UNIVERSITÀ DEGLI STUDI FIRENZE

FLORE
Repository istituzionale dell'Università degli Studi di Firenze

A derivative-free algorithm for bound constrained optimization.

Questa è la Versione finale referata (Post print/Accepted manuscript) della seguente pubblicazione:

Original Citation:
A derivative-free algorithm for bound constrained optimization / S. LUCIDI; M. SCIANDRONE. - In: COMPUTATIONAL OPTIMIZATION AND APPLICATIONS. - ISSN 0926-6003. - STAMPA. - 21:(2002), pp. 119-142.

Availability:
This version is available at: 2158/256057 since:

Publisher:
Kluwer Academic Publishers / Massachusetts:PO Box 358, Accord Station:Hingham, MA 02018:(617)871-

Terms of use:
Open Access
La pubblicazione è resa disponibile sotto le norme e i termini della licenza di deposito, secondo quanto stabilito dalla Policy per l'accesso aperto dell'Università degli Studi di Firenze (https://www.sba.unifi.it/upload/policy-oa-2016-1.pdf)

Publisher copyright claim:

(Article begins on next page)

12 October 2024

Computational Optimization and Applications, 21, 119–142, 2002
© 2002 Kluwer Academic Publishers. Manufactured in The Netherlands.

A Derivative-Free Algorithm for Bound Constrained Optimization

STEFANO LUCIDI lucidi@dis.uniromal.it
Dipartimento di Informatica e Sistemistica, Università di Roma "La Sapienza", Roma, Italy

MARCO SCIANDRONE sciandro@iasi.rm.cnr.it
Istituto di Analisi dei Sistemi ed Informatica, CNR, Roma, Italy

Revised 15 January, 2001

Abstract. In this work, we propose a new globally convergent derivative-free algorithm for the minimization of a continuously differentiable function in the case that some of (or all) the variables are bounded. This algorithm investigates the local behaviour of the objective function on the feasible set by sampling it along the coordinate directions. Whenever a "suitable" descent feasible coordinate direction is detected a new point is produced by performing a linesearch along this direction. The information progressively obtained during the iterates of the algorithm can be used to build an approximation model of the objective function. The minimum of such a model is accepted if it produces an improvement of the objective function value. We also derive a bound for the limit accuracy of the algorithm in the minimization of noisy functions. Finally, we report the results of a preliminary numerical experience.

Keywords: derivative-free algorithm, bound constraints, linesearch technique

1. Introduction

Practical applications very often lead to the minimization of a smooth function whose variables are subject to bound constraints. In many of these cases the objective function value is obtained by direct measurements or it is the result of a complex system of calculations, such as a simulation. Therefore, even if it is known that the objective function is smooth, its analytical expression is not available and the computation of its values may be expensive and/or affected by the presence of noise. Hence, the first order derivatives cannot be explicitly calculated or approximated. This motivates the increasing interest in studying new derivative-free methods for bound constrained optimization. Such methods should present strong global convergence properties, should be able to make significant progresses with few function evaluations, and should be sufficiently robust in the noisy case.

With this in mind, in this paper we consider the problem

$$\begin{aligned} &\text{minimize} \quad f(x) \\ &\text{subject to} \quad l \le x \le u, \end{aligned} \tag{1}$$

where $x, l, u \in R^n$, with $l < u$, and $f : R^n \to R$ is a continuously differentiable function, but its first order derivatives cannot be explicitly calculated or approximated. We allow the possibility that some of the variables are unbounded by permitting both $l^i = -\infty$ and

$u^i = \infty$ for some $i \in \{1, \ldots, n\}$. We denote the feasible set by

$$\mathcal{F} = \{x \in R^n : l \leq x \leq u\}.$$

We define a stationary point of problem (1) a feasible point $\bar{x}$ that satisfies the following first-order necessary optimality condition:

$$\nabla f(\bar{x})^T (y - \bar{x}) \geq 0 \quad \text{for all } y \in \mathcal{F}. \tag{2}$$

We recall that at any stationary point $\bar{x}$ of $f(x)$ ($\bar{x} \in \mathcal{F}$), we have that

$$\nabla^{red} f(\bar{x}) = 0$$

where the *reduced gradient* $\nabla^{red} f(x)$ is defined as follows

$$\nabla_i^{red} f(x) = \begin{cases} \max(\nabla_i f(x), 0) & \text{if } x^i = u^i \\ \min(\nabla_i f(x), 0) & \text{if } x^i = l^i \\ \nabla_i f(x) & \text{otherwise} \end{cases} \tag{3}$$

In order to overcome the lack of gradient information, many globally convergent derivative-free algorithms proposed in literature are based on the idea of performing finer and finer samplings of the objective function along suitable sets of search directions (see, for instance, [2, 11, 14–16] and the references quoted there).

In [8] it has been performed a general analysis of the requirements on the search directions and the sampling techniques, which ensure the global convergence of a derivative-free algorithm for unconstrained minimization problems. Roughly speaking, at every non-stationary point, the set of search directions must contain a descent direction, and the sampling technique must produce a suitable point along such a direction.

As clearly described in [7], the presence of bound constraints imposes stronger restrictions on the choice of search directions. In particular, at every non-stationary point, the set of search directions must contain a descent direction which is also feasible, in the sense that (sufficiently) small stepsizes along such direction must produce feasible points where the objective function is reduced. In [7] it is shown that the set of coordinate directions satisfies this property. Globally convergent algorithms using these directions have been proposed in [7] and [3]. More in particular, the algorithm proposed in [7] follows a pattern search strategy by evaluating the objective function on specified geometric patterns. While the method introduced in [3] is based on the idea of using approximating models of the objective function which are built by using suitable grid points.

In this paper, we propose a new algorithm model for solving problem (1). In order to try to guarantee both global convergence properties and a "good" efficiency of the proposed algorithm, we have drawn our inspiration from the strategy underlying the gradient based methods. The global convergence and the good computational behaviour of these methods follow from the fact that they are able

(i) to find a "good" feasible descent direction (namely a feasible direction along which the objective function sufficiently decreases);
(ii) to perform a "sufficiently" large steplength along this direction;

(iii) to exploit the information on the objective function obtained during the iterates of the algorithm.

Here, starting from the approaches proposed in [3, 7] and [8], we have defined an algorithm which tries to follow the points (i)–(iii) without using any information on the first order derivatives and taking account the particular structure of the feasible set. The main features of the algorithm are the following:

- a "good" feasible descent direction is determined by investigating the local behaviour of the objective function on the feasible set along the coordinate directions;
- whenever a "suitable" descent feasible coordinate direction is detected, a new point is produced by performing a derivative free linesearch along this direction;
- the information progressively obtained during the iterates of the algorithm can be used to build an approximation model of the objective function in order to improve the local behaviour of the algorithm.

As regards the theoretical properties of the proposed algorithm, we prove that every limit point of the sequence produced is a stationary point for problem (1).

Similarly to [1, 3, 6, 17], we consider also the minimization of noisy functions which are perturbations of smooth functions. For this case, by requiring the standard assumption that the gradient of the objective function is Lipschitz continuous (which is not required in the noiseless case), we derive a bound for the limit accuracy of the algorithm.

The paper is organized as follows. In Section 2 we describe the proposed algorithm model. In Section 3 we prove the global convergence of the algorithm. In Section 4, we characterize the behaviour of the algorithm in the minimization of noisy functions. Finally, in Section 5 we report the results of a preliminary numerical experience performed on both standard test problems and a real application.

Notation. The j-th component of a vector $v \in R^n$ is indicated by v^j. We denote the Euclidean norm (on the appropriate space) by $\|\cdot\|$. A subsequence of $\{x_k\}$ corresponding to an infinite subset K will be denoted by $\{x_k\}_K$. We indicate by e_i, with $i = 1, \ldots, n$, the orthonormal set of the coordinate directions and by $\partial\mathcal{F}$ the boundary of the feasible set $\mathcal{F}$. Finally, given a real number a, $\lfloor a \rfloor$ represents the largest integer that is not greater than a.

2. A new algorithm model

In this section we propose a new class of derivative-free algorithms for the minimization of a continuously differentiable function in the case that some of (or all) the variables are bounded. As said before, the approach is based on the idea of performing suitable samplings of the objective function along the coordinate directions. As pointed out in [7], the coordinate directions allow us to cope with the presence of box constraints. This can be easily derived from the optimality conditions (2). In fact, if a feasible point $\bar{x}$ is not a stationary point of f, then there must exist a feasible point y and an integer $h \in \{1, \ldots, n\}$ such that $\nabla_h f(\bar{x})^T (y - \bar{x})^h < 0$. If $\bar{\alpha} = (y - \bar{x})^h > 0$, then, taking into account that $\mathcal{F}$ is defined by box constraints, we have

$$\bar{\alpha} \nabla f(\bar{x})^T e_h < 0, \quad \bar{x} + \bar{\alpha} e_h \in \mathcal{F}$$

The continuity of ∇f and the convexity of the feasible set $\mathcal{F}$ imply that there exists a positive value $\tilde{\alpha}$ such that:

$$f(\bar{x} + \alpha e_h) < f(\bar{x}), \quad \bar{x} + \alpha e_h \in \mathcal{F},$$

for all $\alpha \in (0, \tilde{\alpha})$. The case $\bar{\alpha} = (y - \bar{x})^h < 0$ leads to the same conclusions with e_h replaced by $-e_h$. Hence, in correspondence to any feasible point $\bar{x}$ which is not a stationary point, there is a coordinate direction along which (or along its opposite) there must exist feasible points where the function is strictly decreased (this property is not necessary true for different sets of n linearly independent directions). Therefore, performing finder and finer samplings of the objective function along the coordinate directions and their opposite, it is possible either to understand that a point is a good approximation of a stationary point of f, or to determine a specific direction along which the objective function decreases.

On this basis, we propose an algorithm model which samples the objective function along the coordinate directions, with the aim of detecting a feasible direction where the objective function is sufficiently decreased. Once such a direction has been individuated, a derivative-free linesearch technique is adopted for performing a sufficiently large step along it, so as to exploit the descent property of the search direction as much as possible. Periodically, it is admitted the possibility of generating a point by a movement of arbitrary length along an arbitrary direction.

The use of the coordinate directions as search directions and the particular sampling technique adopted allow us to overcome the lack of gradient information and to ensure that every limit point of the sequence produced is a stationary point for problem (1). Formally, the algorithm model is described as follows.

Algorithm Model

Data. $x_0 \in \mathcal{F}, \theta \in (0, 1), \gamma > 0, 0 < \tilde{\alpha}_0^i < \infty, d_i = e_i$ for $i = 1, \ldots, n$.

Step 0. Set $k = 0, i = 1, h_k = 1$.

Step 1. Compute α_{max} s.t. $x_k + \alpha_{max} d_i \in \partial F$ and set $\alpha = \min\{\tilde{\alpha}_k^i, \alpha_{max}\}$.
If $\alpha > 0$ and $f(x_k + \alpha d_i) \leq f(x_k) - \gamma(\alpha)^2$ go to Step 3.

Step 2. Compute α_{max} s.t. $x_k - \alpha_{max} d_i \in \partial \mathcal{F}$ and set $\alpha = \min\{\tilde{\alpha}_k^i, \alpha_{max}\}$.
If $\alpha > 0$ and $f(x_k - \alpha d_i) \leq f(x_k) - \gamma(\alpha)^2$ then
set $d_i = -d_i$ and go to Step 3.
else set $\alpha_k = 0, \tilde{\alpha}_{k+1}^i = \theta\alpha$, and go to Step 4.

Step 3. Compute α_k by the *Expansion Step*$(d_i, \alpha, \alpha_{max}, \gamma)$ and set $\tilde{\alpha}_{k+1}^i = \alpha_k$.

Step 4. Set $\tilde{x}_{k+1} = x_k + \alpha_k d_i, \tilde{\alpha}_{k+1}^j = \tilde{\alpha}_k^j$, for $j \in \{1, \ldots, n\}$ and $j \neq i$.

Step 5. If $h_k \geq n$ then find x_{k+1} such that

$$f(x_{k+1}) \leq f(\tilde{x}_{k+1}) \quad \text{and} \quad x_{k+1} \in \mathcal{F}, \tag{4}$$

else set $x_{k+1} = \tilde{x}_{k+1}$.
If $x_{k+1} \neq \tilde{x}_{k+1}$ then set $h_{k+1} = 1$ else set $h_{k+1} = h_k + 1$.
Set $i = mod(i, n) + 1$, $k = k + 1$ and go to Step 1.

Expansion Step $(d_i, \alpha, \alpha_{max}, \gamma)$**.**

Data. $\delta \in (0, 1)$.
Step 1. Let $\tilde{\alpha} = \min\{\alpha_{max}, \frac{\alpha}{\delta}\}$.
If $\alpha = \alpha_{max}$ or $f(x_k + \tilde{\alpha} d_i) > f(x_k) - \gamma \tilde{\alpha}^2$ set $\alpha_k = \alpha$ and stop.
Step 2. Set $\alpha = \tilde{\alpha}$ and go to Step 1.

More in particular, the steps of the algorithm can be summarized as follows.

- At Step 1 the direction d_i is examined with the aim of determining (if possible) a feasible point where the objective function is sufficiently decreased. First, it is computed the maximum feasible steplength α_{max} which can be performed along the direction d_i starting from the point x_k. Then, the trial stepsize α is determined by choosing the minimum between α_{max} and $\tilde{\alpha}_k^i$. The scalar $\tilde{\alpha}_k^i$ has been computed on the basis of the behaviour of the objective function along the same direction showed at the previous iterations. Therefore, the scalar $\tilde{\alpha}_k^i$ should take into account the sensitivity of the objective function with respect to the i-th variable, and hence it should provide a promising initial stepsize for the direction d_i. Finally, it is verified if the moving of length α along d_i produces a feasible point where the function is sufficiently reduced. If such a point is produced then a linesearch technique is performed along d_i to provide a suitable stepsize α_k (Step 3). Otherwise, the direction $-d_i$ is considered (Step 2).
- Step 2 is similar to Step 1, with d_i replaced by $-d_i$. In this case, if the trial point $x_k - \alpha d_i$ does not produce a sufficient decrease of f then the stepsize α_k is set equal to zero and the scalar $\tilde{\alpha}_k^i$ is reduced. In this way, when the directions d_i and $-d_i$ will be considered again by the algorithm, the initial stepsize will be chosen in an interval containing smaller values.
- At Step 3 a suitable large stepsize α_k is computed by a derivative-free linesearch technique. Such technique derives from the ones proposed in [4]. It computes a sufficiently good estimate of the minimum of f along d_i without requiring any information on the slope of the objective function. The aim of this step is to exploit the good descent direction d_i identified at Step 1 or Step 2. Then, the scalar $\tilde{\alpha}_{k+1}^i$ is set equal to α_k. The motivation of this choice derives from the fact that the stepsize α_k produced by a linesearch technique should identify promising values for the initial stepsize when the direction d_i (or $-d_i$) will be investigated.
- At Step 4 the candidate point $\tilde{x}_{k+1}$ is generated.
- At Step 5 the new point x_{k+1} is produced and, for the next iteration, a coordinate direction is selected by following the cyclic order. At each iteration x_{k+1} can be always set equal to the candidate point $\tilde{x}_{k+1}$ produced at Step 5. The index h_k counts the number of successive iterations characterized by the fact that $x_{k+1} = \tilde{x}_{k+1}$. The condition $h_k \geq n$ indicates that the last n points generated have been produced by investigating the behaviour of f along all the coordinate directions and that, roughly speaking, the algorithm has obtained "enough" information about the local behaviour of the objective function. In this case, the algorithm admits the possibility of accepting as x_{k+1} any feasible point which produces a reduction of the objective function. Therefore, in these iterations the point x_{k+1} can be generated, for instance, by minimizing any approximation model of the objective function built by using the information progressively obtained by the algorithm. This

possibility does not affect the convergence properties of the algorithm, but it can increase its efficiency.

3. Convergence analysis

In this section we study the theoretical properties of Algorithm Model. In particular, we show that any accumulation point of the sequence generated by the proposed algorithm is a stationary point of problem (1).

First, we state the following proposition.

Proposition 1. *Suppose that f is bounded below on the feasible set $\mathcal{F}$ and let $\{x_k\}$ be the sequence produced by Algorithm Model. Then*:

(*i*) *Algorithm Model is well defined*;

(*ii*) *every limit point of $\{x_k\}$ belongs to $\mathcal{F}$*;

(*iii*) *we have*

$$\lim_{k\to\infty} \alpha_k = 0 \tag{5}$$

$$\lim_{k\to\infty} \tilde{\alpha}_k^i = 0 \quad \text{for } i = 1, \ldots, n. \tag{6}$$

Proof: In order to prove that Algorithm Model is well defined, we have to ensure that the Expansion Step, when performed along a direction d_i, with $i \in \{1, \ldots, n\}$, terminates in a finite number j of steps. At this aim, by contradiction we assume that for a given d_i

$$x_k + \delta^{-j}\alpha d_i \in \mathcal{F} \quad \text{for all } j,$$

and

$$f(x_k + \delta^{-j}\alpha d_i) < f(x_k) - \gamma(\delta^{-j}\alpha)^2\|d_i\|^2 \quad \text{for all } j,$$

which violates the assumption that f is bounded below on $\mathcal{F}$.

As regards assertion (ii), we have that the instructions of Algorithm Model imply that $x_k \in \mathcal{F}$ for all k. Since $\mathcal{F}$ is a closed set, the assertion is proved.

In order to prove (5), we split the iteration sequence $\{k\}$ into two parts, K' and K''. We identify with K' those iterations where

$$\alpha_k = 0 \tag{7}$$

and with K'' those iterations where $\alpha_k \neq 0$ is produced by Expansion Step. Then, Steps 3 and 5 imply

$$f(x_{k+1}) \leq f(x_k + \alpha_k d_i) \leq f(x_k) - \gamma(\alpha_k)^2\|d_i\|^2. \tag{8}$$

Taking into account the boundedness assumption on f, it follows from (8) that $\{f(x_k)\}$ tends to a limit $\bar{f}$. If K'' is an infinite subset, recalling that $\|d_i\| = 1$ we obtain

$$\lim_{k\to\infty, k\in K''} \alpha_k = 0. \tag{9}$$

Therefore, (7) and (9) imply (5).

In order to prove (6), for each $i \in \{1, \ldots, n\}$ we split the iteration sequence $\{k\}$ into three parts, K_1, K_2 and K_3. We identify with K_1 those iterations where Expansion Step has been performed using the direction d_i, namely

$$f(x_k + \alpha_k d_i) \leq f(x_k) - \gamma(\alpha_k)^2 \|d_i\|^2 \tag{10}$$

$$\tilde{\alpha}^i_{k+1} = \alpha_k. \tag{11}$$

We denote by K_2 those iterations where we have failed in decreasing the objective function along the directions d_i and $-d_i$. By the instructions of the algorithm it follows that for all $k \in K_2$

$$\tilde{\alpha}^i_{k+1} \leq \theta \tilde{\alpha}^i_k, \tag{12}$$

where $\theta \in (0, 1)$. Finally, K_3 denotes the iterations where the directions d_i and $-d_i$ are not used as search directions. Then, for $k \in K_3$ we have

$$\tilde{\alpha}^i_{k+1} = \tilde{\alpha}^i_k. \tag{13}$$

If K_1 is an infinite subset, from (11) and (5) we get that

$$\lim_{k \to \infty, k \in K_1} \tilde{\alpha}^i_{k+1} = 0. \tag{14}$$

Now, let us assume that K_2 is an infinite subset. For each $k \in K_2$, let m_k (we omit the dependence from i) be the biggest index such that $m_k < k$ and $m_k \in K_1$. Then we have:

$$\tilde{\alpha}^i_{k+1} \leq \theta^{\lfloor \frac{(k+1-m_k)}{n} \rfloor} \tilde{\alpha}^i_{m_k} \leq \tilde{\alpha}^i_{m_k} \tag{15}$$

(we can assume $m_k = 0$ if the index m_k does not exist, that is, K_1 is empty).

As $k \to \infty$ and $k \in K_2$, either K_1 is an infinite subset implying $m_k \to \infty$, or K_1 is finite implying $(k + 1 - m_k) \to \infty$. Hence, if K_2 is an infinite subset, (15) together with (14), or the fact that $\theta \in (0, 1)$, yields

$$\lim_{k \to \infty, k \in K_2} \tilde{\alpha}^i_{k+1} = 0. \tag{16}$$

Finally, let us consider the infinite subset K_3 (note that the instructions of the algorithm imply that K_3 is always an infinite set). The instructions of the algorithm imply that, for all $k \in K_3$ and k sufficiently large, there exists a nonnegative index $\nu_k \leq n$ such that

$$k - \nu_k \in K_1 \cup K_2$$
$$\tilde{\alpha}^i_{k+1} = \tilde{\alpha}^i_{k-\nu_k+1}.$$

Therefore, from (14) and (16), we get that

$$\lim_{k \to \infty, k \in K_3} \tilde{\alpha}^i_{k+1} = 0, \tag{17}$$

so that (6) is proved, and this concludes the proof. □

Now we are ready to prove the main convergence result.

Proposition 2. *Suppose that f is bounded below on the feasible set $\mathcal{F}$ and let $\{x_k\}$ be the sequence produced by Algorithm Model. Then every limit point of $\{x_k\}$ is a stationary point for problem* (1).

Proof: Let $\bar{x}$ be any limit point of $\{x_k\}$, that is

$$\lim_{k\to\infty, k\in K} x_k = \bar{x}, \tag{18}$$

where $K \subseteq \{0, 1, \ldots\}$. By (ii) of Proposition 1 we have that $\bar{x} \in \mathcal{F}$.

In order to prove the thesis, let us suppose by contradiction that $\bar{x}$ is not a stationary point. Therefore, there exists a point $\bar{y} \in \mathcal{F}$ such that

$$\nabla f(\bar{x})^T(\bar{y} - \bar{x}) < 0. \tag{19}$$

This implies that

$$\nabla_h f(\bar{x})(\bar{y} - \bar{x})^h < 0, \tag{20}$$

for some $h \in \{1, \ldots, n\}$. Let us define the sequence of scalars

$$\beta_k = (\bar{y} - x_k)^h.$$

Now, by (20) we have

$$\lim_{k\to\infty, k\in K} \beta_k \nabla f(x_k)^T e_h = \bar{\beta}\nabla f(\bar{x})^T e_h < 0, \tag{21}$$

where

$$|\bar{\beta}| = |(\bar{y} - \bar{x})^h| > 0. \tag{22}$$

For each $k \in K$, let us consider the smallest index $m_k^+ \geq k$ and the biggest index $m_k^- \leq k$ such that in the iterations m_k^+ and m_k^- the direction e_h or/and the direction $-e_h$ are investigated. The instructions imply

$$m_k^+ - k \leq n \qquad k - m_k^- \leq n. \tag{23}$$

Moreover we have that at least one of the following occurrence must happen

(i) all the points x_{k+j}, with $0 \leq j \leq (m_k^+ - k)$, are produced by Step 4, that is

$$x_{k+j+1} = x_{k+j} + \alpha_{k+j} e_{l(j)} \quad (\text{or } x_{k+j+1} = x_{k+j} - \alpha_{k+j} e_{l(j)});$$

(ii) all the points x_{k-j}, with $0 \leq j \leq k - m_k^-$, are produced by Step 4, that is

$$x_{k-j+1} = x_{k-j} + \alpha_{k-j} e_{l(j)} \quad (\text{or } x_{k-j+1} = x_{k-j} - \alpha_{k-j} e_{l(j)}),$$

where the index $l(j) \in \{1, \ldots, n\}$ identifies the search direction investigated. Now let $K' \subseteq K$ and $K'' \subseteq K$ be the subsets such that condition (i) is verified for all $k \in K'$ and condition (ii) is verified for all $k \in K''$. Suppose first that K' is infinite. For semplicity we rename K' as K. Hence, (iii) of Proposition 1 implies that, for $1 \leq j \leq m_k^+ - k$, it follows

$$\lim_{k\to\infty, k\in K} \|x_{k+j} - x_{k+j-1}\| = 0, \tag{24}$$

from which, recalling (23), we get

$$\lim_{k\to\infty, k\in K} x_{m_x^+} = \bar{x}. \tag{25}$$

By (21), (22) and (25) we obtain

$$\lim_{k\to\infty, k\in K} \beta_{m_k^+} = \bar{\beta}. \tag{26}$$

Conditions (21), (25) and (26) imply that for $k \in K$ and k sufficiently large the direction $-\mathrm{sign}(\beta_{m_k^+})e_h$ is an ascent direction in $x_{m_k^+}$ because it makes an acute angle with the gradient, and moreover, recalling (6) we can prove that for $k \in K$ and k sufficiently large

$$f\left(x_{m_k^+} - \mathrm{sign}\big(\beta_{m_k^+}\big)\frac{\tilde{\alpha}^h_{m_k^+ + 1}}{\theta} e_h\right) > f\big(x_{m_k^+}\big). \tag{27}$$

In fact, by applying the Mean Value Theorem we have

$$\begin{aligned} &f\left(x_{m_k^+} - \mathrm{sign}\big(\beta_{m_k^+}\big)\frac{\tilde{\alpha}^h_{m_k^+ + 1}}{\theta} e_h\right) - f\big(x_{m_k^+}\big) \\ &\quad = -\mathrm{sign}\big(\beta_{m_k^+}\big)\frac{\tilde{\alpha}^h_{m_k^+ + 1}}{\theta}[\nabla f(\bar{x})^T e_h + (\nabla f(\xi_k)^T e_h - \nabla f(\bar{x})^T e_h)], \end{aligned}$$

where $\xi_k = x_{m_k^+} - \lambda_k \mathrm{sign}(\beta_{m_k^+})\frac{\tilde{\alpha}^h_{m_k^+ + 1}}{\theta} e_h$, with $\lambda_k \in (0, 1)$. Then, since $\|\xi_k - x_{m_k^+}\| \to 0$, recalling (21), (26) and the continuity of ∇f, it follows that (27) holds. Hence, for k sufficiently large, at the m_k^+-th iteration, the instructions of the algorithm (in particular, Step 2) imply that the direction $\mathrm{sign}(\beta_{m_k^+})e_h$ is investigated. Since $x_k \in \mathcal{F}$ and $l^h \leq \bar{y}^h \leq u^h$ it follows

$$x_{m_k^+} + t\,\mathrm{sign}\big(\beta_{m_k^+}\big)e_h = x_{m_k^+} + \frac{t}{\big|\beta_{m_k^+}\big|}\big(\bar{y} - x_{m_k^+}\big)^h e_h \in \mathcal{F} \quad \forall t \in \big[0, \big|\beta_{m_k^+}\big|\big]. \tag{28}$$

Now, by (iii) of Proposition 1 we have that $\alpha_k \to 0$ and $\tilde{\alpha}^h_{m_k^+} \to 0$ for $k \to \infty$, and hence, for $k \in K$ and k sufficiently large, (22), (26) and (28) yield

$$x_{m_k^+} + \mathrm{sign}\big(\beta_{m_k^+}\big)\frac{\tilde{\alpha}^h_{m_k^+ + 1}}{\theta} e_h \in \mathcal{F} \tag{29}$$

and

$$x_{m_k^+} + \operatorname{sign}\big(\beta_{m_k^+}\big)\frac{\alpha^h_{m_k^+}}{\delta}e_h \in \mathcal{F} \tag{30}$$

Therefore we have that either

$$\alpha_{m_k^+} = 0,\ (29)\ \text{ holds and}$$
$$f\left(x_{m_k^+} + \operatorname{sign}\big(\beta_{m_k^+}\big)\frac{\tilde{\alpha}^h_{m_k^+ +1}}{\theta}e_h\right) > f\big(x_{m_k^+}\big) - \gamma\left(\frac{\tilde{\alpha}^h_{m_k^+}}{\theta}\right)^2. \tag{31}$$

or

$$\alpha_{m_k^+} \neq 0,\ (30)\ \text{ holds and}$$
$$f\left(x_{m_k^+} + \operatorname{sign}\big(\beta_{m_k^+}\big)\frac{\alpha_{m_k^+}}{\delta}e_h\right) > f\big(x_{m_k^+}\big) - \gamma\left(\frac{\alpha_{m_k^+}}{\delta}\right)^2. \tag{32}$$

By applying the Mean Value Theorem in (31) and (32) we can write either

$$\operatorname{sign}\big(\beta_{m_k^+}\big)\nabla f\big(u_{m_k^+}\big)^T e_h \geq -\frac{\tilde{\alpha}^h_{m_k^+ +1}}{\theta}$$

or

$$\operatorname{sign}\big(\beta_{m_k^+}\big)\nabla f\big(v_{m_k^+}\big)^T e_h > -\frac{\alpha_{m_k^+}}{\delta}$$

where

$$u_{m_k^+} = x_{m_k^+} + \operatorname{sign}\big(\beta_{m_k^+}\big)\lambda_k^1\frac{\tilde{\alpha}^h_{m_k^+ +1}}{\theta}e_h,$$
$$v_{m_k^+} = x_{m_k^+} + \operatorname{sign}\big(\beta_{m_k^+}\big)\lambda_k^2\frac{\alpha_{m_k^+}}{\delta}e_h,$$

with $\lambda_k^1, \lambda_k^2 \in (0, 1)$.

Therefore, taking the limits for $k \to \infty$ and $k \in K$, by using (5), (6) and the continuity assumption on ∇f, it follows

$$\operatorname{sign}(\bar{\beta})\nabla f(\bar{x})^T e_h \geq 0,$$

which contradicts (21).

Now, if K'' is an infinite set then we can repeat the same reasonings by minor modifications and obtain a contradiction with (21). □

Remark 1. If we replace at Step 5 of Algorithm Model condition (4) with the following stronger condition

$$f(x_{k+1}) \leq f(\tilde{x}_{k+1}) \quad \text{and} \quad x_{k+1} \in \mathcal{F} \tag{33}$$
$$\|x_{k+1} - x_k\| \leq \max\{\Delta f_k, \rho\alpha_k\} \tag{34}$$

where $\Delta f_k = f(x_{k+1}) - f(x_k)$ and $\rho \in (0, 1)$, we obtain that the results of Proposition 1 and Proposition 2 still hold with the additional property of the sequence $\{x_k\}$:

$$\lim_{k\to\infty} \|x_{k+1} - x_k\| = 0. \tag{35}$$

In fact, since f is bounded below on $\mathcal{F}$ and the sequence $\{f(x_k)\}$ is decreasing, we have that $\Delta f_k \to 0$. On the other hand, by (5) of Proposition 1 we have that $\alpha_k \to 0$. Therefore, (35) follows immediately.

4. Convergence analysis in presence of noise

In this section, we consider the case where the values of the objective function of problem (1) are corrupted by the presence of noise. In other words, we can only observe the perturbation $\tilde{f}(x)$ given by

$$\tilde{f}(x) = f(x) + \Delta(x),$$

where $\Delta(x)$ represents the amount of noise. Furthermore, we assume that there exists a constant $\Delta > 0$ such that

$$|\Delta(x)| \le \Delta \quad \text{for every } x \in \mathcal{F}. \tag{36}$$

In Algorithm Model the value $f(x)$ has to be replaced by $\tilde{f}(x)$.

First of all, we remark that it is possible to show that Proposition 1 still hold also in presence of noise satisfying (36). Further results about the properties of the proposed algorithm can be stated under the assumption that the gradient of the objective function is Lipschitz continuous. In particular, it is possible to derive a bound on the norm of the reduced gradient. At this aim, for each iteration k we consider a ball B_k of radius

$$r_k = \frac{2}{\min\{\theta, \delta\}} \sum_{l=1}^{n} \tilde{\alpha}_k^l \tag{37}$$

about x_k (where θ and δ are user chosen parameters of the algorithm). For each iteration k such that $h_k \ge n$ we have that every point where the function has been evaluated in one of the preceding n iterations is contained in the ball B_k about x_k. In fact, as regards the generated points we have

$$\|x_k - x_{k-n}\| \le \sum_{l=1}^{n} \alpha_{k-1} \le \sum_{l=1}^{n} \tilde{\alpha}_k^l.$$

Then, in correspondence to each iteration $k - j$, the algorithm can sample the objective function along a given coordinate direction e_h in the interval

$$\left[x_{k-j} - \frac{\tilde{\alpha}_{k-j+1}^h}{\min\{\theta, \delta\}} e_h, x_{k-j} + \frac{\tilde{\alpha}_{k-j+1}^h}{\min\{\theta, \delta\}} e_h \right].$$

Then, we have the following proposition.

Proposition 3. *Suppose that f is bounded below on the feasible set $\mathcal{F}$ and that its gradient is Lipschitz continuous (with constant L) on $\mathcal{F}$. Let $\{x_k\}$ be the sequence produced by* Algorithm Model *and let $\Delta_k = \sup_{x \in B_k} |\Delta(x)|$. Then, there exist two constants $c_1, c_2 > 0$ such that, for each iteration k for which $h_k \geq n$, we have*

$$\|\nabla^{red} f(x_k)\| \leq c_1 \max_{l=1,\ldots,n} \left\{\tilde{\alpha}_k^l\right\} + c_2 \frac{\Delta_k}{\min_{l=1,\ldots,n} \left\{\tilde{\alpha}_k^l\right\}}.$$

Proof: Let us consider the i-th component of $\nabla^{red} f(x_k)$. Now let $i(k) \leq k$ be the biggest index such that at the $i(k)$–iteration the direction e_i (and/or $-e_i$) has been investigated.

First of all, we note that by the instructions of the algorithm, and recalling the assumption $h_k \geq n$, it follows that

$$k - i(k) \leq n,$$

and by the definitions of B_k and Δ_k we have that the objective function values computed in the iterations $i(k), i(k)+1, \ldots, k$ are corrupted by an amount of noise bounded by Δ_k.

Now we distinguish the following cases:

(i) $x_k^i = u^i$
(ii) $x_k^i = l^i$
(iii) $l^i < x_k^i < u^i$

Case (i). In this case we have either

(ia) $x_{i(k)}^i = u^i$

or

(ib) $x_{i(k)}^i < u^i$.

In case (ia), as $x_k^i = x_{i(k)}^i$, from the instructions of the algorithm we have $\alpha_{i(k)} = 0$, $\tilde{\alpha}_{i(k)+1}^i = \tilde{\alpha}_k^i$, and

$$f\left(x_{i(k)} - \frac{\tilde{\alpha}_k^i}{\theta} e_i\right) > f\left(x_{i(k)}\right) - \gamma \left(\frac{\tilde{\alpha}_k^i}{\theta}\right)^2 - 2\Delta_k.$$

By applying the Mean Value Theorem we obtain

$$-\frac{\tilde{\alpha}_k^i}{\theta} \nabla f\left(u_{i(k)}\right)^T e_i > -\gamma \left(\frac{\tilde{\alpha}_k^i}{\theta}\right)^2 - 2\Delta_k,$$

where $u_{i(k)} = x_{i(k)} - \lambda_{i(k)} \frac{\tilde{\alpha}_k^i}{\theta} e_i$ with $\lambda_{i(k)} \in (0, 1)$. Then, we can write

$$\left[\nabla f\left(u_{i(k)}\right) - \nabla f(x_k) + \nabla f(x_k)\right]^T e_i < \gamma \frac{\tilde{\alpha}_k^i}{\theta} + \frac{2\Delta_k \theta}{\tilde{\alpha}_k^i},$$

from which, taking into account the Lipschitz assumption on ∇f, it follows

$$\begin{aligned}\nabla f(x_k)^T e_i &< \gamma \frac{\tilde{\alpha}_k^i}{\theta} + \frac{2\Delta_k \theta}{\tilde{\alpha}_k^i} + L\left\|x_k - u_{i(k)}\right\| \\ &\leq \gamma \frac{\tilde{\alpha}_k^i}{\theta} + \frac{2\Delta_k \theta}{\tilde{\alpha}_k^i} + L\left\|x_k - x_{i(k)}\right\| + L\frac{\tilde{\alpha}_k^i}{\theta}. \end{aligned} \tag{38}$$

We have

$$x_k = x_{i(k)} + \sum_{j=0}^{k-i(k)-1} \alpha_{i(k)+j} d_{i(k)+j},$$

where $d_{i(k)+j} \in \{e_1, -e_1, \ldots, e_n, -e_n\}$. For each j such that $\alpha_{i(k)+j} \neq 0$, recalling the instructions of the algorithm, we have that there exists an index $l \in \{1, \ldots, n\}$ such that $\tilde{\alpha}_{i(k)+j+1}^l = \alpha_{i(k)+j}$, and $\tilde{\alpha}_k^l = \tilde{\alpha}_{i(k)+j+1}^l$. Therefore, it follows

$$\alpha_{i(k)+j} \leq \max_{l=1,\ldots,n} \left\{\tilde{\alpha}_k^l\right\},$$

and we can write

$$\left\|x_k - x_{i(k)}\right\| \leq n \max_{l=1,\ldots,n} \left\{\tilde{\alpha}_k^l\right\}.$$

From (38) we get

$$\nabla_i^{red} f(x_k) < \frac{(\gamma + L(n+1))}{\theta} \max_{l=1,\ldots,n} \left\{\tilde{\alpha}_k^l\right\} + \frac{2\Delta_k \theta}{\min_{l=1,\ldots,n} \left\{\tilde{\alpha}_k^l\right\}}. \tag{39}$$

In case (ib), we have $\alpha_{i(k)} \neq 0$, $\tilde{\alpha}_{i(k)+1}^i = \alpha_{i(k)} = \tilde{\alpha}_k^i$, and

$$f\left(x_{i(k)} + \tilde{\alpha}_k^i e_i\right) \leq f\left(x_{i(k)}\right) - \gamma\left(\tilde{\alpha}_k^i\right)^2 + 2\Delta_k \leq f\left(x_{i(k)}\right) + \gamma\left(\tilde{\alpha}_k^i\right)^2 + 2\Delta_k.$$

Then, by applying the Mean Value Theorem, we obtain

$$\nabla f\left(v_{i(k)}\right)^T e_i < \gamma \tilde{\alpha}_k^i + \frac{2\Delta_k}{\tilde{\alpha}_k^i}.$$

where $v_{i(k)} = x_{i(k)} + \lambda_{i(k)} \tilde{\alpha}_k^i e_i$ with $\lambda_{i(k)} \in (0, 1)$. Then, by repeating the preceding reasonings we obtain

$$\nabla_i^{red} f(x_k) < (\gamma + L(n+1)) \max_{l=1,\ldots,n} \left\{\tilde{\alpha}_k^l\right\} + \frac{2\Delta_k}{\min_{l=1,\ldots,n} \left\{\tilde{\alpha}_k^l\right\}} \tag{40}$$

Case (ii). It is analogous to Case (i), so that conditions (39) and (40) hold.

Case (iii). In this case we have either

(iiia) $x^i_{i(k)} = x^i_k$

or

(iiib) $x^i_{i(k)} \neq x^i_k$.

In case (iiia), from the instructions of the algorithm, recalling that $\tilde{\alpha}^i_{i(k)+1} = \tilde{\alpha}^i_k = \theta\alpha$, where α is the initial stepsize, we have

$$f\left(x_{i(k)} + \frac{\tilde{\alpha}^i_k}{\theta} e_i\right) \geq f\left(x_{i(k)}\right) - \gamma\left(\frac{\tilde{\alpha}^i_k}{\theta}\right)^2 - 2\Delta_k \tag{41}$$

$$f\left(x_{i(k)} - \frac{\tilde{\alpha}^i_k}{\theta} e_i\right) \geq f\left(x_{i(k)}\right) - \gamma\left(\frac{\tilde{\alpha}^i_k}{\theta}\right)^2 - 2\Delta_k. \tag{42}$$

By applying the Mean Value Theorem we obtain

$$\nabla_i f\left(u_{i(k)}\right)^T e_i \geq -\gamma\left(\frac{\tilde{\alpha}^i_k}{\theta}\right)^2 - \frac{2\Delta_k\theta}{\tilde{\alpha}^i_k} \tag{43}$$

$$\nabla_i f\left(v_{i(k)}\right)^T e_i \leq \gamma\left(\frac{\tilde{\alpha}^i_k}{\theta}\right)^2 + \frac{2\Delta_k\theta}{\tilde{\alpha}^i_k}, \tag{44}$$

where $u_{i(k)} = x_{i(k)} + \lambda^1_{i(k)} \frac{\tilde{\alpha}^i_k}{\theta} e_i$, $v_{i(k)} = x_{i(k)} - \lambda^2_{i(k)} \frac{\tilde{\alpha}^i_k}{\theta} e_i$, with $\lambda^1_{i(k)}, \lambda^2_{i(k)} \in (0, 1)$.

From (43), taking into account the Lipschitz assumption on ∇f we get

$$\begin{aligned}
\nabla f(x_k)^T e_i &\geq -\gamma\frac{\tilde{\alpha}^i_k}{\theta} - \frac{2\Delta_k\theta}{\tilde{\alpha}^i_k} - L\left\|x_k - u_{i(k)}\right\| \\
&\geq -\gamma\frac{\tilde{\alpha}^i_k}{\theta} - \frac{2\Delta_k\theta}{\tilde{\alpha}^i_k} - L\left\|x_k - x_{i(k)}\right\| - L\frac{\tilde{\alpha}^i_k}{\theta} \\
&\geq -\gamma\frac{\tilde{\alpha}^i_k}{\theta} - \frac{2\Delta_k\theta}{\tilde{\alpha}^i_k} - nL\max_{l=1,\ldots,n}\left\{\tilde{\alpha}^l_k\right\} - L\frac{\tilde{\alpha}^i_k}{\theta} \\
&= -(\gamma + L)\frac{\tilde{\alpha}^i_k}{\theta} - \frac{2\Delta_k\theta}{\tilde{\alpha}^i_k} - nL\max_{l=1,\ldots,n}\left\{\tilde{\alpha}^l_k\right\}.
\end{aligned}$$

Hence, it follows

$$\nabla^{red}_i f(x_k) \geq -\frac{(\gamma + L(n+1))}{\theta}\max_{l=1,\ldots,n}\left\{\tilde{\alpha}^l_k\right\} - \frac{2\Delta_k\theta}{\min_{l=1,\ldots,n}\left\{\tilde{\alpha}^l_k\right\}}. \tag{45}$$

From (44), by repeating the same reasonings, we obtain

$$\nabla^{red}_i f(x_k) \leq \frac{(\gamma + L(n+1))}{\theta}\max_{l=1,\ldots,n}\left\{\tilde{\alpha}^l_k\right\} + \frac{2\Delta_k\theta}{\min_{l=1,\ldots,n}\left\{\tilde{\alpha}^l_k\right\}}. \tag{46}$$

From (45) and (46) it follows

$$\left|\nabla_i^{red} f(x_k)\right| \leq \frac{(\gamma + L(n+1))}{\theta} \max_{l=1,\ldots,n} \left\{\tilde{\alpha}_k^l\right\} + \frac{2\Delta_k \theta}{\min_{l=1,\ldots,n} \left\{\tilde{\alpha}_k^l\right\}}. \tag{47}$$

Let us consider the case (iiib). Without loss of generality, we can assume that in this case we have $\alpha_{i(k)} \neq 0$, $\tilde{\alpha}_{i(k)+1}^i = \tilde{\alpha}_k^i = \alpha_{i(k)}$,

$$f\left(x_{i(k)} + \tilde{\alpha}_k^i e_i\right) \leq f\left(x_{i(k)}\right) - \gamma\left(\tilde{\alpha}_k^i\right)^2 + 2\Delta_k$$

and

$$f\left(x_{i(k)} + \bar{\alpha}_k^i e_i\right) > f\left(x_{i(k)}\right) - \gamma\left(\bar{\alpha}_k^i\right)^2 - 2\Delta_k,$$

with

$$\bar{\alpha}_k^i = \frac{\tilde{\alpha}_k^i}{\delta_{i(k)}}$$

where $\delta_{i(k)} = \delta$ if $x_{i(k)} + \frac{\tilde{\alpha}_k^i}{\delta} e_i \in \mathcal{F}$, and $\delta_{i(k)} \in (\delta, 1)$ otherwise (in this case $\delta_{i(k)}$ is such that $x_{i(k)} + \frac{\tilde{\alpha}_k^i}{\delta_{i(k)}} e_i \in \partial\mathcal{F}$). By applying the Mean Value Theorem, we can write

$$\nabla f\left(u_{i(k)}\right)^T e_i \leq -\gamma\tilde{\alpha}_k^i + 2\frac{\Delta_k}{\tilde{\alpha}_k^i} \leq \gamma\tilde{\alpha}_k^i + 2\frac{\Delta_k}{\alpha_k^i},$$

and

$$\nabla f\left(v_{i(k)}\right)^T e_i > -\gamma\frac{\tilde{\alpha}_k^i}{\delta_{i(k)}} - 2\frac{\Delta_k \delta_{i(k)}}{\tilde{\alpha}_k^i},$$

where $u_{i(k)} = x_{i(k)} + \lambda_{i(k)}^1 \alpha_{i(k)} e_i$, $v_{i(k)} = x_{i(k)} + \lambda_{i(k)}^2 \frac{\tilde{\alpha}_k^i}{\delta_{i(k)}} e_i$ with $\lambda_{i(k)}^1, \lambda_{i(k)}^2 \in (0, 1)$.

Taking into account the Lipschitz assumption on ∇f we can write

$$\begin{aligned}
\nabla f(x_k)^T e_i &\leq \gamma\tilde{\alpha}_k^i + \frac{2\Delta_k}{\tilde{\alpha}_k^i} + L\left\|x_k - u_{i(k)}\right\| \\
&\leq \gamma\tilde{\alpha}_k^i + \frac{2\Delta_k}{\tilde{\alpha}_k^i} + L\left\|x_k - x_{i(k)}\right\| + L\tilde{\alpha}_{i(k)+1}^i \\
&\leq (\gamma + L)\tilde{\alpha}_k^i + \frac{2\Delta_k}{\tilde{\alpha}_k^i} + nL \max_{l=1,\ldots,n} \left\{\tilde{\alpha}_k^l\right\}, \\
\nabla f(x_k)^T e_i &> -\gamma\frac{\tilde{\alpha}_k^i}{\delta_{i(k)}} - \frac{2\Delta_k \delta_{i(k)}}{\tilde{\alpha}_k^i} - L\left\|x_k - v_{i(k)}\right\| \\
&\geq -\gamma\frac{\tilde{\alpha}_k^i}{\delta} - \frac{2\Delta_k}{\tilde{\alpha}_k^i} - L\left\|x_k - x_{i(k)}\right\| - L\tilde{\alpha}_k^i \\
&\geq -(\gamma + L)\frac{\tilde{\alpha}_k^i}{\delta} - \frac{2\Delta_k}{\tilde{\alpha}_k^i} - nL \max_{l=1,\ldots,n} \left\{\tilde{\alpha}_k^l\right\}.
\end{aligned}$$

Then we have

$$\left|\nabla_i^{red} f(x_k)\right| \leq \frac{(\gamma + L(n+1))}{\delta} \max_{l=1,\ldots,n} \left\{\tilde{\alpha}_k^l\right\} + \frac{2\Delta_k}{\min_{l=1,\ldots,n} \left\{\tilde{\alpha}_k^l\right\}} \tag{48}$$

Finally, from (39), (40) and (48) we obtain

$$\|\nabla^{red} f(x_k)\| \leq n^{1/2} \left(\frac{(\gamma + L(n+1))}{\max\{\theta, \delta\}} \max_{l=1,\ldots,n} \left\{\tilde{\alpha}_k^l\right\} + \frac{2\Delta_k}{\min_{l=1,\ldots,n} \left\{\tilde{\alpha}_k^l\right\}} \right),$$

and this concludes the proof. □

Remark 2. We note that if at each iteration k we set $x_{k+1} = \tilde{x}_{k+1}$ (namely $x_{k+1} = x_k + \alpha_k d_k$, where $d_k \in \{e_1, \ldots, e_n, -e_1, \ldots, -e_n\}$ and where α_k is equal to zero or is produced by the Expansion Step) then the assertion of Proposition 3 holds for all k. In fact, in this case we have $h_k \geq n$ for all $k > n$.

Finally, by requiring a stronger assumption on the noise, we have the following convergence result.

Proposition 4. *Suppose that f is bounded below on the feasible set $\mathcal{F}$ and that its gradient is Lipschitz continuous (with constant L) on $\mathcal{F}$. Let $\{x_k\}$ be the sequence produced by* Algorithm Model. *Then if*

$$\lim_{k\to\infty} \frac{\Delta_k}{\min_{l=1,\ldots,n} \left\{\tilde{\alpha}_k^l\right\}} = 0, \tag{49}$$

then every limit point of $\{x_k\}$ is a stationary point for problem (1).

Proof: Let $\bar{x}$ be any limit point of $\{x_k\}$, that is

$$\lim_{k\to\infty, k\in K} x_k = \bar{x},$$

where $K \subseteq \{0, 1, \ldots\}$. In order to prove the thesis, by contradiction, let us suppose that $\bar{x}$ is not a stationary point for problem (1), that is

$$\|\nabla^{red} f(\bar{x})\| \geq \epsilon, \tag{50}$$

for some $\epsilon > 0$.

From the instructions of the algorithm we have that, for each $k \in K$, there exists an index ν_k, with $0 \leq \nu_k < n$ such that $h_{k+\nu_k} \geq n$ and

$$x_{k+j} = x_{k+j-1} + \alpha_{k+j-1} e_{l(j)} \quad \text{for } j = 1, \ldots, \nu_k. \tag{51}$$

By Proposition 3 we have

$$\|\nabla^{red} f(x_{k+\nu_k})\| \le c_1 \max_{l=1,\dots,n} \left\{\tilde{\alpha}^l_{k+\nu_k}\right\} + c_2 \frac{\Delta_{k+\nu_k}}{\min_{l=1,\dots,n} \left\{\tilde{\alpha}^l_{k+\nu_k}\right\}},$$

hence, recalling (49) and that point (iii) of Proposition 1 still holds also in presence of bounded noise, it follows

$$\lim_{k\to\infty, k\in K} \|\nabla^{red} f(x_{k+\nu_k})\| = 0. \tag{52}$$

Then, by using the Lipschitz assumption on ∇f, we can write

$$\|\nabla^{red} f(x_k) - \nabla^{red} f(x_{k+\nu_k})\| \le L\|x_k - x_{k+\nu_k}\| \le L \sum_{j=1}^{\nu_k} \alpha_{k+j-1},$$

from which, recalling again (iii) of Proposition 1, the continuity assumption of ∇f and (52) we obtain

$$\|\nabla^{red} f(\bar{x})\| = 0,$$

which contradicts (50). □

5. Preliminary computational results

In order to evaluate a possible practical interest of the proposed algorithm model, we have used 41 standard test problems of dimension n ranking form 2 to 10. These problems are selected from two test sets: the first one is made of 23 problems defined in [3], which are obtained from the set of functions suggested in [10]. The second set consists of the all box constrained problems of the Hock-Schittkowski collections [5, 12]. Furthermore, we have considered a real test problem which derives from an application [13] regarding the design of instruments for magnetic resonance.

On the basis of the proposed algorithm model, we have implemented in Fortran code two different algorithms.

Algorithm 1. *The point x_{k+1} produced at Step 5 is set equal to $\tilde{x}_{k+1}$, therefore, either $x_{k+1} = x_k$ or $x_{k+1} = x_k + \alpha_k e_i(x_k - \alpha_k e_i)$, where $i \in \{1, \dots, n\}$.*

Algorithm 2. *It tries to produce the point x_{k+1} at Step 5 by using an approximation scheme.*

Now we describe more in detail the choices made in these implementations.

5.1. Choice of parameters

The parameters which appear in the algorithm model have been set as follows.

$$\gamma = 10^{-6} \quad \delta = 0.25 \quad \theta = 0.5 \quad \tilde{\alpha}^i_0 = 0.5 \quad i = 1, \dots, n.$$

We note that we have not performed an extensive empirical tuning of the parameters of the algorithm. We have adapted choices usually adopted in linesearch techniques of gradient-based algorithms.

5.2. *Approximation scheme used in Algorithm 2*

In Algorithm 2 we have made an initial attempt to exploit the information on the objective function obtained at the previous iterations in order to produce a "better" point x_{k+1} with respect to $\tilde{x}_{k+1}$. In particular, every n iterations, we construct (when the algorithm has produced a sufficient number of points "close" to x_k) a simple quadratic model

$$q(x) = \frac{1}{2}x^T Qx + c^T x + b$$

of the objective function. The $N = n(n+1)/2 + n + 1$ free parameters of the model are determined by minimizing the error

$$\sum_{j=1}^{M} (q(x_j) - f(x_j))^2 \tag{53}$$

being $M > 0$ and $x_1, \ldots, x_M$ the last M points where f has been evaluated and such that

$$\tilde{l}_k \leq x_j \leq \tilde{u}_k,$$

with

$$\begin{aligned} \tilde{l}_k^i &= \max\left\{l_i, x_k^i - 100\tilde{\alpha}_k^i\right\} \quad i = 1, \ldots, n \\ \tilde{u}_k^i &= \max\left\{u_i, x_k^i + 100\tilde{\alpha}_k^i\right\} \quad i = 1, \ldots, n. \end{aligned}$$

Then, we apply a minimization method for computing a stationary point of the defined quadratic problem

$$\min_{\tilde{l} \leq x \leq \tilde{u}} q(x) \tag{54}$$

and the obtained point is accepted if it produces a reduction of f. We note that the box $[\tilde{l}, \tilde{u}]$ has the role of selecting points which are not too far from the current point x_k.

In our tests we have chosen $M = N + 5$ and we have used the routines F04JAF and E04NAF of NAG library to solve, respectively, the linear least squares problem (53) and the box constrained quadratic problem (54).

5.3. *Stopping criterion*

Since the definition of an efficient stopping criterion is out of the scope of the work, we have adapted the same approach proposed in [3]. Let f_0 be the value of f at the starting

point x_0 and f^* be the best known function value. Then we introduce the quotient

$$q_k = \frac{f(x_k) - f^*}{f_0 - f^*},$$

which can be considered a measure of the speed of convergence and we have stopped an algorithm whenever

$$q_k \leq \epsilon, \tag{55}$$

where $\epsilon > 0$ is a prefixed value. By using different values of ϵ we can have an idea on the efficiency of an algorithm. However, in some test problems the global minimum is not the unique stationary point. Therefore, an algorithm could generate a sequence converging towards a stationary point $\bar{x}$, with $f(\bar{x}) > f^*$, and could never satisfy the criterion (55). To tackle this possible occurrence, we have introduced also the following stopping criterion

$$\max_{i=1,\ldots,n} \left\{\tilde{\alpha}_k^i\right\} \leq 10^{-5}. \tag{56}$$

Finally, we say that an algorithm *has failed* when it has performed a number $N_{max} = 1000$ of function evaluations without satisfying any of the two the stopping criteria.

5.4. *Numerical experience*

We have tested Algorithms 1 and 2 on the set of standard problems with three different values of ϵ in criterion (55), namely $\epsilon = 10^{-1}$, $\epsilon = 10^{-3}$, $\epsilon = 10^{-6}$. The complete results are reported in [9]. Here we describe some summaries of these results. In the following table we report the total number of function evaluations needed to solve the problems where both the algorithms have been able to satisfy criterion (55), the total number of failures and the total number of stops due to criterion (56).

From the results of Table 1 we can note that for $\epsilon = 10^{-1}$ both the algorithms have been able to solve all the test problems. For $\epsilon = 10^{-3}$ most of the test problems are still solved by the two algorithms. Whereas the case $\epsilon = 10^{-6}$ points out the utility of approximation techniques in a derivative-free algorithm. In fact, when the degree of required precision is high, it seems to be necessary to exploit as much as possible the information on the

Table 1. Cumulative results.

		Total n_f	No. of failures	No. of stops
$\epsilon = 10^{-1}$	Alg. 1	2045	0	1
	Alg. 2	1099	0	1
$\epsilon = 10^{-3}$	Alg. 1	3556	5	1
	Alg. 2	1974	3	2
$\epsilon = 10^{-6}$	Alg. 1	3375	10	5
	Alg. 2	1590	6	5

Table 2. Number of WINS.

		No. of wins in terms of n_f	No. of balances
$\epsilon = 10^{-1}$	Alg. 1	0	34
	Alg. 2	6	34
$\epsilon = 10^{-3}$	Alg. 1	3	22
	Alg. 2	10	22
$\epsilon = 10^{-6}$	Alg. 1	4	14
	Alg. 2	7	14

objective function obtained during the iterations of the algorithm. From Table 1 we can also note that the simple approximation scheme used in Algorithm 2 has allowed us a significant computational saving in terms of number of function evaluations.

Then we have analysed more in detail the behaviour of Algorithms 1 and 2. In particular, we say that an algorithm *wins* if the number of function evaluations required to solve a test problem is smaller or equal to the 95% of the number required by the other algorithm. In Table 2 we report the total number of wins.

Table 2 shows that in many solved problems the performances of the two algorithms are comparable and hence, that Algorithm 2 outperforms Algorithm 1 only in few problems. However, taking into account also the results of Table 1, we can conclude that in such problems the improvement due to the use of an approximation is considerable.

In order to have a first idea on the practical interest of the proposed approach, we have compared Algorithm 2 with a method using finite-differences gradients. As it is well known, methods of this class are very efficient in absence of noise. Therefore, the comparison is an hard test for Algorithm 2 which is a direct search method, namely a method which does not try to approximate explicitly the first order derivatives. In particular, we have used E04UCF routine of NAG library, which is a sequential quadratic programming method. Similarly to Algorithm 2 we have stopped the NAG routine whenever (55) has been satisfied (also in this case we have used three different values of ϵ: 10^{-1}, 10^{-3}, 10^{-6}). In some runs E04UCF routine was terminated by its stopping criterion and the vector returned $\bar{x}$ was a stationary point such that $f(\bar{x}) > f^*$. In two test problems, E04UCF was unable to produce a point different from the starting point. We have considered these cases as failures of E04UCF. The complete results regarding E04UCF are reported in [9].

Here, for brevity's sake, we report a summary of the comparisons between Algorithm 2 and E04UCF (in Tables 3 and 4). In particular, in Table 3 we report the total number of function evaluations needed to solve the problems where both the Algorithm 2 and the considered NAG routine have been able to satisfy criterion (55), the total number of failures and the total number of stops due to criterion (56) for Algorithm 2, and to the default stopping criterion for the NAG routine. In Table 4 we show the total number of wins in terms of number of function evaluations.

From the results of Table 3, we can note that for $\epsilon = 10^{-1}$ Algorithm 2 outperforms E04UCF. For $\epsilon = 10^{-3}$ it is still competitive in terms of number of function evaluations with E04UCF and the two algorithms are comparable in terms of failures. When $\epsilon = 10^{-6}$, the

Table 3. Cumulative results.

		Total n_f	No. of failures	No. of stops
$\epsilon = 10^{-1}$	Alg. 2	1069	0	1
	E04UCF	1601	2	1
$\epsilon = 10^{-3}$	Alg. 2	1989	3	2
	E04UCF	2052	2	4
$\epsilon = 10^{-6}$	Alg. 2	2381	6	5
	E04UCF	2195	2	4

Table 4. Number of WINS.

		No. of wins in terms of n_f	No. of balances
$\epsilon = 10^{-1}$	Alg. 2	29	1
	E04UCF	8	1
$\epsilon = 10^{-3}$	Alg. 2	21	1
	E04UCF	10	1
$\epsilon = 10^{-6}$	Alg. 2	15	0
	E04UCF	11	0

behaviour of Algorithm 2 is comparable with the one of E04UCF in terms of number of function evaluations. However, E04UCF is able to solve a (slightly) larger number of problems.

The preceding results, although far from being exhaustive, show a satisfactory behaviour of Algorithm 2 compared with a method using finite-differences derivatives. We recall that methods of this kind are efficient in the case that the objective function values are not affected by any noise, but they are not considered reliable for solving noisy problems. In fact, the behaviour of methods using finite-differences derivatives deteriorates also with a small noise. In particular, we have repeated the tests by considering function values given by

$$\tilde{f}(x) = f(x)(1 + \eta), \quad \eta \in N(0, \sigma^2),$$

where $N(0, \sigma^2)$ denotes a Gaussian distributed random number with zero mean and variance $\sigma^2 = 10^{-9}$. We note that the convergence analysis developed in Section 4 holds for bounded noise, while the one considered is unbounded. However, due to the small value of the variance, the noise can be considered bounded in practice. We have obtained the following results.

- **Algorithm 2:** for the three values of ϵ, the number of failures is not changed with respect to the noiseless case.
- **E04UCF routine:**
  - for $\epsilon = 10^{-1}$ the number of failures changes from 2 to 27;
  - for $\epsilon = 10^{-3}$ the number of failures changes from 2 to 32;
  - for $\epsilon = 10^{-6}$ the number of failures changes from 2 to 32.

5.5. *A real design problem*

We consider a real problem arising from an application [13] which regards the optimal design of apparati for magnetic resonance.

The application deals with the construction of apparati with reduced sizes to be used for clinical analysis of peripheral regions of the body. These apparati are based on resistive magnets that can be manufactured at very low cost. However, they require a good magnetic field uniformity in a large part of their volume, and the reduced sizes make it difficult to obtain this field uniformity.

In particular, the magnetic field is generated by four currents: I_1, I_2, I_3, I_4. The required field is B^0 and it must be as uniform as possible in a spherical region Ω at the center of the magnet. The vector of currents is denoted by $I = (I_1, I_2, I_3, I_4)^T$ and the z component of the magnetic field generated in a point r^j by $B_z(I; r^j)$. More formally, the problem is

$$\min_{\text{s.t. } 0 \le I \le U} f(I) = \sum_{j=1}^{N_p} [B_z(I; r^j) - B^0]^2 \tag{57}$$

The lower bounds ($0 \le I$) are imposed because the currents values must have the same sign for reasons related to the construction of the magnet. The upper bounds ($I \le U$) depend on the allowed power dissipation.

We note that, for each r^j, with $j = 1, \ldots, N_p$, the function $B_z(.; r^j) : R^4 \to R$ is not known analytically, but for each $I \in R^4$ the value $B_z(I; r^J)$ can be directly measured by a Gaussmeter. However, the evaluation of the objective function is not expensive.

In [13], neural models have been defined for approximating the functions $B_z(.; r^j)$, with $j = 1, \ldots, N_p$, by using a massive data set generated "offline". Then, a gradient-based method has been applied to problem (57), where the terms $B_z(I; r^j)$ have been replaced by the analytical neural model determined.

In practice some parameters characterizing the magnet will change over the time, and hence, the field uniformity obtained by the procedure proposed in [13] can become unacceptable. In this case, an "online" procedure is necessary for recomputing quickly the vector of currents starting from the currents previously determined "offline". This procedure should be able to solve problem (57) by using direct measurements for computing the values $B_z(I; r^J)$. Since the objective function values will be affected by an amount of noise (due to the direct measurements), the use of a method based on finite-differences gradients will be impractical, while the adoption of a derivative-free method appears suitable.

In order to evaluate the potentialities of this approach, we have applied Algorithm 2 to problem (57), where for each feasible vector I, the values $B_z(I; r^j)$ are determined by a simulation program. In particular, the stopping criterion (55) is well suited for this class of applications. In fact, the value f^* can correspond to an acceptable value of the field uniformity. Starting from a vector of initial currents such that $f_0 = 11.986$, assuming $f^* = 0.049$, we have used three different values of ϵ in the stopping criterion, and we have obtained the following results (where k and n_f are, respectively, the number of iterations and the number of function evaluations required to satisfy the stopping criterion (55), while f_k is the objective function attained).

$\epsilon = 10^{-1}$	$k = 32$	$n_f = 102$	$f_k = 0.067$
$\epsilon = 10^{-3}$	$k = 37$	$n_f = 120$	$f_k = 0.061$
$\epsilon = 10^{-6}$	$k = 377$	$n_f = 1060$	$f_k = 0.050$

To examine more in detail the behaviour of Algorithm 2, we also report some intermediate results. The results show the practicability of using Algorithm 2 to update the current values.

k	0	5	10	20	50	100	200	377
n_f	1	31	45	73	156	293	573	1060
f_k	11.9869	0.4751	0.2203	0.1237	0.05430	0.0515	0.0507	0.0500

In fact Algorithm 2 has been able to reobtain a good uniform field by performing a limited number of function evaluations.

References

1. E.J. Anderson and M.C. Ferris, "A direct search algorithm for optimization with noisy functions evaluations," Mathematical Programming Technical Report 96-11, Computer Sciences Department, University of Wisconsin, Madison, Wisconsin, 1996.
2. A.R. Conn, K. Scheinberg, and Ph.L. Toint, "Recent progress in unconstrained nonlinear optimization without derivatives," Mathematical Programming, vol. 79, pp. 397–415, 1997.
3. C. Elster and A. Neumaier, "A grid algorithm for bound constrained optimization of noisy functions," IMA Journal of Numerical Analysis, vol. 15, pp. 585–608, 1995.
4. L. Grippo, F. Lampariello, and S. Lucidi, "Global convergence and stabilization of unconstrained minimization methods without derivatives," Journal of Optimization Theory and Applications, vol. 56, no. 3, pp. 385–406, 1988.
5. W. Hock and K. Schittkowski, Test Examples for Nonlinear Programming Codes, Springer-Verlag: Berlin, 1981. Lectures Notes in Economics and Mathematical Systems, vol. 187.
6. C.T. Kelley, Iterative Methods for Optimization. Frontiers in Applied Mathematics, Society for Industrial and Applied Mathematics: Philadelphia, 1999.
7. R.M. Lewis and V. Torczon, "Pattern search methods for bound constrained minimization," SIAM Journal on Optimization, vol. 9, pp. 1082–1099, 1999.
8. S. Lucidi and M. Sciandrone, "On the global convergence of derivative free methods for unconstrained optimization," Technical Report 18-96, DIS, Università di Roma "La Sapienza".
9. S. Lucidi and M. Sciandrone, "A derivative-free algorithm for bound constrained optimization," R. 498 1999, IASI, Consiglio Nazionale delle Ricerche.
10. J.J. Moré, B.S. Garbow, and K.E. Hillstrom, "Testing unconstrained optimization software; ACM Trans. on Math. Software, vol. 7, pp. 17–41, 1981.
11. M.J.D. Powell, "Direct search algorithms for optimization calculations," Acta Numerica, vol. 7, pp. 287–336, 1998.
12. K. Schittkowski, More Test Examples for Nonlinear Programming Codes. Springer-Verlag: Berlin, 1987. Lectures Notes in Economics and Mathematical Systems, vol. 282.
13. M. Sciandrone, G. Placidi, L. Testa, and A. Sotgiu, "Compact low field magnetic resonance imaging magnet: Design and optimization," Review of Scientific Instruments, vol. 71, no. 3, pp. 1534–1538, 2000.
14. V. Torczon, "On the convergence of pattern search methods, "SIAM Journal on Optimization, vol. 7, pp. 1–25, 1997.

15. P. Tseng, "Fortified-descent simplicial search method: A general approach," SIAM Journal on Optimization, vol. 10, pp. 269–288, 2000.
16. M.H. Wright, "Direct search methods: Once scorned, now respectable," Numerical Analysis 1995, "Proceedings of the 1995 Dundee Biennial Conference in Numerical Analysis, D.F. Griffiths and G.A. Watson (Eds.) Addison-Wesley Longman: Harlow, UK, 1996, pp. 191–208.
17. S.K. Zavriev, "On the global optimization properties of finite-differences local descent algorithms," Journal of Global Optimization, vol. 3, pp. 67–78, 1993.