Complexity results and active-set identification of a derivative-free method for bound-constrained problems

Andrea Brilli*, Andrea Cristofari†, Giampaolo Liuzzi*, Stefano Lucidi*

*Department of Computer, Control and Management Engineering
Sapienza University of Rome, Italy
Via Ariosto, 25, 00185 Rome, Italy
E-mail: `brilli@diag.uniroma1.it`, `liuzzi@diag.uniroma1.it`, `lucidi@diag.uniroma1.it`

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

Abstract. In this paper, we analyze a derivative-free line search method designed for bound-constrained problems. Our analysis demonstrates that this method exhibits a worst-case complexity comparable to other derivative-free methods for unconstrained and linearly constrained problems. In particular, when minimizing a function with n variables, we prove that at most $\mathcal{O}\left(n\epsilon^{-2}\right)$ iterations are needed to drive a criticality measure below a predefined threshold ϵ, requiring at most $\mathcal{O}\left(n^2\epsilon^{-2}\right)$ function evaluations. We also show that the total number of iterations where the criticality measure is not below ϵ is upper bounded by $\mathcal{O}\left(n^2\epsilon^{-2}\right)$. Moreover, we investigate the method capability to identify active constraints at the final solutions. We show that, after a finite number of iterations, all the active constraints satisfying the strict complementarity condition are correctly identified.

Keywords. Worst-case complexity. Active-set identification. Derivative-free methods.

MSC2000 subject classifications. 90C56. 90C30. 90C26.

1 Introduction

Let us consider the following nonlinear bound-constrained optimization problem:

$$\begin{aligned} \min\ & f(x) \\ & l_i \le x_i \le u_i, \qquad i = 1, \ldots, n, \end{aligned} \tag{1}$$

where $l_i, u_i \in \mathbb{R} \cup \{\pm\infty\}$, $l_i < u_i$, $i = 1, \ldots, n$.

We restrict ourselves to considering problem (1) when function values are given by a time consuming black-box oracle. Hence, the analytical expression of f is not available and first-order information cannot be explicitly used nor approximated within a reasonable amount of time. In such a context, derivative-free methods [2, 13, 30] are usually employed to solve the problem.

In the literature, several derivative-free methods have been proposed to solve problem (1) (even with more general constraints). In particular, we can distinguish among model-based methods [10, 13, 22, 24, 27, 38], where the objective function is sampled in a neighborhood of the current point to build an appropriate model to be minimized, direct-search methods [1, 23, 28, 29, 31], where the objective function is sampled in a neighborhood of the current point in order to find descent, and

1

line search methods [33, 35], where directions are explored by allowing the stepsize to dynamically expand.

For model-based and direct-search methods applied to problems with linear constraints (thus including (1)), a worst-case analysis can be found in [23, 27], providing upper bounds on the maximum number of iterations and function evaluations needed to drive a criticality measure below a prespecified threshold. In particular, in [23], it is shown that at most $\mathcal{O}\left(n\epsilon^{-2}\right)$ iterations and $\mathcal{O}\left(n^2\epsilon^{-2}\right)$ function evaluations are needed, for a (deterministic) direct-search method, to produce the first point with a criticality measure below $\epsilon > 0$, matching the same complexity for the unconstrained case [19, 40]. In [27], similar bounds of $\mathcal{O}\left(k_D^2\epsilon^{-2}\right)$ iterations and $\mathcal{O}\left(nk_D^2\epsilon^{-2}\right)$ function evaluations are obtained, matching the same complexity for the unconstrained case [20], with k_D being a problem dimension-dependent constant which define a fully linear model. Methods based on finite-difference gradient approximations may be used in this context. Such methods usually guarantee favorable worst-case evaluation complexity (e.g., $\mathcal{O}\left(n\epsilon^{-2}\right)$ in the unconstrained case [21]), but they can be inefficient in the presence of noise, unless carefully implemented [13, 39]. Furthermore, concerning the complexity bound it can be noted that the constant defining the $\mathcal{O}\left(\epsilon^{-2}\right)$ bound in [21] depends quadratically on the Lipschitz constant of the gradient whereas a linear dependence is typical for derivative-free methods. As discussed in [41], the Lipschitz constant of the gradient of the objective function can in turn depend exponentially on the dimension of the problem.

In the current paper, we analyze a line search method to solve problem (1). The algorithm under analysis is a modification of the one proposed in [34], equipped with a line search technique described in [8]. In particular, for any considered direction (i.e., a vector of the canonical basis), the line search technique first checks for a sufficient decrease in the objective function using a given stepsize. Then, if such a decrease is obtained, an extrapolation (or expansion) phase starts, where increasingly larger values of the stespize are tried until some conditions are met. This approach allows us to obtain complexity and identification results that extend those existing in the literature for direct-search [23] and model-based [27] methods. In particular, let us summarize the main contributions of this paper below.

- The first contribution of the current paper is providing a worst-case analysis for the proposed line search method, which yields to the same bounds for direct search [23], that is, $\mathcal{O}\left(n\epsilon^{-2}\right)$ iterations and $\mathcal{O}\left(n^2\epsilon^{-2}\right)$ function evaluations to produce the first point with a criticality measure below $\epsilon > 0$. Additionally, for the proposed algorithm, we are able to bound the *total* number of iterations where the criticality measure is not below ϵ, thus going beyond the complexity results for direct-search methods given in [23].

- The second contribution of the current paper is to show finite identification of the active constraints for the proposed line search method. Such a property is usually desirable for an optimization algorithm due to, among other things, the possibility of saving function evaluations if one recognizes the surfaces where a stationary points lies. Furthermore, in several applications, we might be interested only in the identification of the surface containing an optimal solution (or its support). In the literature, finite active-set identification was established for smooth optimization algorithms and proximal methods (see, e.g., [4, 6, 9, 26, 42]), also providing complexity bounds in some cases [7, 14, 37]. Moreover, many active-set approaches were designed for derivative-based optimization with bound constraints (see, e.g., [5, 16, 17, 25, 43, 44]).

  In a derivative-free setting, parameter-dependent estimates were used in [22, 32, 35], allowing for finite identification of active constraints if certain conditions hold. Moreover, finite identification results have been shown in [15] for a method using an inner approximation approach to minimize a function over the convex hull of a given set of vectors, meaning that, in finite time, the algorithm is able to identify, under appropriate assumptions, the vectors with zero weight in the convex combination representing the final solution. Also note that even though the proposed

algorithm, as highlighted above, gives the same complexity bound as the direct-search scheme analyzed in [23], the latter does not guarantee finite active-set identification as the iterates may get very close to the boundary but never lie at the boundary.

Here, we show that the proposed algorithm correctly identifies the active constraints satisfying the strict complementarity condition in a finite number of iterations, without using any parameter-dependent estimate. Namely, this feature is just an intrinsic property of the proposed algorithm. Such a result is obtained by using some tools from the analysis of derivative-based methods [14, 37]. More specifically, we define a measure which represents the minimum strict complementarity among the active constraints, thereby providing a neighborhood of the limit points where the active-set identification holds. Let us remark that also this identification result is obtained thanks to the extrapolation technique used in the line search procedure, which allows the stepsize to expand until we hit the border of the feasible set when we are in a neighborhood of a stationary point.

1.1 Notations

Given $l, u \in \mathbb{R}^n$, we denote $[l, u] = \{x \in \mathbb{R}^n : l_i \leq x_i \leq u_i, i = 1, \ldots, n\}$ the feasible set of problem (1). If $\mathcal{A} = \{a_1, \ldots, a_p\} \subset \mathbb{R}^n$ is a (finite) set of vectors, we denote

$$\text{cone}(\mathcal{A}) = \left\{ x \in \mathbb{R}^n : x = \sum_{i=1}^{p} \beta_i a_i, \ \beta_i \geq 0, \ i = 1, \ldots, p \right\}.$$

Given $v \in \mathbb{R}^n$ and a set $S \subseteq \mathbb{R}^n$, we denote by v_S the projection of v onto S. Given $\beta \in \mathbb{R}$, we indicate the sign of β by $\text{sign}(\beta)$, that is, $\text{sign}(\beta)$ is -1 if $\beta < 0$, 0 if $\beta = 0$ and 1 if $\beta > 0$. Finally, $\|v\|$ denotes the ℓ_2-norm of vector v.

The paper is organized as follows. In Section 2, we define a line search algorithm for the solution of problem (1). Section 3 is devoted to the analysis of the asymptotic convergence of the proposed algorithm, followed by the derivation of worst-case complexity bounds. In Section 4, we show finite active-set identification of the proposed algorithm. Finally, in Section 5, we draw some conclusions.

2 The algorithm

This section is concerned with the definition of a line search algorithm to solve problem (1). The proposed method, denoted as Algorithm 1, is inspired by the method proposed in [33] and uses some ideas from the one proposed in [8] for unconstrained problems.

At each iteration k, starting from the current iterate x_k, the algorithm sets $y_k^1 = x_k$ and explores the coordinate directions $\pm e_i$, $i = 1, \ldots, n$, using stepsizes $\nu_k^i = \max\{\tilde{\alpha}_k^i, c\Delta_k\}$, where

$$\Delta_k = \max_{i=1,\ldots,n} \{\tilde{\alpha}_k^i\},$$

and the quantities $\tilde{\alpha}_k^i$, $i = 1, \ldots, n$, are *tentative stepsizes* updated throughout the iterations. Then, the scheme produces *actual stepsizes* α_k^i to compute intermediate points $y_k^{i+1} = y_k^i + \alpha_k^i d_k^i$, with $d_k^i \in \{\pm e_i\}$, $i = 1, \ldots, n$. In particular, for any y_k^i, if $y_k^i + \nu_k^i d_k^i$ is infeasible (i.e., y_k^i is too close to a bound) or does not provide a sufficient decrease in the objective function, then we set $\alpha_k^i = 0$ (i.e., $y_k^{i+1} = y_k^i$). Otherwise, a sufficient decrease in the objective function is achieved by moving along d_k^i with a feasible stepsize α_k^i determined by a line search procedure which will be described later. Then, we set $x_{k+1} = y_k^{n+1}$ and prepare for the next iteration $k + 1$.

As a final note regarding our proposed scheme, we refer to k as a *successful iteration* if $x_{k+1} \neq x_k$, indicating that at least one positive stepsize α_k^i, $i = 1, \ldots, n$, has been computed. Conversely, we

refer to k as an *unsuccessful iteration* if $x_{k+1} = x_k$, that is, if $\alpha_k^i = 0$ for all $i = 1, \ldots, n$. Depending on whether an iteration k is successful or not, we use specific rules to update the tentative stepsizes for the next iteration $k + 1$. In more detail, for a successful iteration k, each $\tilde{\alpha}_{k+1}^i$ is set to α_k^i if the latter is positive, whereas $\tilde{\alpha}_{k+1}^i$ is set to ν_k^i otherwise. For an unsuccessful iteration k, each $\tilde{\alpha}_{k+1}^i$ is set to $\theta \nu_k^i$, with $\theta \in (0, 1)$.

Algorithm 1 Derivative-free line search algorithm

1: **given** $x_0 \in [l, u]$, $\theta \in (0, 1)$, $\delta \in (0, 1)$, $\gamma > 0$, $c \in (0, 1]$, $\tilde{\alpha}_0^i > 0$, $i = 1, \ldots, n$
2: **for** $k = 0, 1, \ldots$ **do**
3:     set $\Delta_k = \max_{i=1,\ldots,n}\{\tilde{\alpha}_k^i\}$
4:     set $y_k^1 = x_k$
5:     **for** $i = 1, \ldots, n$ **do**
6:         set $\nu_k^i = \max\{\tilde{\alpha}_k^i, c\Delta_k\}$
7:         compute d_k^i and α_k^i by the `line search`$(y_k^i, i, \gamma, \delta, \nu_k^i)$
8:         set $y_k^{i+1} = y_k^i + \alpha_k^i d_k^i$
9:     **end for**
10:    set $x_{k+1} = y_k^{n+1}$
11:    **if** $x_{k+1} \neq x_k$ **then**
12:        set $\tilde{\alpha}_{k+1}^i = \begin{cases} \alpha_k^i & \text{if } \alpha_k^i > 0 \\ \nu_k^i & \text{otherwise} \end{cases} \quad i = 1, \ldots, n$
13:    **else**
14:        set $\tilde{\alpha}_{k+1}^i = \theta \nu_k^i$, $i = 1, \ldots, n$
15:    **end if**
16: **end for**

Given a feasible point x, the exploration of the ith coordinate direction e_i is performed by a line search procedure outlined in Algorithm 2. First, we check if the given stepsize ν is feasible along $\pm e_i$, that is, if either $x + \nu e_i$ or $x - \nu e_i$ is feasible. If this is not the case, then we quit the line search returning a zero step length to indicate a failure. Otherwise, we try to determine if one between e_i and $-e_i$ is a "good" descent direction, that is, if a sufficient decrease in the objective function can be obtained by using a feasible stepsize. If neither e_i nor $-e_i$ qualifies as a suitable descent direction, then the line search procedure terminates, returning a zero step length to indicate a failure. Conversely, if a sufficient decrease of f is obtained, then an *extrapolation* (or *expansion*) phase starts (i.e., lines 12–15), where we try to increase the stepsize to the maximum extent while preserving feasibility and guaranteeing the sufficient decrease condition. Specifically, the *while* loop keeps expanding the stepsize as long as the most recently accepted point remains strictly within the bounds (i.e., $\alpha < \alpha_{\max}$) and the new tentative point is sufficiently better than the last accepted one (i.e., $f(x + \omega d) \leq f(x + \alpha d) - \gamma(\omega - \alpha)^2$).

Note that the proposed method is an adaptation of standard line search schemes with a few technical modifications introduced solely to obtain the desired complexity bounds. While numerical results illustrating the performance of line search variants are available in the literature [8, 33], here we focus on highlighting new theoretical properties about worst-case complexity and finite active-set identification of such a scheme.

3   Convergence and worst-case complexity

This section is devoted to the theoretical analysis of Algorithm 1.

Algorithm 2 line search(x,i,γ,δ,ν)

1: **if** $\nu > \max\{u_i - x_i, x_i - \ell_i\}$ **then** return $d = e_i$, $\alpha = 0$
2: **end if**
3: set $\bar{\alpha} = \nu$
4: **if** $\bar{\alpha} \le x_i - l_i$ and $f(x - \bar{\alpha}e_i) \le f(x) - \gamma\bar{\alpha}^2$ **then**
5:     set $d = -e_i$, $\alpha_{\max} = x_i - l_i$ and go to line 12
6: **end if**
7: **if** $\bar{\alpha} \le u_i - x_i$ and $f(x + \bar{\alpha}e_i) \le f(x) - \gamma\bar{\alpha}^2$ **then**
8:     set $d = e_i$, $\alpha_{\max} = u_i - x_i$ and go to line 12
9: **else**
10:     return d and $\alpha = 0$
11: **end if**
12: set $\alpha = \bar{\alpha}$ and $\omega = \min\{\alpha/\delta, \alpha_{\max}\}$
13: **while** $\left(\alpha < \alpha_{\max} \text{ and } f(x + \omega d) \le f(x + \alpha d) - \gamma(\omega - \alpha)^2\right)$ **do**
14:     set $\alpha = \omega$ and $\omega = \min\{\alpha/\delta, \alpha_{\max}\}$
15: **end while**
16: **return** d and α

3.1   Assumptions and preliminary results

First, let us define the following level set:

$$\mathcal{L}^0 = \{x \in \mathbb{R}^n \colon f(x) \le f(x_0)\}.$$

Throughout the paper, the following assumptions will be always considered satisfied, even if not explicitly invoked.

Assumption 1. *There exists an open convex set $\mathcal{S} \supseteq \mathcal{L}^0$ such that the objective function $f \colon \mathbb{R}^n \to \mathbb{R}$ is continuously differentiable with a Lipschitz continuous gradient ∇f with constant $L > 0$ over $\mathcal{S}$, i.e.,*

$$\|\nabla f(x) - \nabla f(y)\| \le L\|x - y\| \quad \forall x, y \in \mathcal{S}.$$

Moreover, f is bounded from below over $[l, u]$, i.e., a constant $f_{\min} \in \mathbb{R}$ exists such that

$$f_{\min} \le f(x) \quad \forall x \in [l, u].$$

Assumption 2. *A constant $M_g \ge 0$ exists such that*

$$\|\nabla f(x)\| \le M_g$$

for all $x \in \mathcal{L}^0$.

Under Assumption 1, we can also provide the following definition of coordinate-wise Lipschitz constants.

Definition 1. *The coordinate-wise Lipschitz constants $L_i > 0$, $i = 1, \dots, n$, of ∇f are such that, for all $x \in \mathcal{S}$,*

$$|\nabla_i f(x + se_i) - \nabla_i f(x)| \le L_i|s| \quad \forall s \in \mathbb{R} \colon x + se_i \in [l, u], \quad i = 1, \dots, n,$$

where the set $\mathcal{S}$ is defined as in Assumption 1. Moreover,

$$L^{max} = \max_{i=1,\dots,n} L_i. \tag{2}$$

Now, given $x \in [l, u]$, let us introduce the following criticality measure:

$$\chi(x) = \max_{\substack{x+d \in [l,u] \\ \|d\| \leq 1}} -\nabla f(x)^\top d.$$

The above measure has been successfully used in the analysis of some direct search methods for linearly constrained problems [23, 28, 29]. It can be interpreted as the progress on a first-order model in a ball centered at x with unit radius subject to feasibility constraints [12, 29], thus generalizing $\|\nabla f(x)\|$ from the unconstrained setting. Originally proposed in [11] for more general constraints and further analyzed in [12], $\chi(x)$ is continuous, non-negative and such that $\chi(x) = 0$ if and only if x is a KKT point. So, we can define a stationary point as follows.

Definition 2. *A point $x^* \in [l, u]$ is said to be a stationary point of problem (1) if $\chi(x^*) = 0$.*

Next, given $\epsilon \geq 0$, we define the set of ϵ-active constraints at $x \in [l, u]$ as

$$I_l(x, \epsilon) = \{i : x_i \leq l_i + \epsilon\},$$
$$I_u(x, \epsilon) = \{i : x_i \geq u_i - \epsilon\}.$$

Namely, $I_l(x, \epsilon)$ and $I_u(x, \epsilon)$ denote the sets of lower and upper bound constraints, respectively, that are nearly active at x with a tolerance ϵ. Accordingly, let us define $N(x, \epsilon)$ as the ϵ-normal cone generated by the ϵ-active constraints, that is,

$$N(x, \epsilon) = \text{cone}\left(\{-e_i, i \in I_l(x, \epsilon)\} \cup \{e_i, i \in I_u(x, \epsilon)\} \cup \{0\}\right),$$

while the ϵ-tangent cone $T(x, \epsilon)$ is the polar of $N(x, \epsilon)$, that is,

$$T(x, \epsilon) = N(x, \epsilon)^\circ = \{d \in \mathbb{R}^n : d^\top v \leq 0, \ \forall \ v \in N(x, \epsilon)\}.$$

The use of ϵ-normal and ϵ-tangent cones is a well known tool in the analysis of direct search methods applied to linearly constrained problems [23, 28, 29]. Essentially, the set $x + T(x, \epsilon)$ is an approximation of the feasible region near a feasible point x, that is, moving from x along any direction in $T(x, \epsilon)$ with a stepsize less than or equal to ϵ ensures that all constraints stay satisfied.

In our case, considering the structure of the feasible set of problem (1), it is straightforward to verify that a set of generators for $T(x, \epsilon)$ is given by

$$G_{T(x,\epsilon)} = \{-e_i, i \notin I_l(x, \epsilon)\} \cup \{e_i, i \notin I_u(x, \epsilon)\} \cup \{0\}, \tag{3}$$

that is, $T(x, \epsilon) = \text{cone}\,(G_{T(x,\epsilon)})$.

The following two propositions from [29] show how $\chi(x_k)$ can be upper bounded by using of the projection of $-\nabla f(x_k)$ onto $T(x_k, \epsilon)$ and $N(x_k, \epsilon)$.

Proposition 1 ([28, Proposition 8.2]). *If $x \in [l, u]$, then for all $\epsilon \geq 0$ we have that*

$$\chi(x) \leq \|(-\nabla f(x))_{T(x,\epsilon)}\| + \epsilon\sqrt{n}\|(-\nabla f(x))_{N(x,\epsilon)}\|.$$

Proposition 2 ([28, Proposition 8.1]). *Given $\epsilon \geq 0$, let $G_{T(x,\epsilon)}$ be defined as in (3). If $(-\nabla f(x))_{T(x,\epsilon)} \neq 0$, then there exists $d \in G_{T(x,\epsilon)}$ such that*

$$\frac{1}{\sqrt{n}}\|(-\nabla f(x))_{T(x,\epsilon)}\| \leq -\nabla f(x)^\top d.$$

In the convergence analysis of Algorithm 1, Propositions 1–2 will allow us to relate $\chi(x_k)$ with Δ_{k+1} for every iteration k (see Theorem 2 below). In particular, this will be obtained by applying the above results with $\epsilon = \Delta_k$ and using the following relation between Δ_k and Δ_{k+1}.

Lemma 1. *Let $\{\Delta_k\}$ be the sequence of maximum tentative stepsizes produced by Algorithm 1. Then, for all k we have*

$$\Delta_{k+1} \begin{cases} \geq \Delta_k & \text{if } x_{k+1} \neq x_k \\ = \theta\Delta_k & \text{if } x_{k+1} = x_k. \end{cases}$$

Therefore, $\Delta_k \leq \Delta_{k+1}/\theta$ for all k.

Proof. First, let us consider the case where $x_{k+1} \neq x_k$ (i.e., k is a successful iteration). In this case, the algorithm sets $\tilde{\alpha}_{k+1}^i = \alpha_k^i$ if $\alpha_k^i > 0$ and $\tilde{\alpha}_{k+1}^i = \nu_k^i$ if $\alpha_k^i = 0$. Since, from the line search procedure, $\alpha_k^i \geq \nu_k^i$ for all $i = 1, \ldots, n$, we can write

$$\tilde{\alpha}_{k+1}^i \geq \nu_k^i \geq \tilde{\alpha}_k^i, \quad i = 1, \ldots, n,$$

where the last inequality follows from the definition of ν_k^i. Then, using the definition of Δ_k, we get $\Delta_{k+1} \geq \Delta_k$.

Now, let us consider the case where $x_{k+1} = x_k$ (i.e., k is an unsuccessful iteration). In this case, the algorithm sets $\tilde{\alpha}_{k+1}^i = \theta\nu_k^i = \theta\max\{\tilde{\alpha}_k^i, c\Delta_k\}$. Namely, for all $i = 1, \ldots, n$, we have that

$$\tilde{\alpha}_{k+1}^i = \begin{cases} \theta\tilde{\alpha}_k^i & \text{if } \tilde{\alpha}_k^i \geq c\Delta_k, \\ \theta c\Delta_k & \text{if } \tilde{\alpha}_k^i < c\Delta_k. \end{cases} \tag{4}$$

From the definition of Δ_k and the fact that $c \in (0, 1]$, it follows that $\tilde{\alpha}_{k+1}^i \leq \theta\Delta_k$ for all $i = 1, \ldots, n$, implying that

$$\Delta_{k+1} = \max_{i=1,\ldots,n} \tilde{\alpha}_{k+1}^i \leq \theta\Delta_k. \tag{5}$$

Now, let $\bar{\imath} \in \{1, \ldots, n\}$ be such that $\tilde{\alpha}_k^{\bar{\imath}} = \Delta_k$. Since $c \in (0, 1]$, we have $\tilde{\alpha}_k^{\bar{\imath}} \geq c\Delta_k$ and then, recalling (4), we get $\tilde{\alpha}_{k+1}^{\bar{\imath}} = \theta\tilde{\alpha}_k^{\bar{\imath}} = \theta\Delta_k$. It follows from (5) that

$$\Delta_{k+1} = \tilde{\alpha}_{k+1}^{\bar{\imath}} = \theta\Delta_k,$$

which concludes the proof. □

3.2 Global convergence

In the following proposition, we show that Algorithm 1 and Algorithm 2 are well defined, i.e., Algorithm 2 cannot cycle so that Algorithm 1 produces infinite sequences of points and stepsizes.

Proposition 3. *Algorithm 1 is well defined, i.e., it produces infinite sequences $\{x_k\}$, $\{\alpha_k^i\}$, $\{\tilde{\alpha}_k^i\}$, $i = 1, \ldots, n$.*

Proof. To prove that Algorithm 1 is well defined, we have to show that the line search procedure cannot infinitely cycle over steps 13–15 in the while loop. Let us suppose, by contradiction, that the while loop does not terminate, i.e., we always have $\alpha < \alpha_{\max}$. At the j-th iteration of the while loop, we have

$$\alpha = \frac{\bar{\alpha}}{\delta^j}$$

with $\delta < 1$. If $\alpha_{\max}$ is finite, then we have $\alpha > \alpha_{\max}$, for j sufficiently large, which is a contradiction. Otherwise, if $\alpha_{\max} = +\infty$, then, for every j we have

$$f(x + \omega d) \leq f(x + \alpha d) - \gamma(\omega - \alpha)^2, \tag{6}$$

with $\omega = \alpha/\delta = \bar{\alpha}/\delta^{j+1}$. Now, for j sufficiently large, (6) contradicts Assumption 1, i.e., that f is bounded from below on the feasible set. □

In the following, we present some results concerning the global convergence of Algorithm 1 to stationary points. More specifically, by extending some results from [8, 33], we establish a relationship between $\nabla f(x_k)$, with specific directions d, and the largest tentative step length at Δ_{k+1}. We will show that the bound depends, besides on the problem dimension n, on the Lipschitz constant L and the algorithm parameters γ, θ and δ.

From now on, let us denote

$$T_k := T(x_k, \Delta_k) \quad \text{and} \quad G_k := G(x_k, \Delta_k), \tag{7}$$

where G_k is the set of generators of T_k and is defined as in (3).

Theorem 1. *Let $\{x_k\}$ be the sequence produced by Algorithm 1. Then, for all k and for all $d \in G_k$, we have that*

$$-\nabla f(x_k)^\top d \leq \begin{cases} \left(\dfrac{\gamma + L}{\delta} + L\sqrt{n}\right)\Delta_{k+1} & \text{if } x_{k+1} \neq x_k, \\[3mm] \left(\dfrac{\gamma + L^{max}}{\theta}\right)\Delta_{k+1} & \text{if } x_{k+1} = x_k. \end{cases} \tag{8}$$

Proof. For all k, the result trivially holds for $d = 0$. Now, consider an iteration k such that $x_{k+1} \neq x_k$ (i.e., a successful iteration). The following cases can occur, recalling that the analysis is limited to considering directions $\pm e_i$, $i = 1, \ldots, n$, belonging to G_k.

(1a) $e_i \in G_k$, $(y_k^{i+1})_i = l_i$ and $(y_k^i)_i = l_i$. Then, $\alpha_k^i = 0$ and $\tilde{\alpha}_{k+1}^i = \nu_k^i$. From the instructions of the line search procedure, we have that

$$f(y_k^i + \tilde{\alpha}_{k+1}^i e_i) > f(y_k^i) - \gamma(\tilde{\alpha}_{k+1}^i)^2.$$

By the mean value theorem, we have

$$f(y_k^i + \tilde{\alpha}_{k+1}^i e_i) - f(y_k^i) = \tilde{\alpha}_{k+1}^i \nabla_i f(\xi_k^i),$$

where $\xi_k^i = y_k^i + t_k^i \tilde{\alpha}_{k+1}^i e_i$ and $t_k^i \in (0,1)$. Then,

$$-\nabla_i f(\xi_k^i) < \gamma \tilde{\alpha}_{k+1}^i.$$

It follows that

$$-\nabla_i f(\xi_k^i) + \nabla_i f(x_k) - \nabla_i f(x_k) < \gamma \tilde{\alpha}_{k+1}^i$$

and we can write

$$\begin{aligned} -\nabla_i f(x_k) &< \gamma \tilde{\alpha}_{k+1}^i - (\nabla_i f(x_k) - \nabla_i f(\xi_k^i)) \\ &\leq \gamma \tilde{\alpha}_{k+1}^i + L\|x_k - \xi_k^i\| \\ &\leq \gamma \tilde{\alpha}_{k+1}^i + L\|x_k - y_k^i\| + L\|y_k^i - \xi_k^i\|. \end{aligned} \tag{9}$$

Moreover, since $t_k^i \in (0,1)$, we have $t_k^i \tilde{\alpha}_{k+1}^i \leq \tilde{\alpha}_{k+1}^i$. Then,

$$\|y_k^i - \xi_k^i\| \leq \tilde{\alpha}_{k+1}^i.$$

Hence, since $\alpha_{k+1}^i \leq \tilde{\alpha}_{k+1}^i \leq \Delta_{k+1}$ for all $i = 1, \ldots, n$, and taking into account that $y_k^i = x_k + \sum_{j=1}^{i-1} \alpha_k^j d_k^j$, so that $\|x_k - y_k^i\| \leq \sqrt{n}\Delta_{k+1}$, we get

$$(-\nabla f(x_k))^\top e_i < \gamma \tilde{\alpha}_{k+1}^i + L\|x_k - y_k^i\| + L\tilde{\alpha}_{k+1}^i \leq \left(\gamma + L + L\sqrt{n}\right)\Delta_{k+1}. \tag{10}$$

(1b) $-e_i \in G_k$, $(y_k^{i+1})_i = l_i$ and $(y_k^i)_i > l_i$. Then, $\tilde{\alpha}_{k+1}^i = \alpha_k^i > 0$. From the instructions of the line search procedure, there exists $\beta_k^i \in \{0, \delta \tilde{\alpha}_{k+1}^i\}$ such that

$$f(y_k^i - \tilde{\alpha}_{k+1}^i e_i) \leq f(y_k^i - \beta_k^i e_i) - \gamma(\tilde{\alpha}_{k+1}^i - \beta_k^i)^2 \leq f(y_k^i - \beta_k^i e_i).$$

By the mean value theorem, we have

$$f(y_k^i - \tilde{\alpha}_{k+1}^i e_i) - f(y_k^i - \beta_k^i e_i) = -(\tilde{\alpha}_{k+1}^i - \beta_k^i)\nabla_i f(\xi_k^i),$$

where $\xi_k^i = y_k^i - \beta_k^i e_i + t_k^i(\beta_k^i - \tilde{\alpha}_{k+1}^i)e_i$ and $t_k^i \in (0,1)$. Then,

$$-\nabla_i f(\xi_k^i) \leq 0.$$

Moreover, from the definition of β_k^i and the fact that $t_k^i \in (0,1)$, we have $(1-t_k^i)\beta_k^i \leq (1-t_k^i)\tilde{\alpha}_{k+1}^i$. Then,

$$\|y_k^i - \xi_k^i\| = (1 - t_k^i)\beta_k^i + t_k^i \tilde{\alpha}_{k+1}^i \leq \tilde{\alpha}_{k+1}^i.$$

Hence, by similar reasoning as in case (1a), we obtain

$$(-\nabla f(x_k))^\top(-e_i) \leq \left(L + L\sqrt{n}\right)\Delta_{k+1}. \tag{11}$$

(2a) $-e_i \in G_k$, $(y_k^{i+1})_i = u_i$ and $(y_k^i)_i = u_i$. Then, $\alpha_k^i = 0$ and $\tilde{\alpha}_{k+1}^i = \nu_k^i$. Reasoning as in case (1a), with minor differences, we get

$$(-\nabla f(x_k))^\top(-e_i) \leq \left(\gamma + L + L\sqrt{n}\right)\Delta_{k+1}. \tag{12}$$

(2b) $e_i \in G_k$, $(y_k^{i+1})_i = u_i$ and $(y_k^i)_i < u_i$. Then, $\tilde{\alpha}_{k+1}^i = \alpha_k^i > 0$. Reasoning as in case (1b), with minor differences, we get

$$(-\nabla f(x_k))^\top e_i \leq \left(L + L\sqrt{n}\right)\Delta_{k+1}. \tag{13}$$

(3a) $\{\pm e_i\} \cap G_k \neq \emptyset$, $l_i < (y_k^{i+1})_i < u_i$ and $y_k^i = y_k^{i+1}$. Then, $\alpha_k^i = 0$ and $\tilde{\alpha}_{k+1}^i = \nu_k^i$. From the instructions of the line search procedure, we have that

$$f(y_k^i + \tilde{\alpha}_{k+1}^i e_i) > f(y_k^i) - \gamma(\tilde{\alpha}_{k+1}^i)^2 \quad \text{if } e_i \in G_k,$$
$$f(y_k^i - \tilde{\alpha}_{k+1}^i e_i) > f(y_k^i) - \gamma(\tilde{\alpha}_{k+1}^i)^2 \quad \text{if } -e_i \in G_k.$$

By the mean value theorem, we have

$$f(y_k^i + \tilde{\alpha}_{k+1}^i e_i) - f(y_k^i) = \nabla_i f(\xi_k^i)\tilde{\alpha}_{k+1}^i \quad \text{if } e_i \in G_k,$$
$$f(y_k^i - \tilde{\alpha}_{k+1}^i e_i) - f(y_k^i) = -\nabla_i f(\bar{\xi}_k^i)\tilde{\alpha}_{k+1}^i \quad \text{if } -e_i \in G_k.$$

where $\xi_k^i = y_k^i + t_k^i \tilde{\alpha}_{k+1}^i e_i$ and $\bar{\xi}_k^i = y_k^i - \bar{t}_k^i \tilde{\alpha}_{k+1}^i e_i$, with $t_k^i, \bar{t}_k^i \in (0,1)$. Then,

$$-\nabla_i f(\xi_k^i) < \gamma \tilde{\alpha}_{k+1}^i \quad \text{if } e_i \in G_k,$$
$$\nabla_i f(\bar{\xi}_k^i) < \gamma \tilde{\alpha}_{k+1}^i \quad \text{if } -e_i \in G_k.$$

and, recalling that $t_k^i, \bar{t}_k^i \in (0,1)$, we also have

$$\|y_k^i - \xi_k^i\| = t_k^i \tilde{\alpha}_{k+1}^i \leq \tilde{\alpha}_{k+1}^i \quad \text{if } e_i \in G_k,$$
$$\|y_k^i - \bar{\xi}_k^i\| = t_k^i \tilde{\alpha}_{k+1}^i \leq \tilde{\alpha}_{k+1}^i \quad \text{if } -e_i \in G_k.$$

Hence, reasoning as in case (1a) (i.e., using (9) for e_i and applying minor changes to (9), with ξ_k^i replaced by $\bar{\xi}_k^i$, for $-e_i$), we obtain

$$-\nabla f(x_k)^\top e_i \leq \left(\gamma + L + L\sqrt{n}\right)\Delta_{k+1} \quad \text{if } e_i \in G_k,$$
$$-\nabla f(x_k)^\top(-e_i) \leq \left(\gamma + L + L\sqrt{n}\right)\Delta_{k+1} \quad \text{if } -e_i \in G_k. \tag{14}$$

9

(3b) $d_k^i \in G_k$, $l_i < (y_k^{i+1})_i < u_i$ and $y_k^i \neq y_k^{i+1}$. Then, $\alpha_k^i > 0$, $\tilde{\alpha}_{k+1}^i = \alpha_k^i$. Let us assume that $d_k^i = -e_i$, i.e., $y_k^{i+1} = y_k^i - \tilde{\alpha}_{k+1}^i e_i$ (the proof for the case $d_k^i = e_i$ is identical, except for minor changes). Then, from the instructions of the line search procedure, there exist β_k^i and $\bar{\beta}_k^i$ such that $\beta_k^i \in \{0, \delta\tilde{\alpha}_{k+1}^i\}$, $\tilde{\alpha}_{k+1}^i < \bar{\beta}_k^i \leq \tilde{\alpha}_{k+1}^i/\delta$ and

$$f(y_k^i - \tilde{\alpha}_{k+1}^i e_i) \leq f(y_k^i - \beta_k^i e_i) - \gamma(\tilde{\alpha}_{k+1}^i - \beta_k^i)^2 \leq f(y_k^i - \beta_k^i e_i),$$
$$f\left(y_k^i - \bar{\beta}_k^i e_i\right) > f(y_k^i - \tilde{\alpha}_{k+1}^i e_i) - \gamma\left(\bar{\beta}_k^i - \tilde{\alpha}_{k+1}^i\right)^2.$$

By the mean value theorem, we have

$$f(y_k^i - \tilde{\alpha}_{k+1}^i e_i) - f(y_k^i - \beta_k^i e_i) = -(\tilde{\alpha}_{k+1}^i - \beta_k^i)\nabla_i f(\xi_k^i),$$
$$f\left(y_k^i - \bar{\beta}_k^i e_i\right) - f(y_k^i - \tilde{\alpha}_{k+1}^i e_i) = -(\bar{\beta}_k^i - \tilde{\alpha}_{k+1}^i)\nabla_i f(\bar{\xi}_k^i),$$

where $\xi_k^i = y_k^i - \beta_k^i e_i - t_k^i(\tilde{\alpha}_{k+1}^i - \beta_k^i)e_i$ and $\bar{\xi}_k^i = y_k^i - \tilde{\alpha}_{k+1}^i e_i - \bar{t}_k^i(\bar{\beta}_k^i - \tilde{\alpha}_{k+1}^i)e_i$, with $t_k^i, \bar{t}_k^i \in (0,1)$. Then,

$$-\nabla_i f(\xi_k^i) \leq 0,$$
$$\nabla_i f(\bar{\xi}_k^i) < \gamma(\bar{\beta}_k^i - \tilde{\alpha}_{k+1}^i).$$

Moreover, from the definition of $\bar{\beta}_k^i$, it follows that $0 \leq \bar{\beta}_k^i - \tilde{\alpha}_{k+1}^i \leq (1/\delta - 1)\tilde{\alpha}_{k+1}^i \leq \tilde{\alpha}_{k+1}^i/\delta$, where we have used the fact that $\delta \in (0,1)$. Then,

$$-\nabla_i f(\xi_k^i) \leq 0,$$
$$\nabla_i f(\bar{\xi}_k^i) < \gamma\left(\frac{1-\delta}{\delta}\right)\tilde{\alpha}_{k+1}^i \leq \gamma\frac{\tilde{\alpha}_{k+1}^i}{\delta},$$

and, recalling that $t_k^i, \bar{t}_k^i \in (0,1)$, we also have

$$\|y_k^i - \xi_k^i\| = \beta_k^i + t_k^i(\tilde{\alpha}_{k+1}^i - \beta_k^i) \leq \tilde{\alpha}_{k+1}^i \leq \frac{\tilde{\alpha}_{k+1}^i}{\delta},$$

$$\|y_k^i - \bar{\xi}_k^i\| = \tilde{\alpha}_{k+1}^i + t_k^i(\bar{\beta}_k^i - \tilde{\alpha}_{k+1}^i) \leq \bar{\beta}_k^i \leq \frac{\tilde{\alpha}_{k+1}^i}{\delta}.$$

Hence, reasoning as in the previous case, we obtain

$$-\nabla f(x_k)^\top(-e_i) = \nabla_i f(x_k) \leq \left(\frac{\gamma + L}{\delta} + L\sqrt{n}\right)\Delta_{k+1},$$
$$-\nabla f(x_k)^\top e_i = -\nabla_i f(x_k) \leq \left(\frac{\gamma + L}{\delta} + L\sqrt{n}\right)\Delta_{k+1}. \tag{15}$$

Hence, from (10), (11), (12), (13), (14) and (15), we conclude that

$$-\nabla f(x_k)^\top d \leq \left(\frac{\gamma + L}{\delta} + L\sqrt{n}\right)\Delta_{k+1} \quad \forall d \in G_k.$$

Now, let us analyze an iteration k such that $x_{k+1} = x_k$ (i.e., an unsuccessful iteration). In such a case, only cases (1a), (2a) and (3a) can occur, although we have to consider that $y_k^i = x_k$ and $\tilde{\alpha}_{k+1}^i = \theta\nu_k^i$. Hence, replacing L with L_i, we have that the following relations hold:

$$-\nabla f(x_k)^\top e_i \leq \frac{\gamma + L^{\max}}{\theta}\Delta_{k+1} \quad \text{if } e_i \in G_k,$$
$$-\nabla f(x_k)^\top(-e_i) \leq \frac{\gamma + L^{\max}}{\theta}\Delta_{k+1} \quad \text{if } -e_i \in G_k.$$

As above, we finally conclude

$$-\nabla f(x_k)^\top d \leq \frac{\gamma + L^{\max}}{\theta} \Delta_{k+1} \quad \forall d \in G_k.$$

$\square$

Remark 1. *The result expressed in Theorem 1 strongly relies on the extrapolation phase of the line search procedure (i.e., lines 12–15 of Algorithm 2). In particular, since we quit the expansion with a failure in the objective decrease when we do not hit the border of the feasible set, then we are able to upper bound $\chi(x_k)$ for all iterations, including the successful ones. This represents a relevant difference over direct-search [23] methods, where $\chi(x_k)$ is usually upper bounded only for the unsuccessful iterations. Moreover, this property will allow us to give a complexity bound on the total number of iterations where $\chi(x_k)$ is above a prespecified threshold (see Theorem 5 below).*

Combining Proposition 2 and Theorem 1, it is now straightforward to relate $\chi(x_k)$ with Δ_{k+1}, as stated in the following result.

Theorem 2. *Let $\{x_k\}$ be the sequence of points produced by Algorithm 1. Then,*

$$\chi(x_k) \leq \begin{cases} \sqrt{n}\left(\dfrac{\gamma + L}{\delta} + L\sqrt{n} + \dfrac{M_g}{\theta}\right)\Delta_{k+1} & \text{if } x_{k+1} \neq x_k, \\[2ex] \sqrt{n}\left(\dfrac{\gamma + L^{max} + M_g}{\theta}\right)\Delta_{k+1} & \text{if } x_{k+1} = x_k. \end{cases}$$

Proof. Using Proposition 1 with $\epsilon = \Delta_k$ and Lemma 1, we obtain

$$\chi(x_k) \leq \|(-\nabla f(x_k))_{T_k}\| + \frac{\Delta_{k+1}}{\theta}\sqrt{n}M_g,$$

where we have used the fact that $\|(-\nabla f(x_k))_{N(x_k,\epsilon)}\| \leq \|\nabla f(x_k)\| \leq M_g$, where the last inequality follows from Assumption 2. So, using Proposition 2, it follows that there exists a direction $d \in G_k$ such that

$$\chi(x_k) \leq \sqrt{n}\left(-\nabla f(x_k)^\top d + \frac{M_g}{\theta}\Delta_{k+1}\right).$$

The desired result hence follows from Theorem 1. $\square$

In order to get convergence to stationary points and provide worst-case complexity bounds for the proposed algorithm, for each iteration k let us define

$$\Phi_k = f(x_k) + \eta\Delta_k^2, \tag{16}$$

where η satisfies

$$0 < \eta < \gamma(\delta(1-\delta))^2. \tag{17}$$

Recalling Assumption 1, note that

$$\Phi_k \geq f_{\min} \quad \forall k \geq 0. \tag{18}$$

Now, in the next theorem, we bound the difference $\Phi_k - \Phi_{k-1}$ for each iteration k.

Theorem 3. *Let $\{x_k\}$ be the sequence produced by Algorithm 1. Then,*

$$\Phi_k - \Phi_{k-1} \leq -c_1\Delta_k^2 \quad \forall k \geq 1, \tag{19}$$

with

$$c_1 = \min\left\{\gamma c^2, \gamma(\delta(1-\delta))^2 - \eta, \eta\left(\frac{1-\theta^2}{\theta^2}\right)\right\} > 0, \tag{20}$$

where c and θ are defined in Algorithm 1.

11

Proof. For each iteration $k \geq 1$, consider the iteration $k - 1$. The following cases can occur.

- $x_k \neq x_{k-1}$ (i.e., $k - 1$ is a successful iteration) and $\Delta_k = \Delta_{k-1}$.

  Hence, there exists a coordinate $\bar{\imath}$ such that we have moved from x_{k-1} along $\pm e_{\bar{\imath}}$ and we have

  $$\alpha_{k-1}^{\bar{\imath}} \geq \nu_{k-1}^{\bar{\imath}} = \max\left\{\tilde{\alpha}_{k-1}^{\bar{\imath}}, c\Delta_{k-1}\right\} \geq c\Delta_{k-1} = c\Delta_k.$$

  Then, from the line search procedure, we can write

  $$f(x_k) \leq f(x_{k-1}) - \gamma c^2 \Delta_k^2.$$

  Hence, we have

  $$\Phi_k - \Phi_{k-1} = f(x_k) - f(x_{k-1}) + \eta\left(\Delta_k^2 - \Delta_{k-1}^2\right) \leq -\gamma c^2 \Delta_k^2. \tag{21}$$

- $x_k \neq x_{k-1}$ (i.e., $k - 1$ is a successful iteration) and $\Delta_k > \Delta_{k-1}$. Hence, there exists a coordinate $\bar{\imath}$ such that

  $$\alpha_{k-1}^{\bar{\imath}} = \Delta_k.$$

  Let h_{k-1}^i be the number of times the stepsize related to the ith coordinate is expanded on iteration $k - 1$, that is, if $\alpha_k^i > 0$

  $$\alpha_{k-1}^i = \min\{\alpha_{\max}, \delta^{-h_{k-1}^i}\nu_{k-1}^i\},$$

  where $\alpha_{\max}$ is the distance to the bound of the ith coordinate over d_{k-1}^i. Now, considering $\bar{\imath}$, we have

  $$\alpha_{k-1}^{\bar{\imath}} > \Delta_{k-1} \geq \nu_{k-1}^{\bar{\imath}}.$$

  In particular, the first inequality implies that $h_{k-1}^{\bar{\imath}} \geq 1$ (otherwise we would have $\alpha_{k-1}^{\bar{\imath}} \in \{0, \nu_{k-1}^{\bar{\imath}}\}$). If $h_{k-1}^{\bar{\imath}} = 1$, then we have

  $$\Delta_k = \alpha_{k-1}^{\bar{\imath}} = \min\{\alpha_{\max}, \delta^{-1}\nu_{k-1}^{\bar{\imath}}\} \leq \delta^{-1}\nu_{k-1}^{\bar{\imath}},$$

  that is,

  $$\nu_{k-1}^{\bar{\imath}} \geq \delta\Delta_k.$$

  Thus, we get

  $$f(x_k) \leq f(x_{k-1}) - \gamma(\nu_{k-1}^{\bar{\imath}})^2 \leq f(x_{k-1}) - \gamma\delta^2\Delta_k^2. \tag{22}$$

  Let us now consider $h_{k-1}^{\bar{\imath}} \geq 2$. We have

  $$\Delta_k = \alpha_{k-1}^{\bar{\imath}} = \min\{\alpha_{\max}, \delta^{-h_{k-1}^{\bar{\imath}}}\nu_{k-1}^{\bar{\imath}}\} \leq \delta^{-h_{k-1}^{\bar{\imath}}}\nu_{k-1}^{\bar{\imath}},$$

  that is,

  $$\delta^{-(h_{k-1}^{\bar{\imath}}-1)}\nu_{k-1}^{\bar{\imath}} \geq \delta\Delta_k. \tag{23}$$

  Therefore, we can write

  $$\begin{aligned}
  f(x_k) &\leq f(x_{k-1}) - \gamma\left(\delta^{-(h_{k-1}^{\bar{\imath}}-1)}\nu_{k-1}^{\bar{\imath}} - \delta^{-(h_{k-1}^{\bar{\imath}}-2)}\nu_{k-1}^{\bar{\imath}}\right)^2 \\
  &= f(x_{k-1}) - \gamma\left(\delta^{-(h_{k-1}^{\bar{\imath}}-1)}\nu_{k-1}^{\bar{\imath}}(1-\delta)\right)^2 \\
  &\leq f(x_{k-1}) - \gamma\left(\delta(1-\delta)\right)^2\Delta_k^2,
  \end{aligned} \tag{24}$$

where the last inequality follows from (23). Finally, using (22) and (24), and considering that $\min\{(1-\delta)^2, (\delta(1-\delta))^2\} = (\delta(1-\delta))^2$, we can write

$$f(x_k) \leq f(x_{k-1}) - \gamma(\delta(1-\delta))^2 \Delta_k^2.$$

Hence, we have

$$\begin{aligned}
\Phi_k - \Phi_{k-1} &= f(x_k) - f(x_{k-1}) + \eta \left(\Delta_k^2 - \Delta_{k-1}^2 \right) \\
&\leq -\gamma(\delta(1-\delta))^2 \Delta_k^2 + \eta \Delta_k^2 \\
&= -\left(\gamma(\delta(1-\delta))^2 - \eta \right) \Delta_k^2.
\end{aligned} \tag{25}$$

- $x_k = x_{k-1}$ (i.e., $k-1$ is an unsuccessful iteration). By Lemma 1, we have

$$\Delta_k = \theta \Delta_{k-1}.$$

Then,

$$\begin{aligned}
\Phi_k - \Phi_{k-1} &= f(x_k) - f(x_{k-1}) + \eta \left(\Delta_k^2 - \Delta_{k-1}^2 \right) \\
&= \eta \left(\Delta_k^2 - \frac{1}{\theta^2} \Delta_k^2 \right) \\
&= -\eta \left(\frac{1-\theta^2}{\theta^2} \right) \Delta_k^2.
\end{aligned} \tag{26}$$

Finally, we get (19) by combining (21), (25) and (26), where $c_1 > 0$ since $c \in (0,1]$. $\qquad\square$

Using the above result, we can easily show the convergence to zero of the sequences of tentative and actual stepsizes produced by the algorithm, i.e., $\{\tilde{\alpha}_k^i\}$, $\{\alpha_k^i\}$, $i = 1, \ldots, n$, respectively, together with the convergence to zero of the sequence of maximum tentative stepsizes $\{\Delta_k\}$.

Proposition 4. *Let* $\{\tilde{\alpha}_k^i\}$, $\{\alpha_k^i\}$, $i = 1, \ldots, n$, *and* $\{\Delta_k\}$ *be the sequences produced by Algorithm 1. We have that*

$$\lim_{k \to \infty} \tilde{\alpha}_k^i = 0, \quad i = 1, \ldots, n; \tag{27}$$

$$\lim_{k \to \infty} \alpha_k^i = 0, \quad i = 1, \ldots, n; \tag{28}$$

$$\lim_{k \to \infty} \Delta_k = 0. \tag{29}$$

Proof. From (18) and (19), we get (29). Then, using the definition of Δ_k given in Algorithm 1, also (27) holds. Since, for all $k \geq 1$ and for all $i = 1, \ldots, n$, from the instructions of the algorithm either $\alpha_{k-1}^i = 0$ or $\alpha_{k-1}^i = \tilde{\alpha}_k^i$, then (28) follows from (27). $\qquad\square$

Using Theorem 2 and Proposition 4, it is now possible to prove the convergence of the algorithm to stationary points.

Theorem 4. *Let* $\{x_k\}$ *be the sequence of points produced by Algorithm 1. Then,*

- $\lim_{k \to \infty} \chi(x_k) = 0$, *i.e., every limit point of* $\{x_k\}$ *is stationary,*

- $\lim_{k \to \infty} \|x_{k+1} - x_k\| = 0.$

Proof. From Theorem 2 and (29) in Proposition 4, taking the limit for $k \to \infty$ it follows that $\chi(x_k) \to 0$, that is, every limit point of $\{x_k\}$ is stationary. Finally, since $x_{k+1} - x_k = \sum_{i=1}^{n} \alpha_k^i d_k^i$, using (28) in Proposition 4 we also have that $\|x_{k+1} - x_k\| \to 0$. $\qquad\square$

3.3 Worst-case complexity

This section is devoted to analyze the worst-case complexity of Algorithm 1. In particular,

(i) we give an upper bound of $\mathcal{O}\left(n^2\epsilon^{-2}\right)$ on the total number of iterations where $\chi(x_k)$ is not below a prespecified threshold ϵ;

(ii) we give an upper bound of $\mathcal{O}\left(n\epsilon^{-2}\right)$ on the number of iterations required to generate the first point x_k where $\chi(x_k)$ is below a prespecified tolerance ϵ;

(iii) we give an upper bound of $\mathcal{O}\left(n^2\epsilon^{-2}\right)$ on the number of function evaluations required to generate the first point x_k where $\chi(x_k)$ is below a prespecified tolerance ϵ.

We start by providing an upper bound of $\mathcal{O}\left(n^2\epsilon^{-2}\right)$ on the total number of iterations where $\chi(x_k) \geq \epsilon$, with a given $\epsilon > 0$.

Theorem 5. *Let $\{x_k\}$ be the sequence of points produced by the algorithm. Given any $\epsilon > 0$, let*

$$K_\epsilon = \{k \colon \chi(x_k) \geq \epsilon\}.$$

Then, $|K_\epsilon| \leq \mathcal{O}\left(n^2\epsilon^{-2}\right)$. In particular,

$$|K_\epsilon| \leq \left\lfloor \frac{c_2^2(\Phi_0 - f_{\min})}{c_1}\epsilon^{-2} \right\rfloor,$$

where c_1 is defined as in Theorem 3 and

$$c_2 = \sqrt{n} \max\left\{ \frac{\gamma + L}{\delta} + L\sqrt{n} + \frac{M_g}{\theta}, \frac{\gamma + L^{max} + M_g}{\theta}\right\}.$$

Proof. From Theorem 3, it follows that the sequence $\{\Phi_k\}$ is monotonically non-increasing. Furthermore, for all $k \geq 1$, we can write

$$\Phi_k - \Phi_0 \leq -c_1 \sum_{j=1}^{k} \Delta_j^2 = -c_1 \sum_{j=0}^{k-1} \Delta_{j+1}^2. \tag{30}$$

Since the sequence $\{\Phi_k\}$ is bounded from below, there exists Φ^* such that

$$\lim_{k\to\infty} \Phi_k = \Phi^* \geq f_{\min},$$

with $f_{\min}$ defined as in (18). Taking the limit for $k \to \infty$ in (30) we obtain

$$\Phi_0 - f_{\min} \geq \Phi_0 - \Phi^* \geq c_1 \sum_{k=0}^{\infty} \Delta_{k+1}^2 \geq c_1 \sum_{k \in K_\epsilon} \Delta_{k+1}^2.$$

Therefore, using the definition of K_ϵ and Theorem 2, we get

$$\Phi_0 - f_{\min} \geq c_1 \sum_{k \in K_\epsilon} \Delta_{k+1}^2 \geq |K_\epsilon| c_1 \frac{\epsilon^2}{c_2^2}.$$

Thus, the desired result is obtained. $\qquad\square$

As appears from the proof of Theorem 5, the above result relies on Theorem 2 which, in turn, uses Theorem 8. The latter, as pointed out in Remark 1, strongly relies on the extrapolation phase of the line search procedure (i.e., lines 12–15 of Algorithm 2), which allows us to bound $\chi(x_k)$ at both successful and unsuccessful iterations.

In the following theorem, we give an upper bound of $\mathcal{O}\left(n\epsilon^{-2}\right)$ on the maximum number of iterations required to produce a point x_k such that $\chi(x_k)$ is below a given threshold $\epsilon > 0$. This bound aligns with established findings for direct-search [23] and model-based [27] methods.

Theorem 6. *Let $\{x_k\}$ be the sequence of points produced by Algorithm 1. Given any $\epsilon > 0$, let $j_\epsilon \geq 1$ be the first iteration such that $\chi(x_{j_\epsilon}) < \epsilon$, that is, $\chi(x_k) \geq \epsilon$ for all $k \in \{0, \dots, j_\epsilon - 1\}$.*

Then, $j_\epsilon \leq \mathcal{O}\left(n\epsilon^{-2}\right)$. In particular,

$$j_\epsilon \leq \left\lfloor \frac{nc_3^2(\Phi_0 - f_{\min})}{c_1}\epsilon^{-2} \right\rfloor,$$

where c_1 is given by (20) *and*

$$c_3 = \frac{\gamma + L^{max} + M_g}{\theta}. \tag{31}$$

Proof. Let Φ_k the function defined in (16). We can write

$$\Phi_{j_\epsilon} - \Phi_0 = \sum_{k=0}^{j_\epsilon - 1}(\Phi_{k+1} - \Phi_k)$$

and, using Theorem 3, we have that

$$\Phi_{j_\epsilon} - \Phi_0 \leq -c_1 \sum_{k=0}^{j_\epsilon - 1} \Delta_{k+1}^2.$$

Recalling (18) and the fact that $\Phi_k \geq f(x_k)$ for $k \geq 0$, we get

$$f_{\min} - \Phi_0 \leq \Phi_{j_\epsilon} - \Phi_0 \leq -c_1 \sum_{k=0}^{j_\epsilon - 1} \Delta_{k+1}^2. \tag{32}$$

Now, we can partition the set of iteration indices $\{0, \dots, j_\epsilon - 1\}$ into $\mathcal{S}_{j_\epsilon}$ and $\mathcal{U}_{j_\epsilon}$ such that

$$k \in \mathcal{S}_{j_\epsilon} \Leftrightarrow x_k \neq x_{k-1}, \quad k \in \mathcal{U}_{j_\epsilon} \Leftrightarrow x_k = x_{k-1}, \quad \mathcal{S}_{j_\epsilon} \cup \mathcal{U}_{j_\epsilon} = \{0, \dots, j_\epsilon - 1\},$$

that is, $\mathcal{S}_{j_\epsilon}$ and $\mathcal{U}_{j_\epsilon}$ contain the successful and unsuccessful iterations up to $j_\epsilon - 1$, respectively. So, from (32), we can write

$$\Phi_0 - f_{\min} \geq c_1 \sum_{k \in \mathcal{S}_{j_\epsilon}} \Delta_{k+1}^2 + c_1 \sum_{k \in \mathcal{U}_{j_\epsilon}} \Delta_{k+1}^2. \tag{33}$$

For all $k \in \mathcal{S}_{j_\epsilon}$, let us define the index $m(k)$ as follows:

- if $\mathcal{U}_{j_\epsilon} \cap \{0, \dots, k-1\} \neq \emptyset$, then $m(k)$ in the largest index of $\mathcal{U}_{j_\epsilon} \cap \{0, \dots, k-1\}$;

- otherwise, $m(k) = -1$.

Note that, by definition, $m(k)$ is the last unsuccessful iteration before iteration k, i.e., all the iterations from $m(k)+1$ to k are successful iterations. Lemma 1 guarantees that $\Delta_{k+1} \geq \Delta_{m(k)+1}$ for all $k \in \mathcal{S}_{j_\epsilon}$. Using (33), we obtain

$$\Phi_0 - f_{\min} \geq c_1 \sum_{k \in \mathcal{S}_{j_\epsilon}} \Delta_{m(k)+1}^2 + c_1 \sum_{k \in \mathcal{U}_{j_\epsilon}} \Delta_{k+1}^2.$$

15

From Theorem 2, we have that $\Delta_{m(k)+1} \geq \chi(x_{m(k)})/(\sqrt{n}c_3)$ for all $k \in \mathcal{S}_{j_\epsilon}$ and $\Delta_{k+1} \geq \chi(x_k)/(\sqrt{n}c_3)$ for all $k \in \mathcal{U}_{j_\epsilon}$. Since $\chi(x_k) \geq \epsilon$ for all $k \in \{0, \ldots, j_\epsilon - 1\}$, with $\mathcal{S}_{j_\epsilon} \cup \mathcal{U}_{j_\epsilon} = \{0, \ldots, j_\epsilon - 1\}$, we get

$$\Phi_0 - f_{\min} \geq j_\epsilon \frac{c_1}{nc_3^2} \epsilon^2.$$

Thus, the desired result is obtained. □

The last complexity result we give is about the maximum number of function evaluations required to produce a point x_k such that $\chi(x_k)$ is less than or equal to a given threshold $\epsilon > 0$. Using arguments from the related literature [8, 40], we obtain an upper bound of $\mathcal{O}(n^2\epsilon^{-2})$, which still aligns with established findings for direct-search [23] and model-based [27] methods

Theorem 7. *Let $\{x_k\}$ be the sequence of points produced by Algorithm 1. Given any $\epsilon > 0$, let $j_\epsilon \geq 1$ be the first iteration such that $\chi(x_{j_\epsilon}) < \epsilon$, that is, $\chi(x_k) \geq \epsilon$ for all $k \in \{0, \ldots, j_\epsilon - 1\}$.*

Denoting by Nf_{j_ϵ} the number of function evaluations required by Algorithm 1 up to iteration j_ϵ, then $Nf_{j_\epsilon} \leq \mathcal{O}(n^2\epsilon^{-2})$. In particular,

$$Nf_{j_\epsilon} \leq 2n \left\lfloor \frac{nc_3^2(\Phi_0 - f_{\min})}{c_1} \epsilon^{-2} \right\rfloor + \left\lfloor \frac{nc_3^2(f_0 - f_{\min})}{\gamma c^2} \max\left\{1, \left(\frac{\delta}{1-\delta}\right)^2\right\} \epsilon^{-2} \right\rfloor,$$

where c_1 and c_3 are given in (20) *and* (31), *respectively.*

Proof. First, let us partition the set of iteration indices $\{0, \ldots, j_\epsilon - 1\}$ into $\mathcal{S}_{j_\epsilon}$ and $\mathcal{U}_{j_\epsilon}$ such that

$$k \in \mathcal{S}_{j_\epsilon} \Leftrightarrow x_k \neq x_{k-1}, \quad k \in \mathcal{U}_{j_\epsilon} \Leftrightarrow x_k = x_{k-1}, \quad \mathcal{S}_{j_\epsilon} \cup \mathcal{U}_{j_\epsilon} = \{0, \ldots, j_\epsilon - 1\},$$

that is, $\mathcal{S}_{j_\epsilon}$ and $\mathcal{U}_{j_\epsilon}$ contain the successful and unsuccessful iterations up to $j_\epsilon - 1$, respectively.

When the algorithm evaluates a new point, the latter can either succeed to decrease the objective function or fail to do so. Let us then define $Nf_{j_\epsilon}^{\mathcal{S}}$ as the total number of function evaluations related to points which succeed to decrease the objective function up to iteration j_ϵ. Note that, at each iteration, the maximum number of function evaluations related to points which fail to decrease the objective function is $2n$ (and it can be equal to $2n$ only when $T_k = \mathbb{R}^n$). So, we can write

$$Nf_{j_\epsilon} \leq 2nj_\epsilon + Nf_{j_\epsilon}^{\mathcal{S}} \leq 2n \left\lfloor \frac{nc_3^2(\Phi_0 - f_{\min})}{c_1} \epsilon^{-2} \right\rfloor + Nf_{j_\epsilon}^{\mathcal{S}}, \tag{34}$$

where the last inequality follows from Theorem 6. Now, let us consider any iteration $k < j_\epsilon$ and any index $i \in \{1, \ldots, n\}$ such that the line search succeeds to produce a decrease in the objective function. For each α used in the extrapolation phase of the line search, we have that either

$$f(y_k^i) - f(y_k^i + \alpha d_k^i) \geq \gamma\alpha^2 \geq \gamma c^2 \Delta_k^2, \tag{35}$$

or

$$f(y_k^i + \alpha d_k^i) - f(y_k^i + (\alpha/\delta)d_k^i) \geq \gamma\left(\frac{1-\delta}{\delta}\right)^2 \alpha^2 \geq \gamma\left(\frac{1-\delta}{\delta}\right)^2 c^2 \Delta_k^2. \tag{36}$$

Let us define the index $m(k)$ as follows:

- if $\mathcal{U}_{j_\epsilon} \cap \{0, \ldots, k-1\} \neq \emptyset$, then $m(k)$ is the largest index of $\mathcal{U}_{j_\epsilon} \cap \{0, \ldots, k-1\}$;

- otherwise, $m(k) = 0$.

Note that, by definition, $m(k)$ is the last unsuccessful iteration before iteration k, i.e., all the iterations from $m(k)+1$ to k are successful iterations. Lemma 1 guarantees that $\Delta_k \geq \Delta_{m(k)+1}$ for all $k \in \mathcal{S}_{j_\epsilon}$. Hence, from (35) and (36), it follows that

$$f(y_k^i) - f(y_k^i + \alpha d_k^i) \geq \gamma\alpha^2 \geq \gamma c^2 \Delta_{m(k)+1}^2 \geq \gamma\min\left\{1, \left(\frac{1-\delta}{\delta}\right)^2\right\}c^2\Delta_{m(k)+1}^2$$

$$f(y_k^i + \alpha d_k^i) - f(y_k^i + (\alpha/\delta)d_k^i) \geq \gamma\left(\frac{1-\delta}{\delta}\right)^2\alpha^2 \geq \gamma\min\left\{1, \left(\frac{1-\delta}{\delta}\right)^2\right\}c^2\Delta_{m(k)+1}^2.$$

From Theorem 2, we have that $\Delta_{m(k)+1} \geq \chi(x_{m(k)})/(\sqrt{n}c_3)$. Since $\chi(x_k) \geq \epsilon$ for all $k \in \{0, \ldots, j_\epsilon-1\}$, we can write

$$f(y_k^i + \alpha d_k^i) - f(y_k^i + (\alpha/\delta)d_k^i) \geq \gamma\min\left\{1, \left(\frac{1-\delta}{\delta}\right)^2\right\}c^2\frac{\epsilon^2}{nc_3^2}.$$

Then, recalling Assumption 1 and summing up the above relation over all function evaluations producing an objective decrease, we obtain

$$f_0 - f_{\min} \geq Nf_{j_\epsilon}^{\mathcal{S}}\gamma\min\left\{1, \left(\frac{1-\delta}{\delta}\right)^2\right\}c^2\frac{\epsilon^2}{nc_3^2},$$

that is,

$$Nf_{j_\epsilon}^{\mathcal{S}} \leq \left\lfloor\frac{nc_3^2(f_0-f_{\min})}{\gamma c^2}\max\left\{1, \left(\frac{\delta}{1-\delta}\right)^2\right\}\epsilon^{-2}\right\rfloor.$$

The desired results hence follows from (34). □

4 Finite active-set identification

In this section, we show that Algorithm 1 identifies the components of the final solution lying on the lower or the upper bounds (the so called *active set*) in a finite number of iterations.

First, let us give an equivalent definition of stationarity for problem (1), which will be useful in our analysis.

Definition 3. *A point $x^* \in [l, u]$ is said to be a stationary point of problem* (1) *(i.e., $\chi(x^*) = 0$) if, for all $i \in \{1, \ldots, n\}$, we have that*

$$\nabla_i f(x^*)\begin{cases}\geq 0 & \text{if } x_i^* = l_i, \\ = 0 & \text{if } l_i < x_i^* < u_i, \\ \leq 0 & \text{if } x_i^* = u_i.\end{cases}$$

Now, let us recall the definition of *strict complementarity* and *non-degenerate solutions*.

Definition 4. *Given a stationary point x^* of problem* (1)*, we say that a component x_i^* satisfies the strict complementarity condition if $x_i^* \in \{l_i, u_i\}$ and $\nabla_i f(x^*) \neq 0$. If the strict complementarity condition is satisfied by all components x_i^*, we say that x^* is non-degenerate.*

In particular, we define $\mathscr{Z}(x^*)$ as the *active set* for a stationary point x^*, that is, the index set for the active components of x^*. We also define $\mathscr{Z}^+(x^*)$ as the index set for those components satisfying the strict complementarity condition. Namely,

$$\mathscr{Z}(x^*) = \{i\colon x_i^* = l_i\} \cup \{i\colon x_i^* = u_i\} \qquad \text{and} \qquad \mathscr{Z}^+(x^*) = \mathscr{Z}(x^*) \cap \{i\colon \nabla_i f(x^*) \neq 0\}.$$

Furthermore, for any stationary point x^* such that $\mathscr{Z}^+(x^*) \neq \emptyset$, let us define

$$\zeta(x^*) = \min_{i \in \mathscr{Z}^+(x^*)} |\nabla_i f(x^*)|. \tag{37}$$

We see that $\zeta(x^*)$ is a measure of the minimum amount of strict complementarity among the variables in $\mathscr{Z}^+(x^*)$. This quantity will be used to define a neighborhood of x^* where the active components are correctly identified, following a similar approach as in [14, 37].

Before diving into the main theorem of this section, we need some preliminary results following from the Lipschitz continuity of ∇f. Recalling Definition 1, using standard arguments (see, e.g., [36]) one can prove that, for all $x \in [l, u]$, we have

$$|f(x + se_i) - f(x) - s\nabla_i f(x)| \leq \frac{L_i}{2} s^2 \quad \forall s \in \mathbb{R} \colon x + se_i \in [l, u], \quad i = 1, \dots, n.$$

Hence, for all $x \in [l, u]$, we have

$$f(x + se_i) \leq f(x) + s\nabla_i f(x) + \frac{L_i}{2} s^2 \quad \forall s \in \mathbb{R} \colon x + se_i \in [l, u], \quad i = 1, \dots, n, \tag{38}$$

$$f(x + se_i) \geq f(x) + s\nabla_i f(x) - \frac{L_i}{2} s^2 \quad \forall s \in \mathbb{R} \colon x + se_i \in [l, u], \quad i = 1, \dots, n. \tag{39}$$

The two following results provide bounds for the objective function when exploring any coordinate direction.

Proposition 5. *Given* $x \in [l, u]$, $\gamma \geq 0$ *and* $i \in \{1, \dots, n\}$, *then*

$$f(x - s\operatorname{sign}(\nabla_i f(x))e_i) \leq f(x) - \gamma s^2$$

for all $0 \leq s \leq 2\dfrac{|\nabla_i f(x)|}{L_i + 2\gamma}$ *such that* $x - s\operatorname{sign}(\nabla_i f(x))e_i \in [l, u]$.

Proof. From (38), we can write

$$f(x + se_i) \leq f(x) + s\left(\nabla_i f(x) + \frac{L_i}{2} s\right) \quad \forall s \in \mathbb{R} \colon x + se_i \in [l, u].$$

The right-hand side of the above inequality is less than or equal to $f(x) - \gamma s^2$ if

$$s\left(\nabla_i f(x) + \frac{L_i}{2} s\right) \leq -\gamma s^2.$$

If $\nabla_i f(x) \neq 0$, solving with respect to s we obtain

$$-\frac{2\nabla_i f(x)}{L_i + 2\gamma} \leq s \leq 0 \quad \text{if } \nabla_i f(x) > 0,$$

$$0 \leq s \leq -\frac{2\nabla_i f(x)}{L_i + 2\gamma} \quad \text{if } \nabla_i f(x) < 0,$$

leading to the desired result. $\qquad\square$

Proposition 6. *Given* $x \in [l, u]$ *and* $i \in \{1, \dots, n\}$, *then*

$$f(x + s\operatorname{sign}(\nabla_i f(x))e_i) \geq f(x)$$

for all $0 \leq s \leq \dfrac{2|\nabla_i f(x)|}{L_i}$ *such that* $x + s\operatorname{sign}(\nabla_i f(x))e_i \in [l, u]$.

Proof. From (39), we can write

$$f(x + se_i) \geq f(x) + s\left(\nabla_i f(x) - \frac{L_i}{2}s\right) \quad \forall s \in \mathbb{R} \colon x + se_i \in [l, u].$$

The right-hand side of the above inequality is greater than or equal to $f(x)$ if

$$s\left(\nabla_i f(x) - \frac{L_i}{2}s\right) \geq 0.$$

If $\nabla_i f(x) \neq 0$, solving with respect to s we obtain

$$0 \leq s \leq \frac{2\nabla_i f(x)}{L_i} \quad \text{if } \nabla_i f(x) > 0,$$

$$\frac{2\nabla_i f(x)}{L_i} \leq s \leq 0 \quad \text{if } \nabla_i f(x) < 0,$$

leading to the desired result. $\qquad\square$

The next proposition shows that, when ν_k^i is sufficiently small at a given iteration, Algorithm 1 cannot move along an ascent direction.

Proposition 7. *Consider an iteration k of Algorithm 1. If $\nu_k^i \leq 2|\nabla_i f(y_k^i)|/L_i$ for an index $i \in \{1, \dots, n\}$, then*

$$\alpha_k^i > 0 \implies d_k^i = -\operatorname{sign}(\nabla_i f(y_k^i))e_i.$$

Proof. Using Proposition 6, for all $\alpha \leq \nu_k^i$ and $\gamma > 0$ we have

$$f(y_k^i + \alpha \operatorname{sign}(\nabla_i f(y_k^i))e_i) \geq f(y_k^i) > f(y_k^i) - \gamma\alpha^2.$$

Thus, the line search in Algorithm 2 fails when using the direction $\operatorname{sign}(\nabla_i f(y_k^i))e_i$ with any stepsize $0 < \alpha \leq \nu_k^i$. So, if the line search returns $\alpha_k^i > 0$, necessarily $d_k^i = -\operatorname{sign}(\nabla_i f(y_k^i))e_i$. $\qquad\square$

Now, we are ready to state the main result of this section, establishing finite active-set identification of Algorithm 1.

Theorem 8. *Let $\{x_k\}$ be the sequence of points produced by Algorithm 1 and let x^* be a limit point of $\{x_k\}$, i.e., there exists an infinite subsequence $\{x_k\}_K \to x^*$. Then,*

(i) $\displaystyle\lim_{k \to \infty, \, k \in K} x_{k+1} = x^*$;

(ii) an iteration $\bar{k} \in K$ exists such that, for all $k \geq \bar{k}$, $k \in K$, we have that $(x_{k+1})_i = x_i^*$ for all $i \in \mathscr{L}^+(x^*)$.

Proof. Since $y_k^1 = x_k$ and $y_k^{i+1} = x_k + \sum_{j=1}^i \alpha_k^j d_k^j$, from Proposition 4 and the fact that $\|d_k^i\| = 1$, $i = 1, \dots, n+1$, we have

$$\lim_{\substack{k \to \infty \\ k \in K}} y_k^i = \lim_{\substack{k \to \infty \\ k \in K}} x_k = x^*, \quad i = 1, \dots, n+1. \tag{40}$$

Since $x_{k+1} = y_k^{n+1}$, then point (i) follows.

19

To show point (ii), assume that $\mathscr{Z}^+(x^*) \neq \emptyset$. Let $\bar{k} \in K$ be the first iteration such that the two following relations hold for all $k \geq \bar{k}$, $k \in K$:

$$\|y_k^i - x^*\| \leq \min\left\{\frac{1}{L}, \frac{2}{2L + L^{\max} + 2\gamma}\right\}\zeta(x^*), \quad i = 1, \ldots, n, \tag{41a}$$

$$\|y_k^i - x^*\| + \frac{L^{\max}}{2L} \max_{j=1,\ldots,n} \tilde{\alpha}_k^j \leq \frac{\zeta(x^*)}{L}, \quad i = 1, \ldots, n. \tag{41b}$$

Note that (40) and Proposition 4 imply the existence of $\bar{k} \in K$ such that (41) holds for all $k \geq \bar{k}$, $k \in K$.

Consider an index $i \in \mathscr{Z}^+(x^*)$ and an iteration $k \geq \bar{k}$, $k \in K$. To prove that $(x_{k+1})_i = x_i^*$, we have to show that $(y_k^{i+1})_i = x_i^*$ since, from the instructions of the algorithm, $(x_{k+1})_i = (y_k^{i+1})_i$. Without loss of generality, assume that $x_i^* = l_i$ (the proof for the case $x_i^* = u_i$ is identical, except for minor changes). So, we have to show that

$$(y_k^{i+1})_i = l_i. \tag{42}$$

Preliminarily, we want to prove that

$$|z_i - l_i| \leq \frac{2\nabla_i f(z)}{L_i + 2\gamma} \quad \forall z \text{ such that } \|z - x^*\| \leq \|y_k^i - x^*\|, \tag{43}$$

$$\nu_k^i \leq \frac{2\nabla_i f(y_k^i)}{L_i}. \tag{44}$$

According to Definitions 3 and 4, we have that $\nabla_i f(x^*) > 0$ and, from the definition of $\zeta(x^*)$ given in (37), it follows that

$$0 < \zeta(x^*) \leq \nabla_i f(x^*). \tag{45}$$

Consider any $z \in \mathbb{R}^n$ such that $\|z - x^*\| \leq \|y_k^i - x^*\|$. Using the Lipschitz continuity of ∇f, we have

$$\nabla_i f(x^*) - \nabla_i f(z) \leq \|\nabla f(z) - \nabla f(x^*)\| \leq L\|z - x^*\|. \tag{46}$$

Moreover, from (41a) we can write

$$\|z - x^*\| \leq \|y_k^i - x^*\| \leq \frac{2\zeta(x^*)}{2L + L^{\max} + 2\gamma}.$$

Multiplying the first and last terms above by $(2L + L^{\max} + 2\gamma)/(L^{\max} + 2\gamma)$, we have

$$\left(\frac{2L}{L^{\max} + 2\gamma} + 1\right)\|z - x^*\| \leq \frac{2\zeta(x^*)}{L^{\max} + 2\gamma},$$

that is,

$$\|z - x^*\| \leq \left(\zeta(x^*) - L\|z - x^*\|\right)\frac{2}{L^{\max} + 2\gamma}. \tag{47}$$

Since, from (45) and (46), we have

$$\zeta(x^*) \leq \nabla_i f(z) + L\|z - x^*\|, \tag{48}$$

then, using (47), we obtain

$$\|z - x^*\| \leq \frac{2\nabla_i f(z)}{L^{\max} + 2\gamma}.$$

Taking into account that $L_i \leq L^{\max}$ and recalling that $x_i^* = l_i$, it follows that (43) holds. To prove (44), from (41b) and the definition of ν_k^i we can write

$$\nu_k^i \leq \max_{j=1,\dots,n} \tilde{\alpha}_k^j \leq \left(\zeta(x^*) - L\|y_k^i - x^*\|\right) \frac{2}{L^{\max}}.$$

Using (48) with $z = y_k^i$ and the fact that $L_i \leq L^{\max}$, we thus get (44).

In view of (44) and Proposition 7, it follows that

$$d_k^i = -e_i, \tag{49}$$

that is, $y_k^{i+1} = y_k^i - \alpha_k^i e_i$. Using $z = y_k^i$ in (43), we also have

$$\alpha_{\max} = |(y_k^i)_i - l_i| \leq \frac{2\nabla_i f(y_k^i)}{L_i + 2\gamma}, \tag{50}$$

where $\alpha_{\max}$ is the largest feasible stepsize along the direction d_k^i at y_k^i. So, if $\alpha_{\max} = 0$, then $\alpha_k^i = 0$ and, using (49), we have $(y_k^{i+1})_i = (y_k^i)_i = l_i$, thus proving (42). If $\alpha_{\max} > 0$, using Proposition 5, (49) and (50), it follows that a sufficient decrease of f along d_k^i is obtained with the first stepsize $\bar{\alpha}$ used in the line search, that is, the condition at line 4 of Algorithm 2 is satisfied. Now, consider the extrapolation phase in the line search procedure, that is, lines 12–15 of Algorithm 2. Recalling (49), each stepsize $\omega = \min\{\alpha/\delta, \alpha_{\max}\}$ is such that $\alpha \leq \omega \leq (y_k^i)_i - l_i$, that is,

$$0 \leq \omega - \alpha \leq (y_k^i + \alpha d_k^i)_i - l_i.$$

So, from (49) and the fact that $x_i^* = l_i$, it follows that $\|y_k^i + \alpha d_k^i - x^*\| \leq \|y_k^i - x^*\|$. So, we can apply (43) with $z = y_k^i + \alpha d_k^i$ and then we obtain

$$0 \leq \omega - \alpha \leq (y_k^i + \alpha d_k^i)_i - l_i \leq \frac{2\nabla_i f(y_k^i + \alpha d_k^i)}{L_i + 2\gamma}$$

for every stepsize ω used in the extrapolation. Then, using Proposition 5 with $x = y_k^i + \alpha d_k^i$, $s = \omega - \alpha$ and d_k^i as in (49), it follows that

$$f(y_k^i + \omega d_k^i) \leq f(y_k^i + \alpha d_k^i) - \gamma(\omega - \alpha)^2.$$

Namely, a sufficient decrease of f is obtained with all stepsizes used in the extrapolation and we quit when we get the largest feasible stepsize, meaning that $(y_k^{i+1})_i$ will be at the lower bound l_i and thus proving (42). $\square$

Note that Theorem 8 establishes finite identification for any limit point of $\{x_k\}$, thus not requiring the convergence of the whole sequence. Note also, in the proof of Theorem 8, the crucial role played by the extrapolation in the line search procedure. Loosely speaking, when we are sufficiently close to a stationary point, expanding the stepsize allows us to hit the lower or the upper bound, provided the strict complementarity condition holds. This guarantees to identify all the variables satisfying the strict complementarity after a finite number of iterations.

Now, let us point out a useful property for the limit points of $\{x_k\}$. To this aim, let us define $X^\star$ as the set of all limit points of $\{x_k\}$, i.e.,

$$X^\star := \{x \colon \exists K \subseteq \mathbb{N} \text{ such that } \lim_{k \to \infty, k \in K} x_k = x\}.$$

The following result roughly states that, if for all $x \in X^\star$ there does not exists i such that x_i violates the strict complementarity, then either all x_i lie on the same bound or they all are strictly feasible.

21

Proposition 8. *Let $\{x_k\}$ be the sequence of points produced by Algorithm 1 and consider an index $i \in \{1, \ldots, n\}$. Assume that there is no $x \in X^\star$ such that $i \in \mathscr{Z}(x) \setminus \mathscr{Z}^+(x)$. Then,*

- *if there exists $x^* \in X^\star$ such that $i \in \mathscr{Z}^+(x^\star)$, we have that $x_i = x_i^*$ for all $x \in X^\star$;*

- *otherwise, $x_i \in (l_i, u_i)$ for all $x \in X^\star$.*

Proof. We limit ourselves to show only the first point since the second one can be obtained as a logical consequence. From Theorem 4 and recalling Ostrowski's theorem [3], the set $X^\star$ of limit points of the sequence $\{x_k\}$ is a connected set. Now, let us consider any two points in $X^\star$, say $\bar{x} \in X^\star$ and $\bar{y} \in X^\star$, such that $\bar{x} \neq \bar{y}$, and $\bar{y}_i \in \{l_i, u_i\}$. Since $X^\star$ is connected, there exists a continuous function $\rho : [a, b] \to \mathbb{R}^n$ such that $\rho(a) = \bar{x}$, $\rho(b) = \bar{y}$ and $\rho(t) \in X^\star$, i.e., $\rho(t)$ is stationary, for all $t \in [a, b]$. Let us assume, without loss of generality, that $\bar{y}_i = l_i$ (the proof for the case $\bar{y}_i = u_i$ is identical, except for minor changes). By contradiction, now assume that $\bar{x}_i > l_i$. Since $\rho(a)_i = \bar{x}_i > l_i$ and $\rho(b)_i = \bar{y}_i = l_i$, then there exists $\bar{t} \in (a, b]$ such that $\rho(t)_i > l_i$ for all $t \in [a, \bar{t})$ and $\rho(\bar{t})_i = l_i$. Furthermore, by the stationarity conditions given in Definition 3, we have that $\nabla_i f(\rho(t)) = 0$ for all $t \in [a, \bar{t})$ and $\nabla_i f(\rho(\bar{t})) > 0$, where the last inequality follows from the stated hypothesis. Then, by continuity of ∇f, a scalar $\hat{t} \in (a, \bar{t})$ must exist such that $\nabla_i f(\rho(t)) > 0$ for all $t \in (\hat{t}, \bar{t}]$. This is a contradiction since $\nabla_i f(\rho(t)) = 0$ for all $t \in (\hat{t}, \bar{t})$. $\qquad\square$

Applying the above proposition for all indices $i \in \{1, \ldots, n\}$, the following result immediately follows, enforcing the finite active-set identification property established in Theorem 8 when all the limit points of $\{x_k\}$ are non-degenerate.

Corollary 1. *Let $\{x_k\}$ be the sequence of points produced by Algorithm 1 and assume that every $x \in X^\star$ is non-degenerate. Then, for any pair $x', x'' \in X^\star$, we have $\mathscr{Z}(x') = \mathscr{Z}(x'')$ and $x_i' = x_i''$ for all $i \in \mathscr{Z}(x')$.*

5   Conclusions

In this paper, we have analyzed a derivative-free line search method for bound-constrained problems where the objective function has a Lipschitz continuous gradient. For this algorithm, we have first provided complexity results. In more detail, given a prespecified threshold $\epsilon > 0$, we have shown that the criticality measure $\chi(x_k)$ (which vanishes at stationary points) falls below ϵ after at most $\mathcal{O}\left(n\epsilon^{-2}\right)$ iterations, requiring at most $\mathcal{O}\left(n^2\epsilon^{-2}\right)$ function evaluations. These bounds match those obtained for (deterministic) direct-search [23] and model-based [27] methods. Additionally, we have established an upper bound of $\mathcal{O}\left(n^2\epsilon^{-2}\right)$ on the *total* number of iterations where $\chi(x_k) \geq \epsilon$. The latter result is obtained thanks to the extrapolation strategy used in the proposed line search, allowing us to upper bound $\chi(x_k)$ on both successful and unsuccessful iterations.

In the last part of the paper, we have considered the active-set identification property of the proposed method, i.e., the capability to detect the variables lying at the lower or the upper bound in the final solutions. In this respect, we have shown that, in a finite number of iterations, the algorithm identifies the active constraints satisfying the strict complementarity condition. Also this property is obtained by exploiting the extrapolation used in the proposed line search, allowing the stepsize to expand, when we are in a neighborhood of a stationary point, until we hit the boundary of the feasible set.

Finally, some topics for future research can be envisaged. In particular, under convexity assumptions, the worst-case complexity of the algorithm might be tightened, in order to match the results given in [18], and a bound on the maximum number of iterations required to identify the active constraints might be given. We wish to report more results in future works.

References

[1] C. Audet and J. E. Dennis Jr. Mesh Adaptive Direct Search Algorithms for Constrained Optimization. *SIAM Journal on optimization*, 17(1):188–217, 2006. doi: 10.1137/040603371.

[2] C. Audet and W. L. Hare. *Derivative-free and blackbox optimization*. Springer Series in Operations Research and Financial Engineering. Springer, 2017. doi: 10.1007/978-3-319-68913-5.

[3] D. Bertsekas. *Nonlinear Programming*. Athena Scientific, Massachusetts, 2016.

[4] D. P. Bertsekas. On the Goldstein-Levitin-Polyak gradient projection method. *IEEE Transactions on Automatic Control*, 21(2):174–184, 1976. doi: 10.1109/TAC.1976.1101194.

[5] E. G. Birgin and J. Mario Martínez. Large-scale active-set box-constrained optimization method with spectral projected gradients. *Computational Optimization and Applications*, 23(1):101–125, 2002. doi: 10.1023/A:1019928808826.

[6] I. M. Bomze, F. Rinaldi, and S. R. Bulo. First-order methods for the impatient: Support identification in finite time with convergent Frank-Wolfe variants. *SIAM Journal on Optimization*, 29(3):2211–2226, 2019. doi: 10.1137/18M1206953.

[7] I. M. Bomze, F. Rinaldi, and D. Zeffiro. Active set complexity of the away-step Frank-Wolfe algorithm. *SIAM Journal on Optimization*, 30(3):2470–2500, 2020. doi: 10.1137/19M1309419.

[8] A. Brilli, M. Kimiaei, G. Liuzzi, and S. Lucidi. Worst case complexity bounds for linesearch-type derivative-free algorithms. *Journal of Optimization Theory and Applications*, 203:419–454, 2024. doi: 10.1007/s10957-024-02519-x.

[9] J. V. Burke and J. J. Moré. On the identification of active constraints. *SIAM Journal on Numerical Analysis*, 25(5):1197–1211, 1988. doi: 10.1137/0725068.

[10] P. Conejo, E. W. Karas, L. G. Pedroso, A. A. Ribeiro, and M. Sachine. Global convergence of trust-region algorithms for convex constrained minimization without derivatives. *Applied Mathematics and Computation*, 220:324–330, 2013. doi: 10.1016/j.amc.2013.06.041.

[11] A. R. Conn, N. Gould, A. Sartenaer, and P. L. Toint. Convergence Properties of an Augmented Lagrangian Algorithm for Optimization with a Combination of General Equality and Linear Constraints. *SIAM Journal on Optimization*, 6(3):674–703, 1996. doi: 10.1137/S1052623493251463.

[12] A. R. Conn, N. I. Gould, and P. L. Toint. *Trust-Region Methods*. MOS-SIAM Series on Optimization, 2000. doi: 10.1137/1.9780898719857.

[13] A. R. Conn, K. Scheinberg, and L. N. Vicente. *Introduction to derivative-free optimization*. MOS-SIAM Series on Optimization, 2009. doi: 10.1137/1.9780898718768.

[14] A. Cristofari. Active-set identification with complexity guarantees of an almost cyclic 2-coordinate descent method with Armijo line search. *SIAM Journal on Optimization*, 32(2):739–764, 2022. doi: 10.1137/20M1328014.

[15] A. Cristofari and F. Rinaldi. A derivative-free method for structured optimization problems. *SIAM Journal on Optimization*, 31(2):1079–1107, 2021. doi: 10.1137/20M1337417.

[16] A. Cristofari, M. De Santis, S. Lucidi, and F. Rinaldi. A two-stage active-set algorithm for bound-constrained optimization. *Journal of Optimization Theory and Applications*, 172(2):369–401, 2017. doi: 10.1007/s10957-016-1024-9.

[17] M. De Santis, G. Di Pillo, and S. Lucidi. An active set feasible method for large-scale minimization problems with bound constraints. *Computational Optimization and Applications*, 53(2):395–423, 2012. doi: 10.1007/s10589-012-9506-7.

[18] M. Dodangeh and L. N. Vicente. Worst case complexity of direct search under convexity. *Mathematical Programming*, 155(1):307–332, 2016. doi: 10.1007/s10107-014-0847-0.

[19] M. Dodangeh, L. N. Vicente, and Z. Zhang. On the optimal order of worst case complexity of direct search. *Optimization Letters*, 10:699–708, 2016. doi: 10.1007/s11590-015-0908-1.

[20] R. Garmanjani, D. Júdice, and L. N. Vicente. Trust-region methods without using derivatives: worst case complexity and the nonsmooth case. *SIAM Journal on Optimization*, 26(4):1987–2011, 2016. doi: 10.1137/151005683.

[21] G. N. Grapiglia. Quadratic regularization methods with finite-difference gradient approximations. *Computational Optimization and Applications*, 85(3):683–703, 2023. doi: 10.1007/s10589-022-00373-z.

[22] S. Gratton, P. L. Toint, and A. Tröltzsch. An active-set trust-region method for derivative-free nonlinear bound-constrained optimization. *Optimization Methods and Software*, 26(4-5):873–894, 2011. doi: 10.1080/10556788.2010.549231.

[23] S. Gratton, C. W. Royer, L. N. Vicente, and Z. Zhang. Direct search based on probabilistic feasible descent for bound and linearly constrained problems. *Computational Optimization and Applications*, 72:525–559, 2019. doi: 10.1007/s10589-019-00062-4.

[24] E. A. Gumma, M. Hashim, and M. M. Ali. A derivative-free algorithm for linearly constrained optimization problems. *Computational Optimization and Applications*, 57(3):599–621, 2014. doi: 10.1007/s10589-013-9607-y.

[25] W. W. Hager and H. Zhang. A new active set algorithm for box constrained optimization. *SIAM Journal on Optimization*, 17(2):526–557, 2006. doi: 10.1137/050635225.

[26] W. L. Hare and A. S. Lewis. Identifying active constraints via partial smoothness and prox-regularity. *Journal of Convex Analysis*, 11(2):251–266, 2004.

[27] M. Hough and L. Roberts. Model-Based Derivative-Free Methods for Convex-Constrained Optimization. *SIAM Journal on Optimization*, 32(4):2552–2579, 2022. doi: 10.1137/21M1460971.

[28] T. G. Kolda, R. M. Lewis, and V. Torczon. Optimization by direct search: New perspectives on some classical and modern methods. *SIAM review*, 45(3):385–482, 2003. doi: 10.1137/S003614450242889.

[29] T. G. Kolda, R. M. Lewis, and V. Torczon. Stationarity results for generating set search for linearly constrained optimization. *SIAM Journal on Optimization*, 17(4):943–968, 2007. doi: 10.1137/S1052623403433638.

[30] J. Larson, M. Menickelly, and S. M. Wild. Derivative-free optimization methods. *Acta Numerica*, 28:287–404, 2019. doi: 10.1017/S0962492919000060.

[31] R. M. Lewis and V. Torczon. Pattern Search Algorithms for Bound Constrained Minimization. *SIAM Journal on optimization*, 9(4):1082–1099, 1999. doi: 10.1137/S1052623496300507.

[32] R. M. Lewis and V. Torczon. Active set identification for linearly constrained minimization without explicit derivatives. *SIAM Journal on Optimization*, 20(3):1378–1405, 2010. doi: 10.1137/08073545X.

[33] S. Lucidi and M. Sciandrone. A derivative-free algorithm for bound constrained optimization. *Computational Optimization and applications*, 21:119–142, 2002. doi: 10.1023/A:1013735414984.

[34] S. Lucidi and M. Sciandrone. On the global convergence of derivative-free methods for unconstrained optimization. *SIAM Journal on Optimization*, 13(1):97–116, 2002. doi: 10.1137/S1052623497330392.

[35] S. Lucidi, M. Sciandrone, and P. Tseng. Objective-derivative-free methods for constrained optimization. *Mathematical Programming*, 92:37–59, 2002. doi: 10.1007/s101070100266.

[36] Y. Nesterov. *Introductory Lectures on Convex Optimization: A Basic Course*, volume 87. Springer Science & Business Media, 2013. doi: 10.1007/978-1-4419-8853-9.

[37] J. Nutini, M. Schmidt, and W. Hare. "Active-set complexity" of proximal gradient: How long does it take to find the sparsity pattern? *Optimization Letters*, 13:645–655, 2019. doi: 10.1007/s11590-018-1325-z.

[38] M. J. Powell. On fast trust region methods for quadratic models with linear constraints. *Mathematical Programming Computation*, 7:237–267, 2015. doi: 10.1007/s12532-015-0084-4.

[39] H.-J. M. Shi, M. Qiming Xuan, F. Oztoprak, and J. Nocedal. On the numerical performance of finite-difference-based methods for derivative-free optimization. *Optimization Methods and Software*, 38(2):289–311, 2023. doi: 10.1080/10556788.2022.2121832.

[40] L. N. Vicente. Worst case complexity of direct search. *EURO Journal on Computational Optimization*, 1(1):143–153, 2013. doi: 10.1007/s13675-012-0003-7.

[41] L. N. Vicente. Worst case complexity of direct search. *EURO Journal on Computational Optimization*, 1:143–153, 2013.

[42] S. J. Wright. Identifiable surfaces in constrained optimization. *SIAM Journal on Control and Optimization*, 31(4):1063–1079, 1993. doi: 10.1137/0331048.

[43] G. Yuan and X. Lu. An active set limited memory BFGS algorithm for bound constrained optimization. *Applied Mathematical Modelling*, 35(7):3561–3573, 2011. doi: 10.1016/j.apm.2011. 01.036.

[44] G. Yuan, Z. Wei, and M. Zhang. An active-set projected trust region algorithm for box constrained optimization problems. *Journal of Systems Science and Complexity*, 28(5):1128–1147, 2015. doi: 10.1007/s11424-014-2199-5.