorname{cm}(D_k)$, it follows that for all $i \in I$ it exists $d_k^i \in D_k$ such that

$$\operatorname{cm}(D_k) \|\nabla f_i(x_k)\| \|d_k^i\| \leq -\nabla f_i(x_k)^\top d_k^i.$$

Then

$$\max_{i\in I} \left( -\operatorname{cm}(D_k) \|\nabla f_i(x_k)\| \|d_k^i\| \right) \geq \max_{i\in I} \nabla f_i(x_k)^\top d_k^i,$$

which is equivalent to

$$\min_{i\in I} \left( \operatorname{cm}(D_k) \|\nabla f_i(x_k)\| \|d_k^i\| \right) \leq -\max_{i\in I} \nabla f_i(x_k)^\top d_k^i.$$

From (10) and (11), taking into account that $d_k^i \in D_k$, we have:

$$\operatorname{cm}(D_k) d_{\min} \min_{i\in I} \|\nabla f_i(x_k)\| \leq -\max_{i\in I} \nabla f_i(x_k)^\top d_k^i \leq \tilde{\mu}_k = -\max_{i\in I} \nabla f_i(x_k)^\top \tilde{d}_k.$$

■

Theorem 3.1 is then adapted in Lemma 4.2 when Algorithm 2 is regarded.

**Lemma 4.2** *Under Assumptions 3.1 and 3.3, suppose that the k-th iteration of Algorithm 2 is unsuccessful. Then*

$$\min_{i\in I} \|\nabla f_i(x_k)\| \leq \frac{L_{\max} d_{\max}^2 + c}{2 d_{\min} \kappa} \alpha_k.$$

*Proof* Using Lemma 4.1, there exists a direction $\tilde{d}_k \in D_k$ such that

$$\operatorname{cm}(D_k) d_{\min} \min_{i\in I} \|\nabla f_i(x_k)\| \leq -\max_{i\in I} \nabla f_i(x_k)^\top \tilde{d}_k. \tag{12}$$

Moreover, since the $k$-th iteration is unsuccessful and $\tilde{d}_k \in D_k$, one has

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

Hence,

$$\begin{aligned}
-\frac{c}{2}\alpha_k^2 \leq & f(x_k + \alpha_k \tilde{d}_k) - f(x_k) = \max_{i\in I} f_i(x_k + \alpha_k \tilde{d}_k) - \max_{i\in I} f_i(x_k) \\
\leq & \max_{i\in I} f_i(x_k) + \max_{i\in I} \int_0^1 \nabla f_i(x_k + t\alpha_k \tilde{d}_k)^\top \alpha_k \tilde{d}_k \, dt - \max_{i\in I} f_i(x_k) \\
= & \max_{i\in I} \int_0^1 \nabla f_i(x_k + t\alpha_k \tilde{d}_k)^\top \alpha_k \tilde{d}_k \, dt
\end{aligned}$$

Adding (12) multiplied by $\alpha_k$ to both sides yields:

$$\begin{aligned}-\frac{c}{2}\alpha_k^2 + \alpha_k \,\mathrm{cm}(D_k) d_{\min} \min_{i\in I} \|\nabla f_i(x)\| \;&\leq\; \max_{i\in I} \int_0^1 \nabla f_i(x_k + t\alpha_k \tilde{d}_k)^\top \alpha_k \tilde{d}_k \, dt \\ &\quad - \alpha_k \max_i \nabla f_i(x)^\top \tilde{d}_k \\ &\leq \alpha_k \int_0^1 (\nabla f_j(x_k + t\alpha_k \tilde{d}_k) - \nabla f_j(x_k))^\top \tilde{d}_k \, dt \\ &\leq \alpha_k^2 \frac{L_{\max}}{2} d_{\max}^2,\end{aligned}$$

where $j \equiv \arg\max_{i\in I} \int_0^1 \left(\nabla f_i(x_k + t\alpha_k \tilde{d}_k) - \nabla f_i(x_k)\right)^\top \tilde{d}_k \, dt$. Then, the thesis follows. ■

The following lemma states that the sequence $\sum_{k=0}^{\infty} \alpha_k^2$ (where $\{\alpha_k\}_{k\geq 0}$ is generated by Algorithm 2) is finite.

**Lemma 4.3** *[7, Lemma 4.1] Under Assumption 3.2, the sequence of $\{\alpha_k\}_{k\geq 0}$ generated by Algorithm 2 satisfies*

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

*where $\gamma, \beta_2, c$ are defined in Algorithm 2.*

Finally, in the main result of this section, we will prove that Algorithm 2 takes at most $\mathcal{O}(\epsilon^{-2})$ iterations for driving $\mu_k$ below $\epsilon > 0$.

**Theorem 4.1** *Let Assumptions 3.1, 3.2, and 3.3 hold. For $\epsilon \in (0, 1]$, let $k_\epsilon$ be the first iteration index such that $\mu_{k_\epsilon+1} \leq \epsilon$. Then,*

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

*where $\Omega$ is defined as in Lemma 4.3.*

*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 4.2 that

$$\alpha_k^2 \geq \frac{4\kappa^2 d_{\min}^2}{(L_{\max} d_{\max}^2 + c)^2} \min_{0\leq l\leq k} \min_{i\in I} \|\nabla f_i(x_l)\|^2. \tag{13}$$

Since, for all $0 \leq l \leq k$, $\mu_l > \epsilon$, Lemma 2.2 implies that $\|\nabla f_i(x_l)\| > \epsilon$ for all $i \in I$, i.e., $\min_{i\in I} \|\nabla f_i(x_l)\| > \epsilon$ for all $0 \leq l \leq k$.

Hence, (13) implies

$$\alpha_k^2 \geq \frac{4\kappa^2 d_{\min}^2}{(L_{\max} d_{\max}^2 + c)^2} \min_{0\leq l\leq k} \min_{i\in I} \|\nabla f_i(x_l)\|^2 \geq \frac{4\kappa^2 d_{\min}^2}{(L_{\max} d_{\max}^2 + c)^2} \epsilon^2. \tag{14}$$

On the other hand, using the updating rules on the stepsize, for any successful iteration of index $k_\epsilon \geq k \geq j_1$, 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 (14) yields:

$$\forall k \in S, k_\epsilon \geq k \geq j_1, \quad \alpha_k^2 \geq \frac{4\kappa^2 d_{\min}^2 \beta_1^2}{(L_{\max} d_{\max}^2 + c)^2} \epsilon^2,$$

where $S$ denotes the set of successful iterations. Using now the result of Lemma 4.3, we have:

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

Thus

$$k_\varepsilon - j_1 \;\leq\; \frac{\Omega(L_{\max} d_{\max}^2 + c)^2}{4\kappa^2 d_{\min}^2 \beta_1^2} \epsilon^{-2}.$$

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. ■

## 5 Conclusions

In this work, we analyzed the worst-case complexity of derivative-free algorithms for multiobjective nonconvex smooth optimization problems. In the case of Direct Multisearch [1], we derived a complexity bound of $\mathcal{O}(\epsilon^{-2m})$ 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 to the one established in [4] 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 [4] could be established for the algorithms considered in this paper.

## References


1. A. L. Custódio, J. F. A. Madeira, A. I. F. Vaz, and L. N. Vicente. Direct multisearch for multiobjective optimization. *SIAM J. Optim.*, 21:1109–1140, 2011.
2. C. Cartis, N. I. M. Gould, and Ph. L. Toint. On the oracle complexity of first-order and derivative-free algorithms for smooth nonconvex minimization. *SIAM J. Optim.*, 22:66–86, 2012.
3. M. Dodangeh and L. N. Vicente. Worst case complexity of direct search under convexity. *Math. Program.*, 155:307–332, 2016.
4. J. Fliege, A. I. F. Vaz, and L. N. Vicente. Complexity of gradient descent for multiobjective optimization. *Optim. Methods Softw.*, pages 1–11, 2018.
5. R. Garmanjani and L. N. Vicente. Smoothing and worst-case complexity for direct-search methods in nonsmooth optimization. *IMA J. Numer. Anal.*, 33:1008–1028, 2013.
6. G. N. Grapiglia, J. Yuan, and Y. Yuan. On the convergence and worst-case complexity of trust-region and regularization methods for unconstrained optimization. *Math. Program.*, 152:491–520, 2015.
7. S. Gratton, C. W. Royer, L. N. Vicente, and Z. Zhang. Direct search based on probabilistic descent. *SIAM J. Optim.*, 25:1515–1541, 2015.
8. J. Konečný and P. Richtárik. Simple complexity analysis of simplified direct search. Technical report, Available at https://arxiv.org/abs/1410.0390, 2014.
9. Y. Nesterov. *Introductory Lectures on Convex Optimization*. Springer, New York, 2004.
10. L. N. Vicente. Worst case complexity of direct search. *EURO Journal on Computational Optimization*, 1:143–153, 2013.
11. S. Gratton, A. Sartenaer, and Ph. L. Toint. Recursive trust-region methods for multiscale nonlinear optimization. *SIAM J. Optim.*, 19:414–444, 2008.
12. C. Cartis, Ph. R. Sampaio, and Ph. L. Toint. Worst-case evaluation complexity of non-monotone gradient-related algorithms for unconstrained optimization. *Optimization*, 64:1349–1361, 2015.
13. Y. Nesterov and B. T. Polyak. Cubic regularization of Newton method and its global performance. *Math. Program.*, 108:177–205, 2006.
14. C. Cartis, N. I. M. Gould, and Ph. L. Toint. Adaptive cubic regularisation methods for unconstrained optimization. Part II: worst-case function- and derivative-evaluation complexity. *Math. Program.*, 130:295–319, 2011.
15. E. Bergou, E. Gorbunov, and P. Richtárik. Stochastic three points method for unconstrained smooth optimization. Technical report, Available at https://arxiv.org/abs/1902.03591, 2019.
16. L. Calderón, M. A. Diniz-Ehrhardt, and J. M. Martínez. On high-order model regularization for multiobjective optimization. Technical report, Available at http://www.optimization-online.org/DB_HTML/2018/12/6997.html, 2018.
17. R. T. Marler and J. S. Arora. Survey of multi-objective optimization methods for engineering. *Struct. Multidisciplinary Optim.*, 26:369–395, 2004.
18. K. V. Villacorta, P. R. Oliveira, and A. Soubeyran. A trust-region method for unconstrained multiobjective problems with applications in satisficing processes. *J. Optim. Theory Appl.*, 160:865–889, 2014.
19. J. Fliege and B. F. Svaiter. Steepest descent methods for multicriteria optimization. *Math. Methods Oper. Res.*, 51:479–494, 2000.

20. T. G. Kolda, R. M. Lewis, and V. Torczon. Optimization by direct search: New perspectives on some classical and modern methods. *SIAM Rev.*, 45:385–482, 2003.
21. A. L. Custódio, M. Emmerich, and J. F. A. Madeira. Recent developments in derivative-free multiobjective optimization. *Computational Technology Reviews*, 5:1–30, 2012.
22. A. R. Conn, K. Scheinberg, and L. N. Vicente. *Introduction to derivative-free optimization*, volume 8 of *MPS-SIAM Series on Optimization*. SIAM, Philadelphia, 2009.