Noname manuscript No.
(will be inserted by the editor)

An interior point method for nonlinear constrained derivative-free optimization

Andrea Brilli · Giampaolo Liuzzi · Stefano Lucidi

the date of receipt and acceptance should be inserted later

Abstract In this paper we consider constrained optimization problems where both the objective and constraint functions are of the black-box type. Furthermore, we assume that the nonlinear inequality constraints are non-relaxable, i.e. their values and that of the objective function cannot be computed outside of the feasible region. This situation happens frequently in practice especially in the black-box setting where function values are typically computed by means of complex simulation programs which may fail to execute if the considered point is outside of the feasible region. For such problems, we propose a new derivative-free optimization method which is based on the use of a merit function that handles inequality constraints by means of a log-barrier approach and equality constraints by means of a quadratic penalty approach. We prove convergence of the proposed method to KKT stationary points of the problem under quite mild assumptions. Furthermore, we also carry out a preliminary numerical experience on standard test problems and comparison with a state-of-the-art solver which shows efficiency of the proposed method.

keywords Derivative-free optimization, Nonlinear programming, Interior point methods

AMS 65K05, 90C30, 90C56

"Sapienza" Università di Roma, Dipartimento di Ingegneria Informatica Automatica e Gestionale "A. Ruberti," Via Ariosto 25, 00185 Rome, Italy (brilli@diag.uniroma1.it).

"Sapienza" Università di Roma, Dipartimento di Ingegneria Informatica e Sistemistica "A. Ruberti," Via Ariosto 25, 00185 Rome, Italy (liuzzi@diag.uniroma1.it).

"Sapienza" Università di Roma, Dipartimento di Ingegneria Informatica e Sistemistica "A. Ruberti," Via Ariosto 25, 00185 Rome, Italy (lucidi@diag.uniroma1.it).

Address(es) of author(s) should be given

1 Introduction

In this paper we consider the nonlinear constrained minimization problem

$$\begin{aligned} \min f(x),& \\ g(x) \leq 0,& \\ h(x) = 0,& \\ l \leq x \leq u,& \end{aligned} \tag{1}$$

where $f : \mathbb{R}^n \to \mathbb{R}$, $g : \mathbb{R}^n \to \mathbb{R}^m$, $h : \mathbb{R}^n \to \mathbb{R}^q$, and $l, u \in \mathbb{R}^n$, with $l < u$, are vectors of lower and upper bounds on the variables $x \in \mathbb{R}^n$. Furthermore, we assume that f, g and h are continuously differentiable functions even though their derivatives can be neither calculated nor explicitly approximated, and we assume that g_i, $i = 1, \dots, m$, are also convex functions. We denote by $\mathcal{S}$ the set defined by the nonlinear convex constraints and by X the set defined by simple bounds on the variables, that is,

$$\mathcal{S} = \{x \in \mathbb{R}^n : g(x) \leq 0\},$$

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

and by $\mathcal{F}$ the feasible set of problem (1), namely,

$$\mathcal{F} = \{x \in \mathbb{R}^n : h(x) = 0\} \cap \mathcal{S} \cap X.$$

Furthermore, we assume that a point $x_0 \in \overset{\circ}{\mathcal{S}}$ exists. We note that, by definition, X is a compact set so that $\mathcal{F}$ is compact as well. In many engineering problems, the values of the functions defining the objective and constraints of the problem are computed by means of complex simulation programs. For this reason, their analytic expressions are not available. Hence, derivatives are not available or, at the very least, they are untrustworthy.

Many real world applications fit into the derivative-free or black-box optimization paradigm. Such problems usually present nonlinear constraints along with bound constraints on the variables. Black-box optimization problems are widely studied in the literature (see, e.g., [2,6,11]) and many algorithms have been proposed for the solution of constrained black-box optimization problems. In particular, in [13] the use of an augmented Lagrangian function in connection with a pattern search algorithm has been proposed. In [16] a sequential penalty derivative-free linesearch approach has been studied, whereas in [15] the use of a nonsmooth exact penalty function has been proposed. A mesh adaptive direct search method, namely NOMAD, has been firstly introduced and analyzed in [1] to solve constrained black-box problems by using an extreme penalty function to manage general and hidden constraints.

According to [12,8] inequality constraints can be either relaxable or unrelaxable. Unrelaxable constraints are those constraints that must always be satisfied by the points produced by the optimization algorithm. Hence, when unrelaxable balck-box constraints are present, the optimization algorithm should take into proper account this feature. Typically, such constraints can be managed by a so-called *extreme* or *death* penalty approach (see e.g. [3]). In particular, an objective function value of $+\infty$ is assigned to points that are unfeasible with respect to one or more unrelaxable constraints. However, it should also be mentioned that such penalization strategy, by making the objective function discontinuous on the boundary of the feasible region, introduces many difficulties and ill-conditioning in the problem. As a result, solving the problem could become impractical or, at the very least, the computed solution could be far away from the real solution point.

A possible way of handling the above mentioned difficulty, consists in the use of some sort of interior penalization that modifies the landscape of the objective function in the interior of the feasible region by adding to the objective function terms that gradually tend to $+\infty$ as the points approach the boundary of the feasible region (see e.g. [10, 7, 19]).

The paper is organized as follows. In Section 2, we introduce some notation and preliminary results that will be used in the paper. Section 3 is devoted to the definition and description of the proposed derivative-free algorithm. In section 4, convergence of the proposed method is studied. Section 5 is devoted to the numerical experimentation and comparison of the proposed method with a stato-of-the-art solver, namely NOMAD [3]. Finally, in Section 6 we draw some conclusions and possible developments for future research. The paper also has an appendix in which the more technical results (which are used in Section 4) are proved.

2 Notation and preliminary results

In this section we introduce some notation and assumptions that will be used throughout the paper.

Given a vector $v \in \mathbb{R}^n$, a subscript will be used to denote either one of its components (v_i) or the fact that it is an element of an infinite sequence of vectors (v_k). To avoid possible misunderstanding or ambiguities, the ith component of a vector will be denoted by $(v)_i$. We denote by v^j the generic jth element of a finite set of vectors. Given two vectors $a, b \in \mathbb{R}^n$, we denote by $y = \max\{a, b\}$ the vector such that $y_i = \max\{a_i, b_i\}$, $i = 1, \dots, n$. Furthermore, given a vector v, we denote $v^+ = \max\{0, v\}$.

Definition 1 (cone of feasible directions) Given a point $x \in X$, let

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

be the cone of feasible directions at x with respect to the simple bound constraints.

Let $L(x, \lambda)$ be the Lagrangian function associated with the nonlinear constraints of problem (1),

$$L(x, \lambda) = f(x) + \lambda^T g(x) + \mu^T h(x)$$

We recall the Mangasarian–Fromovitz constraint qualification (MFCQ).

Definition 2 A point $x \in X$ is said to satisfy the MFCQ if two conditions are satisfied:

(a) There does not exist a nonzero vector $\alpha = (\alpha_1, ..., \alpha_q)$ such that:

$$\left(\sum_{i=1}^{q} \alpha_i \nabla h_i(x) \right)^T d \geq 0, \qquad \forall d \in D(x), \tag{2}$$

(b) there exists a feasible direction $d \in D(x)$:

$$\nabla g_l(x)^T d < 0 \quad \forall l \in I(x), \quad \nabla h_j(x)^T d = 0 \quad \forall j = 1, ..., q \tag{3}$$

where $I(x) = \{i : g_i(x) \geq 0\}$.

The following proposition is a well-known result (see, for instance, [4]) which states necessary optimality conditions for problem (1).

Proposition 1 *Let $x^\star \in \mathcal{F}$ be a local minimum of problem* (1)*. Then, there exists a vectors $\lambda^\star \in \mathbb{R}^m$, $\mu^\star \in \mathbb{R}^q$ such that*

$$\nabla_x L(x^\star, \lambda^\star, \mu^\star)^T (x - x^\star) \geq 0 \qquad \forall x \in X, \tag{4}$$

$$(\lambda^\star)^T g(x^\star) = 0, \qquad \lambda^\star \geq 0. \qquad \square \tag{5}$$

Definition 3 (stationary point) A point $x^\star \in \mathcal{F}$ is said to be a stationary point for problem (1) if a vector $\lambda^\star \in R^m$ exists such that (4) and (5) are satisfied.

Now we recall two results from [14] and [16] concerning the set $D(x)$.

Proposition 2 *Let $\{x_k\}$ be a sequence of points such that $x_k \in X$ for all k. Assume further that $x_k \to \bar{x}$ for $k \to \infty$. Then, given any direction $\bar{d} \in D(\bar{x})$, there exists a scalar $\bar{\beta} > 0$ such that, for sufficiently large k, we have*

$$x_k + \beta \bar{d} \in X \quad \forall \beta \in [0, \bar{\beta}].$$

Hence, given a sequence $\{x_k\} \subset X$ such that $x_k \to \bar{x}$ for $k \to \infty$, it results $D(\bar{x}) \subseteq D(x_k)$ for k sufficiently large.

Now we define the set of unit vectors

$$D = \{\pm e^1, \ldots, \pm e^n\},$$

where e^i, $i = 1, \ldots, n$, is the ith unit coordinate vector.

The following proposition shows that set D contains the generators of the cone of feasible directions $D(x)$ at any point $x \in X$.

Proposition 3 *Let $x \in X$. We have*

$$cone\{D \cap D(x)\} = D(x). \tag{6}$$

3 The LOG-DFL algorithm

In this section, we introduce the log-barrier penalty function [9]

$$P(x;\epsilon) = f(x) - \epsilon\sum_{j=1}^{m} \log\left[-g_j(x)\right] + \frac{1}{\epsilon}\sum_{j=1}^{q} |h_j(x)|^{\nu},$$

where $\nu > 1$ and only the nonlinear constraints have been taken into account. We assume that, $P(x;\epsilon) = +\infty$, for all $x \in \mathbb{R}^n$ such that $g(x) \not< 0$.

Then, we consider the problem [5,4]

$$\begin{array}{l} \min\ P(x;\epsilon) \\ s.t.\ x \in \overset{\circ}{\mathcal{S}} \cap X \end{array} \tag{7}$$

For every fixed value of the penalty parameter ϵ, function $P(x;\epsilon)$ is continuously differentiable in $\overset{\circ}{\mathcal{S}}$ under the stated assumptions.

The minimization process of a Derivative-free methods is based on suitable sampling techniques along a set of directions that are able to convey, in the limit, sufficient knowledge of the problem functions to recover first order information. In particular, for box constrained optimization problems, suitable choice for set of directions are the unit coordinate vectors e^i, $i = 1, \dots, n$, (see Proposition 3 and [11,17])

Below, we report the scheme of the proposed algorithm using the log-barrier penalty function $P(x;\epsilon)$ just introduced.

The algorithm that we propose and that is reported below is based on an exploration of the coordinate directions d_k^i, $i = 1, \dots, n$ by means of suitable extrapolation techniques. Indeed, in the main loop of the algorithm, the coordinate directions (or their opposites) are exploited by trying to compute feasible stepsizes α_k^i that sufficiently reduce the penalty function (see e.g. steps 1.2 and 1.3).

If such a sufficient reduction cannot be achieved, the algorithm does not move the point by setting α_k^i to zero and computes new tentative stepsize for the next iteration $\tilde{\alpha}_{k+1}^i$ by reducing the current one by a constant factor.

It is worth noting that two quantities are computed during the inner for loop of the algorithm, namely

i) a "maximum stepsize" α_{max}, i.e. the maximum stepsize used by the algorithm in the entire inner for loop. We recall that this quantity can be roughly considered as a measure of stationarity for the penalty function, see e.g. [17,11];
ii) a "minimum value" for the non-relaxable inequality constraints $g_{\min}$, i.e. the smallest absolute value of the inequality constraints at any point sampled by the inner for loop.

These two quantities play a crucial role in the penalty-barrier parameter updating rule that we shall describe shortly.

At the end of the inner for loop, the algorithm checks whether the penalty-barrier parameter should be updated. Finally, the new point x_{k+1} is computed by selecting any point which is better than the one produced by the inner for loop.

As concerns the updating rule performed at step 2 of the algorithm, a few comments are in order to help better understand its meaning. First of all, let us recall that α_{max} can be thought as a measure of stationarity for the merit function

P, see e.g., [11]. Then, the algorithm updates the penalty-barrier parameter when the measure of stationarity α_{max} is smaller than the smallest between ϵ_k^p and $g_{\min}^2$. In more details, ϵ_k is reduced when both the following conditions are satisfied.

i) α_{max} is sufficiently smaller than ϵ_k;
ii) α_{max} is sufficiently smaller than $g_{\min}$.

Condition (i) requires that the measure of stationarity is better than the quality of the approximation performed by the merit function w.r.t. the constrained problem. Note that this implies that the maximum stepsize α_{max} must go to zero faster than the penalty-barrier parameter (which is required to go to zero in order for the iterate to approach a KKT point in the limit).

On the other hand, condition (ii) requires that the step size used by the algorithm is too small to drive the iterates toward the boundary of the feasible region.

Algorithm LOG-DFL.

Data. $x_0 \in X$ such that $g(x_0) < 0$, $\epsilon_0 > 0$, $\gamma > 0$, $\theta \in (0,1)$, $p > 1$, $\tilde{\alpha}_0^i > 0$, and set $d_0^i = e^i$ for $i = 1, \dots, n$.

For $k = 0, 1, 2, \dots$ **do** *(Main iteration loop)*

Set $y_k^1 = x_k$, $g_{\min} = \min_j\{|g_j(x_k)|\}$, $\alpha_{max} = 0$.
For $i = 1, \dots, n$ **do** *(Exploration of the search directions)*

Step 1.2. Compute $\hat{\alpha}_k^i \le \tilde{\alpha}_k^i$ s.t. $y_k^i + \hat{\alpha}_k^i d_k^i \in \overset{\circ}{\mathcal{S}} \cap X$
If $\hat{\alpha}_k^i > 0$, and $P(y_k^i + \hat{\alpha}_k^i d_k^i; \epsilon_k) \le P(y_k^i; \epsilon_k) - \gamma(\hat{\alpha}_k^i)^2$,
compute α_k^i and g_{min} by the
Expansion Step$(\bar{\alpha}^i, \hat{\alpha}_k^i, y_k^i, d_k^i, \gamma, g_{min}; \alpha_k^i, g_{min})$;
$\alpha_{max} = \max\{\alpha_{max}, \alpha_k^i\}$
set $\tilde{\alpha}_{k+1}^i = \alpha_k^i$, $d_{k+1}^i = d_k^i$ and go to **Step 1.5.**
Else
set $\alpha_k^+ = \hat{\alpha}_k^i$

Step 1.3. Compute $\hat{\alpha}_k^i \le \tilde{\alpha}_k^i$ s.t. $y_k^i - \hat{\alpha}_k^i d_k^i \in \overset{\circ}{\mathcal{S}} \cap X$
If $\hat{\alpha}_k^i > 0$, and $P(y_k^i - \hat{\alpha}_k^i d_k^i; \epsilon_k) \le P(y_k^i; \epsilon_k) - \gamma(\hat{\alpha}_k^i)^2$,
compute α_k^i and g_{min} by the
Expansion Step$(\bar{\alpha}^i, \hat{\alpha}_k^i, y_k^i, -d_k^i, \gamma, g_{min}; \alpha_k^i, g_{min})$;
$\alpha_{max} = \max\{\alpha_{max}, \alpha_k^i\}$
set $\tilde{\alpha}_{k+1}^i = \alpha_k^i$, $d_{k+1}^i = -d_k^i$, and go to **Step 1.5.**
Else
set $\alpha_k^- = \hat{\alpha}_k^i$

Step 1.4. Set $\alpha_k^i = 0$, $\tilde{\alpha}_{k+1}^i = \theta\tilde{\alpha}_k^i$, $\alpha_{max} = \max\{\alpha_{max}, \tilde{\alpha}_k^i\}$
$g_{\min} = \min_j\{g_{\min}, |g_j(y_k^i + \alpha_k^+ d_k^i)|, |g_j(y_k^i - \alpha_k^- d_k^i)|\}$

Step 1.5. Set $y_k^{i+1} = y_k^i + \alpha_k^i d_k^i$.

Endfor

Step 2. If $\alpha_{max} \le \min\{\epsilon_k^p,\ g_{\min}^2\}$

Then choose $\epsilon_{k+1} = \theta\epsilon_k$ **Else** set $\epsilon_{k+1} = \epsilon_k$.

Step 3. Find $x_{k+1} \in X$ such that $P(x_{k+1}; \epsilon_k) \le P(y_k^{n+1}; \epsilon_k)$.

Endfor

Expansion Step $(\bar{\alpha}, \hat{\alpha}, y, p, \gamma, g_{min}; \alpha, g_{min})$.

Data. $\delta \in (0, 1)$.
Step 1. Set $\alpha \leftarrow \hat{\alpha}$, $\alpha_\ell \leftarrow \alpha$, $\alpha_u \leftarrow +\infty$.
Step 2. **If** $\alpha_u = +\infty$ **then** set $\breve{\alpha} \leftarrow \min\{\bar{\alpha}, (\alpha/\delta)\}$
Else set $\breve{\alpha} \leftarrow \frac{\alpha_\ell + \alpha_u}{2}$
Step 3. **If** $y + \breve{\alpha} p \notin \overset{\circ}{\mathcal{S}}$ **then** set $\alpha_u \leftarrow \breve{\alpha}$
Elseif $\breve{\alpha} < \bar{\alpha}$ and $P(y + \breve{\alpha} p; \epsilon_k) \leq P(y; \epsilon_k) - \gamma \breve{\alpha}^2$ **then**
set $\alpha \leftarrow \breve{\alpha}$, $\alpha_\ell \leftarrow \alpha$
Else ($\breve{\alpha} = \bar{\alpha}$ or $P(y + \breve{\alpha} p; \epsilon_k) > P(y; \epsilon_k) - \gamma \breve{\alpha}^2$)
set $g_{\min} = \min_j \{g_{\min}, |g_j(y_k^i + \alpha_l p)|, |g_j(y_k^i + \breve{\alpha} p)|\}$
set $\alpha \leftarrow \alpha_\ell$ and **return**
Step 4. Go to Step 2.

4 Convergence analysis

This section is devoted to the analysis of th convergence properties of the proposed algorithm. To this aim, in the following proposition, we first prove that the algorithm is well defined.

Proposition 4 *The* Expansion Step *is well defined, i.e. it always returns a step* α.

Proof We proceed by contradiction and assume that the procedure infinitely cycles, i.e. the “else” branch at Step 3 is never executed. Then, the procedure will generate four sequences $\{\alpha^j\}$, $\{\tilde{\alpha}^j\}$, $\{\alpha_u^j\}$ and $\{\alpha_\ell^j\}$.
First we prove that α_u is eventually updated. Indeed, if this were not the case, we would always have that $\tilde{\alpha}^j = \min\{\bar{\alpha}, \alpha^j/\delta\}$ and $\alpha^{j+1} = \tilde{\alpha}^j$. Then, since $\bar{\alpha}$ is fixed, a $\bar{\jmath}$ exists such that $\tilde{\alpha}^{\bar{\jmath}} = \bar{\alpha}$. When this happens, two possibilities can occur:

i) either $y + \tilde{\alpha}^{\bar{\jmath}} p \notin \overset{\circ}{\mathcal{S}}$, in which case the procedure will set $\alpha_u^{\bar{\jmath}+1} = \tilde{\alpha}^{\bar{\jmath}}$;
ii) or $y + \tilde{\alpha}^{\bar{\jmath}} p \in \overset{\circ}{\mathcal{S}}$, in which case the procedure will return, thus contradicting the hypothesis that it infinitely cycles.

Hence, for j sufficiently large we surely have $\alpha_u^j < +\infty$ and $\tilde{\alpha}^j = \frac{\alpha_\ell^j + \alpha_u^j}{2}$.
Now, for j sufficiently large, we have that:

$$y + \alpha_u^j p \notin \overset{\circ}{\mathcal{S}}, \quad \alpha_u^j < +\infty \tag{8}$$

$$y + \alpha_\ell^j p \in \overset{\circ}{\mathcal{S}} \tag{9}$$

$$P(y + \alpha_\ell^j p; \epsilon_k) \leq P(y; \epsilon_k) - \gamma (\alpha_\ell^j)^2 \tag{10}$$

Furthermore, for every j sufficiently large, we also know that

i) either

$$\alpha_\ell^{j+1} = \frac{\alpha_u^j + \alpha_\ell^j}{2} > \alpha_\ell^j, \text{ and } \alpha_u^{j+1} = \alpha_u^j$$

ii) or

$$\alpha_u^{j+1} = \frac{\alpha_u^j + \alpha_\ell^j}{2} < \alpha_u^j, \text{ and } \alpha_\ell^{j+1} = \alpha_\ell^j.$$

Whichever the case, for all j sufficiently large, it holds that

$$\alpha_u^{j+1} - \alpha_\ell^{j+1} = \frac{\alpha_u^j - \alpha_\ell^j}{2}.$$

Then, taking the limit for $j \to \infty$, we obtain

$$\lim_{j\to\infty} \alpha_u^j - \alpha_\ell^j = 0$$

which, considering that both sequences $\{\alpha_u^j\}$ and $\{\alpha_\ell^j\}$ are monotone and limited, implies that

$$\lim_{j\to\infty} \alpha_u^j = \lim_{j\to\infty} \alpha_\ell^j = \breve{\alpha}.$$

Then, taking the limit in (10) and considering (8) and (9), we get

$$\lim_{j\to\infty} y + \alpha_u^j p = y + \breve{\alpha} p \in \partial\mathcal{S}$$
$$P(y + \breve{\alpha} p; \epsilon_k) \leq P(y; \epsilon_k) - \gamma(\breve{\alpha})^2$$

which is a contradiction with the fact that $P(y; \epsilon_k) = +\infty$ on $\partial\mathcal{S}$. $\square$

The next proposition ensures that the updating rule of the algorithm produces a sequence of values of the penalty parameter which tends to zero. This result is of paramount importance since the parameter ϵ multiplies the log-barrier terms of the merit function.

Proposition 5 *Let $\{\epsilon_k\}$ be the sequence produced by Algorithm LOG-DFL, then*

$$\lim_{k\to\infty} \epsilon_k = 0$$

Proof By the instructions of the algorithm, $\{\epsilon_k\}$ is a monotonically non-increasing sequence of positive numbers. Hence, it is convergent to a limit $\bar{\epsilon} \geq 0$. Then, we proceed by contradiction and assume that $\bar{\epsilon} > 0$. This means that, for k sufficiently large, ϵ_k is no longer updated. Hence, we can assume that ϵ_k stays fixed, i.e. $\epsilon_k = \bar{\epsilon}$, definitely, i.e. the test at step 2 of Algorithm LOG-DFL is no longer satisfied that is

$$\alpha_{\max} > \min\{\bar{\epsilon}^p, (g_{\min})_k\}. \tag{11}$$

We first prove that, recalling $\epsilon_k = \bar{\epsilon}\ \forall k$ sufficiently large, then:

$$\lim_{k\to\infty} \tilde{\alpha}_k^i = 0 \qquad \textit{for } i = 1, \dots, n, \tag{12}$$

$$\lim_{k\to\infty} \alpha_k^i = 0 \qquad \textit{for } i = 1, \dots, n, \tag{13}$$

For every $i = 1, \dots, n$ we prove (13) by splitting the iteration sequence $\{k\}$ into two parts, K' and K''. We identify with K' those iterations where

$$\alpha_k^i = 0, \tag{14}$$

and with K'' those iterations where $\alpha_k^i \neq 0$ is produced by the Expansion Step. Then the instructions of the algorithm imply

$$P(x_{k+1}; \bar{\epsilon}) \leq P(y_k^i + \alpha_k^i d_k^i; \bar{\epsilon}) \leq P(y_k^i; \bar{\epsilon}) - \gamma(\alpha_k^i)^2 \|d_k^i\|^2 \leq P(x_k; \bar{\epsilon}) - \gamma(\alpha_k^i)^2 \|d_k^i\|^2. \tag{15}$$

Taking into account the compactness assumption on X, it follows from (15) that $\{P(x_k;\bar\epsilon)\}$ tends to a limit $\bar P$. If K' is infinite, then from (14) we trivially have that

$$\lim_{k\to\infty, k\in K'} \alpha_k^i = 0$$

If, on the other hand, K'' is an infinite subset, recalling that $\|d_k^i\| = 1$, we obtain

$$\lim_{k\to\infty, k\in K''} \alpha_k^i = 0. \tag{16}$$

Therefore, (14) and (16) imply (13).

In order to prove (12), for each $i \in \{1,\dots,n\}$ we split the iteration sequence $\{k\}$ into two parts, K_1 and K_2. We identify with K_1 those iterations where the Expansion Step has been performed using the direction d_k^i, for which we have

$$\tilde\alpha_{k+1}^i = \alpha_k^i. \tag{17}$$

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

$$\tilde\alpha_{k+1}^i \le \theta\tilde\alpha_k^i, \tag{18}$$

where $\theta \in (0,1)$.

If K_1 is an infinite subset, from (17) and (13) we get that

$$\lim_{k\to\infty, k\in K_1} \tilde\alpha_{k+1}^i = 0. \tag{19}$$

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

$$\tilde\alpha_{k+1}^i \le \theta^{(k+1-m_k)}\tilde\alpha_{m_k}^i \tag{20}$$

(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 $m_k \to \infty$ (namely, K_1 is an infinite subset) or $(k+1-m_k) \to \infty$ (namely, K_1 is finite). Hence, if K_2 is an infinite subset, (20) together with (19), or the fact that $\theta \in (0,1)$, yields

$$\lim_{k\to\infty, k\in K_2} \tilde\alpha_{k+1}^i = 0, \tag{21}$$

so that (12) is proved.

Now, recalling (11) and the fact that $\epsilon_k = \bar\epsilon$ for all k sufficiently large, we have that:

$$\lim_{k\to\infty} (g_{min})_k = 0,$$

Given the definition of $(g_{min})_k$ in the algorithm and the fact that the number of constraints m and of the variables n are both finite, indices $\bar\jmath$ and $\bar\imath$ must exist such that, either

$$(g_{min})_k = \min\{g_{\bar\jmath}(y_k^{\bar\imath}), g_{\bar\jmath}(y_k^{\bar\imath} + \alpha_k^{\bar\imath} d_k^{\bar\imath}), g_{\bar\jmath}(y_k^{\bar\imath} + \check\alpha_k^{\bar\imath} d_k^{\bar\imath})\}$$

or

$$(g_{min})_k = \min\{g_{\bar\jmath}(y_k^{\bar\imath}), g_{\bar\jmath}(y_k^{\bar\imath} + (\alpha_k^{\bar\imath})^+ d_k^{\bar\imath}), g_{\bar\jmath}(y_k^{\bar\imath} - (\alpha_k^{\bar\imath})^- d_k^{\bar\imath})\}$$

where $(\alpha_k^{\bar\imath})^- \le \tilde\alpha_k$ and $(\alpha_k^{\bar\imath})^+ \le \tilde\alpha_k$, and

$$y_k^{\bar\imath} = x_k + \sum_{\ell=1}^{\bar\imath-1} \alpha_k^\ell d_k^\ell.$$

Now, since $x_k \in X$ then, a subset of indices K''' exists such that

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

Then, we have that

$$\lim_{k\to\infty, k\in K'''} g_{\bar\jmath}(x_k) = 0$$

i.e. $\bar x \in \partial\mathcal{S}$, which is a contradiction with the fact that $\{P(x_k;\bar\epsilon)\}$ tends to a limit $\bar P$. □

We introduce the following index set

$$K = \{k: \ \epsilon_{k+1} < \epsilon_k\}. \tag{22}$$

Note that, by virtue of Proposition 5, K is an infinite index set.

The next propositions describe two technical results needed to show the convergence properties of the algorithm. The first one guarantees the convergence to zero of the sequences of the step sizes produced by the algorithm. The second one points out that, eventually, the algorihm performs suitable samplings of the merit function along all the generators of the cone of feasible directions.

Proposition 6 *Let $\{\tilde\alpha_k^i\}$ and $\{\alpha_k^i\}$ be the sequences produced by Algorithm LOG-DFL. Then,*

$$\lim_{k\to\infty} \max_{i=1,\dots,n} \{\tilde\alpha_k^i, \alpha_k^i\} = 0.$$

Proof The proof follows from the updating rule of the algorithm

$$\alpha_{max} \le \min\{\epsilon_k, g_{min}^2\},$$

and Proposition 5. □

Proposition 7 *Let $\{x_k\}$, $\{\epsilon_k\}$, and $\{y_k^i\}$, $i = 1,\dots,n+1$, be the sequences produced by Algorithm LOG-DFL and let $\{x_k\}_{\tilde K}$ be a subsequence conevrging to the point $\bar x$. Then, for $k \in \tilde K$ sufficiently large, for all $d^i \in D \cap D(\bar x)$, there exist scalars $\xi_k^i > 0$ such that*

$$y_k^i + \xi_k^i d^i \in X \cap \mathring{\mathcal{S}}, \tag{23}$$
$$P(y_k^i + \xi_k^i d^i;\epsilon_k) \ge P(y_k^i;\epsilon_k) - o(\xi_k^i), \tag{24}$$
$$\lim_{k\to\infty, k\in\tilde K} \xi_k^i = 0 \tag{25}$$
$$\lim_{k\to\infty, k\in\tilde K} \|y_k^i - x_k\| = 0 \tag{26}$$

Proof We recall that, by the instructions of Algorithm DFL, at every iteration k, the following set of directions is considered:

$$D_k = \{d_k^1, -d_k^1, \dots, d_k^n, -d_k^n\} = \{\pm e^1, \dots, \pm e^n\} = D.$$

At every iteration k, Algorithm DFL extracts information on the behavior of the penalty function along both d_k^i and $-d_k^i$.

In particular, along all d_k^i, $i = 1, \dots, n$, the algorithm identifies the following circumstances:

i) (Step 1.4 is executed) recall that $(\alpha_k^i)^+$, $(\alpha_k^i)^-$ are such that

$$(\alpha_k^i)^+ = 0, \text{ or } y_k^i+(\alpha_k^i)^+ d_k^i \in X\cap\mathcal{S}, \quad P(y_k^i+(\alpha_k^i)^+ d_k^i;\epsilon_k) \geq P(y_k^i;\epsilon_k)-\gamma((\alpha_k^i)^+)^2$$

$$(\alpha_k^i)^- = 0, \text{ or } y_k^i-(\alpha_k^i)^- d_k^i \in X\cap\mathcal{S}, \quad P(y_k^i-(\alpha_k^i)^- d_k^i;\epsilon_k) \geq P(y_k^i;\epsilon_k)-\gamma((\alpha_k^i)^-)^2$$

ii) (Expansion step is executed at Step 1.3) let us define $(\alpha_k^i)^+$ and $\breve{\alpha}_k^i$ such that

$$(\alpha_k^i)^+ = 0, \text{ or } y_k^i+(\alpha_k^i)^+ d_k^i \in X\cap\mathcal{S}, \quad P(y_k^i+(\alpha_k^i)^+ d_k^i;\epsilon_k) \geq P(y_k^i;\epsilon_k)-\gamma((\alpha_k^i)^+)^2$$

$$y_k^i - \breve{\alpha}_k^i d_k^i \in X \cap \mathcal{S}, \quad P(y_k^i - \breve{\alpha}_k^i d_k^i;\epsilon_k) \geq P(y_k^i;\epsilon_k) - \gamma(\breve{\alpha}_k^i)^2$$

iii) (Expansion step is executed at Step 1.2) let us define $\tilde{y}_k^i = y_k^i + \alpha_k^i d_k^i$. α_k^i and $\breve{\alpha}_k^i$ such that

$$y_k^i + \alpha_k^i d_k^i \in X \cap \mathcal{S}, \quad P(\tilde{y}_k^i;\epsilon_k) \geq P(\tilde{y}_k^i - \alpha_k^i d_k^i;\epsilon_k) - (-\gamma(\alpha_k^i)^2)$$

$$y_k^i + \breve{\alpha}_k^i d_k^i \in X \cap \mathcal{S}, \quad P(y_k^i + \breve{\alpha}_k^i d_k^i;\epsilon_k) \geq P(y_k^i;\epsilon_k) - \gamma(\breve{\alpha}_k^i)^2$$

Furthermore, recalling Proposition 6 , we also have that

$$\lim_{k\to\infty} \alpha_k^i = 0 \qquad \text{for } i = 1, \dots, n. \tag{27}$$

$$\lim_{k\to\infty} \tilde{\alpha}_k^i = 0 \qquad \text{for } i = 1, \dots, n. \tag{28}$$

Then, since $(\alpha_k^i)^+ \leq \tilde{\alpha}_k^i$, $(\alpha_k^i)^- \leq \tilde{\alpha}_k^i$, $\breve{\alpha}_k^i \leq \frac{\alpha_k^i}{\delta}$, we also have that

$$\lim_{k\to\infty} (\alpha_k^i)^+ = 0 \qquad \text{for } i = 1, \dots, n. \tag{29}$$

$$\lim_{k\to\infty} (\alpha_k^i)^- = 0 \qquad \text{for } i = 1, \dots, n. \tag{30}$$

$$\lim_{k\to\infty} \breve{\alpha}_k^i = 0 \qquad \text{for } i = 1, \dots, n. \tag{31}$$

By recalling the definitions of the search direction d_k^i, $i = 1, \dots, n$, we obtain

$$D \cap D(\bar{x}) \subseteq \{d_k^1, -d_k^1, \dots, d_k^n, -d_k^n\}. \tag{32}$$

Now by using (28), (32) and Proposition 2, we have that, for sufficiently large $k \in \tilde{K}$ and for all $d_k^i \in D \cap D(\bar{x})$, $(\alpha_k^i)^+ = 0$ can not happen and that, for sufficiently large $k \in \tilde{K}$ and for all $-d_k^i \in D \cap D(\bar{x})$, $(\alpha_k^i)^- = 0$ can not happen.

Let us consider all the directions $d^i \in D \cap D(\bar{x})$.

If $d^i = d_k^i$, by setting $\xi_k^i = (\alpha_k^i)^+$, and $o(\xi_k^i) = \gamma((\alpha_k^i)^+)^2$ if we are in i) or ii); otherwise by setting $\xi_k^i = \check{\alpha}_k^i$, and $o(\xi_k^i) = \gamma(\check{\alpha}_k^i)^2$ if we are in iii), for sufficiently large $k \in \tilde{K}$, we can write

$$y_k^i + \xi_k^i d^i \in X \cap \overset{\circ}{\mathcal{S}}, \tag{33}$$

$$P(y_k^i + \xi_k^i d^i; \epsilon_k) \geq P(y_k^i; \epsilon_k) - o(\xi_k^i), \tag{34}$$

On the other hand, if $d^i = -d_k^i$, (33), (34) hold, for sufficiently large $k \in \tilde{K}$, by setting $\xi_k^i = (\alpha_k^i)^-$, and $o(\xi_k^i) = \gamma((\alpha_k^i)^-)^2$ if we are in case i); by setting $\xi_k^i = \check{\alpha}_k^i$ and $o(\xi_k^i) = \gamma(\check{\alpha}_k^i)^2$ if we are in case ii); by setting $\xi_k^i = \alpha_k^i$, $y_k^i = \tilde{y}_k^i$ and $o(\xi_k^i) = -\gamma(\alpha_k^i)^2$ if we are in case iii).

Then, given the definition of the scalars ξ_k^i, we have that (25) is satisfied.

Finally, since $y_k^i = x_k + \sum_{j=1}^{i-1} \alpha_k^j d_k^j$, recalling (27), we obtain that (26) is also satisfied. □

Finally it is possible to state the main result concerning the convergence properties of the proposed algorithm.

Theorem 1 *Let $\{x_k\}$ be the sequence generated by Algorithm* DFL. *Let K be the set of indices defined in (22). Assume that every limit point of the sequence $\{x_k\}_K$ satisfies the MFCQ; then, every limit point $\bar{x}$ of the subsequence $\{x_k\}_K$ is a stationary point of problem* (1).

Proof Since $\{x_k\}_K \subseteq X$ and X is compact, the subsequence $\{x_k\}_K$ admits limit points. Let us consider one such limit point $\bar{x}$, i.e. an index set $\bar{K} \subseteq K$ exists such that

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

Let us denote $\bar{D} = D \cap D(\bar{x})$. Recalling Proposition 7 we have that (23), (24), (25) and (26) hold.

By applying the mean-value theorem to (24), we can write

$$-o(\xi_k^i) \leq P(y_k^i + \xi_k^i d^i; \epsilon_k) - P(y_k^i; \epsilon_k) = \xi_k^i \nabla P(u_k^i; \epsilon_k)^T d^i \qquad \forall d^i \in \bar{D},$$

where $u_k^i = y_k^i + t_k^i \xi_k^i d^i$, with $t_k^i \in (0,1)$. By convexity of $\mathcal{S}$, $u_k^i \in int(\mathcal{S})$. Thus, we have

$$-\frac{o(\xi_k^i)}{\xi_k^i} \leq \nabla P(u_k^i; \epsilon_k)^T d^i \qquad \forall d^i \in \bar{D}.$$

By considering the expression of $P(x;\epsilon)$, we can write

$$\begin{aligned}\nabla P(u_k^i; \epsilon_k)^T d^i = & \Bigg(\nabla f(u_k^i) + \sum_{l=1}^m \frac{\epsilon_k}{-g_l(x)} \nabla g_l(u_k^i) \\ & + \sum_{j=1}^q \frac{\nu}{\epsilon_k} |h_j(u_k^i)|^{\nu-1} \nabla h_j(u_k^i)\Bigg)^T d^i \geq -\frac{o(\xi_k^i)}{\xi_k^i} \qquad \forall d^i \in \bar{D}.\end{aligned} \tag{35}$$

Recalling that $u_k^i = y_k^i + t_k^i \xi_k^i d^i$, with $t_k^i \in (0,1)$, we have that, for all i such that $d^i \in \bar{D}$,

$$\lim_{k\to\infty, k\in\bar{K}} u_k^i = \bar{x}. \tag{36}$$

Now it is possible to define the following approximations of the multipliers. For $l = 1, \dots, m$ set

$$\lambda_l(x;\epsilon) = \frac{\epsilon}{-g_l(x)}$$

For $j = 1, \dots, q$ set

$$\mu_j(x;\epsilon) = \frac{\nu}{\epsilon}\ |h_j(x)|^{\nu-1}$$

The sequences $\{\lambda_l(x_k;\epsilon_k)\}_K$, $l = 1, \dots, m$, and $\{\mu_j(x_k;\epsilon_k)\}_K$, $j = 1, \dots, q$ are bounded. The proof of this property is rather technical and, to simplify the exposition, it is reported in the appendix as Proposition 8.

Then there exists a subset of $\bar{K}$, which we relabel again $\bar{K}$, such that

$$\lim_{k\to\infty, k\in\bar{K}} \lambda_l(x_k;\epsilon_k) = \bar{\lambda}_l \geq 0, \qquad l = 1, \dots, m,$$

$$\lim_{k\to\infty, k\in\bar{K}} \mu_j(x_k;\epsilon_k) = \bar{\mu}_j \geq 0, \qquad j = 1, \dots, q,$$

where $\bar{\lambda}_l = 0$ for $l \notin I(\bar{x})$.

Since $y_k^i \in \mathcal{S}$ and by the closedness of $\mathcal{S}$, any accumulation point of $\{y_k^i\} \in \mathcal{S}$. We consider now the sequence of positive penalty parameters ϵ_k. By Proposition 5, we have that:

$$\lim_{k\to\infty} \epsilon_k = 0$$

recalling assumption (i), recalling the continuity assumptions of multipliers, multiplying (35) by ϵ_k and taking the limit, we have:

$$\left(\sum_{j=1}^{p} \nu |h_j(\bar{x})|^{\nu-1} \nabla h_j(\bar{x})\right)^T d^i \geq 0 \qquad \forall d^i \in \bar{D}. \tag{37}$$

Since $\bar{x}$ satisfies MFCQ, by (2), it must result:

$$h_j(\bar{x}) = 0 \qquad \forall j = 1, \dots, p.$$

Therefore the point $\bar{x}$ is feasible.

By simple manipulations, (35) can be rewritten as

$$\begin{aligned}\Bigg(&\nabla f(u_k^i) + \sum_{l=1}^{m} \nabla g_l(u_k^i) \lambda_l(x_k;\epsilon_k) \\ &+ \sum_{l=1}^{m} \nabla g_l(u_k^i) \left(\lambda_l(u_k^i;\epsilon_k) - \lambda_l(x_k;\epsilon_k)\right) + \sum_{j=1}^{q} \nabla h_j(u_k^i) \mu_j(x_k;\epsilon_k) \\ &+ \sum_{j=1}^{1} \nabla h_j(u_k^i) \left(\mu_j(u_k^i;\epsilon_k) - \mu_j(x_k;\epsilon_k)\right)\Bigg)^T d^i \geq -\frac{o(\xi_k^i)}{\xi_k^i} \qquad \forall i : d^i \in \bar{D}.\end{aligned} \tag{38}$$

Taking the limits for $k \to \infty$ and $k \in \bar{K}$ in relation (38) and recalling (44) and (49) from the proof of Proposition 8 previously invoked, we obtain

$$\left(\nabla f(\bar{x}) + \sum_{l=1}^{m} \nabla g_l(\bar{x}) \bar{\lambda}_l + \sum_{j=1}^{q} \nabla h_j(\bar{x}) \bar{\mu}_j\right)^T d^i \geq 0 \qquad \forall i : d^i \in \bar{D}.$$

Recalling that $\bar{D} = D \cap D(\bar{x})$, from Proposition 3 we get

$$\nabla L(\bar{x}, \bar{\lambda}, \bar{\mu})^T d \geq 0 \qquad \forall d \in D(\bar{x}),$$

which concludes the proof. □

5 Numerical experiments

In this section we report the numerical performance of the proposed log-barrier derivative-free Algorithm LOG-DFL on a set of test problems.

The proposed method has been implemented in Python, and all the experiments have been conducted by choosing the following values for the parameters defining Algorithm LOG-DFL: $\gamma = 10^{-6}$, $\theta = 0.5$, $p = 2$, $\nu = 1.1$,

$$\tilde{\alpha}_0^i = \max\left\{10^{-3}, \min\{1, |(x_0)^i|\}\right\}, \qquad i = 1, \dots, n.$$

Concerning the parameter ϵ_k, in the implementation of Algorithm LOG-DFL we use a vector of parameters $\epsilon \in \mathbb{R}^m$ and choose, for $j = 1, \dots, m$,

$$(\epsilon_0)^j = \begin{cases} 10^{-3} & \text{if } 0 < g_j(x_0)^+ < 1, \\ \min\{10^{-1}, |\log(-g_j(x_0))|^{-1}\} & \text{if } g_j(x_0) < -10^{-12} \text{ and } g_j(x_0) \neq -1 \\ 10^{-1} & \text{otherwise,} \end{cases} \tag{39}$$

In order to preserve all the theoretical results, the test at Step 2 of Algorithm LOG-DFL, $\alpha_{\max} \leq \min\{\epsilon_k^p, (g_{\min})_k^2\}$, has been replaced by

$$\alpha_{\max} \leq \min\left\{\max_{i=1,\dots,m}\{(\epsilon_k)^i\}^p, (g_{\min})_k^2\right\}.$$

As termination criterion, we stop the algorithm whenever $\alpha_{\max} \leq 10^{-12}$. Finally, we allow a maximum of 20000 function evaluations.

5.1 Test problem collection

In this subsection we report the set of constrained test problems selected from the CUTEst collection. In particular, we selected all the problems with $n \leq 50$ variables and having at least one inequality constraint for which the provided initial point is strictly feasible, i.e. such that at least an index j exists with $g_j(x_0) < 0$ (the constraints such that $g_j(x_0) \geq 0$ are taken into account by an exterior penalty term). This gives us a total of 96 problems.

In figure 1 we report the cumulative distributions, respectively, of the number of variables and of the proportion of strictly satisfied constraints with respect to the total number of constraints, i.e.

$$D(\alpha) = \frac{1}{N} \left|\{p \in \mathcal{P} : \ n_p \leq \alpha\}\right|$$

$$M(\alpha) = \frac{1}{N} \left|\left\{p \in \mathcal{P} : \ \frac{\bar{m}_p}{m_p} \leq \alpha\right\}\right|$$

where

Problem	n_p	m_p	$\bar{m}_p$
ANTWERP	27	10	2
DEMBO7	16	21	16
ERRINBAR	18	9	1
HS117	15	5	5
HS118	15	29	28
LAUNCH	25	29	20
LOADBAL	31	31	20
MAKELA4	21	40	20
MESH	33	48	17
OPTPRLOC	30	30	28
RES	20	14	2
SYNTHES2	11	15	1
SYNTHES3	17	23	1
TENBARS1	18	9	1
TENBARS4	18	9	1
TRUSPYR1	11	4	1
TRUSPYR2	11	11	8
HS12	2	1	1
HS13	2	1	1
HS16	2	2	2
HS19	2	2	1
HS20	2	3	3
HS21	2	1	1
HS23	2	5	4
HS30	3	1	1
HS43	4	3	3
HS65	3	1	1
HS74	4	5	2
HS75	4	5	2
HS83	5	6	5
HS95	6	4	3
HS96	6	4	3
HS97	6	4	2
HS98	6	4	2
HS100	7	4	4
HS101	7	6	2
HS104	8	6	3
HS105	8	1	1
HS113	10	8	8
HS114	10	11	8
HS116	13	15	10
S365	7	5	2
ALLINQP	24	18	9
BLOCKQP1	35	16	1
BLOCKQP2	35	16	1
BLOCKQP3	35	16	1
BLOCKQP4	35	16	1
BLOCKQP5	35	16	1

Problem	n_p	m_p	$\bar{m}_p$
CAMSHAPE	30	94	90
CAR2	21	21	5
CHARDIS1	28	14	13
EG3	31	90	60
GAUSSELM	29	36	11
GPP	30	58	58
HADAMARD	37	93	36
HANGING	15	12	8
JANNSON3	30	3	2
JANNSON4	30	2	2
KISSING	37	78	32
KISSING1	33	144	113
KISSING2	33	144	113
LIPPERT1	41	80	64
LIPPERT2	41	80	64
LUKVLI1	30	28	28
LUKVLI10	30	28	14
LUKVLI11	30	18	3
LUKVLI12	30	21	6
LUKVLI13	30	18	3
LUKVLI14	30	18	18
LUKVLI15	30	21	7
LUKVLI16	30	21	13
LUKVLI17	30	21	21
LUKVLI18	30	21	21
LUKVLI2	30	14	7
LUKVLI3	30	2	2
LUKVLI4	30	14	4
LUKVLI6	31	15	15
LUKVLI8	30	28	14
LUKVLI9	30	6	6
MANNE	29	20	10
MOSARQP1	36	10	10
MOSARQP2	36	10	10
NGONE	29	134	106
NUFFIELD	38	138	28
OPTMASS	36	30	6
POLYGON	28	119	94
POWELL20	30	30	15
READING4	30	60	30
SINROSNB	30	58	29
SVANBERG	30	30	30
VANDERM1	30	59	29
VANDERM2	30	59	29
VANDERM3	30	59	29
VANDERM4	30	59	29
YAO	30	30	1
ZIGZAG	28	30	5

Table 1 Set of test problems selected from the CUTEst collection. n_p, m_p, and $\bar{m}_p$ denote, respectively, the number of variables, of constraints and of strictly feasible inequality constraints for the given problem.

- $\mathcal{P}$ is the set of problems;
- $N = |\mathcal{P}|$;
- n_p is the number of variables of problem $p \in \mathcal{P}$;
- m_p is the number of constraints of problem $p \in \mathcal{P}$;
- $\bar{m}_p$ is the number of strictly satisfied inequality constraints at the initial point for problem $p \in \mathcal{P}$.

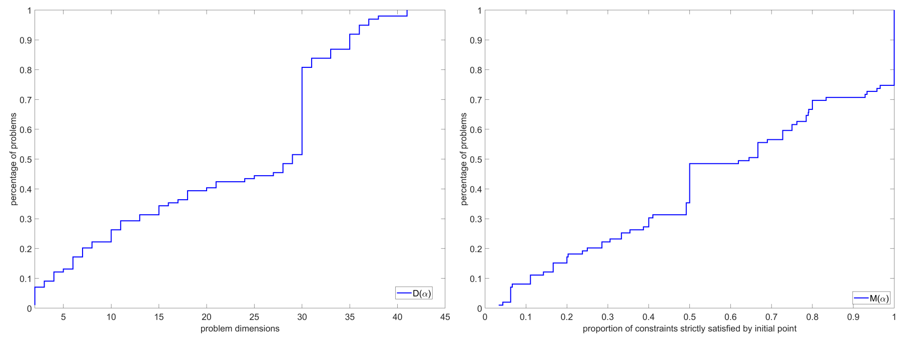

Fig. 1 Cumulative distributions, respectively, of the number of variables and of the proportion of strictly satisfied constraints with respect to the total number of constraints.

5.2 Results and comparison

In this subsection, we report the results of the comparison between our algorithm and the well-known NOMAD package (version 3.9.1) [3]. NOMAD has been run using its default settings (in particular the use of quadratic models is enabled) except that we specified for the type of constraint g_j, $j = 1, \dots, m$,

$$\texttt{EB} \text{ if } g_j(x_0) < 0, \quad \texttt{PEB} \text{ otherwise.}$$

The results are in terms of performance and data profiles as proposed in [18]. In particular, let $\mathcal{S}$ be the set of solvers to be compared against each other. For each $s \in S$ and $p \in P$, the number of function evaluations required by algorithm s to satisfy the convergence condition on problem p is denoted as $t_{p,s}$. Given a tolerance $0 < \tau < 1$ and denoted as f_L the smallest objective function value computed by any algorithm on problem p within a given number of function evaluations, the convergence test is

$$f(x_k) \le f_L + \tau(\hat{f}(x_0) - f_L),$$

where $\hat{f}(x_0)$ is the objective function value of the worst feasible point determined by all the solvers (note that in the bound-constrained case, $\hat{f}(x_0) = f(x_0)$). The above convergence test requires the best point to achieve a sufficient reduction from the value $\hat{f}(x_0)$ of the objective function at the starting point. We set to $+\infty$ the value of the objective function at infeasible points, i.e. points that have a feasibility violation $c(x) > 10^{-4}$, where

$$c(x) = \sum_{i=1}^{m} \max\{0, g_i(x)\} + sum_{j=1}^{q} |h_j(x)|.$$

Note that the smaller the value of the tolerance τ is, the higher accuracy is required at the best point. In particular, three levels of accuracy are considered in this paper for the parameter τ, namely, $\tau \in \{10^{-1}, 10^{-3}, 10^{-5}\}$.

Performance and data profiles of solver s can be formally defined as follows

$$\rho_s(\alpha) = \frac{1}{|P|} \left| \left\{ p \in P : \frac{t_{p,s}}{\min\{t_{p,s'} : s' \in S\}} \le \alpha \right\} \right|,$$

$$d_s(\kappa) = \frac{1}{|P|} \left| \{ p \in P : t_{p,s} \le \kappa(n_p + 1) \} \right|,$$

where n_p is the dimension of problem p. While α indicates that the number of function evaluations required by algorithm s to achieve the best solution is α–times the number of function evaluations needed by the best algorithm, κ denotes the number of simplex gradient estimates, with $n_p + 1$ being the number of function evaluation required to obtain one simplex gradient. Important features for the comparison are $\rho_s(1)$, which is a measure of the efficiency of the algorithm, since it is the percentage of problems for which the algorithm s performs the best, and the height reached by each profile as the value of α or κ increases, which measures the reliability of the solver.

In Figure 2, performance and data profiles are reported. It can be noted that LOG-DFL is slightly more efficient than NOMAD. Further the two methods seem to have approximately the same robustness. As a second experiment, we selected a subset of the considered problems, namely those with a number of variables greater than or equal to 10. The results referring to this set of problems are reported in terms of performance and data profiles in Figure 3. From the latter profiles, it clearly emerges that LOG-DFL is both more efficient and robust than NOMAD.

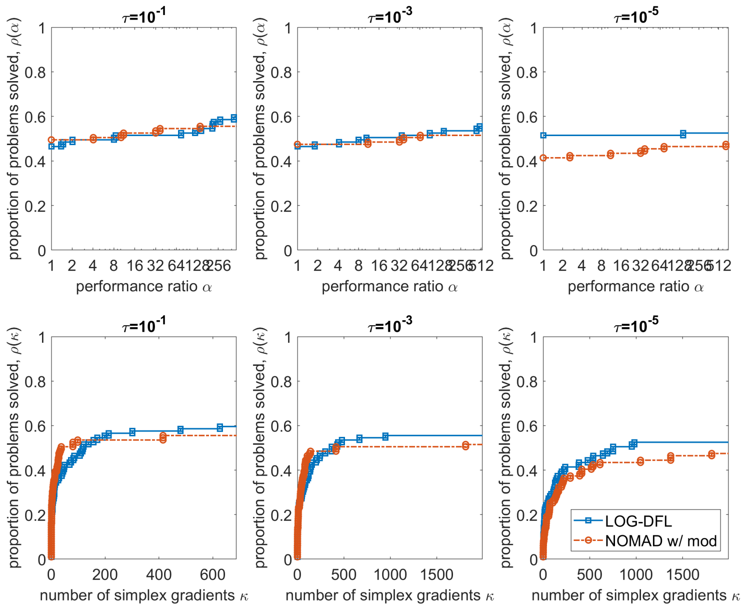

Fig. 2 Performance and data profiles for the comparison between LOG-DFL and NOMAD (3.9.1).

6 Concluding remarks

In this paper we proposed a new algorithm based on the use of a mixed penalty-barrier merit function for the solution of constrained black-box problems. In par-

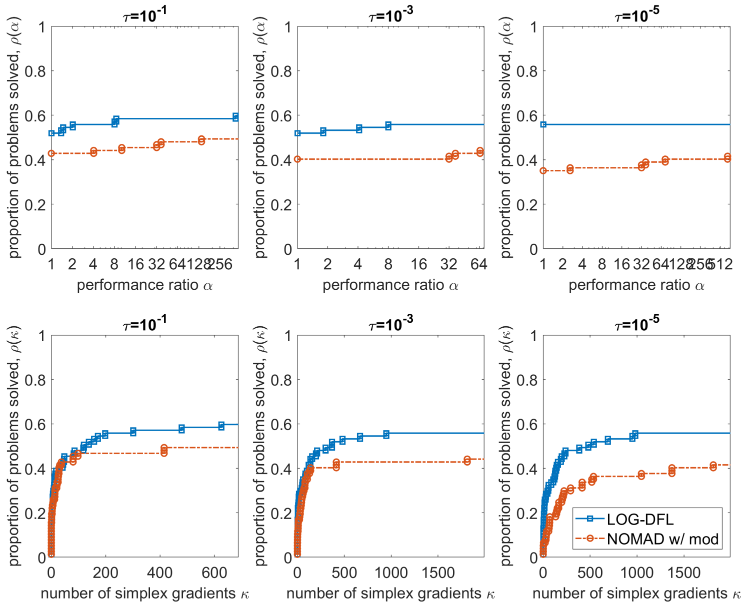

Fig. 3 Performance and data profiles for the comparison between LOG-DFL and NOMAD (3.9.1) on problems with $n_p \geq 10$.

ticular, non-relaxable inequality (convex) constraints are handled by means of a log-barrier penalization. For the proposed algorithm, we managed to prove convergence toward stationary points of the problems under quite mild assumptions. Furthermore, we also report a numerical experience and comparison with state-of-the-art solver on a large set of test problems from the CUTEst test set. We point out that the selected problems include also problems with nonconvex inequality constraints. The numerical results and comparison show that the proposed algorithm is both efficient and robust.

Finally we note that the proposed algorithm and its theoretical properties can be easily adapted to optimization problems with more complex structures than (1). In particular, inequality constraints violated at the starting point could be present and treated with the external penalization approach.

A Technical results

First we recall a result concerning a property of sequences of nonzero scalars which will be used in the proof of the next proposition.

Lemma 1 (see [16]) *Let $\{a_k^i\}$, $i = 1, \dots, p$, be sequences of nonzero scalars. There exist an index $i^\star \in \{1, \dots, p\}$ and an infinite subset $K \subseteq \{0, 1, \dots\}$ such that*

$$\lim_{k\to\infty, k\in K} \frac{a_k^i}{|a_k^{i^\star}|} = z_i, \qquad |z_i| < +\infty, \qquad i = 1, \dots, p. \tag{40}$$

Then, we report a technical result related to the behavior of Algorithm LOG-DFL which is necessary to prove boundedness of the approximations of multipliers introduced in Theroem 1.

Proposition 8 *Let the assumptions of Theroem 1 be satisfied and let K be the set of indices defined in (22). If*

$$\lambda_l(x;\epsilon) = -\frac{\epsilon}{g_l(x)}, \qquad l = 1,\dots,m.$$

$$\mu_j(x;\epsilon) = \frac{\nu}{\epsilon}\ |h_j(x)|^{\nu-1}, \qquad j = 1,\dots,q$$

then the subsequences $\{\lambda_l(x_k;\epsilon_k)\}_K$, $l = 1,\dots,m$, and $\{\mu_j(x_k;\epsilon_k)\}_K$, $j = 1,\dots,q$ are bounded.

Proof By Propositions 7, we have that (23), (24), (25) and (26) hold.

Let $\bar{x}$ be a limit point of the sequence $\{x_k\}_K$, then there exists a subset of K, which we relabel again K, such that

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

Let us denote $\bar{D} = D \cap D(\bar{x})$. By applying the mean-value theorem to (24), we can write

$$-o(\xi_k^i) \le P(y_k^i + \xi_k^i d^i;\epsilon_k) - P(y_k^i;\epsilon_k) = \xi_k^i \nabla P(u_k^i;\epsilon_k)^T d^i \qquad \forall d^i \in \bar{D},$$

where $u_k^i = y_k^i + t_k^i \xi_k^i d^i$, with $t_k^i \in (0,1)$. Thus, we have

$$-\frac{o(\xi_k^i)}{\xi_k^i} \le \nabla P(u_k^i;\epsilon_k)^T d^i \qquad \forall d^i \in \bar{D}.$$

By considering the expression of $P(x;\epsilon)$, we can write

$$\begin{aligned}
&\nabla P(u_k^i;\epsilon_k)^T d^i \\
&= \left(\nabla f(u_k^i) + \sum_{l=1}^m \frac{\epsilon_k}{-g_l(x)} \nabla g_l(u_k^i)\right. \\
&\quad \left. + \sum_{j=1}^q \frac{\nu}{\epsilon_k} \left|h_j(u_k^i)\right|^{\nu-1} \nabla h_j(u_k^i)\right)^T d^i \ge -\frac{o(\xi_k^i)}{\xi_k^i} \qquad \forall d^i \in \bar{D}.
\end{aligned} \tag{41}$$

Recalling that $u_k^i = y_k^i + t_k^i \xi_k^i d^i$, with $t_k^i \in (0,1)$, we have that

$$\lim_{k\to\infty, k\in K} u_k^i = \bar{x}. \tag{42}$$

By recalling the expression of $\lambda_l(x;\epsilon)$, $l = 1,\dots,m$, and the expression of $\mu_j(x;\epsilon)$, $j = 1,\dots,q$, we can rewrite relation (41) as

$$\begin{aligned}
&\left(\nabla f(u_k^i) + \sum_{l=1}^m \lambda_l(u_k^i;\epsilon_k) \nabla g_l(u_k^i)\right. \\
&\quad \left. + \sum_{j=1}^q \mu_j(u_k^i;\epsilon_k) \nabla h_j(u_k^i)\right)^T d^i \ge -\frac{o(\xi_k^i)}{\xi_k^i} \qquad \forall i : d^i \in \bar{D}.
\end{aligned} \tag{43}$$

First we prove that

$$\lim_{k\to\infty, k\in K} \left|\lambda_l(u_k^i;\epsilon_k) - \lambda_l(x_k;\epsilon_k)\right| = 0, \qquad l = 1,\dots,m \quad \forall\ i : d^i \in \bar{D}. \tag{44}$$

In fact,

$$\left|\frac{\epsilon_k}{-g_l(u_k^i)} - \frac{\epsilon_k}{-g_l(x_k)}\right| = \epsilon_k \left|\frac{g_l(x_k) - g_l(u_k^i)}{(-g_l(u_k^i))(-g_l(x_k))}\right| = \tag{45}$$

$$= \epsilon_k \frac{\left|\nabla g_l(\tilde{u}_k^{i,l})^T(x_k - u_k^i)\right|}{g_l(u_k^i)g_l(x_k)} \leq \epsilon_k \frac{\|\nabla g_l(\tilde{u}_k^{i,l})\|\|(u_k^i - x_k)\|}{|g_l(u_k^i)||g_l(x_k)|}$$

$$\leq c_1\epsilon_k \frac{\max_{i:d^i\in\bar{D}}\{\xi_k^i, \|y_k^i - x_k\|\}}{|g_l(u_k^i)||g_l(x_k)|}$$

$$\leq c_1\epsilon_k \frac{\max_{i:d^i\in\bar{D}}\{\xi_k^i, \|y_k^i - x_k\|\}}{\min_{i:d^i\in\bar{D}}\{|g_l(y_k^i)|, |g_l(y_k^i + \xi_k^i d^i)|\}|g_l(x_k)|} \tag{46}$$

$$\leq c_1\epsilon_k \frac{\max_{i:d^i\in\bar{D}}\{\xi_k^i, \|y_k^i - x_k\|\}}{\min_{i:d^i\in\bar{D},l}\{|g_l(y_k^i)|, |g_l(y_k^i + \xi_k^i d^i)|, |g_l(x_k)|\}^2},$$

where $\tilde{u}_k^{i,l} = u_k^i + \tilde{t}_k^{i,l} x_k$ with $\tilde{t}_k^{i,l} \in (0,1)$.

Inequality (46) follows from the convexity assumptions on $g_i(x)$, for $i = 1, \dots, m$, which implies:

$$g_l(u_k^i) \leq \max\{g_l(y_k^i), g_l(y_k^i + \xi_k^i d^i)\}$$

from which

$$|g_l(u_k^i)| \geq \min\{|g_l(y_k^i)|, |g_l(y_k^i + \xi_k^i d^i)|\}$$

Now, recalling that $y_k^i = x_k + \sum_{j=1}^{i-1} \alpha_k^j d_k^j$ and the possible choices for ξ_k^i described in Proposition 7, we can write

$$\epsilon_k \frac{\max_{i:d^i\in\bar{D}}\{\xi_k^i, \|y_k^i - x_k\|\}}{\min_{i:d^i\in\bar{D},j}\{|g_j(y_k^i)|, |g_j(y_k^i + \xi_k^i d^i), |g_j(x_k)|\}^2} \tag{47}$$
$$\leq n\epsilon_k \frac{(\alpha_{max})_k}{(g_{min})_k^2}.$$

The instructions of Step 2 imply that, for all $k \in K$,

$$(\alpha_{max})_k \leq \min\{\epsilon_k^p, (g_{\min})_k^2\} \tag{48}$$

so that

$$n\epsilon_k \frac{(\alpha_{max})_k}{(g_{min})_k^2} \leq n\epsilon_k$$

Then, (44) is proved by (45) and recalling Proposition 5.

Now, we prove that:

$$\lim_{k\to\infty, k\in K} \left|\mu_j(u_k^i; \epsilon_k) - \mu_j(x_k; \epsilon_k)\right| = 0, \qquad j = 1, \dots, q \quad \forall\, i : d^i \in \bar{D}. \tag{49}$$

In fact,

$$\left| \left|\frac{h_j(u_k^i)}{\epsilon_k}\right|^{\nu-1} - \left|\frac{h_j(x_k)}{\epsilon_k}\right|^{\nu-1} \right| \tag{50}$$

$$= \left| \left|\frac{h_j(x_k)}{\epsilon_k} + \frac{1}{\epsilon_k}\nabla h_j(\tilde{u}_k^{i,l})^T(u_k^i - x_k)\right|^{\nu-1} - \left|\frac{h_j(x_k)}{\epsilon_k}\right|^{\nu-1} \right|$$

$$\leq \left| \left|\frac{h_j(x_k)}{\epsilon_k}\right|^{\nu-1} + \left|\frac{1}{\epsilon_k}\nabla h_j(\tilde{u}_k^{i,l})^T(u_k^i - x_k)\right|^{\nu-1} - \left|\frac{h_j(x_k)}{\epsilon_k}\right|^{\nu-1} \right|$$

$$= \left|\frac{1}{\epsilon_k}\nabla h_j(\tilde{u}_k^{i,l})^T(u_k^i - x_k)\right|^{\nu-1} \leq \frac{\|\nabla h_j(\tilde{u}_k^{i,l})\|^{\nu-1}\|(u_k^i - x_k)\|^{\nu-1}}{e_k^{\nu-1}},$$

where again $\tilde{u}_k^{i,l} = u_k^i + \tilde{t}_k^{i,l}x_k$ with $\tilde{t}_k^{i,l} \in (0,1)$. Now, recalling that h_i, $i = 1,\dots,q$, are continuously differentiable functions and that $u_k^i = y_k^i + t_k^i\xi_k^i d^i$, with $t_k^i \in (0,1)$, from (50) we can write

$$\left| \left|\frac{h_j(u_k^i)}{\epsilon_k}\right|^{\nu-1} - \left|\frac{h_j(x_k)}{\epsilon_k}\right|^{\nu-1} \right| \leq c_2 \left(\frac{\max_{i:d^i\in\bar{D}}\{\xi_k^i, \|y_k^i - x_k\|\}}{\epsilon_k}\right)^{\nu-1} \tag{51}$$

Note that

$$\frac{\max_{i:d^i\in\bar{D}}\{\xi_k^i, \|y_k^i - x_k\|\}}{\epsilon_k} \leq \frac{(\alpha_{max})_k}{\epsilon_k}.$$

By the instructions of Algorithm LOG-DFL, for $k \in K$ we can write

$$(\alpha_{max})_k \leq \min\{\epsilon_k^p, (g_{\min})_k^2\}$$

that is

$$\frac{(\alpha_{max})_k}{\epsilon_k} \leq \epsilon_k^{p-1}.$$

Then, (49) is proved by (51) and recalling Proposition 5.

Now, we are ready to show that the sequences $\{\lambda_l(x_k;\epsilon_k)\}_K$, $l = 1,\dots,m$, and $\{\mu_j(x_k;\epsilon_k\}_K$, $j = 1,\dots,p$ are bounded.
In fact, by simple manipulations (43) can be rewritten as

$$\Bigg(\nabla f(u_k^i) + \sum_{l=1}^m \nabla g_l(u_k^i)\lambda_l(x_k;\epsilon_k) \tag{52}$$
$$+ \sum_{l=1}^m \nabla g_l(u_k^i)\left(\lambda_l(u_k^i;\epsilon_k) - \lambda_l(x_k;\epsilon_k)\right) + \sum_{j=1}^q \nabla h_j(u_k^i)\mu_j(x_k;\epsilon_k)$$
$$+ \sum_{j=1}^1 \nabla h_j(u_k^i)\left(\mu_j(u_k^i;\epsilon_k) - \mu_j(x_k;\epsilon_k)\right)\Bigg)^T d^i \geq -\frac{o(\xi_k^i)}{\xi_k^i} \qquad \forall i : d^i \in \bar{D}.$$

Let

$$\{a_k^1,\dots,a_k^m\} = \{\lambda_1(x_k;\epsilon_k),\dots,\lambda_m(x_k;\epsilon_k)\}.$$
$$\{a_k^{m+1},\dots,a_k^{m+q}\} = \{\mu_1(x_k;\epsilon_k),\dots,\mu_q(x_k;\epsilon_k)\}$$

By contradiction let us assume that there exists at least an index $h \in \{1,\dots,m+q\}$ such that

$$\lim_{k\to\infty, k\in K} |a_k^h| = +\infty.$$

From Lemma 1, we get that there exist an infinite subset (which we again relabel K) and an index $s \in \{1,\dots,m+q\}$ such that,

$$\lim_{k\to\infty, k\in K} \frac{a_k^i}{|a_k^s|} = z_i, \quad |z_i| < +\infty, \quad i = 1,\dots,q. \tag{53}$$

Note that

$$z_i \geq 0, \quad i \in \{1,\dots,m\}, \qquad z_s = 1 \qquad \text{and} \qquad |a_k^s| \to +\infty. \tag{54}$$

Dividing relation (52) by $|a_k^s|$, we have

$$\begin{aligned}
&\Bigg(\frac{\nabla f(u_k^i)}{|a_k^s|} + \sum_{l=1}^m \frac{\nabla g_l(u_k^i) a_k^l}{|a_k^s|} \qquad\qquad (55)\\
&\quad + \sum_{l=1}^m \nabla g_l(u_k^i) \frac{\lambda_l(u_k^i;\epsilon_k) - \lambda_l(x_k;\epsilon_k)}{|a_k^s|} + \sum_{j=1}^q \frac{\nabla h_j(u_k^i) a_k^{m+j}}{|a_k^s|}\\
&\quad + \sum_{j=1}^1 \nabla h_j(u_k^i) \frac{\mu_j(u_k^i;\epsilon_k) - \mu_j(x_k;\epsilon_k)}{|a_k^s|}\Bigg)^T d^i \geq -\frac{o(\xi_k^i)}{\xi_k^i} \qquad \forall i : d^i \in \bar{D}.
\end{aligned}$$

Taking the limits for $k \to \infty$ and $k \in K$, recalling that $|a_k^s| \to \infty$, and using (44), (49), (53), and (42), we obtain

$$\left(\sum_{l=1}^m z_l \nabla g_l(\bar{x}) + \sum_{j=1}^q z_{m+j} \nabla h_j(\bar{x})\right)^T d^i \geq 0 \qquad \forall i : d^i \in \bar{D}. \tag{56}$$

We recall that, $\bar{x}$ satisfies the MFCQ by assumption. Now, let $\hat{d} \in D(\bar{x})$ be the direction considered in Definition 2, which, from Proposition 3, can be written as

$$\hat{d} = \sum_{i:d^i\in\bar{D}} \hat{\beta}_i d^i. \tag{57}$$

Thus, from (57) and (56), we obtain

$$\begin{aligned}
&\left(\sum_{l=1}^m z_l \nabla g_l(\bar{x}) + \sum_{j=1}^q z_{m+j} \nabla h_j(\bar{x})\right)^T \hat{d} \ = \qquad (58)\\
&= \sum_{l=1}^m z_l \nabla g_l(\bar{x})^T \hat{d} + \sum_{j=1}^q z_{m+j} \nabla h_j(\bar{x})^T \hat{d} \geq 0.
\end{aligned}$$

The above relation, considering definition 2, implies

$$\sum_{l=1}^m z_l \nabla g_l(\bar{x})^T \hat{d} \geq 0. \tag{59}$$

By definition, we note that

$$z_i = 0 \qquad \text{for all} \quad i \in \{1,\dots,m\} \quad \text{and} \quad i \notin I^+(\bar{x}). \tag{60}$$

Furthermore, by Definition 2, (59) and (60) imply

$$z_i = 0 \qquad \text{for all} \quad i \in \{1,\dots,m\} \quad \text{and} \quad i \in I^+(\bar{x}). \tag{61}$$

Hence, recalling (56), (60) and (61) we have that

$$\left(\sum_{j=1}^q z_{m+j} \nabla h_j(\bar{x})\right)^T d^i \geq 0 \qquad \forall i : d^i \in \bar{D}. \tag{62}$$

By using again Definition 2, Proposition 3 and (62), we obtain

$$z_{m+j} = 0 \qquad \text{for all} \quad j \in \{1,\dots,q\}. \tag{63}$$

In conclusion we get that (59), (60) and (63) contradict (54). and this concludes the proof. ◻

References

1. Audet, C., Dennis Jr, J.E.: Mesh adaptive direct search algorithms for constrained optimization. SIAM Journal on optimization **17**(1), 188–217 (2006)
2. Audet, C., Hare, W.: Derivative-free and blackbox optimization. Springer (2017)
3. Audet, C., Le Digabel, S., Tribes, C., Montplaisir, V.: The NOMAD project
4. Bertsekas, D.P.: Nonlinear Programming. Athena Scientific Belmont, MA (1999)
5. Bertsekas, D.P.: Constrained optimization and Lagrange multiplier methods. Academic press (2014)
6. Conn, A.R., Scheinberg, K., Vicente, L.N.: Introduction to derivative-free optimization. SIAM (2009)
7. Curtis, F.E.: A penalty-interior-point algorithm for nonlinear constrained optimization. Mathematical Programming Computation **4**(2), 181–209 (2012)
8. Digabel, S.L., Wild, S.M.: A taxonomy of constraints in simulation-based optimization (2015)
9. Fiacco, A.V., McCormick, G.P.: Nonlinear programming: sequential unconstrained minimization techniques. SIAM (1990)
10. Forsgren, A., Gill, P.E., Wright, M.H.: Interior methods for nonlinear optimization. SIAM Review **44**(4), 525–597 (2002). DOI 10.1137/S0036144502414942. URL `https://doi.org/10.1137/S0036144502414942`
11. Kolda, T.G., Lewis, R.M., Torczon, V.: Optimization by direct search: New perspectives on some classical and modern methods. SIAM Review **45**, 385–482 (2003)
12. Larson, J., Menickelly, M., Wild, S.M.: Derivative-free optimization methods. Acta Numerica **28**, 287–404 (2019). DOI 10.1017/S0962492919000060
13. Lewis, R.M., Torczon, V.: A globally convergent augmented lagrangian pattern search algorithm for optimization with general constraints and simple bounds. SIAM Journal on Optimization **12**(4), 1075–1089 (2002)
14. Lin, C.J., Lucidi, S., Palagi, L., Risi, A., Sciandrone, M.: Decomposition algorithm model for singly linearly-constrained problems subject to lower and upper bounds. Journal of Optimization Theory and Applications **141**(1), 107–126 (2009)
15. Liuzzi, G., Lucidi, S.: A derivative-free algorithm for inequality constrained nonlinear programming via smoothing of an ell_∞ penalty function. SIAM Journal on Optimization **20**(1), 1–29 (2009)
16. Liuzzi, G., Lucidi, S., Sciandrone, M.: Sequential penalty derivative-free methods for nonlinear constrained optimization. SIAM Journal on Optimization **20**(5), 2614–2635 (2010)
17. Lucidi, S., Sciandrone, M.: On the global convergence of derivative-free methods for unconstrained optimization. SIAM Journal on Optimization **13**, 97–116 (2002)
18. Moré, J.J., Wild, S.M.: Benchmarking derivative-free optimization algorithms. SIAM Journal on Optimization **20**(1), 172–191 (2009)
19. Nocedal, J., Wright, S.: Numerical optimization. Springer Science & Business Media (2006)