ISTITUTO DI ANALISI DEI SISTEMI ED INFORMATICA
"Antonio Ruberti"
CONSIGLIO NAZIONALE DELLE RICERCHE

G. Fasano, G. Liuzzi, S. Lucidi, F. Rinaldi

A LINESEARCH-BASED DERIVATIVE-FREE APPROACH FOR NONSMOOTH OPTIMIZATION

R. 1, January 2013

Giovanni Fasano – Università Ca'Foscari Venezia, Dipartimento di Management, San Giobbe, Cannaregio 873, 30121 Venice, Italy. fasano@unive.it.

Giampaolo Liuzzi – Consiglio Nazionale delle Ricerche, Istituto di Analisi dei Sistemi ed Informatica "A. Ruberti", Viale Manzoni 30, 00185 Rome, Italy. liuzzi@iasi.cnr.it.

Stefano Lucidi – "Sapienza" Università di Roma, Dipartimento di Ingegneria Informatica Automatica e Gestionale "A. Ruberti", Via Ariosto 25, 00185 Rome, Italy. lucidi@dis.uniroma1.it.

Francesco Rinaldi – Università di Padova, Dipartimento di Matematica, Via Trieste 63, 35121 Padua, Italy. rinaldi@math.unipd.it.

This work has been partially funded by the Italian national project RITMARE 2012-2016.

ISSN: 1128–3378

Collana dei Rapporti dell'Istituto di Analisi dei Sistemi ed Informatica "Antonio Ruberti", CNR

viale Manzoni 30, 00185 ROMA, Italy

tel. ++39-06-77161
fax ++39-06-7716461
email: iasi@iasi.cnr.it
URL: http://www.iasi.cnr.it

Abstract

In this paper, we propose new linesearch-based methods for nonsmooth constrained optimization problems when first-order information on the problem functions is not available. In the first part, we describe a general framework for bound-constrained problems and analyze its convergence towards stationary points, using the Clarke-Jahn directional derivative. In the second part, we consider inequality constrained optimization problems where both objective function and constraints can possibly be nonsmooth. In this case, we first split the constraints into two subsets: difficult general nonlinear constraints and simple bound constraints on the variables. Then, we use an exact penalty function to tackle the difficult constraints and we prove that the original problem can be reformulated as the bound-constrained minimization of the proposed exact penalty function. Finally, we use the framework developed for the bound-constrained case to solve the penalized problem, and we prove that every accumulation point of the generated sequence of iterates is a stationary point of the original constrained problem. In the last part of the paper, we report extended numerical results on both bound-constrained and nonlinearly constrained problems, showing that our approach is promising when compared to some state-of-the-art codes from the literature.

Key words: Derivative-free optimization, Lipschitz optimization, Exact penalty functions, Inequality constrained optimization, Stationarity conditions

1. Introduction

In this paper, we consider the optimization of a nonsmooth function $f : \mathbb{R}^n \to \mathbb{R}$ over a feasible set defined by lower and upper bounds on the variables and, possibly, by nonlinear and nonsmooth inequality constraints $g : \mathbb{R}^n \to \mathbb{R}^m$, namely

$$\begin{array}{ll} \min & f(x) \\ s.t. & g(x) \leq 0 \\ & l \leq x \leq u, \end{array}$$

where $l, u \in \mathbb{R}^n$, and $l < u$. We observe that, to our purposes, the fact that l and u must be finite might be relaxed by suitable assumptions on the objective function of the problem. We assume that the problem functions (though nonsmooth) are Lipschitz continuous and that first-order information is unavailable, or impractical to obtain (e.g. when problem functions are expensive to evaluate or somewhat noisy).
Such a kind of optimization problems encompasses many real-world problems arising in different fields like e.g. computational mathematics, physics and engineering, and presents a twofold difficulty. On the one hand, problem functions are typically of the black-box type, so that first order information is unavailable; on the other hand, the functions present a certain level of nonsmoothness (see e.g. [3], [9] and [23]).
In [4] and [5] the Mesh Adaptive Direct Search (MADS) class of algorithms is introduced, where an asymptotically dense set of directions is generated and combined with an extreme barrier approach, in order to provide a general and flexible framework for nonsmooth constrained problems. In [2] the use of a deterministic scheme for the generation of an asymptotically dense set of search directions is proposed, thus defining the ORTHOMADS method. A different way to handle the constraints within MADS-type algorithms is proposed in [6] where the authors combine a filter-based strategy [16] with a progressive barrier approach. In [45], the most general result for direct search methods of this type is given (for functions directionally Lipschitz) and, moreover, it is shown that integer lattices can be replaced by sufficient decrease when using polling directions asymptotically dense in the unit sphere.
In [11] it is proved that the efficiency of direct search methods (like e.g. MADS), when applied to nonsmooth problems, can be improved by using simplex gradients to order poll directions.

In this work, we extend the linesearch approach with sufficient decrease for unconstrained minimization problems in [17, 34] to the case of nonsmooth bound-constrained and nonlinearly-constrained minimization problems. This approach gives us a twofold achievement. On the one hand, by means of the sufficient decrease we can avoid the use of integer lattices. On the other hand, the extrapolation phase allows us to better exploit a descent direction and hence to prove stronger convergence results, i.e. stationarity of all the limit points of the sequence of iterates.
In the first part of this paper, we describe a general framework for solving bound-constrained nonsmooth optimization problems. The approach, called DFN$_{simple}$ (Derivative-Free Nonsmooth simple), combines a projected linesearch with the use of search directions which are asymptotically dense in the unit sphere. These two features make the algorithm very flexible as for the way to generate the asymptotically dense set of search directions, and allow us to prove convergence to stationary points of the problem in the Clarke-Jahn sense [21]. Then, we propose an improved version of the algorithm, namely CS-DFN (Coordinate Search Derivative-Free Nonsmooth), which further performs linesearches along the coordinate directions.
In the second part, we focus on nonlinearly constrained problems. We assume that two different classes of constraints exist, namely, difficult general nonlinear constraints ($g(x) \leq 0$) and simple

bound constraints on the problem variables ($l \leq x \leq u$). The main idea consists of getting rid of the nonlinear constraints by means of an exact penalty approach. Therefore, we construct a merit function that penalizes the general nonlinear inequality constraints and we resort to the minimization of the penalty function subject to the simple bound constraints. We acknowledge that the idea of only penalizing the nonlinear constraints is not new in the context of derivative-free optimization. Indeed, for smooth problems, the same handling of easy constraints has been previously adopted in several papers like, for instance, [27, 32] (for a combination of inequality and bound constraints) and [24] (for a combination of general and linear constraints). For nonsmooth problems, in [31] finite minimax problems are considered with explicit handling of linear inequality constraints; in [19] the idea of projecting onto simple constraints is proposed for Lipschitz problems. Following this approach, we can resort to the framework developed for the bound-constrained case and define an algorithm (which is called DFN$_{con}$) to tackle nonlinearly constrained nonsmooth problems. We are able to prove that the new bound-constrained problem is to a large extent equivalent to the original problem and that the sequence generated by means of the described approach converges to stationary points of the original problem, in the sense that every accumulation point is stationary for the constrained problem.
In the last part of the paper, an extended numerical experience (on 142 bound-constrained and 296 nonlinearly constrained problems) is carried out. We first test two versions of then DFN$_{simple}$ algorithm, obtained by embedding into the scheme two different pseudo-random sequences to generate the asymptotically dense set of search directions. In particular, we compare the Halton [18] and Sobol sequences [42, 7] within our method. Then we analyze the performances of both our methods DFN$_{simple}$ and CS-DFN, and we compare CS-DFN with two state-of-the-art solvers on a large set of 142 bound-constrained nonsmooth problems. Finally, we focus on nonlinearly constrained problems. In the latter case, we compare our code DFN$_{con}$ with two well-known codes on a large test set of 296 nonsmooth constrained problems. The codes DFN$_{simple}$, CS-DFN and DFN$_{con}$ are freely available for download at the url: `http://www.dis.uniroma1.it/∼lucidi/DFL`.

The paper has the following structure: Section 2 contains some preliminaries and technical definitions used throughout the paper. In Section 3, we analyze the approach for the bound-constrained case. In Section 4, we extend the approach to nonlinearly constrained problems. The numerical results are reported in Section 5. We summarize our conclusions in Section 6, and an Appendix completes the paper, including auxiliary results.
As regards the notation used in this paper, given a vector $v \in \mathbb{R}^n$, a subscript will be used to denote either the ith of its entries v_i or the fact that it is an element of an infinite sequence of vectors $\{v_k\}$. In case of possible misunderstanding or ambiguities, the ith component of a vector will be denoted by $(v)_i$ or $\{v\}_i$. By $\|\cdot\|$ we indicate the Euclidean norm. We denote by v^j the generic jth element of a finite set of vectors, and in particular $e^1, \ldots, e^n$ represent the coordinate unit vectors. Given two vectors $a, b \in \mathbb{R}^n$, we indicate with $y = \max\{a, b\}$ the vector such that $y_i = \max\{a_i, b_i\}$, $i = 1, \ldots, n$. Furthermore, given a vector $v \in \mathbb{R}^n$ we denote by $v^+ = \max\{0, v\}$. By $S(0, 1)$ we indicate the unit sphere with center in the origin, i.e. $S(0, 1) = \{x \in \mathbb{R}^n \ : \ \|x\| = 1\}$. Following [40, Theorem 3.3], given an arbitrary collection of finitely many non-empty convex sets $\Gamma = \{A_i \subseteq \mathbb{R}^n : i = 1, \ldots, m\}$, $Co(\Gamma)$ indicates its convex hull, namely

$$Co(\Gamma) = \bigcup \left\{ \sum_{i=1}^{m} \lambda_i A_i \right\},$$

where the union is taken over all finite convex combinations of the coefficients λ_i, $i = 1, \ldots, m$.

Finally, $[x]_{[l,u]} = \max\{l, \min\{u, x\}\}$ denotes the orthogonal projection over the set $\{x \in \mathbb{R}^n : l \le x \le u\}$, and $\partial f(x) = \{s \in \mathbb{R}^n : f^{Cl}(x; d) \ge v^T s, \forall\, v \in \mathbb{R}^n\}$ is the generalized gradient of f at x, where

$$f^{Cl}(x; d) = \limsup_{y \to x, t \downarrow 0} \frac{f(y + td) - f(y)}{t}. \tag{1}$$

2. Preliminary results

In this section, we introduce some definitions and report some results that will be used throughout the paper.
Recalling the convergence analysis carried out in [5, 2] for derivative-free nonsmooth optimization, the directions used in the optimization algorithm have to satisfy a condition which is intimately connected with the nonsmoothness of the problem itself. Indeed, since the cone of feasible descent directions can be arbitrarily narrow, there is no theoretical guarantee that by using finite sets of search directions one of them yields a negative Clarke directional derivative of the objective function (see e.g. [9, 23]). Hence, in the nonsmooth case, it seems necessary to assume that the limit points of particular subsequences of search directions are dense in $S(0, 1)$, in order to prove convergence to critical points.
Similarly to the above requirement, in the sequel we require the following assumption.

Assumption 1. *Let $\mathcal{M} : \mathbb{N} \mapsto S(0, 1)$ be a map and $\{d_k\}$ be a sequence of search directions with $d_k = \mathcal{M}(k)$. Let $\{x_k\}$ be a sequence of points and K be any subset of indices such that*

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

Then, $\mathcal{M}$ is such that the set of all the accumulation points of the subsequence $\{d_k\}_K$ is dense in $S(0, 1)$.

Note that the above Assumption 1 is slightly stronger than the one used in the references [5, 2] but it allows us to guarantee stationarity of all accumulation points of the sequence of iterates generated by the algorithm.

Lemma 2.1. *Let $\{x_k\} \subset A$ be a sequence converging to $\bar{x} \in A$ and let Assumption 1 hold, with $\{d_k\}$ being the sequence of search directions generated by map $\mathcal{M}$. Given a direction $\bar{d} \in S(0, 1)$ there exist subsequences $\{x_k\}_K \subseteq \{x_k\}$ and $\{d_k\}_K \subseteq \{d_k\}$ such that*

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

Proof. By contradiction, let us assume that an index $\bar{k}$ exists such that

$$\|d_k - \bar{d}\| \ge \epsilon, \quad k \ge \bar{k}, \tag{2}$$

for some $\epsilon > 0$. By Assumption 1 a subset $\hat{K}$ exists such that, for any $\epsilon > 0$,

$$\lim_{k \to \infty, k \in \hat{K}} d_k = \hat{d} \tag{3}$$

with

$$\|\hat{d} - \bar{d}\| \leq \frac{\epsilon}{4}. \tag{4}$$

Further, by (3), an index $\hat{k}$ exists such that, for all $k \geq \hat{k}$, $k \in \hat{K}$,

$$\|d_k - \hat{d}\| \leq \frac{\epsilon}{4}. \tag{5}$$

From (4) and (5) it follows that

$$\|d_k - \bar{d}\| \leq \|d_k - \hat{d}\| + \|\hat{d} - \bar{d}\| \leq \frac{\epsilon}{2}$$

which contradicts (2) and concludes the proof. □

Now, to conclude this section, we present a condition on the mapping $\mathcal{M}$ that allows to guarantee Assumption 1. To this aim, we first introduce the following definition of asymptotically dense sequence of sets (of directions).

Definition 2.2. *The sequence of normalized directions $\{d_k\}$ is said to be asymptoticaly dense in the unit sphere $S(0,1)$, if for any $\bar{d} \in S(0,1)$ and for any $\epsilon > 0$ there exists an index k such that $\|d_k - \bar{d}\| \leq \epsilon$.*

Finally we report a sufficient condition for Assumption 1 to hold.

Lemma 2.3. *Let $\mathcal{M} : \mathbb{N} \mapsto S(0,1)$ be a map and $\{d_k\}$ be a sequence of search directions with $d_k = \mathcal{M}(k)$. For any infinite index set $K \subseteq \mathbb{N}$, if the subsequence of directions $\{d_k\}_K$ is asymptotically dense in $S(0,1)$, then Assumption 1 holds.*

Proof. We proceed by contradiction and assume that Assumption 1 does not hold. Hence, a direction $\bar{d} \in S(0,1)$ exists such that for any subset K with

$$\lim_{k\to\infty, k\in K} d_k = d^K,$$

it results

$$\|d^K - \bar{d}\| > \epsilon$$

with $\epsilon > 0$. Then, an index $\bar{k} \in K$ exists such that, for all $k \in K$, $k \geq \bar{k}$ we have

$$\|d_k - \bar{d}\| > \frac{\epsilon}{2}. \tag{6}$$

Now, let us define

$$\bar{K} = \{k \in K : k \geq \bar{k}\}.$$

Since by assumption $\{d_k\}_{\bar{K}}$ is asymptotically dense in $S(0,1)$, an index $k \in \bar{K}$ (hence $k \in K$ and $k \geq \bar{k}$) exists such that

$$\|d_k - \bar{d}\| \leq \frac{\epsilon}{2}. \tag{7}$$

The above relation (7) is a contradiction with (6) thus concluding the proof. □

3. The bound-constrained case

In this section we consider the bound-constrained problem

$$\begin{array}{l} \min\ f(x) \\ s.t.\ x \in X, \end{array} \tag{8}$$

where we indicate by X the set of bound constraints on the variables, i.e.

$$X = \{x \in \mathbb{R}^n : l \leq x \leq u\},$$

and f is Lipschitz continuous. We recall that, since l and u are both finite, set X is compact. For points in the feasible set X we address also the definition of cone of feasible directions, as follows.

Definition 3.1 (Cone of feasible directions) *Given problem* (8) *and any point* $x \in X$

$$D(x) = \{d \in \mathbb{R}^n : d_i \geq 0 \text{ if } x_i = l_i,\ d_i \leq 0 \text{ if } x_i = u_i,\ d_i \in \mathbb{R} \text{ if } l_i < x_i < u_i,\ i = 1, \dots, n\}$$

is the cone of feasible directions at x *with respect to* X.

We also report a technical proposition whose proof can be found in [28, Proposition 2.3].

Proposition 3.2. *Given problem* (8), *let* $\{x_k\} \subset X$ *for all* k, *and* $\{x_k\} \to \bar{x}$ *for* $k \to \infty$. *Then, for* k *sufficiently large,*

$$D(\bar{x}) \subseteq D(x_k).$$

The necessary optimality conditions for problem (8) can be characterized in terms of the Clarke-Jahn generalized directional derivative of the objective function, instead of using the definition (1). Given a point $x \in X$, the Clarke-Jahn generalized directional derivative of the function f along the direction $d \in D(x)$ is given by (see [21, Section 3.5]):

$$f^\circ(x; d) = \limsup_{\substack{y \to x,\, y \in X \\ t \downarrow 0,\, y + td \in X}} \frac{f(y + td) - f(y)}{t}. \tag{9}$$

From [21, Theorem 4.14] we recall that every local minimum of problem (8) satisfies the following definition.

Definition 3.3. *Given problem (8),* $x^\star$ *is a Clarke-Jahn-stationary point if*

$$f^\circ(x^\star; d) \geq 0, \qquad \forall\ d \in D(x^\star). \tag{10}$$

We propose in the next sections two algorithms, having different performances on the nonsmooth bound-constrained problem (8).

3.1. A simple derivative-free algorithm

As discussed in the Introduction, even in the simpler case of bound constraints, since the objective function f is possibly not continuously differentiable on X, a finite number of search directions is not sufficient to investigate the local behavior of $f(x)$ on X [23, Section 6.4]. Hence, recalling [5], we resort to the use of a set of search directions which is eventually dense in the unit sphere. We prove that the use of such a simple set of search directions is sufficient to enforce convergence to stationary points of problem (8).

On this purpose, here we propose a very simple Derivative-Free algorithm for solving the Nonsmooth problem (8), namely DFN$_{simple}$, where a map satisfying Assumption 1 is adopted, in order to generate a set of search directions dense in the unit sphere.
In this algorithm, apart from the initializations, at any iteration k we use the map $\mathcal{M}$ to generate the search direction d_k. Then, we investigate the behavior of the function $f(x)$ along the direction d_k, by means of the linesearch procedure *Projected Continuous Search*. Given the current iterate x_k at step k, the latter procedure first evaluates the function at $[x_k \pm \tilde{\alpha}_k d_k]_{[l,u]}$. In case a sufficient reduction of the function value is obtained, then an extrapolation along the search direction is performed, so that a suitable step-length α_k is computed, and is used as a tentative steplength for the next iteration, i.e. $\tilde{\alpha}_{k+1} = \alpha_k$. On the other hand, if at $[x_k \pm \tilde{\alpha}_k d_k]_{[l,u]}$ we do not obtain a sufficient reduction of the function value, then the tentative steplength at the next iteration is suitably reduced by a scale factor, i.e. $\tilde{\alpha}_{k+1} = \theta\tilde{\alpha}_k$, $\theta \in (0, 1)$. More formally the resulting algorithm and the corresponding linesearch procedure adopted are summarized in the next schemes.

Algorithm DFN$_{simple}$

Data. $\theta \in (0,1)$, $x_0 \in X$, $\tilde{\alpha}_0 > 0$, the map $\mathcal{M} : \mathbb{N} \mapsto \mathbb{R}^n$ such that for $k \geq 0$, $d_k = \mathcal{M}(k)$ and

$$\|d_k\| = 1.$$

For $k = 0, 1, \ldots$

Set $d_k = \mathcal{M}(k)$.

Compute α_k and $\tilde{d}_k$ by the *Projected Continuous Search*$(\tilde{\alpha}_k, x_k, d_k; \alpha_k, \tilde{d}_k)$.

If $(\alpha_k = 0)$ **then** $\tilde{\alpha}_{k+1} = \theta\tilde{\alpha}_k$ and $\tilde{x}_k = x_k$

else $\tilde{\alpha}_{k+1} = \alpha_k$ and $\tilde{x}_k = [x_k + \alpha_k \tilde{d}_k]_{[l,u]}$

Find $x_{k+1} \in X$ such that $f(x_{k+1}) \leq f(\tilde{x}_k)$.

End For

Projected Continuous Search $(\tilde{\alpha}, y, p; \alpha, p^+)$

Data. $\gamma > 0$, $\delta \in (0,1)$.

Step 0. Set $\alpha = \tilde{\alpha}$.

Step 1. **If** $f([y + \alpha p]_{[l,u]}) \le f(y) - \gamma\alpha^2$ **then** set $p^+ = \ \ p$ and go to Step 4.

Step 2. **If** $f([y - \alpha p]_{[l,u]}) \le f(y) - \gamma\alpha^2$ **then** set $p^+ = -p$ and go to Step 4.

Step 3. Set $\alpha = 0$ and **return** α, p^+.

Step 4. Let $\beta = \alpha/\delta$.

Step 5. **If** $f([y + \beta p^+]_{[l,u]}) > f(y) - \gamma\beta^2$ **return** α, p^+.

Step 6. Set $\alpha = \beta$ and go to Step 4.

For the sake of clarity, we note that the Projected Continuous Search procedure takes in input $\tilde{\alpha}, y$, and p (that is the arguments before the semicolon) and gives in output α, and p^+ (that is the arguments after the semicolon).

It is worth noting that in Algorithm DFN$_{simple}$ the next iterate x_{k+1} is required to satisfy $f(x_{k+1}) \le f(\tilde{x}_k)$. This allows in principle to compute x_{k+1} by minimizing suitable approximating models of the objective function, thus possibly improving the efficiency of the overall scheme.

Furthermore, since we are interested in studying the asymptotic convergence properties of DFN$_{simple}$, its formal definition does not include a stopping condition. We note that this is in accordance with most of the papers concerning convergence of derivative free methods, see, e.g., [43, 15, 13, 6, 10, 12] among others. We refer the reader to Section 5 for a practical stopping condition.

In the following results we analyze the global convergence properties of Algorithm DFN$_{simple}$. In particular, in the next proposition we prove that the procedure described in Algorithm DFN$_{simple}$ cannot cycle.

Proposition 3.4. *The* Projected Continuous Search *cannot infinitely cycle between Step 4 and Step 6.*

Proof. Let us consider the *Projected Continuous Search*, we proceed by contradiction assuming that an infinite monotonically increasing sequence of positive numbers $\{\beta_j\}$ exists such that

$$f([y + \beta_j p^+]_{[l,u]}) \le f(y) - \gamma\beta_j^2.$$

The above relation contradicts the fact that X is compact, by definition, and that function f is continuous, thus concluding the proof. □

Now, in the following proposition we prove that the stepsizes computed by the procedure *Projected Continuous Search* eventually go to zero.

Proposition 3.5. *Let* $\{\alpha_k\}$, $\{\tilde{\alpha}_k\}$ *be the sequences generated by Algorithm DFN*$_{simple}$*, then*

$$\lim_{k\to\infty} \max\{\alpha_k, \tilde{\alpha}_k\} = 0. \tag{11}$$

Proof. We split the iteration sequence $\{k\}$ into two sets K_1, K_2, with $K_1 \cup K_2 = \{k\}$ and $K_1 \cap K_2 = \emptyset$. We denote by

- K_1 the set of iterations when $\tilde{\alpha}_{k+1} = \alpha_k$;
- K_2 the set of iterations when $\tilde{\alpha}_{k+1} = \theta\tilde{\alpha}_k$ and $\alpha_k = 0$.

Note that K_1 and K_2 cannot be both finite. Let us first suppose that K_1 is infinite, then the instructions of the algorithm imply, for $k \in K_1$,

$$f(x_{k+1}) \le f([x_k + \alpha_k \tilde{d}_k]_{[l,u]}) \le f(x_k) - \gamma\alpha_k^2. \tag{12}$$

Taking into account the compactness of X and the continuity of f, we get from the above relation that $\{f(x_k)\}$ tends to a limit $\bar{f}$. Then, by (12), it follows

$$\lim_{k\to\infty, k\in K_1} \alpha_k = 0, \tag{13}$$

which also implies

$$\lim_{k\to\infty, k\in K_1} \tilde{\alpha}_k = 0. \tag{14}$$

Now, let us suppose that K_2 is infinite. Since, by definition, $\alpha_k = 0$, $k \in K_2$, we have

$$\lim_{k\to\infty, k\in K_2} \alpha_k = 0. \tag{15}$$

Then, let $m_k < k$ be the largest integer such that $m_k \in K_1$. By the instructions of the algorithm, we can write

$$\tilde{\alpha}_{k+1} = \theta^{k-m_k}\tilde{\alpha}_{m_k}. \tag{16}$$

Note that, in case the index m_k does not exist (when K_1 is empty), we set $m_k = 0$. When $k \to \infty$ and $k \in K_2$, we have only the following two cases: either $m_k \to \infty$ (i.e. K_1 is an infinite subset) or $(k - m_k) \to \infty$ (i.e. K_1 is finite). Therefore, (14) and (16) along with $\theta \in (0, 1)$ give

$$\lim_{k\to\infty, k\in K_2} \tilde{\alpha}_k = 0. \tag{17}$$

Relations (13), (14), (15) and (17) yield (11), thus concluding the proof. □

Using the latter result, along with map $\mathcal{M}$ satisfying Assumption 1, we can provide the next technical lemma, which will be necessary to prove the main global convergence result for algorithm DFN$_{simple}$.

Lemma 3.6. *Let $\{x_k\}$, $\{d_k\}$ be the sequences generated by Algorithm DFN$_{simple}$, let $\{\eta_k\}$ be a sequence such that $\eta_k > 0$, for all k, and $\lim_{k\to\infty} \eta_k = 0$. Further, let Assumption 1 hold. Then, for any accumulation point $\bar{x}$ of $\{x_k\}$ and for any direction $\bar{d} \in D(\bar{x})$, $\bar{d} \neq 0$, there exists a subsequence of indices K such that*

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

$$\lim_{k\to\infty, k\in K} d_k = \bar{d}, \tag{19}$$

$$\lim_{k\to\infty, k\in K} \eta_k = 0. \tag{20}$$

Moreover,

(i) for all $k \in K$ sufficiently large,

$$[x_k + \eta_k d_k]_{[l,u]} \neq x_k,$$

(ii) the following limit holds

$$\lim_{k\to\infty, k\in K} v_k = \bar{d},$$

where

$$v_k = \frac{[x_k + \eta_k d_k]_{[l,u]} - x_k}{\eta_k}. \tag{21}$$

Proof. Relations (18) and (19) follow from Lemma 2.1; (20) follows by assumption. Now, in order to prove items (i) and (ii), let us recall that

$$[x_k + \eta_k d_k]_{[l,u]} = \max\{l, \min\{u, (x_k + \eta_k d_k)\}\}.$$

Now we show that, for $k \in K$ sufficiently large

$$[x_k + \eta_k d_k]_{[l,u]} \neq x_k. \tag{22}$$

By contradiction, let us assume that, for $k \in K$ sufficiently large, we have

$$[x_k + \eta_k d_k]_{[l,u]} = x_k. \tag{23}$$

Since, by assumption, $\bar{d} \neq 0$, an index i with $\bar{d}_i \neq 0$ exists and one of the following three cases holds.

1) $\bar{x}_i = l_i$ (which implies $\bar{d}_i > 0$): we can write

$$\{[x_k + \eta_k d_k]_{[l,u]}\}_i = \max\{l_i, (x_k + \eta_k d_k)_i\};$$

since x_k is feasible and (19) holds, for k sufficiently large we have

$$\max\{l_i, (x_k + \eta_k d_k)_i\} > \max\left\{l_i, \left(x_k + \frac{\eta_k}{2}\bar{d}\right)_i\right\},$$

and by (20) we get

$$\max\left\{l_i, \left(x_k + \frac{\eta_k}{2}\bar{d}\right)_i\right\} = \left(x_k + \frac{\eta_k}{2}\bar{d}\right)_i \neq (x_k)_i. \tag{24}$$

2) $\bar{x}_i = u_i$ (which implies $\bar{d}_i < 0$): we can write

$$\{[x_k + \eta_k d_k]_{[l,u]}\}_i = \min\{u_i, (x_k + \eta_k d_k)_i\};$$

since x_k is feasible and (19) holds, for k sufficiently large we have

$$\min\{u_i, (x_k + \eta_k d_k)_i\} < \min\left\{u_i, \left(x_k + \frac{\eta_k}{2}\bar{d}\right)_i\right\},$$

and by (20) we get

$$\min\left\{u_i, \left(x_k + \frac{\eta_k}{2}\bar{d}\right)_i\right\} = \left(x_k + \frac{\eta_k}{2}\bar{d}\right)_i \neq (x_k)_i. \tag{25}$$

3) $l_i < \bar{x}_i < u_i$ (which implies $\bar{d}_i \neq 0$): we can write

$$\{[x_k + \eta_k d_k]_{[l,u]}\}_i = (x_k + \eta_k d_k)_i;$$

since x_k is feasible and (19) holds, for k sufficiently large we have

$$(x_k + \eta_k d_k)_i \neq (x_k)_i. \tag{26}$$

Then, by (24), (25) and (26) we have a contradiction with (23), which proves *(i)*.
Now, we recall definition (21) and note that, by (22), the vector v_k is eventually nonzero. By the definition of the vector v_k, we have for its $i-$th entry

$$\begin{aligned}(v_k)_i &= \frac{\max\{l_i, \min\{u_i, (x_k + \eta_k d_k)_i\}\} - (x_k)_i}{\eta_k} &(27)\\ &= \frac{\min\{u_i, \max\{l_i, (x_k + \eta_k d_k)_i\}\} - (x_k)_i}{\eta_k}. &(28)\end{aligned}$$

Now, let us distinguish among the following three cases, for k sufficiently large and $k \in K$:

1) $\bar{x}_i = l_i$: then by (27) we have

$$(v_k)_i = \frac{\max\{l_i, (x_k + \eta_k d_k)_i\} - (x_k)_i}{\eta_k}$$

and recalling that whenever $\bar{x}_i = l_i$ it must be $\bar{d}_i \geq 0$, we distinguish two subcases:

a) when $\bar{d}_i > 0$, then $(v_k)_i = \max\left\{\frac{l_i-(x_k)_i}{\eta_k}, (d_k)_i\right\} = (d_k)_i$;

b) when $\bar{d}_i = 0$, then

$$\lim_{k\to\infty, k\in K} (v_k)_i = \lim_{k\to\infty, k\in K} \max\left\{\frac{l_i - (x_k)_i}{\eta_k}, (d_k)_i\right\} = 0 = (\bar{d})_i.$$

2) $\bar{x}_i = u_i$: then by (28) we have

$$(v_k)_i = \frac{\min\{u_i, (x_k + \eta_k d_k)_i\} - (x_k)_i}{\eta_k}$$

and recalling that whenever $\bar{x}_i = u_i$ it must be $\bar{d}_i \leq 0$, we distinguish two subcases:

a) when $\bar{d}_i < 0$, then $(v_k)_i = \min\left\{\frac{u_i-(x_k)_i}{\eta_k}, (d_k)_i\right\} = (d_k)_i$;

b) when $\bar{d}_i = 0$, then

$$\lim_{k\to\infty, k\in K} (v_k)_i = \lim_{k\to\infty, k\in K} \min\left\{\frac{u_i - (x_k)_i}{\eta_k}, (d_k)_i\right\} = 0 = (\bar{d})_i;$$

3) $l_i < \bar{x}_i < u_i$: then by (27) or (28) we have $(v_k)_i = (x_k + \eta_k d_k - x_k)_i/\eta_k = (d_k)_i$;

which imply that $\lim_{k\to\infty, k\in K} v_k = \bar{d}$, so that *(ii)* is proved. □

Finally, we are now ready to prove the main convergence result for Algorithm DFN$_{simple}$. We highlight that according to the following proposition, every limit point of the sequence of iterates $\{x_k\}$, generated by Algorithm DFN$_{simple}$, is a stationary point for problem (8).

Proposition 3.7. *Let Assumption 1 hold and let $\{x_k\}$, $\{d_k\}$, $\{\alpha_k\}$, and $\{\tilde{\alpha}_k\}$ be the sequences generated by Algorithm DFN$_{simple}$. Then, every limit point of $\{x_k\}$ is stationary for problem (8).*

Proof. We recall that by Definition 3.3 we consider the stationarity condition at $\bar{x}$:

$$f^\circ(\bar{x};\bar{d}) = \limsup_{\substack{y \to \bar{x}, y \in X \\ t \downarrow 0, y + t\bar{d} \in X}} \frac{f(y + t\bar{d}) - f(y)}{t} \geq 0, \qquad \forall \bar{d} \in D(\bar{x}). \tag{29}$$

Let $\bar{x}$ be a limit point of $\{x_k\}$ and $\tilde{K} \subseteq \{1, 2, \dots\}$ be a subset of indices such that

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

We proceed by contradiction and assume that a direction $\bar{d} \in D(\bar{x}) \cap S(0, 1)$ exists such that

$$f^\circ(\bar{x};\bar{d}) = \limsup_{\substack{x_k \to \bar{x}, x_k \in X, \\ t \downarrow 0, x_k + t\bar{d} \in X}} \frac{f(x_k + t\bar{d}) - f(x_k)}{t} < 0. \tag{30}$$

By recalling the instructions of the Projected Continuous Search, if the condition at Step 1 is satisfied, we have $\alpha_k > 0$ and

$$f([x_k + (\alpha_k/\delta)d_k]_{[l,u]}) > f(x_k) - \gamma(\alpha_k/\delta)^2, \tag{31}$$

otherwise, we have

$$f([x_k + \tilde{\alpha}_k d_k]_{[l,u]}) > f(x_k) - \gamma\tilde{\alpha}_k^2, \tag{32}$$

where $d_k = \mathcal{M}(k)$. Now, for every index k, let us set

$$\eta_k = \begin{cases} \alpha_k/\delta & \text{if (31) holds} \\ \tilde{\alpha}_k & \text{if (32) holds.} \end{cases}$$

and let v_k be defined as in relation (21) of Lemma 3.6, that is

$$v_k = \frac{[x_k + \eta_k d_k]_{[l,u]} - x_k}{\eta_k}.$$

The instructions of Algorithm DFN$_{simple}$ and definition of η_k guarantee that $\eta_k > 0$, for all k. Moreover, by Proposition 3.5,

$$\lim_{k\to\infty} \eta_k = 0.$$

Hence, the assumptions of Lemma 2.1 are satisfied, so that we can consider the subset $K \subseteq \tilde{K}$, introduced in Lemma 2.1, such that

$$\lim_{k\to\infty, k\in K} x_k = \bar{x}, \tag{33}$$

$$\lim_{k\to\infty, k\in K} d_k = \bar{d}, \tag{34}$$

$$\lim_{k\to\infty, k\in K} \eta_k = 0. \tag{35}$$

Now, from point (i) of Lemma 3.6, we have $v_k \neq 0$, for $k \in K$ and sufficiently large, so that relations (31) and (32) can be eqivalently expressed as

$$f(x_k + \eta_k v_k) > f(x_k) - \gamma \eta_k^2,$$

that is, recalling that $\eta_k > 0$,

$$\frac{f(x_k + \eta_k v_k) - f(x_k)}{\eta_k} > -\gamma \eta_k, \tag{36}$$

for $k \in K$ and sufficiently large.
Then we can write

$$\begin{aligned}
&\limsup_{\substack{x_k \to \bar{x}, x_k \in X \\ t \downarrow 0, x_k + t\bar{d} \in X}} \frac{f(x_k + t\bar{d}) - f(x_k)}{t} \geq \limsup_{k \to \infty, k \in K} \frac{f(x_k + \eta_k \bar{d}) - f(x_k)}{\eta_k} = \\
&\limsup_{k \to \infty, k \in K} \frac{f(x_k + \eta_k \bar{d}) + f(x_k + \eta_k v_k) - f(x_k + \eta_k v_k) - f(x_k)}{\eta_k} \geq \\
&\limsup_{k \to \infty, k \in K} \frac{f(x_k + \eta_k v_k) - f(x_k)}{\eta_k} - L\|\bar{d} - v_k\|,
\end{aligned}$$

where L is the Lipschitz constant of f. By (36)–(35) and *(ii)* of Lemma 3.6 we get, from the latter relation,

$$\limsup_{\substack{x_k \to \bar{x}, x_k \in X \\ t \downarrow 0, x_k + t\bar{d} \in X}} \frac{f(x_k + t\bar{d}) - f(x_k)}{t} \geq 0$$

which contradicts (30) and concludes the proof. □

3.2. Combining DFN$_{simple}$ with coordinate searches

A possible way to improve the efficiency of Algorithm DFN$_{simple}$ can be to take advantage of the experience in the smooth case. For example, we can draw inspiration from the paper [33] where the objective function is repeatedly investigated along the directions $\pm e^1, \dots, \pm e^n$ in order to capture the local behavior of the objective function. In fact, the use of a set of search directions, which is constant with iterations, allows to store the actual and tentative steplengths, i.e. α^i and $\tilde{\alpha}^i$, respectively, that roughly summarize the sensitivity of the function along those directions. Thus, when the function is further investigated along such search directions, we can exploit information gathered in the previous searches along them.

In the following, we propose a new algorithm, where the search along coordinate directions is performed until the steplengths α^i and $\tilde{\alpha}^i$ are greater than a given threshold $\eta > 0$. In particular, the sampling along the coordinate directions is performed by means of a Continuous Search procedure [33, 30].

Algorithm CS-DFN

Data. $\theta \in (0,1)$, $\eta > 0$, $x_0 \in X$, $\tilde{\alpha}_0 > 0$, $\tilde{\alpha}_0^i > 0$, $d_0^i = e^i$, for $i = 1,\dots,n$, the map $\mathcal{M} : \mathbb{N} \mapsto \mathbb{R}^n$ such that, for $k \geq 0$, $d_k = \mathcal{M}(k)$ and $\|d_k\| = 1$.

For $k = 0, 1, \dots$

Set $y_k^1 = x_k$

For $i = 1, \dots, n$

Compute α and d_{k+1}^i by the *Continuous Search*($\tilde{\alpha}_k^i, y_k^i, d_k^i; \alpha, d_{k+1}^i$).

If $(\alpha = 0)$ **then** set $\alpha_k^i = 0$ and $\tilde{\alpha}_{k+1}^i = \theta\tilde{\alpha}_k^i$

else set $\alpha_k^i = \alpha$ and $\tilde{\alpha}_{k+1}^i = \alpha$

Set $y_k^{i+1} = y_k^i + \alpha_k^i d_{k+1}^i$.

End For

If $\left(\max_{i=1,\dots,n}\{\alpha_k^i, \tilde{\alpha}_k^i\} \leq \eta\right)$ **then**

Set $d_k = \mathcal{M}(k)$.

Compute α_k and $\tilde{d}_k$ by the *Projected Continuous Search*($\tilde{\alpha}_k, y_k^{n+1}, d_k; \alpha_k, \tilde{d}_k$).

If $(\alpha_k = 0)$ **then** $\tilde{\alpha}_{k+1} = \theta\tilde{\alpha}_k$ and $y_k^{n+2} = y_k^{n+1}$

else $\tilde{\alpha}_{k+1} = \alpha_k$ and $y_k^{n+2} = [y_k^{n+1} + \alpha_k\tilde{d}_k]_{[l,u]}$

else Set $\tilde{\alpha}_{k+1} = \tilde{\alpha}_k$ and $y_k^{n+2} = y_k^{n+1}$

Find $x_{k+1} \in X$ such that $f(x_{k+1}) \leq f(y_k^{n+2})$.

End For

Continuous Search $(\tilde{\alpha}, y, p; \alpha, p^+)$

Data. $\gamma > 0$, $\delta \in (0,1)$.

Step 1. Compute the largest $\bar{\alpha}$ such that $y + \bar{\alpha}p \in X$. Set $\alpha = \min\{\bar{\alpha}, \tilde{\alpha}\}$.

Step 2. **If** $\alpha > 0$ and $f(y + \alpha p) \leq f(y) - \gamma\alpha^2$ **then** set $p^+ = p$ and go to Step 6.

Step 3. Compute the largest $\bar{\alpha}$ such that $y - \bar{\alpha}p \in X$. Set $\alpha = \min\{\bar{\alpha}, \tilde{\alpha}\}$.

Step 4. **If** $\alpha > 0$ and $f(y - \alpha p) \leq f(y) - \gamma\alpha^2$ **then** set $p^+ = -p$ and go to Step 6.

Step 5. Set $\alpha = 0$ and **return** α, p^+.

Step 6. Let $\beta = \min\{\bar{\alpha}, (\alpha/\delta)\}$.

Step 7. **If** $\alpha = \bar{\alpha}$ or $f(y + \beta p^+) > f(y) - \gamma\beta^2$ **return** α, p^+.

Step 8. Set $\alpha = \beta$ and go to Step 6.

Concerning the above definition of Algorithm CS-DFN, we again remark that the lack of a stopping condition allows us to study the asymptotic convergence properties of CS-DFN.

The following three propositions concern the convergence analysis of Algorithm CS-DFN. The third proof is omitted since it is very similar to the corresponding one for Algorithm DFN_{simple}.

Proposition 3.8. *The* Continuous Search *cannot infinitely cycle between Step 6 and Step 8.*

Proof. We proceed by contradiction and assume that an infinite monotonically increasing sequence of positive numbers $\{\beta_j\}$ exists such that

$$\beta_j < \bar{\alpha} \quad \text{and} \quad f(y + \beta_j p^+) \le f(y) - \gamma \beta_j^2.$$

The above relation contradicts the fact that X is compact, by definition, and that function f in problem (8) is continuous. □

The proposition that follows concerns convergence to zero of the steplengths in Algorithm CS-DFN. In particular, since α_k^i and $\tilde{\alpha}_k^i$ tend to zero, it results that the search along the dense direction d_k is performed eventually infinitely many times.

Proposition 3.9. *Let $\{\alpha_k^i\}$, $\{\tilde{\alpha}_k^i\}$, $\{\alpha_k\}$ and $\{\tilde{\alpha}_k\}$ be the sequences generated by Algorithm CS-DFN, then*

$$\lim_{k\to\infty} \max\{\alpha_k^1, \tilde{\alpha}_k^1, \dots, \alpha_k^n, \tilde{\alpha}_k^n\} = 0, \tag{37}$$

$$\lim_{k\to\infty} \max\{\alpha_k, \tilde{\alpha}_k\} = 0. \tag{38}$$

Proof. Reasoning as in the proof of Proposition 1 in [33], we can prove (37).

Now we have to show (38). By virtue of (37), we know that an index $\bar{k}$ exists such that the dense direction d_k is investigated for all $k \ge \bar{k}$.

Then, without loss of generality, we split the iteration sequence $\{k\}$ into two sets K_1 and K_2, with $K_1 \cup K_2 = \{k\}$ and $K_1 \cap K_2 = \emptyset$. We denote by

- K_1 the set of iterations when $\tilde{\alpha}_{k+1} = \alpha_k$;
- K_2 the set of iterations when $\tilde{\alpha}_{k+1} = \theta \tilde{\alpha}_k$.

Hence, the proof follows by reasoning as in the proof of Proposition 3.5. □

Proposition 3.10. *Let Assumption 1 hold and let $\{x_k\}$ and $\{d_k\}$ be the sequences generated by Algorithm CS-DFN. Then, every limit point of $\{x_k\}$ is stationary for problem (8).*

Proof. The proof trivially follows from Proposition 3.7. □

4. The nonsmooth nonlinearly constrained case

In this section, we consider Lipschitz-countinuous nonlinearly constrained problems of the following form:

$$\begin{aligned} \min \ & f(x) \\ s.t. \ & g(x) \le 0, \\ & l \le x \le u, \end{aligned} \tag{39}$$

where $f : \mathbb{R}^n \to \mathbb{R}$, $g : \mathbb{R}^n \to \mathbb{R}^m$ and $l, u \in \mathbb{R}^n$. The vectors l and u correspond respectively to lower and upper bounds on the variables $x \in \mathbb{R}^n$, and satisfy the additional condition $l < u$. We also assume throughout the paper that $f(x)$ and $g(x)$ are Lipschitz continuous functions, though they may be possibly *nondifferentiable*. Furthermore, $\mathcal{F}$ indicates the feasible set of problem (39), i.e.

$$\mathcal{F} = \{x \in X : g(x) \le 0\}.$$

We highlight that, by definition, $X = \{x \in \mathbb{R}^n \ : \ l \le x \le u\}$ is a compact subset of $\mathbb{R}^n$.

4.1. Assumption and preliminary material

We further introduce some definitions and assumptions related to problem (39). First, in order to carry out the theoretical analysis, we use a version of the Mangasarian-Fromowitz Constraint Qualification (MFCQ) condition for nonsmooth problems.

Assumption 2 (MFCQ) *Given problem* (39)*, for any* $x \in X \setminus \mathcal{F}$ *a direction* $d \in D(x)$ *exists such that*

$$(\xi^{g_i})^\top d < 0, \tag{40}$$

for all $\xi^{g_i} \in \partial g_i(x)$*,* $i \in \{1, \dots, m : \ g_i(x) \geq 0\}$*.*

The nonlinearly constrained problem (39) can be handled partitioning the constraints in two different sets, the first one defined by general inequality constraints, and the second one consisting of simple bound constraints. Then, for this kind of problem, we can state necessary optimality conditions that explicitly take into account the presence of these two different sets of constraints. The following propositions extend the results in [20, Theorem 6] to the case where inequality constraints and an additional convex set of constraints are present.

Proposition 4.1 (Fritz John Optimality Conditions) *Let* $x^\star \in \mathcal{F}$ *be a local minimum of the problem* (39)*. Then, multipliers* $\lambda_0^\star, \lambda_1^\star, \dots, \lambda_m^\star \in \mathbb{R}$ *not all zero exist such that*

$$\begin{aligned} \max_{\xi} \quad & \xi^\top d \ \geq 0, \qquad \forall\ d \in D(x^\star) \\ & \xi \in \lambda_0^\star \partial f(x^\star) + \sum_{i=1}^{m} \lambda_i^\star \partial g_i(x^\star), \end{aligned} \tag{41}$$

$$\lambda_i^\star \geq 0 \quad \textit{and} \quad \lambda_i^\star g_i(x^\star) = 0 \qquad \forall\ i = 1, \dots, m. \tag{42}$$

Proof. The proof can be found in Appendix A. □

Corollary 4.2 (KKT Necessary Optimality Conditions) *Let* $x^\star \in \mathcal{F}$ *be a local minimum of problem* (39) *and assume that a direction* $d \in D(x^\star)$ *exists such that, for all* $i \in \{1, \dots, m : g_i(x^\star) = 0\}$*,*

$$(\xi^{g_i})^\top d < 0, \quad \forall\ \xi^{g_i} \in \partial g_i(x^\star). \tag{43}$$

Then, multipliers $\lambda_1^\star, \dots, \lambda_m^\star \in \mathbb{R}$ *exist such that*

$$\begin{aligned} \max_{\xi} \quad & \xi^\top d \ \geq 0, \qquad \forall\ d \in D(x^\star) \\ & \xi \in \partial f(x^\star) + \sum_{i=1}^{m} \lambda_i^\star \partial g_i(x^\star), \end{aligned} \tag{44}$$

$$\lambda_i^\star \geq 0 \quad \text{and} \quad \lambda_i^\star g_i(x^\star) = 0 \qquad \forall\ i = 1, \dots, m.$$

Proof. The proof can be found in Appendix A. □

As regards the stationarity conditions for problem (39), taking into account the above propositions, we can now give the following definition.

Definition 4.3 (Stationary point) *Given the problem (39), $\bar{x}$ is a stationary point of (39) if multipliers $\bar{\lambda}_1, \dots, \bar{\lambda}_m$ exist such that the following conditions hold:*

$$\begin{aligned} \max_{\xi} \quad & \xi^\top d \ \geq 0, \qquad \forall\, d \in D(\bar{x}) \\ & \xi \in \partial f(\bar{x}) + \sum_{i=1}^{m} \bar{\lambda}_i \partial g_i(\bar{x}), \end{aligned} \tag{45}$$

$$\bar{\lambda}_i \geq 0 \quad \text{and} \quad \bar{\lambda}_i g_i(\bar{x}) = 0 \qquad \forall\, i = 1, \dots, m. \tag{46}$$

4.2. The penalty approach

Given problem (39), we introduce the following penalty function

$$Z_\varepsilon(x) = f(x) + \frac{1}{\varepsilon} \sum_{i=1}^{m} \max\{0, g_i(x)\}$$

and define the penalized problem

$$\begin{array}{ll} \min & Z_\varepsilon(x) \\ s.t. & x \in X. \end{array} \tag{47}$$

Remark 4.4. *Observe that, since f and g_i, $i = 1, \dots, m$, are Lipschitz continuous, with Lipschitz constants L_f and L_{g_i}, $i = 1, \dots, m$, the penalty function Z_ε is Lipschitz continuous too, with Lipschitz constant*

$$L = L_f + \frac{1}{\varepsilon} \sum_{i=1}^{m} L_{g_i}.$$

Remark 4.5. *Note that problem (47), for any $\varepsilon > 0$, has the same structure and properties of problem (8).*

We further note that our penalty approach differs from the ones previously proposed in the literature (see e.g. [14] and references therein), since only the general nonlinear constraints are penalized. The minimization of the penalty function is then carried out on the set defined by the bound constraints. We report in the following proposition the equivalence between the problem (47) and the nonlinearly constrained problem (39).

Proposition 4.6. *Let Assumption 2 hold. Given problem* (39) *and considering problem* (47)*, a threshold value $\varepsilon^\star > 0$ exists such that, for every $\varepsilon \in (0, \varepsilon^\star]$, every Clark-Jahn-stationary point of Problem* (47) *is stationary for Problem (39).*

Proof. The proof is reported in Appendix B. □

4.3. A derivative-free algorithm

Now we report the algorithm adopted for solving problem (47), which is obtained from Algorithm CS-DFN by replacing f with Z_ε, for given $\varepsilon > 0$. For the sake of simplicity, we omit to report also the extension of Algorithm DFN$_{simple}$ to the general inequality constrained case, which requires trivial modifications.

Algorithm DFN$_{con}$

Data. $\theta \in (0,1)$, $x_0 \in X$, $\varepsilon > 0$, $\tilde\alpha_0 > 0$, $\tilde\alpha_0^i > 0$, $d_0^i = e^i$, for $i = 1,\dots,n$, the map $\mathcal{M} : \mathbb{N} \mapsto \mathbb{R}^n$ such that $d_k = \mathcal{M}(k)$ and $\|d_k\| = 1$.

For $k = 0, 1, \dots$
  Set $y_k^1 = x_k$
  For $i = 1,\dots,n$
    Compute α and d_{k+1}^i by the *Continuous Search*$(\tilde\alpha_k^i, y_k^i, d_k^i; \alpha, d_{k+1}^i)$.
    If $(\alpha = 0)$ **then** set $\alpha_k^i = 0$ and $\tilde\alpha_{k+1}^i = \theta\tilde\alpha_k^i$
    else set $\alpha_k^i = \alpha$ and $\tilde\alpha_{k+1}^i = \alpha$
    Set $y_k^{i+1} = y_k^i + \alpha_k^i d_{k+1}^i$.
  End For
  If $\left(\max_{i=1,\dots,n}\{\alpha_k^i, \tilde\alpha_k^i\} \le \eta\right)$ **then**
      Set $d_k = \mathcal{M}(k)$.
      Compute α_k and $\tilde d_k$ by the *Projected Continuous Search*$(\tilde\alpha_k, y_k^{n+1}, d_k; \alpha_k, \tilde d_k)$.
      If $(\alpha_k = 0)$ **then** $\tilde\alpha_{k+1} = \theta\tilde\alpha_k$ and $y_k^{n+2} = y_k^{n+1}$
      else $\tilde\alpha_{k+1} = \alpha_k$ and $y_k^{n+2} = [y_k^{n+1} + \alpha_k \tilde d_k]_{[l,u]}$
  else set $\tilde\alpha_{k+1} = \tilde\alpha_k$, $y_k^{n+2} = y_k^{n+1}$
  Find $x_{k+1} \in X$ such that $Z_\varepsilon(x_{k+1}) \le Z_\varepsilon(y_k^{n+2})$.
End For

We remark that in Algorithm DFN$_{con}$ the *Continuous Search* procedure is performed replacing f with Z_ε. Further, observe that Algorithm DFN$_{con}$ can be used to solve the constrained problem (39) provided that the penalty parameter ε is sufficiently small, as the following proposition states.

Proposition 4.7. *Let Assumptions 1 and 2 hold, and let $\{x_k\}$ and $\{d_k\}$ be the sequences generated by Algorithm DFN$_{con}$. Then, a threshold value ε^* exists such that for all $\varepsilon \in (0, \varepsilon^*]$ every limit point of $\{x_k\}$ is stationary for problem (39).*

Proof. The proof follows from Proposition 3.10 and Proposition 4.6. □

5. Implementation details and numerical results

This section is devoted to investigate the numerical issues related to the implementation of the proposed algorithms. We first report the numerical experience related to bound-constrained problems, then we analyze the computational results related to the nonlinearly constrained case. All the experiments, have been conducted allowing for a maximum number of 20000 function evaluations (which is quite reasonable considered the dimensions of our test problems).

For the parameters included in the proposed algorithms (DFN$_{simple}$, CS-DFN, DFN$_{con}$) we considered the following setting: $\theta = 0.5$, $\gamma = 10^{-6}$, $\delta = 0.5$, $\eta = 10^{-3}$,

$$\begin{aligned}\tilde{\alpha}_0^i &= \max\left\{10^{-3}, \min\{1, |(x_0)_i|\}\right\}, \qquad i = 1,\dots,n,\\ \tilde{\alpha}_0 &= \frac{1}{n}\sum_{i=1}^{n}\tilde{\alpha}_0^i.\end{aligned}$$

Regarding the choice of the new iterate x_{k+1}, we remark that:

- in Algorithm DFN$_{simple}$, x_{k+1} is computed starting from $\tilde{x}_k$ and performing Projected continuous searches along a set of $n-1$ directions which define an orthonormal basis in $\mathbb{R}^n$ along with d_k;
- in Algorithms CS-DFN and DFN$_{con}$, if $(\max_{i=1,\dots,n}\{\alpha_k^i, \tilde{\alpha}_k^i\} \le \eta)$ then x_{k+1} is computed as above but starting from y_k^{n+2}. Otherwise, we set $x_{k+1} = y_k^{n+2}$.

As a final note, by drawing inspiration from [25, Theorem 6.4] and from the proof of Proposition 3.7, and by recalling that by Proposition 3.5 $\lim_{k\to\infty}\max\{\alpha_k, \tilde{\alpha}_k\} = 0$, in the implementation of our algorithms, we used as termination condition the following heuristic rule

$$\max\{\alpha_k, \tilde{\alpha}_k\} \le 10^{-13}. \tag{48}$$

However, we highlight that the algorithms are compared by means of performance and data profiles [36], that is by using a normalized convergence test on the function values. Thus, we adopted the tight convergence test (48) in order to provide enough information on the progress of all the codes compared.

The codes DFN$_{simple}$, CS-DFN and DFN$_{con}$ are freely available for download at the url:

`http://www.dis.uniroma1.it/~lucidi/DFL`

5.1. Bound constrained problems

The first part of the numerical experience has been carried out on a set of 142 bound-constrained problems from [44], [31] and [36], with a number of variables in the range $[1, 200]$ (see Table 1).

n	1	2	3	4	5	6	7	8	9	10	11	12	15	20	50	100	200
# of problems	4	34	12	17	10	8	6	6	6	10	6	6	1	8	4	2	2

Table 1: Distribution of problem dimensions for the bound-constrained case

As showed in the theoretical analysis of the different algorithms, our linesearch-based approach is able to guarantee convergence towards stationary points of the nonsmooth problem, by using search directions generated by a mapping $\mathcal{M}$ satisfying Assumption 1. In particular, we can adopt the mapping based on the Halton sequence [18], which is the one implemented in the NOMAD package [1, 26, 2]. But, unlike NOMAD, further mappings can be easily embedded into our algorithms, since we are not committed to use a modified Halton sequence in order to generate points on a mesh (see e.g. [18]). For instance, we implemented a mapping based on the Sobol sequence [42, 7], which is a pseudo-random sequence widely used in practice.

In order to show the behavior of the above pseudo-random sequences, we preliminarily compared two versions of the Algorithm DFL$_{simple}$, which respectively use the Halton and the Sobol sequence, on the test set of bound-constrained problems described above. The resulting experience

is reported in Figure 1 using data and performance profiles [36].
As we can see, the Sobol pseudo-random sequence outperforms the Halton one for all precision levels, both in terms of efficiency and robustness. Then, we compared both our Algorithms DFN_{simple} and CS-DFN and reported the results in Figure 2 in terms of performance and data profiles.
As we can see, the combination of coordinate and dense directions can improve the performance of the algorithm.

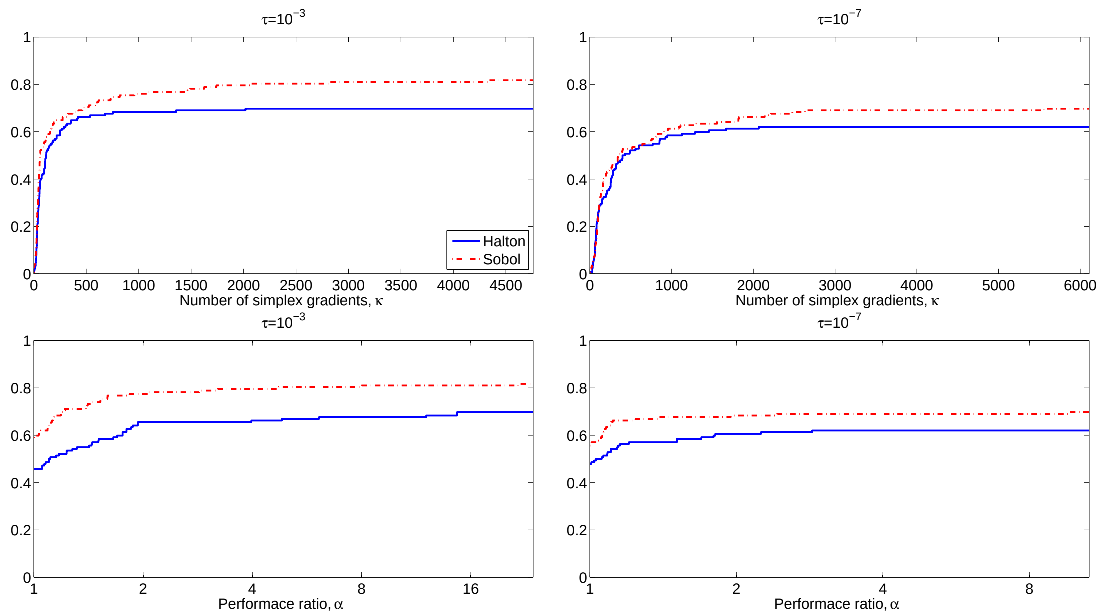

Figure 1: Data (top) and performance (bottom) profiles for the 142 bound-constrained problems. Comparison between Sobol and Halton pseudo-random sequences within DFN_{simple}

Finally, we compare CS-DFN with two state-of-the-art derivative free optimization codes, namely NOMAD [1, 26, 2] and BOBYQA [39]. We run NOMAD by using its default settings and BOBYQA by specifying `RHOBEG` $= 1$ and `RHOEND` $= 10^{-13}$. The results of this comparison are summarized in Figure 3. By looking at how rapidly data and performance profiles raise for small values of κ and α respectively, we can say that: (i) BOBYQA is very efficient for small precision; (ii) when a high precision, i.e. $\tau = 10^7$, is required, CS-DFN is the most efficient in terms of number of simplex gradients evaluations whereas NOMAD is the best one in terms of performance ratio. As concerns the robustness of the compared solvers, which is represented by the asympthotic behavior of the reported data and performance profiles, the robustness of CS-DFN is between NOMAD, which is the most robust one, and BOBYQA. The above comments about efficiency and robustness of the methods were definitely expected and can be explained by the following considerations.

(i) BOBYQA is quite efficient because it is a model based method whose performances are strongly related to the smoothness of the objective function;

(ii) both CS-DFN and NOMAD are more robust than BOBYQA since they use globally convergent strategies which do not assume any continuous differentiability. For this reason,

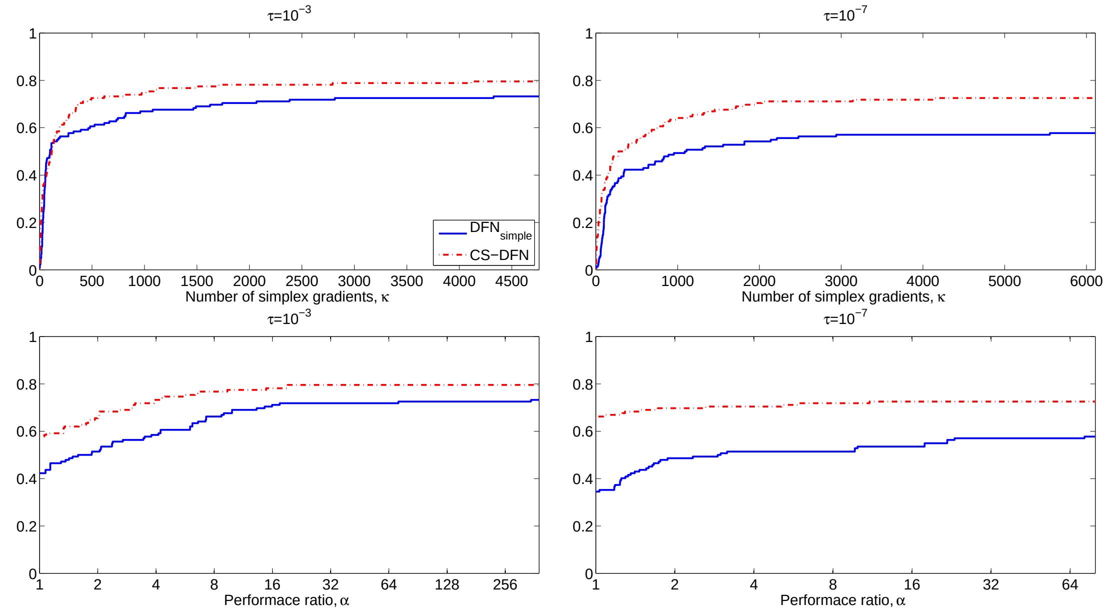

Figure 2: Data (top) and performance (bottom) profiles for the 142 bound-constrained problems. Comparison between DFN_{simple} and CS-DFN

they result less efficient than BOBYQA and (potentially) more expensive from a computational point of view;

(iii) NOMAD is the most robust code because it incorporates an heuristic search phase (as opposed to the poll phase) in which quadratic models are used to try and improve the current iterate. This phase can surely help improving the quality of the solution, especially for non-convex problems.

To better understand the behaviors of CS-DFN and NOMAD, we now limit the comparison to those problems where both methods find the same solution. More precisely, given a problem and the solution points $x^{*,1}$ and $x^{*,2}$ returned by the two solvers, the solutions are considered the same if

$$\frac{|f(x^{*,1}) - f(x^{*,2})|}{f(x_0) - \min\{f(x^{*,1}), f(x^{*,2})\}} \leq \tau,$$

for a given precision $\tau > 0$. These results are reported in Figure 4 where it can be noted that the gap between CS-DFN and NOMAD in terms of robustness is considerably reduced. Figure 4 also shows the good behavior of the linesearch strategy of CS-DFN in terms of efficiency. Since the results in Figure 4 report convergence to the same stationary point, we claim that, for these problems, the search phase of NOMAD possibly does not help to improve efficiency.
These latter results motivate us to better investigate the behavior of the codes. We report the results of a further comparison between CS-DFN and NOMAD, where we run NOMAD disabling the search phase with quadratic models by setting `MODEL_SEARCH` to `NO`, again on the whole set of 142 bound constrained problems. The results reported in Figure 5 suggest that, when NOMAD does not exploit the model search phase, to a large extent CS-DFN and NOMAD show similar performance in terms of robustness (with a slight preference for CS-DFN). On the other hand, it can be noted that the globalization strategy of CS-DFN, based on the use of linesearches,

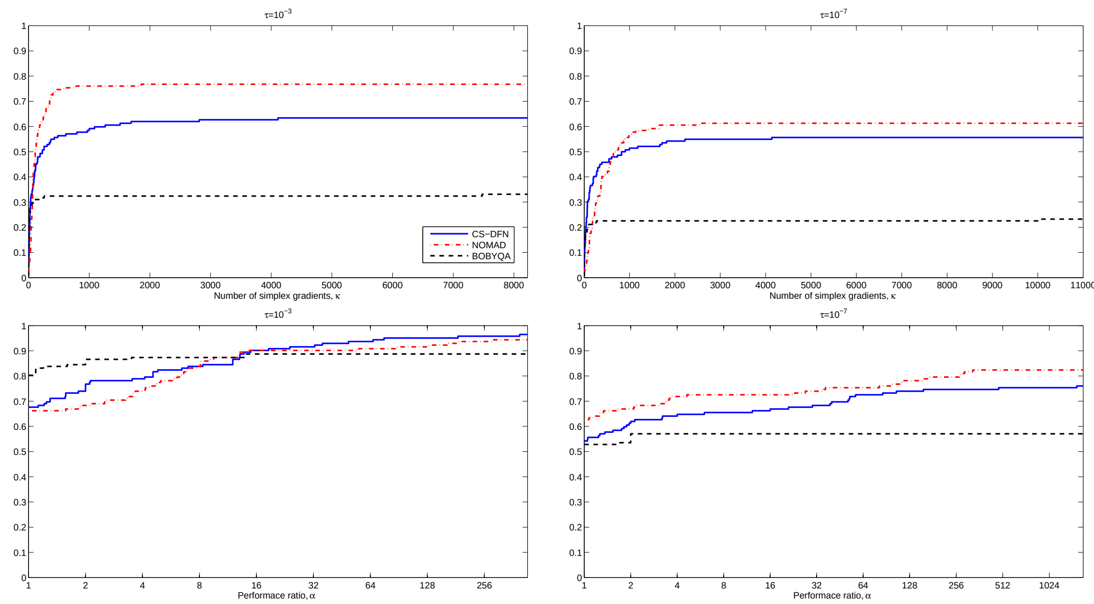

Figure 3: Data (top) and performance (bottom) profiles for the 142 bound-constrained problems. Comparison among CS-DFN, NOMAD and BOBYQA

outperforms the strategy of NOMAD, based on the use of a mesh adaptive direct search, in terms of efficiency.

5.2. Nonlinearly constrained problems

In the second part of our numerical experience, we defined a set of hard nonsmooth nonlinearly constrained test problems, by pairing the objective functions of the collection [31] with the constraint families proposed in [22], thus obtaining 296 problems. The problems in this collection have a number of constraints m in the range $[1, 199]$ and a number of variables n in the range $[1, 200]$ (see Table 2). We note that 205 out of 296 problems have a starting point x_0 which is

n	2	3	4	5	6	10	20	50	100	200
# of problems	96	30	40	10	10	20	40	10	20	20

m	1	2	3	4	5	6	8	9	11	12
# of problems	151	39	17	24	9	1	3	2	3	2
m	18	19	22	23	48	49	98	99	198	199
# of problems	9	6	3	2	3	2	6	4	6	4

Table 2: Distribution of problem dimensions (n number of variables, m number of constraints) for the nonlinearly constrained test set

not feasible, that is

$$h(x_0) > 10^{-6}, \qquad \text{with } h(x) = \max\left\{0, \max_{i=1,\dots,m}\{g_i(x)\}\right\}. \tag{49}$$

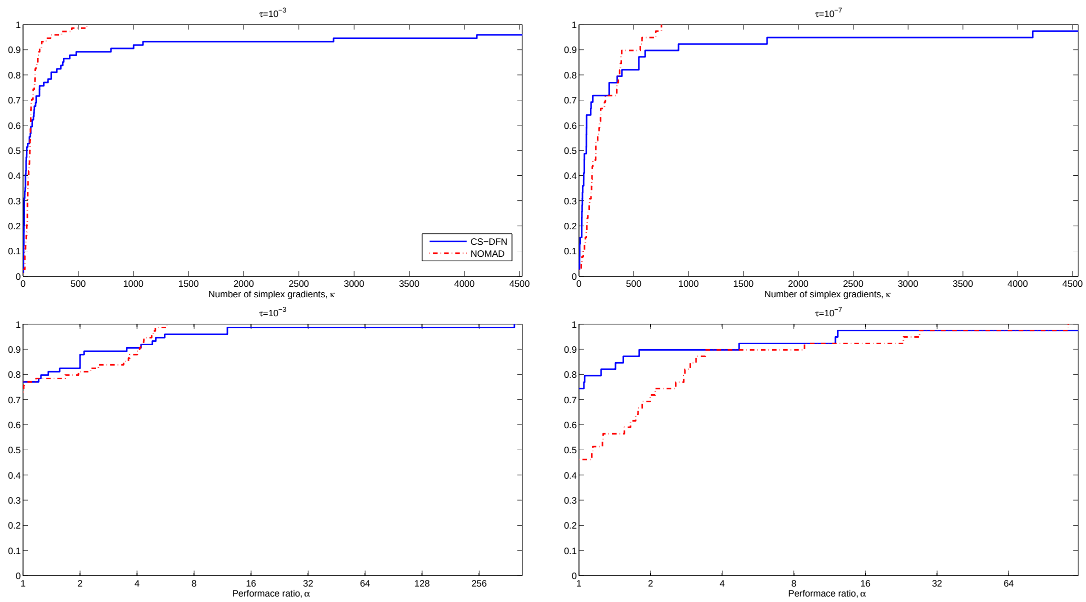

Figure 4: Data (top) and performance (bottom) profiles for CS-DFN and NOMAD on the set of problems where they find the same solution

In order to adapt the procedure for constructing performance and data profiles, as proposed in [36], to the nonlinearly constrained case, we considered the convergence test

$$\tilde{f}_0 - f(x) \geq (1-\tau)(\tilde{f}_0 - f_L),$$

where $\tilde{f}_0$ is the objective function value of the *worst feasible point* determined by all the solvers (note that in the bound-constrained case, $\tilde{f}_0 = f(x_0)$), $\tau > 0$ is a tolerance, and f_L is computed for each problem as the smallest value of f (at a feasible point) obtained by any solver within 20000 function evaluations. We notice that when a point is not feasible (i.e. $h(x) > 10^{-6}$) we set $f(x) = +\infty$.

As concerns the penalty parameter ε that defines Algorithm DFN$_{con}$, we first tried different fixed values for this parameter, namely $10^{-1}, 10^{-3}, 10^{-5}$. Then, we tried a more sophisticated managing and updating strategy. In particular, we used a vector of penalty parameters $\varepsilon \in \mathbb{R}^m$ and considered the penalty function

$$Z_\varepsilon(x) = f(x) + \sum_{i=1}^{m} \frac{1}{\varepsilon_i} \max\{0, g_i(x)\},$$

which trivially preserves all the theoretical results proved in Section 4. The vector of penalty parameters is iteratively updated during progress of the algorithm and, in particular, we chose

$$(\varepsilon_0)_i = \begin{cases} 10^{-3} & \text{if } \max\{0, g_i(x_0)\} < 1 \\ 10^{-1} & \text{otherwise,} \end{cases} \qquad i = 1, \dots, m, \tag{50}$$

and adopted the updating rule

$$(\varepsilon_{k+1})_i = \begin{cases} 10^{-2}(\varepsilon_k)_i & \text{if } (\varepsilon_k)_i g_i(x_k) > \max\{\alpha_k, \tilde{\alpha}_k\} \\ (\varepsilon_k)_i & \text{otherwise,} \end{cases} \qquad i = 1, \dots, m. \tag{51}$$

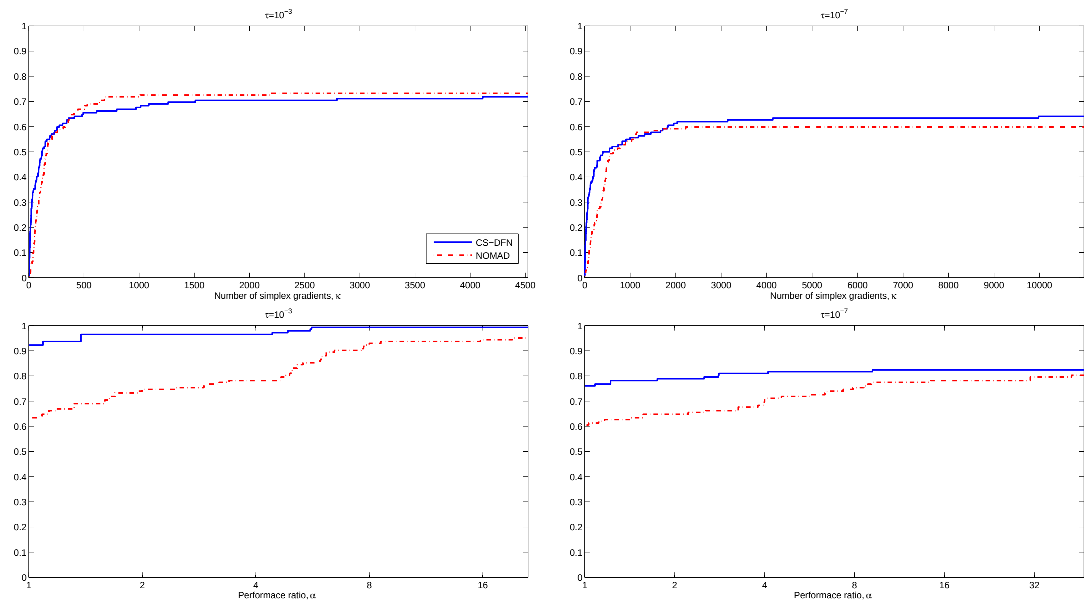

Figure 5: Data (top) and performance (bottom) profiles for the 142 bound-constrained problems. Comparison among CS-DFN and NOMAD (without quadratic models)

The above updating rule is applied right before computation of the new iterate x_{k+1}. We notice that the rule described above takes inspiration from derivative-based exact penalty approaches (see e.g. [29], [37]) where the updating rule for the penalty parameter is based on the (scaled) comparison between the stationarity measure of the point and the constraint violation. In a derivative-free context, the stationarity measure can be approximated by means of the steplengths selected along the search directions, as showed in [25].

First of all we compare the different versions of DFN$_{con}$ with the above described different strategies for the parameter ε. The results of this comparison are reported in Figure 6 from which it emerges that, though the performances of the algorithms are quite similar to each other, the scheme where parameter ε is adaptively updated looks preferable.
Then, in Figure 7, we report the comparison among DFN$_{con}$, NOMAD and COBYLA [38]. NOMAD was run by setting the constraints type to `PEB`, so that constraints are treated first with the progressive barrier, and once satisfied, with the extreme barrier approach. COBYLA was run by setting `RHOBEG` $= 1$ and `RHOEND` $= 10^{-13}$. As already said, a maximum number of 20000 function evaluations was specified for all of the solvers. As it can be seen, when relatively small precision is required, COBYLA has an initial fast progress, but is not as robust as the other two codes. When high precision is required, NOMAD is the more robust solver and DFN$_{con}$ is slightly more efficient with respect to the data profiles.
Again, as already done for the bound constrained case, to better understand the behavior of NOMAD and DFN$_{con}$, we now limit the comparison of the two codes to those problems where both the solvers find the same feasible solution. These results are reported in Figure 8. It emerges that the gap between NOMAD and DFN$_{con}$ is considerably reduced. This seems to confirm that NOMAD has a greater ability to find better solutions with respect to those found by DFN$_{con}$.

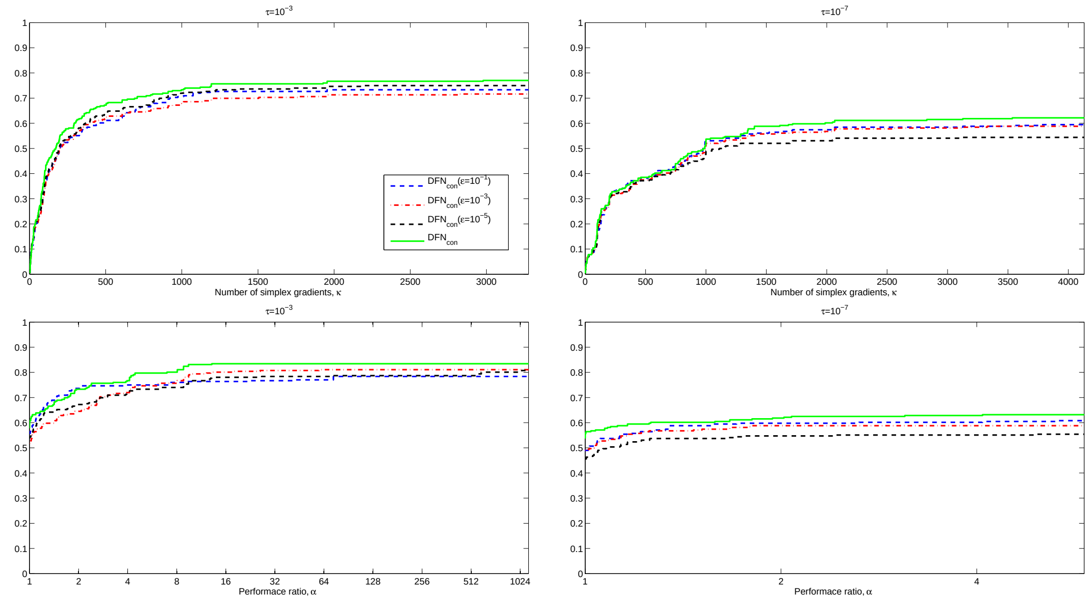

Figure 6: Data (top) and performance (bottom) profiles for the 296 constrained problems. Comparison among different versions of DFN$_{con}$ with different strategies for updating the ϵ parameter

Finally, we again compare DFN$_{con}$ and NOMAD on the whole testset but by setting the parameter `MODEL_SEARCH` to `NO` in NOMAD. The results are reported in Figure 9 and confirm that the robusteness of NOMAD was largely due to the use of quadratic models to heuristically improve the current iterate.

6. Conclusions

In this paper, we described new methods for dealing with nonsmooth optimization problems when no first-order information is available. We adopted a projected linesearch approach and we combined it with asymptotically dense sequences of search directions. In particular, we extended the linesearch approach with sufficient decrease for smooth minimization problems. This approach gives a twofold achievement. On the one hand, by means of the sufficient decrease we can avoid the use of integer lattices. On the other hand, the extrapolation phase allows us to better exploit a descent direction and hence to prove stronger convergence results, i.e. stationarity of all the limit points of the sequence of iterates.

In the first part of the paper, we considered problems with only bound constraints on the variables and we proposed two different algorithms for their solution. We showed that every accumulation point of the sequence of iterates produced by both the algorithms is Clark stationary. As concerns nonlinear inequality constrained problems, we introduced the use of an exact penalty function to transform the given problem into a bound-constrained one, which is solved by adapting the method proposed for the bound-constrained case. Similarly to the bound constrained case, we were able to prove again that every accumulation point of the generated sequence of iterates is Clarke stationary for the original constrained problem.

The numerical results reported in the paper show that the use of linesearches gives a large

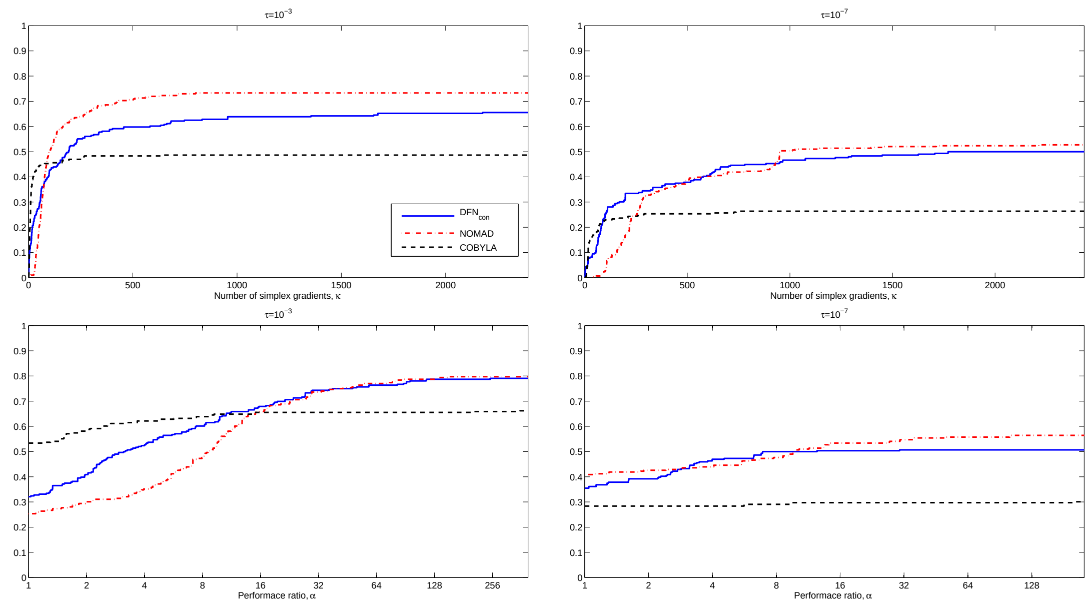

Figure 7: Data (top) and performance (bottom) profiles for the 296 constrained problems. Comparison among DFN$_{con}$, NOMAD and COBYLA

freedom in the choice of the set of used search directions. Furthermore, our analysis highlights the fact that coordinate searches can often improve the performance of the proposed algorithms. Finally, we compared the proposed methods with other state-of-the-art codes on two large test sets of bound-constrained and nonlinearly-constrained nonsmooth problems. The numerical experimentation carried out evidenced that our globalization strategy is promising, as compared to the mesh adaptive direct search strategy, and at the same time showed the importance of using approximating models to determine good solution points.

Acknowledgments

We are indebted with two anonymous Reviewers whose many suggestions and stimulating comments greatly helped us improving the quality of the paper.

A. Necessary optimality conditions

Proof of Proposition 4.1. By definition of local optimality of $x^\star$, we know that a constant $\rho > 0$ exists such that

$$f(x^\star) \le f(x), \qquad \text{for all } x \in \mathcal{F} \cap \mathcal{B}(x^\star, \rho), \tag{52}$$

where $\mathcal{B}(x^\star, \rho) = \{x \in \mathbb{R}^n : \|x - x^\star\| \le \rho\}$. Then, let us introduce the following functional

$$\Phi(x) = \max\Big\{f(x) - f(x^\star), g_1(x), \ldots, g_m(x), (l_1 - x_1), \ldots, (l_n - x_n), (x_1 - u_1), \ldots, (x_n - u_n)\Big\}$$

and show that, for all $x \in \mathcal{B}(x^\star, \rho)$, $\Phi(x) \ge 0$. Indeed, by contradiction, suppose that $\hat{x} \in \mathcal{B}(x^\star, \rho)$ exists such that $\Phi(\hat{x}) < 0$. This implies that $g(\hat{x}) < 0$ and $l < \hat{x} < u$ yielding $\hat{x} \in \mathcal{F}$. Further, $f(\hat{x}) < f(x^\star)$. This latter condition contradicts the optimality condition (52) of $x^\star$.

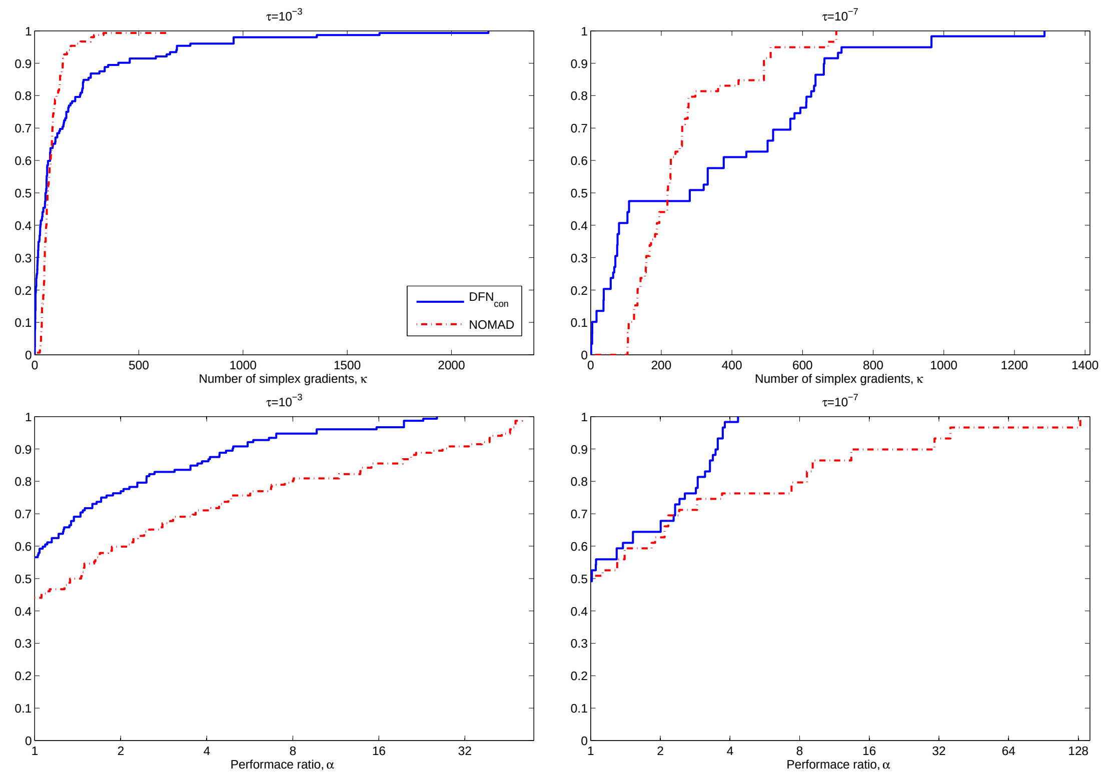

Figure 8: Data (top) and performance (bottom) profiles for DFN$_{con}$ and NOMAD on the set of problems where they find the same solution

Now, since $x^\star \in \mathcal{F}$ and $\Phi(x^\star) = 0$, we know that $x^\star$ is a local minimum of $\Phi(x)$ onto $\mathbb{R}^n$. Hence, by definition of Clarke stationarity

$$0 \in \partial\Phi(x^\star).$$

Then, considering [8, Proposition 2.3.12], we have that

$$0 \in \tilde\lambda_0 \partial f(x^\star) + \sum_{i\in I_0(x^\star)} \tilde\lambda_i \partial g_i(x^\star) - \sum_{j\in I_l(x^\star)} \tilde\mu_j e_j + \sum_{h\in I_u(x^\star)} \tilde\mu_h e_h, \tag{53}$$

with $I_0(x^\star) = \{i : g_i(x^\star) = 0\}$, $I_l(x^\star) = \{j : x_j^\star = l_j\}$, $I_u(x^\star) = \{h : x_h^\star = u_h\}$, $\tilde\lambda_0 \geq 0$, $\tilde\lambda_i \geq 0$, $i \in I_0(x^\star)$, $\tilde\mu_j \geq 0$, $j \in I_l(x^\star)$, $\tilde\mu_h \geq 0$, $h \in I_u(x^\star)$, and

$$\tilde\lambda_0 + \sum_{i\in I_0(x^\star)} \tilde\lambda_i + \sum_{j\in I_l(x^\star)} \tilde\mu_j + \sum_{h\in I_u(x^\star)} \tilde\mu_h = 1. \tag{54}$$

Now, from the linear independence of the set $\{e_j, e_h,\ j \in I_l(x^\star),\ h \in I_u(x^\star)\}$, it turns out that

$$\tilde\lambda_0 + \sum_{i\in I_0(x^\star)} \tilde\lambda_i \neq 0.$$

Indeed, if this was not the case, relation (53) would yield

$$0 = -\sum_{j\in I_l(x^\star)} \tilde\mu_j e_j + \sum_{h\in I_u(x^\star)} \tilde\mu_h e_h,$$

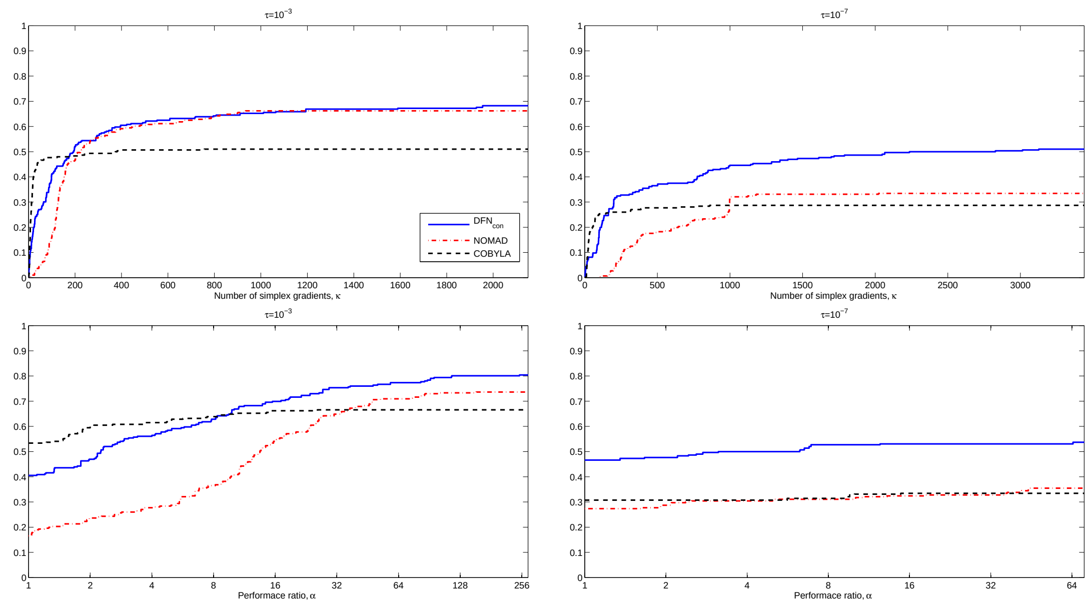

Figure 9: Data (top) and performance (bottom) profiles for the 296 constrained problems. Comparison among DFN$_{con}$, NOMAD (without quadratic models) and COBYLA

which would then impose $\tilde\mu_j$'s and $\tilde\mu_h$'s to be all zero thus contradicting (54).
Hence, by dividing (53) by $\tilde\lambda_0 + \sum_{i\in I_0(x^\star)} \tilde\lambda_i = \Lambda \neq 0$ and by posing

$$\begin{aligned} \lambda_0 &= \frac{\tilde\lambda_0}{\Lambda}, \\ \lambda_i &= \frac{\tilde\lambda_i}{\Lambda},\ \forall\ i \in I_0(x^\star), \\ \mu_j &= \frac{\tilde\mu_j}{\Lambda},\ \forall\ j \in I_l(x^\star), \\ \mu_h &= \frac{\tilde\mu_h}{\Lambda},\ \forall\ h \in I_u(x^\star), \end{aligned}$$

we get

$$0 \in \lambda_0 \partial f(x^\star) + \sum_{i\in I_0(x^\star)} \lambda_i \partial g_i(x^\star) - \sum_{j\in I_l(x^\star)} \mu_j e_j + \sum_{h\in I_u(x^\star)} \mu_h e_h,$$

with $\lambda_0 \geq 0$, $\lambda_i \geq 0$, $i \in I_0(x^\star)$, $\mu_j \geq 0$, $j \in I_l(x^\star)$, $\mu_h \geq 0$, $h \in I_u(x^\star)$, and

$$\lambda_0 + \sum_{i\in I_0(x^\star)} \lambda_i = 1. \tag{55}$$

Hence, it exists

$$\xi \in \lambda_0 \partial f(x^\star) + \sum_{i\in I_0(x^\star)} \lambda_i \partial g_i(x^\star),$$

such that

$$0 = \xi\ - \sum_{j\in I_l(x^\star)} \mu_j e_j + \sum_{h\in I_u(x^\star)} \mu_h e_h.$$

Then, recalling Definition 3.1 of $D(x)$, we have $\xi^\top d \geq 0$, for all $d \in D(x^\star)$ which shows

$$\begin{array}{l} \max_{\xi} \ \xi^\top d \geq 0 \\ \quad \xi \in \lambda_0 \partial f(x^\star) + \sum_{i \in I_0(x^\star)} \lambda_i \partial g_i(x^\star), \end{array}$$

with $\lambda_0 \geq 0$, $\lambda_i \geq 0$, $i \in I_0(x^\star)$, and, by (55), not all zero. This concludes the proof, chosing $\lambda_i = 0$, for all $i \notin I_0(x^\star)$. □

Proof of Corollary 4.2. By assumption, in particular by condition (43), we know that $\bar{d} \in \mathbb{R}^n$ exists such that

$$\begin{aligned} (\xi^{g_i})^\top \bar{d} < 0, &\qquad \forall\, \xi^{g_i} \in \partial g_i(x^\star), \quad \forall\, i \in I_0(x^\star), \\ -e_j^\top \bar{d} < 0, &\qquad \forall\, j \in I_l(x^\star), \\ e_h^\top \bar{d} < 0, &\qquad \forall\, h \in I_u(x^\star), \end{aligned}$$

where $I_0(x^\star) = \{i : g_i(x^\star) = 0\}$, $I_l(x^\star) = \{j : x_j^\star = l_j\}$, and $I_u(x^\star) = \{h : x_h^\star = u_h\}$.
Now, by the alternative theorem in [41, Theorem 2.3.4] and [46], there cannot exist multipliers $\tilde{\lambda}_i \geq 0$, $i \in I_0(x^\star)$, $\tilde{\mu}_j \geq 0$, $j \in I_l(x^\star)$, $\tilde{\mu}_h \geq 0$, $h \in I_u(x^\star)$, with

$$\sum_{i \in I_0(x^\star)} \tilde{\lambda}_i + \sum_{j \in I_l(x^\star)} \tilde{\mu}_j + \sum_{h \in I_u(x^\star)} \tilde{\mu}_h = 1,$$

such that

$$0 \in \sum_{i \in I_0(x^\star)} \tilde{\lambda}_i \partial g_i(x^\star) - \sum_{j \in I_l(x^\star)} \tilde{\mu}_j e_j + \sum_{h \in I_u(x^\star)} \tilde{\mu}_h e_h. \tag{56}$$

On the other hand, by Proposition 4.1, we know that multipliers $\lambda_0^\star \geq 0$, $\lambda_i^\star \geq 0$, with $\lambda_i^\star = 0$ when $g_i(x^\star) < 0$, exist such that (41) holds. Now, we proceed by contradiction and assume that $\lambda_0^\star = 0$. In this case, we would have

$$\begin{array}{l} \max_{\xi} \ \xi^\top d \geq 0 \qquad \forall\, d \in D(x^\star) \\ \quad \xi \in \sum_{i \in I(x^\star)} \lambda_i^\star \partial g_i(x^\star), \\ \quad \lambda_i^\star \geq 0, \quad \forall\, i \in I_0(x^\star). \end{array} \tag{57}$$

Note that the multipliers $\lambda_i^\star$, $i = 1, \ldots, m$, cannot be all zero, since in this case all the multipliers would be zero thus contradicting Proposition 4.1. Then, we can define new multipliers

$$\bar{\lambda}_i = \lambda_i^\star / \beta, \qquad i \in I_0(x^\star),$$

where $\beta = \sum_{i \in I_0(x^\star)} \lambda_i^\star > 0$. Hence, we have that $\bar{\lambda}_i \geq 0$ and $\sum_{i \in I_0(x^\star)} \bar{\lambda}_i = 1$. Now, (57) can be written as

$$\begin{array}{l} \max_{\xi} \ \tilde{\xi}^\top d \geq 0 \qquad \forall\, d \in D(x^\star) \\ \quad \tilde{\xi} \in \sum_{i \in I_0(x^\star)} \bar{\lambda}_i \partial g_i(x^\star), \\ \quad \sum_{i \in I_0(x^\star)} \bar{\lambda}_i = 1, \qquad \bar{\lambda}_i \geq 0, \quad \forall\, i \in I_0(x^\star). \end{array} \tag{58}$$

Then, there exists a vector $\bar{\xi}$ such that

$$\bar{\xi} \in \sum_{i \in I_0(x^\star)} \bar{\lambda}_i \partial g_i(x^\star), \tag{59}$$

and the following system

$$\begin{array}{rcll} -\bar{\xi}^\top d & < & 0, & \\ -e_j^\top d & \leq & 0, & \forall\, j \in I_l(x^\star), \\ e_h^\top d & \leq & 0, & \forall\, h \in I_u(x^\star), \end{array}$$

does not have a solution, where the latter two sets of constraints imply $d \in D(x^\star)$. As a consequence, by the Farkas' theorem (see e.g. [35, Chapter 2]), we have that scalars not all zero $\alpha_j \geq 0$, $j \in I_l(x^\star)$, and $\alpha_h \geq 0$, $h \in I_u(x^\star)$, exist such that

$$-\bar{\xi} = -\sum_{j \in I_l(x^\star)} \alpha_j e_j + \sum_{h \in I_u(x^\star)} \alpha_h e_h.$$

The above equation along with (59) is in contradiction with (56). □

B. Exactness properties of $Z_\varepsilon(x)$

In this section we first prove that any Clarke-stationary point of Problem (47) is stationary for Problem (39). We recall from [8] the definition of Clarke-stationary point, namely, a point $\bar{x} \in X$ such that

$$Z_\varepsilon^{Cl}(\bar{x}; d) \geq 0, \qquad \forall\, d \in D(\bar{x}).$$

Furthermore, we assume throughout this section that the Assumption 2 holds.

Proposition B.1. *Given problem* (39) *and considering problem* (47)*, a threshold value* $\varepsilon^\star > 0$ *exists such that, for every* $\varepsilon \in (0, \varepsilon^\star]$*, the function* $Z_\varepsilon(x)$ *has no Clarke-stationary points in* $X \setminus \mathcal{F}$*.*

Proof. We proceed by contradiction and assume that for any integer k an $\varepsilon_k \leq 1/k$ and a stationary point for Problem (47) $x_k \in X \setminus \mathcal{F}$ exists. Then, let us consider a limit point $\bar{x} \in X \setminus \mathcal{F}$ of the sequence $\{x_k\}$ and let us relabel the corresponding subsequence $\{x_k\}$ again. Since $\bar{x} \notin \mathcal{F}$, Assumption 2 guarantees that a direction $\bar{d} \in D(\bar{x})$ exists such that

$$\left(\xi^{g_i}\right)^\top \bar{d} < 0, \qquad \text{for all } \xi^{g_i} \in \partial g_i(\bar{x}),\ i \in \{1, \dots, m : g_i(\bar{x}) \geq 0\}.$$

In particular, it holds that

$$\left(\xi^{g_i}\right)^\top \bar{d} < 0, \qquad \text{for all } \xi^{g_i} \in \partial g_i(\bar{x}),\ i \in I(\bar{x}), \tag{60}$$

where $I(\bar{x}) = \{i \in \{1, \dots, m\} : g_i(\bar{x}) = \phi(\bar{x}) > 0\}$ and $\phi(x) = \max\{0, g_1(x), \dots, g_m(x)\}$. The above property can be equivalently expressed by saying that a positive scalar η exists, such that

$$\max_{\substack{\xi^{g_i} \in \partial g_i(\bar{x}) \\ i \in I(\bar{x})}} \left(\xi^{g_i}\right)^\top \bar{d} = -\eta < 0. \tag{61}$$

Recalling that, for k sufficiently large, $D(\bar{x}) \subseteq D(x_k)$ (see e.g. [28]), so that $\bar{d} \in D(x_k)$, we get, by considering that x_k is a Clarke-stationary point of Problem (47), that

$$Z_{\varepsilon}^{Cl}(x_k; \bar{d}) \geq 0. \tag{62}$$

Since, by [8, Proposition 2.1.2], $Z_{\varepsilon}^{Cl}(x; \bar{d}) = \max_{\xi \in \partial Z_{\varepsilon}(x)} \xi^{\top} \bar{d}$, and we know that

$$\partial Z_{\varepsilon}(x) \subseteq \partial f(x) + \frac{1}{\varepsilon} \partial(\max\{0, g_1(x), \ldots, g_m(x)\})$$

and (see [8, Proposition 2.3.12])

$$\partial(\max\{0, g_1(x), \ldots, g_m(x)\}) \subseteq Co\left(\{\partial g_i(x) : i \in I(x)\}\right) = \bigcup \left\{ \sum_{i \in I(x)} \beta^i \partial g_i(x) \right\}.$$

Hence, β_k^i, $i \in I(x_k)$, exist such that (62) can be written as

$$\left(\xi_k^f + \frac{1}{\varepsilon_k} \sum_{i \in I(x_k)} \beta_k^i \xi_k^{g_i} \right)^{\top} \bar{d} \geq 0. \tag{63}$$

$$\sum_{i \in I(x_k)} \beta_k^i = 1, \qquad \beta_k^i \geq 0,$$

for some $\xi_k^f \in \partial f(x_k)$, $\xi_k^{g_i} \in \partial g_i(x_k)$.
Now, recalling that m is a finite number, we can consider the subsequence of $\{x_k\}$ where $I(x_k) = \bar{I}$.
Then, since the generalized gradient of a locally Lipschitz continuous function is locally bounded, it results that all the considered sequences $\{\xi_k^f\}$, $\{\xi_k^{g_i}\}$, $i \in \bar{I}$, where $\xi_k^f \in \partial f(x_k)$, $\xi_k^{g_i} \in \partial g_i(x_k)$, $x_k \in X$, are bounded[1]. Hence, we get that

$$\xi_k^f \to \bar{\xi}^f, \tag{64a}$$
$$\xi_k^{g_i} \to \bar{\xi}^{g_i}, \qquad \text{for all } i \in \bar{I}, \tag{64b}$$
$$\beta_k^i \to \bar{\beta}^i, \qquad \text{for all } i \in \bar{I}. \tag{64c}$$

Further, since ∂f and ∂g_i, $i \in \bar{I}$ are upper semicontinuous at $\bar{x}$ (see Proposition 2.1.5 in [8]), it results $\bar{\xi}^f \in \partial f(\bar{x})$, $\bar{\xi}^{g_i} \in \partial g_i(\bar{x})$, $i \in \bar{I}$.
Now, since by continuity of the problem functions we have for k sufficiently large

$$\{i : g_i(\bar{x}) - \phi(\bar{x}) < 0\} \subseteq \{i : g_i(x_k) - \phi(x_k) < 0\},$$

it results, for k sufficiently large,

$$\{i : g_i(x_k) - \phi(x_k) = 0\} = I(x_k) \subseteq I(\bar{x}) = \{i : g_i(\bar{x}) - \phi(\bar{x}) = 0\},$$

so that

$$\bar{I} \subseteq I(\bar{x}). \tag{65}$$

[1]This result follows by considering that a finite covering of X by bounded sets exists and that any $\xi_k^f, \xi_k^{g_i}$, $i \in \bar{I}$, are bounded on the latter sets.

Then, by (61), (64), and (65), we get, for k sufficiently large,

$$\left(\xi_k^{g_i}\right)^\top \bar{d} \leq -\frac{\eta}{2}, \qquad \forall\, i \in \bar{I}. \tag{66}$$

Now, by multiplying (63) by ε_k we have

$$\left(\varepsilon_k \xi_k^f + \sum_{i\in\bar{I}} \beta_k^i \xi_k^{g_i}\right)^\top \bar{d} \geq 0,$$

which, by (66), yields

$$0 \leq \left(\varepsilon_k \xi_k^f + \sum_{i\in\bar{I}} \beta_k^i \xi_k^{g_i}\right)^\top \bar{d} \leq \left(\varepsilon_k \xi_k^f\right)^\top \bar{d} - \frac{\eta}{2}. \tag{67}$$

Finally, the above relation, considering (64a), gives raise to a contradiction when $\varepsilon_k \to 0$. □

Now we report three further results concerning the exactness of $Z_\varepsilon(x)$ from reference [14].

Proposition B.2. *A threshold value $\varepsilon^\star > 0$ exists such that, for any $\varepsilon \in (0, \varepsilon^\star]$, every local minimum point of problem (47) is also a local minimum point of Problem (39).*

Proposition B.3. *A threshold value $\varepsilon^\star > 0$ exists such that, for any $\varepsilon \in (0, \varepsilon^\star]$, every global minimum point of problem (47) is also a global minimum point of Problem (39), and conversely.*

In order to give stationarity results for problem (47), we have the following proposition.

Proposition B.4. *For any $\varepsilon > 0$, every Clarke-stationary point $\bar{x}$ of problem (47), such that $\bar{x} \in \mathcal{F}$, is also a stationary point of problem (39).*

Proof. Since $\bar{x}$ is, by assumption, a Clarke-stationary point of Problem (47), then, by definition of Clarke stationarity, we know that, for all $d \in D(\bar{x})$,

$$\begin{aligned} \max_{\xi} \;& \xi^\top d \geq 0 \\ & \xi \in \partial Z_\varepsilon(\bar{x}), \end{aligned}$$

that is $\xi^\star \in \partial Z_\varepsilon(\bar{x})$ exists such that $(\xi^\star)^\top d \geq 0$, for all $d \in D(\bar{x})$. Now, we recall that

$$\partial Z_\varepsilon(x) \subseteq \partial f(x) + \frac{1}{\varepsilon} \sum_{i\in I(x)} \beta_i \partial g_i(x),$$

for some β_i, $i \in I(x)$, such that $\sum_{i\in I(x)} \beta_i = 1$ and $\beta_i \geq 0$, for all $i \in I(x)$. Hence, we have that $\xi^* \in \partial f(\bar{x}) + \frac{1}{\varepsilon}\sum_{i\in I(\bar{x})} \beta_i \partial g_i(\bar{x})$. Then, denoting $\lambda_i = \beta_i/\varepsilon$, $i \in I(\bar{x})$, we can write

$$\begin{aligned} \max_{\xi} \;& \xi^\top d \geq 0 \\ & \xi \in \partial f(\bar{x}) + \sum_{i\in I(\bar{x})} \lambda_i \partial g_i(\bar{x}), \end{aligned}$$

with $\lambda_i \geq 0$. The above condition, along with $\bar{x} \in \mathcal{F}$, proves stationarity of $\bar{x}$ for Problem (39) and concludes the proof. □

Finally we can prove Proposition 4.6.

Proof of Proposition 4.6. Since $\bar{x}$ is Clarke-Jahn-stationary for Problem (47), we have, by Definition 3.3,

$$Z_\varepsilon^\circ(\bar{x}; d) \geq 0, \quad \forall\, d \in D(\bar{x}). \tag{68}$$

Then, by definitions (1) and (9), we have that

$$\limsup_{y \to \bar{x},\, t \downarrow 0} \frac{Z_\varepsilon(y + td) - Z_\varepsilon(y)}{t} = Z_\varepsilon^{Cl}(\bar{x}; d) \geq Z_\varepsilon^\circ(\bar{x}; d), \quad \forall\, d \in D(\bar{x}),$$

which, by (68), gives

$$Z_\varepsilon^{Cl}(\bar{x}; d) \geq 0, \quad \forall\, d \in D(\bar{x}),$$

Now, the proof follows by considering Propositions B.1 and B.4. □

References

[1] M.A. Abramson, C. Audet, G. Couture, J.E. Dennis Jr., S. Le Digabel, and C. Tribes. The NOMAD project. Software available at url `http://www.gerad.ca/nomad`.

[2] M.A. Abramson, C. Audet, J.E. Dennis Jr., and S. Le Digabel. Orthomads: A deterministic mads instance with orthogonal directions. *SIAM Journal on Optimization*, 20(2):948–966, 2009.

[3] N. Alexandrov and M.Y. Hussaini. *Multidisciplinary Design Optimization - state of the art*. SIAM Proceedings Series, SIAM, Philadelphia, 1997.

[4] C. Audet, A. Custódio, and J.E. Dennis Jr. Erratum: Mesh adaptive direct search algorithms for constrained optimization. *SIAM Journal on Optimization*, 18:1501–1503, 2008.

[5] C. Audet and J.E. Dennis Jr. Mesh adaptive direct search algorithms for constrained optimization. *SIAM Journal on Optimization*, 17(1):188–217, 2006.

[6] C. Audet and J.E. Dennis Jr. A progressive barrier for derivative-free nonlinear programming. *SIAM Journal on Optimization*, 20(4):445–472, 2009.

[7] P. Bratley and B. Fox. Algorithm 659: Implementing Sobol's quasirandom sequence generator. *ACM Transactions on Mathematical Software*, 14(1):88–100, 1988.

[8] F.H. Clarke. *Optimization and Nonsmooth Analysis*. John Wiley & Sons, New York, 1983.

[9] A. Conn, K. Scheinberg, and L.N. Vicente. *Introduction to Derivative-Free Optimization*. MPS/SIAM Ser. Optim., SIAM, Philadelphia, 2009.

[10] A.R. Conn, K. Scheinberg, and L.N. Vicente. Global convergence of general derivative-free trust-region algorithms to first and second order critical points. *SIAM Journal on Optimization*, 20:387–415, 2009.

[11] A. L. Custódio, J.E. Dennis Jr., and L.N. Vicente. Using simplex gradients of nonsmooth functions in direct search methods. *IMA Journal of Numerical Analysis*, 28:770–784, 2008.

[12] A.L. Custódio, J.F.A. Madeira, A.I.F. Vaz, and L.N. Vicente. Direct multisearch for multiobjective optimization. *SIAM Journal on Optimization*, 21:1109–1140, 2011.

[13] A.L. Custódio and L.N. Vicente. Using sampling and simplex derivatives in pattern search methods. *SIAM Journal on Optimization*, 18:537–555, 2007.

[14] G. Di Pillo and F. Facchinei. Exact barrier function methods for Lipschitz programs. *Applied Mathematics and Optimization*, 32:1–31, 1995.

[15] E.D. Dolan, R.M. Lewis, and V. Torczon. On the local convergence of pattern search. *SIAM Journal on Optimization*, 14(2):567–583, 2003.

[16] R. Fletcher and S. Leyffer. Nonlinear programming without a penalty function. *Mathematical Programming*, 91:239–269, 2002.

[17] L. Grippo, F. Lampariello, and S. Lucidi. Global convergence and stabilization of unconstrained minimization methods without derivatives. *Journal of Optimization Theory and Applications*, 56:385–406, 1988.

[18] J.H. Halton. On the efficiency of certain quasi-random sequences of points in evaluating multi-dimensional integrals. *Numerische Mathematik*, 2:84–90, 1960.

[19] W.L. Hare. Using derivative free optimization for constrained parameter selection in a home and community care forecasting model. In J. Blake and M. Carter, editors, *International Perspectives on Operations Research and Health Care*, pages 61–73, 2010. Proceedings of the 34th Meeting of the EURO Working Group on Operational Research Applied to Health Sciences.

[20] J.B. Hiriart-Urruty. On optimality conditions in nondifferentiable programming. *Mathematical Programming*, 14:73–86, 1978.

[21] J. Jahn. *Introduction to the theory of nonlinear optimization.* Springer-Verlag, Berlin, 1996.

[22] N. Karmitsa. Test problems for large-scale nonsmooth minimization. Technical Report No. B. 4/2007, Department of Mathematical Information Technology, University of Jyväskylä, 2007.

[23] T.G. Kolda, R.M. Lewis, and V. Torczon. Optimization by direct search: new perspectives on some classical and modern methods. *SIAM Review*, 45:385–482, 2003.

[24] T.G. Kolda, R.M. Lewis, and V. Torczon. A generating set direct search augmented lagrangian algorithm for optimization with a combination of general and linear constraints. Technical Report Tech. Report SAND2006-5315, Sandia National Laboratories, 2006.

[25] 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.

[26] S. Le Digabel. Algorithm 909: NOMAD: Nonlinear optimization with the MADS algorithm. *ACM Transactions on Mathematical Software*, 37(4):1–15, 2011.

[27] R.M. Lewis and V. Torczon. A globally convergent augmented lagrangian pattern search algorithm for optimization with general constraints and simple bounds. *SIAM Journal on Optimization*, 12:1075–1089, 2002.

[28] C.J. Lin, S. Lucidi, L. Palagi, A. Risi, and M. Sciandrone. Decomposition algorithm model for singly linearly-constrained problems subject to lower and upper bounds. *Journal of Optimization Theory and Applications*, 141:107–126, 2009.

[29] G. Liuzzi and S. Lucidi. New results on a continuously differentiable exact penalty function. *SIAM Journal on Optimization*, 2(4):558–574, 1992.

[30] G. Liuzzi, S. Lucidi, and F. Rinaldi. Derivative-free methods for bound constrained mixed-integer optimization. *Computational Optimization and Applications*, 53:505–526, 2012.

[31] G. Liuzzi, S. Lucidi, and M. Sciandrone. A derivative-free algorithm for linearly constrained finite minimax problems. *SIAM Journal on Optimization*, 16(4):1054–1075, 2006.

[32] G. Liuzzi, S. Lucidi, and M. Sciandrone. Sequential penalty derivative-free methods for nonlinear constrained optimization. *SIAM Journal on Optimization*, 20(5):2614–2635, 2010.

[33] S. Lucidi and M. Sciandrone. A derivative-free algorithm for bound constrained optimization. *Computational Optimization and Applications*, 21(2):119–142, 2002.

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

[35] O.L. Mangasarian. *Nonlinear Programming*. Classics in Applied Mathematics. SIAM, Philadelphia, USA, 1994.

[36] J.J. More' and S.M. Wild. Benchmarking derivative-free optimization algorithms. *SIAM Journal on Optimization*, 20(1):172–191, 2009.

[37] G. Di Pillo and S. Lucidi. An augmented lagrangian function with improved exactness properties. *SIAM Journal on Optimization*, 12(2):376–406, 2001.

[38] M.J.D. Powell. A direct search optimization method that models the objective and constraint functions by linear interpolation. In S. Gomez and J-P. Hennart, editors, *Advances in Optimization and Numerical Analysis*, pages 51–67. Kluwer Academic, Dordrecht, 1994.

[39] M.J.D. Powell. The bobyqa algorithm for bound constrained optimization without derivatives. Technical report, Department of Applied Mathematics and Theoretical Physics, University of Cambridge, 2009.

[40] R.T. Rockafellar. *Convex Analysis*. Princeton Mathematical Series. Princeton University Press, Princeton, New Jersey (USA), 1970.

[41] K. Shimizu, Y. Yshizuka, and J.F. Bard. *Nondifferentiable and Two-level Mathematical Programming*. Kluwer Academic Publishers, Norwell, Massachusetts, USA, 1997.

[42] I. Sobol. Uniformly distributed sequences with an additional uniform property. *USSR Computational Mathematics and Mathematical Physics*, 16:236–242, 1977.

[43] V. Torczon. On the convergence of pattern search algorithms. *SIAM Journal on Optimization*, 7(1):1–25, 1997.

[44] J. Vlček V. Lukšan. Test problems for nonsmooth unconstrained and linearly constrained optimization. *Technical report, Institute of Computer Science, Academy of Sciences of the Czech Republic*, 2000.

[45] L.N. Vicente and A. L. Custódio. Analysis of direct searches for discontinuous functions. *Mathematical Programming*, 133:299–325, 2012.

[46] Y. Yshizuka and K. Shimizu. Necessary and sufficient conditions for the efficient solutions of nondifferentiable multi-objective problems. *IEEE Transactions on Systems, Man and Cybernetics*, SMC-14(4):624–629, 1984.